ICIT 2015 The 7th International Conference on Information Technology
doi:10.15849/icit.2015.0008

Hierarchical Singular Value Decomposition for Halftone Images

Roumen Kountchev
Department of RCTV
Technical University of Sofia
Sofia, Bulgaria
rkountch@tu-sofia.bg

Roumiana Kountcheva
TK Engineering
Sofia
Bulgaria
kountcheva_4@yahoo.com

***Abstract* - This work is devoted to one new approach for decomposition of images represented by matrices of size $2^n \times 2^n$, based on the multiple application of the Singular Value Decomposition (SVD) over image blocks of relatively small size (2×2), obtained after division of the original image matrix. The new decomposition, called Hierarchical SVD, has tree structure of the kind binary tree of n hierarchical levels. Its basic advantages over the famous SVD are: the reduced computational complexity, the opportunity for parallel and recursive processing of the image blocks, based on relatively simple algebraic relations, the high concentration of the image energy in the first decomposition components, and the ability to accelerate the calculations through cutting-off the tree branches in the decomposition levels, where the corresponding eigen values are very small. The HSVD algorithm is generalized for images of unspecified size. The new decomposition opens numerous opportunities for fast image processing in various application areas: image compression, filtration, segmentation, merging, digital watermarking, extraction of minimum number of features sufficient for the objects recognition, etc.**

Keywords - Singular Value Decomposition (SVD), block SVD, Hierarchical SVD, binary tree, computational complexity.

I. INTRODUCTION

The SVD decomposition had significant influence on the processing and analysis of digital images used in computer vision systems. This decomposition was the target of significant number of investigations, presented in scientific monographs [1-6] and papers [7-12].

The SVD has the following basic features: 1) it is an optimum decomposition, because it concentrates maximum part of the image energy in a minimum number of components; 2) the image, restored after the reduction of the low-energy components has minimum mean square error. One of the basic problems, which restrict the practical use of the famous SVD, is its high computational complexity, which grows together with the size of the image matrix. Several approaches are offered to overcome this problem. The first approach is based on the SVD calculation through iterative methods, which do not demand to define the characteristic polynomial of the matrix. In this case the SVD is executed in two stages: in the first, the matrix is transformed into triangular form through the QR decomposition, and then - into bi-diagonal through Householder's transforms [13]; in the second stage on the bi-diagonal matrix is applied an iterative algorithm, whose iterations stop when the needed accuracy is obtained. Such is, for example, the iterative method of Jacobi [3, 6, 25], in accordance with which to calculate the SVD for a bi-diagonal matrix, is needed to execute a sequence of orthogonal transforms with matrices, which differ from the singular matrix in the elements of the rotation matrix of size 2×2 only. The second approach is based on the relation between the SVD and the Principal Component Analysis (PCA). It could be implemented through neural networks [14] of the kind generalized Hebbian or multilayer perceptron networks, which use iterative learning algorithms. One more approach is based on the algorithm, known as Sequential KL/SVD [15]. Its basic idea is given in brief as follows: the image matrix is divided into blocks of small size, on which is applied SVD, based on the QR decomposition [6]. The SVD is initially calculated for the first block, and then iterative SVD calculation is executed for each block, using the transform matrix, already obtained for the preceding block (update procedure). In the iteration process the SVD components, which have very small values, are eliminated.

II. RELATED WORK

Several methods had already been developed, aimed at the enhancement of the SVD calculation [16-19]. The first, called

doi:10.15849/icit.2015.0008

Randomized SVD [16, 17], is based on the algorithm in accordance with which, are randomly selected some rows (or columns) of the transform matrix. After scaling, they build a small matrix, for which is calculated the SVD, which is then used as an approximation of the original matrix. In [18] is offered the QUIC-SVD algorithm, which is suitable for matrices of very large size. Using this algorithm is achieved fast sample-based SVD approximation with automatic relative error control. This algorithm also uses a sampling mechanism, called "the cosine tree", to achieve best-rank approximation. The experimental investigation on the QUIC-SVD, given in [19], offers better results than these, obtained with MATLAB SVD and Tygert SVD [17]. The speedup achieved is 6-7 times higher compared to that of the exact SVD, but it depends on the selected value for the parameter δ which defines the higher limit of the approximation error with a probability of size (1-δ).

Significant number of SVD-based methods had been developed, aimed at the image compression efficiency enhancement [20-24]. The method, called Multiresolution SVD [20], comprises 3 steps: 1) image transform through 9/7 bi-orthogonal wavelets of 2 levels; 2) decomposition of the transformed image through SVD executed on blocks of size 2×2 up to level six, and 3) execution of the SPIHT and gzip algorithms. In [21] is offered a hybrid KLT-SVD algorithm for efficient image compression. The K-SVD [22] for facial image compression is a generalization of the K-means clusterization method and is applied in the iterative learning of over-complete sparse coding dictionaries. In correspondence with the combined compression algorithm presented in [23], the SVD is executed individually for each of the color components R, G, B, segregated from the image stored in the JPEG file format. In [24] is introduced the Higher-Order SVD (HOSVD), which is an extension of the SVD matrix to tensors with application in the data compression. In [26, 27] are presented some parallel hardware implementations of the SVD for symmetrical matrices, based on the Jacobi's method.

In this work is offered one new approach for hierarchical image decomposition, based on the multiple SVD execution on blocks of small size. This decomposition, called here the "Hierarchical SVD" (HSVD), has a tree structure of the kind binary or 3-nodes tree (full or truncated). The SVD calculation for blocks of size *2×2* is based on the adaptive KLT [28]. The HSVD algorithm [29, 30] is aimed at the achievement of decomposition with high computational efficiency, which is also suitable for parallel recursive processing with relatively simple algebraic operations, and permits calculation speedup through cutting-off the branches with very small eigenvalues.

The paper comprises the following sections: SVD calculation for a matrix of size *2×2*; representation of the hierarchical SVD for a matrix of size $2^n\times 2^n$; evaluation of the computational complexity of the hierarchical SVD of size $2^n\times 2^n$; representation of the HSVD algorithm through tree-like structure, and conclusions.

III. CALCULATION OF SVD WITH A MATRIX OF SIZE *2×2*

A. General case: SVD execution on image of size N×N

In the general case, the decomposition of the square image $[X(N)]$, represented by a matrix of size $N\times N$ is based on the direct SVD, defined by the relation below [10, 11]:

$$[X(N)]=[U(N)][\Lambda(N)]^{1/2}[V(N)]^t=\sum_{s=1}^{N}\sqrt{\lambda_s}\vec{U}_s\vec{V}_s^t. \quad (1)$$

The <u>inverse</u> SVD is respectively represented as:

$$[\Lambda(N)]^{1/2}=[U(N)]^t[X(N)][V(N)]. \quad (2)$$

In the equations above, the terms $\left[U(N)\right]=[\vec{U}_1,\vec{U}_2,..,\vec{U}_N]$ and $\left[V(N)\right]=[\vec{V}_1,\vec{V}_2,..,\vec{V}_N]$ are matrices, composed by the vectors $\vec{U}_s$ and $\vec{V}_s$ for $s=1,2,..,N$. Here $\vec{U}_s$ are the eigen vectors of the matrix $[Y(N)]=[X(N)]^t[X(N)]$ (left-singular vectors of $[X(N)]$), and $\vec{V}_s$ are the eigen vectors of the matrix $[Z(N)]=[X(N)][X(N)]^t$ (right-singular vectors of $[X(N)]$), for which:

$$[Y(N)]\vec{U}_s=\lambda_s\vec{U}_s, \quad (3)$$

$$[Z(N)]\vec{V}_s=\lambda_s\vec{V}_s. \quad (4)$$

$[\Lambda(N)]=diag[\lambda_1,\lambda_2,..,\lambda_N]$ is a diagonal matrix, composed of the eigenvalues λ_s of both matrices $[Y(N)]$ and $[Z(N)]$, which are same.

From (1) it follows that for the description of a matrix of size $N\times N$ are needed $N\times(2N+1)$ parameters in total, i.e., in the general case the SVD is an of over-complete decomposition.

B. Particular case: SVD for one image block of size 2×2

The direct SVD for the square block [X] of size 2×2 (N=2) is represented by the relation:

$$[X]=\begin{bmatrix} a & b \\ c & d \end{bmatrix}=[U][\Lambda]^{1/2}[V]^t= \\ =\sqrt{\lambda_1}\vec{U}_1\vec{V}_1^t+\sqrt{\lambda_2}\vec{U}_2\vec{V}_2^t=\sum_{s=1}^{2}\sqrt{\lambda_s}\vec{U}_s\vec{V}_s^t, \quad (5)$$

where *a,b,c,d* are pixels; λ_1, λ_2 - common eigen values of the symmetrical matrices $[Y]$ and $[Z]$:

$$[Y]=[X]^t[X]=\begin{bmatrix} a & c \\ b & d \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix}=\begin{bmatrix} (a^2+c^2) & (ab+cd) \\ (ab+cd) & (b^2+d^2) \end{bmatrix}; \quad (6)$$

$$[Z]=[X][X]^t=\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} a & c \\ b & d \end{bmatrix}=\begin{bmatrix} (a^2+b^2) & (ac+bd) \\ (ac+bd) & (c^2+d^2) \end{bmatrix}. \quad (7)$$

$\vec{U}_1$ and $\vec{U}_2$ are the eigenvectors of the matrix $[Y]$, for which: $[Y]\vec{U}_s=\lambda_s\vec{U}_s$, $s = 1, 2;$

$\vec{V}_1$ and $\vec{V}_2$ are the eigenvectors of the matrix $[Z]$, for which: $[Z]\vec{V}_s=\lambda_s\vec{V}_s$, $s = 1, 2.$

$\left[U\right]=[\vec{U}_1,\vec{U}_2]$ and $\left[V\right]^t=\begin{bmatrix}\vec{V}_1^t\\ \vec{V}_2^t\end{bmatrix}$ - matrices, composed of the eigenvectors $\vec{U}_s$ and $\vec{V}_s$.

C. Calculation of the eigenvalues and vectors of the symmetrical matrix of size 2×2

Let for *N=2* the corresponding matrix $\left[G\right]=\begin{bmatrix}g_{11} & g_{12}\\ g_{21} & g_{22}\end{bmatrix}$ is symmetrical in respect of its main diagonal. Then here could be assumed the simplified symbols: $g_{11}=g_1$, $g_{22}=g_2$, $g_{12}=g_{21}=g_3$. The eigenvalues λ_1,λ_2 of the matrix $[G]$ are the solution of the characteristic equation:

$$\det\left|[G]-\lambda[I]\right|=\lambda^2-(g_1+g_2)\lambda+(g_1g_2-g_3^2)=0. \tag{8}$$

Since the matrix $[G]$ is symmetrical, its eigenvalues are real numbers:

$$\begin{aligned}\lambda_1&=\frac{1}{2}\left[(g_1+g_2)+\sqrt{(g_1-g_2)^2+4g_3^2}\right],\\ \lambda_2&=\frac{1}{2}\left[(g_1+g_2)-\sqrt{(g_1-g_2)^2+4g_3^2}\right].\end{aligned} \tag{9}$$

The eigenvectors $\vec{\Phi}_s$ of the matrix $[G]$ for $s=1, 2$ are the solutions of the system of equations below:

$$\begin{aligned}&(g_1-\lambda_s)\Phi_{1s}+g_3\Phi_{2s}=0,\\ &g_1\Phi_{1s}+(g_2-\lambda_s)\Phi_{2s}=0,\\ &\Phi_{1s}^2+\Phi_{2s}^2=1.\end{aligned} \tag{10}$$

The eigenvector $\vec{\Phi}_s=\left[\Phi_{1s},\Phi_{2s}\right]^T$, which corresponds to the eigenvalue λ_s, is:

- For $s=1$

$$\begin{aligned}\vec{\Phi}_1&=\frac{1}{\sqrt{2[(g_1-g_2)^2+4g_3^2+(g_1-g_2)\sqrt{(g_1-g_2)^2+4g_3^2}\,]}}\times\\ &[(g_1-g_2)+\sqrt{(g_1-g_2)^2+4g_3^2},\,2g_3]^t=\frac{1}{\sqrt{2\gamma(\gamma+\alpha)}}[\alpha+\gamma,\beta]^t,\end{aligned} \tag{11}$$

- For $s=2$

$$\begin{aligned}\vec{\Phi}_2&=\frac{1}{\sqrt{2[(g_1-g_2)^2+4g_3^2-(g_1-g_2)\sqrt{(g_1-g_2)^2+4g_3^2}\,]}}\times\\ &[(g_1-g_2)-\sqrt{(g_1-g_2)^2+4g_3^2},2g_3]^t=\frac{1}{\sqrt{2\gamma(\gamma-\alpha)}}[\alpha-\gamma,\beta]^t,\end{aligned} \tag{12}$$

$$\alpha=g_1-g_2,\ \gamma=\sqrt{\alpha^2+\beta^2}=\sqrt{(g_1-g_2)^2+4g_3^2}\ \text{and}\ \beta=2g_3$$

The marix $\left[\Phi\right]$ consists of the eigenvectors $\vec{\Phi}_1=[\Phi_{11},\Phi_{21}]^t$ and $\vec{\Phi}_2=[\Phi_{12},\Phi_{22}]^t$:

$$\left[\Phi\right]=\begin{bmatrix}\vec{\Phi}_1^t\\ \vec{\Phi}_2^t\end{bmatrix}=\begin{bmatrix}\Phi_{11} & \Phi_{21}\\ \Phi_{12} & \Phi_{22}\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}\dfrac{\alpha+\gamma}{\sqrt{\gamma^2+\alpha\gamma}} & \dfrac{\beta}{\sqrt{\gamma^2+\alpha\gamma}}\\ \dfrac{\alpha-\gamma}{\sqrt{\gamma^2-\alpha\gamma}} & \dfrac{\beta}{\sqrt{\gamma^2-\alpha\gamma}}\end{bmatrix}. \tag{13}$$

Then the corresponding transposed matrix is:

$$\begin{aligned}\left[\Phi\right]^t&=[\vec{\Phi}_1,\vec{\Phi}_2]=\begin{bmatrix}\Phi_{11} & \Phi_{12}\\ \Phi_{21} & \Phi_{22}\end{bmatrix}=\\ &=\frac{1}{\sqrt{2}}\begin{bmatrix}\dfrac{\alpha+\gamma}{\sqrt{\gamma^2+\alpha\gamma}} & \dfrac{\alpha-\gamma}{\sqrt{\gamma^2-\alpha\gamma}}\\ \dfrac{\beta}{\sqrt{\gamma^2+\alpha\gamma}} & \dfrac{\beta}{\sqrt{\gamma^2-\alpha\gamma}}\end{bmatrix}\end{aligned} \tag{14}$$

The elements Φ_{ij} of the matrix $\left[\Phi\right]$ could be represented as a function of the angle θ, on which the coordinate system, defined by the vectors $\vec{\Phi}_1$ and $\vec{\Phi}_2$ is rotated in respect to the original coordinate system. In this case:

$$\left[\Phi(\theta)\right]=\begin{bmatrix}\Phi_{11}(\theta) & \Phi_{21}(\theta)\\ \Phi_{12}(\theta) & \Phi_{22}(\theta)\end{bmatrix}=\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}, \tag{15}$$

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

where

$$=arctg\left(\frac{2g_3}{(g_2-g_1)+\sqrt{(g_1-g_2)^2+4g_3^2}}\right).$$

The elements Φ_{ij} of the matrix $[\Phi(\theta)]$ are:

$$\begin{aligned}\cos\theta&=\cos[arctg\left(\frac{\beta}{\alpha+\gamma}\right)]=\\ &=\frac{\alpha+\gamma}{\sqrt{2\gamma(\gamma+\alpha)}}=\frac{\beta}{\sqrt{2\gamma(\gamma-\alpha)}}\end{aligned}, \tag{16}$$

$$\begin{aligned}\sin\theta&=\sin[arctg\left(\frac{\beta}{\alpha+\gamma}\right)]=\\ &=\frac{\beta}{\sqrt{2\gamma(\gamma+\alpha)}}=-\frac{\alpha-\gamma}{\sqrt{2\gamma(\gamma-\alpha)}}.\end{aligned} \tag{17}$$

Since $tg2\theta=(2tg\theta)/(1-tg^2\theta)$, the angle θ is defined as:

$$\theta=\frac{1}{2}arc\,tg\left(\frac{\beta}{\alpha}\right)=\frac{1}{2}arctg\left(\frac{2g_3}{g_1-g_2}\right). \tag{18}$$

From (15) it follows, that:

$$\left[\Phi(\theta)\right]=\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}=\frac{\alpha+\gamma}{\sqrt{2\gamma(\alpha+\gamma)}}\begin{bmatrix}1 & \dfrac{\beta}{\alpha+\gamma}\\ -\dfrac{\beta}{\alpha+\gamma} & 1\end{bmatrix}. \tag{19}$$

In this case the eigenvectors are correspondingly [6]:

doi:10.15849/icit.2015.0008 © ICIT 2015 (http://icit.zuj.edu.jo/ICIT15)

$$\vec{\Phi}_1=[\cos\theta,\sin\theta]^t \text{ and } \vec{\Phi}_2=[-\sin\theta,\cos\theta]^t, \tag{20}$$

where the angle θ is defined by (18).

D. Calculation of the eigenvalues and the eigenvectors of matrices [*Y*] *and* [*Z*]

The characteristic equation of the matrices [*Y*] and [*Z*], defined in accordance with (8), is:

$$\lambda^2-(a^2+b^2+c^2+d^2)\lambda+(ad+bc)^2=0.\ \alpha=g_1-g_2,$$
$$\beta=2g_3,\ \gamma=\sqrt{\alpha^2+\beta^2}=\sqrt{(g_1-g_2)^2+4g_3^2} \tag{21}$$

On the basis of (20), (11) and (12) are calculated the values of λ_s, $\vec{U}_s$ and $\vec{V}_s$ for $s=1,2$:

$$\lambda_{1,2}=\frac{1}{2}\left[(a^2+b^2+c^2+d^2)\pm\sqrt{(a^2+c^2-b^2-d^2)^2+4(ab+cd)^2}\right]=$$
$$=\frac{1}{2}(\omega\pm\sqrt{\nu^2+4\eta^2})=\frac{1}{2}(\omega\pm A), \tag{22}$$

$$\vec{U}_1=\frac{1}{\sqrt{2(\nu^2+4\eta^2+\nu\sqrt{\nu^2+4\eta^2})}}[\nu+\sqrt{\nu^2+4\eta^2},2\eta]^t=$$
$$=\frac{1}{\sqrt{2A(A+\nu)}}[\nu+A,2\eta]^t, \tag{23}$$

$$\vec{U}_2=\frac{1}{\sqrt{2(\nu^2+4\eta^2-\nu\sqrt{\nu^2+4\eta^2})}}[\nu-\sqrt{\nu^2+4\eta^2},2\eta]^t=$$
$$=\frac{1}{\sqrt{2A(A-\nu)}}[\nu-A,2\eta]^t, \tag{24}$$

$$\vec{V}_1=\frac{1}{\sqrt{2(\mu^2+4\delta^2+\mu\sqrt{\mu^2+4\delta^2})}}[\mu+\sqrt{\mu^2+4\delta^2},2\delta]^t=$$
$$=\frac{1}{\sqrt{2B(B+\mu)}}[\mu+B,2\delta]^t, \tag{25}$$

$$\vec{V}_2=\frac{1}{\sqrt{2(\mu^2+4\delta^2-\mu\sqrt{\mu^2+4\delta^2})}}[\mu-\sqrt{\mu^2+4\delta^2},2\delta]^t=$$
$$=\frac{1}{\sqrt{2B(B-\mu)}}[\mu-B,2\delta]^t, \tag{26}$$

where:

$$\omega=a^2+b^2+c^2+d^2,\ \nu=a^2+c^2-b^2-d^2,\ \mu=a^2+b^2-c^2-d^2, \tag{27}$$

$$\eta=ab+cd,\ \delta=ac+bd,\ A=\sqrt{\nu^2+4\eta^2},\ B=\sqrt{\mu^2+4\delta^2}. \tag{28}$$

The direct SVD for a matrix of size 2×2 could be represented by the relation:

$$[X]=\begin{bmatrix} a & b\\ c & d\end{bmatrix}=[U][\Lambda]^{1/2}[V]^t=$$
$$=\sqrt{\lambda_1}[T_1]+\sqrt{\lambda_2}[T_2]=\sum_{s=1}^{2}\sqrt{\lambda_s}[T_s] \tag{29}$$

where $[U]=[\vec{U}_1,\vec{U}_2]$, $[\Lambda]=diag[\lambda_1,\lambda_2]$, $[V]=[\vec{V}_1,\vec{V}_2]$.

The eigen images of the matrix [*X*] are the matrices [T_1] and [T_2], defined by the relations:

$$[T_1]=\vec{U}_1\vec{V}_1^t=\frac{1}{\sqrt{4AB(A+\nu)(B+\mu)}}\begin{bmatrix}(\nu+A)(\mu+B) & 2(\nu+A)\delta\\ 2(\mu+B)\eta & 4\eta\delta\end{bmatrix}, \tag{30}$$

$$[T_2]=\vec{U}_2\vec{V}_2^t=\frac{1}{\sqrt{4AB(A-\nu)(B-\mu)}}\begin{bmatrix}(\nu-A)(\mu-B) & 2(\nu-A)\delta\\ 2(\mu-B)\eta & 4\eta\delta\end{bmatrix} \tag{31}$$

If the vectors $\vec{U}_s$ and $\vec{V}_s$ are defined in accordance with (20), the eigen images are:

$$[T_1]=\vec{U}_1\vec{V}_1^t=\begin{bmatrix}\cos\theta_1\\ \sin\theta_1\end{bmatrix}[\cos\theta_2,\sin\theta_2]=$$
$$\begin{bmatrix}\cos\theta_1\cos\theta_2 & \cos\theta_1\sin\theta_2\\ \sin\theta_1\cos\theta_2 & \sin\theta_1\sin\theta_2\end{bmatrix}, \tag{32}$$

$$[T_2]=\vec{U}_2\vec{V}_2^t=\begin{bmatrix}-\sin\theta_1\\ \cos\theta_1\end{bmatrix}[-\sin\theta_2,\cos\theta_2]=$$
$$=\begin{bmatrix}\sin\theta_1\sin\theta_2 & -\sin\theta_1\cos\theta_2\\ -\cos\theta_1\sin\theta_2 & \cos\theta_1\cos\theta_2\end{bmatrix}, \tag{33}$$

$$\theta_1=\frac{1}{2}arctg\left(\frac{2\eta}{\nu}\right),\quad \theta_2=\frac{1}{2}arctg\left(\frac{2\delta}{\mu}\right). \tag{34}$$

The <u>inverse SVD</u> for a matrix of size 2×2 is defined by the relation:

$$[\Lambda]^{1/2}=\begin{bmatrix}\lambda_1^{1/2} & 0\\ 0 & \lambda_2^{1/2}\end{bmatrix}=[U]^t[X][V], \tag{35}$$

where

$$[U]^t=\begin{bmatrix}\vec{U}_1^t\\ \vec{U}_2^t\end{bmatrix}=\begin{bmatrix}U_{11} & U_{21}\\ U_{12} & U_{22}\end{bmatrix}=$$
$$\frac{1}{\sqrt{2}}\begin{bmatrix}\frac{\nu+A}{\sqrt{A(A+\nu)}} & \frac{2\eta}{\sqrt{A(A+\nu)}}\\ \frac{\nu-A}{\sqrt{A(A-\nu)}} & \frac{2\eta}{\sqrt{A(A-\nu)}}\end{bmatrix}=\begin{bmatrix}\cos\theta_1 & \sin\theta_1\\ -\sin\theta_1 & \cos\theta_1\end{bmatrix}, \tag{36}$$

$$[V]=[\vec{V}_1,\vec{V}_2]=\begin{bmatrix}V_{11} & V_{12}\\ V_{21} & V_{22}\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}\frac{\mu+B}{\sqrt{B(B+\mu)}} & \frac{\mu-B}{\sqrt{B(B-\mu)}}\\ \frac{2\delta}{\sqrt{B(B+\mu)}} & \frac{2\delta}{\sqrt{B(B-\mu)}}\end{bmatrix}=$$
$$=\begin{bmatrix}\cos\theta_2 & -\sin\theta_2\\ \sin\theta_2 & \cos\theta_2\end{bmatrix}. \tag{37}$$

doi:10.15849/icit.2015.0008 © ICIT 2015 (http://icit.zuj.edu.jo/ICIT15)

The couple Direct/Inverse SVD could be then represented as follows:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \frac{1}{2\sqrt{2}} \left\{ \begin{array}{l} \sqrt{\dfrac{\omega + A}{AB(A+\nu)(B+\mu)}} \begin{bmatrix} \nu+A \\ 2\eta \end{bmatrix} [\mu+B, 2\delta] + \\ + \sqrt{\dfrac{\omega - A}{AB(A-\nu)(B-\mu)}} \begin{bmatrix} \nu-A \\ 2\eta \end{bmatrix} [\mu-B, 2\delta] \end{array} \right\} = \tag{38}$$

$$= \frac{1}{2\sqrt{2}} \left\{ \begin{array}{l} \sqrt{\dfrac{\omega + A}{AB(A+\nu)(B+\mu)}} \begin{bmatrix} (\nu+A)(\mu+B) & 2(\nu+A)\delta \\ 2(\mu+B)\eta & 4\eta\delta \end{bmatrix} + \\ + \sqrt{\dfrac{\omega - A}{AB(A-\nu)(B-\mu)}} \begin{bmatrix} (\nu-A)(\mu-B) & 2(\nu-A)\delta \\ 2(\mu-B)\eta & 4\eta\delta \end{bmatrix} \end{array} \right\}$$

or

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \sigma_1 \begin{bmatrix} \cos\theta_1 \cos\theta_2 & \cos\theta_1 \sin\theta_2 \\ \sin\theta_1 \cos\theta_2 & \sin\theta_1 \sin\theta_2 \end{bmatrix} + \\ + \sigma_2 \begin{bmatrix} \sin\theta_1 \sin\theta_2 & -\sin\theta_1 \cos\theta_2 \\ -\cos\theta_1 \sin\theta_2 & \cos\theta_1 \cos\theta_2 \end{bmatrix}. \tag{39}$$

$$\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \dfrac{\nu+A}{\sqrt{A(A+\nu)}} & \dfrac{2\eta}{\sqrt{A(A+\nu)}} \\ \dfrac{\nu-A}{\sqrt{A(A-\nu)}} & \dfrac{2\eta}{\sqrt{A(A-\nu)}} \end{bmatrix} \times \\ \times \begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} \dfrac{\mu+B}{\sqrt{B(B+\mu)}} & \dfrac{\mu-B}{\sqrt{B(B-\mu)}} \\ \dfrac{2\delta}{\sqrt{B(B+\mu)}} & \dfrac{2\delta}{\sqrt{B(B-\mu)}} \end{bmatrix} \tag{40}$$

$$\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 \end{bmatrix} \tag{41}$$

where $\sigma_1 = \sqrt{\lambda_1} = \sqrt{\dfrac{\omega + A}{2}}$ and $\sigma_2 = \sqrt{\lambda_2} = \sqrt{\dfrac{\omega - A}{2}}$. (42)

Check: if $a=b=c=d$, then $\omega = 4a^2$, $\mu = \nu = 0$, $\delta = \eta = 2a^2$, $A = B = \sqrt{4^2 a^4} = 4a^2$, $\theta_1 = \theta_2 = \pi/4$.

In this case the couple Direct/Inverse SVD is correspondingly:

$$\begin{bmatrix} a & a \\ a & a \end{bmatrix} = \sqrt{\frac{4a^2 + 4a^2}{2}} \begin{bmatrix} (\sqrt{2}/2)^2 & (\sqrt{2}/2)^2 \\ (\sqrt{2}/2)^2 & (\sqrt{2}/2)^2 \end{bmatrix} + \\ + \sqrt{\frac{4a^2 - 4a^2}{2}} \begin{bmatrix} (\sqrt{2}/2)^2 & -(\sqrt{2}/2)^2 \\ -(\sqrt{2}/2)^2 & (\sqrt{2}/2)^2 \end{bmatrix} = \\ = 2a\frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = a\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \tag{43}$$

$$\begin{bmatrix} 2a & 0 \\ 0 & 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} a & a \\ a & a \end{bmatrix}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = \\ = \frac{1}{2}\begin{bmatrix} 2a & 2a \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 4a & 0 \\ 0 & 0 \end{bmatrix}. \tag{44}$$

The equations above confirm the correctness of (39) and (41) for the SVD of size *2*×*2*, executed for the image matrix of same size and with constant brightness of the image pixels. From (39) it follows, that that for the representation of the matrix [*X*] of size *2*×*2* through SVD are needed four parameters altogether: σ_1, σ_2, θ_1 and θ_2, calculated on the basis of (22) and (32). Hence, the SVD of size *2*×*2* (SVD$_{2\times2}$), defined in accordance with (39), is not over-complete.

E. Energy distribution in the SVD components for a matrix of size 2×2

The energy of the matrix $[X] = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ (or of its quadratic Euclidean norm) is defined by the sum of the squares of its elements:

$$E_X = \sum_{i=1}^{2}\sum_{j=1}^{2} x_{i.j}^2 = a^2 + b^2 + c^2 + d^2 = \omega. \tag{43}$$

In correspondence with (5) and (39) the matrix [*X*] is represented as the sum of two components:

$$[X] = \sigma_1[T_1] + \sigma_2[T_2] = [C_1] + [C_2]. \tag{44}$$

$$[C_1] = \begin{bmatrix} c_{11}(1) & c_{12}(1) \\ c_{13}(1) & c_{14}(1) \end{bmatrix} = \\ = \sigma_1 \begin{bmatrix} \cos\theta_1 \cos\theta_2 & \cos\theta_1 \sin\theta_2 \\ \sin\theta_1 \cos\theta_2 & \sin\theta_1 \sin\theta_2 \end{bmatrix}, \tag{45}$$

$$[C_2] = \begin{bmatrix} c_{11}(2) & c_{12}(2) \\ c_{13}(2) & c_{14}(2) \end{bmatrix} = \\ = \sigma_2 \begin{bmatrix} \sin\theta_1 \sin\theta_2 & -\sin\theta_1 \cos\theta_2 \\ -\cos\theta_1 \sin\theta_2 & \cos\theta_1 \cos\theta_2 \end{bmatrix} \tag{46}$$

C_1 and C_2 are the eigen images of the matrix [*X*].

The energy of each eigen image $[C_1], [C_2]$ is respectively:

$$E_{C_1} = \sum_{i=1}^{2}\sum_{j=1}^{2} c_{i.j}^2(1) = \sigma_1^2 = \frac{\omega + A}{2}, \tag{47}$$

$$E_{C_2} = \sum_{i=1}^{2}\sum_{j=1}^{2} c_{i,j}^2(2) = \sigma_2^2 = \frac{\omega - A}{2}. \tag{48}$$

From the Parseval's theorem for energy preservation, ($E_X = E_{C_1} + E_{C_2}$) and from (47) and (48) it follows, that $E_{C_1} >> E_{C_2}$, i.e., the energy E_X of the matrix [*X*] is concentrated mainly in the first SVD component. The concentration degree is defined by the relation:

$$\xi = \frac{E_{C_1}}{E_{C_1} + E_{C_2}} = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} = \frac{\omega + A}{2\omega}. \tag{49}$$

In particular, for the case, when the matrix [*X*] is with equal values of the elements ($x_{i,j}=a$), from (39), (47), (48) and (49) is obtained $E_X = E_{C_1} = 4a^2$, $E_{C_2} = 0$ and $\xi = 1$. Hence, the total energy of the matrix [*X*] is concentrated in the first SVD component only.

IV. Hierarchical SVD for a Matrix of Size $2^N\times 2^N$

The hierarchical SVD (HSVD) for the image matrix $[X(N)]$ of size $2^n\times 2^n$ pixels ($N=2^n$) is implemented through multiple execution of the n-levels SVD on image blocks (sub-matrices) of size 2×2. Let the matrix $[X(4)]$ is of size $2^2\times 2^2$ ($N=2^2=4$). In this case the number of hierarchical levels is $n=2$.

In the first HSVD level ($r=1$), the matrix $[X(4)]$ is divided into 4 sub-matrices of size 2×2, as it is shown in the left part of Fig. 1. The elements of the sub-matrices are as follows:

$$[X(4)]=\begin{bmatrix}[X_1(2)] & [X_2(2)]\\ [X_3(2)] & [X_4(2)]\end{bmatrix}=\begin{bmatrix}\begin{bmatrix}a_1 & b_1\\ c_1 & d_1\end{bmatrix} & \begin{bmatrix}a_2 & b_2\\ c_2 & d_2\end{bmatrix}\\ \begin{bmatrix}a_3 & b_3\\ c_3 & d_3\end{bmatrix} & \begin{bmatrix}a_4 & b_4\\ c_4 & d_4\end{bmatrix}\end{bmatrix}. \quad (50)$$

On each sub-matrix $[X_k(2)]$ of size 2×2 ($k=1,2,3,4$) is applied $SVD_{2\times 2}$, in accordance with (39). As a result, it is decomposed into 2 components:

$$[X_k(2)]=\sigma_{1,k}[T_{1,k}(2)]+\sigma_{2,k}[T_{2,k}(2)]=[C_{1,k}(2)]+[C_{2,k}(2)] \quad (51)$$

From (45) and (46) it follows that each sub-matrix $[C_{m,k}(2)]$ of size 2×2 could be represented as shown below:

For $m=1$,

$$[C_{1,k}(2)]=\begin{bmatrix}c_{11}(1,k) & c_{12}(1,k)\\ c_{13}(1,k) & c_{14}(1,k)\end{bmatrix}= =\sigma_{1,k}\begin{bmatrix}\cos\theta_{1,k}\cos\theta_{2,k} & \cos\theta_{1,k}\sin\theta_{2,k}\\ \sin\theta_{1,k}\cos\theta_{2,k} & \sin\theta_{1,k}\sin\theta_{2,k}\end{bmatrix}, \quad (52)$$

For $m=2$,

$$[C_{2,k}(2)]=\begin{bmatrix}c_{11}(2,k) & c_{12}(2,k)\\ c_{13}(2,k) & c_{14}(2,k)\end{bmatrix}= =\sigma_{2,k}\begin{bmatrix}\sin\theta_{1,k}\sin\theta_{2,k} & -\sin\theta_{1,k}\cos\theta_{2,k}\\ -\cos\theta_{1,k}\sin\theta_{2,k} & \cos\theta_{1,k}\cos\theta_{2,k}\end{bmatrix}, \quad (53)$$

$$\sigma_{1,k}=\sqrt{\frac{\omega_k+A_k}{2}},\ \sigma_{2,k}=\sqrt{\frac{\omega_k-A_k}{2}},$$

$$\theta_{1,k}=\frac{1}{2}arctg\left(\frac{2\eta_k}{\nu_k}\right),\ \theta_{2,k}=\frac{1}{2}arctg\left(\frac{2\delta_k}{\mu_k}\right), \quad (54)$$

$$\omega_k=a_k^2+b_k^2+c_k^2+d_k^2,\ A_k=\sqrt{\nu_k^2+4\eta_k^2},\ \nu_k=a_k^2+c_k^2-b_k^2-d_k^2, \quad (55)$$

$$\eta_k=a_kb_k+c_kd_k,\ \mu_k=a_k^2+b_k^2-c_k^2-d_k^2,\ \delta_k=a_kc_k+b_kd_k. \quad (56)$$

The sub-matrices $[C_{m,k}(2)]$ of size 2×2 for $k=1,2,3,4$ compose the matrices $[C_m(4)]$, each of size 4×4 (for $m=1,2$):

$$[C_{m,k}(4)]=\begin{bmatrix}[C_{m,1}(2)] & [C_{m,2}(2)]\\ [C_{m,3}(2)] & [C_{m,4}(2)]\end{bmatrix}= =\begin{bmatrix}\begin{bmatrix}c_{11}(m,1) & c_{12}(m,1)\\ c_{13}(m,1) & c_{14}(m,1)\end{bmatrix} & \begin{bmatrix}c_{11}(m,2) & c_{12}(m,2)\\ c_{13}(m,2) & c_{14}(m,2)\end{bmatrix}\\ \begin{bmatrix}c_{11}(m,3) & c_{12}(m,3)\\ c_{13}(m,3) & c_{14}(m,3)\end{bmatrix} & \begin{bmatrix}c_{11}(m,4) & c_{12}(m,4)\\ c_{13}(m,4) & c_{14}(m,4)\end{bmatrix}\end{bmatrix}. \quad (57)$$

Hence, the SVD decomposition of the matrix $[X]$ in the first level is represented by two components only:

$$[X(4)]=[C_1(4)]+[C_2(4)]= =\begin{bmatrix}([C_{1,1}(2)]+[C_{2,1}(2)]) & ([C_{1,2}(2)]+[C_{2,2}(2)])\\ ([C_{1,3}(2)]+[C_{2,3}(2)]) & ([C_{1,4}(2)]+[C_{2,4}(2)])\end{bmatrix}. \quad (58)$$

In the Second HSVD level ($r=2$), on each matrix $[C_m(4)]$ of size 4×4 is applied 4 times $SVD_{2\times 2}$. Unlike the preceding 1st level, in the second level the $SVD_{2\times 2}$ is applied on the sub-martices $[C_{m,k}(2)]$ of size 2×2 each, whose elements are mutually interlaced and are defined in accordance with the scheme, shown in the right part of Fig. 1. Here the elements of the sub-matrices, on which is applied the $SVD_{2\times 2}$ in the first and second hierarchical level ($r=1,2$), are tinted in same color. As it could be seen, the elements of the sub-matrices of size 2×2 in the second level are not neighbors, but are placed at one element interval in horizontal and vertical directions. In result of the $SVD_{2\times 2}$ execution, in the second level each matrix $[C_m(4)]$ is decomposed into two components:

$$[C_m(4)]=[C_{m,1}(4)]+[C_{m,2}(4)]\ \text{for}\ m=1,2. \quad (59)$$

The full decomposition of $[X]$ is then represented as:

$$[X(4)]=[C_{1,1}(4)]+[C_{1,2}(4)]+[C_{2,1}(4)]+[C_{2,2}(4)]= =\sum_{m=1}^{2}\sum_{s=1}^{2}[C_{m,s}(4)], \quad (60)$$

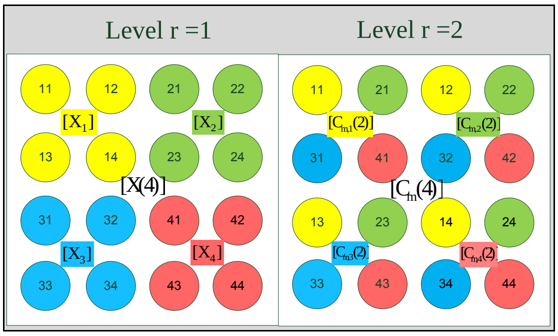

Fig. 1. The elements of the sub-matrices of size 2×2, over which is applied the $SVD_{2\times 2}$ in the 1st and 2nd HSVD levels

Hence, the decomposition of the image of size 4×4 comprises four components altogether. When the matrix $[X(8)]$ is of size $2^3\times 2^3$ ($N=2^3=8$ for $n=3$), three HSVD levels are executed through multiple applying of $SVD_{2\times 2}$ over the image blocks of size 2×2. In this case the total number of decomposition components is eight. In the first and second level the $SVD_{2\times 2}$ is executed in accordance with the scheme, shown on Fig. 1. In the third level, the $SVD_{2\times 2}$ is applied again on the sub-matrices of size 2×2. Their elements are defined in a way, similar with this, shown on Fig. 1. The only difference is that the elements of same color (i.e., belonging to same sub-matrix) are moved 3 elements away in horizontal and vertical directions. The presented HSVD algorithm could be generalized for the cases, when the image $[X(2^n)]$ is of size $2^n\times 2^n$ pixels:

$$[X(2^n)] = \sum_{p_1=1}^{2} \sum_{p_2=1}^{2} \dots \sum_{p_n}^{2} [C_{p_1,p_2,\dots p_n}(2^n)] \qquad (61)$$

In this case, the total number of levels is n, and the displacement in horizontal and vertical directions between the elements of the blocks of size 2×2 in the current level r, is correspondingly ($2^{r-1}-1$) elements for $r=1,2,..,n$.

V. COMPUTATIONAL COMPLEXITY OF THE HIERARCHICAL SVD WITH MATRIX *OF SIZE* $2^n\times2^n$

A. Computational complexity of SVD with a matrix of size 2×2

The computational complexity is defined on the basis of (39), taking into account the number of operations multiplication and addition, needed for the preliminary calculation of the components ω, μ, δ, ν, η, A, θ_1, θ_2, σ_1, σ_1, defined by Eqs. (27), (28), (34) and (42). Then:

- The number of multiplications needed for the calculation of (39), is $\Sigma_m = 39$;
- The corresponding number of additions is $\Sigma_s = 15$.

The total number of the needed algebraic operations for the execution of SVD of size 2×2, is:

$$SS_{SVD}(2\times2) = \Sigma_m + \Sigma_s = 54. \qquad (62)$$

B. Computational complexity of the hierarchical SVD with a matrix of size $2^n\times2^n$

The computational complexity of the hierarchical SVD is defined in similar way, as that for the $SVD_{2\times2}$. In this case, the number M of the sub-matrices of size 2×2, contained in the image of size $2^n\times2^n$, is $2^{n-1}\times2^{n-1}=4^{n-1}$, and the number of levels is n.

- The number of $SVD_{2\times2}$ in the first level is $M_1 = M = 4^{n-1}$;
- The number of $SVD_{2\times2}$ in the second level is $M_2=2M=2\times4^{n-1}$;
- ……………………
- The number of $SVD_{2\times2}$ in level n is $M_n=2^{n-1}M=2^{n-1}\times4^{n-1}$;

The total number of $SVD_{2\times2}$ is $M_\Sigma=M(1+2+..+2^{n-1})=4^{n-1}(2^n-1)=2^{2n-2}(2^n-1)$ correspondingly, and the total number of algebraic operations for the HSVD of size $2^n\times2^n$ is:

$$SS_{HSVD}(2^n\times2^n) = M_\Sigma . SS_{SVD}(2\times2) = 55\times2^{2n-2}(2^n-1). \qquad (63)$$

C. Computational complexity of SVD with a $2^n\times2^n$ matrix

For the calculation of matrices $[Y(N)]$ and $[Z(N)]$ each of size $N\times N$, when $N=2^n$, are needed $\Sigma_m = 2^{2n+2}$ multiplications and $\Sigma_s = 2^{n+1}(2^n-1)$ additions. The total number of operations is:

$$SS_{Y,Z}(N) = 2^{2n+2} + 2^{n+1}(2^n-1) = 2^{n+1}(3\times2^n-1). \qquad (64)$$

In accordance with the analysis, given in [23], the number of the operations $SS(N)$ needed for the iterative calculation of all N eigenvalues and the eigen N-dimensional vectors of the matrix of size $N\times N$ for $N=2^n$ with L iterations is:

$$\begin{aligned} SS_{val}(N) &= (1/6)(N-1)(8N^2+17N+42) = \\ &= (1/6)(2^n-1)(2^{2n+3}+17.2^n+42), \end{aligned} \qquad (65)$$

$$SS_{vec}(N) = N[2N(LN+L+1)-1] = 2^n[2^{n+1}(2^nL+L+1)-1]. \qquad (66)$$

From (3) and (4) it follows that two kinds of eigen vectors ($\vec{U}_s$ and $\vec{V}_s$) have to be calculated, so the number of operations needed for their definition in accordance with (64), should be doubled.

From the analysis of (1) it follows, that:

- The number of multiplications needed to calculate all components is: $\Sigma_m=2^n(2^{2n}+2^{2n})=2^{3n+1}$;
- The number of additions needed to calculate all components is: $\Sigma_s = 2^n-1$.

The global number of operations needed for the calculations in accordance with Eq. (1), is:

$$SS_D(N) = 2^{3n+1}+2^n-1 = 2^n(2^{2n+1}+1)-1 = 2^n(2^{2n+1}+1)-1. \qquad (67)$$

Hence, the global number of algebraic operations needed to calculate the SVD of size $2^n\times2^n$, is:

$$\begin{aligned} SS_{SVD}(2^n\times2^n) &= SS_{Y,Z}(2^n) + SS_{val}(2^n) + 2SS_{vec}(2^n) + \\ &+ SS_D(2^n) = 2^{2n+1}[2L(2^n+1)+2^{n-1}+5] + \\ &+ (1/6)(2^{2n+3}+17.2^n+42)-1. \end{aligned} \qquad (68)$$

D. Determination of the relative computational complexity of the HSVD

The relative computational complexity of the HSVD could be calculated on the basis of (62) and (68), from which is defined the relation below:

$$\begin{aligned} \psi_1(n,L) &= \frac{SS_{SVD}(2^n\times2^n)}{SS_{HSVD}(2^n\times2^n)} = \frac{1}{165.2^{2n-1}(2^n-1)} \times \\ &\times \begin{Bmatrix} 3.2^{n+1}[2^{n+2}(2^nL+L+1)+2^{n+1}(2^n+3)-3]+ \\ +(2^n-1)(2^{2n+3}+17.2^n+42)-6 \end{Bmatrix} \end{aligned} \qquad (69)$$

The computational complexity of the HSVD is defined by (69). For $n=2,3,4,5$ (i.e., for image blocks, of size 4×4, 8×8, 16×16 and 32×32 pixels) the values of $\psi_1(n,L)$ for $L=10$, obtained in accordance with (69), are given in Table 1. For big values of n the relation $\psi_1(n,L)$ does not depend on n and trends towards:

$$\psi_1(n,L)_{n\to\infty} \Rightarrow (16/165)(3L+1). \qquad (70)$$

TABLE 1. THE COEFFICIENT $\psi_1(n,L)$ OF THE RELATIVE LESSENING OF THE COMPUTATIONAL COMPLEXITY OF HSVD TOWARDS THE SVD FOR L=10.

n	2	3	4	5
$\psi_1(n,10)$	5.44	4.14	3.61	3.37

doi:10.15849/icit.2015.0008 © ICIT 2015 (http://icit.zuj.edu.jo/ICIT15)

Hence, for big values of n, in the case, when the number of iterations is $L \geq 4$ and $\psi(L)>1$, the computational complexity of the HSVD is lower than that of the SVD. Practically, the value of L is much larger than four. For the case, given here, $\psi_1(10) = 3$, i.e., the computational complexity of the HSVD is three times lower than that of the SVD.

VI. Representation of the HSVD algorithm through tree structure

The presented algorithm for 2-level HSVD with blocks of size 4×4 ($n=2$), represented by (60), could be generalized also for blocks of size $2^n\times2^n$. In this case, the matrix $[X]$ of each block could be represented by (61).

The number of the HSVD components is n. On the basis of the relations above, on Fig. 2 are shown the corresponding tree structures for the two-level case ($n=2$). As it could be seen from the figures, in accordance with (61) the HSVD algorithm is represented as a binary tree. For a HSVD with a block of size 8×8, the binary tree should be of levels ($n=3$), while for the tree with three nodes, two levels only are enough. This means, that for the second case the computational complexity is lower.

Each branch of the trees, shown on Fig. 2, has a corresponding eigenvalue $\lambda_{s,k}$, or $\sigma_{s,k}=\sqrt{\lambda_{s,k}}$ for the level 1, and $\lambda_{s,k}(m)$ or $\sigma_{s,k}(m)=\sqrt{\lambda_{s,k}(m)}$ for the level 2 respectively. The total number of branches in the tree from Fig. 2 is equal to 24. A part of the branches in each level could be cut-off, if for them the condition: $\sigma_{s,k} \cup \sigma_{s,k}(m)=0$, is satisfied, or if their values are close to zero.

To remove one component $[C]$ from given HSVD level, it is necessary all values of σ in this component to be equal or close to zero. In result, the decomposition for the corresponding branch could be stopped before it had reached the last level (n). In this way the HSVD algorithm is adapted in respect to the block contents. In this sense the HSVD algorithm is adaptive and easily adjustable to the requirements of each application.

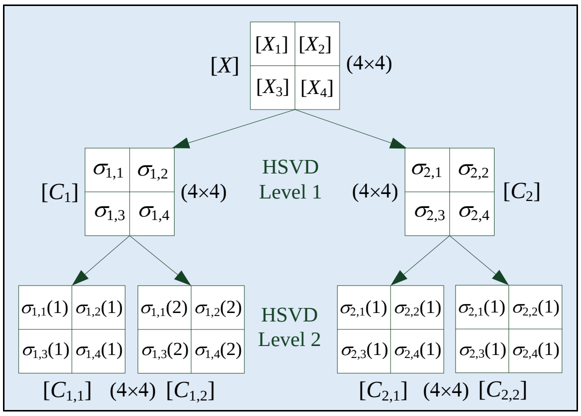

Fig. 2. Representation of the 2-level HSVD algorithm through binary tree

VII. Conclusions

From the analysis of the presented HSVD algorithm it follows that its basic advantages to SVD are:

1. Its computational complexity, represented as a full tree (without truncation), for a matrix of size $2^n\times2^n$ ($n=2$) is at least three times lower than that of the SVD, for similar matrix;
2. The HSVD algorithm is represented as a tree structure of n levels, which permits parallel and recursive processing of blocks of size 2×2 in each level. On each block in the corresponding decomposition level is applied the SVD, calculated by using simple algebraic relations;
3. The HSVD algorithm retains the SVD quality to concentrate the basic part of the image energy in the first decomposition components. After removal of the low-energy elements, the restored matrix has minimum mean square error and is an optimal approximation of the original;
4. The tree structure of the HSVD algorithm (a binary tree) facilitates the ability to stop the decomposition in one or more of the tree branches, for which the corresponding eigenvalue is zero, or approximately zero. In result, the HSVD computational complexity is additionally reduced compared to that of the "classic" SVD;
5. The HSVD algorithm could be easily generalized for matrices of any size (not for $2^n\times2^n$ only). In these cases the matrix should be divided into blocks of size 8×8, and on each to be applied the HSVD, i.e., will be executed a decomposition of eight components. Beforehand, all incomplete boundary blocks should be expanded through extrapolation. This approach is feasible, when the number of decomposition components, limited up to 8, is sufficient for the application. To increase the number of the HSVD components, the image should be divided into blocks of size 16×16 or larger;
6. The HSVD algorithm opens new opportunities for image processing in various application areas, such as: compression, filtration, segmentation, merging and digital watermarking, extraction of minimum number of features, sufficient for the objects recognition, etc.

References

[1] R. Vaccaro, Ed., SVD and Signal Processing II, Elsevier, New York, 1991.
[2] M. Moonen and B. de Moor, SVD and Signal Processing III, Elsevier, New York, 1995.
[3] I. Jolliffe, Principal Component Analysis, 2nd ed., Springer-Verlag, New York, 2002.
[4] A. Hyvarinen, J. Hurri, P. Hoyer, Natural image statistics, a probabilistic approach to early computational vision, Springer, 2009.
[5] P. Fieguth, Statistical image processing and multidimensional modeling, Springer, Science+Business Media, 2011.
[6] E. Carlen, Calculus++, Chapter 3: The Symmetric Eigenvalue Problem, Georgia Tech, 2003.
[7] H. Andrews and C. Patterson, Singular Value Decompositions and Digital Image Processing, IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-24, 1976, pp. 26–53.
[8] G. Stewart, The decompositional approach to matrix computation, Computing in Science and Engineering, Vol. 2, No. 1, 2000, pp. 50–59.

[9] J. Gerbrands, On the relationships between SVD, KLT, and PCA, Pattern Recognition, Vol. 14, No. 6, 1981, pp. 375–381.
[10] D. Kalman, A Singularly Valuable Decomposition: The SVD of a Matrix, The College Mathematics J., Vol. 27, No. 1, Jan. 1996, pp. 2-23.
[11] S. Orfanidis, SVD, PCA, KLT, CCA, and All That, Rutgers University Electrical & Computer Engineering Dept., Optimum Signal Processing, 2007, pp. 1-77.
[12] R. Sadek, SVD Based Image Processing Applications: State of The Art, Contributions and Research Challenges, Intern. J. of Advanced Computer Science and Applications, Vol. 3, No. 7, 2012, pp. 26-34.
[13] A. Householder, The Theory of Matrices in Numerical Analysis, New York: Dover, 1975.
[14] K. Diamantaras and S. Kung, Principal Component Neural Networks, Wiley, New York, 1996.
[15] A. Levy and M. Lindenbaum, Sequential Karhunen-Loeve Basis Extraction and its Application to Images. IEEE Trans. on Image Processing, Vol. 9, No. 8, August 2000, pp. 1371-1374.
[16] E. Drinea, P. Drineas, P. Huggins, A Randomized Singular Value Decomposition Algorithm for Image Processing Applications, "Proc. of the 8th Panhellenic Conference on Informatics", Y. Manolopoulos and S. Evripidou (Eds.), Nicosia, Cyprus, Nov. 2001, pp. 278-288.
[17] V. Rokhlin, A. Szlam and M. Tygert, A randomized algorithm for principal component analysis. SIAM J. Matrix Anal. Appl. 31, 2009, pp. 1100-1124.
[18] M. Holmes, A. Gray and C. Isbell, QUIC-SVD: Fast SVD using Cosine trees. Proc. of NIPS. 2008, pp. 673-680.
[19] B. Foster, S. Mahadevan, and R. Wang, A GPU-based Approximate SVD Algorithm. 9th Intern. Conference on Parallel Processing and Applied Mathematics, 2011, LN in CS, Vol. 7203, 2012, pp 569-578.
[20] M. Yoshikawa, Y. Gong, R. Ashino, R. Vaillancourt, Case study on SVD multiresolution analysis, CRM-3179, Jan. 2005, pp. 1-18.
[21] P. Waldemar and T. Ramstad, Hybrid KLT-SVD image compression, IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing, IEEE Comput. Soc. Press, Los Alamitos, 1997, pp. 2713-2716.
[22] M. Aharon, M. Elad and A. Bruckstein, The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation, IEEE Trans. on Signal Processing 54, 2006, pp. 4311-4322.
[23] V. Reha, M. Jeyakumar, Singular Value Decomposition based Image Coding for Achieving Additional Compression to JPEG Images, Intern. J. of Image Processing and Vision Sciences, Vol. 1, No 2, 2012, pp. 56-61.
[24] L. De Lathauwer, B. De Moor, and J. Vandewalle, A Multilinear Singular Value Decomposition. SIAM J. of Matrix Analysis and Applications, 21, 2000, pp. 1253-1278.
[25] W. Press, S. Teukolsky and W. Vetterling, Numerical Recipes in C, The Art of Scientific Computing, 2nd Ed., Cambridge University Press, 2002.
[26] K. Dickson, Z. Liu and J. McCanny, QRD and SVD processor design based on an approximate rotations algorithm, IEEE Workshop on Signal Processing Systems, SIPS, 2004, pp. 42-47.
[27] Y. Liu, C. Bouganis, P. Cheung, P. Leong and S. Motley, Hardware Efficient Architectures for Eigenvalue Computation, Proc. Design, Automation and Test in Europe (DATE'06), 2006, pp. 953-958.
[28] R. Kountchev, K. Nakamatsu, Adaptive multi-level 2D Karhunen-Loeve-based transform for still images, International Journal of Reasoning-Based Intelligent Systems (IJRIS), Vol. 6, Nos. 1/2, 2014, pp. 49-58.
[29] R. Kountchev, Applications of the Hierarchical Adaptive PCA for Processing of Medical CT Images, Egyptian Computer Science Journal, Vol. 37, No 3, May 2013, pp. 1-25.
[30] R. Kountchev, R. Kountcheva, Adaptive Hierarchical KL-based Transform: Algorithms and Applications, Chapter in “Computer Vision in Advanced Control Systems: Mathematical Theory”, M. Favorskaya and L. Jain (Eds.), Springer, Vol. 1, 2015, pp. 91-136.