

Article

# Hierarchical Cubical Tensor Decomposition through Low Complexity Orthogonal Transforms

Roumen K. Kountchev <sup>1,\*</sup>, Rumen P. Mironov <sup>1</sup> and Roumiana A. Kountcheva <sup>2</sup>

<sup>1</sup> Department of Radio Communications and Video Technologies, Faculty of Telecommunications, Technical University of Sofia, Sofia 1756, Bulgaria; rmironov@tu-sofia.bg

<sup>2</sup> TK Engineering, Sofia 1712, Bulgaria; kountcheva\_r@yahoo.com

\* Correspondence: rkountch@tu-sofia.bg; Tel.: +35-9992-4284

Received: 5 May 2020; Accepted: 19 May 2020; Published: 25 May 2020



**Abstract:** In this work, new approaches are proposed for the 3D decomposition of a cubical tensor of size  $N \times N \times N$  for  $N = 2^n$  through hierarchical deterministic orthogonal transforms with low computational complexity, whose kernels are based on the Walsh-Hadamard Transform (WHT) and the Complex Hadamard Transform (CHT). On the basis of the symmetrical properties of the real and complex Walsh-Hadamard matrices are developed fast computational algorithms whose computational complexity is compared with that of the famous deterministic transforms: the 3D Fast Fourier Transform, the 3D Discrete Wavelet Transform and the statistical Hierarchical Tucker decomposition. The comparison results show the lower computational complexity of the offered algorithms. Additionally, they ensure the high energy concentration of the original tensor into a small number of coefficients of the so calculated transformed spectrum tensor. The main advantage of the proposed algorithms is the reduction of the needed calculations due to the low number of hierarchical levels compared to the significant number of iterations needed to achieve the required decomposition accuracy based on the statistical methods. The choice of the 3D hierarchical decomposition is defined by the requirements and limitations related to the corresponding application area.

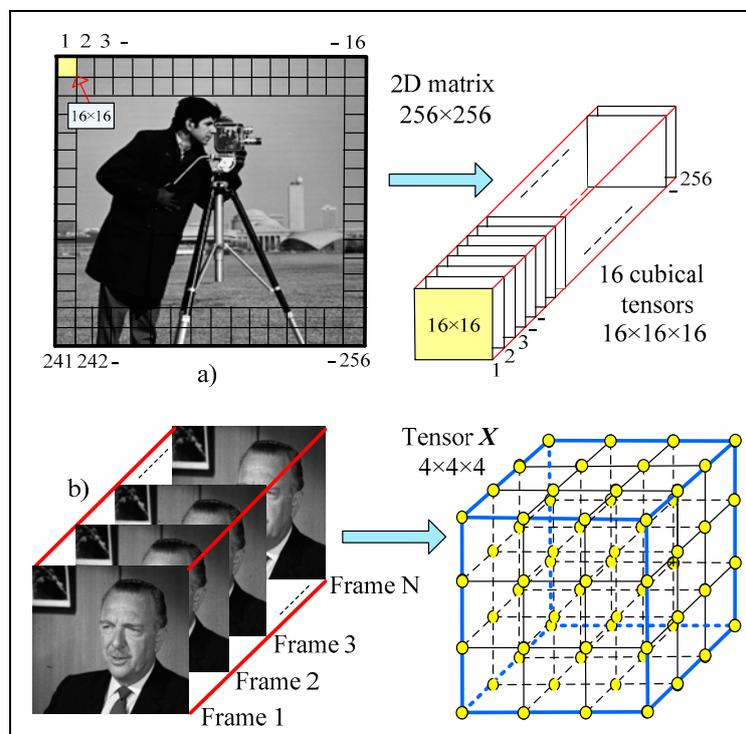
**Keywords:** decomposition of 3D cubical tensor; hierarchical 3D Fast Walsh-Hadamard Transform (3D-FWHT); 3D Hierarchical Fast Complex Walsh-Hadamard Transform (3D-FCHT)

## 1. Introduction

The famous tensor decompositions—Canonical Polyadic Decomposition (CPD), Higher-Order Singular Value Decomposition (HOSVD) [1–3], Tensor Trains Decomposition (TTD) [4], and Hierarchical Tucker decomposition (H Tucker) [5] - and their modifications [6] are based on the calculation of the eigen values and eigen vectors of the decomposed tensor. Their basic advantage is that they are optimum with respect to the Mean Square Error (MSE) of the approximation in the case of the truncation of the low-energy decomposition components. For the calculation of the retained components are used various iterative methods (the power method [7], the Jacoby method [8], etc.), which require relatively high numbers of calculations to achieve the needed approximation accuracy.

As an alternative, in this work are presented new hierarchical approaches for 3D tensor decomposition, based on well-known deterministic orthogonal transforms: the Walsh-Hadamard Transform (WHT) [9–12] and the Complex Hadamard Transform (CHT) [13]. The offered decompositions are not optimum with respect to MSE minimization, but due to the lack of iterations, they have low Computational Complexity (CC), and as a result, they are suitable for the fast processing of the 3D tensors obtained, for example, from single or sequences of 2D correlated images. In the first case, each single 2D image is divided into  $N^2$  blocks of size  $N \times N$  for  $N = 2^n$ , from which is obtained a sequence of  $N$  cubical tensors, each of size  $N \times N \times N$ . In the second case, from each sequence of

$N \times N$  2D images of size  $N \times N$  is obtained a single cubical tensor of size  $N \times N \times N$ . Examples for the transformation of a single 2D image and of a sequence of 2D video frames into a cubical (3D) tensor are shown on Figure 1.



**Figure 1.** Transformation into a 3D cubical tensor for (a) a single 2D image and (b) a sequence of 2D frames.

The presented approach for fast hierarchical cubical tensor decomposition is applicable not only for images but also for various kinds of multidimensional signals (medical, seismic, spectrometric, etc.) and big data analysis. In the next sections, Sections 2–5, are presented the proposed algorithms for 1D Hierarchical Fast Walsh-Hadamard Transform (1D-FWHT), 1D Hierarchical Fast Complex Hadamard Transform (1D-FCHT), and the corresponding 3D Hierarchical Fast real and complex transforms (3D-FWHT and 3D-FCHT). In Section 6 is shown a comparative analysis of the CC of the presented algorithms for hierarchical orthogonal tensor transforms, with respect to the famous deterministic and statistical transforms: the 3D Fast Fourier Transform (3D-FFT), 3D Discrete Wavelet Transform (3D-DWT) and Hierarchical Tucker Decomposition (H-Tucker). The last section, Section 7, contains the conclusions.

In Appendices A.1 and A.2 is given the factorization of the matrices for the  $n$ -level one-dimensional fast hierarchical transforms 1D-FWHT and 1D-FCHT.

## 2. One-Dimensional Hierarchical Fast Walsh-Hadamard Transform

The following symbols are introduced for tensors, matrices and vectors, respectively:  $X$  for a tensor,  $X$  for a matrix and  $\vec{x}$  for a vector.

The forward and inverse 1D Walsh-Hadamard Transform (1D-WHT) (frequency-ordered) is represented in scalar form by the equations below [9–11]:

$$s(u) = \sum_{i=0}^{N-1} x(i) \text{wal}(i, u); \quad x(i) = (1/N) \sum_{u=0}^{N-1} s(u) \text{wal}(i, u) \quad \text{for } i, u = 0, 1, \dots, N-1, \quad (1)$$

where  $N = 2^n$  and  $x(i)$  and  $s(u)$  are, respectively, the  $N$ -dimensional discrete signal and its spectrum. The discrete Walsh functions are defined by the following relations [10,12]:

$$\text{wal}(i, u) = (-1)^{\sum_{r=0}^{n-1} q_r(i)u_r}; q_r(i) = i_{n-r} \oplus i_{n-r-1} \text{ for } r = 1, 2, \dots, n-1; q_0(i) = i_{n-1}; i = \sum_{r=0}^{n-1} i_r 2^r, u = \sum_{r=0}^{n-1} u_r 2^r. \quad (2)$$

Here, the operation “exclusive OR” is represented by the symbol  $\oplus$ .

The algorithm for the one-dimensional hierarchical fast Walsh-Hadamard transform (1D-FWHT) for the cubical tensor  $X$  of size  $N \times N \times N$  when  $N = 2^n$  (a sequence of matrices  $X_k$ , each of size  $N \times N$  for  $k = 1, 2, \dots, N$ ) is presented for  $N = 8$  (i.e., for  $n = 3$  levels). The general case of the transform when  $N = 2^n$  and  $n > 3$ , is given in Appendix A.1.

The execution of the direct and inverse 1D-WHT is grounded on the basic operation “butterfly” for  $N = 2$ , in accordance with the following relations [9,11]:

$$\begin{aligned} y(1) &= x(1) + x(2), & x(1) &= (1/2)[y(1) + y(2)], \\ y(2) &= x(1) - x(2), & x(2) &= (1/2)[y(1) - y(2)]. \end{aligned} \quad (3)$$

The application of the operation “butterfly” to the elements of the couples of matrices in each hierarchical level is represented by the equations below.

In Level 1 of the 1D-FWHT:

$$Y = G_1(8)X. \quad (4)$$

The matrix  $G_1(8)$  of size  $8 \times 8$  is used to execute the direct 1D-WHT when  $N = 2$  for each couple of neighbor matrices in the sequence  $X_k$  for  $k = 1, 2, \dots, 8$ . These matrices are the components of the matrix-column  $X = [X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8]^T$ . From Equation (3), it follows that the matrix  $G_1(8)$  could be represented in the following way:

$$G_1(8) = I(4) \otimes H(2) = \begin{bmatrix} H(2) & 0 & 0 & 0 \\ 0 & H(2) & 0 & 0 \\ 0 & 0 & H(2) & 0 \\ 0 & 0 & 0 & H(2) \end{bmatrix}, H(2) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} - \text{Hadamard matrix of size } 2 \times 2. \quad (5)$$

Here,  $I(4)$  is the identity matrix of size  $4 \times 4$ , the symbol  $\otimes$  stands for the Kronecker product of the matrices [3], and  $Y = [Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, Y_7, Y_8]^T$  corresponds to the transformed matrix-column whose components are the matrices  $Y_k$ , for  $k = 1, 2, \dots, 8$ .

After the *rearrangement 1* is obtained:

$$Y' = P_1(8)Y, \quad (6)$$

where

$$P_1(8) = L_1(4) \otimes I(2) = \begin{bmatrix} L_1(4) & 0 \\ 0 & L_1(4) \end{bmatrix}; L_1(4) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \text{permutation sub - matrix of size } 4 \times 4; \quad (7)$$

$P_1(8)$ —permutation matrix of size  $8 \times 8$ ;  $Y' = [Y'_1, Y'_2, Y'_3, Y'_4, Y'_5, Y'_6, Y'_7, Y'_8]^T$ —rearranged matrix-column with components  $Y'_1 = Y_1, Y'_2 = Y_3, Y'_3 = Y_2, Y'_4 = Y_4, Y'_5 = Y_5, Y'_6 = Y_7, Y'_7 = Y_6, Y'_8 = Y_8$ .

In the Level 2 of the 1D-FWHT is obtained:

$$Z = G_2(8)Y', \quad (8)$$

where  $G_2(8) = G_1(8)$  and  $Z = [Z_1, Z_2, Z_3, Z_4, Z_5, Z_6, Z_7, Z_8]^T$ . The components of the matrix Z are:

$$Z_1 = Y_1 + Y_3; Z_2 = Y_1 - Y_3; Z_3 = Y_2 + Y_4; Z_4 = Y_2 - Y_4; \\ Z_5 = Y_5 + Y_7; Z_6 = Y_5 - Y_7; Z_7 = Y_6 + Y_8; Z_8 = Y_6 - Y_8.$$

After the rearrangement 2 is obtained:

$$Z' = P_2(8)Z, \tag{9}$$

where

$$P_2(8) = L_2(4) \otimes I(2) = \begin{bmatrix} L_2(4) & 0 \\ 0 & L_2(4) \end{bmatrix}; L_2(4) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} - \text{permutation sub - matrix of size } 4 \times 4; \tag{10}$$

$P_2(8)$ —permutation matrix of size  $8 \times 8$ ;  $Z' = [Z'_1, Z'_2, Z'_3, Z'_4, Z'_5, Z'_6, Z'_7, Z'_8]^T$ - rearranged matrix-column with components  $Z'_1 = Z_1, Z'_2 = Z_2, Z'_3 = Z_4, Z'_4 = Z_3, Z'_5 = Z_5, Z'_6 = Z_6, Z'_7 = Z_8, Z'_8 = Z_7$ .

In the Level 3 of the 1D-FWHT is obtained:

$$D = G_3(8)Z', \tag{11}$$

where

$$G_3(8) = H(2) \otimes I(4) = \begin{bmatrix} I(4) & I(4) \\ I(4) & -I(4) \end{bmatrix} \text{ for } I(4) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \text{identity matrix of size } 4 \times 4; \tag{12}$$

$D = [D_1, D_2, D_3, D_4, D_5, D_6, D_7, D_8]^T$ - transformed matrix-column with components  $D_1 = Z_1 + Z_5; D_2 = Z_1 - Z_5; D_3 = Z_2 + Z_6; D_4 = Z_2 - Y_6; D_5 = Z_4 + Z_8; D_6 = Z_4 - Z_8; D_7 = Z_3 + Z_7; D_8 = Z_3 - Z_7$ .

After the rearrangement 3 is obtained:

$$E = P_3(8)D, \tag{13}$$

where  $P_3(8) = P_2(8)$ ;  $E = [E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8]^T$ - output matrix-column with components  $E_1 = D_1, E_2 = D_2, E_3 = D_4, E_4 = D_3, E_5 = D_5, E_6 = D_6, E_7 = D_8, E_8 = D_7$ .

The relation between the input and output matrix-column X and E, correspondingly, defined by the relations in Equations (4)–(13), is:

$$E = P_3(8)G_3(8)P_2(8)G_2(8)P_1(8)G_1(8)X = H_w(8)X, \tag{14}$$

where  $H_w(8)$  is a frequency-ordered Hadamard matrix, defined as follows:

$$H_w(8) = P_3(8)G_3(8)P_2(8)G_2(8)P_1(8)G_1(8) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \tag{15}$$

The rearrangement of the intermediate matrix components in each consecutive transform level must satisfy the requirement for the frequency-ordering of the output matrices [9,14]. The chosen method for matrix  $H_w(8)$  factorization permits the operation “butterfly” in the first two levels to be applied to the elements of the neighbor couples of matrices  $X_k$  and  $X_{k+1}$  (for  $k = 1, 2, \dots, 8$ ) from the sequence, which builds the cubical tensor  $X$  of size  $8 \times 8 \times 8$ . In this way, the decorrelation efficiency for the elements of each couple of transformed matrices is enhanced, which results in higher energy concentration in the low-frequency components of the output matrix-column,  $E$ . In the result of the 1D-FWHT is obtained the cubical tensor  $E$  of size  $8 \times 8 \times 8$ , represented by the sequence of matrices  $E_l$ , each of size  $8 \times 8$ , when  $l = 1, 2, \dots, 8$ .

The graph of the 3-level 1D-FWHT algorithm for the tensor  $X$  of size  $8 \times 8 \times 8$  is shown in Figure 2. Here, the basic operation “butterfly” in the first two levels is framed in red.

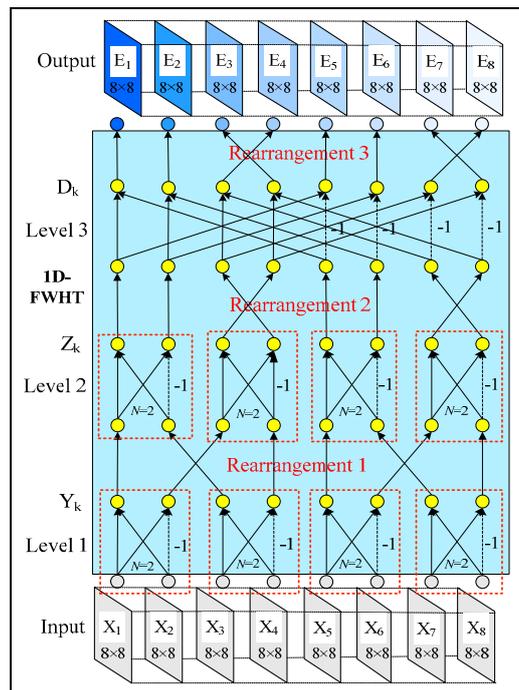


Figure 2. Hierarchical 1D fast real Walsh-Hadamard transform for a sequence of matrices,  $X_k$ .

An example 1D-FWHT is given below for the processing of a sequence of matrices  $X_k$  for  $k = 1, 2, \dots, 8$ , in a particular case: the transform of one element  $x(i, m, k)$  of the same spatial position  $(i, m)$  in each matrix,  $k$ :

Let:

$$x(i, m, 1) = 1; x(i, m, 2) = 2; x(i, m, 3) = 2; x(i, m, 4) = 3; \\ x(i, m, 5) = 4; x(i, m, 6) = 1; x(i, m, 7) = 6; x(i, m, 8) = 5;$$

Then, in Level 1 of the 1D-HFWHT:

$$y(i, m, 1) = 3; y(i, m, 2) = -1; y(i, m, 3) = 5; y(i, m, 4) = -1; \\ y(i, m, 5) = 5; y(i, m, 6) = 3; y(i, m, 7) = 11; y(i, m, 8) = 1;$$

After the rearrangement 1:

$$y(i, m, 1) = 3; y(i, m, 3) = 5; y(i, m, 2) = -1; y(i, m, 4) = -1; \\ y(i, m, 5) = 5; y(i, m, 7) = 11; y(i, m, 6) = 3; y(i, m, 8) = 1;$$

In Level 2 of the 1D-HFWHT:

$$z(i, m, 1) = 8; z(i, m, 2) = -2; z(i, m, 3) = -2; z(i, m, 4) = 0; \\ z(i, m, 5) = 16; z(i, m, 6) = -6; z(i, m, 7) = 4; z(i, m, 8) = 2;$$

After the rearrangement 2:

$$z(i, m, 1) = 8; z(i, m, 2) = -2; z(i, m, 4) = 0; z(i, m, 3) = -2; \\ z(i, m, 5) = 16; z(i, m, 6) = -6; z(i, m, 8) = 2; z(i, m, 7) = 4;$$

In Level 3 of the 1D-HFWHT:

$$d(i, m, 1) = 24; d(i, m, 2) = -8; d(i, m, 3) = -8; d(i, m, 4) = 4; \\ d(i, m, 5) = 2; d(i, m, 6) = -2; d(i, m, 7) = 2; d(i, m, 8) = -6;$$

After the rearrangement 3:

$$e(i, m, 1) = 24; e(i, m, 2) = -8; e(i, m, 3) = 4; e(i, m, 4) = -8; \\ e(i, m, 5) = 2; e(i, m, 6) = -2; e(i, m, 7) = -6; e(i, m, 8) = 2.$$

Here,  $y(i, m, k)$ ,  $z(i, m, k)$ ,  $d(i, m, k)$  and  $e(i, m, k)$  are, correspondingly, the elements of the sequences of matrices  $Y_k$ ,  $Z_k$ ,  $D_k$  and  $E_k$  for  $k = 1, 2, \dots, 8$ , calculated in levels 1, 2 and 3, and for the graph output.

In accordance with the generalized equation of Parseval [15] is obtained:

$$\sum_{i=1}^N \sum_{m=1}^N \sum_{k=1}^N x(i, m, k)^2 = (1/N^3) \sum_{u=1}^N \sum_{v=1}^N \sum_{l=1}^N e(u, v, l)^2. \quad (16)$$

Here,  $x(i, m, k)$  and  $e(u, v, l)$  are, respectively, the elements of the input tensor  $X$  and of the spectral tensor,  $E$ . Then, for the example above, for  $N = 8$ , from Equation (16) is obtained

$$\sum_{k=1}^8 x(i, j, k)^2 = (1/8) \sum_{n=1}^8 e(u, v, l)^2 = 96.$$

In Appendix A.1 is given, in detail, the frequency-ordered hierarchical n-level 1D-FWHT.

### 3. One-Dimensional Hierarchical Fast Complex Hadamard Transform

The direct and inverse 1D Complex Hadamard Transform (1D-CHT) (frequency-ordered) is represented in a scalar form by the equations below [14]:

$$s(u) = \sum_{i=0}^{N-1} x(i) c(i, u) = s_{\text{Re}}(u) + js_{\text{Im}}(u); x(i) = (1/N) \sum_{u=0}^{N-1} s(u) c^*(i, u) \text{ for } i, u = 0, 1, \dots, N-1, \quad (17)$$

where  $N = 2^n \geq 4$  (the minimum possible value is  $n = 2$ );  $x(i)$  and  $s(u)$  are, correspondingly, the  $N$ -dimensional discrete signal and its complex spectrum;  $j = \sqrt{-1}$ ; and  $s_{\text{Re}}(u)$ , and  $s_{\text{Im}}(u)$  are the real and the imaginary parts of the spectrum  $s(u)$ . The coefficients of the 1D-CHT matrix  $C(N)$  of size  $N \times N$  are represented by the equation:

$$c(i, u) = j^{iu} h(i, u); c^*(i, u) = j^{-iu} h(i, u), \quad (18)$$

where  $i, u = 0, 1, \dots, 2^n - 1$ . The sign function  $h(i, u)$  is:

$$h(i, u) = \begin{cases} 1 & \text{for } n = 2, \\ \prod_{r=3}^n (-1)^{[i/2^{i-r}] [u/2^{i-r}]} & \text{for } n = 3, 4, \dots \end{cases} \quad (19)$$

Here,  $[\circ]$  is an operator, which represents the integer part of the result, obtained after the division. The even coefficients of the 1D-CHT are real, and the odd ones are complex-conjugated.

For  $N = 4$  (respectively,  $n = 2$ ), Equations (17)–(19) for  $i, u = 0, 1, 2, 3$  are transformed into:

$$\begin{bmatrix} s(0) \\ s(1) \\ s(2) \\ s(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & j & -j \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -j & j \end{bmatrix} \begin{bmatrix} x(0) \\ x(2) \\ x(1) \\ x(3) \end{bmatrix}; \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -j & j \\ 1 & 1 & -1 & -1 \\ 1 & -1 & j & -j \end{bmatrix} \begin{bmatrix} s(0) \\ s(2) \\ s(1) \\ s(3) \end{bmatrix}. \quad (20)$$

In this case, in accordance with Equation (20), for the calculation of the direct/inverse 1D-CHT are needed four basic “butterfly” operations, whose weight coefficients are, respectively,  $(+1, -1)$  for three of the “butterflies” and  $(+j, -j)$  for the fourth.

In Figure 3 is shown the calculation graph of the direct 3-level fast hierarchical 1D-CHT ( $n = 3$ ) for  $N = 8$ , built in accordance with Figure 2.

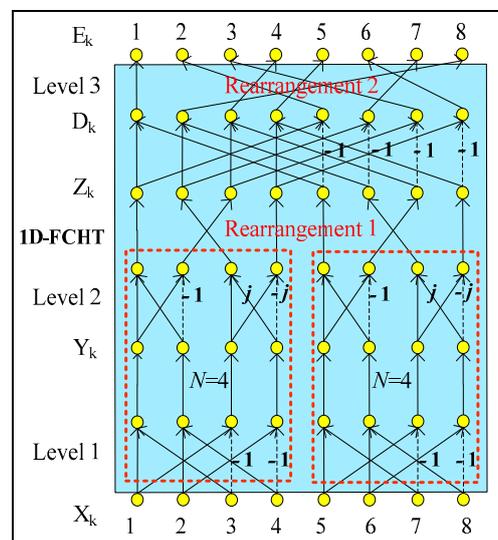


Figure 3. Hierarchical 1D Fast Complex Hadamard Transform (1D-FCHT), for  $N = 8$ .

The graph for the 1D-FCHT when  $N = 8$  is created on the basis of Equation (20), to which are added four “butterfly” operations with weight coefficients  $(+1, -1)$  in the last, third level. In the result of the use of the frequency-ordered 1D-FCHT [14] is obtained the following system of equations, which represent the relations between the sequences of the input and output matrices, respectively  $X_k$  and  $E_k$ , for  $k = 1, 2, \dots, 8$  and  $N = 8$ :

$$\begin{aligned} E_1 &= X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 + X_8; & E_5 &= X_1 - X_3 + X_5 - X_7 - j(X_2 - X_4 + X_6 - X_8); \\ E_2 &= X_1 + X_2 + X_3 + X_4 - X_5 - X_6 - X_7 - X_8; & E_6 &= X_1 - X_3 - X_5 + X_7 - j(X_2 - X_4 - X_6 + X_8); \\ E_3 &= X_1 - X_3 - X_5 + X_7 + j(X_2 - X_4 - X_6 + X_8); & E_7 &= X_1 - X_2 + X_3 - X_4 - X_5 + X_6 - X_7 + X_8; \\ E_4 &= X_1 - X_3 + X_5 - X_7 + j(X_2 - X_4 + X_6 - X_8); & E_8 &= X_1 - X_2 + X_3 - X_4 + X_5 - X_6 + X_7 - X_8. \end{aligned} \quad (21)$$

Unlike Equation (14), part of the equations in the system above contain the complex variable  $j = \sqrt{-1}$ , and they represent complex matrices (i.e., these are relations  $E_3, E_4, E_5$  and  $E_6$ ). The remaining four equations ( $E_1, E_2, E_7$  and  $E_8$ ) represent real matrices. The components of each complex matrix  $E_k = A_k + jB_k$  ( $k = 3, 4, 5, 6$ ) are  $A_k = \sum_{p=1,3,5,7} \alpha_p X_p$  and  $B_k = \sum_{p=2,4,6,8} \beta_p X_p$ ;  $\alpha_p$  and  $\beta_p$  are sign functions with values of  $+1$  or  $-1$ , defined by Equation (21). The elements  $e(i, m, k)$  of the complex matrix  $E_k$  are defined as follows:

$$e(i, m, k) = e(i, m, k)_{\text{Re}} + je(i, m, k)_{\text{Im}} = M_k(i, m) e^{j\Phi_k(i, m)}, \quad (22)$$

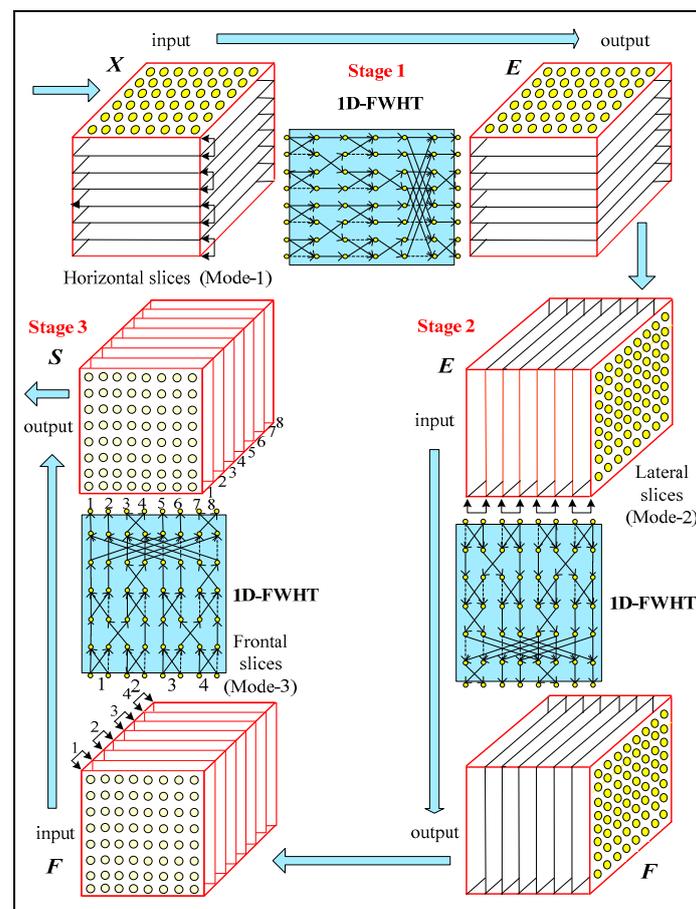
where  $M_k(i, m) = \sqrt{e(i, m, k)_{\text{Re}}^2 + e(i, m, k)_{\text{Im}}^2}$  is a module; and  $\Phi_k(i, m) = \arctg[e(i, m, k)_{\text{Im}}/e(i, m, k)_{\text{Re}}]$ , a phase of the element  $e(i, m, k)$ .

The matrix  $\Phi_k$  permits the phase modification of its elements, which could be used (for example) for the resistant digital watermarking of multidimensional signals and images [14].

In Appendix A.2 is presented in detail the factorization of the matrix for an n-level hierarchical frequency-ordered 1D-FCHT, based on the “butterfly” operation (Equation (20)) used for the 1D-CHT execution, if  $N = 4$ .

#### 4. Hierarchical Cubical Tensor Decomposition through the 3D-FWHT

At first, here is described the decomposition of the cubical tensor  $X$  of size  $8 \times 8 \times 8$ . In this case, the decomposition could be represented in the simplified form shown in Figure 4.



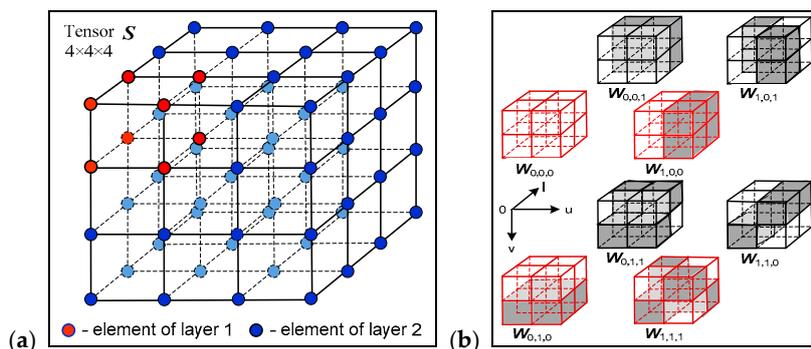
**Figure 4.** The execution stages of the 3D Hierarchical Fast Walsh-Hadamard Transform (3D-FWHT) for the tensor  $X$  of size  $8 \times 8 \times 8$ , based on the 1D Hierarchical Fast Walsh-Hadamard Transform (1D-FWHT).

The decomposition based on the 3D-FWHT comprises three stages. In the first stage, the tensor is divided into horizontal slices (mode-1), after which, on each couple of matrices, is applied the 1D-FWHT, and from the so obtained eight slices is restored the tensor  $E$ . In a similar way are executed the operations in the second and third stages. Unlike in the first, in the second stage, the input tensor  $E$  is divided into eight lateral slices (mode-2). After their transform through the 1D-FWHT, from the obtained eight slices is restored the tensor  $F$ . In the third stage, the tensor  $F$  is divided into eight frontal slices (mode-3), from which, after the 1D-FWHT, is obtained the spectrum tensor  $S$  with elements  $s(u, v, l)$ . From this tensor, after the reverse execution of the three stages of the inverse 1D-FWHT,

is restored the initial tensor  $X$  with elements  $x(i,j,k)$ . In the result of the decomposition, it is represented as a sum of weighted “basic” tensors  $W_{u,v,l}$  of size  $8 \times 8 \times 8$  with elements  $wal(u,v,l)$ , which are 3D Walsh functions:

$$X = (1/512)[s(0,0,0)W_{0,0,0} + s(1,0,0)W_{1,0,0} + \dots + s(7,7,7)W_{7,7,7}] = (1/8^3) \sum_{u=0}^7 \sum_{v=0}^7 \sum_{l=0}^7 s(u,v,l)W_{u,v,l}. \quad (23)$$

The 3D functions  $wal(u,v,l)$  are divisible [16,17], and they could be represented as the product of three 1D Walsh functions:  $wal(u).wal(v).wal(l)$ . The coefficients  $s(u,v,l)$  for  $u = 0,1, v = 0,1$  and  $l = 0,1$  in the first layer correspond to the lowest spatial frequencies of the spectrum tensor,  $S$ . In Figure 5a is shown an example for a spectrum tensor  $S$  of size  $4 \times 4 \times 4$ , and in Figure 5b, the 8 “basic” tensors, which correspond to coefficients  $s(u,v,l)$  from the initial layer of the tensor  $S$  [18].



**Figure 5.** (a) The spectrum tensor  $S$  of size  $4 \times 4 \times 4$ ; (b) “Basic” tensors  $W_{u,v,l}$ , which correspond to the first layer of  $S$ .

The 3D-FWHT algorithm, shown in Figure 4, could be generalized for the case when  $N = 2^n$  by creating an extended n-level computational graph in correspondence with the transform matrix factorization explained in Appendix A.1. The number of stages needed for the execution of the 1D-FWHT (in accordance with Figure 4) is three, not only for  $N = 8$  but also for the higher values of  $N = 16, 32, \dots$

In the general case, the spectrum coefficients  $s(u,v,l)$  are calculated through the direct 3D-WHT, in accordance with the relation [12,18]:

$$s(u,v,l) = \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} x(i,j,k) wal(i,u) wal(j,v) wal(k,l) \text{ for } u,v,l = 0,1,\dots,N-1. \quad (24)$$

The 3D inverse WHT (3D-IWHT) is defined by the relation:

$$x(i,m,k) = \frac{1}{N^3} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \sum_{l=0}^{N-1} s(u,v,l) wal(i,u) wal(m,v) wal(k,l) \text{ for } i,m,k = 0,1,\dots,N-1. \quad (25)$$

Here, the discrete one-dimensional Walsh-Hadamard (WH) functions of  $N^{\text{th}}$  order  $wal(i,u)$ ,  $wal(m,v)$ ,  $wal(k,l)$  for  $i,m,k = 0,1, \dots, N-1$ , are defined by the following relations:

$$wal(i,u) = (-1)^{\sum_{r=0}^{n-1} q_r(i)u_r}, \quad wal(m,v) = (-1)^{\sum_{r=0}^{n-1} q_r(m)v_r}, \quad wal(k,l) = (-1)^{\sum_{r=0}^{n-1} q_r(k)l_r}, \quad (26)$$

where  $q_r(i), q_r(m), q_r(k)$  and  $u_r, v_r, l_r$  are defined by analogy with Equation (2) after converting the decimal numbers  $i, m, k$  and  $u, v, l$  into binary, respectively:  $i = \sum_{r=0}^{n-1} i_r 2^r, m = \sum_{r=0}^{n-1} m_r 2^r, k = \sum_{r=0}^{n-1} k_r 2^r$  and  $v = \sum_{r=0}^{n-1} v_r 2^r, u = \sum_{r=0}^{n-1} u_r 2^r, l = \sum_{r=0}^{n-1} l_r 2^r$  for  $n = \log_2 N$ .

Each 3<sup>d</sup>-order tensor  $X$  of size  $N \times N \times N$  could be calculated by using the weighted sum of the  $N^3$  3D WH functions represented through tensors  $W_{u,v,l}$ , each of size  $N \times N \times N$ :

$$X = (1/N^3)[s(0,0,0)W_{0,0,0} + s(1,0,0)W_{1,0,0} + \dots + s(N-1,N-1,N-1)W_{N-1,N-1,N-1}] = (1/N^3) \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \sum_{l=0}^{N-1} s(u,v,l) W_{u,v,l}. \quad (27)$$

Each "basic" tensor  $W_{u,v,l}$  with frequency  $(u,v,l)$  could be represented through the outer product of the vectors  $\vec{w}_u, \vec{w}_v, \vec{w}_l$ :

$$W_{u,v,l} = \vec{w}_u \circ \vec{w}_v \circ \vec{w}_l, \quad (28)$$

where the vectors, which represent the tensor  $W_{u,v,l}$ , are defined by the relations below:

$$\vec{w}_u = [(-1)^{\sum_{r=0}^{n-1} q_r(0)u_r}, (-1)^{\sum_{r=0}^{n-1} q_r(1)u_r}, \dots, (-1)^{\sum_{r=0}^{n-1} q_r(N-1)u_r}]^T; \quad (29)$$

$$\vec{w}_v = [(-1)^{\sum_{r=0}^{n-1} q_r(0)v_r}, (-1)^{\sum_{r=0}^{n-1} q_r(1)v_r}, \dots, (-1)^{\sum_{r=0}^{n-1} q_r(N-1)v_r}]^T; \quad (30)$$

$$\vec{w}_l = [(-1)^{\sum_{r=0}^{n-1} q_r(0)l_r}, (-1)^{\sum_{r=0}^{n-1} q_r(1)l_r}, \dots, (-1)^{\sum_{r=0}^{n-1} q_r(N-1)l_r}]^T. \quad (31)$$

Then, the tensor decomposition based on the 3D-WHT, is defined by the relation:

$$X = (1/N^3) \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \sum_{l=0}^{N-1} s(u,v,l) [\vec{w}_u \circ \vec{w}_v \circ \vec{w}_l]. \quad (32)$$

As follows from the relations in Equations (25) and (32), the 3D-WHT is reversible, i.e.,  $X$  could be restored from the tensor  $S$  through the 3D-IWTH. The decomposition in Equation (32) of the tensor  $X$  of size  $N \times N \times N$  corresponds to the Tucker decomposition [1,3,5]:

$$X = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} g_{r_1,r_2,r_3} (\vec{a}_{r_1} \circ \vec{b}_{r_2} \circ \vec{c}_{r_3}). \quad (33)$$

Here,  $g_{r_1,r_2,r_3}$  are the entries of the core tensor  $G$  of size  $R_1 \times R_2 \times R_3$ , and  $R_1, R_2, R_3$ , the multilinear rank of the tensor  $X$ . For the case, when it is a cube of size  $N \times N \times N$ , the tensor  $G$  is diagonal and of size  $R \times R \times R$ ;  $g_r$  (for  $r = 1, 2, \dots, R$ ) are its eigen values, and  $\vec{a}_{r_1}, \vec{b}_{r_2}, \vec{c}_{r_3}$ , the eigen vectors. In this case, the Tucker decomposition is transformed into CPD [1], and is represented by the relation below:

$$X = \sum_{r=1}^R g_r (\vec{a}_r \circ \vec{b}_r \circ \vec{c}_r). \quad (34)$$

Here,  $R$  is the rank of the tensor  $X$ , whose value is limited in the range  $N \leq R \leq N^2$  [19]. The difference between the decompositions, represented by Equations (33) and (32), is in the numbers of their components; in the first case, it is  $\leq N^2$ , and in the second, it is  $\leq N^3$ . This shows the higher energy concentration in the first components of the H-Tucker decomposition (respectively, CPD) compared to that of the 3D-WHT.

## 5. Hierarchical Cubical Tensor Decomposition through the 3D-FCHT

The comparison of the relations in Equations (14), (15) and (21) shows that the 1D-FWHT and 1D-FCHT are executed in accordance with algorithms with similar structure (Figures 2 and 3). From this, it follows that for the decomposition of the tensor  $X$  of size  $8 \times 8 \times 8$  through 3D-FCHT could be used the three-stage scheme shown in Figure 4. The difference is that in the three decomposition stages, the 1D-FWHT algorithm from Figure 2 must be replaced by this for the 1D-FCHT, shown in Figure 3. After the triple execution of the algorithm is obtained the spectrum tensor  $S$  (shown on Figure 4) with elements  $s(u, v, l)$ , half of which are complex. The 3D-FCHT algorithm, shown in Figure 3, could be generalized for the case when  $N = 2^n$  by creating an extended computational graph of  $n$  levels. The number of stages needed for the execution of the 1D-FCHT by analogy with the 3D-FWHT is also three (not for  $N = 8$  only but also for the next values of  $N = 16, 32, \dots$ ).

In the general case, the decomposition of the tensor  $X$  of size  $N \times N \times N$  in accordance with Equation (27) is represented as the sum of the weighted “basic” tensors:

$$X = (1/N^3) \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \sum_{l=0}^{N-1} s(u, v, l) C_{u,v,l} = (1/N^3) \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \sum_{l=0}^{N-1} s(u, v, l) [\vec{c}_u \circ \vec{c}_v \circ \vec{c}_l], \quad (35)$$

where  $C_{u,v,l} = \vec{c}_u \circ \vec{c}_v \circ \vec{c}_l$ . The vectors which define the tensor  $C_{u,v,l}$  are defined by the relations below:

$$\vec{c}_u = [h(0, u), j^u h(1, u), \dots, j^{iu} h(i, u), \dots, j^{(N-1)u} h(N-1, u)]^T; \quad (36)$$

$$\vec{c}_v = [h(0, v), j^v h(1, v), \dots, j^{mv} h(m, v), \dots, j^{(N-1)v} h(N-1, v)]^T; \quad (37)$$

$$\vec{c}_l = [h(0, l), j^l h(1, l), \dots, j^{kl} h(k, l), \dots, j^{(N-1)l} h(N-1, l)]^T. \quad (38)$$

In the above relations, the sign functions  $h(i, u)$ ,  $h(m, v)$  and  $h(k, l)$  for  $i, m, k = 0, 1, \dots, N-1$  are defined by Equation (19). The spectrum coefficients  $s(u, v, l)$  in Equation (35) are calculated through the direct 3D-CHT, in accordance with the relation:

$$s(u, v, l) = \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} x(i, m, k) j^{(iu+mv+kl)} h(i, u) h(m, v) h(k, l) \text{ for } u, v, l = 0, 1, \dots, N-1. \quad (39)$$

The 3D inverse CHT (3D-ICHT) is defined as follows:

$$x(i, m, k) = (1/N^3) \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} s(u, v, l) j^{-(iu+mv+kl)} h(i, u) h(m, v) h(k, l) \text{ for } i, m, k = 0, 1, \dots, N-1. \quad (40)$$

## 6. Comparative CC Evaluation for the 3D-FWHT and 3D-FCWHT Algorithm

In this section is evaluated the CC of the decomposition algorithms for a tensor of size  $N \times N \times N$ , when  $N = 2^n$ .

### 6.1. CC of the Algorithm Based on the Real Orthogonal Transform, 3D-FWHT

The number of additions,  $A_W(1)$ , needed for the execution of the 1D-WHT based on the “butterfly” operation for all  $N^2$  elements of a couple of matrices, both of size  $N \times N$ , is:

$$A_{1H}(1) = 2N^2. \quad (41)$$

For all  $N/2$  couples of matrices in one stage of the hierarchical 3D-FWHT of  $n$  levels, this number is:

$$A_{1H}(n) = (N/2)2N^2 \lg_2 N = 2^{3n} n. \quad (42)$$

Then, for the three stages of the hierarchical 3D-FWHT is obtained:

$$A_{3H}(n) = 3 \times 2^{3n}n. \quad (43)$$

The needed number of multiplications for the execution of the direct 3D-FWHT is  $M_{3H}(n) = 0$ . Then, the CC of the 3D-FWHT evaluated through the total number of operations  $O_{3H}(n)$  is defined by the relation:

$$O_{3H}(n) = A_{3H}(n) + M_{3H}(n) = 3 \times 2^{3n}n. \quad (44)$$

### 6.2. CC of the Algorithm Based on the Complex Orthogonal Transform, 3D-FCHT

The number of additions for all  $N^2$  elements of  $N/2$  couples of matrices of size  $N \times N$  in one stage of the hierarchical  $n$ -level 1D-FCHT is:

$$A_{1C}(n) = N^2\{(3/2)N + (n-2)[(N/4) + 2(3N/4)]\} = 2^{3n+1}(0.875n - 1). \quad (45)$$

In the relation above is taken into account that in the first level of the 1D-FCHT are added  $N$  real numbers; in the second level, there are  $N/2$  real numbers, and in the remaining  $(n-2)$  levels,  $N/4$  real and  $3N/4$  complex numbers. Additionally, the summarizing of two complex numbers is executed as a sum of two couples of real numbers.

Then, for the three stages of the 3D Fast CHT (3D-FCHT), the number of needed additions is:

$$A_{3C}(n) = 3 \times 2^{3n}n = 6 \times 2^{3n}(0.875n - 1). \quad (46)$$

The number of complex multiplications needed for the direct 3D-FCHT is  $M_{3C}(n) = 0$  (the multiplication of  $j$  by a real number is not an arithmetical operation but only needs larger memory in which to save the so obtained complex number). Then, the CC of the 3D-FCHT, evaluated on the basis of the global number of needed operations  $O_{3C}(n)$ , is defined by the relation:

$$O_{3C}(n) = A_{3C}(n) + M_{3C}(n) = 6 \times 2^{3n}(0.875n - 1). \quad (47)$$

### 6.3. CC of the Algorithm Based on the Complex Orthogonal Transform, 3D-DFT

The divisibility of the 3D fast Fourier transform (3D-DFT) permits the tensor decomposition to be executed through applying the 1D-DFT on all fibers (vectors) obtained in the result of the tensor transform, called vectorization [1,3]. The vectors are oriented in three directions:  $x$  (mode-1),  $y$  (mode-2) and  $z$  (mode-3). The calculation of the 1D-DFT is based on the Cooley-Tukey algorithm for Fast Fourier Transform (FFT) [15]. In the result of the calculation of the radix-2 FFT for an  $N$ -dimensional vector, the number of operations (respectively, CC) is reduced from  $N^2$  to  $N \log_2 N = Nn$ . Then, the CC of the 3D-FFT is  $O(N_c \log_2 N_c)$ , where  $N_c = N_1 \times N_2 \times N_3$ . In the case that  $N_1 = N_2 = N_3 = N = 2^n$ , for a tensor of  $N^3$  voxels, the number of the complex additions is  $A_{3F}(n) = N^3 \log_2 N^3 = 3N^3n$ , and that of the complex multiplications,  $M_{3F}(n) = (N/2)^3 \log_2 N^3 = (3/8)N^3n$  [20]. The multiplication of two complex numbers is equivalent to four multiplications of real numbers, and one complex addition, to two additions of real numbers. Then, for the CC of the 3D-FFT is obtained:

$$O_{3F}(n) = A_{3F}(n) + M_{3F}(n) = 6.75N^3n + 0.75N^3n = 7.5 \times 2^{3n}n. \quad (48)$$

#### 6.4. CC of the Algorithm Based on the 3D Discrete Wavelet Transform, 3D-DWT and H-Tucker

The global number of additions and multiplications needed to calculate the cubical tensor of size  $N \times N \times N$  through the 3D Discrete Wavelet Transform (3D-DWT) of  $n$  levels with orthogonal filters (3, 5), are correspondingly [20]:

$$A_{3WT}(n) = 18N^3 \sum_{p=0}^{n-1} 8^{-p} = 20.57(2^{3n} - 1) \text{ and } M_{3WT}(n) = 24N^3 \sum_{p=0}^{n-1} 8^{-p} = 27.43(2^{3n} - 1). \quad (49)$$

Then, the CC of the 3D-DWT is:

$$O_{3WT}(n) = A_{3WT}(n) + M_{3WT}(n) \approx 48 \times 2^{3n}n. \quad (50)$$

The H-Tucker decomposition [5] of the cubical tensor of minimum rank  $R = 2^n$  (as defined in [19]), size  $N = 2^n$  and order  $d = 3$  requires  $O_{HT}(3 \times 2^{3n} + 2 \times 2^{4n})$  operations. For the TT-decomposition [4], the CC for the same tensor is  $O_{TT}(3 \times 2^{4n})$ , i.e., it is approximately 1.5 times higher than that for the H-Tucker. This is why the H-Tucker transform was selected for the CC comparison with the analyzed 3D deterministic orthogonal transforms.

#### 6.5. CC of 3D-FWHT and 3D-FCHT, Compared to 3D-FFT, 3D-DWT and H-Tucker

The relative acceleration  $\psi(n)$  of the calculations needed for the execution of the 3D-FWHT, 3D-FCHT and 3D-FFT compared to that of the 3D-DWT and H-Tucker is given in detail below:

3D-FWHT, 3D-FCHT and 3D-FFT compared to 3D-DWT:

$$\psi_1(n) = O_{3WT}(n)/O_{3H}(n) = (48 \times 2^{3n})/(3 \times 2^{3n}) \approx 16/n; \quad (51)$$

$$\psi_3(n) = O_{3WT}(n)/O_{3C}(n) = (48 \times 2^{3n})/[6 \times 2^{3n}(0.875n - 1)] \approx 8/(0.87n - 1); \quad (52)$$

$$\psi_5(n) = O_{3WT}(n)/O_{3F}(n) = (48 \times 2^{3n})/(7.5 \times 2^{3n}) \approx 6.4/n. \quad (53)$$

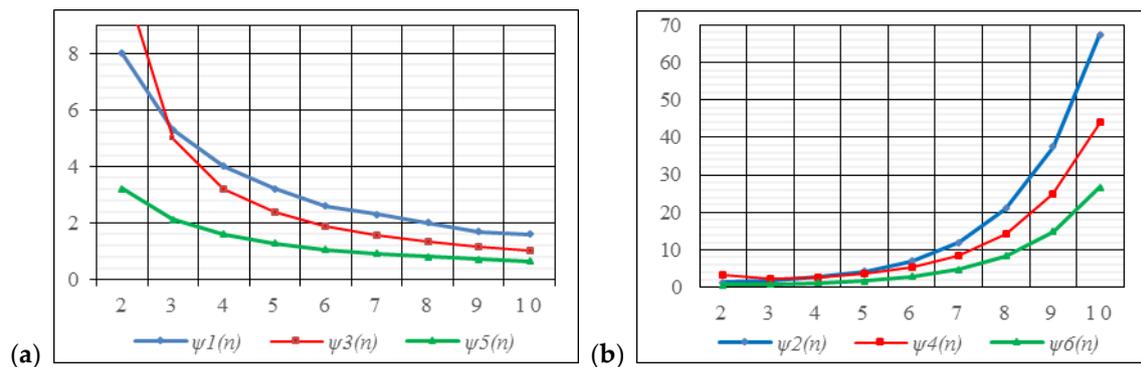
3D-FWHT, 3D-FCHT and 3D-FFT compared to H-Tucker:

$$\psi_2(n) = O_{HT}(n)/O_{3H}(n) = 2^{3n}(2^{n+1} + 3)/(3 \times 2^{3n}) \approx 0.66 \times (2^n/n); \quad (54)$$

$$\psi_4(n) = O_{HT}(n)/O_{3C}(n) = 2^{3n}(2^{n+1} + 3)/[6 \times 2^{3n}(0.875n - 1)] \approx (0.33 \times 2^n + 1)/(0.87n - 1); \quad (55)$$

$$\psi_6(n) = O_{HT}(n)/O_{3F}(n) = 2^{3n}(2^{n+1} + 3)/(7.5 \times 2^{3n}) \approx 0.26 \times (2^n/n). \quad (56)$$

In Figure 6a are shown the graphics of functions  $\psi_i(n)$  for  $i = 1, 3, 5$ , which define the relative CC of the 3D-FWHT, 3D-FCHT and 3D-FFT towards the 3D-DWT, and in Figure 6b, the functions  $\psi_i(n)$  for  $i = 2, 4, 6$ , which define the relative CC for same transforms towards the H-Tucker. The graphics in Figure 6a show that, together with the growth of the hierarchical levels  $n = \log_2 N$ , the relative CC of the 3D-FWHT, 3D-FCHT and 3D-FFT decreases inversely proportionally to  $n$  towards the 3D-DWT, while towards H-Tucker, it grows proportionally to  $2^n/n$ . As follows from the graphics on Figure 6b, the CC of the 3D-FCHT is the lowest compared to that of the 3D-FWHT, H-Tucker and 3D-FFT only for levels  $n = 2, 3$ , while for the next levels ( $n = 4, 5, \dots, 10$ ), the CC of the 3D-FWHT is the lowest. For the same range of  $n$  (from 4 to 10), the value of the function  $\psi_2(n)$  changes from 2.6 to 67.5, and that of  $\psi_4(n)$ , from 2.5 to 44. The comparison results permit the choosing of the number of hierarchical decomposition levels  $n$  for which the developed new algorithms 3D-FWHT and 3D-FCHT are more efficient than 3D-FFT, 3D-DWT and H-Tucker.



**Figure 6.** Relative Computational Complexity (CC) of the 3D-FWHT (blue), 3D-FCHT (red) and 3D Fast Fourier Transform (3D-FFT) (green) towards the 3D-DWT (a) and H-Tucker (b).

## 7. Conclusions

In this work is presented one new approach for the decomposition of the 3D tensors of size  $N \times N \times N$ , based on the deterministic orthogonal transforms 3D-FWHT and 3D-FCHT. The results of the comparative analysis for the CC of these decompositions outline their advantages over 3D-FFT, 3D-DWT and H-Tucker. Another advantage is that no iterative calculations are needed, and instead are used hierarchical calculations only, of low CC. The comparison results show that after the 1D-transform of the couples of tensor slices (matrices) in three mutually orthogonal directions, the energy is concentrated into a small number of low-frequency spectrum coefficients. The disadvantage of the method is that unlike the famous 3D decompositions, which are based on the tensor eigen vectors, the offered deterministic orthogonal transforms do not ensure minimum MSE, due to low-energy decomposition component truncation. The choice of the most suitable 3D deterministic hierarchical tensor decomposition depends on the corresponding application area.

The method could be also used for parallelepiped tensors. In such a case, the original tensor is divided into cubical sub-tensors, and on each should be applied the presented decomposition. The incomplete sub-tensors at the main tensor boundaries should be supplemented through 3D interpolation based on the closest boundary data. Besides, instead of WHT or CHT could be used orthogonal transforms whose matrices are of size different from  $2^n$ .

Future investigations will be aimed at the modeling of the offered algorithms, to permit the full evaluation of their characteristics. Additionally, the algorithm for tensor decomposition will be further developed to process tensors of any size (not cubical only). As a result, the most efficient applications will be detected for various areas such as data compression and filtration; the analysis, recognition and searching of visual information; deep learning, digital image watermarking, etc.

**Author Contributions:** Conceptualization, R.K.K., R.P.M. and R.A.K.; methodology, formal analysis, R.A.K.; investigation, R.P.M.; writing-original draft preparation, R.K.K.; writing-review and editing, R.A.K.; visualization, R.K.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the Bulgarian National Science Fund: Project No. KP-06-H27/16: "Development of efficient methods and algorithms for tensor-based processing and analysis of multidimensional images with application in interdisciplinary areas".

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Multi-Level Representation of One-Dimensional Hierarchical Hadamard Transforms

The mathematical descriptions of the n-level 1D-FWHT and 1D-FCHT are given below in correspondence with the transformation graphs from Figures 2 and 3, for  $N = 8$ . The factorization of the matrices used for the n-level transforms permits their execution based on matrix processors, which will additionally accelerate all calculations.

### Appendix A.1. Multi-Level Representation of Hierarchical 1D-FWHT

The  $n$ -level hierarchical frequency-ordered Walsh-Hadamard matrix  $H_W(N)$  of size  $N \times N$  for  $N = 2^n$  and  $n \geq 3$  can be represented as the matrix product of the following  $2n$  sparse matrices:

$$H_W(2^n) = \prod_{r=1}^n P_r(2^n) G_r(2^n), \quad (A1)$$

where

$$P_1(2^n) = L_1(2^2) \otimes I(2^{n-2}) = \begin{bmatrix} L_1(4) & 0 & - & 0 \\ 0 & L_1(4) & - & 0 \\ - & - & - & - \\ 0 & 0 & - & L_1(4) \end{bmatrix}, L_1(4) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, I(2^{n-2}) = \begin{bmatrix} 1 & 0 & - & 0 \\ 0 & 1 & - & 0 \\ - & - & - & - \\ 0 & 0 & - & 1 \end{bmatrix}; \quad (A2)$$

$$P_r(2^n) = L_2(2^2) \otimes I(2^{n-2}) = \begin{bmatrix} L_2(4) & 0 & - & 0 \\ 0 & L_2(4) & - & 0 \\ - & - & - & - \\ 0 & 0 & - & L_2(4) \end{bmatrix}, \text{ for } r = 2, 3, \dots, n, L_2(4) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}; \quad (A3)$$

$P_r(2^n)$  are permutation matrices of size  $2^n \times 2^n$  for level  $r = 1, 2, \dots, n$ ;  $I(2^{n-2})$  is the identity matrix,  $2^{n-2} \times 2^{n-2}$ ;

$$G_{r=1,2}(2^n) = I(2^{n-1}) \otimes H(2) = \begin{bmatrix} H(2) & 0 & - & 0 \\ 0 & H(2) & - & - \\ - & - & - & - \\ 0 & 0 & - & H(2) \end{bmatrix}, G_r(2^n) = \begin{bmatrix} A(2^r) & 0 & - & 0 \\ 0 & A(2^r) & - & - \\ - & - & - & - \\ 0 & 0 & - & A(2^r) \end{bmatrix} \text{ for } r = 3, 4, \dots, n; \quad (A4)$$

$$A(2^r) = H(2) \otimes I(2^{r-1}) = \begin{bmatrix} I(2^{r-1}) & I(2^{r-1}) \\ I(2^{r-1}) & -I(2^{r-1}) \end{bmatrix}. \quad (A5)$$

$G_r(2^n)$  are the sparse transform matrices of size  $2^n \times 2^n$  for the levels  $r = 1, 2, \dots, n$ , respectively; The algorithm 1D-FWHT is executed in  $n$  consecutive levels:

$$\begin{aligned} Y_1 &= P_1(2^n) G_1(2^n) X; \\ Y_r &= P_r(2^n) G_r(2^n) Y_{r-1} \text{ for } r = 2, 3, \dots, n-1; \\ E &= P_n(2^n) G_n(2^n) Y_{n-1}. \end{aligned} \quad (A6)$$

Here,  $X$  and  $E$  are the input and the output matrix-columns, respectively, and the matrices  $Y_r$  for  $r = 2, 3, \dots, n-1$  are the corresponding intermediate results. The input matrix  $X$  is obtained through slice transform mode  $-1$  of the tensor  $X$ . In the result of the 1D-FWHT execution, the so calculated components of the matrix  $E$  build the output tensor  $E$ . The matrix representation of the 1D-FWHT is consecutively applied in all three stages of the 3D transform, in accordance with the algorithm shown on Figure 4.

### Appendix A.2. Multi-Level Representation of Hierarchical 1D-FCHT

The hierarchical  $n$ -level frequency-ordered complex Hadamard matrix  $C(N)$  of size  $N \times N$  for  $N = 2^n$  and  $n \geq 3$  could be represented as the product of the following  $2n-1$  sparse matrices of order  $N$ , in accordance with [14]:

$$C(2^n) = \left[ \prod_{r=3}^n P_r(2^n) G_r(2^n) \right] P_1(2^n) C_J(2^n) G_1(2^n), \quad (A7)$$

where

$$G_1(2^n) = \begin{bmatrix} I(2) & I(2) \\ I(2) & -I(2) \end{bmatrix} \otimes I(2^{n-2}) = \begin{bmatrix} I(2) & I(2) & 0 & 0 & 0 & - & 0 \\ I(2) & -I(2) & 0 & 0 & 0 & - & 0 \\ 0 & 0 & I(2) & I(2) & 0 & - & - \\ 0 & 0 & I(2) & -I(2) & 0 & - & - \\ 0 & 0 & 0 & 0 & - & 0 & 0 \\ - & - & - & - & 0 & I(2) & I(2) \\ - & - & - & - & 0 & I(2) & -I(2) \end{bmatrix} \text{ for } I(2) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad (\text{A8})$$

$$C_J(2^n) = C_J(4) \otimes I(2^{n-2}) = \begin{bmatrix} C_J(4) & 0 & - & 0 \\ 0 & C_J(4) & - & - \\ - & - & - & 0 \\ - & - & 0 & C_J(4) \end{bmatrix} \text{ for } C_J(4) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & j \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -j \end{bmatrix}. \quad (\text{A9})$$

$I(2^{n-2})$  is the identity matrix of size  $2^{n-2} \times 2^{n-2}$ ; the sparse matrices  $G_r(2^n)$  of size  $2^n \times 2^n$  for levels  $r = 2, 3, \dots, n$  are defined by the relation (A4); the permutation matrix  $P_1(2^n)$ , by the relation (A2); and the permutation matrix  $P_r(2^n)$  for  $r = 3, 4, \dots, n-1$ , by (A3). The permutation matrix  $P_n(2^n)$  is used to rearrange the output matrix-column components in ascending order from the lowest to highest frequencies. For the case when  $2^n = 8$ , the matrix  $P_n(2^n) = P_2(2^3)$  is:

$$P_2(8) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (\text{A10})$$

In the general case, for the rearrangement of the components in the  $n$ -th level should be inversely followed the binary code  $e_{n-1}, e_{n-2}, \dots, e_0$  of the sequential number  $0, 1, \dots, 2^n-1$  of the output component (i.e.,  $g_i = e_{n-i-1}$  for  $0 \leq i \leq n-1$ ). Then, the obtained values  $g_{n-1}, g_{n-2}, \dots, g_0$  are transformed from Gray code into binary  $b_{n-1}, b_{n-2}, \dots, b_0$ , in accordance with the rule [9]:

$$b_{n-1} = g_{n-1}, \quad b_i = b_{i-1} \oplus g_i \text{ for } 0 \leq i \leq n-2. \quad (\text{A11})$$

The algorithm 1D-FCFT is executed in  $n$  consecutive levels:

$$\begin{aligned} Y_1 &= P_1(2^n)C_J(2^n)G_1(2^n)X \text{ - for levels 1 and 2;} \\ Y_{r-1} &= P_{r-1}(2^n)G_{r-1}(2^n)Y_{r-2} \text{ for } n > 3 \text{ and } r = 3, 4, \dots, n; \\ E &= P_n(2^n)G_n(2^n)Y_{n-1} \text{ - for level } n. \end{aligned} \quad (\text{A12})$$

## References

- Li, Y.; Wang, Q.; Li, X.; Liu, J. A Survey on tensor techniques and applications in machine learning. *IEEE Access* **2019**, *7*, 162950–162990. [\[CrossRef\]](#)
- Rabanser, S.; Shchur, O.; Günnemann, S. Introduction to tensor decompositions and their applications in machine learning. *arXiv* **2017**, arXiv:1711.10781v1 [stat.ML].
- Cichocki, A.; Mandic, D.; Phan, A.-H.; Caiafa, C.; Zhou, G.; Zhao, Q.; De Lathauwer, L. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Process Mag.* **2015**, *32*, 145–163. [\[CrossRef\]](#)
- Oseledets, I. Tensor-train decomposition. *SIAM J. Sci. Comput.* **2011**, *33*, 2295–2317. [\[CrossRef\]](#)

5. Grasedyck, L. Hierarchical singular value decomposition of tensors. *SIAM J. Matrix Anal. Appl.* **2010**, *31*, 2029–2054. [[CrossRef](#)]
6. Ozdemir, A.; Zare, A.; Iwen, M.; Aviyente, S. Extension of PCA to higher order data structures: An introduction to tensors, tensor decompositions, and tensor PCA. *Proc. IEEE* **2018**, *106*, 1341–1358. [[CrossRef](#)]
7. Wang, P.-A.; Lu, C.-J. Tensor decomposition via simultaneous power iteration, Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia. *Proc. Mach. Learn. Res. (PMLR)* **2017**, *70*, 3665–3673.
8. Ishteva, M.; Absil, P.; Van Dookken, P. Jacoby algorithm for the best low multi-linear rank approximation of symmetric tensors. *SIAM J. Matrix Anal. Appl.* **2013**, *34*, 651–672. [[CrossRef](#)]
9. Ahmed, N.; Rao, K. Walsh-Hadamard transform. In *Orthogonal Transforms for Digital Signal Processing*; Springer: Berlin/Heidelberg, Germany, 1975; pp. 99–152.
10. Horadam, K. Hadamard matrices, their applications and generalisations. In *Hadamard Matrices and Their Applications*; Princeton University Press: Princeton, NJ, USA, 2007.
11. Agaian, S.; Sarukhanyan, H.; Egiazarian, K.; Astola, J. Classical Hadamard matrices and arrays. In *Hadamard Transforms*; SPIE Press Book: Washington, DC, USA, 2011; pp. 1–50.
12. Yaroslavsky, L. Hadamard, Walsh, Wavelet and Other Transforms. In *Fast Transform Methods in Digital Signal Processing: Theory, Applications, Efficient Algorithms*; Bentham eBooks: Saif Zone Sharjah, UAE, 2011. [[CrossRef](#)]
13. Aung, A.; Ng, B. Natural-ordered complex Hadamard transform. *Signal Process.* **2010**, *90*, 874–879. [[CrossRef](#)]
14. Mironov, R.; Kountchev, R. Complex Hadamard transform of digital signals: Properties and applications. *Int. J. Reason. Based Intell. Syst. (IJRIS)* **2012**, *4*, 1–16. [[CrossRef](#)]
15. Rao, K.; Kim, D.; Hwang, J. Two-dimensional discrete Fourier transform. In *Fast Fourier Transform: Algorithms and Applications*; Springer: Heidelberg, Germany, 2010; pp. 127–184.
16. Testoni, V.; Costa, M. Entropy coders and 3D-Hadamard coefficients sequency scan order for a fast embedded color video codec. *Comput. Electr. Eng.* **2008**, *36*, 676–690. [[CrossRef](#)]
17. Scheibler, R.; Haghighatshoar, S.; Vetterli, M. A fast Hadamard transform for signals with sub-linear sparsity in the transform domain. *IEEE Trans. Inf. Theory* **2015**, *61*, 2115–2132. [[CrossRef](#)]
18. Kountchev, R.; Inatovicz, B.; Kountcheva, R. Hierarchical third-order tensor decomposition through Inverse difference pyramid, based on the 3D Walsh-Hadamard transform with applications in data mining. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2020**, *10*, e1314. [[CrossRef](#)]
19. Sidiropoulos, N.; De Lathauwer, L.; Fu, X.; Huang, K.; Papalexakis, E.; Faloutsos, C. Tensor decomposition for signal processing and machine learning. *arXiv* **2016**, arXiv:1607.01668v2 [stat.ML]. [[CrossRef](#)]
20. Kountchev, R.; Mironov, R.; Kountcheva, R. Complexity evaluation of tensor decomposition through 3D Inverse spectrum pyramid in respect of deterministic orthogonal transforms. *WSEAS Trans. Signal Process.* **2019**, *15*, 142–148.

