Complexity Evaluation of Tensor Decomposition through 3D Inverse Spectrum Pyramid in respect of Deterministic Orthogonal Transforms

ROUMEN KOUNTCHEV RUMEN MIRONOV ROUMIANA KOUNTCHEVA
Radio Communications and Video Technologies
Technical University of Sofia TK Engineering
Sofia1756, Bul. Kl. Ohridsky 8 Sofia 1712, Mladost 3, 12
BULGARIA
rkountch@tu-sofia.bg rmironov@tu-sofia.bg kountcheva_r@yahoo.com
http://www.tu-sofia.bg

Abstract: - Recently, orthogonal 3D tensor decompositions are widely involved in the processing of various kinds of 3D data such as multimedia signals, correlated image sequences, etc. The methods for tensor decomposition could be divided into two main groups: statistical, based on various modifications of the Principal Component Analysis and the Singular Value Decomposition, and deterministic, based on the pyramidal 3D Discrete Wavelet Transform decompositions, Curvelet/Contourlet Discrete Transform and the Shearlet Discrete Transform. The methods from the first group surpass these from the second in the higher decorrelation of the decomposition components, but these from the second group have much lower computational complexity. In this work are compared the structures and is evaluated the computational complexity of the tensor decompositions, mentioned above, with the decomposition developed by the authors, which is based on the Reduced 3D Inverse Spectrum Pyramid (3D-RISP). The comparison results show that: the global computational complexity of the 3D-RISP is much lower than that of the remaining pyramidal decompositions; the compared schemes for recursive calculation have similar structures but this of the RISP does not need operations of the kind decimation and interpolation which are the reason for distortions in the reconstructed tensors.

Key-Words: - Deterministic pyramidal decompositions, Computational complexity, Recursive calculation, 3D Inverse Spectrum Pyramid, 3D Walsh-Hadamard transform

1 Introduction

The methods for tensor decomposition are recently particularly topical in the 3D data processing. Depending on the kind of the used orthogonal transforms, the methods could be divided into two main groups: statistical and deterministic. In the first case, when methods are based on the eigen-decomposition approach, are used various extensions of the Principal Component Analysis (PCA) for higher-order data. Most famous are the following methods: the CANDECOMP/PARAFAC Decomposition (CPD), the Higher Order PCA (HoPCA), the Hierarchical Tucker (HT) decomposition, the Tensor Train PCA (TT-PCA), the Multilinear PCA (MPCA), the Hierarchical Tensor PCA (HT-PCA), the Tensor Singular Value Decomposition (t-SVD), etc. In this way is achieved very high decorrelation of the decomposition components, maximum energy concentration in the first components and minimum mean square error due to limitation of the used components [1,2,3]. The statistical methods are executed through iterative algorithms for calculation of eigen vectors which have relatively high computational complexity. The deterministic methods for tensor decomposition are based on various 3D orthogonal transforms (DFT, DCT, DWT, DHT, etc.) [4,5] through which is achieved much lower computational complexity at the expense of lower decorrelation of the decomposition components. The choice of a method from these two groups is determined of the requirements imposed by the needed application for 3D data processing, for example: hyperspectral imaging [6], high resolution videos [7], medical imaging (EEG, MRI, fMRI) [8], biometrics [9,10], chemometrics [11], social network analysis [12,13], etc. The main feature of the algorithm for the execution of each tensor decomposition method is its Computational Complexity (CC) defined by the number of needed mathematical operations "addition" and

"multiplication". They determine to a high degree the needed execution time and memory which restrict the application areas of the algorithm.

In this work is evaluated and compared the CC of tensor decompositions grounded on some basic methods for deterministic 3D orthogonal transforms. The comparison covers the following methods: the 3D Discrete Fourier Transform (3D-DFT) [14,15,16]; the 3D Discrete Wavelet Transform (3D-DWT) [17,18,19]; pyramidal transforms with similar structure: 3D Curvelet and the Contourlet Discrete Transform (3D-CDT) [20,21,22,23] and the 3D Shearlet Discrete Transform (3D-SDT) [24]. As a basis for the comparison was selected the tensor decomposition through 3D Inverse Spectrum Pyramid (3D-ISP) [25,26] with 3D Walsh-Hadamard Transform (3D-WHT) [27,28]. In Sections 2-7 is given the evaluation and the comparative analysis of their CC.

2 CC of tensor decomposition based on the 3D Discrete Fourier Transform

For the calculation of the 3D-DFT is used the well-known Fast Fourier Transform (FFT) [14,16]. The separability of 3D-DFT permits the decomposition of the tensor $\boldsymbol{X}$ of size N×N×N ($N=2^m$) to be executed through applying 1D-DFT on each of the vectors which comprise the tensor: first, in direction x (Fig.1a), after that in direction y (Fig.1b) and finally - in direction z (Fig.1c) [16]. On the other hand, the calculation of the 1D-DFT is based on the Cooley-Tukey algorithm, called Fast Fourier Transform (FFT). After applying the radix-2 FFT algorithm on the N-dimensional vector, the number of needed operations is reduced from N^2 to $N\log_2 N=Nm$. Then, the CC of the 3D-FFT becomes $O(N_c\log_2 N_c)$, where $N_c=N_1\times N_2\times N_3$.

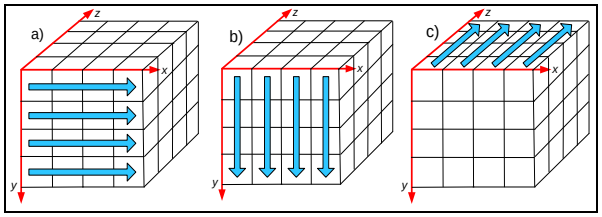


Fig. 1.a,b,c. Calculation of the 3D-DFT through applying 1D-FFT on the vectors of the tensor $\boldsymbol{X}$ in three directions (x,y,z).

On condition that $N_1=N_2=N_3=N=2^m$, the number of additions needed to calculate the tensor $\boldsymbol{X}$ of N^3 voxels is $A=N^3\log_2 N^3=3N^3m$ and the number of complex multiplications is correspondingly $M=(N/2)^3\log_2 N^3=(3/8)N^3m$.

Then, the normalized number of additions and complex multiplications for one voxel is:

$$A^0_{3D-FFT}=A/N^3=3m,\ M^0_{3D-FFT}=M/N^3=(3/8)m. \quad (1)$$

For the real data calculation the algorithm split-radix 1D-FFT [17] requires (N/2)(3m-5)+4 real additions and (N/2)(m-3)+2 real multiplications. Hence, for a tensor of N^3 voxels $A=(N/2)^3(3\log_2 N^3-5)+4=(N^3/8)(9m-5)+4$ and $M=(N/2)^3(3m-3)+2=(3N^3/8)(m-1)+2$. Then, the normalized number of real additions and multiplications is:

$$A^0_{3D-FFT}=(1/8)(9m-5)+4/N^3\approx(1/8)(9m-5),$$
$$M^0_{3D-FFT}=(3/8)(m-1)+2/N^3\approx(3/8)(m-1) \quad (2)$$

3 CC of 3D-DWT in the tensor decomposition

According to Mallat's pyramid algorithm [19] the 3D-DWT is executed through consecutive applying 1D-DWT along columns, rows and slices of tensor $\boldsymbol{X}_p$ in each decomposition level (p=0,1,..,m-1). At that in the level p=1 the sizes of $\boldsymbol{X}_1$ (i.e. $2^{m-1}\times2^{m-1}\times2^{m-1}$) are twice smaller than these of $\boldsymbol{X}_0$ ($2^m\times2^m\times2^m$) in the level p=0 for $N=2^m$, etc. The elements $x_p(i,j,k)$ for i=0,1,..,N-1 in the row j of the slice k of $\boldsymbol{X}_p$ are transformed column-by-column through 1D-DWT with 1D high-pass filters (h) of (2q+1) coefficients, and low-pass filters (g) of (2t+1) coefficients each:

$$x_p(i,j,k)=\sum_{r=-q}^{q}h(r)x_{p-1}(i-r,j,k),$$
$$d_p(i,j,k)=\sum_{r=-t}^{t}g(r)x_{p-1}(i-r,j,k) \quad (3)$$

Here $x_{p-1}(i,j,k)$ is one element of the tensor $\boldsymbol{X}_{p-1}$ in the pyramid level (p-1). The elements $x_p(i,j,k)$ of $\boldsymbol{X}_p$ calculated in accordance with Eq. (3) are transformed in similar way row by row for j=0,1,..,N-1, and after that - between the slices- for k=0,1,..,N-1. On Fig. 2 is shown the recursive scheme for 1D-DWT calculation in the pyramid level p=0 [17].

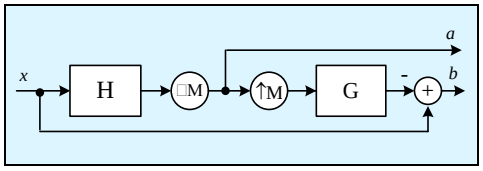


Fig. 2. Recursive calculation of 1D-DWT (p=0)

For the decomposition of tensor $\boldsymbol{X}$ of size N×N×N ($N=2^m$) through the 3D Wavelet Pyramid (3D-WP) of m levels which uses a bank of separable digital filters with 3 and 5 coefficients (1/2)(1,2,1) and (1/8)(-1,2,6,2,-1), is needed the following number of additions and multiplications:

- for p=0: $A_0=18N^3$; $M_0=24N^3$;
- for p=1: $A_1=18N^3(1/2^3)$; $M_1=24N^3(1/2^3)$;

- for p : $A_p=18N^3(1/2^{3p})$; $M_p=24N^3(1/2^{3p})$;
- for p=m-1: $A_{m-1}=18N^3(1/2^{3m-3})$; $M_{m-1}=24N^3(1/2^{3m-1})$.

Then, the total number of additions and multiplications for 3D-DWT(3,5) of m levels, is:

$$A_{3D-DWT}(m)=\sum_{p=0}^{m-1}A_p=18N^3\sum_{p=0}^{m-1}8^{-p}=20.5(N^3-1);$$

$$M_{3D-DWT}(m)=\sum_{p=0}^{m-1}M_p=24N^3\sum_{p=0}^{m-1}8^{-p}=27.4(N^3-1)$$

For m=2 $\sum_0^1 A_p=1296$; $\sum_0^1 M_p=1726$.

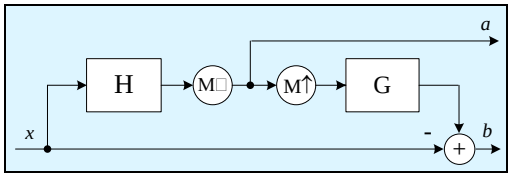
For m levels: $\sum_{p=0}^{m-1}A_p\approx 20.5N^3$, $\sum_{p=0}^{m-1}M_p\approx 27.4N^3$.

In this case, the normalized number of additions and multiplications for one voxel is:

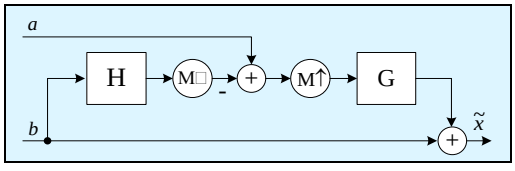
$$A^0_{3D-DWT}=\frac{1}{N^3}\sum_{p=0}^{m-1}A_p=20.5,\ M^0_{3D-DWT}=\frac{1}{N^3}\sum_{p=0}^{m-1}M_p=27.4 \quad (4)$$

4 Tensor decomposition through multiscale pyramid and multi-directional filter bank

Towards tensor decomposition methods based on the multiscale pyramid and multidirectional filter banks can also be related several transforms with similar structures, for example Curvelet, Contourlet and Shearlet Discrete Transforms [20,21,22,23]. The Contourlet Discrete Transform (CDT) [20] comprises two basic steps. In the first step is executed the tensor $\boldsymbol{X}$ decomposition through Laplacian Pyramid (LP). On Fig. 3 a,b are shown the schemes for tensor decomposition and reconstruction in one level of the LP [20].



a) LP analysis: the coarse approximation, *a*;



b) LP reconstruction and the difference *b* between the original signal and the prediction.

Fig. 3. One-level Laplacian Pyramid (LP)

The used abbreviations are: M↓ / M↑ - sampling / interpolation matrix; H and G - analysis (low pass) and synthesis filters.

The 3D Laplacian Pyramid (3D-LP) is built in the frequency 3D space of the tensor transformed through FFT. The Fourier space is divided into slices and sub-bands. In the second step, on each sub-band is applied 3D Directional Filter Bank (3D-DFB) of quincunx filters. For their implementation are used fan filters [22]. In the first step of the 3D Fast Discrete Curvelet Transform (3D-FDCurT) is used Unequally-Spaced FFT (USFFT) [23] which needs $O(N^3\log_2 N)$ flops. The structure of the 3D Shearlet Discrete Transform (3D-SDT) is similar to that of the 3D-FDCurT.

The 3D-SDT algorithm comprises the following basic steps [24]:

- applying 3D Odd-Frequency DFT (3D OF-DFT) on the input image, using 3D-FFT;
- analysis of the OF-DFT using 3D-LP filter bank with sub-bands;
- applying 3D-DFB to the resulting sub-bands;
- shearing the sub-bands: in result, the spectrum content of each band is placed in the frequency space center;
- two dimensional sub-sampling aimed at the zero-DFT coefficients removal;
- inverse 3D OF-DFT (3D IOF-DFT) computation of each sub-band using the inverse 3D-FFT.

The Direct 3D OF-DFT is defined as follows [24]:

$$s(u,v,l)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}\{x(i,j,k)\exp[-(\pi\sqrt{-1}/N)(i+j+k)]\}\times$$
$$\times\exp[-(2\pi\sqrt{-1}/N)(iu+jv+kl)] \quad (5)$$

To calculate the 3D OF-DFT for the pre-modulated image $x(i,j,k)\exp[-(\pi\sqrt{-1}/N)(i+j+k)]$ is used 3D-FFT. After that, the decomposition of each sub-band is executed through 3D-DFB based on Meyer wavelet filters [24]:

$$H_i(\omega)=\begin{cases}1, & |\omega|\le\pi/4\\ \cos\{(\pi/2)v[(4|\omega|/\pi)-1]\}, & \pi/4\le|\omega|\le\pi/2;\\ 0, & \text{elsewhere}\end{cases}$$

$$G_i(\omega)=\begin{cases}0, & |\omega|\le\pi/4;\\ \sin\{(\pi/2)v[(4|\omega|/\pi)-1]\}, & \pi/4\le|\omega|\le\pi/2;\\ 1, & \text{elsewhere.}\end{cases}$$

where i=1,2; ν(.) is function satisfying: ν(ω)=0 for ω≤0; ν(ω)=1 for ω≥1, and $\nu(\omega)+\nu(1-\omega)=1$.

The CC of the algorithm 3D-SDT based on 3D-LP of m levels in the spectrum space of the 3D-FFT is defined through the following relations:

- for the level p=0:

$$A_0=2^{3m+1}+3\times 2^{3m}m+3\times 16(2^{3m}-2^{3m}/8)=2^{3m}(5m+44)$$

$$M_0=2^{3m}+(3/8)2^{3m}m+3\times 18(2^{3m}-2^{3m}/8)=(3/8)^{3m}(m+128.6).$$

- for the level p: $A_p=2^{3(m-p)}[5(m-p)+44]$;

$$M_p=(3/8)2^{3(m-p)}[(m-p)+128.6].$$

Then, the global number of additions and multiplications needed for the execution of 3D-SDT pyramid of m levels, is:

$$A_{3D-SDT}=\sum_{p=0}^{m-1}A_p=(8/7)2^{3m}\{(5m+44)[1-(1/8)^m]-(8/7)\times$$
$$\times[(1/8)^{m-1}((1/8)(m-1)-m)+1]\}\approx 2^{3m}(5m+43)$$

$$M_{3D-SDT}=\sum_{p=0}^{m-1}M_p=(3/8)2^{3m}\{(8/7)m[1-(1/8)^m]+$$
$$+128.6m-(1/49)[(1/8)^m((m-1)-m(1/8)^{m-1}+1]\}\approx 2^{3m}48.6m$$

In this case, for one voxel is got (6):

$$A^0_{3D-SDT}=A/N^3=5m+43;\quad M^0_{3D-SDT}=M/N^3=48.6m.$$

5 Tensor decomposition through 3D Inverse Spectrum Pyramid based on the 3D Walsh-Hadamard Transform

The direct/inverse 3D-WHT in scalar form is defined in accordance with the relations [26,27]:

$$s(u,v,l)=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}x(i,j,k)wal(i,u,N)wal(j,v,N)wal(k,l,N) \quad (7)$$

for u,v,l=0,1,..,N-1

$$x(i,j,k)=\frac{1}{N^3}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\sum_{l=0}^{N-1}s(u,v,l)wal(i,u,N)wal(j,v,N)wal(k,l,N) \quad (8)$$

for i,j,k=0,1,..,N-1, where x(i,j,k) is one element of the tensor $\boldsymbol{X}$ of size N×N×N; s(u,v,l) is a spectrum coefficient which is an element of the spectrum tensor $\boldsymbol{S}$, of same size; $wal(z,w,N)=(-1)^{\sum_{r=0}^{m-1}q_r(z)w_r}$, $q_r(z)=z_{m-r}\oplus z_{m-r-1}$ for r=0,1,.,m-1, $q_0(z)=z_{m-1}$;

$$z=\sum_{r=0}^{m-1}z_r2^r \text{ and } w=\sum_{r=0}^{m-1}w_r2^r \text{ for } z/w=i/u \text{ or } j/v \text{ or } k/l.$$

The 3D-WHT is divisible and for its calculation can be used multiple 1D-WHT. For the 1D Fast WHT (1D-FWHT) the needed number of additions A_F is [27,28]:

$$A_F(m)=N\times\lg_2N=2^m m. \quad (9)$$

The number of addition A_{FT} needed for the 1D Fast Truncated WHT (1D-FTWHT) when a reduction (truncation) of the output coefficients from N down to 2 is done, is [26]:

$$A_{FT}(m)=N\sum_{p=0}^{m-1}2^{-p}=2(2^m-1). \quad (10)$$

The acceleration of the calculations for the 1D-FTWHT compared to 1D-FWHT is defined by the relation:

$$R_{1D}(m)=\frac{A_F(m)}{A_{FT}(m)}=\frac{m}{2(1-2^{-m})}\approx m/2. \quad (11)$$

The total number of A_{FT} for the tensor of size N×N×N after 3D-FTWHT is: $A^{3D}_{FT}(N)=2N^2[N(m+1)-1]$. The total number of additions needed for the 3D-WHT is $A^{3D}(N)=3N^3(N-1)$, but for 3D-FWHT it is reduced to $A^{3D}_F(N)=3N^3m$. Then, the normalized number of additions for one voxel is:

$$A^0_{FT}=(2/N)[N(m+1)-1] \text{ and } A^0_F=3m. \quad (12)$$

The acceleration of the 3D-FTWHT calculations compared to 3D-FWHT and 3D-WHT is respectively:

$$R_{3D}(N)=\frac{A^0_F}{A^0_{FT}}=\frac{3Nm}{2[N(m+1)-1]}\approx\frac{1.5m}{m+1} \quad (13)$$

For m=4 $R_{3D}(16)=1.2$;

$$R'_{3D}(N)=\frac{A^0}{A^0_{FT}}=\frac{3N(N-1)}{2[N(m+1)-1]}\approx\frac{1.5\times2^m}{m+1} \quad (14)$$

For m=4 $R'_{3D}(16)=4.8$.

The decomposition of the tensor $\boldsymbol{X}$ of size 8×8×8 through 3D-RISP pyramid of m=3 levels based on the Truncated 3D-WHT (3D-TWHT) is defined by the relation below [26]:

$$\boldsymbol{X}=\tilde{\boldsymbol{X}}+\tilde{\boldsymbol{E}}_0+\boldsymbol{E}_1 \text{ for the levels } p=0,1,2, \quad (15)$$

where $\boldsymbol{E}_0=\boldsymbol{X}-\tilde{\boldsymbol{X}}$, $\tilde{\boldsymbol{X}}=(1/8^3)\sum_{u=0}^{1}\sum_{v=0}^{1}\sum_{l=0}^{1}s(u,v,l)\boldsymbol{W}_{u,v,l}$, $\boldsymbol{E}^t_1=\boldsymbol{E}^t_0-\tilde{\boldsymbol{E}}^t_0$,

$$\tilde{\boldsymbol{E}}^t_0=(1/4^3)\sum_{u=0}^{1}\sum_{v=0}^{1}\sum_{l=0}^{1}s^t_0(u,v,l)\boldsymbol{W}_{u,v,l} \text{ for } t=1,2,..,8;$$

$$s(u,v,l)=\sum_{i=0}^{7}\sum_{j=0}^{7}\sum_{k=0}^{7}x(i,j,k)wal(i,u,8)wal(j,v,8)wal(k,l,8)$$

for p=0;

$$s^t_p(u,v,l)=\sum_{i=0}^{2^{3-p}-1}\sum_{j=0}^{2^{3-p}-1}\sum_{k=0}^{2^{3-p}-1}\tilde{e}^t_p(i,j,k)wal(i,u,2^{3-p})wal(j,v,2^{3-p})wal(k,l,2^{3-p})$$

for t=1,2,.., 8^p and p=1,2.

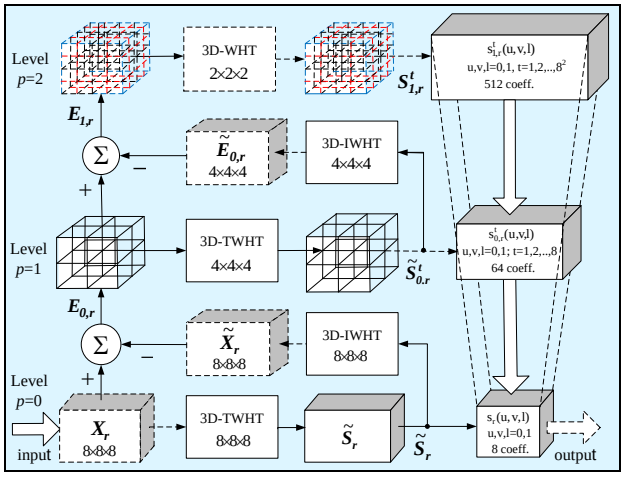
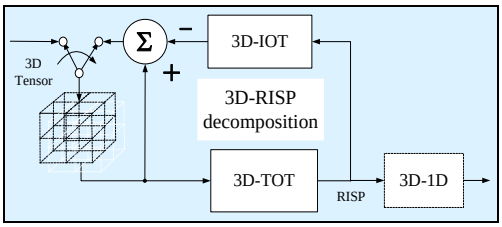


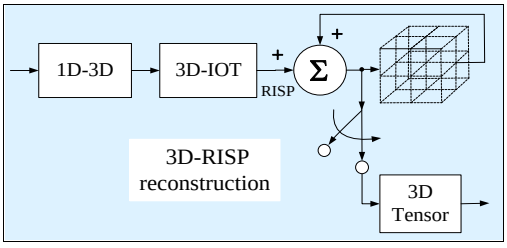
Fig. 4. ISP structure for the decomposition of tensor $\boldsymbol{X}$ of size 8×8×8

After truncation of coefficients $s_p^t(0,0,0)=0$ for levels p=1,2 is got the 3D Reduced ISP (3D-RISP) [25].

The block-diagram for tensor decomposition through 3D-RISP/TWHT of 3 levels is shown on Fig. 4. The reconstruction is implemented in reverse order. The orthogonal transforms (OT) used in the 3D-RISP are statistical or deterministic [4,5].



a) Block diagram of the recursive 3D-RISP decomposition



b) Block diagram of the recursive 3D-RISP reconstruction

Fig. 5. Block diagrams for the recursive 3D-RISP

On Fig. 5 a, b are shown the block diagrams for recursive 3D-RIDP calculation in one level of the pyramidal decomposition/reconstruction. In both cases could be used each of the mentioned OT. The used abbreviations are: Σ - Summator; 3D-TOT/3D-IOT - 3D Truncated Direct/3D Inverse Orthogonal Transform; 3D-1D/1D-3D and 3D-1D/1D-3D - Data Dimension Transform.

6 CC of 3D-RISP based on the Fast 3D-WHT

The computational complexity of the 3D-Fast RISP/WHT of m levels (number of additions/multiplications), is:

- for p=0: $A_0=2^{2m+1}[2^m(m+1)-1]+3m2^{3m}$; $M_0=2^0$;
- for p: $A_p=2^{2m+1+p}[2^{m-p}(m-p+1)-1]+3(m-p)2^{3m}$; $M_p=2^{3p}$;
- for p=m-2: $A_{m-2}=11\times 2^{3m-1}+3\times 2^{3m+1}$; $M_{m-2}=2^{3(m-2)}$;
- for p=m-1: $A_{m-1}=3\times 2^{3m}$; $M_{m-1}=0$.

The total number of additions/multiplications for the Fast 3D RISP/WHT of m levels is:

$$A_{3D-FRISP}(m)=\sum_{p=0}^{m-1}A_p=2^{3m}\{(m-1)[(5/2)m+7)]+4\};$$

$$M_{3D-FRISP}(m)=\sum_{p=0}^{m-1}M_p=(1/7)[2^{3(m-1)}-1].$$

The total number of additions/multiplications for 3D-RISP/WHT of m levels is:

$$A_{3D-RISP}(m)=\sum_{p=0}^{m-1}A_p=3\times 2^{3m+2}(2^{m+1}-1);$$

$$M_{3D-RISP}(m)=\sum_{p=0}^{m-1}M_p=(1/7)[2^{3(m-1)}-1].$$

Example for m=2: $A_{3D-FRISP}(2)=\sum_{p=0}^{1}A_p=768$; $A_{3D-RISP}(2)=\sum_{p=0}^{1}A_p=5376$. The acceleration of the 3D-FRISP computation compared to 3D-RISP in respect of A(m) and M(m) for m=2 is:

$$R_A(2)=A_{3D-RISP}(2)/A_{3D-FRISP}(2)=5376/768=7.$$

For a tensor $\boldsymbol{X}$ of N^3 voxels: $A_{3D\text{-}FRISP}\approx 2.5m^2N^3$ and $M_{3D\text{-}FRISP}\approx(1/56)N^3$. The normalized number of additions/multiplications is:

$$A^0_{3D-FRISP}=\frac{1}{N^3}\sum_{p=0}^{m-1}A_p\approx 2.5m^2;$$
$$M^0_{3D-FRISP}=\frac{1}{N^3}\sum_{p=0}^{m-1}M_p\approx 0.02 \qquad (16)$$

7 Comparison of the CC of 3D-RIDP, 3D-DWT, 3D-FFT and 3D-SDT

The results of the CC comparison for algorithms 3D RIDP, DWT, FFT and SDT are generalized in Tables 1, 2 and 3. On the basis of their information are created the graphics on Figs. 6 and 7.

Table 1. Acceleration R(m) of the computations for the Fast 3D-RISP/WHT towards Fast 3D DWT, FFT, SDT

Decomposition/ acceleration	$A^0(m)$	$M^0(m)$	$R_A(m)$	$R_M(m)$
Fast 3D-RISP	$2.5m^2$	0.02	$R_A(m)=1$	$R_M(m)=1$
Fast 3D-DWT	20.57	27.43	$R_A(m)=8.23/m^2$	$R_M(m)=1371$
Fast 3D-FFT	3m	(3/8)m	$R_A(m)=1.2/m$	$R_M(m)=18.7m$
Fast 3D-SDT	5m+43	48.6m	$R_A(m)\approx(2m+17)/m^2$	$R_M(m)=2430m$

Table 2. Comparison of $R_A(m)$ for the Fast 3D-RISP/WHT towards Fast 3D DWT, FFT, and SDT

$R_A(m)$ / $m\rightarrow$	2	3	4	5	6	7
Fast 3D-RISP	1	1	1	1	1	1
Fast 3D-DWT	2.05	0.91	0.51	0.33	0.22	0.16
Fast 3D-FFT	0.30	0.13	0.07	0.05	0.03	0.02
Fast 3D-SDT	5.25	2.55	1.56	1.08	0.80	0.63

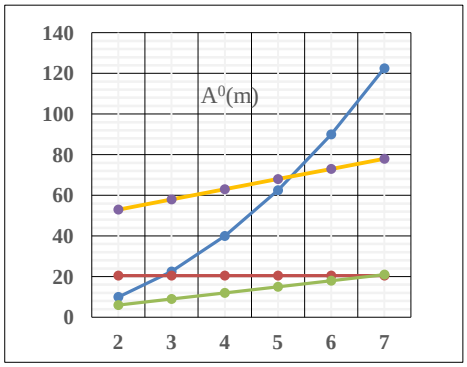


Fig. 6. Graphical relations of $A^0(m)$ for the Fast 3D RISP (blue), DWT (red), FFT (green), SDT (yellow)

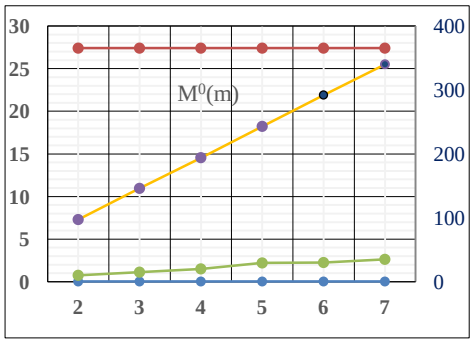


Fig. 7. Graphical relations $M^0(m)$ for the Fast 3D RISP (blue), DWT (red), FFT (green), SDT (yellow)

Table 3. Comparison of $R_M(m)$ for the 3D Fast RISP/WHT towards Fast 3D DWT, FFT, and SDT

$R_M(m)$ / $m \rightarrow$	2	3	4	5	6	7
Fast 3D-RISP	1	1	1	1	1	1
Fast 3D-DWT	1371	1371	1371	1371	1371	1371
Fast 3D-FFT	37.4	56.1	74.9	93.5	112	131
Fast 3D-SDT	4860	7020	9720	12150	14580	17010

The comparison of the investigated four kinds of 3D tensor decompositions shows that:

- regarding the number of "additions" A^0 (Fig. 6): together with the growing of m, the increase of A^0 is slowest for the 3D-FFT and fastest for the Fast 3D-RISP while for the Fast 3D-DWT the normalized number of additions is a constant. For m=3 the value of A^0 for the Fast 3D-RISP is close to that for the Fast 3D-SDT;
- regarding the number of "multiplications", M^0 (Fig. 7): together with the growing of m, for the Fast 3D-RISP it has a constant value (0.02) and for the Fast 3D-DWT - a constant maximum value (27.4) while for the Fast 3D-SDT and the 3D-FFT the growth is linear, with different speed;
- the structures used for the recursive calculation of the compared 3D orthogonal pyramidal transforms from Figs. 2, 3 and 5 are very close. The main advantage of 3D-RISP is that it does not use operations "decimation" and "interpolation" which cause distortions in the reconstructed tensors.

The comparison results show that the global CC (additions+multiplications) of the Fast 3D-RISP is significantly lower than that of the pyramidal decompositions Fast 3D-DWT and Fast 3D-SDT which are based on deterministic orthogonal transforms.

8 Conclusions

In this paper are analysed the schemes for recursive calculation and the computational complexity of the Fast 3D-RISP/WHT towards the deterministic transforms Fast 3D-DWT and Fast 3D-SDT used for the pyramidal tensor decomposition. The so obtained results show the higher efficiency of the Fast 3D-RISP in the general computational complexity as well as in the structures for recursive calculation. These qualities of the 3D-RISP open various possibilities for its application in areas where the lower computational cost is of high significance.

Acknowledgment

This work was supported by the National Science Fund of Bulgaria: Project No. KP-06-H27/16.

References:

[1] T. Kolda, B. Bader, Tensor decompositions and applications, *SIAM Review*, Vol. 51, No.3, 2009, pp. 455–500.

[2] A. Cichocki, N. Lee, I. Oseledets, A. Phan, Q. Zhao, D. Mandic, Tensor networks for dimensionality reduction and large-scale optimization: Part 1. *Foundations and Trends in Machine Learning*, Vol. 9, No. 4-5, 2016, pp. 249-429, https://doi.org/10.1561/2200000059.

[3] A. Zare, A. Ozdemir, M. Iwen, S. Aviyente, *Extension of PCA to Higher Order Data Structures: An Introduction to Tensors, Tensor Decompositions, and Tensor PCA*, arXiv:1803.00704v2 [eess.SP], 26 Jun 2018.

[4] A. Grigoryan, S. Agaian. *Multidimensional Discrete Unitary Transforms: Representation, Partitioning, and Algorithms*, Marcel Dekker, 2003.

[5] R. Wang. *Introduction to Orthogonal Transforms with Applications in Data Processing and Analysis*, Cambridge University Press, 2012.

[6] D. Letexier, S. Bourennane, J. Talon, Nonorthogonal tensor matricization for hyperspectral image ltering, *IEEE Geoscience*

and Remote Sensing Letters, Vol. 5, No. 1, 2008, pp. 3-7.

[7] T. Kim, R. Cipolla, Canonical correlation analysis of video volume tensors for action categorization and detection, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 31, No. 8, 2009, pp. 1415-1428.

[8] F. Miwakeichi, E. Montes, P. Sosa, N. Nishiyama, H. Mizuhara, Y. Yamaguchi, Decomposing EEG data into space-time-frequency components using parallel factor analysis, *NeuroImage*, Vol. 22, No. 3, 2004, pp. 1035-1045.

[9] D. Tao, X. Li, X. Wu, S. Maybank, General tensor discriminant analysis and Gabor features for gait recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 10, 2007, pp. 1700-1715.

[10] H. Fronthaler, K. Kollreider, J. Bigun, J. Fierrez, F. Fernandez, J. Garcia, J. Rodriguez, Fingerprint image quality estimation and its application to multi-algorithm verification, *IEEE Trans. on Information Forensics and Security*, Vol. 3, No. 2, 2008, pp. 331-338.

[11] H. Wu, J. Nie, Y. Yu, R. Yu, Multi-way chemometric methodologies and applications, *Analytica Chimica Acta*, Vol. 650, No. 1, 2009, pp. 131-142.

[12] Y. Zhang, M. Chen, S. Mao, L. Hu, V. Leung, Cap: Community activity prediction based on big data analysis, *IEEE Network*, Vol. 28, No. 4, 2014, pp. 52-57.

[13] K. Maruhashi, F. Guo, C. Faloutsos, Multiaspectforensics: Pattern mining on large-scale heterogeneous networks with tensor analysis, *Intern. Conf. on IEEE, Advances in Social Networks Analysis and Mining* (ASONAM), 2011, pp. 203-210.

[14] K. Rao, D. Kim, J. Hwang, *Fast Fourier Transform: Algorithms and Applications*, Springer, 2010.

[15] T. Sakai, S. Sedukhin, 3D Discrete Transforms with Cubical Data Decomposition on the IBM Blue Gene/Q, *Technical Report* 2013-001, Graduate School of Computer Science and Engineering, University of Aizu Tsuruga, Aizu-Wakamatsu City, Japan, Feb. 2013, pp. 1-31. https://doi.org/10.1103/PhysRevLett.111.10700

[16] S. Keskin, E. Erdi, T. Kocak, An efficient parallel implementation of 3D-FFT on GPU, 978-1-5386-3472-1/17/2017 IEEE.

[17] M. Vetterli, J. Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 2007.

[18] A. Prochazka, L. Grafova, O. Vysata, Three-Dimensional Wavelet Transform in Multidimensional Biomedical Volume Processing, *Proc. of the IASTED International Conf. Graphics and VR*, UK, 2011.

[19] R. Jiang, D. Crookes, FPGA Implementation of 3D Discrete Wavelet Transform for Real-Time Medical Imaging, *IEEE Xplore*, Conf. Paper, Sept. 2007.

[20] M. Do, M. Vetterli, Contourlets: A Directional Multiresolution Image Representation, *IEEE Trans. on Image Processing*, Vol. 14, No. 12, Dec. 2005, pp. 2091-2106.

[21] J. Starck, E. Candes, D. Donoho, The Curvelet Transform for Image Denoising, *IEEE Trans. on Image Processing*, Vol. l, No 11, June 2002, pp. 670-684.

[22] S. Phoong, C. Kim, P. Vaidyanathan, R. Ansari, A new class of two-channel biorthogonal filter banks and wavelet bases, *IEEE Trans. Signal Proc.*, Vol. 43, No. 3, 1995, pp. 649-665.

[23] E. Candes, L. Demanet, D. Donoho, L. Ying, Fast Discrete Curvelet Transforms, *Multiscale Modeling and Simulation*, Vol. 5, No. 3, 2006, pp. 861-899.

[24] B. Goossens, H. Luong, J. Aelterman, A. Pizurica, W. Philips, Efficient Multiscale and Multidirectional Representation of 3D Data using the 3D Discrete Shearlet Transform, *Proc. of SPIE*, M, Papadakis, D. Ville, V. Goyal (Eds.), Vol. 8138, Bellingham, USA, 2011, pp. 81381Z–1–81381Z–13.

[25] R. Kountchev, Vl. Todorov, R. Kountcheva, Linear and Non-linear Inverse Pyramidal Image Representation: Algorithms and Applications, *Advances in Reasoning-based Image Processing, Analysis and Intelligent Systems*, R. Kountchev, K. Nakamatsu (Eds.), Springer, 2012, pp. 35-89.

[26] R. Kountchev, B. Iantovics, R. Kountcheva, Hierarchical Third-Order Tensor Decomposition through Inverse Difference Pyramid based on the three-dimensional Walsh–Hadamard Transform with Applications in Data Mining, *WIREs Data Mining Knowledge Discovery* 2019; e1314, https://doi.org/10.1002/widm.1314.

[27] S. Agaian, H. Sarukhanyan, K. Egiazarian, J. Astola, *Hadamard Transforms*, Washington, USA, SPIE Press Book, 2011.

[28] V. Testoni, M. Costa, 3D-Hadamard Coefficients Sequence Scan Order for a Fast Embedded Color Video Codec, *Elsevier Computers & Electrical Engineering*, Vol. 36, No 4, 2009, pp. 676-690.