symmetry

MDPI

Article

Complexity Estimation of Cubical Tensor Represented through 3D Frequency-Ordered Hierarchical KLT

Roumen Kountchev [1,*], Rumen Mironov [1] and Roumiana Kountcheva [2]

[1] Faculty of Telecommunications, Department Radio Communications and Video Technologies, Technical University of Sofia, 1000 Sofia, Bulgaria; rmironov@tu-sofia.bg

[2] TK Engineering, 1582 Sofia, Bulgaria; kountcheva_r@yahoo.com

* Correspondence: rkountch@tu-sofia.bg; Tel.: +359-899924284

Received: 6 August 2020; Accepted: 25 September 2020; Published: 26 September 2020

Abstract: In this work is introduced one new hierarchical decomposition for cubical tensor of size 2^n, based on the well-known orthogonal transforms Principal Component Analysis and Karhunen–Loeve Transform. The decomposition is called 3D Frequency-Ordered Hierarchical KLT (3D-FOHKLT). It is separable, and its calculation is based on the one-dimensional Frequency-Ordered Hierarchical KLT (1D-FOHKLT) applied on a sequence of matrices. The transform matrix is the product of n sparse matrices, symmetrical at the point of their main diagonal. In particular, for the case in which the angles which define the transform coefficients for the couples of matrices in each hierarchical level of 1D-FOHKLT are equal to $\pi/4$, the transform coincides with this of the frequency-ordered 1D Walsh–Hadamard. Compared to the hierarchical decompositions of Tucker (H-Tucker) and the Tensor-Train (TT), the offered approach does not ensure full decorrelation between its components, but is close to the maximum. On the other hand, the evaluation of the computational complexity (CC) of the new decomposition proves that it is lower than that of the above-mentioned similar approaches. In correspondence with the comparison results for H-Tucker and TT, the CC decreases fast together with the increase of the hierarchical levels' number, n. An additional advantage of 3D-FOHKLT is that it is based on the use of operations of low complexity, while the similar famous decompositions need large numbers of iterations to achieve the coveted accuracy.

Keywords: cubical tensor decomposition; 3D hierarchical adaptive PCA transform; 3D Frequency-Ordered Hierarchical KLT; computational complexity

1. Introduction

The well-known tensor decompositions Canonical Polyadic Decomposition (CPD), Higher-Order Singular Value Decomposition (HOSVD) [1–3], Tensor Train (TT) Decomposition [4], Hierarchical Tucker decomposition (H-Tucker) [5] and some of their modifications [6,7] are based on the calculation of the eigenvalues and eigenvestors of the decomposed tensor. The basic attribute of this group is that they are optimal regarding the minimization of the mean square approximation error derived from the "truncation" of the low-energy components. For the calculation of the retained components, we use iterative methods which need relatively large numbers of mathematical operations to achieve the requested accuracy (Power method [8], Jacoby method [9], etc.).

The hierarchical tensor decompositions based on the H-Tucker algorithm are generalized in publications [7] and [10]. In [7], Vasilescu and Kim introduced a compositional hierarchical tensor factorization that disentangles the hierarchical causal structure of object image formation, but the computational complexity (CC) of the method is not evaluated. In [10], Zniyed et al. offered the TT-based hierarchical decomposition of high-order tensors, for the calculation of which is used the Tensor-Train Hierarchical SVD (TT-HSVD). This approach permits parallel processing, which accelerates the calculations. Unlike the TT-SVD algorithm, the TT-HSVD is based on applying SVDs to matrices of smaller dimensions. As a result, the CC of TT-SVD is O(R^2N^3) operations higher than that of TT-HSVD (here, R is the rank of the hypercubical tensor of size N).

As an alternative, in this work are presented new hierarchical 3D tensor decompositions, based on the famous statistical orthogonal transforms Principal Component Analysis (PCA) and the Karhunen–Loeve Transform (KLT) [11,12]. The offered decompositions are close to the optimal (which ensure full decorrelation of their components), but do not need iterations and have low computational complexity.

In Section 2, we present the methods for one-dimensional hierarchical adaptive transform based on the PCA (1D-HAPCA) and the Hierarchical Adaptive Karhunen–Loeve Transform (1D-HAKLT). In Section 3, on the basis of these 1D transforms we give the details of the cubical tensor decomposition through separable 3D Frequency-Ordered Hierarchical KLT (3D-FOHKLT), and Section 4 contains the related algorithm. Section 5 presents the comparative analysis of the computational complexity of the new approaches with respect to the well-known hierarchical Tucker decomposition (H-Tucker) and the TT decomposition. Section 6 is the Conclusion.

2. One-Dimensional Hierarchical Adaptive PCA/KLT Transform for a Sequence of Matrices

The detailed description of the algorithm for the one-dimensional adaptive PCA/KLT transform (1D-APCA/AKLT) for a couple of matrices is given below. This is the basis of the hierarchical algorithm for processing the sequence of matrices which represent the corresponding tensor.

Throughout this paper, scalars are represented as x, vectors and matrices as **x** and **X** correspondingly, and tensors as $\boldsymbol{\mathcal{X}}$. It is also assumed here that all tensor elements (entries) are nonnegative real numbers x(i,j,k).

2.1. One-Dimensional Adaptive PCA/KLT Transform for a Sequence of Matrices

The transform is executed in the following way.

First, on the elements x_{1s} and x_{2s}, which have the same spatial position in the couple of matrices $\mathbf{X}_1$ and $\mathbf{X}_2$ of size N × N, are applied one-dimensional direct and inverse adaptive PCA, i.e., the 1D-APCA [12] is:

$$\begin{gathered}
y_{1s} = (x_{1s} - m_1)\cos\theta_{1,2} + (x_{2s} - m_2)\sin\theta_{1,2}; \quad x_{1s} = y_{1s}\cos\theta_{1,2} + y_{2s}\sin\theta_{1,2} + m_1; \\
y_{2s} = (x_{1s} - m_1)\sin\theta_{1,2} - (x_{2s} - m_2)\cos\theta_{1,2}; \quad x_{2s} = y_{1s}\sin\theta_{1,2} - y_{2s}\cos\theta_{1,2} + m_2, \\
\text{fors} = 1,2,\ldots,N^2
\end{gathered} \tag{1}$$

Here, y_{1s} and y_{2s} are the corresponding elements in the couple of transformed matrices $\mathbf{Y}_1$ and $\mathbf{Y}_2$, each of size N × N,

$$\theta_{1,2} = \mathrm{arctg}\{2k_3/(k_1 - k_2 + \sqrt{(k_1 - k_2)^2 + 4k_3^2})\}; \tag{2}$$

$$k_1 = E(x_{1s}^2) - m_1^2, k_2 = E(x_{2s}^2) - m_2^2, k_3 = E(x_{1s}x_{2s}) - m_1m_2; \tag{3}$$

$$m_1 = E(x_{1s}), m_2 = E(x_{2s}), \tag{4}$$

where $m = E(x_s) = (1/N^2)\sum_{s=1}^{N^2} x_s$ is the averaging operator.

To calculate the angle $\theta_{1,2}$ it is necessary to define the extended arctg function. If $\beta_2 = 2k_3$ and $\beta_1 = k_1 - k_2 + \sqrt{(k_1 - k_2)^2 + 4k_3^2}$ are two real numbers, the extended arctg function $\varepsilon_{\mathrm{arctg}}(\beta_1, \beta_2)$ is mapped to $(-\pi, \pi]$ by adding π [13]:

$$\varepsilon_{\mathrm{arctg}}(\beta_1, \beta_2) = \begin{cases} \mathrm{arctg}(\beta_2/\beta_1), \text{ if } \beta_1 > 0; \\ \mathrm{arctg}(\beta_2/\beta_1) + \pi, \text{ if } \beta_1 < 0,\ \beta_2 \geq 0; \\ \mathrm{arctg}(\beta_2/\beta_1) - \pi, \text{ if } \beta_1 < 0, \beta_2 < 0; \\ \pi/2, \text{ if } \beta_1 = 0,\ \beta_2 > 0; \\ -\pi/2, \text{ if } \beta_1 = 0, \beta_2 < 0; \\ 0, \text{ if } \beta_2 = \beta_1 = 0. \end{cases} \tag{5}$$

In particular, for $m_1 = m_2 = 0$ (valid for 1D-AKLT) and $\theta_{1,2} = \pi/4$, the relations (1) become as follows:

$$\begin{array}{ll} y_{1s} = (1/\sqrt{2})(x_{1s} + x_{2s}); & x_{1s} = (1/\sqrt{2})(y_{1s} + y_{2s}); \\ y_{2s} = (1/\sqrt{2})(x_{1s} - x_{2s}); & x_{2s} = (1/\sqrt{2})(y_{1s} - y_{2s}). \end{array} \tag{6}$$

Hence, in this case the 1D-KLT coincides with the 1D Walsh–Hadamard Transform (1D-WHT).

2.2. One-Dimensional Hierarchical Adaptive KLT for a Sequence of Matrices

The relations above, used for the calculation of 1D-APCA [12], are the kernel for the processing of a sequence of matrices (for example, a sequence of moving images), and do not need iterative calculations. The procedure is executed in the following way. First, the sequence of matrices X_k, each of size N × N for k = 1, 2, . . . , N is divided into N/2 couples (N = 2^n). For each couple of matrices with elements x_{1s}(p,q) and x_{2s}(p,q), when q = 1, 2, . . . , N/2, in the level p = 1, 2, . . . , n, is executed the one-dimensional adaptive PCA (1D-APCA), in accordance with Equations (1)–(5). For $m_1(p,q) = m_2(p,q) = 0$ (valid for the KLT), the direct 1D-AKLT for the couple of matrices q in the level p of the 1D-HAKLT is defined by the relations

$$\begin{array}{l} y_{1s}(p,q) = x_{1s}(p,q)\cos\theta_{p,q} + x_{2s}(p,q)\sin\theta_{p,q} \\ y_{2s}(p,q) = x_{1s}(p,q)\sin\theta_{p,q} - x_{2s}(p,q)\cos\theta_{p,q} \end{array} \text{ for } s = 1, 2, \ldots, N^2, \tag{7}$$

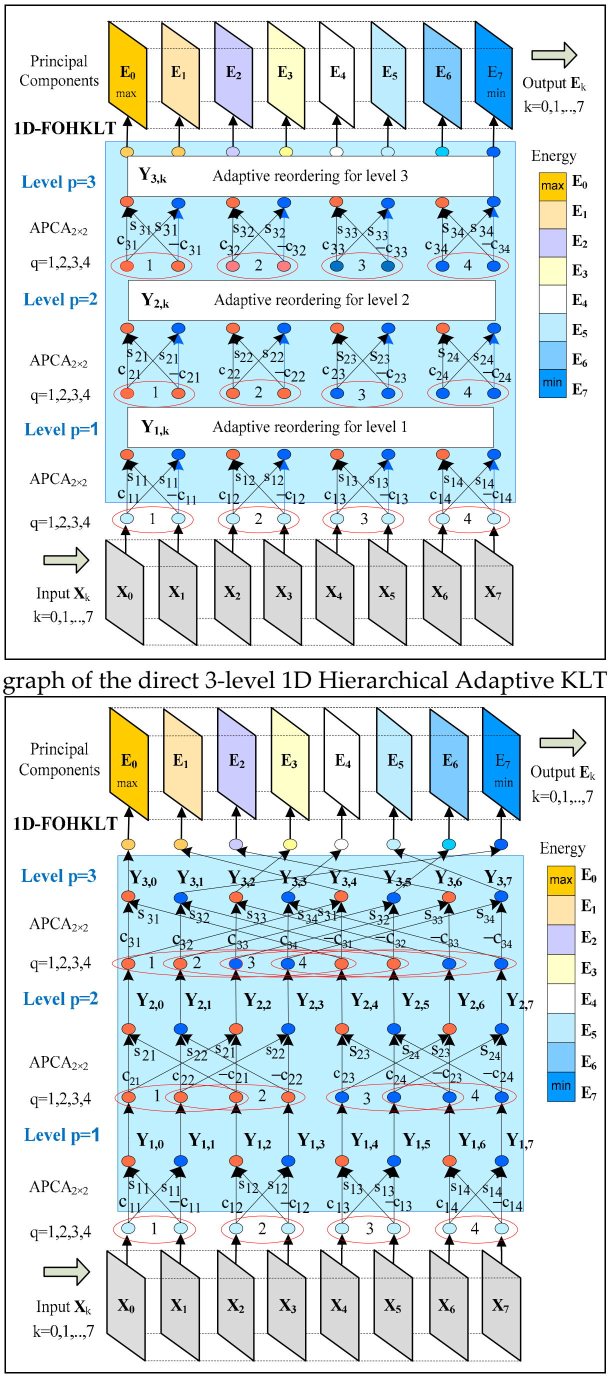
where s is the sequential number of two elements of the same position in the couple of matrices, q,

$$\begin{array}{c} \theta_{p,q} = \mathrm{arctg}\left\{2k_3(p,q)/[k_1(p,q) - k_2(p,q) + \sqrt{(k_1(p,q) - k_2(p,q))^2 + 4k_3^2(p,q)]}\right\}, \\ k_1(p,q) = E(x_{1s}^2(p,q)); k_2(p,q) = E(x_{2s}^2(p,q)); k_3(p,q) = E(x_{1s}(p,q)x_{2s}(p,q)). \end{array} \tag{8}$$

In the equations, figures and text below are introduced the following symbols: $\cos\theta_{p,q} = c_{p,q}$ and $\sin\theta_{p,q} = s_{p,q}$ for p = 1, 2, 3 and q = 1, 2, 3, 4. The execution graph for 1D-HAKLT of three hierarchical levels (N = 8) is shown on Figure 1a. In every level p = 1, 2, 3, and for each couple of neighbor matrices q (fenced in red ellipses) 1D-APCA is executed. In result, the first transformed matrix calculated for each couple q has higher energy than the second matrix, and they are mutually decorrelated. To continue with the operations in the next level of 1D-HAKLT, these matrices are reordered, following the decrease of their energy, i.e.,

$$P_{\mathbf{Y}_{p,0}} \geq P_{\mathbf{Y}_{p,1}} \geq P_{\mathbf{Y}_{p,2}} \geq \ldots P_{\mathbf{Y}_{p,k}} \geq \ldots \geq P_{\mathbf{Y}_{p,7}}, \tag{9}$$

where $P_{\mathbf{Y}_{p,k}} = \sum\limits_{i=1}^{8}\sum\limits_{j=1}^{8}[y_{p,k}(i,j)]^2$ is the energy of matrices $\mathbf{Y}_{p,k}$ with elements $y_{p.k}(i,j)$, obtained in result of the execution of 1D-HAKLT in levels p = 1, 2, 3, before the reordering. In result, the reordered matrices $\mathbf{E_r}$ in the last level are decorrelated to the highest degree [12].



(**a**) Execution graph of the direct 3-level 1D Hierarchical Adaptive KLT (1D-HAKLT)

(**b**) Execution graph of the direct 3-level 1D Frequency-Ordered Hierarchical KLT (1D-FOHKLT)

Figure 1. Execution graphs of the direct 3-level transforms 1D-HAKLT and 1D-FOHKLT.

The reordering in each level p in accordance with Equation (9) needs significant computational resources. For the case when the connections between the outputs $Y_{p,k}$ for $k = 1, 2, \ldots, 8$ in the level p and the inputs in the next level $p + 1$ are fixed, reordering is not performed, and the 1D-HAKLT turns into the one-dimensional Frequency-Ordered Hierarchical KLT (1D-FOHKLT), whose execution graph is shown in Figure 1b. In this case, only the output matrices in the last (third) level are reordered, which does not require mathematical operations. Besides, the energy distribution of matrices $\mathbf{Y}_{p,k}$ is close to that defined by Equation (9). In particular, in accordance with Appendix A, if $\theta_{p,q} = \pi/4$ for $p = 1, 2, 3$ and $q = 1, 2, 3, 4$, then 1D-HAKLT coincides with the one-dimensional frequency-ordered Fast Walsh–Hadamard Transform (1D-FWHT) for $N = 8$ [14].

Both algorithms are generalized below for N = 2^n. The algorithm for fast hierarchical 1D-HAKLT is represented through the sequential transformation of the vectors' elements s in each decomposition level p, in accordance with the relations

$$\mathbf{y}_{1,s} = \mathbf{P}_1(2^n)\mathbf{G}_1(2^n)x_s \text{for } p = 1 \text{ and } s = 1, 2, \ldots, N^2; \tag{10}$$

$$\mathbf{y}_{p,s} = \mathbf{P}_p(2^n)\mathbf{G}_p(2^n)\mathbf{y}_{p-1,s} \text{for } p = 2, 3, \ldots, n-1 \text{ and } s = 1, 2, \ldots, N^2; \tag{11}$$

$$\mathbf{e}_s = \mathbf{P}_n(2^n)\mathbf{G}_n(2^n)\mathbf{y}_{n-1,s}.\text{forp} = n \text{ and } s = 1, 2, \ldots, N^2. \tag{12}$$

Here, $\mathbf{x}_s$ and $\mathbf{e}_s$ are respectively the input and the output column-vector of N components correspondingly, and the column-vectors $\mathbf{y}_{p,s}$ for p = 2, 3, … , n − 1 are the intermediate results. The relation between vectors $\mathbf{e}_s$ and $\mathbf{x}_s$, defined on the basis of Equations (10)–(12), is given below:

$$\mathbf{e}_s = [\prod_{p=1}^{n} \mathbf{P}_p(2^n)\mathbf{G}_p(2^n)]x_s = \mathbf{T}_{\text{HAKLT}}(2^n)x_s \text{ for } s = \ 1, 2, \ \ldots, N^2, \tag{13}$$

where

$$\mathbf{T}_{\text{HAKLT}}(2^n) = \prod_{p=1}^{n} \mathbf{P}_p(2^n)\mathbf{G}_p(2^n) \tag{14}$$

is the matrix for the fast n-level 1D-HAKLT. The sparse transform matrices $\mathbf{G}_p(2^n)$ of size $2^n \times 2^n$ have a symmetrical block-diagonal structure towards the main diagonal, and are defined by the relations

$$\mathbf{G}_p(2^n) = \bigoplus_{q=1}^{2^{n-1}} \mathbf{T}_{p,q}(2) = \text{diag}\{\mathbf{T}_{p,1}(2), \mathbf{T}_{p,2}(2), \ldots, \mathbf{T}_{p,2^{n-1}}(2)\} \text{ for } p = 1,\ 2,\ \ldots,\ n, \tag{15}$$

where $\mathbf{T}_{p,q}(2) = \begin{bmatrix} c_{p,q} & s_{p,q} \\ s_{p,q} & -c_{p,q} \end{bmatrix}$, and ⊕ denotes the direct sum of matrices. The permutation matrices $\mathbf{P}_p(2^n)$ of size $2^n \times 2^n$ for level p = 1, 2, … , n of 1D-HAKLT are defined on the basis of Equation (9) for k = 0, 1, 2, … , 2^{n-1}.

The algorithm 1D-FOHKLT is depicted by a relation similar to Equation (13), but the reordering is executed only in the last level, n;

$$\mathbf{e}_s = \mathbf{P}_n(2^n)[\prod_{p=1}^{n} \mathbf{G}_p(2^n)]x_s = \mathbf{T}_{\text{FOHKLT}}(2^n)x_s \text{ for } s = \ 1,\ 2,\ \ldots,\ N^2, \tag{16}$$

where $\mathbf{T}_{\text{FOHKLT}}(2^n) = \mathbf{P}_n(2^n)[\prod_{p=1}^{n} \mathbf{G}_p(2^n)]$ is the matrix for 1D-FOHKLT, $\mathbf{P}_n(2^n)$ is the permutation matrix $\mathbf{P}_n(2^n)$ for the last level n, and $\prod_{p=1}^{n} \mathbf{G}_p(2^n)$ is the product of n sparse transform matrices $\mathbf{G}_p(2^n)$ for p = 1, 2, 3, … , n.

Each matrix $\mathbf{G}_p(2^n)$ is defined as follows:

$$\mathbf{G}_p(2^n) = \bigoplus_{i=1}^{2^{n-p}} \begin{bmatrix} \bigoplus_{j=1}^{2^{p-1}} c_{p,2^{p-1}(i-1)+j} & \bigoplus_{j=1}^{2^{p-1}} s_{p,2^{p-1}(i-1)+j} \\ \bigoplus_{j=1}^{2^{p-1}} s_{p,2^{p-1}(i-1)+j} & \bigoplus_{j=1}^{2^{p-1}} -c_{p,2^{p-1}(i-1)+j} \end{bmatrix} \text{for } p = 1,\ 2,\ 3, \ldots,\ n, \tag{17}$$

where

$$\bigoplus_{i=1}^{2^{n-p}}\begin{bmatrix}\bigoplus_{j=1}^{2^{p-1}} c_{p,2^{p-1}(i-1)+j} & \bigoplus_{j=1}^{2^{p-1}} s_{p,2^{p-1}(i-1)+j}\\ \bigoplus_{j=1}^{2^{p-1}} s_{p,2^{p-1}(i-1)+j} & \bigoplus_{j=1}^{2^{p-1}} -c_{p,2^{p-1}(i-1)+j}\end{bmatrix}=\begin{bmatrix}\bigoplus_{j=1}^{2^{p-1}} c_{p,j} & \bigoplus_{j=1}^{2^{p-1}} s_{p,j}\\ \bigoplus_{j=1}^{2^{p-1}} s_{p,j} & \bigoplus_{j=1}^{2^{p-1}} -c_{p,j}\end{bmatrix}\oplus$$
$$\oplus\begin{bmatrix}\bigoplus_{j=1}^{2^{p-1}} c_{p,2^{p-1}+j} & \bigoplus_{j=1}^{2^{p-1}} s_{p,2^{p-1}+j}\\ \bigoplus_{j=1}^{2^{p-1}} s_{p,2^{p-1}+j} & \bigoplus_{j=1}^{2^{p-1}} -c_{p,2^{p-1}+j}\end{bmatrix}\oplus\ldots\oplus\begin{bmatrix}\bigoplus_{j=1}^{2^{p-1}} c_{p,2^{p-1}(n-p-1)+j} & \bigoplus_{j=1}^{2^{p-1}} s_{p,2^{p-1}(n-p-1)+j}\\ \bigoplus_{j=1}^{2^{p-1}} s_{p,2^{p-1}(n-p-1)+j} & \bigoplus_{j=1}^{2^{p-1}} -c_{p,2^{p-1}(n-p-1)+j}\end{bmatrix}. \quad (18)$$

In the level n are rearranged the components of the column-vectors $\mathbf{y}_{n,s} = \mathbf{G}_n(2^n)\mathbf{y}_{n-1,s}$, and, respectively, the components of matrices $\mathbf{Y}_{n,k}$ for k = 0, 1, ... , N − 1. For this is used the permutation matrix $\mathbf{P}_n(2^n)$. From the components of the column-vectors $\mathbf{e}_s = \mathbf{P}_n(2^n)\mathbf{y}_{n,s}$ are obtained the matrices $\mathbf{E}_0, \mathbf{E}_1, \ldots, \mathbf{E}_{N-1}$. The frequency-ordered matrices $\mathbf{E}_r$ are calculated in accordance with the relation between their sequential number r and the corresponding sequential number k, for the matrices $\mathbf{Y}_{n,k}$. The relation which defines the matrix $\mathbf{P}_n(2^n)$ is as follows [14]:

- the binary code $k_{n-1}, k_{n-2}, \ldots, k_0$ of the sequential decimal number $k = 0, 1, \ldots, 2^{n-1}$ of the component $\mathbf{Y}_{n,k}$ is arranged inversely (i.e., $k_0, k_1, \ldots, k_{n-1}$), as follows:

$$g_i = k_{n-i-1} \text{ for } 0 \le i \le n-1; \quad (19)$$

- the so-obtained code $g_{n-1}, g_{n-2}, \ldots, g_0$ is transformed from Gray code into binary code $r_{n-1}, r_{n-2}, \ldots, r_0$, in accordance with the operations

$$r_{n-1} = g_{n-1},\ r_i = r_{i-1} \oplus g_i \text{ for } 0 \le i \le n{-}2. \quad (20)$$

Here, $\oplus$ is the symbol for the operation "exclusive OR". The decimal number $r = \sum_{i=0}^{n-1} r_i 2^i$ defines the sequential number of the component $\mathbf{E}_r$ for $r = 0, 1, \ldots, 2^{n-1}$, which before the rearrangement corresponded to the component $\mathbf{Y}_{n,k}$, with sequential number $k = \sum_{i=0}^{n-1} k_i 2^i$.

An example for the direct 1D-FOHKLT of three hierarchical levels is given in Appendix A.

3. Decomposition of Cubical Tensor of Size N × N × N through 3D-FOHKLT

The decomposition 3D-FOHKLT for a cubical tensor $\mathcal{X}$ of size 8 × 8 × 8, executed in three consecutive stages, is shown on Figure 2.

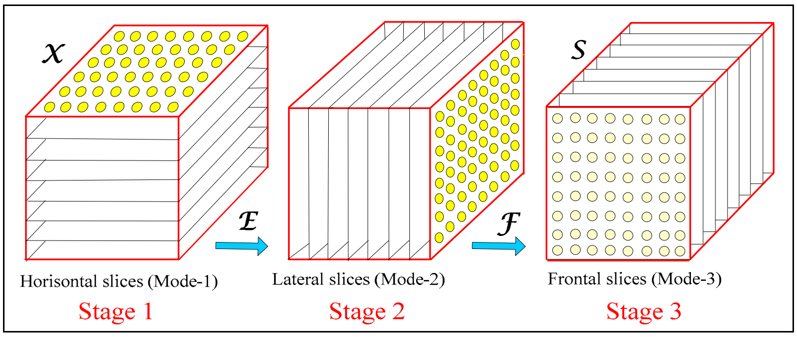


Figure 2. Stages of 3D-FOHKLT for the tensor $\mathcal{X}$ of size 8 × 8 × 8, based on the 1D-FOKLT.

In the first stage, the tensor $\mathcal{X}$ is divided into eight horizontal slices (*mode-1*). Then, in accordance with Figure 1, on each couple of matrices is applied 1D-FOHKLT. In result, the sequence of eight transformed matrices is united into one new tensor, $\boldsymbol{\mathcal{E}}$. In the second stage, the input tensor $\boldsymbol{\mathcal{E}}$ is divided into eight lateral slices (*mode-2*), then the couples of matrices are transformed, and after that united into the tensor $\boldsymbol{\mathcal{F}}$. In the third stage, the input tensor $\boldsymbol{\mathcal{F}}$ is divided into eight frontal slices (*mode-3*). After applying the 1D-AKLT on each couple of matrices which build the tensor $\boldsymbol{\mathcal{F}}$, the final result is obtained—the general spectral tensor $\boldsymbol{\mathcal{S}}$, with elements $\mathrm{s(u,v,l)}$. The initial tensor $\boldsymbol{\mathcal{X}}$ is restored from the tensor $\boldsymbol{\mathcal{S}}$ through summation of all "weighted" basic tensors $\boldsymbol{\mathcal{K}}_{\mathrm{u,v,l}}$ of size 8 × 8 × 8, in accordance with the relation

$$\boldsymbol{\mathcal{X}} = (1/512)[\mathrm{s}(0,0,0)\boldsymbol{\mathcal{K}}_{0,0,0} + \mathrm{s}(1,0,0)\boldsymbol{\mathcal{K}}_{1,0,0} + \ldots + \mathrm{s}(7,7,7)\boldsymbol{\mathcal{K}}_{7,7,7}] = (1/8^3)\sum_{\mathrm{u}=0}^{7}\sum_{\mathrm{v}=0}^{7}\sum_{\mathrm{l}=0}^{7}\mathrm{s(u,v,l)}\boldsymbol{\mathcal{K}}_{\mathrm{u,v,l}}. \tag{21}$$

Here, $\mathrm{s(u,v,l)}$ are the weight coefficients of the basic tensor, $\boldsymbol{\mathcal{K}}_{\mathrm{u,v,l}}$. Each basic tensor is represented as the outer product of three vectors:

$$\boldsymbol{\mathcal{K}}_{\mathrm{u,v,l}} = \mathbf{k}_{\mathrm{u}}(1) \circ \mathbf{k}_{\mathrm{v}}(2) \circ \mathbf{k}_{\mathrm{l}}(3), \tag{22}$$

where o is the symbol for the outer product of the two column-vectors ($\mathbf{x} \circ \mathbf{y} = \mathbf{x}.\mathbf{y}^{\mathrm{T}}$) and $\mathbf{k}_{\mathrm{u}}(1), \mathbf{k}_{\mathrm{v}}(2), \mathbf{k}_{\mathrm{l}}(3)$ are the basic vectors for the first, second and third stages of 3D-FOHKLT, respectively. These vectors are defined by using the matrix $\mathbf{T}_{\mathrm{FOHKLT}}(8)$ for 1D-FOHKLT, in accordance with Equation (A6) from Appendix A. The vector $\mathbf{k}_{\mathrm{u}}(1)$ for u = 0, 1, ... , 7 comprises the elements of the u-th row of the matrix $\mathbf{T}^{1}_{\mathrm{FOHKLT}}(8)$ for the first stage (t = 1) of 3D-FOHKLT. The vectors $\mathbf{k}_{\mathrm{v}}(2)$ and $\mathbf{k}_{\mathrm{l}}(3)$ for v, l = 0, 1, ... , 7 comprise the elements of rows v and l in the matrices $\mathbf{T}^{2}_{\mathrm{FOHKLT}}(8)$ and $\mathbf{T}^{3}_{\mathrm{FOHKLT}}(8)$ for the second (t = 2) and third (t = 3) stages of 3D-FOHKLT, respectively. The basic vectors are defined below for t = 1, 2, 3 and u, v, l = 0, 1, ... , 7:

$$\mathbf{k}_{\mathrm{u}}(\mathrm{t}) = [\mathrm{k}^{\mathrm{t}}_{\mathrm{u},1}, \mathrm{k}^{\mathrm{t}}_{\mathrm{u},2}, \mathrm{k}^{\mathrm{t}}_{\mathrm{u},3}, \mathrm{k}^{\mathrm{t}}_{\mathrm{u},4}, \mathrm{k}^{\mathrm{t}}_{\mathrm{u},5}, \mathrm{k}^{\mathrm{t}}_{\mathrm{u},6}, \mathrm{k}^{\mathrm{t}}_{\mathrm{u},7}, \mathrm{k}^{\mathrm{t}}_{\mathrm{u},8}]^{\mathrm{T}}; \tag{23}$$

$$\mathbf{k}_{\mathrm{v}}(\mathrm{t}) = [\mathrm{k}^{\mathrm{t}}_{\mathrm{v},1}, \mathrm{k}^{\mathrm{t}}_{\mathrm{v},2}, \mathrm{k}^{\mathrm{t}}_{\mathrm{v},3}, \mathrm{k}^{\mathrm{t}}_{\mathrm{v},4}, \mathrm{k}^{\mathrm{t}}_{\mathrm{v},5}, \mathrm{k}^{\mathrm{t}}_{\mathrm{v},6}, \mathrm{k}^{\mathrm{t}}_{\mathrm{v},7}, \mathrm{k}^{\mathrm{t}}_{\mathrm{v},8}]^{\mathrm{T}}; \tag{24}$$

$$\mathbf{k}_{\mathrm{l}}(\mathrm{t}) = [\mathrm{k}^{\mathrm{t}}_{\mathrm{l},1}, \mathrm{k}^{\mathrm{t}}_{\mathrm{l},2}, \mathrm{k}^{\mathrm{t}}_{\mathrm{l},3}, \mathrm{k}^{\mathrm{t}}_{\mathrm{l},4}, \mathrm{k}^{\mathrm{t}}_{\mathrm{l},5}, \mathrm{k}^{\mathrm{t}}_{\mathrm{l},6}, \mathrm{k}^{\mathrm{t}}_{\mathrm{l},7}, \mathrm{k}^{\mathrm{t}}_{\mathrm{l},8}]^{\mathrm{T}}. \tag{25}$$

The general number of the basic tensors in this case is 8^3. For example, if u = 3, v = 2 and l = 6, the components of these vectors for each stage are defined by the relations below:

$$\begin{aligned}\mathbf{k}_3(1) &= [\mathrm{k}^{1}_{3,0}, \mathrm{k}^{1}_{3,1}, \mathrm{k}^{1}_{3,2}, \mathrm{k}^{1}_{3,3}, \mathrm{k}^{1}_{3,4}, \mathrm{k}^{1}_{3,5}, \mathrm{k}^{1}_{3,6}, \mathrm{k}^{1}_{3,7}]^{\mathrm{T}} = \\ &= [\mathrm{c}^{1}_{3,3}\mathrm{s}^{1}_{2,1}\mathrm{c}^{1}_{1,1}, \mathrm{c}^{1}_{3,3}\mathrm{s}^{1}_{2,1}\mathrm{s}^{1}_{1,1}, -\mathrm{c}^{1}_{3,3}\mathrm{c}^{1}_{2,1}\mathrm{c}^{1}_{1,2}, -\mathrm{c}^{1}_{3,3}\mathrm{c}^{1}_{2,1}\mathrm{s}^{1}_{1,2}, \mathrm{s}^{1}_{3,3}\mathrm{s}^{1}_{2,3}\mathrm{c}^{1}_{1,3}, \mathrm{s}^{1}_{3,3}\mathrm{s}^{1}_{2,3}\mathrm{s}^{1}_{1,3}, -\mathrm{s}^{1}_{3,3}\mathrm{c}^{1}_{2,3}\mathrm{c}^{1}_{1,4}, -\mathrm{s}^{1}_{3,3}\mathrm{c}^{1}_{2,3}\mathrm{s}^{1}_{1,4}]^{\mathrm{T}};\end{aligned} \tag{26}$$

$$\begin{aligned}\mathbf{k}_2(2) &= [\mathrm{k}^{2}_{2,1}, \mathrm{k}^{2}_{2,2}, \mathrm{k}^{2}_{2,3}, \mathrm{k}^{2}_{2,4}, \mathrm{k}^{2}_{2,5}, \mathrm{k}^{2}_{2,6}, \mathrm{k}^{2}_{2,7}, \mathrm{k}^{2}_{2,8}]^{\mathrm{T}} = \\ &= [\mathrm{s}^{2}_{3,3}\mathrm{s}^{2}_{2,1}\mathrm{c}^{2}_{1,1}, \mathrm{s}^{2}_{3,3}\mathrm{s}^{2}_{2,1}\mathrm{s}^{2}_{1,1}, -\mathrm{s}^{2}_{3,3}\mathrm{c}^{2}_{2,1}\mathrm{c}^{2}_{1,2}, -\mathrm{s}^{2}_{3,3}\mathrm{c}^{2}_{2,1}\mathrm{s}^{2}_{1,2}, -\mathrm{c}^{2}_{3,3}\mathrm{s}^{2}_{2,3}\mathrm{c}^{2}_{1,3}, -\mathrm{c}^{2}_{3,3}\mathrm{s}^{2}_{2,3}\mathrm{s}^{2}_{1,3}, \mathrm{c}^{2}_{3,3}\mathrm{c}^{2}_{2,3}\mathrm{c}^{2}_{1,4}, \mathrm{c}^{2}_{3,3}\mathrm{c}^{2}_{2,3}\mathrm{s}^{2}_{1,4}]^{\mathrm{T}};\end{aligned} \tag{27}$$

$$\begin{aligned}\mathbf{k}_6(3) &= [\mathrm{k}^{3}_{6,1}, \mathrm{k}^{3}_{6,2}, \mathrm{k}^{3}_{6,3}, \mathrm{k}^{3}_{6,4}, \mathrm{k}^{3}_{6,5}, \mathrm{k}^{3}_{6,6}, \mathrm{k}^{3}_{6,7}, \mathrm{k}^{3}_{6,8}]^{\mathrm{T}} = \\ &= [\mathrm{s}^{3}_{3,2}\mathrm{c}^{3}_{2,2}\mathrm{s}^{3}_{1,1}, -\mathrm{s}^{3}_{3,2}\mathrm{c}^{3}_{2,2}\mathrm{c}^{3}_{1,1}, \mathrm{s}^{3}_{3,2}\mathrm{s}^{3}_{2,2}\mathrm{s}^{3}_{1,2}, -\mathrm{s}^{3}_{3,2}\mathrm{s}^{3}_{2,2}\mathrm{c}^{3}_{1,2}, -\mathrm{c}^{3}_{3,2}\mathrm{c}^{3}_{2,4}\mathrm{s}^{3}_{1,3}, \mathrm{c}^{3}_{3,2}\mathrm{c}^{3}_{2,4}\mathrm{c}^{3}_{1,3}, -\mathrm{c}^{3}_{3,2}\mathrm{s}^{3}_{2,4}\mathrm{s}^{3}_{1,4}, \mathrm{c}^{3}_{3,2}\mathrm{s}^{3}_{2,4}\mathrm{c}^{3}_{1,4}]^{\mathrm{T}}.\end{aligned} \tag{28}$$

The vectors $\mathbf{k}_{\mathrm{u}}(\mathrm{t})$,$\mathbf{k}_{\mathrm{v}}(\mathrm{t})$, $\mathbf{k}_{\mathrm{l}}(\mathrm{t})$ are orthonormalized, i.e., for their components are valid the relations:

$$\sum_{\mathrm{m}=1}^{8}\mathrm{k}^{\mathrm{t}_1}_{\mathrm{u}_1,\mathrm{m}}.\mathrm{k}^{\mathrm{t}_2}_{\mathrm{u}_2,\mathrm{m}} = \begin{cases} 0 \text{ for } \mathrm{t}_1 \neq \mathrm{t}_2; \\ 1 \text{ for } \mathrm{t}_1 = \mathrm{t}_2, \end{cases} \quad \text{where } \mathrm{u}_1,\ \mathrm{u}_2 = 0,\ 1,\ \ldots,\ 7 \text{ and } \mathrm{t}_1,\ \mathrm{t}_2 = 1,\ 2,\ 3. \tag{29}$$

The decomposition 3D-HAKLT, shown in Figure 2, could be easily generalized for the cases when N = 16, 32, ... , 2^{n} through uniting the computational principles from Figures 1 and 2 for decompositions of high numbers of levels (n = 4, 5, ...). From Equation (29), it follows that the rows of the matrices $\mathbf{T}^{\mathrm{t}_1}_{\mathrm{FOHKLT}}(8)$ and $\mathbf{T}^{\mathrm{t}_2}_{\mathrm{FOHKLT}}(8)$ for $\mathrm{t}_1 \neq \mathrm{t}_2$ are not correlated. This conclusion could also be generalized for the rows of the matrices $\mathbf{T}^{t_1}_{\mathrm{FOHKLT}}(2^n)$ and $\mathbf{T}^{t_2}_{\mathrm{FOHKLT}}(2^n)$. Hence, after the execution of each stage of 3D-FOHKLT, the so-calculated N slices in the corresponding direction are highly decorrelated [12,15].

In the general case, the spectral coefficients s(u,v,l) are calculated through direct 3D-KLT, in accordance with the relation

$$s(u,v,l) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} x(i,j,k)\mathrm{kar}(i,u,N)\mathrm{kar}(j,v,N)\mathrm{kar}(k,l,N) \text{ for } u,\ v,\ l = 0,\ 1,\ \ldots,\ N-1. \quad (30)$$

The three-dimensional inverse KLT is defined by the relation

$$x(i,j,k) = \frac{1}{N^3}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\sum_{l=0}^{N-1} s(u,v,l)\mathrm{kar}(i,u,N)\mathrm{kar}(j,v,N)\mathrm{kar}(k,l,N) \text{ for } i,\ j,\ k = 0,\ 1,\ \ldots,\ N-1. \quad (31)$$

Here the 1D-Karhunen–Loeve (KL) functions of the Nth order are kar(i,u,N), kar(j,v,N), and kar(k,l,N).

Each third-order tensor $\boldsymbol{\mathcal{X}}$ of size N × N × N is represented as the weighted sum of N^3 3D-KL functions obtained by using the tensors $\boldsymbol{\mathcal{K}}_{u,v,l}$, each of size N × N × N:

$$\boldsymbol{\mathcal{X}} = (1/N^3)[s(0,0,0)\boldsymbol{\mathcal{K}}_{0,0,0} + s(1,0,0)\boldsymbol{\mathcal{K}}_{1,0,0} + \ldots + s(N-1,N-1,N-1)\boldsymbol{\mathcal{K}}_{N-1,N-1,N-1}] = (1/N^3)\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\sum_{l=0}^{N-1} s(u,v,l)\boldsymbol{\mathcal{K}}_{u,v,l}. \quad (32)$$

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

$$\boldsymbol{\mathcal{X}} = (1/N^3)\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\sum_{l=0}^{N-1} s(u,v,l)[k_{\mathbf{u}}(1) \circ k_v(2) \circ k_l(3)]. \quad (33)$$

Hence, in this case the tensor $\boldsymbol{\mathcal{X}}$ of size N × N × N is decomposed in a way similar to that of the Tucker decomposition [2];

$$\boldsymbol{\mathcal{X}} = \sum_{r_1=1}^{R}\sum_{r_2=1}^{R}\sum_{r_3=1}^{R} d_{r_1,r_2,r_3}(a_{r_1} \circ b_{r_2} \circ c_{r_3}). \quad (34)$$

Here, d_{r_1,r_2,r_3} are the eigen values of the tensor $\boldsymbol{\mathcal{X}}$, $a_{r_1}, b_{r_2}, c_{r_3}$ are the corresponding eigen vectors, and R is the tensor rank. The rank of the three-way tensor $\boldsymbol{\mathcal{X}}$ of size I × J × K is bounded from above in accordance with the inequality, as follows: $\max\{I, J, K\} \le R \le \min\{I \times J, I \times K, J \times K\}$ [3]. If I = J = K = N, then N ≤ R ≤ N^2. The difference between the decompositions from Equations (33) and (34) is the number of their components, which can reach up to N^3 and N^2, respectively (for the maximum value of the rank, R). In the case that the decomposition components number as defined by Equation (33) is not reduced, the 3D-HAKLT is reversible, i.e., the coefficients s(u,v,l) of the tensor $\boldsymbol{\mathcal{S}}$ and its basic tensors $\boldsymbol{\mathcal{K}}_{u,v,l}$ are sufficient to restore the tensor X.

4. Algorithm 3D-FOHKLT for Cubical Tensor Representation

The representation of the tensor $\boldsymbol{\mathcal{X}}$ of size N × N × N, through 3D-FOHKLT based on the spectral tensor $\boldsymbol{\mathcal{S}}$, is executed in correspondence with Figure 2. Each stage of the 3D transform is calculated for N = 8 through 1D-FOHKLT, as shown in Figure 1b.

In result is obtained the spectral tensor **S**, which comprises n layers of coefficients. These coefficients participate as component weights in the decomposition represented by Equation (33).

The tensor **X** is restored from the tensor *S* through inverse 3D-FOHKLT. This transform is based on the algorithm for direct 3D-FOHKLT, wherein is repeatedly performed the inverse 1D-AKLT (with basic operation "butterfly"), defined by the relations:

$$\left.\begin{array}{l} x_{1s}(p,q) = (1/2)[y_{1s}(p,q)\cos\theta_{p,q} + y_{2s}(p,q)\sin\theta_{p,q}]; \\ x_{2s}(p,q) = (1/2)[y_{1s}(p,q)\sin\theta_{p,q} - y_{2s}(p,q)\cos\theta_{p,q}] \end{array}\right. \text{for } s = 1,\ 2,\ \ldots,\ N^2. \tag{35}$$

Algorithm 1: Tensor representation based on the 3D-FOHKLT

```
Input: Third-order tensor $\mathcal{X}$ of size N × N × N ($N = 2^n$) and with elements x(i,j,k)
Output: Spectral tensor $\mathcal{S}$ of n layers, whose elements s(u,v,l) are the coefficients of the 3D-FOHKLT.
begin
  Divide the tensor $\mathcal{X}$ into horisontal slices and compose N/2 couples of matrices $\mathbf{X}^1_{p,q}$ and $\mathbf{X}^2_{p,q}$ of size N × N, for q = 1, 2, ... , N/2;
  for each couple of matrices $\mathbf{X}^1_{p,q}$ and $\mathbf{X}^2_{p,q}$ when q = 1, 2, ... , N/2 and p = 1, 2, ... , n, do
      Calculate the couple of matrices $\mathbf{Y}^1_{p,q}$ and $\mathbf{Y}^2_{p,q}$ obtained through 1D-FOHKLT in accordance with Equations (7) and (8);
      Calculate the matrices $\mathbf{P}_n(2^n)$ and $\mathbf{Y}_{n,k}$ for k = 0, 1, ... , N−1 and p = n in accordance with Equations (18)–(20);
      Define the matrices $\mathbf{E}_r$ for r = 0, 1, ... , N−1 in the level p = n by rearranging the matrices $\mathbf{Y}_{n,k}$ for k = 0, 1, ... , N−1 on the basis of the vector transform $\mathbf{e}_s = \mathbf{P}_n(2^n)\mathbf{y}_{n,s}$, for s = 1, 2, ... , $N^2$;
      Reshape all rearranged matrices $\mathbf{E}_r$ into the corresponding tensor $\mathcal{E}$ of size N × N × N.
  end
  Divide the tensor $\mathcal{E}$ into lateral slices and compose N/2 couples of intermediate matrices $\mathbf{E}^1_{p,q}$ and $\mathbf{E}^2_{p,q}$ of size N × N, for q = 1, 2, ... , N/2;
  for each couple of matrices $\mathbf{E}^1_{p,q}$ and $\mathbf{E}^2_{p,q}$ when q = 1, 2, ... , N/2 and p = 1, 2, ... , n, do
      Calculate the couple of matrices $\mathbf{F}^1_{p,q}$ and $\mathbf{F}^2_{p,q}$ transformed through 1D-FOHKLT in accordance with Equations (7) and (8);
      Calculate the matrices $\mathbf{P}_n(2^n)$ and $\mathbf{F}_{n,k}$ for k = 1, 2, ... , N and p = n in accordance with Equations (18)–(20);
      Define the matrices $\mathbf{F}_r$ for r = 0, 1, ... , N−1 in the level p = n by rearranging the matrices $\mathbf{F}_{n,k}$ for k = 0, 1, ... , N−1 on the basis of the vector transform $\mathbf{F}_s = \mathbf{P}_n(2^n)\mathbf{f}_{n,s}$, for s = 1, 2, ... , $N^2$;
      Reshape all rearranged matrices $\mathbf{F}_r$ in level p = n into the corresponding tensor $\mathcal{F}$ of size N × N × N
  end
  Divide the tensor $\mathcal{F}$ into frontal slices and compose N/2 couples of matrices $\mathbf{F}^1_{p,q}$ and $\mathbf{F}^2_{p,q}$ of size N × N, for q = 1, 2, ... , N/2;
  for each couple of matrices $\mathbf{F}^1_{p,q}$ and $\mathbf{F}^2_{p,q}$ when q = 1, 2, ... , N/2 and p = 1, 2, ... , n, do
      Calculate the couple of matrices $\mathbf{S}^1_{p,q}$ and $\mathbf{S}^2_{p,q}$ transformed through 1D-FOHKLT in accordance with Equations (7) and (8);
      Calculate the matrices $\mathbf{P}_n(2^n)$ and $\mathbf{S}_{n,k}$ for k = 1, 2, ... , N and p = n in accordance with Equations (18)–(20);
      Define the matrices $\mathbf{S}_r$ for r = 0, 1, ... , N−1 in the level p = n by rearranging the matrices $\mathbf{S}_{n,k}$ for k = 0, 1, ... , N−1 on the basis of the vector transform $\mathbf{S}_s = \mathbf{P}_n(2^n)s_{n,s}$, for s = 1, 2, ... , $N^2$;
      Reshape all rearranged matrices $\mathbf{S}_r$ in the level p = n into the spectral tensor $\mathcal{S}$ of size N × N × N
  end
  Arrange the coefficients s(u,v,l) of the spectral cubical tensor $\mathcal{S}$ layer by layer in accordance with their spatial frequencies (u,v,l) - from the lowest (0,0,0), to the highest (N−1, N−1, N−1), for u + v + l = const.
end
```

Here, the angle $\theta_{p,q}$ is defined by Equation (2), on the basis of the covariance matrix elements $K(p,q) = \begin{bmatrix} k_1(p,q) & k_3(p,q) \\ k_3(p,q) & k_2(p,q) \end{bmatrix}$ for the couple of matrices $\mathbf{X}^1_{p,q}$ and $\mathbf{X}^2_{p,q}$ (in levels p = 1, 2, 3 for q = 1, 2, 3, 4).

Unlike 1D-FOHKLT, the decorrelation of the cubical tensor elements through 3D-FOHKLT is executed in three mutually orthogonal directions, which is its main advantage compared to 1D-FOHKLT.

The algorithm for 3D-HAHKLT is similar to Algorithm 1, given above. The only difference is that the reordering of the matrices in each consecutive level is executed in accordance with Equation (9).

5. Comparative Evaluation of the Computational Complexity of 3D-HAKLT and 3D-FOHKLT

The computational complexity is evaluated on the basis of the needed operations "addition" and "multiplication".

5.1. Computational Complexity of 3D-HAKLT for a Cubical Tensor of Size $N \times N \times N$ ($N = 2^n$)

- The number of additions $\mathrm{A_H(n)}$ and multiplications $\mathrm{M_H(n)}$ needed to execute the 1D-HAKLT for a couple of matrices of size $\mathrm{N \times N}$ in one decomposition level is

$$\mathrm{A_H(n) = 3(N^2-1) + 2N^2 + 4 = 5 \times 2^{2n} + 1;} \tag{36}$$

$$\mathrm{M_H(n) = 3N^2 + 4N^2 + 5 = 7 \times 2^{2n} + 5.} \tag{37}$$

- For $N/2$ couples of matrices in one stage of the n-level 3D-HAKLT, these numbers are

$$\mathrm{A_{1H}(n) = [2^{n-1}(5 \times 2^{2n} + 1) + (2^{2n} - 1)(2^n - 1)]n;} \tag{38}$$

$$\mathrm{M_{1H}(n) = [2^{n-1}(7 \times 2^{2n} + 5) + 2^{2n}(2^n - 1)]n.} \tag{39}$$

- Then, for the three-level 3D-HAKLT is obtained

$$\mathrm{A_{3H}(n) = 3 \times [2^{n-1}(5 \times 2^{2n} + 1) + (2^{2n} - 1)(2^n - 1)]n;} \tag{40}$$

$$\mathrm{M_{3H}(n) = 3 \times [2^{n-1}(7 \times 2^{2n} + 5) + 2^{2n}(2^n - 1)]n.} \tag{41}$$

Hence,

$$\mathrm{A_{3H}(n) \approx 3 \times [2.5 \times 2^{3n} + 2^{3n}]n = 10.5 \times 2^{3n}n;} \tag{42}$$

$$\mathrm{M_{3H}(n) \approx 3 \times [3.5 \times 2^{3n} + 2^{3n}]n = 13.5 \times 2^{3n}n.} \tag{43}$$

The computational complexity of 3D-HAKLT, evaluated on the basis of the needed number of operations $\mathrm{O_{3H}(n)}$, is defined by the relation

$$\mathrm{O_{3H}(n) = A_{3H}(n) + M_{3H}(n) \approx 24 \times 2^{3n}n.} \tag{44}$$

The normalized value of $\mathrm{O_{3H}(n)}$ for one tensor element (voxel) is defined by the relation

$$\mathrm{O^0_{3H}(n) \approx (1/2^{3n})24 \times 2^{3n}n = 24n.} \tag{45}$$

For 1D-FOHKLT $\mathrm{A^{FO}_{1H}(n) = 2^{n-1}(5 \times 2^{2n} + 1)n \approx 2.5 \times 2^{3n}n}$ and $\mathrm{M^{FO}_{1H}(n) = 2^{n-1}(7 \times 2^{2n} + 5)n \approx 3.5 \times 2^{3n}n}$. Then

$$\mathrm{O^{FO}_{3H}(n) = A^{FO}_{3H}(n) + M^{FO}_{3H}(n) \approx 18 \times 2^{3n}n.} \tag{46}$$

The comparison of Equations (44) and (46) shows that the CC of 3D-FOHKLT is reduced approximately (4/3) times, compared to that of 3D-HAKLT.

5.2. Comparison of the Computational Complexity of 3D-HAKLT and 3D-FOHKLT to H-Tucker and TT

The CC of 3D-HAKLT for the transformation of the cubical tensor **X** of size 2^n, evaluated through the needed number of operations, is $O_{3H}(24 \times 2^{3n}n)$. In accordance with [5], the CC of the H-Tucker decomposition is $O_{HT}((d-1)R^3 + d \times 2^n \times R)$, where R is the rank, 2^n is the size, and d = 3 is the order of the tensor **X**. For $R = 2^n$ (the minimum value in the range $2^n \leq R \leq 2^{2n}$), the number of the operations is $O_{HT}(2^{4n+1} + 3 \times 2^{3n}) = O_{HT}(2^{3n}(2^{n+1}+3))$. In this case, the relative CCs of 3D-HAKLT and 3D-FOHKLT compared to that of the H-Tucker is defined by the relations:

$$\psi_1(n) = O_{HT}(n)/O_{3H}(n) = 2^{3n}(2^{n+1}+3)/(24 \times 2^{3n}n) = (2^n + 1.5\big)/12n; \tag{47}$$

$$\psi_2(n) = O_{HT}(n)/O_{3H}^{FO}(n) = 2^{3n}(2^{n+1}+3)/(18 \times 2^{3n}n) = (2^n + 1.5)/9n. \tag{48}$$

The CC of the TT decomposition [4] for the cubical tensor $\boldsymbol{\mathcal{X}}$ of size 2^n is $O_{TT}(2^n dR^3)$, i.e., $O_{TT}(3 \times 2^{4n})$. Correspondingly, the relative CC of 3D-HAKLT and 3D-FOHKLT towards the TT decomposition is defined by the relations:

$$\psi_3(n) = O_{TT}(n)/O_{3H}(n) = (3 \times 2^{4n})/(24 \times 2^{3n}n) = 2^n/8n; \tag{49}$$

$$\psi_4(n) = O_{TT}(n)/O_{3H}^{FO}(n) = (3 \times 2^{4n})/(18 \times 2^{3n}n) = 2^n/6n. \tag{50}$$

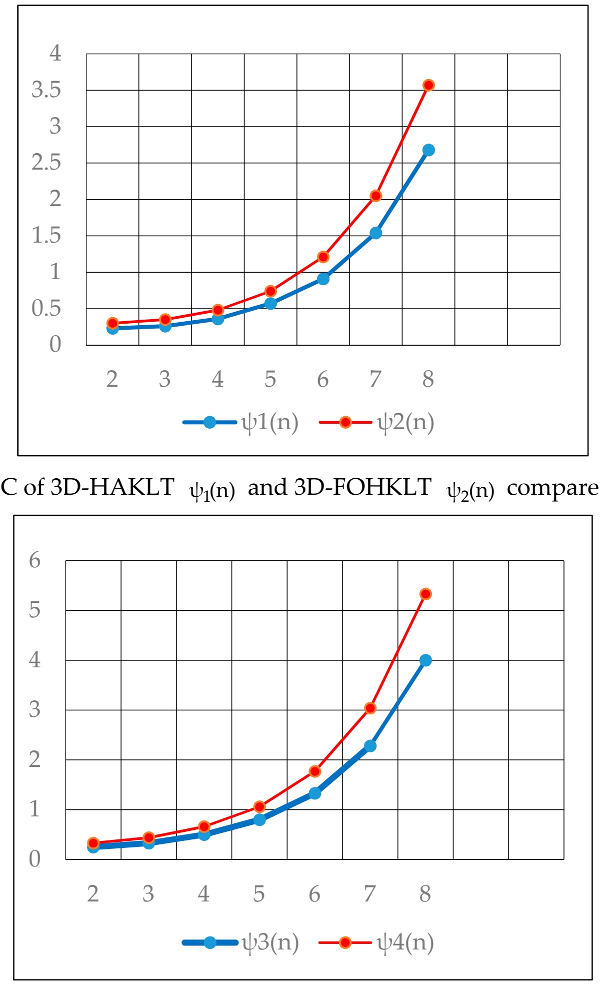
The values of functions $\psi_i(n)$ for i = 1, 2, 3, 4 and n = 2, 3, ... , 8 are given in Table 1, and their graphics are shown in Figure 3a,b. From Equations (47)–(50), it follows that the functions $\psi_i(n)$ for i = 1, 2, 3, 4 grow proportionally to $2^n/n$.

Table 1. Relative CC of 3D-HAKLT and 3D-FOHKLT compared to H-Tucker and TT.

n	2	3	4	5	6	7	8
$\psi_1(n)$	0.23	0.26	0.36	0.57	0.91	1.54	2.68
$\psi_2(n)$	0.30	0.35	0.48	0.74	1.21	2.05	3.57
$\psi_3(n)$	0.25	0.33	0.50	0.80	1.33	2.28	4.00
$\psi_4(n)$	0.33	0.44	0.66	1.06	1.77	3.04	5.33

The obtained results show that for small values of n (in the range from 2 to 5), the decompositions 3D-HAKLT and 3D-FOHKLT have low efficiencies because their CCs are equal or higher than those of H-Tucker and TT.

However, for practical applications (image processing), these values are quite different (for example, n = 8 corresponds to tensor image of size 256 × 256 × 256, which is very small), and in this case the relative CC of the offered decompositions is much lower than those of H-Tucker and TT. An additional advantage is that unlike H-Tucker and TT, the proposed algorithms do not need iterative calculations.



(**a**) Relative CC of 3D-HAKLT $\psi_1(n)$ and 3D-FOHKLT $\psi_2(n)$ compared to H-Tucker

(**b**) Relative CC of 3D-HAKLT $\psi_3(n)$ and 3D-FOHKLT $\psi_4(n)$ compared to TT

Figure 3. Graphical representation of the relative computational complexity.

6. Conclusions

In this work are presented new cubical tensor decompositions based on the 3D orthogonal hierarchical transforms HAKLT and FOHKLT. The comparative analysis for the relative CC of the new decompositions compared to those of H-Tucker and TT shows their advantages for tensors of size 256 and more. From these results it follows that 3D-FOHKLT has minimum CC, and is suitable for real-time applications. One more advantage of the offered algorithms is that they do not need iterative calculations, and instead use hierarchical calculations in the three mutually orthogonal directions of the tensor. Unlike the well-known 3D decompositions based on the N-dimensional eigen vectors of the tensor, the kernel of the offered hierarchical transforms is the AKLT of size 2 × 2, used repeatedly in each level. The presented algorithms 3D-HAKLT and 3D-FOHKLT could also be generalized for tensors with three different dimensions $2^{n_1} \times 2^{n_2} \times 2^{n_3}$ for $n_1 \neq n_2 \neq n_3$. The choice of the offered hierarchical 3D decompositions is made depending on the requirements and limitations of their CC imposed by the corresponding application area.

The future investigations of the 3D-HAKLT and 3D-FOHKLT will be aimed at the evaluation of their characteristics compared to the famous tensor decompositions, in order to define the best settings and to outline the most efficient applications in the 3D compression, filtration, analysis, search and recognition of multidimensional visual information, deep learning, etc.

Author Contributions: Conceptualization, R.K. (Roumen Kountchev), R.M. and R.K. (Roumiana Kountcheva); methodology, formal analysis, R.K. (Roumiana Kountcheva); investigation, R.M.; writing—original draft preparation, R.K. (Roumen Kountchev); writing—review and editing, R.K. (Roumiana Kountcheva); visualization, R.K. (Roumen Kountchev) 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. One-Dimensional Frequency-Ordered Hierarchical KLT for N = 8

The description of 1D-FOHKLT given below, is for the case, when N = 8 (n = 3) in correspondence with Equation (16) and Figure 1b. The transform matrix $\mathbf{T}_{\text{FOHKLT}}(8)$ of size 8 × 8 is decomposed in accordance with the relation:

$$\mathbf{T}_{\text{FOHKLT}}(8) = \mathbf{P}_3(8)\mathbf{G}_3(8)\mathbf{G}_2(8)\mathbf{G}_1(8) \tag{A1}$$

In this case, the transform matrices $\mathbf{G}_1(8)$, $\mathbf{G}_2(8)$, $\mathbf{G}_3(8)$ are defined by Equations (17) and (18):

$$\mathbf{G}_1(8) = \bigoplus_{i=1}^{4}\begin{bmatrix} c_{1,(i-1)+1} & s_{1,(i-1)+1} \\ s_{1,(i-1)+1} & -c_{1,(i-1)+1} \end{bmatrix} = \begin{bmatrix} c_{1,1} & s_{1,1} & 0 & 0 & 0 & 0 & 0 & 0 \\ s_{1,1} & -c_{1,1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_{1,2} & s_{1,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & s_{1,2} & -c_{1,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{1,3} & s_{1,3} & 0 & 0 \\ 0 & 0 & 0 & 0 & s_{1,3} & -c_{1,3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_{1,4} & s_{1,4} \\ 0 & 0 & 0 & 0 & 0 & 0 & s_{1,4} & -c_{1,4} \end{bmatrix} \tag{A2}$$

$$\mathbf{G}_2(8) = \bigoplus_{i=1}^{2}\begin{bmatrix} \bigoplus_{j=1}^{2} c_{2,2(i-1)+j} & \bigoplus_{j=1}^{2} s_{2,2(i-1)+j} \\ \bigoplus_{j=1}^{2} s_{2,2(i-1)+j} & \bigoplus_{j=1}^{2} -c_{2,2(i-1)+j} \end{bmatrix} = \begin{bmatrix} c_{2,1} & 0 & s_{2,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & c_{2,2} & 0 & s_{2,2} & 0 & 0 & 0 & 0 \\ s_{2,1} & 0 & -c_{2,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & s_{2,2} & 0 & -c_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{2,3} & 0 & s_{2,3} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{2,4} & 0 & s_{2,4} \\ 0 & 0 & 0 & 0 & s_{2,3} & 0 & -c_{2,3} & 0 \\ 0 & 0 & 0 & 0 & 0 & s_{2,4} & 0 & -c_{2,4} \end{bmatrix} \tag{A3}$$

$$\mathbf{G}_3(8) = \begin{bmatrix} \bigoplus_{j=1}^{4} c_{3,j} & \bigoplus_{j=1}^{4} s_{3,j} \\ \bigoplus_{j=1}^{4} s_{3,j} & \bigoplus_{j=1}^{4} -c_{3,j} \end{bmatrix} = \begin{bmatrix} c_{3,1} & 0 & 0 & 0 & s_{3,1} & 0 & 0 & 0 \\ 0 & c_{3,2} & 0 & 0 & 0 & s_{3,2} & 0 & 0 \\ 0 & 0 & c_{3,3} & 0 & 0 & 0 & s_{3,3} & 0 \\ 0 & 0 & 0 & c_{3,4} & 0 & 0 & 0 & s_{3,4} \\ s_{3,1} & 0 & 0 & 0 & -c_{3,1} & 0 & 0 & 0 \\ 0 & s_{3,2} & 0 & 0 & 0 & -c_{3,2} & 0 & 0 \\ 0 & 0 & s_{3,3} & 0 & 0 & 0 & -c_{3,3} & 0 \\ 0 & 0 & 0 & s_{3,4} & 0 & 0 & 0 & -c_{3,4} \end{bmatrix}. \tag{A4}$$

Here the angles $\theta_{p,q}$ for p = 1, 2, 3, ... , n and q = 1, 2, ... 2^{n-1} are calculated by using the coefficients $k_1(p,q)$, $k_2(p,q)$, $k_3(p,q)$ of the covariance matrices $K(p,q)$ for the couples of matrices $\mathbf{X}^1_{p,q}$ and $\mathbf{X}^2_{p,q}$ in correspondence with Equation (8).

The permutation matrix $\mathbf{P}_3(8)$ in the level p = 3 is calculated on the basis of Equations (19) and (20). In this case, the values of r = 0, 1, 2, 3, 4, 5, 6, 7 are transformed into k = 0, 7, 3, 4, 1, 6, 2, 5 and the matrix $\mathbf{P}_3(8)$ is defined as follows:

$$\mathbf{P}_3(8) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad \text{(A5)}$$

Then, 1D-FOHKLT for N = 8 is executed in 3 consecutive stages, each of 3 levels. The matrix representation of the 1D-FOHKLT corresponds to the first stage of the 3D transform, shown on Figure 2. The structure of matrices (A2)–(A4) is close to diagonal and for this reason, in the general case 1D-FOHKLT does not ensure full decorrelation of the decomposition components. The matrices (A2)–(A5) are multiplied in correspondence with (A1), and in result is obtained:

$$\mathbf{T}_{\mathrm{FOHKLT}}(8) =$$
$$\begin{bmatrix} c_{3,1}c_{2,1}c_{1,1} & c_{3,1}c_{2,1}s_{1,1} & c_{3,1}s_{2,1}c_{1,2} & c_{3,1}s_{2,1}s_{1,2} & s_{3,1}c_{2,3}c_{1,3} & s_{3,1}c_{2,3}s_{1,3} & s_{3,1}s_{2,3}c_{1,4} & s_{3,1}s_{2,3}s_{1,4} \\ s_{3,1}c_{2,1}c_{1,1} & s_{3,1}c_{2,1}s_{1,1} & s_{3,1}s_{2,1}c_{1,2} & s_{3,1}s_{2,1}s_{1,2} & -c_{3,1}c_{2,3}c_{1,3} & -c_{3,1}c_{2,3}s_{1,3} & -c_{3,1}s_{2,3}c_{1,4} & -c_{3,1}s_{2,3}s_{1,4} \\ s_{3,3}s_{2,1}c_{1,1} & s_{3,3}s_{2,1}s_{1,1} & -s_{3,3}c_{2,1}c_{1,2} & -s_{3,3}c_{2,1}s_{1,2} & -c_{3,3}s_{2,3}c_{1,3} & -c_{3,3}s_{2,3}s_{1,3} & c_{3,3}c_{2,3}c_{1,4} & c_{3,3}c_{2,3}s_{1,4} \\ c_{3,3}s_{2,1}c_{1,1} & c_{3,3}s_{2,1}s_{1,1} & -c_{3,3}c_{2,1}c_{1,2} & -c_{3,3}c_{2,1}s_{1,2} & s_{3,3}s_{2,3}c_{1,3} & s_{3,3}s_{2,3}s_{1,3} & -s_{3,3}c_{2,3}c_{1,4} & -s_{3,3}c_{2,3}s_{1,4} \\ c_{3,4}s_{2,2}s_{1,1} & -c_{3,4}s_{2,2}c_{1,1} & -c_{3,4}c_{2,2}s_{1,2} & c_{3,4}c_{2,2}c_{1,2} & s_{3,4}s_{2,4}s_{1,3} & -s_{3,4}s_{2,4}c_{1,3} & -s_{3,4}c_{2,4}s_{1,4} & s_{3,4}c_{2,4}c_{1,4} \\ s_{3,4}s_{2,2}s_{1,1} & -s_{3,4}s_{2,2}c_{1,1} & -s_{3,4}c_{2,2}s_{1,2} & s_{3,4}c_{2,2}c_{1,2} & -c_{3,4}s_{2,4}s_{1,3} & c_{3,4}s_{2,4}c_{1,3} & c_{3,4}c_{2,4}s_{1,4} & -c_{3,4}c_{2,4}c_{1,4} \\ s_{3,2}c_{2,2}s_{1,1} & -s_{3,2}c_{2,2}c_{1,1} & s_{3,2}s_{2,2}s_{1,2} & -s_{3,2}s_{2,2}c_{1,2} & -c_{3,2}c_{2,4}s_{1,3} & c_{3,2}c_{2,4}c_{1,3} & -c_{3,2}s_{2,4}s_{1,4} & c_{3,2}s_{2,4}c_{1,4} \\ c_{3,2}c_{2,2}s_{1,1} & -c_{3,2}c_{2,2}c_{1,1} & c_{3,2}s_{2,2}s_{1,2} & -c_{3,2}s_{2,2}c_{1,2} & s_{3,2}c_{2,4}s_{1,3} & -s_{3,2}c_{2,4}c_{1,3} & s_{3,2}s_{2,4}s_{1,4} & -s_{3,2}s_{2,4}c_{1,4} \end{bmatrix}. \quad \text{(A6)}$$

From (A6) it follows that the modules of the vectors $\mathbf{k}_i$ for I = 0, 1, 2, . . . , 7, which represent the rows of the matrix $\mathbf{T}_{\mathrm{FOHKLT}}(8)$, are defined by the relation:

$$\|\mathbf{k}_i\| = \sqrt{k_{i,0}^2 + k_{i,1}^2 + k_{i,2}^2 + k_{i,3}^2 + k_{i,4}^2 + k_{i,5}^2 + k_{i,6}^2 + k_{i,7}^2} = 1 \quad \text{(A7)}$$

i.e., these vectors are normalized. For example, the module of the vector $\mathbf{k}_0$ (row i = 0 of the matrix $\mathbf{T}_{\mathrm{FOHKLT}}(8)$), is:

$$\begin{gathered} \|\mathbf{k}_0\|^2 = (c_{3,1}c_{2,1}c_{1,1})^2 + (c_{3,1}c_{2,1}s_{1,1})^2 + (c_{3,1}s_{2,1}c_{1,2})^2 + (c_{3,1}s_{2,1}s_{1,2})^2 + (s_{3,1}c_{2,3}c_{1,3})^2 + \\ (s_{3,1}c_{2,3}s_{1,3})^2 + (s_{3,1}s_{2,3}c_{1,4})^2 + (s_{3,1}s_{2,3}s_{1,4})^2 = (c_{3,1}c_{2,1})^2 + (c_{3,1}s_{2,1})^2 + (s_{3,1}c_{2,3})^2 + (s_{3,1}s_{2,3})^2 = (c_{3,1})^2 + (s_{3,1})^2 = 1. \end{gathered} \quad \text{(A8)}$$

The couples of vectors $\mathbf{k}_i$ and $\mathbf{k}_j$ (for i ≠ j and i, j = 0,1, . . . ,7) which represent different rows of the matrix $\mathbf{T}_{\mathrm{FOHKLT}}(8)$, are mutually orthogonal (non-correlated) only in some particular cases. For example, the scalar product of the vectors $\mathbf{k}_1$ and $\mathbf{k}_2$ is defined by the relation:

$$\begin{gathered} \mathbf{k}_1.\mathbf{k}_2 = s_{3,1}c_{2,1}c_{1,1}s_{3,3}s_{2,1}c_{1,1} + s_{3,1}c_{2,1}s_{1,1}s_{3,3}s_{2,1}s_{1,1} - s_{3,1}s_{2,1}c_{1,2}s_{3,3}c_{2,1}c_{1,2} - s_{3,1}s_{2,1}s_{1,2}s_{3,3}c_{2,1}s_{1,2} + \\ + c_{3,1}c_{2,3}c_{1,3}c_{3,3}s_{2,3}c_{1,3} + c_{3,1}c_{2,3}s_{1,3}c_{3,3}s_{2,3}s_{1,3} - c_{3,1}s_{2,3}c_{1,4}c_{3,3}c_{2,3}c_{1,4} - c_{3,1}s_{2,3}s_{1,4}c_{3,3}c_{2,3}s_{1,4} = \\ = s_{3,1}c_{2,1}s_{3,3}s_{2,1} - s_{3,1}s_{2,1}s_{3,3}c_{2,1} + c_{3,1}c_{2,3}c_{3,3}s_{2,3} - c_{3,1}s_{2,3}c_{3,3}c_{2,3} = s_{3,1}c_{2,1}s_{3,3}(s_{2,1} - c_{2,1}) + c_{3,1}c_{2,3}c_{3,3}(s_{2,3} - c_{2,3}). \end{gathered} \quad \text{(A9)}$$

From this, it follows that $\mathbf{k}_1.\mathbf{k}_2 = 0$ only if the conditions $s_{2,1} = c_{2,1}$ and $s_{2,3} = c_{2,3}$ are satisfied.

In particular, if for a given input tensor $\boldsymbol{X}$ in the three stages of 3D-FOHKLT the angles $\theta_{p.q}$ for p = 1, 2, 3 and q = 1, 2, 3, 4 are equal or close to $\pi/4$, then in correspondence with (A6) the matrix $\mathbf{T}^{\pi/4}_{\mathrm{FOHKLT}}(8)$ coincides with the frequency-ordered matrix of the Wash-Hadamard Transform $\mathbf{T}_{\mathrm{WHT}}(8)$ [14,16], i.e.,:

$$\mathbf{T}^{\pi/4}_{\mathrm{FOHKLT}}(8) = \mathbf{T}_{\mathrm{WHT}}(8) = \frac{1}{2\sqrt{2}}\left[\begin{array}{cccccccc} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \end{array}\right]\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array}. \tag{A10}$$

This property of the matrix $\mathbf{T}_{\mathrm{WHT}}(8)$ shows that when it is used, is got a frequency-ordered spectrum which corresponds to that obtained with the Fast Fourier Transform (FFT). Hence, the FOHKLT spectrum is frequency-ordered. In the cases when the angles $\theta_{p,q}$ are equal or close to 0 or $\pi/2$, the matrix $\mathbf{T}_{\mathrm{FOHKLT}}(8)$ is transformed into the following two matrices:

$$\mathbf{T}^{0}_{\mathrm{FOHKLT}}(8) = \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \tag{A11}$$

$$\mathbf{T}^{\pi/2}_{\mathrm{FOHKLT}}(8) = \frac{1}{2\sqrt{2}}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{A12}$$

These matrices are not transform matrices: $\mathbf{T}^{0}_{\mathrm{FOHKLT}}(8)$ is used to rearrange the components of the input vector and to change the signs in half of them, and the matrix $\mathbf{T}^{\pi/2}_{\mathrm{FOHKLT}}(8)$ only rearranges the input vector components.

The description given above, could be easily extended for $N = 2^n$ on the basis of Equations (10)–(20).

References

1. Ji, Y.; Wang, Q.; Li, X.; Liu, J. A Survey on Tensor Techniques and Applications in Machine Learning. *IEEE Access* **2019**, *7*, 162950–162990. [CrossRef]
2. Cichocki, A.; Mandic, D.; Phan, A.; 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]
3. Sidiropoulos, N.; De Lathauwer, L.; Fu, X.; Huang, K.; Papalexakis, E.; Faloutsos, C. Tensor Decomposition for Signal Processing and Machine Learning. *IEEE Trans. Signal Proc.* **2017**, *65*, 3551–3582. [CrossRef]
4. 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.
7. Vasilescu, M.; Kim, E. Compositional Hierarchical Tensor Factorization: Representing Hierarchical Intrinsic and Extrinsic Causal Factors. In Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD): Tensor Methods for Emerging Data Science Challenges, Anchorage, NY, USA, 4–8 August 2019; p. 9.
8. Wang, P.; Lu, C. Tensor Decomposition via Simultaneous Power Iteration. In Proceedings of the 34th Intern. Conference on Machine Learning (PMLR), Sydney, Australia, 6–9 August 2017; Volume 70, pp. 3665–3673.
9. Ishteva, M.; Absil, P.; Van Dooken, P. Jacoby Algorithm for the Best Low Multi-linear Rank Approximation of Symmetric Tensors. *SIAM J. Matrix Anal. Appl.* **2013**, *34*, 651–672. [CrossRef]
10. Zniyed, Y.; Boyer, R.; Almeida, A.; Favier, G. A TT-based Hierarchical Framework for Decomposing High-Order Tensors. *SIAM J. Sci. Comput.* **2020**, *42*, A822–A848. [CrossRef]
11. Rao, K.; Kim, D.; Hwang, J. *Fast Fourier Transform: Algorithms and Applications*; Springer: Dordrecht, The Netherlands, 2010; pp. 166–170.
12. Kountchev, R.; Kountcheva, R. Adaptive Hierarchical KL-based Transform: Algorithms and Applications. In *Computer Vision in Advanced Control Systems: Mathematical Theory*; Favorskaya, M., Jain, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; Volume 1, pp. 91–136.
13. Lu, H.; Kpalma, K.; Ronsin, J. Motion descriptors for micro-expression recognition. In *Signal Processing: Image Communication*; Elsevier: Amsterdam, The Netherlands, 2018; Volume 67, pp. 108–117.
14. Ahmed, N.; Rao, K. *Orthogonal Transforms for Digital Signal Processing*; Springer: Berlin/Heidelberg, Germany, 1975; pp. 86–91.
15. Kountchev, R.; Kountcheva, R. Decorrelation of Multispectral Images, Based on Hierarchical Adaptive PCA. *WSEAS Trans. Signal Proc.* **2013**, *9*, 120–137.
16. Kountchev, R.; Mironov, R.; Kountcheva, R. Hierarchical Cubical Tensor Decomposition through Low Complexity Orthogonal Transforms. *Symmetry* **2020**, *12*, 864. [CrossRef]