

Article

MLTSP: New 3D Framework, Based on the Multilayer Tensor Spectrum Pyramid

Roumiana A. Kountcheva ¹, Rumen P. Mironov ² and Roumen K. Kountchev ^{2,*}¹ TK Engineering, 1582 Sofia, Bulgaria² Department of Radio Communications and Video Technologies, Faculty of Telecommunications, Technical University of Sofia, 1000 Sofia, Bulgaria

* Correspondence: rkountch@tu-sofia.bg; Tel.: +359-9924284

Abstract: A tensor representation structure based on the multilayer tensor spectrum pyramid (MLTSP) is introduced in this work. The structure is “truncated”, i.e., part of the high-frequency spectrum coefficients is cut-off, and on the retained low-frequency coefficients, obtained at the output of each pyramid layer, a hierarchical tensor SVD (HTSVD) is applied. This ensures a high concentration of the input tensor energy into a small number of decomposition components of the tensors obtained at the coder output. The implementation of this idea is based on a symmetrical coder/decoder. An example structure for a cubical tensor of size $8 \times 8 \times 8$, which is represented as a two-layer tensor spectrum pyramid, where 3D frequency-ordered fast Walsh–Hadamard transform and HTSVD are used, is given in this paper. The analysis of the needed mathematical operations proved the low computational complexity of the new approach, due to a lack of iterative calculations. The high flexibility of the structure in respect to the number of pyramid layers, the kind of used orthogonal transforms, the number of retained spectrum coefficients, and HTSVD components, permits us to achieve the desired accuracy of the restored output tensor, imposed by the application. Furthermore, this paper presents one possible application for 3D object searches in a tensor database. In this case, to obtain the invariant representation of the 3D objects, in the spectrum pyramid, the 3D modified Mellin–Fourier transform is embedded, and the corresponding algorithm is shown.

Keywords: tensor decomposition; multilayer tensor spectrum pyramid; 3D orthogonal transform; hierarchical tensor SVD; 3D modified Mellin–Fourier transform



Citation: Kountcheva, R.A.; Mironov, R.P.; Kountchev, R.K. MLTSP: New 3D Framework, Based on the Multilayer Tensor Spectrum Pyramid. *Symmetry* **2022**, *14*, 1909. <https://doi.org/10.3390/sym14091909>

Academic Editor: Marco Montemurro

Received: 29 July 2022

Accepted: 7 September 2022

Published: 12 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Processing, storage, transfer, and analysis of multidimensional (MD) images, mathematically represented as MD tensors, recently became a question of particular focus studied in numerous research works. In the last years, various tensor decomposition methods have been developed [1–4], among which are the multilinear extensions of the matrix SVD, and the generalizations of the SVD matrix for higher-order tensors, called higher-order SVD (HOSVD). This includes several well-known methods: canonical polyadic decomposition (CPD) also known as parallel factor analysis (PARAFAC), where the tensor is represented as a sum of rank-one tensors; the hierarchical Tucker decomposition (H-Tucker), which is a higher-order form of the principal component analysis (PCA); the multilinear PCA (MPCA) [5]; the non-negative tensor factorization (NTF) [6]; the tensor-train decomposition (TTD) [7], etc. For the tensor decomposition components calculation, different iterative methods are used: the alternating least square (ALS) [8]; the tensor power iteration; the higher-order eigenvalue decomposition; the Jacobi algorithm; the QR factorization [9], etc.

Additionally, for object detection in tensor multispectral images, mathematical models of stochastic random fields for the background description are suggested [10]. In this work, the existing contemporary approaches for object detection is based on convolutional neural network (CNN) deep learning with structures of the kind region-based CNN (R-CNN),

fast R-CNN, YOLO (You Only Live Once), EfficientDet (efficient object detection), SSD (single-shot detector), etc., which are compared with the introduced new approach, and it is indicated that neural networks ensure high efficiency of object detection, but unlike the approach based on the stochastic random fields models, they demand significant computational resources and a huge amount of priori visual information about the objects.

Tensors play, nowadays, the leading part in modern deep learning architectures. As is known, deep neural networks sometimes perform in an extremely complicated regime, manipulating tens of millions of parameters—much larger than the training data amount. Unlike this, the use of tensor decompositions in deep learning architectures results in a significant reduction in the number of unknown parameters [11].

To achieve computational complexity reduction, a new non-iterative approach for MD tensor image representation based on the multilayer tensor spectrum pyramid (MLTSP) [12] with embedded 3D orthogonal transforms (3D OTs) and hierarchical tensor SVD (HTSVD) [13] is proposed in this work. This approach is illustrated by an example representation of a tensor of the size $8 \times 8 \times 8$ through two-layer tensor spectrum pyramid (2LTSP), with embedded 3D frequency-ordered fast Walsh–Hadamard transform (3D FO-FWHT) [14] and HTSVD for a tensor of the size $2 \times 2 \times 2$ (HTSVD $_{2 \times 2 \times 2}$) [13]. Furthermore, one application of the presented pyramidal structure aimed at the accelerated search of 3D objects represented through MLTSP and 3D modified Mellin–Fourier Transform (3D MMFT), which is an upgrade from the previous research of the authors [15], is given in this work.

In the next sections of the paper, more details are given about the new approach; i.e., Section 2 contains the basic relations, which represent the two-layer tensor spectrum pyramid implementation for a tensor of the size $8 \times 8 \times 8$; in Section 3, the HTSVD decomposition algorithm for a spectrum tensor of the size $2 \times 2 \times 2$ (which is the building element in the introduced pyramidal structure) is given; in Section 4, the fast algorithm for 3D FO-FWHT calculation is explained; Section 5 evaluates the computational complexity of 3D-FWHT; in Section 6, the general algorithm for 3D object search in a database of 3D objects represented through MLTSP is given; in Section 7, the invariant 3D object representation based on the 3D MMFT is given; and Section 8 contains the Conclusion.

2. Basic Relations, Which Represent the Two-Layer Tensor Spectrum Pyramid with 3D OT and HTSVD

To explain the structure of the multilayer TSP, we use, as an example, the 2LTSP, which comprises a coder and decoder. The structure of the decoder is mirror-symmetrical to that of the coder. The corresponding block diagrams are shown in Figure 1a,b.

The following symbols are used: \mathbf{X} —cubical tensor of size $N = 2^n$ for $N = 8$; $r = 0, 1$ —the 2LTSP layer number; \mathbf{S} —cubical spectrum tensor of size $N = 2^n$; $F(\cdot)$ —operator for direct 3D orthogonal transform (3D OT) (for example, one of the following: 3D FFT [16], 3D FO-AHKLT [17], 3D FO-FWHT [14], etc.); $F_T(\cdot)$ —operator for direct truncated orthogonal transform (3D TOT); $F^{-1}(\cdot)$ —operator for 3D inverse orthogonal transform (3D IOT); HTSVD $_{2 \times 2 \times 2}$ —hierarchical tensor singular value decomposition for a $2 \times 2 \times 2$ elementary tensor [13].

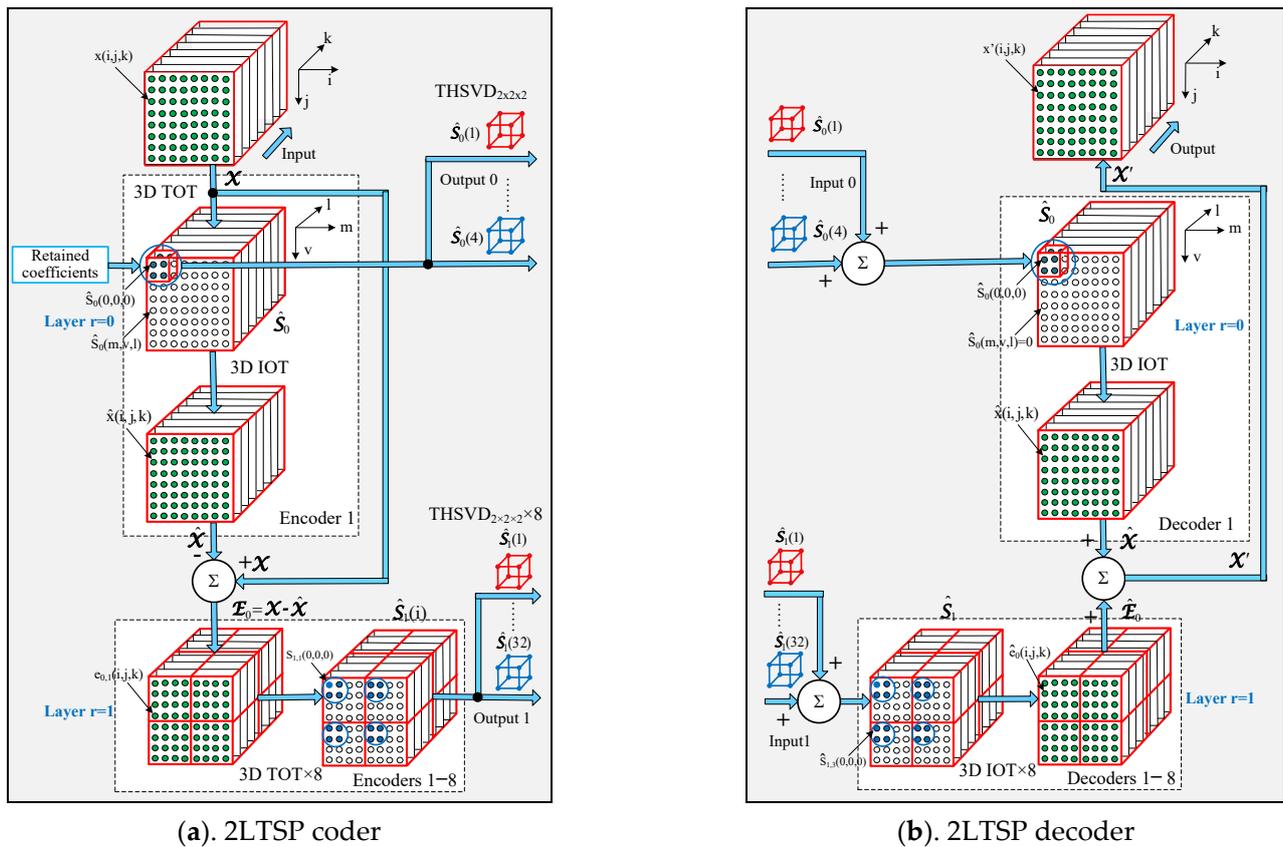


Figure 1. 2LTSP encoder, based on 3D OT and HTSVD $_{2 \times 2 \times 2}$ for $N = 8$.

2.1. Description of the 2LTSP Coder Performance

The performance of the coder (Figure 1a) is presented through the following operations:

In the first layer, for $r = 0$:

$$\hat{\mathbf{S}}_0 = F_T(\mathbf{X}) \quad (1)$$

$$\hat{\mathbf{S}}_{0,2 \times 2 \times 2} = \hat{\mathbf{S}}_{0,2 \times 2 \times 2}(1) + \hat{\mathbf{S}}_{0,2 \times 2 \times 2}(2) + \hat{\mathbf{S}}_{0,2 \times 2 \times 2}(3) + \hat{\mathbf{S}}_{0,2 \times 2 \times 2}(4) = \sum_{t=1}^4 \hat{\mathbf{S}}_{0,2 \times 2 \times 2}(t) \quad (2)$$

$$\hat{\mathbf{X}} = F^{-1}(\hat{\mathbf{S}}_0), \mathbf{E}_0 = \mathbf{X} - \hat{\mathbf{X}} \quad (3)$$

Here, $\hat{\mathbf{S}}_0$ denotes the spectrum tensor, which is the approximation of \mathbf{X} in the layer $r = 0$; \mathbf{E}_0 —the difference tensor in the layer $r = 0$; $\hat{\mathbf{S}}_{0,2 \times 2 \times 2}$ —the spectrum sub-tensor of the size $2 \times 2 \times 2$ in the layer $r = 0$, which comprises eight low-frequency spectrum coefficients of the selected 3D truncated orthogonal transform (3D TOT); $\hat{\mathbf{X}}$ —the approximation of tensor \mathbf{X} ; $\hat{\mathbf{S}}_{0,2 \times 2 \times 2}(t)$ —the SVD component t for the sub-tensor $\hat{\mathbf{S}}_{0,2 \times 2 \times 2}$.

In the second layer, for $r = 1$:

$$\hat{\mathbf{S}}_1(u) = F_T\{\mathbf{E}_0(u)\} \text{ for } u = 1, 2, \dots, 8; \quad (4)$$

$$\hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u) = \hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u, 1) + \hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u, 2) + \hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u, 3) + \hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u, 4) = \sum_{t=1}^4 \hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u, t), \quad (5)$$

for $u = 1, 2, \dots, 8$,

where $\mathbf{E}_0(u)$ denotes the u^{th} difference sub-tensor of the size $4 \times 4 \times 4$; $\hat{\mathbf{S}}_1(u)$ —the u^{th} spectrum sub-tensor of the size $4 \times 4 \times 4$, which is the approximation of the sub-tensor $\mathbf{S}_1(u)$ in the layer $r = 1$; $\hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u)$ —the u^{th} spectrum sub-tensor of the size $2 \times 2 \times 2$ in the layer $r = 1$, which comprises eight low-frequency spectrum coefficients of the selected 3D TOT; $\hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u, t)$ —the SVD component t of the sub-tensor $\hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u)$. At the coder

output in the layer $r = 0$ and in correspondence with Equation (2), four HTSVD $_{2 \times 2 \times 2}$ components are obtained, arranged following the decreasing values of their dispersions. At the coder output in the layer $r = 1$, 32 components of HTSVD $_{2 \times 2 \times 2}$ are obtained, calculated in correspondence with Equation (5) for the sub-tensors $u = 1, 2, \dots, 8$, and arranged following the decreasing values of their dispersions.

2.2. Description of the 2LTSP Decoder Performance

Because of the mirror symmetry existing in the coder/decoder structures (Figure 1), the performance of the decoder is presented through operations, inverse to these in the coder.

In the first layer, for $r = 0$:

$$\hat{\mathbf{S}}_{0,2 \times 2 \times 2} = \sum_{t=1}^4 \hat{\mathbf{S}}_{0,2 \times 2 \times 2}(t), \quad \hat{\mathbf{S}}_0 = \bigcup_{t=1}^4 \hat{\mathbf{S}}_{0,2 \times 2 \times 2}(t) \quad (6)$$

$$\hat{\mathbf{X}} = \mathbf{F}^{-1}(\hat{\mathbf{S}}_0) \quad (7)$$

In accordance with Equation (6), the tensor $\hat{\mathbf{S}}_0$ is restored by uniting the sub-tensors $\hat{\mathbf{S}}_{0,2 \times 2 \times 2}(t)$ in correspondence with Figure 1b, and the values of the remaining spectrum coefficients ($\hat{\mathbf{S}}_0$ voxels) are replaced by zeros.

In the second layer, for $r = 1$:

$$\hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u) = \sum_{t=1}^4 \hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u, t) \text{ for } u = 1, 2, \dots, 8; \quad (8)$$

$$\hat{\mathbf{S}}_1 = \bigcup_{u=1}^8 \bigcup_{t=1}^4 \hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u, t), \quad \hat{\mathbf{E}}_0 = \mathbf{F}^{-1}(\hat{\mathbf{S}}_1) \quad (9)$$

$$\mathbf{X}' = \hat{\mathbf{E}}_0 + \hat{\mathbf{X}}. \quad (10)$$

In accordance with Equation (9), the tensor $\hat{\mathbf{S}}_1$ is restored through uniting the sub-tensors $\hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u, t)$ in correspondence with Figure 1b, and the values of the remaining spectrum coefficients (voxels of $\hat{\mathbf{S}}_1$) are replaced by zeros. In cases that in the process of 2LTSP coder performance is used, 3D OT instead of 3D TOT, at the output of the decoder 2LTSP, the restored tensor \mathbf{X} is obtained, i.e., the transform of the input tensor \mathbf{X} through 2LTSP is reversible.

The difference between the original and restored tensors defines the error tensor, $\Delta\mathbf{X}$:

$$\Delta\mathbf{X} = \mathbf{X} - \mathbf{X}' = \mathbf{X} - (\hat{\mathbf{E}}_0 + \hat{\mathbf{X}}). \quad (11)$$

The restoration accuracy of the tensor \mathbf{X}' depends on:

- (1) The number of retained low-frequency spectrum coefficients, which compose the cubical spectrum tensors $\hat{\mathbf{S}}_{0,2 \times 2 \times 2}$ and $\hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u)$ for $u = 1, 2, \dots, 8$ (in the last case, each spectrum tensor comprises eight coefficients);
- (2) The number of retained components of each HTSVD $_{2 \times 2 \times 2}$, applied on the spectrum tensors $\hat{\mathbf{S}}_{0,2 \times 2 \times 2}$ and $\hat{\mathbf{S}}_{1,2 \times 2 \times 2}(u)$.

The restoration accuracy of tensor \mathbf{X}' increases together with the number of retained HTSVD spectrum coefficients, and the use of 3D FO-AHKLT as a basis for 2LTSP. The presented approach could be easily generalized for MTSP of more than two layers. The increase in the number of layers results in a lower power of the error tensor, $\Delta\mathbf{X}$.

3. HTSVD Algorithm for Spectrum Tensor Decomposition

In this case, on each tensor $\mathbf{S}_{2 \times 2 \times 2}$, obtained at the outputs 0 and 1 of the 2LTSP coder shown in Figure 1a, HTSVD $_{2 \times 2 \times 2}$ is applied. As a result, a sum of four components is obtained, for which the energy of each tensor is concentrated mainly in the first and second

components. This permits us to reduce (cut off) the number of low-energy components without lessening the quality of the restored tensor, in correspondence with Figure 1b.

The block diagram of the computational graph of the two-level HTSVD_{2×2×2} algorithm for decomposition of the elementary tensor **S**_{2×2×2} is shown in Figure 2. The decomposition is based on SVD for the matrix **X** of size 2 × 2, denoted as SVD_{2×2}, and described by the relation below [13]:

$$\mathbf{X} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \frac{1}{2A} \left\{ \sigma_1 \begin{bmatrix} \sqrt{rp} & \sqrt{sp} \\ \sqrt{rq} & \sqrt{sq} \end{bmatrix} + \sigma_2 \begin{bmatrix} \sqrt{sq} & -\sqrt{rq} \\ -\sqrt{sp} & \sqrt{rp} \end{bmatrix} \right\} = \sigma_1 \mathbf{U}_1 \mathbf{V}_1^t + \sigma_2 \mathbf{U}_2 \mathbf{V}_2^t = \mathbf{C}_1 + \mathbf{C}_2 \quad (12)$$

where *a*, *b*, *c*, and *d* are the elements of the matrix **X**, $A = \sqrt{v^2 + 4\eta^2} \neq 0$, $\sigma_1 = \sqrt{\frac{\omega+A}{2}}$, $\sigma_2 = \sqrt{\frac{\omega-A}{2}}$, $\mathbf{U}_1 = \frac{1}{\sqrt{2A}} \begin{bmatrix} \sqrt{p} \\ \sqrt{q} \end{bmatrix}$, $\mathbf{U}_2 = \frac{1}{\sqrt{2A}} \begin{bmatrix} -\sqrt{q} \\ \sqrt{p} \end{bmatrix}$, $\mathbf{V}_1 = \frac{1}{\sqrt{2A}} \begin{bmatrix} \sqrt{r} \\ \sqrt{s} \end{bmatrix}$, $\mathbf{V}_2 = \frac{1}{\sqrt{2A}} \begin{bmatrix} -\sqrt{s} \\ \sqrt{r} \end{bmatrix}$, $r = A + v$, $p = A + \mu$, $s = A - v$, $q = A - \mu$, $v = a^2 + c^2 - b^2 - d^2$, $\eta = ab + cd$, $\mu = a^2 + b^2 - c^2 - d^2$, $\omega = a^2 + b^2 + c^2 + d^2$. Since $\sigma_1 \gg \sigma_2$, the energy of the first decomposition component in (12) is much larger than that of the second. The number of parameters which define SVD_{2×2} is four (*v*, *η*, *μ*, *ω*), i.e., the decomposition is not “over-complete”.

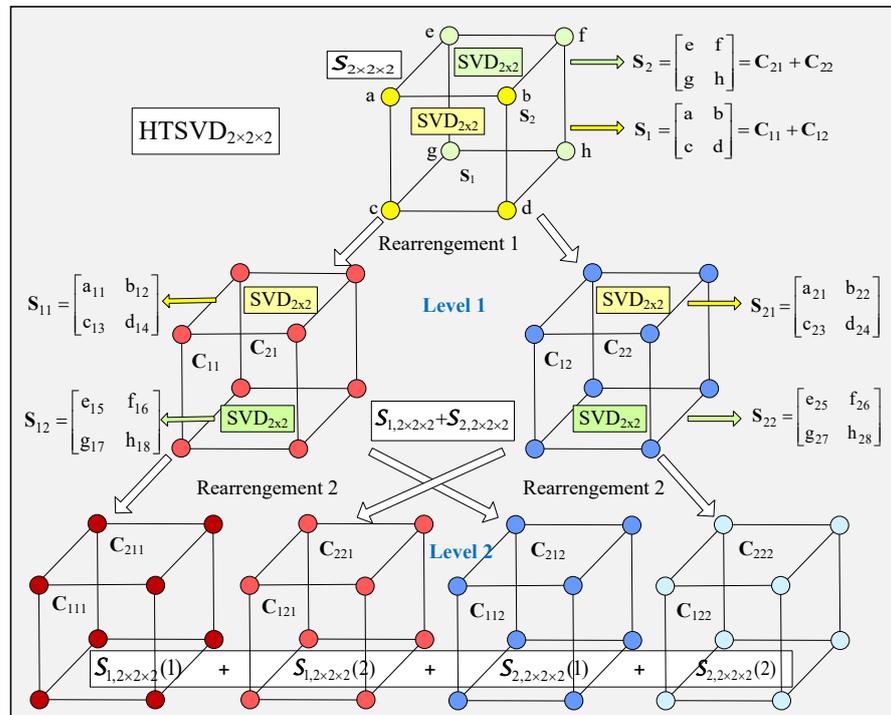


Figure 2. HTSVD_{2×2×2} algorithm for decomposition of tensor **S**_{2×2×2}.

After model unfolding (matricization) of the elementary tensor **S**_{2×2×2}, the following is obtained:

$$\text{unfold}(\mathbf{S}_{2 \times 2 \times 2}) = \begin{bmatrix} a & b & e & f \\ c & d & g & h \end{bmatrix} = [\mathbf{S}_1 \quad \mathbf{S}_2] \quad (13)$$

In the first level of the algorithm HTSVD for the tensor **S**_{2×2×2} (HTSVD_{2×2×2}), for each matrix **S**₁ and **S**₂, an SVD of the size 2 × 2 (SVD_{2×2}) is executed, and as a result we obtain:

$$\mathbf{S}_1 = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \mathbf{C}_{11} + \mathbf{C}_{12}, \mathbf{S}_2 = \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \mathbf{C}_{21} + \mathbf{C}_{22} \quad (14)$$

The matrices **C**_{*ij*} of the size 2 × 2 for *i*, *j* = 1, 2, calculated in accordance with Equation (12), are rearranged into new couples in correspondence with their singular values. After this, the first couple of matrices, **C**₁₁ and **C**₂₁, which have high singular

values, define the tensor $\mathbf{S}_{1(2 \times 2 \times 2)}$ by inverse matricization, and for the second couple, \mathbf{C}_{12} and \mathbf{C}_{22} , which have lower singular values, the corresponding tensor is $\mathbf{S}_{2(2 \times 2 \times 2)}$. Then:

$$\mathbf{S}_{2 \times 2 \times 2} = \mathbf{S}_{1(2 \times 2 \times 2)} + \mathbf{S}_{2(2 \times 2 \times 2)}. \quad (15)$$

After mode-3 unfolding of both tensors, the following is obtained:

$$\text{unfold}(\mathbf{S}_{1(2 \times 2 \times 2)}) + \text{unfold}(\mathbf{S}_{2(2 \times 2 \times 2)}) = [\mathbf{S}_{11} \quad \mathbf{S}_{21}] + [\mathbf{S}_{12} \quad \mathbf{S}_{22}]. \quad (16)$$

In the second HTSVD $_{2 \times 2 \times 2}$ level, SVD $_{2 \times 2}$ is applied for each matrix \mathbf{S}_{ij} of the size 2×2 , and this result is obtained:

$$\mathbf{S}_{11} = \mathbf{C}_{111} + \mathbf{C}_{112}, \mathbf{S}_{21} = \mathbf{C}_{211} + \mathbf{C}_{212}, \mathbf{S}_{12} = \mathbf{C}_{121} + \mathbf{C}_{122}, \mathbf{S}_{22} = \mathbf{C}_{221} + \mathbf{C}_{222}. \quad (17)$$

The matrices $\mathbf{C}_{i,j,k}$ of the size 2×2 for $i, j, k = 1, 2$ are rearranged into four new couples, following the decrease in their singular values. After inverse matricization, each of these four couples of matrices defines a corresponding tensor of size $2 \times 2 \times 2$.

In result of the mode1, unfolding is obtained:

$$\text{unfold}\{\mathbf{S}_{1(2 \times 2 \times 2)}(1)\} + \text{unfold}\{\mathbf{S}_{1(2 \times 2 \times 2)}(2)\} + \text{unfold}\{\mathbf{S}_{2(2 \times 2 \times 2)}(1)\} + \text{unfold}\{\mathbf{S}_{2(2 \times 2 \times 2)}(2)\} = [\mathbf{C}_{111}\mathbf{C}_{211}] + [\mathbf{C}_{121}\mathbf{C}_{221}] + [\mathbf{C}_{112}\mathbf{C}_{212}] + [\mathbf{C}_{122}\mathbf{C}_{222}]. \quad (18)$$

After execution of the two HTSVD $_{2 \times 2 \times 2}$ levels, the tensor $\mathbf{S}_{2 \times 2 \times 2}$ is represented as:

$$\mathbf{S}_{2 \times 2 \times 2} = \mathbf{S}_{1(2 \times 2 \times 2)}(1) + \mathbf{S}_{1(2 \times 2 \times 2)}(2) + \mathbf{S}_{2(2 \times 2 \times 2)}(1) + \mathbf{S}_{2(2 \times 2 \times 2)}(2) = \sum_{i=1}^2 \sum_{j=1}^2 \mathbf{S}_{j(2 \times 2 \times 2)}(i). \quad (19)$$

After the execution of the $\mathbf{S}_{2 \times 2 \times 2}$ decomposition, the tensors in the resulting sum are arranged following the decrease in the dispersion values for the sub-matrices obtained after the unfolding. The voxels of higher values in the decomposition (19) are colored in dark red in Figure 2, and those of lower values are in light blue. The decomposition of tensor $\mathbf{S}_{2 \times 2 \times 2}$ is a basic structure unit of HTSVD, when 3D tensors of size $N \times N \times N$ for $N = 2^n$ are decomposed [13].

4. Algorithm for Calculation of the 3D Frequency-Ordered Fast Walsh–Hadamard Transform

The embedding of the 3D frequency-ordered fast Walsh–Hadamard Transform (3D FO-FWHT) in the 2LTSP coder/decoder ensures maximum acceleration of the tensor \mathbf{X}' restoration. This approach is suitable for applications where high execution speed is requested, as, for example, the efficient compression of multidimensional images, etc.

The calculation of 3D FO-FWHT for tensors of the size $8 \times 8 \times 8$ could be executed in accordance with the diagrams, shown in Figure 3. The transform is separable and is executed through consecutive 1D FO-WHT calculations for the columns, rows, and tubes of the tensor \mathbf{X} of the size $N \times N \times N$, for $N = 2^n$. To speed up the 1D FO-WHT calculations, the famous “fast” algorithm is used [16]. The number of basic operations “addition” (A_R) needed for the execution of the “fast” 1D FO-WHT (1D FO-FWHT) in correspondence with the computational graph from Figure 3 is $A_R(n) = N \times \lg_2 N = 2^n n$. For the case when the “fast” truncated 1D FO-TWHT (i.e., 1D FO-FTWHT) is used with a reduction (truncation) in some of the output coefficients from N down to 2, the number of “additions” $A_T(n)$ is defined by the relation:

$$A_T(n) = N + N/2 + N/4 + \dots + N/2^{n-1} = N \sum_{p=0}^{n-1} 2^{-p} = 2(2^n - 1). \quad (20)$$

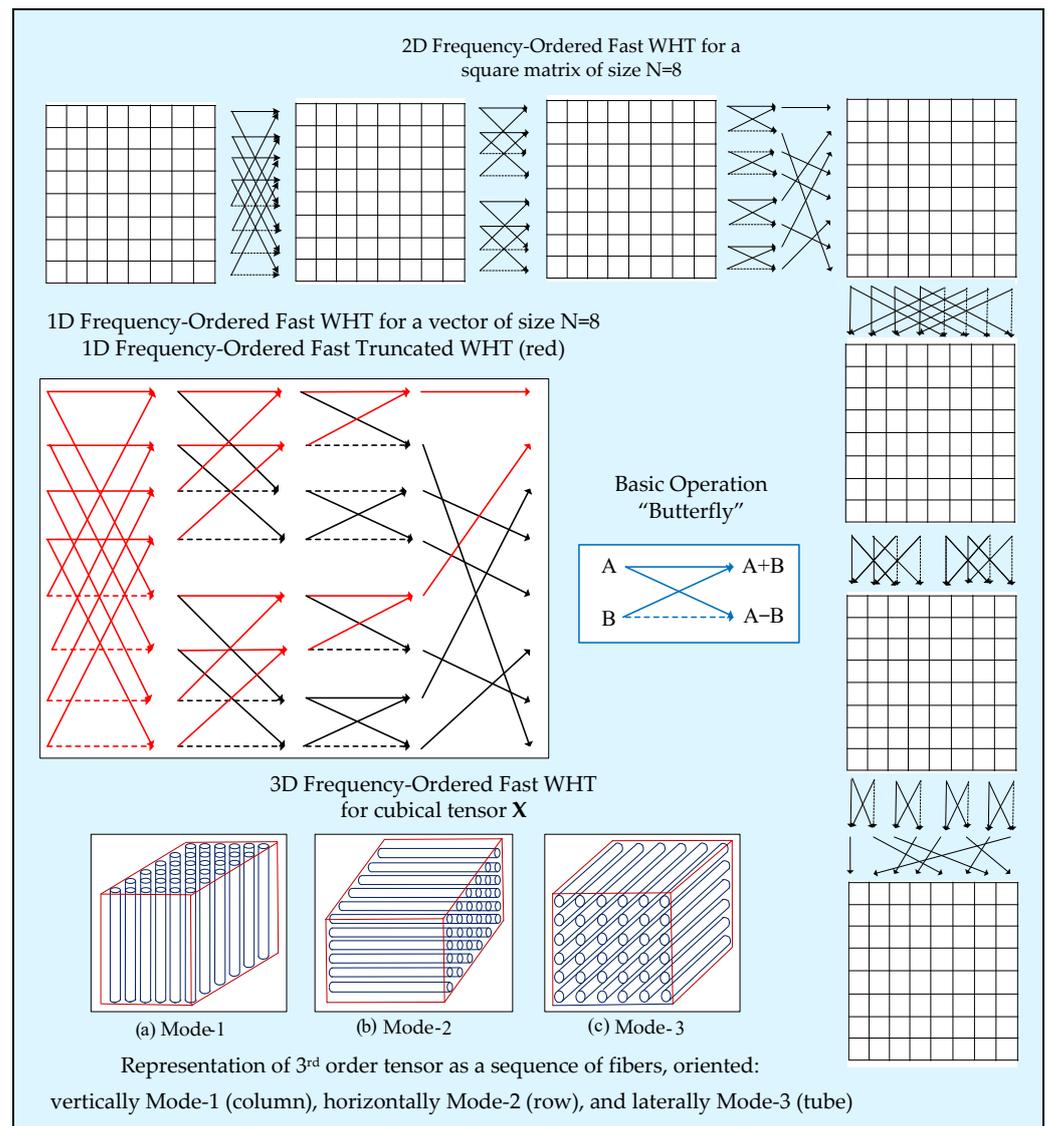


Figure 3. Block diagrams of algorithms 3D FO-FWHT and 3D FO-FTWHT.

The corresponding graph from Figure 3 is colored in red. In this case, the number of calculations needed for the 1D FO-FWHT compared to 1D FO-FTWHT is defined by the relation below:

$$\psi(n) = \frac{A_R(n)}{A_T(n)} = \frac{n}{2(1 - 2^{-n})} \tag{21}$$

Hence, for $n = 3$, $\psi(3) \approx 1.5$

In accordance with Figure 3, for the 3D WHT transform of the tensor X , it is first divided into N frontal slices (2D matrices). Then, for each column of the sequence of 2D matrices, 1D FWHT is executed, followed by a similar operation for each row of the corresponding transformed matrices. The calculated tensor X comprises all matrices transformed this way. Then, the tensor is divided again, but into N horizontal slices (2D matrices), which are transformed column-by-column through 1D FWHT. After that, on each row of the obtained 2D matrices, 1D TFWHT is applied again (for unfolding mode3 only), which, in correspondence with Equation (20), needs $2(N-1)$ additions only. Hence, the total number of additions needed to transform the tensor X of the size $N \times N \times N$ ($N = 2^n$) through 3D FO-FWHT is correspondingly:

$$A_{3D,T}(n) = N^2 [2N \lg_2 N + 2(N - 1)] = 2^{2n+1}(n2^n + 2^n - 1) \approx 2^{3n+1}(n + 1). \tag{22}$$

The total number of additions needed to transform the tensor \mathbf{X} of the size $N \times N \times N$ through 3D WHT is:

$$A_{3D} = 3N^3(N - 1) \approx 3 \times 2^{4n} \quad (23)$$

Hence, the acceleration of the calculations when the modified “fast” 3D FO-WHT (i.e., 3D FO-FWHT) is used, compared to the 3D FO-TFWHT, is:

$$\psi_{3D}(n) = \frac{A_{3D}(n)}{A_{3D,T}(n)} \approx \frac{3 \times 2^{4n}}{2 \times 2^{3n}(n + 1)} = 1.5 \frac{2^n}{n + 1}. \quad (24)$$

For the case when $n = 3$, from Equation (24) it follows that $\psi_{3D}(3) \approx 3$, i.e., the acceleration of the calculations based on the 3D FO-FWHT compared to 3D FO-TFWHT for a tensor \mathbf{X} of the size $8 \times 8 \times 8$, is 2.6 times.

5. Comparative Evaluation of 3D-FWHT Computational Complexity

Here, a comparative evaluation of the computational complexity (CC) of the 3D-FWHT is given with respect to the known hierarchical tensor decompositions 3D discrete wavelet transform (3D-DWT) and H-Tucker. For the evaluation, the number of needed basic operations is used. In the general case, the number of additions (A_{3H}) for the hierarchical 3D-FWHT [14] when $N = 2^n$ (n denotes the number of decomposition levels) is:

$$A_{3H}(n) = 3 \times 2^{3n}n. \quad (25)$$

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:

$$O_{3H}(n) = A_{3H}(n) + M_{3H}(n) = 3 \times 2^{3n}n. \quad (26)$$

The global number of additions and multiplications needed to transform the tensor of a size $N \times N \times N$ through n -level 3D-DWT with orthogonal filters (3, 5) is correspondingly [14]:

$$A_{3WT}(n) = 18N^3 \sum_{p=0}^{n-1} 8^{-p} = 20.5(2^{3n} - 1) \text{ and } M_{3WT}(n) = 24N^3 \sum_{p=0}^{n-1} 8^{-p} = 27.4(2^{3n} - 1). \quad (27)$$

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

$$O_{3WT}(n) = A_{3WT}(n) + M_{3WT}(n) \approx 48 \times 2^{3n}n. \quad (28)$$

The H-Tucker decomposition [2] of the cubical tensor of minimum rank $R = 2^n$, size $N = 2^n$, and order $d = 3$ requires $O_{HT}(n) = (3 \times 2^{3n} + 2 \times 2^{4n})$ operations. For the TT decomposition [7], the needed operations for the same tensor are $O_{TT}(n) = (3 \times 2^{4n})$, i.e., the CC is approximately 1.5 times higher than that of the H-Tucker. This is why the H-Tucker transform was selected for the CC comparison with the analyzed 3D deterministic orthogonal transforms.

The relative CC (denoted as $\psi(n)$) for the execution of the 3D FWHT compared to that of the 3D-DWT and H-Tucker is given below:

$$\psi_1(n) = O_{3WT}(n)/O_{3H}(n) = (48 \times 2^{3n})/(3 \times 2^{3n}n) \approx 16/n; \quad (29)$$

$$\psi_2(n) = O_{HT}(n)/O_{3H}(n) = 2^{3n}(2^{n+1} + 3)/(3 \times 2^{3n}n) \approx 0.66 \times (2^n/n); \quad (30)$$

The relative CC of the 3D FWHT decreases in inverse proportion to n towards the 3D DWT, while compared to H-Tucker, it grows proportionally to $2^n/n$. The CC of the 3D FWHT is lower when compared to that of the 3D DWT for levels $n < 16$, while compared to H-Tucker, it starts to decrease for levels $n \geq 4$. In this case, the value of the function $\psi_2(n) \geq 2.6$. The comparison results permit the choosing of the number of hierarchical

decomposition levels n , for which the algorithm 3D FWHT is more efficient than 3D DWT and H-Tucker.

6. Global Algorithm for 3D Object Search in a Database of 3D Objects Represented through MLTSP

We give here one possible application of the tensor representation based on the n -layer MLTSP, aimed at the accelerated 3D object search in a related database (DB). Such visual information is usually represented by image sequences obtained from various sources: video cameras, multispectral scanners, computer tomography, etc. The authentication degree of the 3D object search is increased significantly compared to the case when 2D image representation only is used. The main problems, which arise when such 3D image systems are created, are related to the high computational complexity of the operations needed for the full search in the DB and the huge number of data needed for the 3D object representation. To overcome these problems, we propose here the global algorithm for accelerated 3D object search, shown in Figure 4.

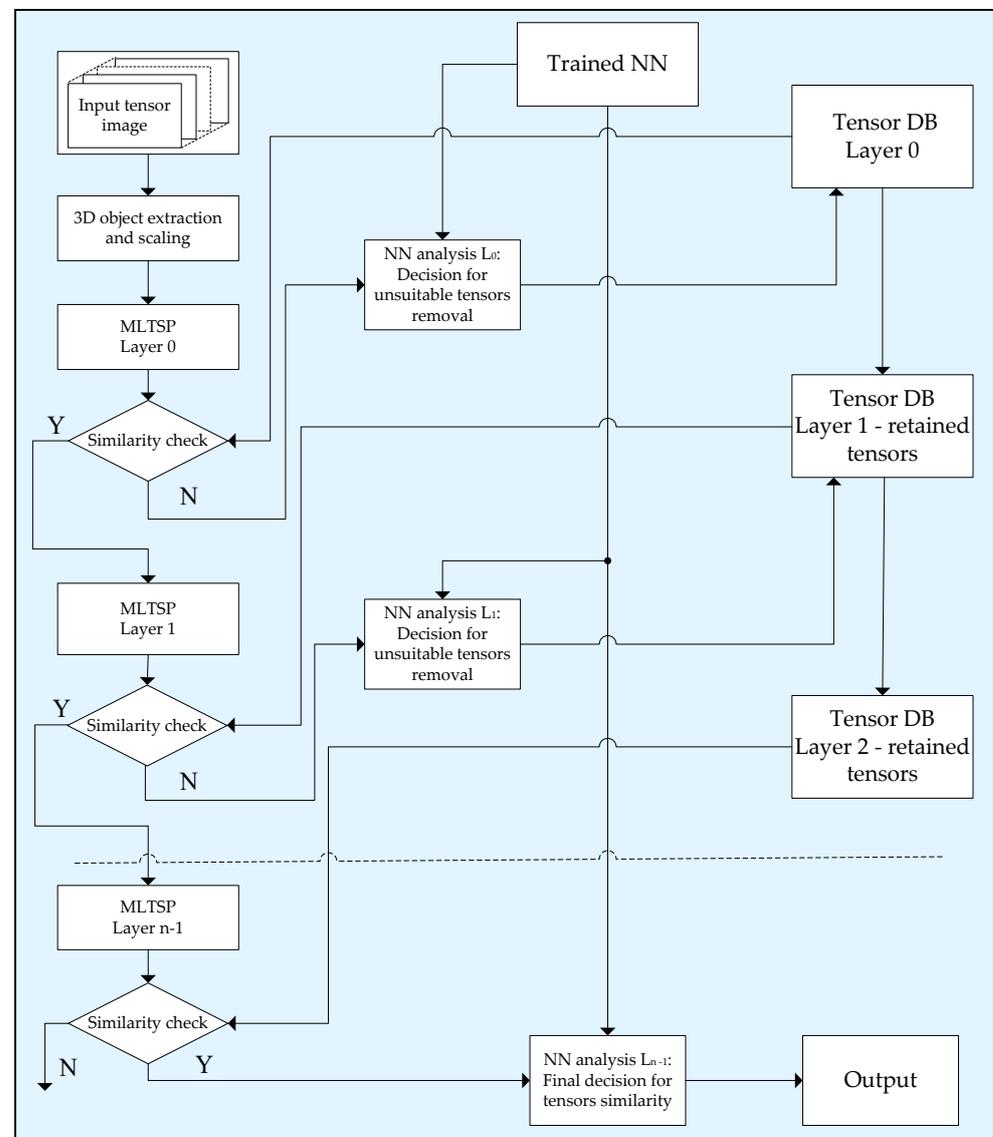


Figure 4. Global algorithm for fast 3D object search in a tensor database, based on MLTSP.

We suppose that the 3D object DB is already created and contains normalized tensor object representations, each of size $N \times N \times N$ (for $N = 2^n$), calculated through the n -layer MLTSP. For the query, the 3D object used is a deep neural network [18]. The segmented object image is scaled so as to match the size of the objects in the DB, and its tensor is represented through the n -layer MLTSP. The basic principle which we used to search for the unknown 3D object is to detect the closest similar objects in the DB through sequential multilayer selection. In the first MLTSP layer ($r = 0$), the search is full, and, as a result, a group of closest similar objects is selected. The selection is based on the similarity degree between homonymous HTSVD components of the spectrum tensors of the size $2 \times 2 \times 2$ in layers $r = 0$ of the corresponding MLTSPs. The small size of the compared THSVD components does not demand the use of significant computational resources for similarity evaluations. The search in the next MLTSP layer ($r = 1$) is also based on the similarity evaluation of the HTSVD components, but for the already selected group in the layer $r = 0$ only. As a result, the total number of computational operations needed for the similarity evaluation of the query tensor is reduced compared to the group from the previous layer. The search continues in the next MLTSP layers and stops when a 3D object is detected in the DB, whose similarity is higher than a predefined threshold. The thresholds used to select the groups of similar tensors in each layer of MLTSP are defined through training the corresponding neural network (NN).

The similarity degree for a couple of P -dimensional vectors $\mathbf{x} = [x_0, x_1, \dots, x_{P-1}]^T$ and $\mathbf{y} = [y_0, y_1, \dots, y_{P-1}]^T$ is calculated by using the squared cosine similarity (SCSim) criterion [15], defined by the relation:

$$\text{SCSim}(\mathbf{x}, \mathbf{y}) = \left[\sum_{i=0}^{P-1} \sum_{j=0}^{P-1} x(i)y(j) \right]^2 / \left[\sum_{i=0}^{P-1} x(i)^2 \right] \times \left[\sum_{j=0}^{P-1} y(j)^2 \right] \quad (31)$$

To use this criterion for similarity evaluation between a couple of tensors of the same size, first of all, unfolding of the same kind (mode 1, 2, or 3) for each tensor, \mathbf{X} and \mathbf{Y} , must be executed. In cases that the total number of vectors obtained after the unfolding is Q , for the similarity evaluation, the mean value of SCSim could be used for each couple of vectors \mathbf{x}_q and \mathbf{y}_q of the same sequential number $q = 1, 2, \dots, Q$, which belongs to \mathbf{X} and \mathbf{Y} , correspondingly:

$$\text{SCSim}(\mathbf{X}, \mathbf{Y}) = \frac{1}{Q} \sum_{q=1}^Q \left\{ \left[\sum_{i=0}^{P-1} \sum_{j=0}^{P-1} x_q(i)y_q(j) \right]^2 / \left[\sum_{i=0}^{P-1} x_q(i)^2 \right] \times \left[\sum_{j=0}^{P-1} y_q(j)^2 \right] \right\} \quad (32)$$

7. Invariant 3D Object Representation, Based on MLTSP with 3D Modified Mellin–Fourier Transform

The description of the 3D object should be invariant to 3D rotations (R) on angles $\alpha_r, \beta_r, \gamma_r$, 3D translation shift (T) in directions x_s, y_s, z_s , 3D scaling (S), and contrast (C) changes. To obtain the invariant representation of a 3D object through MLTSP, the spectrum coefficients at the outputs of the decomposition layers $r = 0, 1, \dots, n-1$ must be calculated by using the 3D modified Mellin–Fourier transform (3D MMFT), which is an upgrade from the previous work of the authors, introduced in [15].

The block diagram of the algorithm for calculation of the direct 3D MMFT spectrum coefficients is shown in Figure 5. The main operations for the 3D MMFT execution are:

- (1) Bi-polar transform of the voxels $x(i, l, k)$ of the input tensor image \mathbf{X} :

$$L(i, l, k) = x(i, l, k) - (x_{\max} + 1)/2 \text{ for } i, l, k = 0, 1, \dots, N - 1, \quad (33)$$

where x_{\max} is the maximum value in the voxel quantization scale.

- (2) First direct 3D discrete Fourier transform (3D DFT):

$$F(a, b, c) = \sum_{i=0}^{N-1} \sum_{l=0}^{N-1} \sum_{k=0}^{N-1} L(i, l, k) \exp\{-j(2\pi/N)(ia + lb + kc)\} \text{ for } a, b, c = 0, 1, \dots, N-1 \quad (34)$$

followed by centering of the Fourier coefficients $F(a, b, c)$:

$$F_0(a, b, c) = F\left(a - \frac{N}{2}, b - \frac{N}{2}, c - \frac{N}{2}\right) \text{ for } a, b, c = 0, 1, \dots, N-1 \quad (35)$$

(3) Retaining the centered low-frequency spectrum Fourier coefficients:

$$F_{0R}(a, b, c) = \begin{cases} F_0(a, b, c), & \text{if } (a, b, c) \in \text{retained region;} \\ 0 & \text{in all other cases.} \end{cases} \quad (36)$$

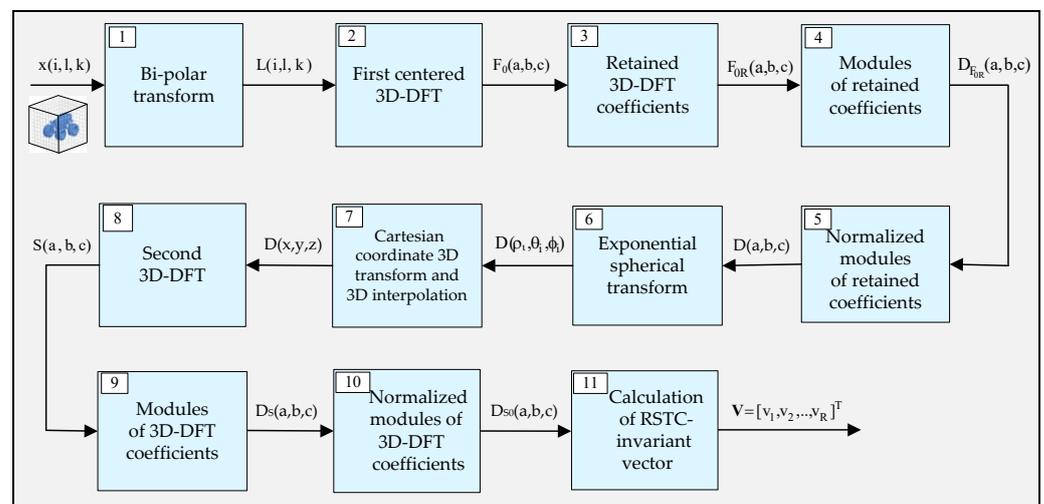


Figure 5. Block diagram of the algorithm for calculation of 3D MMFT coefficients for a cubical tensor of size $N \times N \times N$.

The retained coefficients' region is a cube with a side $H \leq N$, which envelopes the center $(0,0,0)$ of the 3D Fourier spectrum (H —even number). For $H < N$ and $a, b, c = -(H/2), -(H/2) + 1, \dots, -1, 0, 1, \dots, (H/2) - 1$; this cube contains low-frequency coefficients only.

(4) Calculation of modules and phases of the retained coefficients

$$F_{0R}(a, b, c) = D_{F_{0R}}(a, b, c) e^{j\varphi_{F_{0R}}(a, b, c)};$$

$$D_{F_{0R}}(a, b, c) = \sqrt{[A_{F_{0R}}(a, b, c)]^2 + [B_{F_{0R}}(a, b, c)]^2}, \quad (37)$$

$$\varphi_{F_{0R}}(a, b, c) = \arctan [B_{F_{0R}}(a, b, c) / A_{F_{0R}}(a, b, c)]$$

where $A_{0R}(a, b, c)$ and $B_{0R}(a, b, c)$ are the real and the imaginary components of $F_{0R}(a, b, c)$, correspondingly.

(5) Calculation of the retained coefficients' normalized modules:

$$D(a, b, c) = p \ln D_{F_{0R}}(a, b, c) \text{ for } p - \text{the normalization coefficient;} \quad (38)$$

(6) Replacement of the orthogonal 3D discretization grid, which contains the voxels $D(a, b, c)$, by a new grid, defined through 3D logarithmic spherical polar transform (3D LSPT), using the relations below:

$$\rho = \log \sqrt{a^2 + b^2 + c^2}, \quad \theta = \arccos [c / \sqrt{a^2 + b^2 + c^2}], \quad \phi = \arctan [b/a] \quad (39)$$

After discretization of ρ, θ, ϕ , the voxels $D(\rho_i, \theta_j, \phi_j)$ are obtained where:

$$\begin{aligned} \rho_i &= [(\sqrt{2}/2)H]^{i/H} \text{ for } i = 1, 2, \dots, H; \\ \theta_j &= (2\beta/H)j \text{ and } \phi_j = (2\beta/H)j \text{ for } j = (-H/2), \dots, -1, 0, 1, \dots, (H/2) - 1. \end{aligned} \quad (40)$$

Here, instead of the 3D LSPT given in Equation (32), we used the 3D exponential polar transform (3D EPT) from Equation (33), which gives a higher density of the retained voxels $D(a, b, c)$ around the origin $(0,0,0)$ of the discretization grid $(\rho_i, \theta_j, \phi_j)$.

- (7) Replacement of the discretization grid for voxels $D(\rho_i, \theta_j, \phi_j)$ defined through 3D EPT, by the orthogonal discretization 3D grid for voxels $D(x, y, z)$, calculated through trilinear interpolation (Figure 6). In this case, each interpolated voxel $D(x_1, y_1, z_1)$ is calculated taking into consideration the closest eight neighbor voxels on the grid:

$$D(x_1, y_1, z_1) = (1/H^3) \{z_1[x_1(y_1B + y_2F) + x_2(y_1E + y_2A)] + z_2[y_1(x_1G + x_2L) + y_2(x_1C + x_2K)]\} \quad (41)$$

where A, B, C, K, E, F, G, L—neighboring voxels from the discretization grid (x, y, z) ;

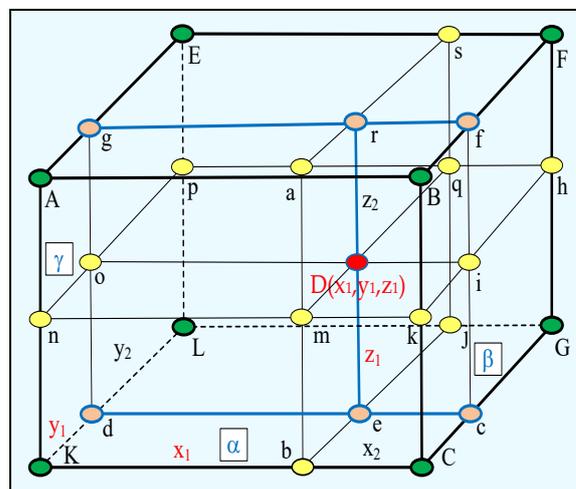


Figure 6. Calculation of the voxel $D(x_1, y_1, z_1)$ through trilinear interpolation on the basis of the closest eight neighbor voxels (colored in green) of the 3D discretization grid $(\rho_i, \theta_j, \phi_j)$.

α, β, γ —distances between the eight neighboring voxels in directions x, y, z ;

$D(x_1, y_1, z_1)$ —linearly interpolated voxel for $x_1 = \alpha - x_2, y_1 = \beta - y_2, z_1 = \gamma - z_2$;

The obtained results are the voxels $D(x, y, z)$ of the interpolated output tensor \mathbf{D} for $x, y, z = 0, 1, 2, \dots, H - 1$.

- (8) Second direct 3D DFT for a tensor \mathbf{D} , with voxels $D(x, y, z)$:

$$S(a, b, c) = (1/H^3) \sum_{x=0}^{H-1} \sum_{y=0}^{H-1} \sum_{z=0}^{H-1} D(x, y, z) \exp\{-j(2\beta/N)(xa + yb + zc)\} \text{ for } a, b, c = 0, 1, \dots, H - 1. \quad (42)$$

- (9) Calculation of the complex coefficient $S(a, b, c)$ modules:

$$D_S(a, b, c) = \sqrt{[A_S(a, b, c)]^2 + [B_S(a, b, c)]^2 + [C_S(a, b, c)]^2}, \quad (43)$$

where $A_S(a, b, c), B_S(a, b, c)$, and $C_S(a, b, c)$ are the real and the imaginary components of $S(a, b, c)$, correspondingly.

- (10) Calculation of the normalized modules $D_{S_0}(a, b, c)$ of the Fourier coefficients $S(a, b, c)$:

$$D_{S_0}(a, b, c) = x_{\max} \times [(D_S(a, b, c)/D_{S_{\max}}(a, b, c))], \quad (44)$$

where $D_{S_{\max}}(a, b, c)$ is the module of the coefficient $S(a, b, c)$ with a maximum value for $a, b, c = 0, 1, \dots, H - 1$.

- (11) Calculation of the vector for RSTC-invariant 3D object representation based on the coefficients $D_{S_0}(a, b, c)$ of highest energy in the amplitude spectrum 3D MMFT, for $a, b, c = 0, 1, \dots, H - 1$. The v_m components for $m = 1, 2, \dots, R$ of the corresponding RSTC-invariant vector $\mathbf{V} = [v_1, v_2, \dots, v_R]^T$ are defined by coefficients $D_{S_{OR}}(a, b, c)$, arranged as a 1D massif after scanning the 3D-MMFT spectrum in the frame of the 3D mask area with $R < H^3$ voxels $D_{S_0}(a, b, c)$.

The main advantages of 3D MMFT compared to other famous spectrum 3D object descriptors [19–22] are that it is an RSTC invariant, and the features' space dimensionality is reduced. This is why the approach based on MLTSP with embedded 3D MMFT accelerates the search of 3D objects in tensor DBs and ensures higher identification accuracy.

8. Conclusions

In this work, a new approach is proposed for 3D tensor representation through MLTSP with embedded 3D FO-FWHT and HTSVD, or 3D FO-FWHT and 3D-MMFT depending on the application, for example, information redundancy reduction in the input tensor, or an accelerated 3D object search in a tensor DB. In both cases, the main advantages of the MLTSP algorithms are the low computational complexity, the high flexibility regarding the choice of their parameters, and the ability for information redundancy reduction in the input tensor. These qualities have good potential for application in various areas where multidimensional signals and images are concerned.

Our future work will be focused on MLTSP modeling with various kinds of 3D OT and on the investigation of spectrum pyramid applications in different areas of tensor image processing, analysis, and computer vision. The future investigation of the MLTSP application in deep learning architectures is extremely promising.

Author Contributions: Conceptualization, R.A.K., R.P.M., and R.K.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”.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Kolda, T.; Bader, B. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500. [[CrossRef](#)]
2. Cichocki, A.; Lee, N.; Oseledets, I.; Phan, A.; Zhao, Q.; Mandic, D. Tensor networks for dimensionality reduction and large-scale optimization: Part 1. Low-rank tensor decompositions. *Found. Trends Mach. Learn.* **2016**, *9*, 249–429. [[CrossRef](#)]
3. Bergqvist, G.; Larsson, E. The Higher-Order Singular Value Decomposition: Theory and an Application. *IEEE Signal Processing Mag.* **2010**, *27*, 151–154. [[CrossRef](#)]
4. De Lathauwer, L.; De Moor, B.; Vandewalle, J. A Multilinear Singular Value Decomposition. *SIAM J. Matrix Anal. Appl.* **2000**, *24*, 1253–1278. [[CrossRef](#)]
5. Lu, H.; Plataniotis, K.; Venetsanopoulos, A. MPCA: Multilinear Principal Component Analysis of tensor objects. *IEEE Trans. Neural. Netw.* **2008**, *19*, 18–39. [[PubMed](#)]
6. Shashua, A.; Hazan, T. Non-negative tensor factorization with applications to statistics and computer vision. In Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 7–11 August 2005; pp. 792–799.
7. Oseledets, I. Tensor-Train Decomposition. *SIAM J. Sci. Comput.* **2011**, *33*, 2295–2317. [[CrossRef](#)]
8. Mohlenkamp, M. Musings on multilinear fitting. *Linear Algebra Its Appl.* **2013**, *438*, 834–852. [[CrossRef](#)]
9. Golub, H.; Loan, V.; Charles, F. *Matrix Computations*, 3rd ed.; Johns Hopkins University Press: Baltimore, MD, USA, 1996.

10. Andriyanov, N.; Dementiev, V.; Tashlinskii, A. Detection of objects in the images: From likelihood relationships towards scalable and efficient neural networks. *Comput. Opt.* **2022**, *46*, 139–159. [[CrossRef](#)]
11. Panagakis, Y.; Kossaifi, J.; Chrysos, G.; Oldfield, J.; Nicolaou, M.; Anandkumar, A.; Zafeiriou, S. Tensor Methods in Computer Vision and Deep Learning. *Proc. IEEE* **2021**, *109*, 863–890. [[CrossRef](#)]
12. Kountchev, R.; Kountcheva, R. Low Computational Complexity Third-Order Tensor Representation through Inverse Spectrum Pyramid. In *Advances in 3D Image and Graphics Representation, Analysis, Computing and Information Technology—Methods and Algorithms*; Kountchev, R., Patnaik, S., Shi, J., Favorskaya, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 1, pp. 63–76.
13. Kountchev, R.; Kountcheva, R. Comparative Analysis of the Hierarchical 3D-SVD and Reduced Inverse Tensor Pyramid in Regard to Famous 3D Orthogonal Transforms. In *New Approaches for Multidimensional Signal Processing—Proceedings of International Workshop NAMSP 2020*; Kountchev, R., Mironov, R., Shengqing, L., Eds.; Springer SIST series: Singapore, 2021; pp. 35–56.
14. Kountchev, R.; Mironov, R.; Kountcheva, R. Hierarchical Cubical Tensor Decomposition through Low Complexity Orthogonal Transforms. *Symmetry* **2020**, *12*, 864. [[CrossRef](#)]
15. Kountchev, R.; Kountcheva, R. Objects color segmentation and RSTC-invariant representation using adaptive color KLT and modified Mellin-Fourier transform. In Proceedings of the 8th IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA), Innsbruck, Austria, 16–18 February 2011; pp. 21–28.
16. Rao, K.; Kim, D.H.; Wang, J. *Fast Fourier Transform: Algorithms and Applications*; Springer: Dordrecht, The Netherlands, 2010.
17. Kountchev, R.; Mironov, R.; Kountcheva, R. Complexity Estimation of Cubical Tensor Represented through 3D Frequency-Ordered Hierarchical KLT. *Symmetry* **2020**, *12*, 1605. [[CrossRef](#)]
18. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza AKehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *arXiv* **2020**, arXiv:2001.05566v5. [[CrossRef](#)] [[PubMed](#)]
19. Tyagi, V. *Content-Based Image Retrieval: Ideas, Influences, and Current Trends*; Springer: Singapore, 2017.
20. Lee, C.; Shih, J.; Liu, Y.; Han, C. 3D-DFT Spectrum and Cepstrum of Dense Local Cubes for 3D Model Retrieval. In Proceedings of the Fifth International Conference on Informatics and Applications, Takamatsu, Japan, 14–16 November 2016; pp. 1–11.
21. Lee, C.; Shih, J.; Liu, Y. 3D model retrieval using 3D-DFT spectral and cepstral descriptors of local octants. In Proceedings of the 3rd IIAE International Conference on Intelligent Systems and Image Processing, Fukuoka, Japan, 2–5 September 2015; pp. 63–69.
22. Vranic, D.; Saupe, D.; Richter, J. Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics. In Proceedings of the IEEE Workshop Multimedia Signal Processing, Cannes, France, 3–5 October 2001; pp. 293–298.