Decorrelation of Multispectral Images, Based on Hierarchical Adaptive PCA

ROUMEN KOUNTCHEV
Department of Radio Communications
Technical University of Sofia
Bul. Kl. Ohridsky 8, Sofia 1000
BULGARIA
rkountch@tu-sofia.bg
http://www.tu-sofia.bg

ROUMIANA KOUNTCHEVA
T&K Engineering
Mladost 3
Pob 12
Sofia 1712
BULGARIA
kountcheva_r@yahoo.com

Abstract: - In this work is presented one new approach for processing of groups of multispectral (MS) images, called Hierarchical Adaptive Principal Component Analysis (HAPCA). The aim is to decorrelate each group of N multispectral images, obtained after dividing the whole set into sub-groups of 2 or 3 images each. In result, the basic part of the power of the images in one group is concentrated in a small number of eigen images only. This is achieved using the well-known method Principal Component Analysis (PCA) with transform matrix of size $N \times N$. In this case however, the method implementation needs high computational power, because it is based on iterative algorithms. Unlike it, the 2-level HAPCA permits to use transform matrices of size 3×3 (or 2×2), instead of the PCA transform matrix of size 9×9 (or 8×8 correspondingly), which makes the needed computational complexity 2 times lower in average. One more advantage of the new algorithm is that it permits parallel processing of each image sub-group in all hierarchical levels. In this work are also given some experimental results for the HAPCA algorithm applied on groups of MS images, which confirm the high decorrelation obtained. The proposed algorithm could be used as a basis for the creation of new algorithms for efficient compression of sets of MS and medical images and video sequences, for minimization of objects feature space in sequences of images, etc.

Key-Words: - Image processing, Image segmentation, Image contents analysis, Lossless image compression, Histogram modification, Inverse pyramid decomposition, Lossy image compression.

1 Introduction

The contemporary research in different application areas sets the task of the efficient processing and archiving of MS images as one of high importance. As it is well-known, MS images are characterized by very high spatial, spectral, and radiometric resolution and, hence, by ever-increasing demands of communication and storage resources. Such demands often exceed the system capacity like, for example, in the downlink from satellite to Earth stations, where the channel bandwidth is often much inferior to the intrinsic data rate of the images, some of which must be discarded altogether. In such situation the high-fidelity image compression is a very appealing alternative. As a matter of fact, there has been intense research activity on this topic [4,5,7,14,18], focusing, particularly, on transform-coding techniques, due to their good performance and limited computational complexity. Linear transform coding, however, does not take into account the nonlinear dependences existing among different bands, due to the fact that multiple land covers, each with its own interband statistics, are present in a single image. Based on this observation, a class-based coder was proposed in [4] that address the problem of interband dependences by segmenting the image into several classes, corresponding as much as possible to the different land covers of the scene. As a consequence, within each class, pixels share the same statistics and exhibit only linear interband dependences, which can be efficiently exploited by the conventional transform coding. Satellite-borne sensors have ever higher spatial, spectral and radiometric resolution. With this huge amount of information comes the problem of dealing

with large volumes of data. The most critical phase is on-board the satellite, where acquired data easily exceed the capacity of the downlink transmission channel, and often large parts of images must be simply discarded, but similar issues arise in the ground segment, where image archival and dissemination are seriously undermined by the sheer amount of data to be managed. The reasonable approach is to resort to data compression, which allows reducing the data volume by one and even two orders of magnitude without serious effects on the image quality and on their diagnostic value for subsequent automatic processing. To this end, however, is not possible to use the general purpose techniques as they do not exploit the peculiar features of multispectral remote-sensing images, which is why several ad hoc coding schemes have been proposed in recent years. The transform coding is one of the most popular approaches for several reasons. First, transform coding techniques are well established and deeply understood; they provide excellent performances in the compression of images, video and other sources, have a reasonable complexity and besides, are at the core of the famous standards JPEG and JPEG2000, implemented in widely used and easily available coders.

The common approach for coding MS images [3, 8] is to use some decorrelating transforms along the spectral dimension followed by JPEG2000 on the transform bands with a suitable rate allocation among the bands. Less attention has been devoted to techniques based on vector quantization (VQ) because, despite its theoretical optimality, VQ is too computationally demanding to be of any practical use. Nonetheless, when dealing with multiband images, VQ is a natural candidate, because the elementary semantic unit in such images is the spectral response vector (or spectrum, for short) which collects the image intensities for a given location at all spectral bands. The values of a spectrum at different bands are not simply correlated but strongly dependent, because they are completely determined (but for the noise) by the land covers of the imaged cell. This observation has motivated the search for constrained VQ techniques [13], which are suboptimal but simpler than full-search VQ, and show promising performances. MS images require large amounts of storage space, and therefore a lot of attention has recently been focused to compress these images. MS images include both spatial and spectral redundancies. Usually we can use vector quantization, prediction and transform coding to reduce redundancies. For example, hybrids transform/VQ coding scheme is proposed [13]. Instead, Karhunen-Loeve transform (KLT) is used to reduce the spectral redundancies, followed by a two-dimensional (2D) discrete cosine transform (DCT) to reduce the spatial redundancies [5]. A quad-tree technique for determining the transform block size and the quantizer for encoding the transform coefficients was applied across KLT-DCT method [18]. In [13] and [14] the researchers use a wavelet transform (WT) to reduce the spatial redundancies and KLT to reduce the spectral redundancies, and then encoded using the 3-dimensional (3D) SPIHT algorithm [9]. The state-of-the-art analysis shows that despite of the vast investigations and various techniques used for the efficient compression of MS images, a recognized general method able to solve the main problems is still not created.

One of the most efficient methods for decorrelation and compression of groups of images is based on the KLT, also known as Hotelling transform, or PCA [6, 11, 12, 24-35]. For its implementation the pixels of same spatial position in a group of N images compose an N-dimensional vector. The basic difficulty of the PCA implementation is related to the large size of the covariance matrix. For the calculation of its eigenvectors is necessary to calculate the roots of a polynomial of n^{th} degree (characteristic equation) and to solve a linear system of N equations. For large values of N, the computational complexity of the algorithm for calculation of the transform matrix is significantly increased.

One of the possible approaches for reduction of the computational complexity of PCA for N-dimensional group of images is based on the so-called “Hierarchical Adaptive PCA” (HAPCA). Unlike the famous Hierarchical PCA (HPCA) [22], this transform is not related to the image sub-blocks, but to the whole image from one group. For this, the HPCA is implemented through dividing the images into groups of length, defined by their correlation range. Each group is divided into sub-groups of 3 or 2 images each, on which is applied Adaptive PCA (APCA), of size 3×3 or *2×2* [15, 36]. This transform is performed using equations, which are not based on iterative calculations, and as a result, they have lower computational complexity. To obtain decorrelation for the whole group of images is necessary to use APCA of size *3×3* or *2×2*, which to be applied in several consecutive stages (hierarchical levels), with rearranging of the obtained intermediate eigen images after each stage. In result is obtained a decorrelated group of "eigen" images, on which could be applied other combined approaches to obtain efficient compression through lossy or lossless coding.

The paper is arranged as follows: in Section 2 is presented the principle for coding MS images through hierarchical APCA, in Section 3 are given some experimental results, in Section 4 is evaluated the computational complexity of the 2-level HAKLT, and Section 5 is the Conclusion.

2 Principle for Decorrelation of MS Images Through Hierarchical APCA

The new algorithm was developed for processing groups of MS images using adaptive PCA (APCA) with transform matrix of size *3×3* or *2×2*. The size choice is concerted with the length of the correlation range for each group, which usually contains from 2 up to 12 images.

2.1 Algorithm for Hierarchical APCA with Transform Matrix of size 3×3

The processed group of MS images (pictures) is divided into smaller groups (GOP) of (for example) 9 images each, for which is supposed that they are highly correlated. On the other hand, each GOP is further divided into 3 sub-groups.

As it is shown on Fig. 1, on each sub-group of 3 images from the first hierarchical level of HAPCA is applied APCA with matrix of size 3×3. In result are obtained 3 "eigen" images (principal components), colored in yellow, blue and green correspondingly. After that, these eigen images are rearranged so that the first sub-group of 3 eigen images to comprise the first images from each group, the second group of 3 eigen images – the second images from each group, etc. For each GOP of 9 intermediate eigen images in the first hierarchical level is applied in similar way the next APCA, with a *3×3* matrix, on each sub-group of 3 eigen values. In result are obtained 3 new eigen images (i.e. the eigen images of the group of 3 intermediate eigen images), colored in yellow, blue, and green correspondingly in the second hierarchical level. Then the eigen images are rearranged again so, that the first group of 3 eigen images to contain the first images from each group before the rearrangement; the second group of 3 eigen images – the second image before the rearrangement, etc. In result is obtained efficient decorrelation for the processed group of images, which permits efficient compression and restoration, since HAPCA is a reversible transform. For this, however, is needed the information for the transform matrix for each triad of images and for all hierarchical levels - 6 matrices for one GOP in total.

2.2. Algebraic Method for Calculation of Eigen Images Through APCA with a *3×3* Matrix

From each group of 3 MS images of *S* pixels each, shown on Fig. 2, are calculated the vectors $\vec{C}_s = [C_{1s}, C_{2s}, C_{3s}]^t$ for *s=1,2,..,S* (on the figure are shown the vectors for the first 4 pixels only: $\vec{C}_1 = [C_{11}, C_{21}, C_{31}]^t$, $\vec{C}_2 = [C_{12}, C_{22}, C_{32}]^t$, $\vec{C}_3 = [C_{13}, C_{23}, C_{33}]^t$ and $\vec{C}_4 = [C_{14}, C_{24}, C_{34}]^t$).

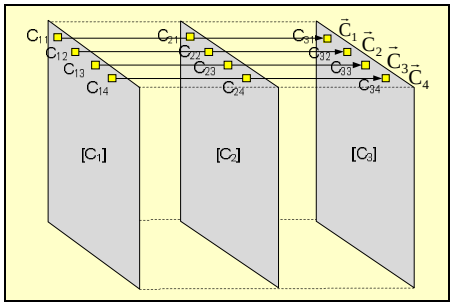


Fig. 2. Sub-group of 3 MS images from original GOP

Each vector is transformed into corresponding vectors $\vec{L}_s = [L_{1s}, L_{2s}, L_{3s}]^t$ through APCA with the matrix $[\Phi]$ of size *3×3* [15]. Its elements Φ_{ij} are defined below:

1. The covariance matrix $[K_C]$ of size *3×3* for vectors $\vec{C}_s$ is calculated:

$$[K_C] = \left[\frac{1}{S}\sum_{s=1}^{S}\vec{C}_s\vec{C}_s^t\right] - \vec{m}_c\vec{m}_c^t = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}, \quad (1)$$

where $\vec{m}_c = [\overline{C}_1, \overline{C}_2, \overline{C}_3]^t$ is the mean vector. Here $\bar{x} = E(x_s) = (1/S)\sum_{s=1}^{S} x_s$; $E(*)$ - the operator for calculation of the mean value of x_s for *s=1,2,..,S*.

2. The elements of the mean vector $\vec{m}_c$ and of the matrix $[K_C]$ are defined in accordance with the relations:

$$\overline{C}_1 = E(C_{1s}),\ \overline{C}_2 = E(C_{2s}),\ \overline{C}_3 = E(C_{3s}), \quad (2)$$

$$k_{11} = k_1 = E(C_{1s}^2) - (\overline{C}_1)^2,$$
$$k_{22} = k_2 = E(C_{2s}^2) - (\overline{C}_2)^2, \quad (3)$$
$$k_{33} = k_3 = E(C_{3s}^2) - (\overline{C}_3)^2,$$

$$k_{12} = k_{21} = k_4 = E(C_{1s}C_{2s}) - (\overline{C}_1)(\overline{C}_2),$$
$$k_{23} = k_{32} = k_6 = E(C_{2s}C_{3s}) - (\overline{C}_2)(\overline{C}_3), \quad (4)$$

$$k_{13} = k_{31} = k_5 = E(C_{1s}C_{3s}) - (\overline{C}_1)(\overline{C}_3). \quad (5)$$

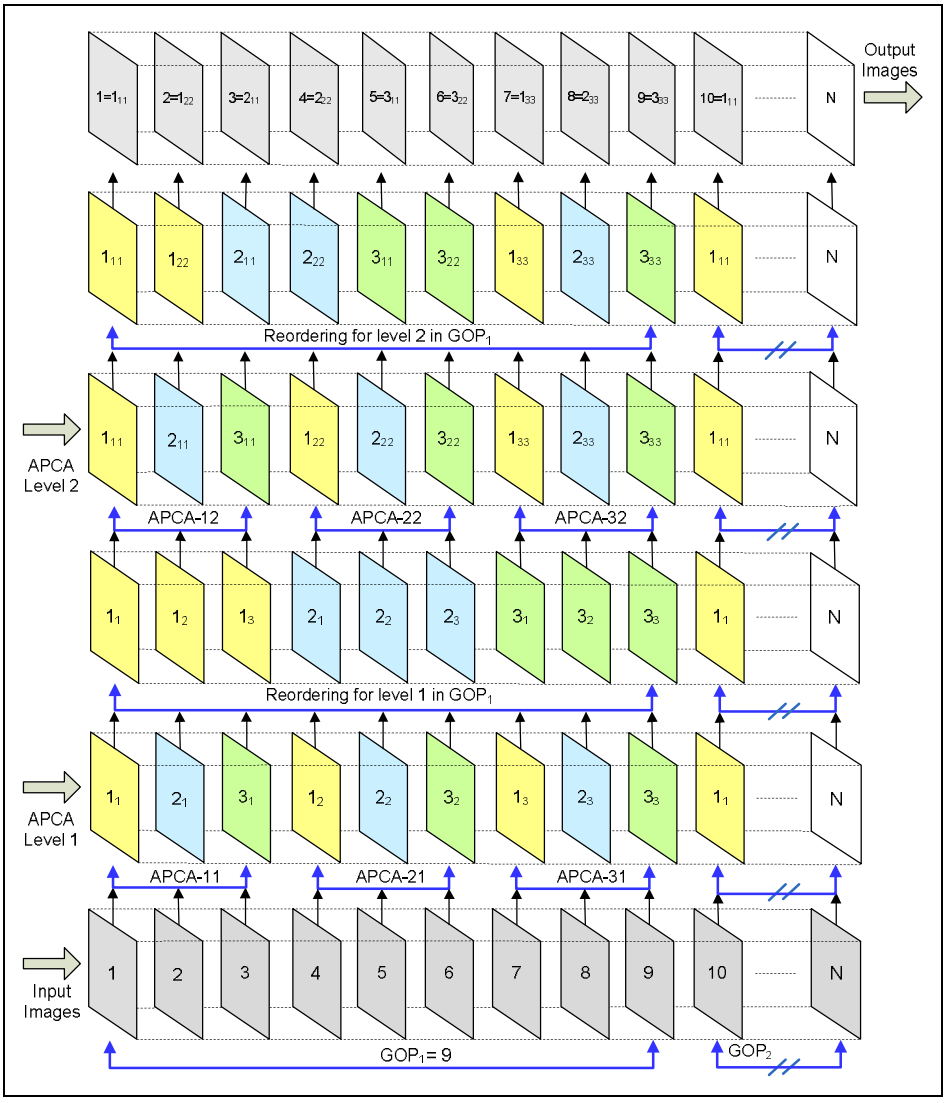


Fig. 1. Processing of a group of MS images with 2-level Hierarchical Adaptive PCA with a matrix of size *3×3*

3. The eigen values $\lambda_1, \lambda_2, \lambda_3$ of the matrix $[K_C]$ are defined in accordance to the solution of the characteristic equation:

$$\det|\, k_{ij} - \lambda\delta_{ij} \,| = \lambda^3 + a\lambda^2 + b\lambda + c = 0, \qquad (6)$$

where: $\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$

$$a = -(k_1 + k_2 + k_3),$$
$$b = k_1k_2 + k_1k_3 + k_2k_3 - (k_4^2 + k_5^2 + k_6^2), \qquad (7)$$
$$c = k_1k_6^2 + k_2k_5^2 + k_3k_4^2 - (k_1k_2k_3 + 2k_4k_5k_6).$$

Since the matrix $[K_C]$ is symmetric, its eigen values are real numbers. For their calculation could be used the equations of Cardano for "casus irreducibilis" (i.e., the so-called "trigonometric solution") [37]:

$$\lambda_1 = 2\sqrt{\frac{|p|}{3}}\cos\left(\frac{\varphi}{3}\right) - \frac{a}{3};$$
$$\lambda_2 = -2\sqrt{\frac{|p|}{3}}\cos\left(\frac{\varphi+\pi}{3}\right) - \frac{a}{3}; \qquad (8)$$
$$\lambda_3 = -2\sqrt{\frac{|p|}{3}}\cos\left(\frac{\varphi-\pi}{3}\right) - \frac{a}{3}$$

for $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$;

$$q = 2(a/3)^3 - (ab)/3 + c,$$
$$p = -(a^2/3) + b < 0. \qquad (9)$$
$$\varphi = \arccos\left[-q/2 \Big/ \sqrt{(|p|/3)^3} \right],$$

4. The eigen vectors $\vec{\Phi}_1, \vec{\Phi}_2, \vec{\Phi}_3$ of the covariance matrix $[K_C]$ are the solution of the system of equations below:

$$[K_C]\vec{\Phi}_m = \lambda_m \vec{\Phi}_m \text{ and } \left|\vec{\Phi}_m\right|^2 = \sum_{i=1}^{3} \Phi_{mi}^2 = 1, \quad (10)$$

for *m=1,2,3.*

Eq. 10 follows from the condition for orthogonality and normalization of the 3 eigenvectors:

$$\vec{\Phi}_s^t \vec{\Phi}_k = \sum_{i=1}^{3} \Phi_{is}\Phi_{ik} = \begin{cases} 1 & for\ \ s = k; \\ 0 & for\ \ s \neq k. \end{cases} \text{for } s, k = 1,2,3 \quad (11)$$

The solution of the system of equations (Eq. 10) is used to calculate the components of m^{th} eigenvector $\vec{\Phi}_m = [\Phi_{1m}, \Phi_{2m}, \Phi_{3m}]^t$, which corresponds to the eigen value λ_m:

$$\Phi_{1m} = A_m / P_m;\ \Phi_{2m} = B_m / P_m;\ \Phi_{3m} = D_m / P_m \quad (12)$$

for *m=1,2,3*;

$$\begin{aligned} A_m &= (k_3 - \lambda_m)[k_5(k_2 - \lambda_m) - k_4 k_6], \\ B_m &= (k_3 - \lambda_m)[k_6(k_1 - \lambda_m) - k_4 k_5], \end{aligned} \quad (13)$$

$$\begin{aligned} D_m &= k_6[2k_4 k_5 - k_6(k_1 - \lambda_m)] - k_5^2(k_2 - \lambda_m), \\ P_m &= \sqrt{A_m^2 + B_m^2 + D_m^2} \neq 0. \end{aligned} \quad (14)$$

The PCA matrix $[\Phi]$ comprises the eigenvectors $\vec{\Phi}_m = [\Phi_{1m}, \Phi_{2m}, \Phi_{3m}]^t$:

$$[\Phi] = \begin{bmatrix} \vec{\Phi}_1^t \\ \vec{\Phi}_2^t \\ \vec{\Phi}_3^t \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{21} & \Phi_{31} \\ \Phi_{12} & \Phi_{22} & \Phi_{32} \\ \Phi_{13} & \Phi_{23} & \Phi_{33} \end{bmatrix} \text{ for } m=1,2,3. \quad (15)$$

The **direct APCA** for vectors $\vec{C}_s = [C_{1s}, C_{2s}, C_{3s}]^t$, from which are obtained vectors $\vec{L}_s = [L_{1s}, L_{2s}, L_{3s}]^t$, is:

$$\begin{bmatrix} L_{1s} \\ L_{2s} \\ L_{3s} \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{21} & \Phi_{31} \\ \Phi_{12} & \Phi_{22} & \Phi_{32} \\ \Phi_{13} & \Phi_{23} & \Phi_{33} \end{bmatrix} \begin{bmatrix} (C_{1s} - \overline{C}_1) \\ (C_{2s} - \overline{C}_2) \\ (C_{3s} - \overline{C}_3) \end{bmatrix} \quad (16)$$

for *s=1,2,...,S*, or $\vec{L}_s = [\Phi]\vec{C}_s - [\Phi]\vec{m}_C$.

The components of vectors $\vec{L}_s = [L_{1s}, L_{2s}, L_{3s}]^t$ could be processed in various ways (such as for example: decimation and interpolation, filtration, orthogonal transforms, quantization, etc.). In result are obtained the vectors $\vec{L}_s^q = \psi(\vec{L}_s) = [\psi_1(L_{1s}), \psi_2(L_{2s}), \psi_3(L_{3s})]^t$ with components $L_{1s}^q = \psi_1(L_{1s})$, $L_{2s}^q = \psi_2(L_{2s})$, and $L_{3s}^q = \psi_3(L_{3s})$, where $\psi_1(.), \psi_2(.), \psi_3(.)$ are the functions of the used transform.

For the restoration of vectors $\vec{L}_s^q$ are used the functions for inverse transform of the components $\hat{L}_{1s} = \psi_1^{-1}(L_{1s}^q)$, $\hat{L}_{2s} = \psi_2^{-1}(L_{2s}^q)$, $\hat{L}_{3s} = \psi_3^{-1}(L_{3s}^q)$ and in result are obtained the decoded vectors $\vec{\hat{L}}_s = [\hat{L}_{1s}, \hat{L}_{2s}, \hat{L}_{3s}]^t$.

Using the **inverse APCA**, the vectors $\vec{\hat{L}}_s$ are transformed into vectors $\vec{\hat{C}}_s = [\hat{C}_{1s}, \hat{C}_{2s}, \hat{C}_{3s}]^t$:

$$\begin{bmatrix} \hat{C}_{1s} \\ \hat{C}_{2s} \\ \hat{C}_{3s} \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} \\ \Phi_{21} & \Phi_{22} & \Phi_{23} \\ \Phi_{31} & \Phi_{32} & \Phi_{33} \end{bmatrix} \begin{bmatrix} \hat{L}_{1s} \\ \hat{L}_{2s} \\ \hat{L}_{3s} \end{bmatrix} + \begin{bmatrix} \overline{\hat{C}}_1 \\ \overline{\hat{C}}_2 \\ \overline{\hat{C}}_3 \end{bmatrix} \quad (17)$$

for *s=1,2,…,S*, or $\vec{\hat{C}}_s = [\Phi]^t.\vec{\hat{L}}_s + \vec{m}_C$.

Here the matrix of the inverse APCA is:

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} & \Phi_{13} \\ \Phi_{21} & \Phi_{22} & \Phi_{23} \\ \Phi_{31} & \Phi_{32} & \Phi_{33} \end{bmatrix} = [\Phi]^{-1} = [\Phi]^t = \left[\vec{\Phi}_1, \vec{\Phi}_2, \vec{\Phi}_3\right]. \quad (18)$$

For the restoration of vectors $\vec{\hat{C}}_s = [\hat{C}_{1s}, \hat{C}_{2s}, \hat{C}_{3s}]^t$ through inverse APCA are needed not only the vectors $\vec{\hat{L}}_s = [\hat{L}_{1s}, \hat{L}_{2s}, \hat{L}_{3s}]^t$, but also the elements Φ_{ij} of the matrix $[\Phi]$, and the values of $\overline{C}_1, \overline{C}_2, \overline{C}_3$ as well. The total number of these elements could be reduced representing the matrix $[\Phi]$ as the product of matrices $[\Phi_1(\alpha)]$, $[\Phi_2(\beta)]$, $[\Phi_3(\gamma)]$, and the rotation around coordinate axes for each transformed vector in Euler angles α, β and γ correspondingly:

$$\begin{aligned} [\Phi] &= \begin{bmatrix} \Phi_{11} & \Phi_{21} & \Phi_{31} \\ \Phi_{12} & \Phi_{22} & \Phi_{32} \\ \Phi_{13} & \Phi_{23} & \Phi_{33} \end{bmatrix} = \\ &= [\Phi_1(\alpha)][\Phi_2(\beta)][\Phi_3(\gamma)] = [\Phi(\alpha, \beta, \gamma)] \end{aligned} \quad (19)$$

where

$$[\Phi_1(\alpha)] = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad [\Phi_2(\beta)] = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix};$$

$$[\Phi_3(\gamma)] = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (20)$$

In this case the elements of the matrix $[\Phi]$ are represented by the relations:

$$\begin{aligned} \Phi_{11} &= \cos\alpha\cos\beta\cos\gamma - \sin\alpha\sin\gamma; \\ \Phi_{21} &= -\cos\alpha\cos\beta\sin\gamma - \sin\alpha\cos\gamma; \\ \Phi_{31} &= -\cos\alpha\sin\beta; \end{aligned} \quad (21)$$

Φ_{12}=sin α cos β cos γ+cos α sin γ;
Φ_{22}= −sin α cos β sin γ+cos α cos γ;
Φ_{32}=−sinα sinβ; Φ_{13}=sinβ cosγ;

The matrix of the inverse APCA is defined by the relation:

$$[\Phi]^{-1}=[\Phi_3(-\gamma)][\Phi_2(-\beta)][\Phi_1(-\alpha)]. \quad (22)$$

Then, to calculate the elements of the inverse matrix $[\Phi]^{-1}$ is enough to know the values of the 3 rotation angles α, β and γ, defined by the relations:

$$\alpha=-\arcsin\left(\Phi_{32}/\sqrt{1-\Phi_{33}^2}\right);\ \beta=\arccos(\Phi_{33});$$
$$\gamma=\arccos\left(\Phi_{13}/\sqrt{1-\Phi_{33}^2}\right). \quad (23)$$

In result, the number of values needed to calculate the matrix $[\Phi]^{-1}$ is reduced from 9 down to 3, i.e. 3 times reduction. The elements L_{1s}, L_{2s}, L_{3s} for $s=1,2,...,S$ comprise the pixels of the first, second and third eigen image in the sub-group of MS images with elements C_{1s}, C_{2s}, C_{3s}.

2.3 APCA without Coordinate System Origin Shift

The famous PCA transform was initially developed for processing of digital data. When applied to images, some principal component pixel brightness could be negative due to the fact that the transformation is a simple axis rotation. As it is known, a combination of positive and negative brightness cannot be displayed, but together with this negative brightness pixels should not be ignored since their appearance relative to the other pixels in a component is needed to define details. The problem with negative values is accommodated by shifting the origin of the principal components space to yield all components with positive and thus displayable brightness. This has no effect on the transformation properties as can be seen by inserting an origin shift term in the definition of the covariance matrix in the principal components axes. Let us define:

$$\vec{L}'_s=\vec{L}_s-\vec{L}_0, \quad (24)$$

where $\vec{L}_0$ is the position of a new origin. In the new $\vec{L}'_s$ coordinates

$$[K_{L'}]=E\{(\vec{L}'_s-\vec{m}_{L'})(\vec{L}'_s-\vec{m}_{L'})^t\}, \quad (25)$$

Then, $\vec{m}_{L'}=\vec{m}_L-\vec{L}_0$, and:
$\vec{L}'_s-\vec{m}_{L'}=\vec{L}_s-\vec{L}_0-\vec{m}_L+\vec{L}_0=\vec{L}_s-\vec{m}_L$. Thus, $[K_{L'}]=[K_L]$ - i.e., the origin shift has no influence on the data covariance in the principal components axes, and can be used for convenience in displaying principal component images.

The direct and inverse APCA without coordinate system origin shift are obtained from Eqs. 16 and 17, where $\vec{L}'_s=\vec{L}_s-\vec{L}_0$ is substituted for $\vec{L}_0=[\Phi]\vec{m}_C$. Then the direct and inverse APCA are represented by the relations below:

$$\begin{bmatrix}L'_{1s}\\L'_{2s}\\L'_{3s}\end{bmatrix}=\begin{bmatrix}\Phi_{11}&\Phi_{21}&\Phi_{31}\\\Phi_{12}&\Phi_{22}&\Phi_{32}\\\Phi_{13}&\Phi_{23}&\Phi_{33}\end{bmatrix}\begin{bmatrix}C_{1s}\\C_{2s}\\C_{3s}\end{bmatrix} \text{ or } \vec{L}'_s=[\Phi].\vec{C}_s; \quad (26)$$

$$\begin{bmatrix}\hat{C}_{1s}\\\hat{C}_{2s}\\\hat{C}_{3s}\end{bmatrix}=\begin{bmatrix}\Phi_{11}&\Phi_{12}&\Phi_{13}\\\Phi_{21}&\Phi_{22}&\Phi_{23}\\\Phi_{31}&\Phi_{32}&\Phi_{33}\end{bmatrix}\begin{bmatrix}\hat{L}'_{1s}\\\hat{L}'_{2s}\\\hat{L}'_{3s}\end{bmatrix} \text{ or } \hat{\vec{C}}_s=[\Phi]^t.\hat{\vec{L}}'_s. \quad (27)$$

2.4 Evaluation of the Transformed Images Decorrelation after HAPCA with a *3×3* Matrix

For *Level 1* of HAPCA the corresponding covariance matrices of size *3×3* for each group of vectors $\vec{C}_{s,p}$ for *p=1,2,3* and *s=1,2,...,S*, are:

$$[K_L^{1,1}]=[\Phi_1^1][K_{C_{1,1}}][\Phi_1^1]^t=\begin{bmatrix}\lambda_1^{1,1}&0&0\\0&\lambda_2^{1,1}&0\\0&0&\lambda_3^{1,1}\end{bmatrix}, \quad (28)$$

$$[K_L^{1,2}]=[\Phi_2^1][K_{C_{1,2}}][\Phi_2^1]^t=\begin{bmatrix}\lambda_1^{1,2}&0&0\\0&\lambda_2^{1,2}&0\\0&0&\lambda_3^{1,2}\end{bmatrix}, \quad (29)$$

$$[K_L^{1,3}]=[\Phi_3^1][K_{C_{1,3}}][\Phi_3^1]^t=\begin{bmatrix}\lambda_1^{1,3}&0&0\\0&\lambda_2^{1,3}&0\\0&0&\lambda_3^{1,3}\end{bmatrix}. \quad (30)$$

Here

$$[K_{C_{1,p}}]=\left[\frac{1}{S}\sum_{s=1}^{S}\vec{C}_{s,p}\vec{C}^t_{s,p}\right]-\vec{m}_{c,p}\vec{m}^t_{c,p}=$$
$$=\begin{bmatrix}k_{11}(p)&k_{12}(p)&k_{13}(p)\\k_{21}(p)&k_{22}(p)&k_{23}(p)\\k_{31}(p)&k_{32}(p)&k_{33}(p)\end{bmatrix}\text{for } p=1,2,3 \quad (31)$$

Here $\lambda_1^{1,p}, \lambda_1^{1,p}, \lambda_1^{1,p}$ are the corresponding eigen values of the covariance matrices $[K_L^{1,p}]$.

The covariance matrix of size *9×9* for the 9-component vectors $\vec{C}_s$ calculated for the group of processed input images (GOP) for Level 1 of HAPCA with a 3×3 matrix, is represented as:

$$[K_L^1]=\left[\frac{1}{S}\sum_{s=1}^{S}\vec{C}_s\vec{C}_s^t\right]-\vec{m}_c\vec{m}_c^t=$$

$$=\begin{bmatrix} \lambda_1^{1,1} & 0 & 0 & & & & & & \\ 0 & \lambda_2^{1,1} & 0 & & [K_{L_1,L_2}^1] & & & [K_{L_1,L_3}^1] & \\ 0 & 0 & \lambda_3^{1,1} & & & & & & \\ & & & \lambda_1^{1,2} & 0 & 0 & & & \\ & [K_{L_1,L_2}^1] & & 0 & \lambda_2^{1,2} & 0 & & [K_{L_2,L_3}^1] & \\ & & & 0 & 0 & \lambda_3^{1,2} & & & \\ & & & & & & \lambda_1^{1,3} & 0 & 0 \\ & [K_{L_1,L_3}^1] & & & [K_{L_2,L_3}^1] & & 0 & \lambda_2^{1,3} & 0 \\ & & & & & & 0 & 0 & \lambda_3^{1,3} \end{bmatrix}, \quad (32)$$

where $[K_{L_k,L_p}^1]=E\{\vec{L}_{sk}^1.\vec{L}_{sp}^{1,t}\}-E\{\vec{L}_{sk}^1\}E\{\vec{L}_{sp}^1\}^t$ (33)

for $k,p=1,2,3$; $k\neq p$ is the mutual covariance matrix of size 3×3 for the 3-component vectors $\vec{L}_{sk}^1$ and $\vec{L}_{sp}^1$ from groups k and p in the Level 1 of HAPCA after rearrangement of the already obtained eigen images.

For the *Level 2* of HAPCA the covariance matrices of size 3×3 for each sub-group of rearranged vectors, obtained after performing the HAPCA Level 1 are:

$$[K_L^{2,1}]=\begin{bmatrix} \lambda_1^{2,1} & 0 & 0 \\ 0 & \lambda_2^{2,1} & 0 \\ 0 & 0 & \lambda_3^{2,1} \end{bmatrix}.$$

$$[K_L^{2,2}]=\begin{bmatrix} \lambda_1^{2,2} & 0 & 0 \\ 0 & \lambda_2^{2,2} & 0 \\ 0 & 0 & \lambda_3^{2,2} \end{bmatrix}. \quad (34)$$

$$[K_L^{2,3}]=\begin{bmatrix} \lambda_1^{2,3} & 0 & 0 \\ 0 & \lambda_2^{2,3} & 0 \\ 0 & 0 & \lambda_3^{2,3} \end{bmatrix}.$$

The covariance matrix of size 9×9 for the 9-component vectors $\vec{C}_s$ for the group of processed input images (GOP) for Level 2 is:

$$[K_L^2]=\begin{bmatrix} \lambda_1^{2,1} & 0 & 0 & & & & & & \\ 0 & \lambda_2^{2,1} & 0 & & [K_{L_1,L_2}^2] & & & [K_{L_1,L_3}^2] & \\ 0 & 0 & \lambda_3^{2,1} & & & & & & \\ & & & \lambda_1^{2,2} & 0 & 0 & & & \\ & [K_{L_1,L_2}^2] & & 0 & \lambda_2^{2,2} & 0 & & [K_{L_2,L_3}^2] & \\ & & & 0 & 0 & \lambda_3^{2,2} & & & \\ & & & & & & \lambda_1^{2,3} & 0 & 0 \\ & [K_{L_1,L_3}^2] & & & [K_{L_2,L_3}^2] & & 0 & \lambda_2^{2,3} & 0 \\ & & & & & & 0 & 0 & \lambda_3^{2,3} \end{bmatrix}, \quad (35)$$

$$[K_{L_k,L_p}^2]=E\{\vec{L}_{sk}^2.\vec{L}_{sp}^{2,t}\}-E\{\vec{L}_{sk}^2\}E\{\vec{L}_{sp}^2\}^t \quad (36)$$

for $k,p=1,2,3$ and $k\neq p$ is the mutual covariance matrix of size 3×3 of the 3-component vectors $\vec{L}_{sk}^2$ and $\vec{L}_{sp}^2$ of groups k and p in the Level 2 of HAPCA after rearrangement of the obtained eigen images.

Using the already obtained matrix $[K_L^2]$ could be evaluated the decorrelation of the corresponding eigen images in the processed GOP. When full decorrelation is obtained, the matrix $[K_L^2]$ is diagonal, for which should be satisfied the condition:

$$[K_{L_k,L_p}^2]=E\{\vec{L}_{sk}^2.\vec{L}_{sp}^{2,t}\}-E\{\vec{L}_{sk}^2\}E\{\vec{L}_{sp}^2\}^t=0 \quad (37)$$

for $k,p=1,2,3$; $k\neq p$.

The iterations of HAPCA could be stopped even without full decorrelation, if the condition below is satisfied:

$$|E\{\vec{L}_k^2.\vec{L}_p^2\}-E\{\vec{L}_k^2\}E\{\vec{L}_p^2\}|\geq\delta, \quad (38)$$

where δ is a pre-defined threshold.

2.5 Algebraic Method for Calculation of Eigen Images Through APCA with a *2×2* Matrix

For any 2 digital images of size $S=M\times N$ pixels each, shown on Fig. 3, are calculated the vectors $\vec{C}_s=[C_{1s},C_{2s}]^t$ for $s=1,2,..,S$ (on the figure are shown the vectors for the first 4 pixels only, resp. $\vec{C}_1=[C_{11},C_{21}]^t$, $\vec{C}_2=[C_{12},C_{22}]^t$, $\vec{C}_3=[C_{13},C_{23}]^t$ and $\vec{C}_4=[C_{14},C_{24}]^t$).

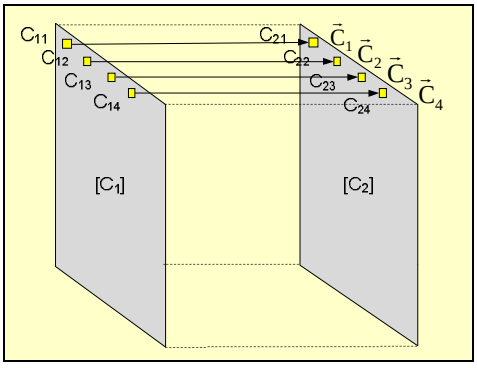


Fig. 3. Sub-group of 2 images from the original GOP

Each vector is transformed into corresponding vectors $\vec{L}_s=[L_{1s},L_{2s}]^t$ through APCA using the matrix $[\Phi]$ of size 2×2. Its' elements Φ_{ij} are defined as follows:

1. The covariance matrix $[K_C]$ of size *2×2* for vectors $\vec{C}_s$ is calculated:

$$[K_C]=\left[\frac{1}{S}\sum_{s=1}^{S}\vec{C}_s\vec{C}_s^t\right]-\vec{m}_c\vec{m}_c^t= \\ =E\{\vec{C}_s\vec{C}_s^t\}-\vec{m}_c\vec{m}_c^t=\begin{bmatrix}k_{11} & k_{12}\\ k_{21} & k_{22}\end{bmatrix}, \quad (39)$$

where $\vec{m}_c=[\overline{C}_1,\overline{C}_2]^t$ is the mean vector and $\overline{x}=E(x_s)=\frac{1}{S}\sum_{s=1}^{S}x_s$ - the mean operator.

2. The elements of the mean vector $\vec{m}_c$ and the matrix $[K_C]$ are defined in accordance with the relations:

$$k_{11}=k_1=E(C_{1s}^2)-(\overline{C}_1)^2,\ k_{22}=k_2=E(C_{2s}^2)-(\overline{C}_2)^2, \quad (40)$$

$$k_{12}=k_{21}=k_3=E(C_{1s}C_{2s})-(\overline{C}_1)(\overline{C}_2). \quad (41)$$

$$\overline{C}_1=E(C_{1s}),\ \overline{C}_2=E(C_{2s}), \quad (42)$$

3. The eigen values λ_1,λ_2 of the matrix $[K_C]$ are defined in accordance to the solution of the characteristic equation:

$$\det|k_{ij}-\lambda\delta_{ij}|=\lambda^2+(k_1+k_2)\lambda+(k_1k_2-k_3^2)=0. \quad (43)$$

Since the matrix $[K_C]$ is symmetric, its eigen values are real numbers:

$$\lambda_1=\frac{1}{2}\left[(k_1+k_2)+\sqrt{(k_1-k_2)^2+4k_3^2}\right], \\ \lambda_2=\frac{1}{2}\left[(k_1+k_2)-\sqrt{(k_1-k_2)^2+4k_3^2}\right] \quad (44)$$

4. The eigen vectors $\vec{\Phi}_1,\vec{\Phi}_2$ of the covariance matrix $[K_C]$ are the solution of the system of equations below:

$$[K_C]\vec{\Phi}_m=\lambda_m\vec{\Phi}_m \text{ and } |\vec{\Phi}_m|^2=\sum_{i=1}^{2}\Phi_{mi}^2=1, \text{ for } m=1,2. \quad (45)$$

The solution of the system of equations (Eq. 45) is used to calculate the components of m^{th} eigenvector $\vec{\Phi}_m=[\Phi_{1m},\Phi_{2m}]^t$, which corresponds to the eigen value λ_m for *m=1,2*:

$$\Phi_{11}=\frac{(k_1-k_2)+\sqrt{(k_1-k_2)^2+4k_3^2}}{\sqrt{2[(k_1-k_2)^2+4k_3^2+(k_1-k_2)\sqrt{(k_1-k_2)^2+4k_3^2}]}}= \\ =\frac{\gamma+\alpha}{\sqrt{2\gamma(\gamma+\alpha)}}; \quad (46)$$

$$\Phi_{21}=\frac{2k_3}{\sqrt{2[(k_1-k_2)^2+4k_3^2+(k_1-k_2)\sqrt{(k_1-k_2)^2+4k_3^2}]}}= \\ =\frac{\beta}{\sqrt{2\gamma(\gamma+\alpha)}}; \quad (47)$$

$$\Phi_{12}=\frac{(k_1-k_2)-\sqrt{(k_1-k_2)^2+4k_3^2}}{\sqrt{2[(k_1-k_2)^2+4k_3^2-(k_1-k_2)\sqrt{(k_1-k_2)^2+4k_3^2}]}}= \\ =\frac{\alpha-\gamma}{\sqrt{2\gamma(\gamma-\alpha)}}=-\frac{\beta}{\sqrt{2\gamma(\gamma+\alpha)}}; \quad (48)$$

$$\Phi_{22}=\frac{2k_3}{\sqrt{2[(k_1-k_2)^2+4k_3^2-(k_1-k_2)\sqrt{(k_1-k_2)^2+4k_3^2}]}}= \\ =\frac{\beta}{\sqrt{2\gamma(\gamma-\alpha)}}. \quad (49)$$

where $\alpha=k_1-k_2,\ \beta=2k_3,\ \gamma=\sqrt{\alpha^2+\beta^2}$.

The divisors in Eqs. 48 and 49 should not be equal to zero, and for this the condition $\kappa_3 \neq 0$, has to be satisfied, i.e. vectors $\vec{C}_s$ should be decorrelated.

The PCA matrix $[\Phi]$ comprises the eigenvectors $\vec{\Phi}_1=[\Phi_{11},\Phi_{21}]^t=(1/\sqrt{2\gamma})[\sqrt{\gamma+\alpha},\sqrt{\gamma-\alpha}]^t$ and $\vec{\Phi}_2=[\Phi_{12},\Phi_{22}]^t=(1/\sqrt{2\gamma})[-\sqrt{\gamma-\alpha},\sqrt{\gamma+\alpha}]^t$.

Then the matrix $[\Phi]$ is defined by the relation:

$$[\Phi]=\begin{bmatrix}\vec{\Phi}_1^t\\ \vec{\Phi}_2^t\end{bmatrix}=\begin{bmatrix}\Phi_{11} & \Phi_{21}\\ \Phi_{12} & \Phi_{22}\end{bmatrix}= \\ =\frac{1}{\sqrt{2}}\begin{bmatrix}\sqrt{\frac{\gamma+\alpha}{\gamma}} & \sqrt{\frac{\gamma-\alpha}{\gamma}}\\ -\sqrt{\frac{\gamma-\alpha}{\gamma}} & \sqrt{\frac{\gamma+\alpha}{\gamma}}\end{bmatrix} \quad (50)$$

5. The direct APCA for vectors $\vec{C}_s=[C_{1s},C_{2s}]^t$, from which are obtained vectors $\vec{L}_s=[L_{1s},L_{2s}]^t$, is:

$$\begin{bmatrix}L_{1s}\\ L_{2s}\end{bmatrix}=\begin{bmatrix}\Phi_{11} & \Phi_{21}\\ \Phi_{12} & \Phi_{22}\end{bmatrix}\begin{bmatrix}(C_{1s}-\overline{C}_1)\\ (C_{2s}-\overline{C}_2)\end{bmatrix} \quad (51)$$

for *s=1,2,...,S*.

6. In result of the inverse APCA, vectors $\vec{L}_s$ are transformed into vectors $\vec{C}_s=[C_{1s},C_{2s}]^t$:

$$\begin{bmatrix}C_{1s}\\ C_{2s}\end{bmatrix}=\begin{bmatrix}\Phi_{11} & \Phi_{12}\\ \Phi_{21} & \Phi_{22}\end{bmatrix}\begin{bmatrix}L_{1s}\\ L_{2s}\end{bmatrix}+\begin{bmatrix}\overline{C}_1\\ \overline{C}_2\end{bmatrix} \quad (52)$$

for *s=1,2,…,S*.

Here the matrix of the inverse APCA is:

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} = [\Phi]^{-1} = [\Phi]^t = [\vec{\Phi}_1, \vec{\Phi}_2] =$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{\frac{\gamma+\alpha}{\gamma}} & -\sqrt{\frac{\gamma-\alpha}{\gamma}} \\ \sqrt{\frac{\gamma-\alpha}{\gamma}} & \sqrt{\frac{\gamma+\alpha}{\gamma}} \end{bmatrix}. \quad (53)$$

7. The elements Φ_{ij} of the matrix $[\Phi]$ are a function of the angle θ on which is rotated the coordinate system (Φ_1, Φ_2) in respect to the original, (C_1, C_2), which follows from the APCA transform. The matrix $[\Phi]$ could be represented by the relation:

$$[\Phi(\theta)] = \begin{bmatrix} \Phi_{11}(\theta) & \Phi_{21}(\theta) \\ \Phi_{12}(\theta) & \Phi_{22}(\theta) \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}, \quad (54)$$

where

$$\theta = arctg\left(\frac{\Phi_{21}(\theta)}{\Phi_{11}(\theta)}\right) = arctg\left(\frac{\beta}{\alpha+\gamma}\right) \quad (55)$$

In particular, if $\alpha = 0$ *(for* $k_1 = k_2$*)* from Eq. 51 follows, that $\theta = arctg(1) = \pi/4$ and then $\cos\theta = \sin\theta = 1/\sqrt{2}$; the corresponding transform matrix is:

$$[\Phi(\pi/4)] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}.$$

From Eqs. (46–49 and 51) for *s=1,2,...,S* are obtained the relations:

$$L_{1s} = (1/\sqrt{2\gamma})[(\sqrt{\gamma+\alpha})(C_{1s} - \overline{C}_1) + (\sqrt{\gamma-\alpha})(C_{2s} - \overline{C}_2)], \quad (56)$$

$$L_{2s} = (1/\sqrt{2\gamma})[-(\sqrt{\gamma-\alpha})(C_{1s} - \overline{C}_1) + (\sqrt{\gamma+\alpha})(C_{2s} - \overline{C}_2)], \quad (57)$$

where L_{1s} and L_{2s} are the pixels of the first and second eigen image correspondingly.

Accordingly, from Eqs. (46-49 and 52) for *s=1,2,..,S* are obtained the pixels of both restored images:

$$C_{1s} = (1/\sqrt{2\gamma})[(\sqrt{\gamma+\alpha})L_{1s} - (\sqrt{\gamma-\alpha})L_{2s}] + \overline{C}_1, \quad (58)$$

$$C_{2s} = (1/\sqrt{2\gamma})[(\sqrt{\gamma-\alpha})L_{1s} + (\sqrt{\gamma+\alpha})L_{2s}] + \overline{C}_2. \quad (59)$$

From Eqs. (56-59) it is easy to notice that for the successful performance of the direct and inverse AKLT should be known the values of the parameters α, β, (or of the angle θ, and $\overline{C}_1, \overline{C}_2$, respectively), i.e. 3 numbers for one transform altogether.

When the coordinate system is right-handed, and the rotated on angle θ system – left-handed (the case of rotation with reflection), the rotation matrix of the direct KLT is of the kind:

$$[\Phi(\theta)] = \begin{bmatrix} \Phi_{11}(\theta) & \Phi_{21}(\theta) \\ \Phi_{12}(\theta) & \Phi_{22}(\theta) \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}. \quad (60)$$

For $\theta = \pi/4$ this matrix is $[\Phi(\pi/4)] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and coincides with the Hadamard matrix of size *2×2*. For the case when this matrix is used in Eqs. (56-59) the signs of the weight elements should be changed in accordance with the elements of the matrix from Eq. 60. In this case, the direct and inverse AKLT are defined by the relations below:

$$L_{1s} = (1/\sqrt{2\gamma})[(\sqrt{\gamma+\alpha})(C_{1s} - \overline{C}_1) + (\sqrt{\gamma-\alpha})(C_{2s} - \overline{C}_2)], \quad (61)$$

$$L_{2s} = (1/\sqrt{2\gamma})[(\sqrt{\gamma-\alpha})(C_{1s} - \overline{C}_1) - (\sqrt{\gamma+\alpha})(C_{2s} - \overline{C}_2)], \quad (62)$$

$$C_{1s} = (1/\sqrt{2\gamma})[(\sqrt{\gamma+\alpha})L_{1s} + (\sqrt{\gamma-\alpha})L_{2s}] + \overline{C}_1, \quad (63)$$

$$C_{2s} = (1/\sqrt{2\gamma})[(\sqrt{\gamma-\alpha})L_{1s} - (\sqrt{\gamma+\alpha})L_{2s}] + \overline{C}_2. \quad (64)$$

In order to avoid negative values for the components of vectors $\vec{L}_s = [L_{1s}, L_{2s}]^t$ they should be transformed in advance, in accordance to relations: $\vec{L}'_s = \vec{L}_s - \vec{L}_0$ ($\vec{L}_0 = [\Phi(\theta)].\vec{m}_C = [\Phi(\theta)].[\overline{C}_1, \overline{C}_2]^t$). As a result Eqs. (56-59) become:

$$L'_{1s} = (1/\sqrt{2\gamma})[(\sqrt{\gamma+\alpha})C_{1s} + (\sqrt{\gamma-\alpha})C_{2s}], \text{ or}$$

$$L'_{1s} = (\cos\theta).C_{1s} + (\sin\theta).C_{2s}, \quad (65)$$

$$L'_{2s} = (1/\sqrt{2\gamma})[-(\sqrt{\gamma-\alpha})C_{1s} + (\sqrt{\gamma+\alpha})C_{2s}], \text{ or}$$

$$L'_{2s} = -(\sin\theta).C_{1s} + (\cos\theta).C_{2s}, \quad (66)$$

$$C_{1s} = (1/\sqrt{2\gamma})[(\sqrt{\gamma+\alpha})L'_{1s} - (\sqrt{\gamma-\alpha})L'_{2s}], \text{ or}$$

$$C_{1s} = (\cos\theta).L'_{1s} - (\sin\theta).L'_{2s}, \quad (67)$$

$$C_{2s} = (1/\sqrt{2\gamma})[(\sqrt{\gamma-\alpha})L'_{1s} + (\sqrt{\gamma+\alpha})L'_{2s}], \text{ or}$$

$$C_{2s} = (\sin\theta).L'_{1s} + (\cos\theta).L'_{2s}. \quad (68)$$

2.6 Algorithm for Hierarchical APCA with a Matrix of size *2×2*

In this case the processing is performed as follows. First, the group of MS images is sub-divided into groups (GOP) of 8 images each, for which is supposed that they are highly correlated. On the other hand, each GOP is further divided into 4 sub-groups.

As it is shown on Fig. 4, on each sub-group of 2 images from the first hierarchical level of HAPCA is applied APCA with a matrix of size *2×2*. In result are obtained 2 eigen images, colored in yellow and blue correspondingly. After that, the eigen images are rearranged so that the first sub-group of 2 eigen

images to comprise the first images from each group, the second group of 2 eigen images - the second images from each group, etc. For each GOP of 8 intermediate eigen images in the first hierarchical level on each sub-group of 2 eigen values is applied in similar way the next APCA, with a matrix of size *2×2*. In result are obtained 2 new eigen images (i.e. the eigen images of the group of 2 intermediate eigen images), colored in yellow, and blue correspondingly. Then the eigen images are rearranged again so, that the first group of 2 eigen images to contain the first images from each group before the rearrangement; the second group of 2 eigen images - the second image before the rearrangement, etc. In result is achieved significant decorrelation for the processed group of images, which is a reliable basis for efficient compression/restoration (HAPCA is a reversible transform). For this is necessary to have information about the transform matrix, used for each couple of images in all hierarchical levels – 4 matrices for one GOP altogether.

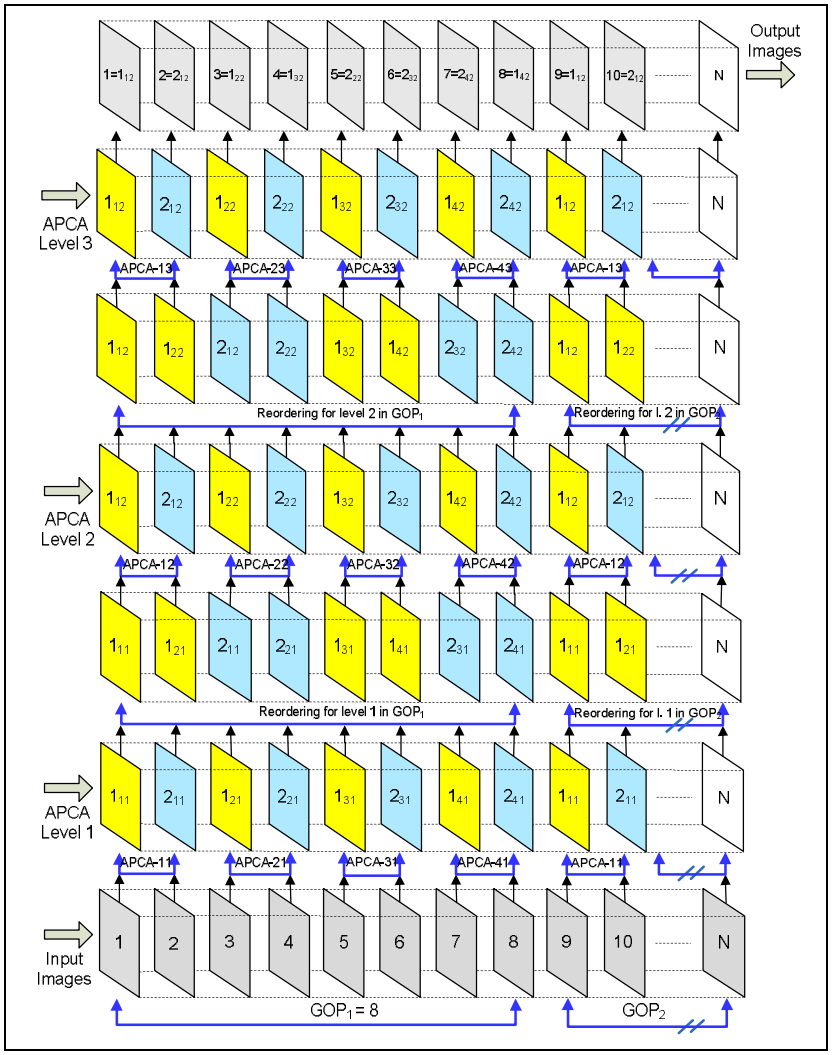


Fig. 4. Processing of a group of MS images with 3-level Hierarchical Adaptive PCA with a matrix of size *2×2*

Numerical example for HAPCA with a matrix of size 2×2 and GOP = 4.

Let's have a group of 4 matrix images $[C_1] \div [C_4]$ of size *2×2* (GOP=4) with pixels, whose values are:

$$[C_1]=\begin{bmatrix} C_{11}(1) & C_{12}(1) \\ C_{21}(1) & C_{22}(1) \end{bmatrix}=\begin{bmatrix} 2 & 3 \\ 4 & 2 \end{bmatrix},$$

$$[C_2]=\begin{bmatrix} C_{11}(2) & C_{12}(2) \\ C_{21}(2) & C_{22}(2) \end{bmatrix}=\begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix}$$

$$[C_3]=\begin{bmatrix} C_{11}(3) & C_{12}(3) \\ C_{21}(3) & C_{22}(3) \end{bmatrix}=\begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix},$$

$$[C_4]=\begin{bmatrix} C_{11}(4) & C_{12}(4) \\ C_{21}(4) & C_{22}(4) \end{bmatrix}=\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

In Level 1 of HAPCA with a matrix of size *2×2* these images are divided into 2 sub-groups: $[C_1]$, $[C_2]$ and $[C_3]$, $[C_4]$, for which are calculated the vectors $\vec{C}_s(i)=\left[C_{1s}(i), C_{2s}(i)\right]^t$ for *s=1,2,..,4* and *i=1,2*:

- for the first sub-group $[C_1]$, $[C_2]$ (*i=1*) these are the vectors:

$$\vec{C}_1(1)=\left[C_{11}(1)=2, C_{21}(1)=3\right]^t;$$
$$\vec{C}_2(1)=\left[C_{12}(1)=3, C_{22}(1)=2\right]^t;$$
$$\vec{C}_3(1)=\left[C_{13}(1)=4, C_{23}(1)=2\right]^t;$$
$$\vec{C}_4(1)=\left[C_{14}(1)=2, C_{24}(1)=3\right]^t;$$

- for the second sub-group of images $[C_3]$, $[C_4]$ (*i=2*) – the vectors:

$$\vec{C}_1(2)=\left[C_{11}(2)=2, C_{21}(2)=1\right]^t;$$
$$\vec{C}_2(2)=\left[C_{12}(2)=1, C_{22}(2)=2\right]^t;$$
$$\vec{C}_3(2)=\left[C_{13}(2)=3, C_{23}(2)=2\right]^t;$$
$$\vec{C}_4(2)=\left[C_{14}(2)=2, C_{24}(2)=4\right]^t;$$

For the <u>first sub-group</u> the covariance matrix $[K_C(1)]$ of vectors $\vec{C}_s(1)$ is:

$$\left[K_C(1)\right]=\begin{bmatrix} k_{11}(1) & k_{12}(1) \\ k_{21}(1) & k_{22}(1) \end{bmatrix}=\begin{bmatrix} 0.687 & -0.375 \\ -0.375 & 0.250 \end{bmatrix}$$

The elements $k_{ij}(1)$ of this matrix are:

$k_{11}(1)=k_1(1)=0.687$, $k_{22}(1)=k_2(1)=0.250$,

$k_{12}(1)=k_{21}(1)=k_3(1)=-0.375$.

Hence, $\alpha(1)=0.437$, $\beta(1)=-0.750$, $\gamma(1)=0.868$ and the rotation angle for the coordinate system $C_{1s}(1), C_{2s}(1)$ in result of APCA-1 is:

$$\theta_1=arctg\left[\frac{\beta(1)}{\alpha(1)+\gamma(1)}\right]=arctg(-0.574)=-0.521.$$

Then, $\cos(-0.521)=0.867$ and $\sin(-0.521)=-0.498$. The matrix of the direct APCA-1, used for the rotation of the coordinate axes $C_{1s}(1), C_{2s}(1)$ on angle θ_1, is:

$$\left[\Phi(1)\right]=\begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{bmatrix}=\begin{bmatrix} 0.867 & -0.498 \\ 0.498 & 0.867 \end{bmatrix}.$$

In this case, the direct transform of components $C_{1s}(1), C_{2s}(1)$ of the vectors $\vec{C}_s(1)$ for *s=1,2,3,4* without shifting the origin of the coordinate system, is represented by the relations:

$$L_{1s}(1)=0.867\,C_{1s}(1)-0,498\,C_{2s}(1);$$
$$L_{2s}(1)=0.498\,C_{1s}(1)+0,867\,C_{2s}(1),$$

where $L_{1s}(1), L_{2s}(1)$ are the components of the transformed vectors, $\vec{L}_s(1)$:

$$\vec{L}_1(1)=[0.240, 3.597]^t;$$
$$\vec{L}_2(1)=[1.605, 3.228]^t;$$
$$\vec{L}_3(1)=[2.472, 3.726]^t;$$
$$\vec{L}_4(1)=[0.240, 3.597]^t;$$

From these relations are obtained the matrices $[L_1], [L_2]$ of the corresponding eigen images:

$$\left[L_1\right]=\begin{bmatrix} L_{11}(1) & L_{12}(1) \\ L_{21}(1) & L_{22}(1) \end{bmatrix}=\begin{bmatrix} 0.240 & 1.605 \\ 2.472 & 0.240 \end{bmatrix},$$
$$\left[L_2\right]=\begin{bmatrix} L_{11}(2) & L_{12}(2) \\ L_{21}(2) & L_{22}(2) \end{bmatrix}=\begin{bmatrix} 3.597 & 3.228 \\ 3.726 & 3.597 \end{bmatrix}.$$

For the <u>second sub-group</u> the covariance matrix $[K_C(2)]$ of vectors $\vec{C}_s(2)$ is:

$$\left[K_C(2)\right]=\begin{bmatrix} k_{11}(2) & k_{12}(2) \\ k_{21}(2) & k_{22}(2) \end{bmatrix}=\begin{bmatrix} 0.500 & 0.000 \\ 0.000 & 1.187 \end{bmatrix}$$

The elements $k_{ij}(2)$ of this matrix are:

$k_{11}(2)=k_1(2)=0.5$, $k_{22}(2)=k_2(2)=1.187$,

$k_{12}(2)=k_{21}(2)=k_3(2)=0$.

Hence, $\alpha(2)=-0.687$, $\beta(2)=0$, $\gamma(2)=-0.687$.

Then, the rotation angle for the coordinate system $C_{1s}(2), C_{2s}(2)$ obtained in result of APCA-2, is $\theta_2=arctg(0)=0$, for which $\cos(0)=1$ and $\sin(0)=0$.

The matrix of the direct AKLT-2, used for the rotation of the coordinate axes, is a single matrix:

$$\left[\Phi(2)\right]=\begin{bmatrix} \cos\theta_2 & \sin\theta_2 \\ -\sin\theta_2 & \cos\theta_2 \end{bmatrix}=\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

In this case, the direct transform of the vectors $\vec{C}_s(2)$ for *s=1,2,3,4* without shifting the origin of the coordinate system, is given by the relations:

$$L_{1s}(2)=C_{1s}(2);\quad L_{2s}(2)=C_{2s}(2),$$

where $L_{1s}(2), L_{2s}(2)$ are the components of the transformed vectors $\vec{L}_s(2)$. From them are obtained the matrices $[L_3], [L_4]$ of the corresponding eigen images:

$$[L_3]=[C_3]=\begin{bmatrix} L_{11}(3) & L_{12}(3) \\ L_{21}(3) & L_{22}(3) \end{bmatrix}=\begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix},$$

$$[L_4]=[C_4]=\begin{bmatrix} L_{11}(4) & L_{12}(4) \\ L_{21}(4) & L_{22}(4) \end{bmatrix}=\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}.$$

In this particular case the vectors $\vec{C}_s(2)$ from the second sub-group are decorrelated, because $k_3(2)=0$, and they coincide with the transformed vectors $\vec{L}_s(2)$, while eigen images $[L_3], [L_4]$ coincide with their corresponding initial images $[C_3]$, $[C_4]$.

By analogy with the algorithm from Fig. 4, the obtained eigen images from first and second group are rearranged, and after that they are divided again into 2 sub-groups: the first comprises the eigen images $[L_1], [L_3]$, and the second - $[L_2], [L_4]$.

- the vectors for the first sub-group of rearranged images $[L_1]$, $[L_3]$ ($i=1$) are:

$$\vec{L}_1(1)=[0.240, 2.0]^t; \quad \vec{L}_2(1)=[1.605, 1.0]^t;$$

$$\vec{L}_3(1)=[2.472, 3.0]^t; \; \vec{L}_4(1)=[0.240, 2.0]^t.$$

- the vectors for the second sub-group of rearranged images $[L_2]$, $[L_4]$ ($i=2$) are correspondingly:

$$\vec{L}_1(2)=[3.597, 1.0]^t; \; \vec{L}_2(2)=[3.228, 2.0]^t;$$

$$\vec{L}_3(2)=[3.726, 2.0]^t; \; \vec{L}_4(2)=[3.597, 4.0]^t.$$

In the Level 2 of HAPCA with a matrix of size 2×2 on the rearranged vectors from first and second sub-group are performed APCA-1 and APCA-2 in similar way, as it was done in Level 1 of the hierarchical transform.

For the <u>first sub-group</u> the covariance matrix $[K_L(1)]$ of the vectors $\vec{L}_s(1)$ is:

$$[K_L(1)]=\begin{bmatrix} k_{11}(1) & k_{12}(1) \\ k_{21}(1) & k_{22}(1) \end{bmatrix}=\begin{bmatrix} 0.902 & 0.216 \\ 0.216 & 0.500 \end{bmatrix}.$$

The elements $k_{ij}(1)$ of this matrix are correspondingly:

$$k_{11}(1)=k_1(1)=0.902, \; k_{22}(1)=k_2(1)=0.500,$$
$$k_{12}(1)=k_{21}(1)=k_3(1)=0.216.$$

Hence, $\alpha(1)=0.402$, $\beta(1)=0.433$, $\gamma(1)=0.591$. Then, the rotation angle of the coordinate system $L_{1s}(1), L_{2s}(1)$ in result of APCA-1 is:

$$\theta_1=arctg\left[\frac{\beta(1)}{\alpha(1)+\gamma(1)}\right]=arctg(0.436)=0.411.$$

Correspondingly, $\cos(0.411)=0.916$ and $\sin(0.411)=0.399$. The matrix of the direct AKLT-1, through which is done the rotation on angle θ_1 of the coordinate axes $L_{1s}(1), L_{2s}(1)$, is:

$$[\Phi(1)]=\begin{bmatrix} cos\theta_1 & sin\theta_1 \\ -sin\theta_1 & cos\theta_1 \end{bmatrix}=\begin{bmatrix} 0.916 & 0.399 \\ -0.399 & 0.916 \end{bmatrix}.$$

In this case the direct transform of the components $L_{1s}(1), L_{2s}(1)$ of vectors $\vec{L}_s(1)$ for $s=1,2,3,4$ without shifting the coordinate system origin, is depicted by the relations:

$$L'_{1s}(1)=0.916.L_{1s}(1)+0.399.L_{2s}(1);$$
$$L'_{2s}(1)=-0.399.L_{1s}(1)+0.916.L_{2s}(1),$$

where $L'_{1s}(1), L'_{2s}(1)$ are the components of the transformed vectors $\vec{L}'_s(1)$:

$$\vec{L}'_1(1)=[0.998, 1.737]^t; \; \vec{L}'_2(1)=[1.870, 0.275]^t;$$

$$\vec{L}'_3(1)=[3.465, 1.762]^t; \; \vec{L}'_4(1)=[0.998, 1.737]^t;$$

From them are obtained the matrices $[L'_1], [L'_2]$ of the corresponding eigen images:

$$[L'_1]=\begin{bmatrix} L_{11}(1) & L_{12}(1) \\ L_{21}(1) & L_{22}(1) \end{bmatrix}=\begin{bmatrix} 0.998 & 1.870 \\ 3.465 & 0.998 \end{bmatrix},$$

$$[L'_2]=\begin{bmatrix} L_{11}(2) & L_{12}(2) \\ L_{21}(2) & L_{22}(2) \end{bmatrix}=\begin{bmatrix} 1.737 & 0.275 \\ 1.762 & 1.737 \end{bmatrix}.$$

For the <u>second sub-group</u> the covariance matrix $[K_L(2)]$ of the vectors $\vec{L}_s(2)$ is:

$$[K_L(2)]=\begin{bmatrix} k_{11}(2) & k_{12}(2) \\ k_{21}(2) & k_{22}(2) \end{bmatrix}=\begin{bmatrix} 0.098 & -0.299 \\ -0.299 & 1.187 \end{bmatrix}.$$

The elements $k_{ij}(2)$ of this matrix are:

$$k_{11}(2)=k_1(2)=0.098, \; k_{22}(2)=k_2(2)=1.187,$$
$$k_{12}(2)=k_{21}(2)=k_3(2)=-0.299.$$

Hence, $\alpha(2)=-1.088$, $\beta(2)=-0.599$, $\gamma(2)=1.243$ and then the rotation angle of the coordinate system $L_{1s}(2), L_{2s}(2)$ in result of APCA-2 is:

$$\theta_2=arctg\left[\frac{\beta(2)}{\alpha(2)+\gamma(2)}\right]=arctg(-3.889)=-1.319.$$

Hence, $\cos(-1.319)=0.248$ and $\sin(-1.319)=-0.968$. The matrix for the direct AKLT-2, through which the coordinate axes $L_{1s}(2), L_{2s}(2)$ are rotated on angle θ_2, is:

$$[\Phi(2)]=\begin{bmatrix} cos\theta_2 & sin\theta_2 \\ -sin\theta_2 & cos\theta_2 \end{bmatrix}=\begin{bmatrix} 0.248 & -0.968 \\ 0.968 & 0.248 \end{bmatrix}.$$

In this case the direct transform of the components $L_{1s}(2), L_{2s}(2)$ of vectors $\vec{L}_s(2)$ for *s=1,2,3,4* without shifting the origin of the coordinate system, is given by the relations:

$$L'_{1s}(2)=0.248.L_{1s}(2)-0.968.L_{2s}(2);$$
$$L'_{2s}(2)=-0.968.L_{1s}(2)+0.248.L_{2s}(2),$$

where $L'_{1s}(2), L'_{2s}(3)$ are the components of the transformed vectors $\vec{L}'_s(2)$:

$$\vec{L}'_1(1)=[0.001, 4.081]^t;\ \vec{L}'_2(1)=[-1.122, 3.624]^t;$$
$$\vec{L}'_3(1)=[-1.009, 4.106]^t;\ \vec{L}'_4(1)=[-2.978, 4.479]^t.$$

From them are obtained the matrices $[L'_3], [L'_4]$ of corresponding eigen images:

$$[L'_3]=\begin{bmatrix} L_{11}(2) & L_{12}(2) \\ L_{21}(2) & L_{22}(2) \end{bmatrix}=\begin{bmatrix} 0.001 & -1.122 \\ -1.009 & -2.978 \end{bmatrix},$$
$$[L'_4]=\begin{bmatrix} L_{11}(2) & L_{12}(2) \\ L_{21}(2) & L_{22}(2) \end{bmatrix}=\begin{bmatrix} 4.081 & 3.624 \\ 4.106 & 4.479 \end{bmatrix}.$$

All eigen images $[L'_3], [L'_4]$ from the first and second sub-group are rearranged again, in result of which are obtained the following images:

$$[E_1]=[L'_4],\ [E_2]=[L'_1],\ [E_3]=[L'_3],\ [E_4]=[L'_2].$$

The power of the rearranged eigen images $[E_1] \div [E_4]$ are:

$$P_{E1}=(1/4)\sum_{i=1}^{2}\sum_{j=1}^{2}E_{ij,1}^2=16.68;$$
$$P_{E2}=(1/4)\sum_{i=1}^{2}\sum_{j=1}^{2}E_{ij,2}^2=4.37;$$
$$P_{E3}=(1/4)\sum_{i=1}^{2}\sum_{j=1}^{2}E_{ij,3}^2=2.87;$$
$$P_{E4}=(1/4)\sum_{i=1}^{2}\sum_{j=1}^{2}E_{ij,3}^2=2.30.$$

Their powers, normalized towards P_{E4} are correspondingly: 7.24, 1.89, 1.2, and 1.0. The powers of the components of images $[C_1]$, $[C_2]$, $[C_3]$, $[C_4]$ are correspondingly:

$$P_{C1}=(1/4)\sum_{i=1}^{2}\sum_{j=1}^{2}C_{ij,1}^2=8.25;$$
$$P_{C2}=(1/4)\sum_{i=1}^{2}\sum_{j=1}^{2}C_{ij,2}^2=6.5;$$
$$P_{C3}=(1/4)\sum_{i=1}^{2}\sum_{j=1}^{2}C_{ij,3}^2=4.5;$$
$$P_{C4}=(1/4)\sum_{i=1}^{2}\sum_{j=1}^{2}C_{ij,4}^2=6.25,$$

and the corresponding powers, normalized towards P_{E3}, are: 1.83; 1.44; 1.0; 1.38. Hence, $\eta_E=P_{E1}/P_{E4}=7.24$ and $\eta_C=P_{C1}/P_{C3}=1.83$, from which follows, that $\eta_E/\eta_C \approx 4$.

This example shows that after applying 2-level HAPCA with a matrix of size 2×2, the relation (η_E/η_C) for the eigen images of maximum and minimum power is 4 times higher than that for the original images of same kind. The power of the set of original images after their transform is concentrated in the first eigen image, and together with this, the power of the remaining eigen images in the set decreases rapidly.

2.7 Generalization about the Algorithm HAPCA with a Matrix of size *2×2* for a Set of 16-Component Vectors

The algorithm HAPCA from Fig. 4 could be generalized for sets of *S* 16-component vectors $\vec{C}_s=[C_{1s},C_{2s},..,C_{16s}]^t$ (*s=1,2,..,S*). These vectors could be calculated for a set of MS images, divided into groups (GOP), which comprise 16 images each. In this case, every group is divided into 8 sub-groups, each of which contains *S* two-component vectors $\vec{C}_{is}=[C_{i1s},C_{i2s}]^t$ (*i=1,2,..,8*). Because of the large number of sub-groups used in the HAPCA with a matrix of size *2×2* algorithm, is necessary to increase the number of transform levels, compared to that of HAPCA with a matrix of size *3×3*. As an example, on Fig. 5 is shown the algorithm for 3-level HAPCA with a matrix of size 2×2, applied on the set of vectors $\vec{C}_{1s}=[C_{11s},C_{12s}]^t$, $\vec{C}_{2s}=[C_{21s},C_{22s}]^t$, ..., $\vec{C}_{8s}=[C_{81s},C_{82s}]^t$ for *s=1,2,..,S*.

In order to obtain higher decorrelation for vectors $\vec{L}_s=[L_{1s},L_{2s},..,L_{16s}]^t$, is necessary the mutual correlation between vectors $\vec{C}_{is}=[C_{i1s},C_{i2s}]^t$ to be high enough. This could be achieved through preliminary space transform for vectors $\vec{C}_{is}$ so, that to obtain clusterization, for example, in accordance with the Least Squares Method. The 16-component vectors $\vec{L}_s=[L_{s1},L_{s2},..,L_{s16}]^t$ obtained after 3-level HAPCA with a matrix of size *2×2* could be done shorter by retaining their most powerful coefficients only. The so obtained vectors could be used for pattern recognition in the reduced vector space.

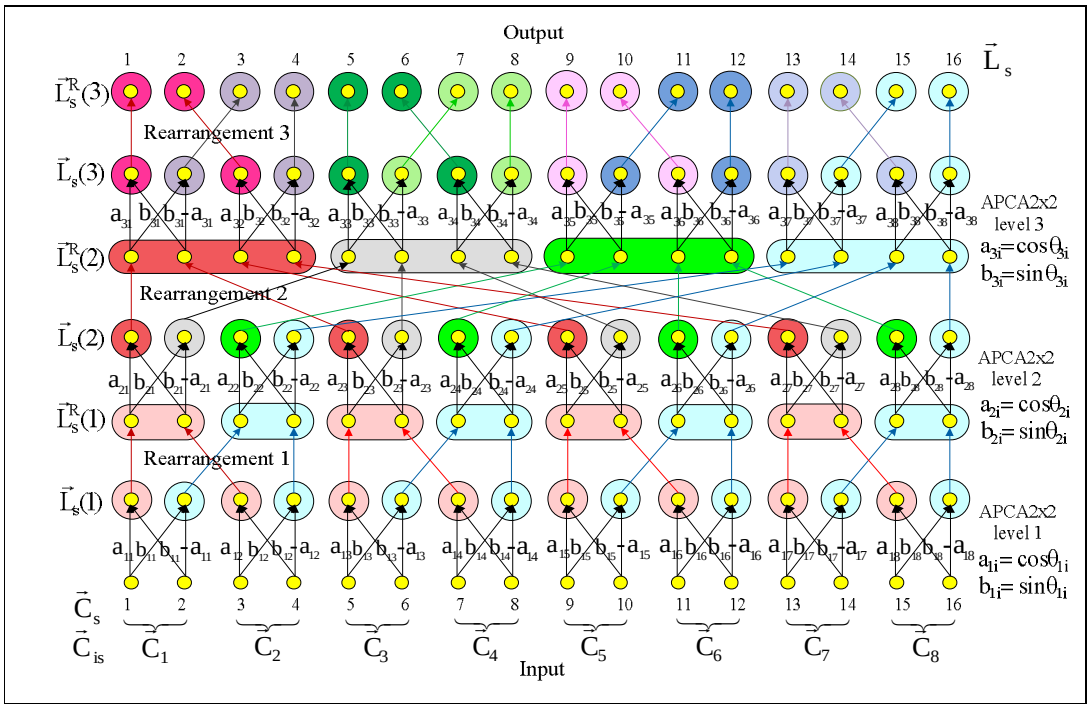


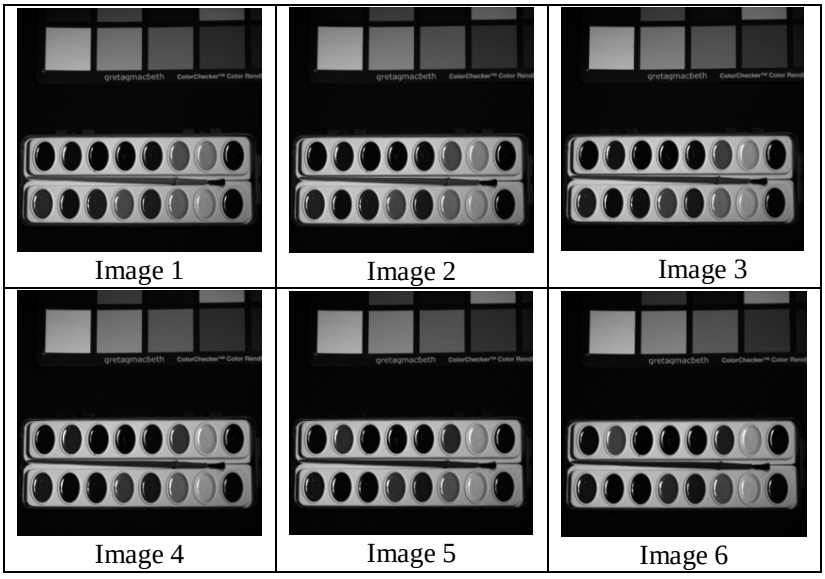
Fig. 5. Processing of a group of 8 vectors with Hierarchical Adaptive 3-level PCA with a matrix of size 2×2

3 Experimental results

On the basis of the 2-level HAPCA algorithm, shown on Fig. 1, were done experiments with sequences of MS images of size *512×512* pixels, 24 bpp. The sequence was divided into groups, each containing 9 MS images. From them were obtained 2^{18} vectors in each group, containing 3 MS images each.

As an example, on Fig. 6 is shown one of the test groups (Set 1), which contains MS Image1,..,Image 9.

On Fig. 7 are shown the corresponding eigen MS images, obtained after applying the 2-level HAPCA algorithm on the group of images (Set 1). As it could be seen from the results shown on Fig. 7, on the first eigen MS image is concentrated the main part of the energy of all 9 images, and the energy of each next eigen image decreases rapidly. In Table 1 is given the power distribution of all eigen MS images in Set 1 and the relative power distribution.



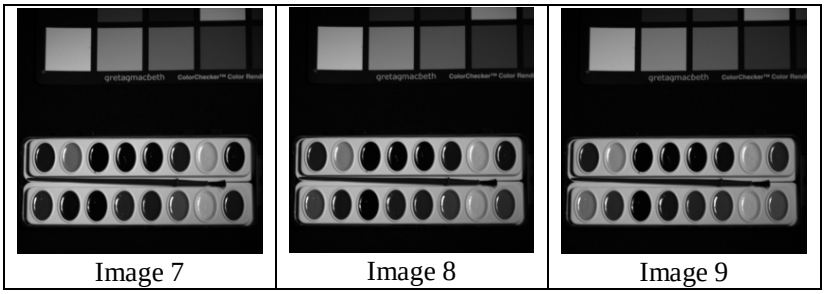


Fig. 6. Group of 9 consecutive MS images in Set 1.

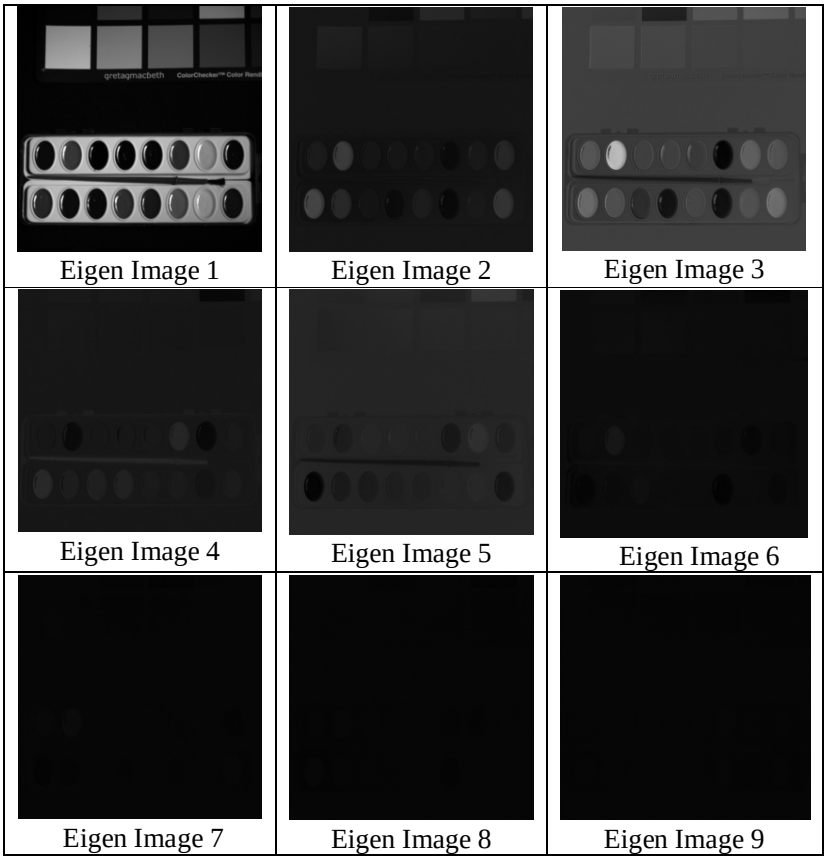


Fig. 7. Eigen images, obtained for Set 3 after performing 2-levels HAPCA.

Table 1. Power distribution of all eigen MS images in Set 1, relative and cumulative power distribution in %.

Name	Level 2 (arranged)	Relative power	Relative power %	Cumulative power %
Eigen Im. 1	35499.0	148722	0.990390	99.039
Eigen Im. 2	47.490	198	0.001320	99.171
Eigen Im. 3	257.13	1077	0.007170	99.888
Eigen Im. 4	13.97	58	0.000380	99.926
Eigen Im. 5	22.0	92	0.000610	99.987
Eigen Im. 6	3.00	14	0.000090	99.996
Eigen Im. 7	0.57	2	0.000012	99.997
Eigen Im. 8	0.26	1	0.000006	99.998
Eigen Im. 9	0.23	1	0.000006	100.00

On the basis of data given in Table 1 are build the corresponding graphics, which represent the power distribution of all 9 eigen images, shown correspondingly on Fig. 8. The data in the last column of Table 1 show, that in the first 3 eigen MS images are concentrated 99,88 % of the total power of all 9 images in the GOP.

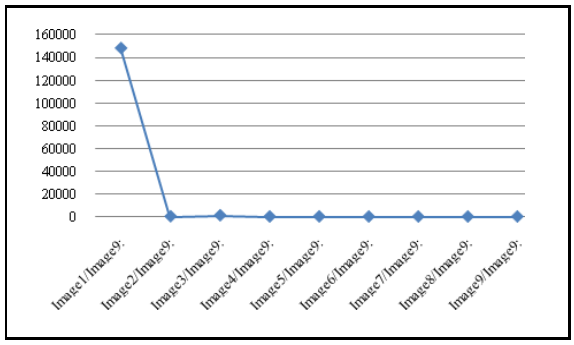


Fig. 8. Relative power distribution for Set 1, Level 2 (arranged)

From Fig. 8 follows that the power of the first eigen MS image for Set 1 is more than 148000 times larger than that of each of the next 8 eigen images. The values for pixels of the eigen images, obtained in result of the direct 2-level HAPCA, were calculated with full accuracy, and after corresponding rounding could be transformed into 24-bit numbers. Then, if on the 24 bpp eigen images is applied the inverse 2-level HAPCA, the quality of corresponding restored images in GOP, evaluated by their Peak Signal-to-Noise Ratio (PSNR), is ≥ 50 dB. This was confirmed by the results from Fig. 9, obtained for the eigen images in Set 1 of Fig. 8 after inverse HAPCA, following the algorithm, shown on Fig. 1. Hence, the group of 9 MS images could be restored with retained visual quality.

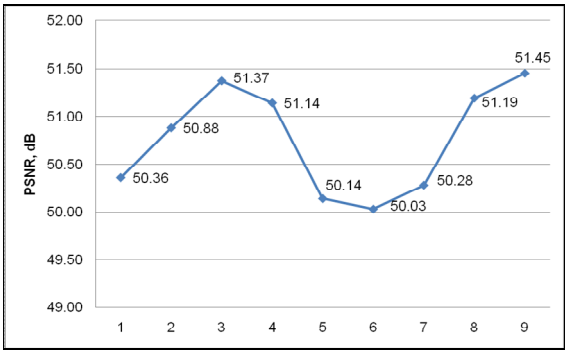


Fig. 9. Evaluation of the quality of restored images from Set 1 with PSNR [dB], after applying inverse 2-level HAPCA on eigen images from Fig. 7.

Similar results were obtained for other sets of MS images: Set 2, Set 3,.., Set 7. This result illustrates the ability for efficient compression of a set of MS images, when HAPCA is used. The experimental results were obtained with the software implementation of HAPCA in Visual C.

4 Evaluation of the 2-level HAPCA Computational Complexity

The computational complexity of the 2-level HAPCA algorithm, based on matrices of size *3×3* will be compared with that of the PCA algorithm with a matrix of size *9×9*. Both algorithms are compared in respect to the performed number of operations *O* (additions and multiplications) [8] needed for the calculation of the following components:

- covariance matrices $[K_C]$ – in total 6 for the first algorithm, each of size *3×3*, and one matrix $[K_C]$ of size *9×9* – for the second algorithm;
- eigen values and eigen vectors of the corresponding matrices $[K_C]$;
- eigen images of each GOP, obtained using both algorithms.

On the basis of the computational complexity analysis given in [34] for APCA with matrix of size *3×3* and for PCA with a matrix of size *N×N* follows, that for the 2-level HAPCA with a *3×3* matrix and for the PCA with a *9×9* matrix, we have:

- The number of operations needed for the calculation of all elements k_{ij} for all 6 matrices $[K_C]$ of size *3×3* (for the 2-level HAPCA) and for one matrix $[K_C]$ of size 9×9 (for the PCA), is:

$$O_k(N)\big|_{N=3} = 3N(N+1)[N(N-1)+2(N+2)] = 576. \quad (69)$$

$$O_k(N)\big|_{N=9} = \frac{1}{2}N(N+1)[N(N-1)+2(N+2)] = 4230. \quad (70)$$

- The number of operations needed for calculation of the eigenvalues of matrices $[K_C]$ for the 2-level HAPCA and of the PCA $[K_C]$ matrix, when the QR decomposition and the Householder transform of (*N-1*) steps [34] were used, is:

$$O_{val}(N)\big|_{N=3} = 282. \quad (71)$$

$$O_{val}(N)\big|_{N=9} = (N-1)\left(\frac{4}{3}N^2 + \frac{17}{6}N + 7\right) = 1124. \quad (72)$$

- The number of operations needed for the calculation of the eigen vectors of matrices $[K_C]$ for the 2-level HAPCA and for the PCA matrix $[K_C]$, in case that iterative algorithm with 4 iterations is used, is correspondingly:

$$O_{vec}(N)\big|_{N=3} = 275. \quad (73)$$

$$O_{vec}(N)\big|_{N=9} = N[2N(4N+5)-1] = 6633. \quad (74)$$

- The number of operations needed for the calculation of a group of 9 eigen images (each of S pixels), obtained in result of the direct 2-level HAPCA and of PCA for zero mean vectors, is correspondingly:

$$O_{HAPCA}(N)\big|_{N=3} = 6SN(2N-1) = 90S. \quad (75)$$

$$O_{PCA}(N)\big|_{N=9} = SN(2N-1) = 153S. \quad (76)$$

Then the total number of operations Σ for the 2-level HAPCA and for PCA is correspondingly:

$$\Sigma_1(3) = [O_k(3) + O_{val}(3) + O_{vec}(3) + O_{HAPCA}(3)] = \\ = 576 + 282 + 275 + 90S = 1133 + 90S, \quad (77)$$

$$\Sigma_2(9) = [O_k(9) + O_{val}(9) + O_{vec}(9) + O_{PCA}(9)] = \\ = 4239 + 1124 + 6633 + 153S = 11996 + 153S. \quad (78)$$

The reduction of the total number of operations needed for the 2-level HAPCA, compared to that of the PCA could be evaluated using the coefficient ξ:

$$\xi(S) = \frac{\Sigma_2(9)}{\Sigma_1(3)} = \frac{11996 + 153S}{1133 + 90S}. \quad (79)$$

For example, for $S=100$ $\xi(100) = 2.96$; for $S=1000$ correspondingly $\xi(1000) = 1.81$ and $\xi(\infty) \to 1.7$. Hence, $\Sigma_1(S)$ is at least 1.7 times smaller than $\Sigma_2(S)$ for each value of S (in average, about 2 times).

5 Conclusions

The basic qualities of the offered HAPCA for processing of sets of MS images are:

1. Lower computational complexity than PCA for the whole GOP, due to the lower complexity of APCA with matrices of size 2×2 and 3×3 compared to the case, for which for the calculation of the PCA matrix are used numerical methods [15];

2. HAPCA could be used not only for efficient compression of sets of MS images, but also for sequences of medical CT images, video sequences, obtained from stationary TV camera, compression of multi-view images [17], image fusion [21], face recognition [23], etc.;

3. There is also a possibility for further development of the HAPCA algorithms, through: use of Integer PCA for lossless coding of MS images [18,19]; HAPCA with a matrix of size $N\times N$ (N - digit, divisible to 2 or 3), but without using numerical methods, etc.

For the further compression of the original group of eigen MS images could be used for example, the "Branched" Inverse Pyramid Decomposition (BIPD) [16,17] with nonlinear pre- and post-processing, based on the pixel-by-pixel Adaptive Histogram Matching transform.

In the future research work will be done software modeling and perfection of the HAPCA algorithm. The modeling and experiments will be performed for a large image database with image sequences/sets of various kinds. The so obtained results will be evaluated and compared to other similar algorithms and will be investigated possible new application areas: remote investigation of the earth surface, medical investigations, automatic manufacturing control, defense, etc.

References

[1] R. Dony, Karhunen-Loeve transform, In: *The Transform and Data Compression Handbook*, K. Rao and P. Yip (Eds.), Boca Raton, CRC Press, 2001.

[2] R. Gonzales, R. Woods, *Digital Image Processing*, Second Ed., Prentice Hall, 2002.

[3] M. Klimesh, *Compression of Multispectral Images*, TDA Progress Rep.42-129,1997, pp. 1-7.

[4] G. Gelli, G. Poggi, Compression of Multispectral Images by Spectral Classification and Transform Coding, *IEEE Trans. on Image Processing*, Vol. 8, No. 4, 1999, pp. 476–489.

[5] P. Dragotti, G. Poggi, A. Ragozini, Compression of Multispectral Images by Three-dimensional SPIHT algorithm, *IEEE Trans. Geosci*. Remote Sens., Vol. 38 (1), 2000, pp. 416–428.

[6]I. Rosca, L State, C. Cocianu, Learning Schemes in Using PCA Neural Networks for Image restoration Purposes, *WSEAS Trans. on Information Science and Applications*, Issue 7, Vol. 5, 2008, pp. 1149-1159.

[7] X. Tang, W. A. Pearlman, J. W. Modestino, Hyperspectral Image Compression Using Three-dimensional Wavelet Coding, *Proc. SPIE*, Vol.5022, 2003, pp. 1037–1047.

[8] J. Saghri, A. Tescher, A. Planinac, KLT/JPEG 2000 Multispectral Bandwidth Compression with Region of Interest Prioritization Capability, *Proc. SPIE*, 2003, pp. 226–235.

[9] J. Wu, C. Wu, Multispectral Image Compression Using 3-dimensional Transform Zeroblock Coding. *Chinese Optic Letters*, Vol. 2,No.6, 2004, pp.1-4.

[10] Sh. Yu, Y. Murakami, T. Ob, M. Yamaguchi, N. Ohuama, Multispectral Image Compression for Improvement of Colorimetric and Spectral Reproducibility by Nonlinear Spectral Transform, *Optical Rev.*,Vol.13, No. 5, 2006, pp. 346–356.

[11] L. Chang, Multispectral Image Compression Using Eigenregion-based Segmentation, *Pattern Recognition*, Vol. 37, 2004, pp. 1233–1243.

[12] J. Richards, X. Jia, *Remote Sensing Digital Image Analysis. An Introduction*, 4th Ed., Springer, 2006.

[13] B. Aiazzi, S. Baronti, C. Lastri, Remote-Sensing Image Coding, In: *Document and Image Compression*, M. Barni Ed. CRC Taylor&Francis, 2006, pp. 389-412.

[14] M. Cagnazzo, S. Parrilli, G. Poggi, L. Verdoliva, Improved Class-Based Coding of Multispectral Images with Shape-Adaptive Wavelet Transform, *IEEE Geoscience and Remote Sensing Letters*, Vol. 4, No. 4, 2007, pp. 566-570.

[15] R. Kountchev, R. Kountcheva, Image Color Space Transform with Enhanced KLT, In: *New Advances in Intelligent Decision Technologies*, Eds. K. Nakamatsu et al, Springer, 2009, pp. 171-182.

[16] R. Kountchev, R. Kountcheva. Compression of CT Images with Branched Inverse Pyramidal Decomposition. *12th WSEAS Int. Conf. on Signal Processing, Computational Geometry and Artificial Vision (ISCGAV '12)*, Istanbul, Turkey, 2012, pp. 74-79.

[17] R. Kountchev, Vl. Todorov, R. Kountcheva, Compression of Multispectral and Multi-view Images with Inverse Pyramid Decomposition, *Intern. J. of Reasoning-Based Intelligent Systems (IJRIS)*, Vol.3, No. 2, 2011, pp. 124-131.

[18] J. Mielikäinen, A. Kaarna, Improved Back End for Integer PCA and Wavelet Transforms for Lossless Compression of Multispectral Images, *Proceedings, 16th Intern. Conf. on Pattern Recognition*, 2002, Vol. 2, pp. 257-260.

[19] P. Hao, Q. Shi, Reversible Integer KLT for Progressive-to-Lossless Compression of Multiple Component Images. *IEEE ICIP*, Spain, 2003.

[20] I. Bitaa, M. Barretb, D. Phamc. On Optimal Transforms in Lossy Compression of Multicomponent Images with JPEG 2000, *Signal Processing*, Vol. 90, No. 3, 2010, pp. 759-773.

[21] P. Ujwala, M. Uma, Image Fusion using Hierarchical PCA, *Intern. Conf. on Image Information Processing (ICIIP)*, 2011, pp. 1-6.

[22] M. Hanafi, A. Kohler, E. Qannari, Shedding New light on Hierarchical Principal Component Analysis, *J. of Chemometrics*, Vol 24, No. 11-12, 2010, pp. 703-709.

[23] G. Langs, H. Bischof, W. Kropatsch, *Hierarchical Top Down Enhancement of Robust PCA*, SSPR&SPR, T. Caelli et al. (Eds.), *LNCS* 2396, Springer 2002, pp. 234–243.

[24] I. Jolliffe, *Principal Component Analysis*, 2nd ed., Springer-Verlag, NY, 2002.

[25] J. Gentle, *Matrix Algebra: Theory, Computations, and Applications in Statistics*, Springer, Science+Business Media, LLC, 2007.

[26] K. Diamantaras, S. Kung, *Principal Component Neural Networks: Theory and Applications*, John Wiley & Sons, New York, 1996.

[27] V. Solo, X. Kong, Performance Analysis of Adaptive Eigen Analysis Algorithms, *IEEE Trans. Signal Processing*, 1998, Vol. 46, No. 3, pp. 636-645.

[28] R. Landqvist, A. Mohammed, Comparative Performance Analysis of Three Algorithms for Principal Component Analysis, *Radioengineering*, Vol. 15, No 4, 2006, pp. 84-90.

[29] S. Orfanidis, *SVD, PCA, KLT, CCA, and All That*, Rutgers University, Optimum Signal Processing, 332:525, 2007.

[30] F. Shao, G. Jiang, M. Yu, Color Correction for Multi-view Images Combined with PCA and ICA. *WSEAS Trans. on Biology and Biomedicine*, Iss. 6, Vol.4, 2007, pp. 73-79.

[31] R. Dony, S. Haykin, Optimally Adaptive Transform Coding, *IEEE Trans. on Image Processing*, Vol. 4, No. 10, 1995, pp. 1358-1370.

[32] V. Thirumalai, *Distributed Compressed Representation of Correlated Image Sets*, Thesis No. 5264, Lausanne, EPFL, 2012.

[33] L. Grasedyck, *Hierarchical Singular Value Decomposition of Tensors*, Preprint 20, AG Numerik/Optimierung, Philipps Universitat Marburg, July 2009, pp. 1-29.

[34] Y. Liu, C. Bouganis, P. Cheung, P. Leong S. Motley, Hardware Efficient Architectures for Eigenvalue Computation, *EDAA*, 2006, pp. 953-958.

[35] M. Yao, X. Qu, Q. Gu, T. Ruan, Z. Lou, Online PCA with Adaptive Subspace method for Real-Time Hand Gesture Learning and recognition, *WSEAS Trans. on Computers*, Issue 6, Vol. 9, 2010, pp. 583-592.

[36] R. Kountchev, K. Nakamatsu, One Approach for Grayscale Image Decorrelation with Adaptive Multi-level 2D KLT, In: *Advances in Knowledge-Based and Intelligent Information and Engineering Systems*, M. Graña et al., (Eds.), IOS Press, 2012, pp. 1303-1312.

[37] G. Korn, T. Korn, *Mathematical Handbook for Scientists and Engineers*, Mc Graw-Hill Book Company, NY, 1968.