

Article

Analysis of the Recursive Locally-Adaptive Filtration of 3D Tensor Images

Roumen Kountchev ¹, Rumen Mironov ^{1,*} and Roumiana Kountcheva ²

¹ Department of Radiocommunications and Videotechnology, Faculty of Telecommunications, Technical University of Sofia, 1000 Sofia, Bulgaria; rkountch@tu-sofia.bg

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

* Correspondence: rmironov@tu-sofia.bg; Tel.: +359-0895-590-555

Abstract: This work is focused on the computational complexity (CC) reduction of the locally-adaptive processing of 3D tensor images, based on recursive approaches. As a basis, a local averaging operation is used, implemented as a sliding mean 3D filter (SM3DF) with a central symmetric 3D kernel. Symmetry plays a very important role in constructing the working window. The presented theoretical approach could be adopted in various algorithms for locally-adaptive processing, such as additive noise reduction, sharpness enhancement, texture segmentation, etc. The basic characteristics of the recursive SM3DF are analyzed, together with the main features of the adaptive algorithms for filtration of Gaussian noises and unsharp masking where the filter is aimed at. In the paper, the ability of SM3DF implementation through recursive sliding mean 1D filters, sequentially bonded together, is also introduced. The computational complexity of the algorithms is evaluated for the recursive and non-recursive mode. The recursive SM3DF also suits the 3D convolutional neural networks which comprise sliding locally-adaptive 3D filtration in their layers. As a result of the lower CC, a promising opportunity is opened for higher efficiency of the 3D image processing through tensor neural networks.

Keywords: recursive sliding mean 3D filter; recursive locally-adaptive denoising 3D filter; recursive locally-adaptive unsharp masking 3D filter; computational complexity



Citation: Kountchev, R.; Mironov, R.; Kountcheva, R. Analysis of the Recursive Locally-Adaptive Filtration of 3D Tensor Images. *Symmetry* **2023**, *15*, 1493. <https://doi.org/10.3390/sym15081493>

Academic Editor: José Carlos R. Alcantud

Received: 25 May 2023

Revised: 24 July 2023

Accepted: 25 July 2023

Published: 27 July 2023



Copyright: © 2023 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

Three-dimensional tensors suit very well the representation of sequences of correlated images from various sources: CTI, MRI, endoscopic, US, multidimensional encephalographic, or electrocardiographic images, and also seismic signals, spatio-temporal video sequences, etc. One of the basic tools used for their processing are the algorithms for locally-adaptive 3D filtration [1–3]. The leading part in this processing belongs to the local averaging, implemented by using the sliding mean 3D filter (SM3DF) with a central symmetric 3D kernel. Symmetry plays a very important role in the working window construction. It is suitable for many algorithms for local processing of tensor images, such as for example, the adaptive reduction of additive noises [4,5], sharpness enhancement through unsharp masking, segmentation through variable thresholding [1], texture extraction through variation methods and wavelets [6], etc. Most of the famous algorithms for locally-adaptive processing are aimed at the 2D matrix images. Their efficiency is significantly increased when applied to image sequences of various kinds [3,7,8]. Recently, tensor neural network development [9] has been aimed at the decomposition of 3D convolutional neural networks (CNN) [10], where multiple locally-adaptive 3D tensor filtration is used. Various solutions already exist, aimed at multidimensional image filtering. In [11], a non-local filter is introduced, which combines the adaptive median filtering technique and the non-local means filtering, and is aimed at the processing of various kinds of medical images. In [12], the authors present one multiresolution version of the non-local means

filter, which adapts the denoising according to the spatial and frequency information in the image. In [13], a new non-local algorithm for image denoising is proposed, and references [14–17] are focused on the significance of image filtering in image processing. Most of these approaches present the obtained results, but the computational complexity is not evaluated in detail.

In [18,19], one approach for fast 2D mean filtering technique is presented. The authors propose a method which reduces the number of additions to a large extent and completely eliminates the need for any division in calculating the mean. The presented algorithms are aimed at the efficiency enhancement of the mathematical implementation in 2D processing, and could be generalized towards the 3D filtering.

In this work, we propose a new approach of higher efficiency in the processing of 3D images, which could be extended for calculation of multidimensional information. Due to the high volume of multidimensional data which have to be processed, the computational complexity (CC) of the operations executed by the locally-adaptive sliding mean filters increases significantly. One possible approach to settle the problem is to utilize the recursive SM3DF.

The main objectives of this work are to present an efficient approach for locally-adaptive processing of tensor 3D images based on the recursive SM3DF, to analyze its most significant characteristics, and to evaluate the CC as compared to that of the similar non-recursive algorithms. The next sections of the work are as follows: in Section 2 the characteristics of the recursive SM3DF in the spatial and frequency domains are analyzed; in Sections 3 and 4 we present the corresponding algorithms for the adaptive suppression of additive noises in tensor 3D images and for sharpness enhancement through adaptive unsharp masking, based on the recursive SM3DF; and Section 5 contains the discussion and conclusions and the ideas for future work.

2. Analysis of the Recursive Sliding Mean 3D Filter Characteristics

For the analysis of the recursive 3D filter, the non-recursive structure should be first analyzed. The non-recursive sliding mean 3D filter is presented below as a generalization of the well-known non-recursive sliding mean 2D filter.

The non-recursive SM3DF is analyzed in the z -area, and on the basis of this, the execution algorithm of the corresponding recursive filter and its implementation are defined, based on recursive sliding mean 1D filters, sequentially bonded together. The exact description of the sliding recursive 3D filter needs to understand its previous states, which define the corresponding causal area. The form of the area depends on the chosen way to scan the tensor voxels. It is supposed here that the filter scans the voxels following a trajectory which corresponds to the well-known progressive 3D linear scan (used in most applications for image processing), as shown on Figure 1, for a tensor of size $5 \times 5 \times 4$.

2.1. Non-Recursive Sliding Mean 3D Filter

The non-recursive SM3DF with a central symmetrizing 3D kernel $w(s, m, n)$ is designed for sliding processing of the voxels $x(i, r, k)$ of tensor \mathbf{X} , of size $M \times N \times K$. At the filter output, the mean value is obtained of the voxels framed by the 3D window, in correspondence with the relation:

$$\mu_x(i, r, k) = \frac{1}{A} \sum_{s=-a}^a \sum_{m=-b}^b \sum_{n=-h}^h w(s, m, n)x(i + s, r + m, k + n), \text{ for } w(s, m, n) = 1, \quad (1)$$

where $\mu_x(i, r, k)$ denotes the filtered output voxel, and $A = (2a + 1)(2b + 1)(2h + 1)$ is the number of voxels, framed by the 3D sliding window $W(i, r, k)$ of size $(2a + 1) \times (2b + 1) \times (2h + 1)$. For the filtration of voxels placed on the first and the last horizontal and lateral rows, and also on the first and the last vertical columns, the tensor \mathbf{X} should be extended in all three directions (zero padding), and as a result it becomes of size $(M + 2a) \times (N + 2b) \times (K + 2h)$.

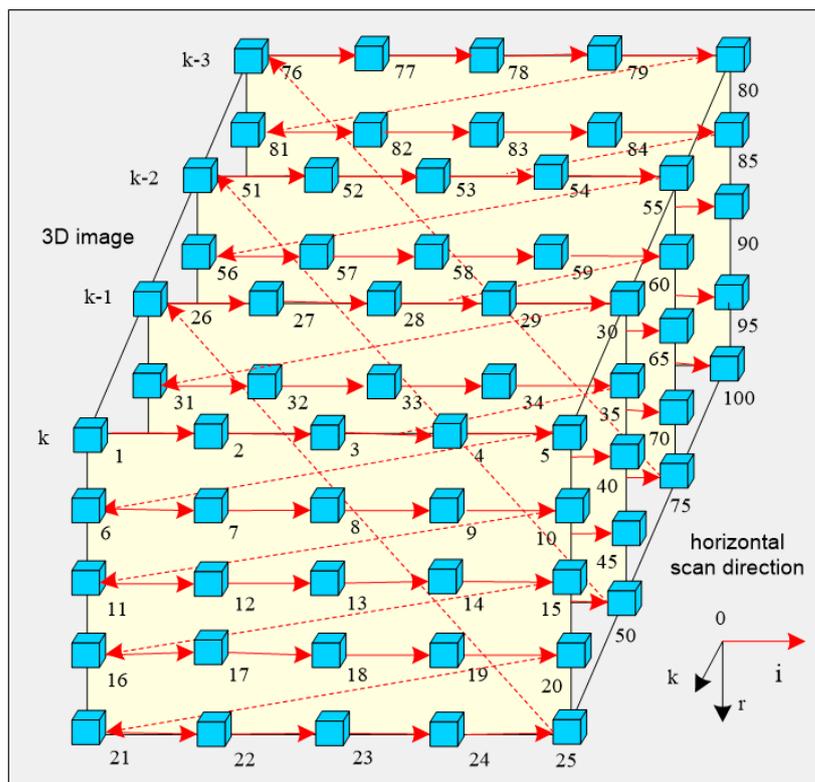


Figure 1. 3D progressive linear scan representation for the tensor X of size 5 × 5 × 4.

2.2. Recursive Sliding Mean 3D Filter

After applying the 3D z-transform [20] on Equation (1), the following is obtained:

$$M_x(z_1, z_2, z_3) = \frac{1}{A} \sum_{s=-a}^a \sum_{m=-b}^b \sum_{n=-h}^h X(z_1, z_2, z_3) z_1^s z_2^m z_3^n = \frac{1}{A} X(z_1, z_2, z_3) \sum_{s=-a}^a \sum_{m=-b}^b \sum_{n=-h}^h z_1^s z_2^m z_3^n. \tag{2}$$

In accordance with the condition for SM3DF kernel divisibility [7], from Equation (2) the following 3D transfer function is obtained:

$$H(z_1, z_2, z_3) = \frac{M_x(z_1, z_2, z_3)}{X(z_1, z_2, z_3)} = \left(\frac{1}{A_1} \sum_{s=-a}^a z_1^s \right) \left(\frac{1}{A_2} \sum_{m=-b}^b z_2^m \right) \left(\frac{1}{A_3} \sum_{n=-h}^h z_3^n \right) = H_1(z_1)H_2(z_2)H_3(z_3), \tag{3}$$

where $A_1 = (2a + 1)$, $A_2 = (2b + 1)$, $A_3 = (2h + 1)$ and $A = A_1A_2A_3$.

Here, $H_1(z_1), H_2(z_2), H_3(z_3)$ are the transfer functions of the corresponding 1D recursive filters, which could be represented in the following way:

$$\begin{aligned} H_1(z_1) &= \frac{1}{A_1} \sum_{s=-a}^a z_1^s = \frac{z_1^a - z_1^{-(a+1)}}{A_1(1 - z_1^{-1})}, \\ H_2(z_2) &= \frac{1}{A_2} \sum_{m=-b}^b z_2^m = \frac{z_2^b - z_2^{-(b+1)}}{A_2(1 - z_2^{-1})}, \\ H_3(z_3) &= \frac{1}{A_3} \sum_{n=-h}^h z_3^n = \frac{z_3^h - z_3^{-(h+1)}}{A_3(1 - z_3^{-1})}. \end{aligned} \tag{4}$$

To answer the requirement for the 1D recursive filter stability, the relations $|z_1| > 1$, $|z_2| > 1$ and $|z_3| > 1$ should be satisfied.

Then the 3D transfer function $H(z_1, z_2, z_3)$ from Equation (3) is reorganized as follows:

$$\begin{aligned}
 H(z_1, z_2, z_3) &= \frac{M_x(z_1, z_2, z_3)}{X(z_1, z_2, z_3)} = \left(\frac{z_1^a - z_1^{-(a+1)}}{A_1(1 - z_1^{-1})} \right) \left(\frac{z_2^b - z_2^{-(b+1)}}{A_2(1 - z_2^{-1})} \right) \left(\frac{z_3^h - z_3^{-(h+1)}}{A_3(1 - z_3^{-1})} \right) = \\
 &= \frac{1}{A} \left(\frac{z_1^a z_2^b z_3^h - z_1^a z_2^{-(b+1)} z_3^h - z_1^{-(a+1)} z_2^b z_3^h + z_1^{-(a+1)} z_2^{-(b+1)} z_3^h - z_1^a z_2^b z_3^{-(h+1)} + \dots}{1 - z_1^{-1} - z_2^{-1} - z_3^{-1} + z_1^{-1} z_2^{-1} + z_1^{-1} z_3^{-1} + z_2^{-1} z_3^{-1} - z_1^{-1} z_2^{-1} z_3^{-1}} + \dots \right) \quad (5)
 \end{aligned}$$

From Equation (5) it follows that:

$$\begin{aligned}
 M_x(z_1, z_2, z_3) &(1 - z_1^{-1} - z_2^{-1} - z_3^{-1} + z_1^{-1} z_2^{-1} + z_1^{-1} z_3^{-1} + z_2^{-1} z_3^{-1} - z_1^{-1} z_2^{-1} z_3^{-1}) = \\
 &= \frac{1}{A} X(z_1, z_2, z_3) [z_1^a z_2^b z_3^h - z_1^a z_2^{-(b+1)} z_3^h - z_1^{-(a+1)} z_2^b z_3^h + z_1^{-(a+1)} z_2^{-(b+1)} z_3^h - z_1^a z_2^b z_3^{-(h+1)} + \\
 &+ z_1^a z_2^{-(b+1)} z_3^{-(h+1)} + z_1^{-(a+1)} z_2^b z_3^{-(h+1)} - z_1^{-(a+1)} z_2^{-(b+1)} z_3^{-(h+1)}]. \quad (6)
 \end{aligned}$$

After applying the inverse 3D z-transform on Equation (6), at the output of SM3DF the mean local value, $\mu_x(i, r, k)$, is obtained:

$$\begin{aligned}
 \mu_x(i, r, k) &= \mu_x(i - 1, r, k) + \mu_x(i, r - 1, k) + \mu_x(i, r, k - 1) - \mu_x(i - 1, r - 1, k) - \mu_x(i - 1, r, k - 1) - \mu_x(i, r - 1, k - 1) + \\
 &+ \mu_x(i - 1, r - 1, k - 1) + \frac{1}{A} [x(i + a, r + b, k + h) - x(i + a, r - b - 1, k + h) - x(i - a - 1, r + b, k + h) \\
 &+ x(i - a - 1, r - b - 1, k + h) - x(i + a, r + b, k - h - 1) + x(i + a, r - b - 1, k - h - 1) + \\
 &+ x(i - a - 1, r + b, k - h - 1) - x(i - a - 1, r - b - 1, k - h - 1)] \quad (7)
 \end{aligned}$$

In Figure 2, the spatial positioning of the voxels, used for the recursive calculation of $\mu_x(i, r, k)$, in correspondence with Equation (7), is shown:

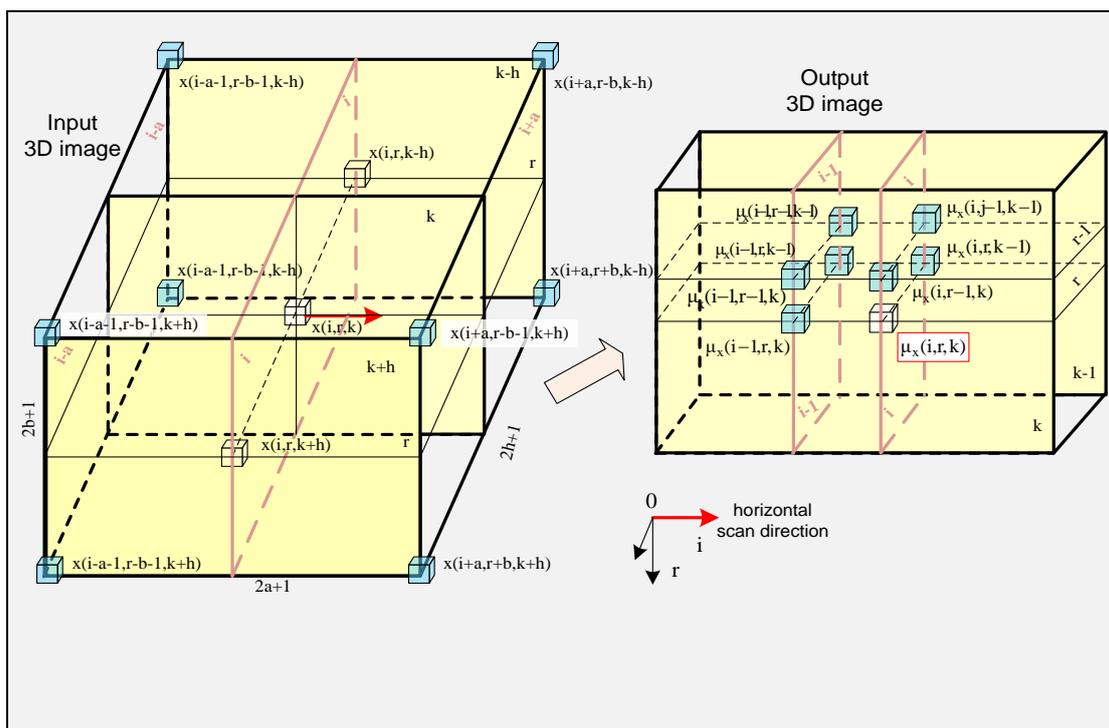


Figure 2. Spatial positioning of voxels (in blue) used for the recursive calculation of $\mu_x(i, r, k)$ in the 3D sliding window $W(i, r, k)$ of size $(2a + 1) \times (2b + 1) \times (2h + 1)$.

The voxels, used for the recursive calculation, are colored in blue, and the current voxel $\mu_x(i, r, k)$ is white. In the causal area of the filter, the following voxels of the output 3D image are placed: $\mu_x(i - 1, r, k)$, $\mu_x(i, r - 1, k)$, $\mu_x(i, r, k - 1)$, $\mu_x(i - 1, r - 1, k)$, $\mu_x(i - 1, r, k - 1)$, $\mu_x(i - 1, r - 1, k - 1)$, and $\mu_x(i, r - 1, k - 1)$. These voxels are already calculated, because for the selected direction of the linear scan (as shown in Figure 1), they correspond to the previous positions of the current voxel $\mu_x(i, j, k)$ at the output of the recursive SM3DF.

2.3. Analysis of the Recursive Sliding Mean 3D Filter Frequency Response

To analyze the frequency response of the recursive sliding mean 3D filter, in Equation (5) the following are set: $z_1 = e^{j\omega_1}$, $z_2 = e^{j\omega_2}$, and $z_3 = e^{j\omega_3}$, where $\omega_1 = 2\pi(f_x T_x)$, $\omega_2 = 2\pi(f_y T_y)$, and $\omega_3 = 2\pi(f_z T_z)$. Here, f_x , f_y , and f_z denote the spatial frequencies, and T_x , T_y , T_z are the distances between the neighboring voxels in a horizontal, vertical and lateral direction, respectively.

After replacing the variables in Equation (5) for the 3D complex frequency response (CFR) of SM3DF, we obtained:

$$H(e^{j\omega_1}, e^{j\omega_2}, e^{j\omega_3}) = \frac{M_x(e^{j\omega_1}, e^{j\omega_2}, e^{j\omega_3})}{X(e^{j\omega_1}, e^{j\omega_2}, e^{j\omega_3})} = H_1(e^{j\omega_1})H_2(e^{j\omega_2})H_3(e^{j\omega_3}), \quad (8)$$

where, for the corresponding one-dimensional CFR in respect to ω_1 , ω_2 , ω_3 , the following relations are obtained:

$$H_1(e^{j\omega_1}) = \frac{e^{ja\omega_1} - e^{-j(a+1)\omega_1}}{A_1(1 - e^{-j\omega_1})} = \frac{\{\cos(a\omega_1) - \cos(a+1)\omega_1\} + j\{\sin(a\omega_1) + \sin(a+1)\omega_1\}}{A_1[(1 - \cos\omega_1) + j\sin\omega_1]}, \quad (9)$$

$$H_2(e^{j\omega_2}) = \frac{e^{jb\omega_2} - e^{-j(b+1)\omega_2}}{A_2(1 - e^{-j\omega_2})} = \frac{\{\cos(b\omega_2) - \cos(b+1)\omega_2\} + j\{\sin(b\omega_2) + \sin(b+1)\omega_2\}}{A_2[(1 - \cos\omega_2) + j\sin\omega_2]}, \quad (10)$$

$$H_3(e^{j\omega_3}) = \frac{e^{jh\omega_3} - e^{-j(h+1)\omega_3}}{A_3(1 - e^{-j\omega_3})} = \frac{\{\cos(h\omega_3) - \cos(h+1)\omega_3\} + j\{\sin(h\omega_3) + \sin(h+1)\omega_3\}}{A_3[(1 - \cos\omega_3) + j\sin\omega_3]}. \quad (11)$$

After applying the necessary trigonometric relations:

$$\cos(a\omega_1) - \cos(a+1)\omega_1 = -2\sin((2a+1)\omega_1/2)\sin(-\omega_1/2) = 2\sin(A_1\omega_1/2)\sin(\omega_1/2) \quad (12)$$

and

$$\sin(a\omega_1) + \sin(a+1)\omega_1 = 2\sin((2a+1)\omega_1/2)\cos(-\omega_1/2) = 2\sin(A_1\omega_1/2)\cos(\omega_1/2), \quad (13)$$

the corresponding 1D quadratic amplitude–frequency response (AFR) is defined:

$$\begin{aligned} M_1^2(\omega_1) &= |H_1(e^{j\omega_1})|^2 = \frac{\{\cos(a\omega_1) - \cos(a+1)\omega_1\}^2 + \{\sin(a\omega_1) + \sin(a+1)\omega_1\}^2}{A_1^2[(1 - \cos\omega_1)^2 + \sin^2\omega_1]} = \\ &= \frac{4\sin^2(A_1\omega_1/2)}{2A_1^2(1 - \cos\omega_1)} = \frac{1}{A_1^2} \left[\frac{\sin(A_1\omega_1/2)}{\sin(\omega_1/2)} \right]^2 = \frac{1}{A_1^2} \left[\frac{A_1 \sin c(A_1\omega_1/2)}{\sin c(\omega_1/2)} \right]^2 = \\ &= \left[\frac{\sin c(A_1\omega_1/2)}{\sin c(\omega_1/2)} \right]^2, \text{ for } \sin c(x) = \frac{\sin(x)}{x}. \end{aligned} \quad (14)$$

$$M_2^2(\omega_2) = \frac{\{\cos(b\omega_2) - \cos(b+1)\omega_2\}^2 + \{\sin(b\omega_2) + \sin(b+1)\omega_2\}^2}{A_2^2[(1 - \cos\omega_2)^2 + \sin^2\omega_2]} = \left[\frac{\sin c(A_2\omega_2/2)}{\sin c(\omega_2/2)} \right]^2; \quad (15)$$

$$M_3^2(\omega_3) = \frac{\{\cos(h\omega_3) - \cos(h+1)\omega_3\}^2 + \{\sin(h\omega_3) + \sin(h+1)\omega_3\}^2}{A_3^2[(1 - \cos\omega_3)^2 + \sin^2\omega_3]} = \left[\frac{\sin c(A_3\omega_3/2)}{\sin c(\omega_3/2)} \right]^2. \quad (16)$$

Then, the 3D AFR of SM3DF is represented as:

$$\begin{aligned} M(\omega_1, \omega_2, \omega_3) &= \sqrt{M_1^2(\omega_1)M_2^2(\omega_2)M_3^2(\omega_3)} = \\ &= \left| \frac{\sin c(A_1\omega_1/2)}{\sin c(\omega_1/2)} \right| \times \left| \frac{\sin c(A_2\omega_2/2)}{\sin c(\omega_2/2)} \right| \times \left| \frac{\sin c(A_3\omega_3/2)}{\sin c(\omega_3/2)} \right|. \end{aligned} \quad (17)$$

The phase–frequency response (PFR) in respect to ω_1 , is:

$$\phi(\omega_1) = \text{arctg} \frac{\beta_1\gamma_1 - \alpha_1\psi_1}{\alpha_1\gamma_1 + \beta_1\psi_1} = \text{arctg} \frac{\sin\omega_1 - \sin\omega_1}{1 + \sin\omega_1 - \cos\omega_1} = 0, \quad (18)$$

where $\alpha_1 = 2 \sin(A_1 \omega_1 / 2) \sin(\omega_1 / 2)$; $\beta_1 = 2s \sin(A_1 \omega_1 / 2) \cos(\omega_1 / 2)$; $\psi_1 = A_1 \sin(\omega_1)$; $\gamma_1 = 2A_1 \sin^2(\omega_1 / 2)$.

In a similar way, we defined $\phi(\omega_2) = \phi(\omega_3) = 0$. Then, for the 3D CFR of SM3DF, the following is obtained:

$$\phi(\omega_1, \omega_2, \omega_3) = \phi(\omega_1) + \phi(\omega_2) + \phi(\omega_3) = 0. \quad (19)$$

In this case, the following relation is obtained for 3D CFR, 3D AFR and 3D PFR:

$$H(e^{j\omega_1}, e^{j\omega_2}, e^{j\omega_3}) = M(\omega_1, \omega_2, \omega_3) e^{j\phi(\omega_1, \omega_2, \omega_3)} = M(\omega_1, \omega_2, \omega_3). \quad (20)$$

Hence, the SM3DF does not introduce phase distortions in the output tensor. From Equations (17) and (20) it follows that AFR and CFR obtain values equal to zero for each frequency, $\omega_1 = 2\pi n / A_1$, $\omega_2 = 2\pi n / A_2$, $\omega_3 = 2\pi n / A_3$, for $n = 1, 2, \dots$

Example 1: Let $a = b = h = 1$, i.e., the sliding window $W(i, r, k)$ is of size $3 \times 3 \times 3$. Moreover, in the case of $M = N = K = 4$, the input tensor \mathbf{X} is of size $4 \times 4 \times 4$. Then, for $i = r = 3$ and $k = 2$, from Equations (1) and (7) we obtain the corresponding relations for the non-recursive and the recursive calculation of $\mu_x(3, 3, 2)$:

- for the non-recursive calculation:

$$\begin{aligned} \mu_x(3, 3, 2) = & \frac{1}{27} \sum_{s=-1}^1 \sum_{m=-1}^1 \sum_{n=-1}^1 x(3+s, 3+m, 2+n) = \frac{1}{27} \{ [x(3, 3, 2) + x(2, 3, 2) + x(4, 3, 2) + x(3, 2, 2) + \\ & + x(3, 4, 2) + x(2, 2, 2) + x(4, 4, 2) + x(4, 4, 2) + x(2, 4, 2)] + [x(3, 3, 1) + x(2, 3, 1) + x(4, 3, 1) + x(3, 2, 1) + \\ & + x(3, 4, 1) + x(2, 2, 1) + x(4, 4, 1) + x(4, 4, 1) + x(2, 4, 1)] + [x(3, 3, 3) + x(2, 3, 3) + x(4, 3, 3) + x(3, 2, 3) + \\ & + x(3, 4, 3) + x(2, 2, 3) + x(4, 4, 3) + x(4, 4, 3) + x(2, 4, 3)] \}. \end{aligned}$$

- for the recursive calculation:

$$\begin{aligned} \mu_x(3, 3, 2) = & \mu_x(2, 3, 2) + \mu_x(3, 2, 2) + \mu_x(3, 3, 1) - \mu_x(2, 2, 2) - \mu_x(2, 3, 1) - \mu_x(3, 2, 1) + \mu_x(2, 2, 1) + \\ & + \frac{1}{27} [x(4, 4, 3) - x(4, 1, 3) - x(1, 4, 3) + x(1, 1, 3) - x(4, 4, 0) + x(4, 1, 0) + x(1, 4, 0) - x(1, 1, 0)]. \end{aligned}$$

Let the values of the tensor \mathbf{X} voxels $x(i, r, k)$ be defined as elements of the four matrices \mathbf{X}_{k+1} , \mathbf{X}_k , \mathbf{X}_{k-1} , and \mathbf{X}_{k-2} (sections of tensor \mathbf{X}), each of size 4×4 , for $i/j = 1, 2, 3, 4$, and $k = 2$:

$$\mathbf{X}_3 = \begin{bmatrix} 1 & 2 & 2 & 1 \\ 2 & 1 & 3 & 1 \\ 1 & 2 & 4 & 2 \\ 2 & 2 & 1 & 1 \end{bmatrix}, \mathbf{X}_2 = \begin{bmatrix} 2 & 2 & 3 & 3 \\ 1 & 2 & 2 & 3 \\ 2 & 3 & 1 & 1 \\ 3 & 4 & 3 & 3 \end{bmatrix}, \mathbf{X}_1 = \begin{bmatrix} 4 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 2 & 3 & 2 & 1 \\ 3 & 2 & 3 & 2 \end{bmatrix}, \mathbf{X}_0 = \begin{bmatrix} 3 & 2 & 1 & 1 \\ 1 & 2 & 2 & 1 \\ 2 & 3 & 2 & 2 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

In this case, the value of $\mu_x(3, 3, 2)$ is calculated non-recursive in accordance with the relation:

$$\begin{aligned} \mu_x(3, 3, 2) = & \frac{1}{27} [(3+1+1+2+2+3+4+3+3) + (3+2+1+2+1+1+2+3+2) + \\ & + (2+4+2+1+3+1+2+1+1)] = \frac{56}{27} = 2.074(074). \end{aligned}$$

To calculate recursively the value of $\mu_x(3, 3, 2)$, we should calculate in advance the preceding values: $\mu_x(2, 3, 2)$, $\mu_x(3, 2, 2)$, $\mu_x(2, 2, 2)$, $\mu_x(2, 3, 1)$, $\mu_x(3, 2, 1)$, $\mu_x(3, 3, 1)$, and $\mu_x(2, 2, 1)$. These values are calculated recursively, in accordance with the relations below:

$$\begin{aligned} \mu_x(2, 3, 2) = & (1/27) [(2+3+1+1+2+2+3+4+3) + (1+2+4+2+1+3+2+2+1) + \\ & + (1+2+1+2+3+2+3+2+3)] = 58/27 \end{aligned}$$

$$\begin{aligned} \mu_x(3, 2, 2) = & (1/27) [(2+2+1+1+3+1+2+4+2) + (2+3+3+2+2+3+3+1+1) + \\ & + (1+1+1+2+1+1+3+2+1)] = 51/27 \end{aligned}$$

$$\mu_x(3,3,1) = (1/27)[(3+2+1+2+1+1+2+3+2) + (3+1+1+2+2+3+4+3+3) + (3+2+2+2+2+1+1+1+2)] = 55/27$$

$$\mu_x(2,2,2) = (1/27)[(1+2+2+2+2+3+2+3+1) + (2+1+3+1+2+2+1+2+4) + (1+2+1+4+1+1+2+3+2)] = 53/27$$

$$\mu_x(2,3,1) = (1/27)[(2+3+2+1+2+1+3+2+3) + (2+3+1+1+2+2+3+4+3) + (2+3+2+1+2+2+3+1+1)] = 57/27$$

$$\mu_x(3,2,1) = (1/27)[(2+1+1+1+1+1+3+2+1) + (2+2+3+2+3+3+3+1+1) + (2+2+1+2+1+1+3+2+2)] = 49/27$$

$$\mu_x(2,2,1) = (1/27)[(1+2+1+4+1+1+2+3+2) + (1+2+2+2+2+3+2+3+1) + (1+2+2+3+2+1+2+3+2)] = 53/27$$

Here, it should be taken into account that in order to avoid division errors in the calculation of the intermediate values used for the recursive case, the division should be executed last.

Then, for the recursive calculation of $\mu_x(3,3,2)$ and in accordance with Equation (7), it follows that:

$$\mu_x(3,3,2) = \frac{58}{27} + \frac{51}{27} + \frac{55}{27} - \frac{53}{27} - \frac{57}{27} - \frac{49}{27} + \frac{53}{27} + \frac{1}{27}(1-1-2+1-2+1+3-3) = \frac{56}{27}.$$

The obtained result confirms that the recursive and the non-recursive filtering are equivalent in respect of the calculated value of $\mu_x(3,3,2)$.

2.4. Computation Complexity of the Recursive SM3DF

The computation complexity (CC) of the recursive SM3DF is evaluated, taking into account the number of operations needed to calculate the central voxel in the part framed by the sliding window $W(i, r, k)$, in accordance with Equation (7). The number of additions is $S = 14$, and the number of multiplications is $M = 1$. Then, the number of operations needed is $O = (S + M) = 15$, i.e., O does not depend on the values of parameters a, b , and h of the window $W(i, r, k)$.

To define the CC regarding the central voxel, for the non-recursive SM3DF, the following should be calculated: the number of additions, $S = (2a + 1)(2b + 1)(2h + 1) - 1$, and the number of multiplications, $M = 1$. Then, the total number of operations is defined by the relation:

$$O = (2a + 1)(2b + 1)(2h + 1) \quad (21)$$

The CC reduction for the recursive SM3DF is:

$$R_1 = (2a + 1)(2b + 1)(2h + 1)/15 \quad (22)$$

2.5. SM3DF Implementation through Cascade Sliding Recursive Mean 1D Filters

The SM3DF transfer function in correspondence with Equation (3), is given by the relation:

$$H(z_1, z_2, z_3) = \frac{M_x(z_1, z_2, z_3)}{X(z_1, z_2, z_3)} = \left(\frac{M_1(z_1)}{X(z_1)} \right) \left(\frac{M_2(z_2)}{M_1(z_2)} \right) \left(\frac{M_3(z_3)}{M_2(z_3)} \right) = H_1(z_1)H_2(z_2)H_3(z_3), \quad (23)$$

where

$$H_1(z_1) = \frac{z_1^a - z_1^{-(a+1)}}{A_1(1 - z_1^{-1})}; H_2(z_2) = \frac{z_2^b - z_2^{-(b+1)}}{A_2(1 - z_2^{-1})}; H_3(z_3) = \frac{z_3^h - z_3^{-(h+1)}}{A_3(1 - z_3^{-1})}. \quad (24)$$

Each of the transfer functions could be presented by the relations:

$$H(z) = \frac{M(z)}{X(z)} = \frac{z^a - z^{-(a+1)}}{A(1 - z^{-1})}, \text{ or } M(z) = M(z)z^{-1} + (1/A)[X(z)z^a - X(z)z^{-a-1}]. \quad (25)$$

Then, 3D filtering is executed by three sequential mean 1D filters, which scan the 3D image in directions x, y, z , and are defined by the relations below:

$$\mu_1(i) = \mu_1(i - 1) + (1/A_1)[x(i + a) - x(i - a - 1)]; \quad (26)$$

$$\mu_2(r) = \mu_2(r - 1) + (1/A_2)[\mu_1(r + b) - \mu_1(r - b - 1)]; \quad (27)$$

$$\mu_3(k) = \mu_3(k - 1) + (1/A_3)[\mu_2(k + h) - \mu_2(k - h - 1)]. \quad (28)$$

The result, obtained at the output of the third 1D recursive filter, coincides with that at the output of the recursive 3D filter.

2.6. CC of the Recursive SM3DF, Implemented through Triple Recursive Mean 1D Filter

In this case, the CC is evaluated by the number of operations, O , needed to calculate the central voxel in the window $W(i, j, k)$. As was mentioned above, $O_i = S_i + M_i$ (here S_i and M_i represent the number of additions and multiplications for the 1D filtering, which is executed 3 times). In this case, for $S_i = 2$ and $M_i = 1$, the total number of operations is:

$$O = O_1 + O_2 + O_3 = 9. \quad (29)$$

In correspondence with Equation (21), the CC of the non-recursive 3D filter is:

$$O = (2a + 1)(2b + 1)(2h + 1).$$

In respect to this, the CC reduction coefficient for one voxel is:

$$R_2 = \frac{(2a + 1)(2b + 1)(2h + 1)}{3[2(a + b + h) + 3]}. \quad (30)$$

Example 2: $a = b = h = 5$ (i.e., 3D window of size $11'11 \times 11$), and $R_2 = 11^3/99 = 13.44$.

The execution time for the 3D filtration of the voxels in a window of size $(2a + 1)(2b + 1)(2h + 1)$, when triple 1D filtration is used, is:

$$T = 3(2a + 1)(2b + 1)(2h + 1)t, \quad (31)$$

where t is the time needed to scan one voxel of the tensor X . The reduced CC of the recursive filtration with one 3D filter, compared to the reduced CC for the case when it is implemented as three sequential 1D filters, for the same values of parameters $a = b = h = c$, is defined by the relation:

$$R_1/R_2 = 3(2c + 1)/5. \quad (32)$$

For example, if $c = 5$ (cubic window of size 11), is obtained $R_1/R_2 = 6.6$, i.e., the global CC is reduced by more than 6 times.

The comparison between the recursive and the non-recursive filtration shows that the recursive filtration, implemented as three sequential 1D filters, reduces the CC significantly.

2.7. Computation Complexity of the Sliding Mean 3D Filter, Based on 3D FFT

The filtration of tensor X of size $M \times N \times K$ could be replaced by similar filtration, but in the frequency domain of the 3D discrete Fourier transform (3D-DFT) [7]. To accelerate calculations, the 3D-DFT is usually replaced by the 3D fast Fourier transform (3D-FFT). For this, the implementation is based on the one-dimensional fast Fourier transform (FFT) of

the M -dimensional vector, which needs $\frac{M}{2} \log_2 M$ mathematical operations [7]. In this case, each operation comprises one complex multiplication and two complex summations. For a tensor of size $M \times N \times K$, the number of M -dimensional vectors in one tensor section is N , and, correspondingly, NK for all K sections. Then, the total number of operations needed to execute FFT for NK vectors with horizontal orientation is $NK \frac{M}{2} \log_2 M$.

For the 3D FFT calculation, FFT should be applied sequentially on all NK vectors with horizontal, vertical, and lateral orientation, respectively, and as a result, the total number of operations is:

$$O_{\text{FFT}}^1 = 3NK \frac{M}{2} \log_2 M. \quad (33)$$

The number of operations needed to calculate the module of each complex frequency coefficient in the 3D spectrum of the tensor \mathbf{X} comprises two multiplications: one summation and one rooting. Hence, for all MNK frequency coefficients, this number is:

$$O_{\text{FFT}}^2 = 4MNK. \quad (34)$$

The total number of operations needed to calculate the filtered spectrum coefficients after multiplication with the corresponding coefficients of 3D AFR of SM3DF is:

$$O_{\text{FFT}}^3 = MNK. \quad (35)$$

The total number of operations needed to execute the 3D inverse FFT for all $3NK$ vectors in directions x, y, z , differs from that in Equation (33) by one operation only, needed for the restored voxels' normalization:

$$O_{\text{FFT}}^4 = 3NK \frac{M}{2} \log_2 M + 1 \approx 3NK \frac{M}{2} \log_2 M. \quad (36)$$

Hence, the total number of operations needed for the filtration of a tensor of size $M \times N \times K$ in the frequency domain is:

$$O_{\text{FFT}} = O_{\text{FFT}}^1 + O_{\text{FFT}}^2 + O_{\text{FFT}}^3 + O_{\text{FFT}}^4 = MNK(5 + 3 \log_2 M). \quad (37)$$

The total number of operations needed for the recursive 3D locally-adaptive filtration of the tensor \mathbf{X} of size $M \times N \times K$ does not depend on the window dimensions $(2a + 1) \times (2b + 1) \times (2h + 1)$, and is defined by the relation:

$$O_{\text{RF}} = 15MNK. \quad (38)$$

Then, the relation R_3 of O_{FFT} towards O_{RF} is:

$$R_3 = O_{\text{FFT}}/O_{\text{RF}} = 0.33 + 0.2 \log_2 M. \quad (39)$$

For example, if $M = 512$, then $R_3 = 32/15 = 2.13$, i.e., the filtration CC in the tensor frequency space is twice as low as that in the voxel space. Together with the M increase, the relation R_3 grows proportionally to $\log_2 M$, and is not influenced by the filter window size. For the case when $M = 2^m$, the values obtained for the relation of R_3 for $m = 3, 4, \dots, 10$, are shown in Table 1.

Table 1. Experimental results for R_3 values.

m	3	4	5	6	7	8	9	10
R_3	0.93	1.13	1.33	1.53	1.73	1.93	2.13	2.33

The total number of operations needed for the non-recursive 3D locally-adaptive filtration of the tensor \mathbf{X} of size $M \times N \times K$, through a filter window of size $(2c + 1) \times (2c + 1) \times (2c + 1)$, for $c = a = b = h$, is:

$$O_{\text{NRF}} = MNK(2c + 1)^3. \quad (40)$$

Then, the relation R_4 for O_{FFT} regarding O_{NRF} , is represented as given below:

$$R_4 = O_{\text{FFT}}/O_{\text{NRF}} = (5 + 3 \log_2 M)/(2c + 1)^3. \quad (41)$$

To satisfy the condition $R_4 > 1$ (i.e., to obtain a lower CC for the filtration in the tensor space compared to that for the frequency domain of 3D-DFT), the parameter c of the window must satisfy the condition:

$$c < \frac{1}{2} \left(\sqrt[3]{5 + 3 \log_2 M} - 1 \right). \quad (42)$$

For example, for $N = 512$, $c < 1.08$. In fact, for $c = 1$ (window of size $3 \times 3 \times 3$), we obtain $R_2 = 1.18 > 1$. Unlike the recursive calculation, for the non-recursive calculation the filter window size is limited by the horizontal size of the tensor M , in correspondence with Equation (42), which is a significant advantage of the recursive filtration.

3. Locally-Adaptive 3D Image Noise Filtration, Based on the Recursive SM3DF

3.1. Reduction of the Additive Gaussian Noise through Locally-Adaptive 3D Filter

The locally-adaptive 3D filter aimed at the reduction of the additive Gaussian noise is the generalized Wiener 2D filter [8], represented by the relation:

$$g(i, r, k) = \begin{cases} \mu_x(i, r, k) + \frac{\sigma_x^2(i, r, k) - v^2}{\sigma_x^2(i, r, k)} [x(i, r, k) - \mu_x(i, r, k)] & \text{for } \sigma_x^2(i, r, k) \geq v^2, \\ \mu_x(i, r, k) & \text{for } \sigma_x^2(i, r, k) < v^2, \end{cases} \quad (43)$$

where:

- $g(i, r, k)$ denotes the filtered input pixel $x(i, r, k)$;
- $\mu_x(i, r, k) = \frac{1}{A} \sum_{s=-a}^a \sum_{m=-b}^b \sum_{n=-h}^h x(i + s, r + m, k + n)$ is the mean value of pixels $x(i, r, k)$ in the sliding window $W(i, r, k)$ of size $(2a + 1) \times (2b + 1) \times (2h + 1)$;
- $A = (2a + 1)(2b + 1)(2h + 1)$ —the number of voxels in the sliding window $W(i, r, k)$ (it is recommended that a square window $W(i, r, k)$ of size $(2c + 1) \times (2c + 1) \times (2c + 1)$ is used for $(a = b = h = c)$);
- $\sigma_x^2(i, r, k) = \frac{1}{A} \sum_{s=-a}^a \sum_{m=-b}^b \sum_{n=-h}^h x^2(i + s, r + m, k + n) - \mu_x^2(i, r, k)$ —the local variance of the voxels $x(i, r, k)$, enveloped by the sliding window $W(i, r, k)$;
- $v^2 = \frac{1}{MNK} \sum_{i=1}^M \sum_{r=1}^N \sum_{k=1}^K \sigma_x^2(i, r, k)$ —the mean noise variance in the tensor image \mathbf{X} , of size $M \times N \times K$.

For the case when $\sigma_x^2(i, r, k) \geq v^2$, from Equation (33) it follows that:

$$g(i, r, k) = \left\{ \frac{\sigma_x^2(i, r, k) - v^2}{\sigma_x^2(i, r, k)} \right\} x(i, r, k) + [v^2 / \sigma_x^2(i, r, k)] \mu_x(i, r, k) \quad (44)$$

To accelerate the calculation of the local mean value $\mu_x(i, r, k)$ in the above relation for the case of a linear scan of the input tensor \mathbf{X} , (excluding the first and the last horizontal and lateral row, and the first and the last column), Equation (7) should be used.

For the acceleration of the local variance $\sigma_x^2(i, r, k) = p(i, r, k) - \mu_x^2(i, r, k)$ calculation, for its first component $p(i, r, k) = \frac{1}{A} \sum_{s=-a}^a \sum_{m=-b}^b \sum_{n=-h}^h x^2(i + s, r + m, k + n)$ a rela-

tion similar to the recursive Equation (7) should be used, where the values of voxels $x(i + s, r + m, k + n)$ are replaced by their squares, $x^2(i + s, r + m, k + n)$:

$$\begin{aligned}
 p(i, r, k) = & p(i - 1, r, k) + p(i, r - 1, k) + p(i, r, k - 1) - p(i - 1, r - 1, k) - p(i - 1, r, k - 1) - p(i, r - 1, k - 1) + \\
 & + p(i - 1, r - 1, k - 1) + \frac{1}{A} [x^2(i + a, r + b, k + h) - x^2(i + a, r - b - 1, k + h) - x^2(i - a - 1, r + b, k + h) + \\
 & + x^2(i - a - 1, r - b - 1, k + h) - x^2(i + a, r + b, k - h - 1) + x^2(i + a, r - b - 1, k - h - 1) + \\
 & + x^2(i - a - 1, r + b, k - h - 1) - x^2(i - a - 1, r - b - 1, k - h - 1)].
 \end{aligned}
 \tag{45}$$

3.2. Transfer Function of the Locally-Adaptive Denoising 3D Filter

Let us present Equation (43) as given below:

$$g(i, r, k) = Cx(i, r, k) + \frac{D}{A} \sum_{s=-a}^a \sum_{m=-b}^b \sum_{n=-h}^h x(i + s, r + m, k + n),
 \tag{46}$$

where $C = [\sigma_x^2(i, r, k) - v^2] / \sigma_x^2(i, r, k)$; $D = v^2 / \sigma_x^2(i, r, k)$.

After applying the 3D z-transform on Equation (46), the 3D transfer function is defined:

$$H_g(z_1, z_2, z_3) = \frac{G(z_1, z_2, z_3)}{X(z_1, z_2, z_3)} = C + \frac{D}{A} \left(\sum_{s=-a}^a z_1^s \right) \left(\sum_{m=-b}^b z_2^m \right) \left(\sum_{n=-h}^h z_3^n \right).
 \tag{47}$$

In accordance with Equation (24), the relation (47) is transformed as follows:

$$\begin{aligned}
 H_g(z_1, z_2, z_3) = & C + \frac{D}{A} \left(\frac{z_1^a - z_1^{-(a+1)}}{1 - z_1^{-1}} \right) \left(\frac{z_2^b - z_2^{-(b+1)}}{1 - z_2^{-1}} \right) \left(\frac{z_3^h - z_3^{-(h+1)}}{1 - z_3^{-1}} \right) = \\
 = & \frac{AC(1 - z_1^{-1})(1 - z_2^{-1})(1 - z_3^{-1}) + D(z_1^a - z_1^{-(a+1)})(z_2^b - z_2^{-(b+1)})(z_3^h - z_3^{-(h+1)})}{A(1 - z_1^{-1})(1 - z_2^{-1})(1 - z_3^{-1})},
 \end{aligned}
 \tag{48}$$

from which it follows:

$$\begin{aligned}
 H_g(z_1, z_2, z_3) = & \frac{G(z_1, z_2, z_3)}{X(z_1, z_2, z_3)} = C \frac{1 - z_1^{-1} - z_2^{-1} - z_3^{-1} + z_1^{-1}z_2^{-1} + z_1^{-1}z_3^{-1} + z_2^{-1}z_3^{-1} - z_1^{-1}z_2^{-1}z_3^{-1}}{1 - z_1^{-1} - z_2^{-1} - z_3^{-1} + z_1^{-1}z_2^{-1} + z_1^{-1}z_3^{-1} + z_2^{-1}z_3^{-1} - z_1^{-1}z_2^{-1}z_3^{-1}} + \\
 & + \frac{D}{A} \frac{z_1^a z_2^b z_3^h - z_1^{a-(b+1)} z_2^{b-(a+1)} z_3^{h-(a+1)} - z_1^{b-(a+1)} z_2^{a-(b+1)} z_3^{h-(a+1)} - z_1^{a-(b+1)} z_2^{b-(a+1)} z_3^{h-(a+1)}}{1 - z_1^{-1} - z_2^{-1} - z_3^{-1} + z_1^{-1}z_2^{-1} + z_1^{-1}z_3^{-1} + z_2^{-1}z_3^{-1} - z_1^{-1}z_2^{-1}z_3^{-1}} + \\
 & + \frac{D}{A} \frac{z_1^a z_2^{-(b+1)} z_3^{-(h+1)} + z_1^{-(a+1)} z_2^b z_3^{-(h+1)} - z_1^{-(a+1)} z_2^{-(b+1)} z_3^{-(h+1)}}{1 - z_1^{-1} - z_2^{-1} - z_3^{-1} + z_1^{-1}z_2^{-1} + z_1^{-1}z_3^{-1} + z_2^{-1}z_3^{-1} - z_1^{-1}z_2^{-1}z_3^{-1}}
 \end{aligned}
 \tag{49}$$

After applying the inverse 3D z-transform on Equation (49), the recursive relation is obtained, which represents the performance of the sliding locally-adaptive denoising 3D filter:

$$\begin{aligned}
 g(i, r, k) = & g(i - 1, r, k) + g(i, r - 1, k) + g(i, r, k - 1) - g(i - 1, r - 1, k) - g(i - 1, r, k - 1) - \\
 & - g(i, r - 1, k - 1) + g(i - 1, r - 1, k - 1) + C[x(i, r, k) + x(i - 1, r, k) + x(i, r - 1, k) + \\
 & + x(i, r, k - 1) - x(i - 1, r - 1, k) - x(i - 1, r, k - 1) - x(i, r - 1, k - 1) + x(i - 1, r - 1, k - 1)] + \\
 & + \frac{D}{A} [x(i + a, r + b, k + h) - x(i + a, r - b - 1, k + h) - x(i, a - 1, r + b, k + h) - \\
 & - x(i - a - 1, r - b - 1, k + h) - x(i + a, r + b, k - h - 1) + x(i + a, r - b - 1, k - h - 1) + \\
 & + x(i - a - 1, r + b, k - h - 1) - x(i - a - 1, r - b - 1, k - h - 1)]
 \end{aligned}
 \tag{50}$$

From Equation (50) it follows that for the recursive calculation of $g(i, r, k)$, and in accordance with the above relation, 25 additions and 2 multiplications only are needed. In Figure 3, we show the spatial positioning of the voxels, used for the recursive calculation of the $g(i, r, k)$ value.

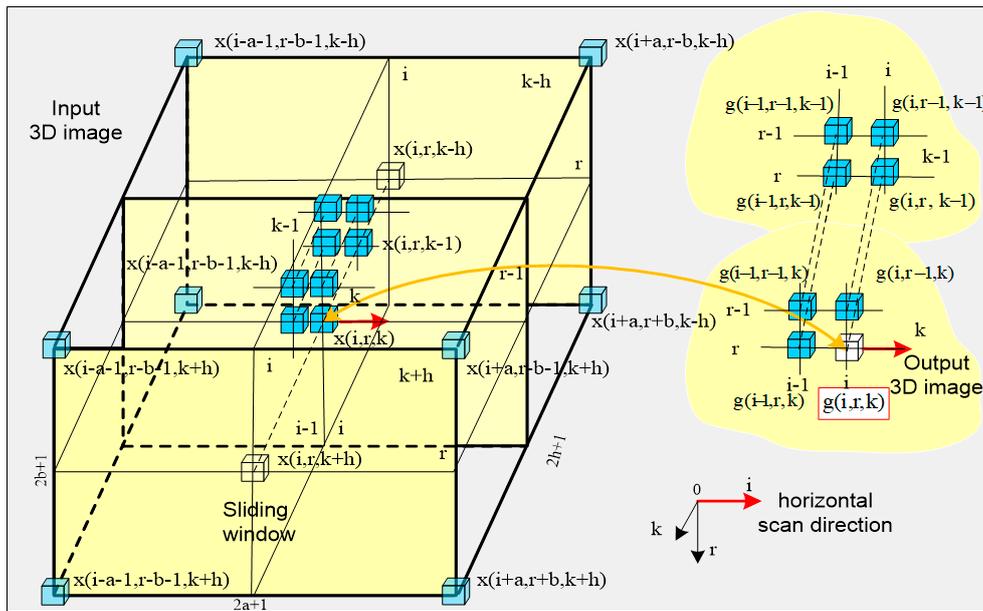


Figure 3. Spatial positioning of voxels (in blue) used for the recursive calculation of $g(i, r, k)$ in a 3D sliding window $W(i, r, k)$ of size $(2a + 1) \times (2b + 1) \times (2h + 1)$.

3.3. Frequency Response Analysis of the Recursive Sliding Locally-Adaptive Denoising 3D Filter

The CFP of the filter is defined after replacing $z_1 = e^{j\omega_1}$, $z_2 = e^{j\omega_2}$, and $z_3 = e^{j\omega_3}$ in Equation (48):

$$H_g(e^{j\omega_1}, e^{j\omega_2}, e^{j\omega_3}) = C + \frac{D}{A} \left(\frac{e^{ja\omega_1} - e^{-j(a+1)\omega_1}}{1 - e^{-j\omega_1}} \right) \left(\frac{e^{jb\omega_2} - e^{-j(b+1)\omega_2}}{1 - e^{-j\omega_2}} \right) \left(\frac{e^{jh\omega_3} - e^{-j(h+1)\omega_3}}{1 - e^{-j\omega_3}} \right). \quad (51)$$

Taking into account that:

$$\left| \frac{e^{ja\omega_1} - e^{-j(a+1)\omega_1}}{1 - e^{-j\omega_1}} \right| = \frac{\sin c(A_1 \omega_1 / 2)}{\sin c(\omega_1 / 2)}, \quad \left| \frac{e^{jb\omega_2} - e^{-j(b+1)\omega_2}}{1 - e^{-j\omega_2}} \right| = \frac{\sin c(A_2 \omega_2 / 2)}{\sin c(\omega_2 / 2)}, \quad \text{and}$$

$$\left| \frac{e^{jh\omega_3} - e^{-j(h+1)\omega_3}}{1 - e^{-j\omega_3}} \right| = \frac{\sin c(A_3 \omega_3 / 2)}{\sin c(\omega_3 / 2)}, \quad \text{for } A_1 = (2a + 1), A_2 = (2b + 1), \text{ and } A_3 = (2h + 1),$$

from Equation (51), it follows that:

$$H_g(e^{j\omega_1}, e^{j\omega_2}, e^{j\omega_3}) = C + \frac{D}{A} \left| \frac{\sin c(A_1 \omega_1 / 2)}{\sin c(\omega_1 / 2)} \right| \left| \frac{\sin c(A_2 \omega_2 / 2)}{\sin c(\omega_2 / 2)} \right| \left| \frac{\sin c(A_3 \omega_3 / 2)}{\sin c(\omega_3 / 2)} \right| \quad (52)$$

Correspondingly, the filter AFR and PFR are defined by the relations:

$$M_g(\omega_1, \omega_2, \omega_3) = C + \left(\frac{D}{A} \right) \left| \frac{\sin c(A_1 \omega_1 / 2)}{\sin c(\omega_1 / 2)} \right| \left| \frac{\sin c(A_2 \omega_2 / 2)}{\sin c(\omega_2 / 2)} \right| \left| \frac{\sin c(A_3 \omega_3 / 2)}{\sin c(\omega_3 / 2)} \right|, \quad (53)$$

and $\phi(\omega_1, \omega_2, \omega_3) = 0$.

The obtained results prove that the filter does not introduce phase distortions. For the low frequencies $\omega_1 = \omega_2 = \omega_3 = 0$, $M_g(0, 0, 0) = C + (D/A)$, while for the high frequencies $\omega_1 = 2\pi n/A_1$, $\omega_2 = 2\pi n/A_2$, and $\omega_3 = 2\pi n/A_3$, for $n = 1, 2, \dots$, $M_g(\omega_1, \omega_2, \omega_3) = C$. The term $D/A = v^2/A\sigma_x^2(i, r, k)$ defines the reduction of the AFR value for the high frequencies divisible by n towards that for the low frequencies, for $\omega_1 = \omega_2 = \omega_3 = 0$. The reduction is proportional to the relation $v^2/\sigma_x^2(i, r, k)$, and correspondingly, together with the increase in v^2 towards $\sigma_x^2(i, r, k)$, the suppression grows with regard to the high frequencies, divisible by n , whose main part are the additive noise components.

3.4. CC of the Algorithm for Locally-Adaptive Recursive 3D Image Denoising

The number of operations O_R needed for the recursive calculation of $g(i, r, k)$ for all voxels of tensor X of size $M \times N \times K$ in correspondence with Equation (39), is defined on the basis of the relations below:

- The number of operations needed for the recursive calculation of $\mu_x(i, r, k)$: $O_1 = 15 MNK$;
- The number of operations needed for the recursive calculation of $p(i, r, k)$: $O_2 = 31 MNK$;
- The number of operations needed for the calculation of $\sigma_x^2(i, r, k)$: $O_3 = 2 MNK$;
- The number of operations needed for the calculation of v^2 : $O_4 = MNK$;
- The number of operations needed for the calculation of C and D : $O_5 = 3 MNK$;
- The number of operations needed for the recursive calculation of $g(i, r, k)$: $O_6 = 27 MNK$.

The total number of operations needed for the recursive calculation of $g(i, r, k)$ is:
 $O_R = \sum_{i=1}^6 O_i = 79 MNK$. For the non-recursive calculation of $g(i, r, k)$: $O_{NR} = \sum_{i=1}^4 O_i = MNK[3(2a+1)(2b+1)(2h+1)+3]$.

In this case, the reduction coefficient for CC is:

$$R_5 = O_{NR}/O_R = [3(2a+1)(2b+1)(2h+1)+3]/79. \quad (54)$$

Example 3: Let $a = b = h = 5$ (3D window of size $11 \times 11 \times 11$). Then $R_5 = 50.58$. In this case, the calculations needed for the recursive approach are accelerated more than 50 times, compared to the non-recursive. On Figure 4 we show the graphic representation of R_c for $c = a = b = h$, i.e., for a cubic window of size $c \times c \times c$.

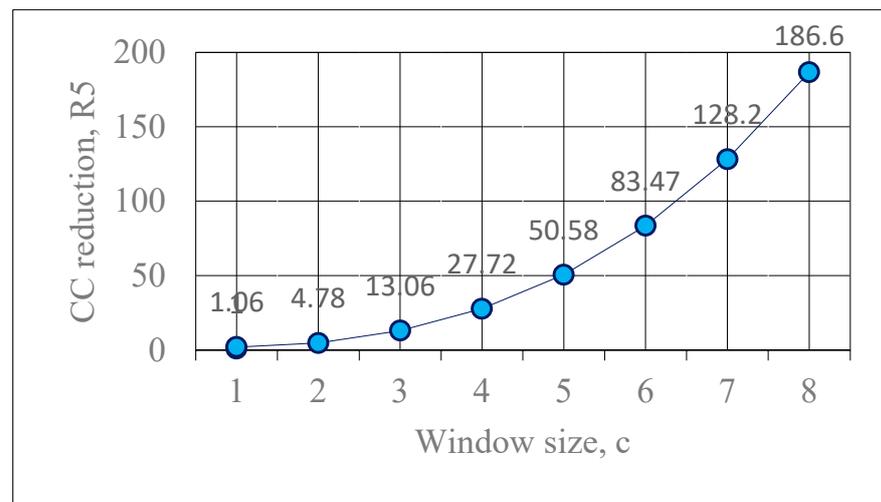


Figure 4. CC reduction of the locally-adaptive recursive 3D image denoising, as a function of the cubic window size.

It is easy to notice from Figure 4, that for small parameter values the reduction coefficient R is also relatively small, and increases together with the linear growth of the parameter.

4. Recursive 3D Image Sharpness through Adaptive Unsharp Masking

4.1. Description of the Non-Recursive 3D Image Sharpness through Sliding Adaptive Unsharp Masking 3D Filter

In this work, the non-recursive 3D image filtering is based on the mathematical description of the sliding adaptive unsharp masking 2D filter. The 3D generalization is represented by the relation below:

$$g(i, r, k) = \begin{cases} x(i, r, k) + \lambda[x(i, r, k) - \mu_x(i, r, k)] & \text{for } |x(i, r, k) - \mu_x(i, r, k)| \geq \delta, \\ x(i, r, k) & \text{in other cases} \end{cases}, \text{ for } 0 < \lambda < 1, \quad (55)$$

where $g(i, r, k)$ denotes a voxel from the filtered image, with enhanced sharpness; λ is a coefficient, used to control the sharpness enhancement, whose value is chosen experimentally in the range $0 < \lambda < 1$; δ is a threshold, $\delta \geq 2\sigma$; (σ —the noise variance); $\mu_x(i, r, k)$ is the local mean value of the voxels in the window $W(i, r, k)$, recursively calculated in correspondence with Equation (7).

4.2. Transfer Function of the Recursive Adaptive Unsharp Masking 3D Filter

For the case when $|x(i,r,k) - \mu_x(i,r,k)| \geq \delta$, the filter transfer function is defined by the relation:

$$\begin{aligned}
 H_g(z_1, z_2, z_3) &= \frac{G(z_1, z_2, z_3)}{X(z_1, z_2, z_3)} = 1 + \lambda \left[1 - \frac{(z_1^a - z_1^{-(a+1)})(z_2^b - z_2^{-(b+1)})(z_3^h - z_3^{-(h+1)})}{A(1-z_1^{-1})(1-z_2^{-1})(1-z_3^{-1})} \right] = \\
 &= \frac{(1+\lambda)(1-z_1^{-1}-z_2^{-1}-z_3^{-1}+z_1^{-1}z_2^{-1}+z_1^{-1}z_3^{-1}+z_2^{-1}z_3^{-1}-z_1^{-1}z_2^{-1}z_3^{-1})}{1-z_1^{-1}-z_2^{-1}-z_3^{-1}+z_1^{-1}z_2^{-1}+z_1^{-1}z_3^{-1}+z_2^{-1}z_3^{-1}-z_1^{-1}z_2^{-1}z_3^{-1}} - \\
 &\quad - \left(\frac{\lambda}{A} \right) \frac{z_1^a z_2^b z_3^h - z_1^{-(b+1)} z_2^h - z_1^{-(a+1)} z_2^b z_3^h + z_1^{-(a+1)} z_2^{-(b+1)} z_3^h - z_1^a z_2^b z_3^{-(h+1)}}{1-z_1^{-1}-z_2^{-1}-z_3^{-1}+z_1^{-1}z_2^{-1}+z_1^{-1}z_3^{-1}+z_2^{-1}z_3^{-1}-z_1^{-1}z_2^{-1}z_3^{-1}} - \\
 &\quad - \left(\frac{\lambda}{A} \right) \frac{z_1^a z_2^{-(b+1)} z_3^{-(h+1)} + z_1^{-(a+1)} z_2^b z_3^{-(h+1)} - z_1^{-(a+1)} z_2^{-(b+1)} z_3^{-(h+1)}}{1-z_1^{-1}-z_2^{-1}-z_3^{-1}+z_1^{-1}z_2^{-1}+z_1^{-1}z_3^{-1}+z_2^{-1}z_3^{-1}-z_1^{-1}z_2^{-1}z_3^{-1}}.
 \end{aligned} \tag{56}$$

After inverse 3D z-transform, from Equation (56) the following is obtained:

$$\begin{aligned}
 g(i, r, k) &= g(i-1, r, k) + g(i, r-1, k) + g(i, r, k-1) - g(i-1, r-1, k) - g(i-1, r, k-1) - g(i, r-1, k-1) + \\
 &+ g(i-1, r-1, k-1) + (1+\lambda)[x(i, r, k) - x(i-1, r, k) - x(i, r-1, k) - x(i, r, k-1) + x(i-1, r-1, k) + \\
 &+ x(i-1, r, k-1) + x(i, r-1, k-1) - x(i-1, r-1, k-1)] - (\lambda/A)[x(i+a, r+b, k+h) - \\
 &- x(i+a, r-b-1, k+h) - x(i-a-1, r+b, k+h) + x(i-a-1, r-b-1, k+h) - x(i+a, r+b, k-h-1) + \\
 &+ x(i+a, r-b-1, k-h-1) + x(i-a-1, r+b, k-h-1) - x(i-a-1, r-b-1, k-h-1)].
 \end{aligned} \tag{57}$$

The spatial positioning of the voxels, used for the recursive calculation of $g(i, r, k)$ in correspondence with Equation (57), coincides with those of the voxels, defined for the previous algorithm (shown in Figure 3).

4.3. Amplitude–Frequency Response of the Recursive Adaptive Unsharp Masking 3D Filter

The AFR is defined by the SM3DF response, in correspondence with Equations (17) and (47):

$$\begin{aligned}
 M_g(\omega_1, \omega_2, \omega_3) &= 1 + \lambda[1 - M(\omega_1, \omega_2, \omega_3)] = \\
 &= 1 + \lambda \left[1 - \left| \frac{\sin c(A_1 \omega_1/2)}{\sin c(\omega_1/2)} \right| \left| \frac{\sin c(A_2 \omega_2/2)}{\sin c(\omega_2/2)} \right| \left| \frac{\sin c(A_3 \omega_3/2)}{\sin c(\omega_3/2)} \right| \right].
 \end{aligned} \tag{58}$$

$$\text{If } \omega_2 = \omega_3 = 0, \text{ then } M_g(\omega_1) = 1 + \lambda[1 - M(\omega_1)] = 1 + \lambda \left[1 - \left| \frac{\sin c(A_1 \omega_1/2)}{\sin c(\omega_1/2)} \right| \right] \tag{59}$$

$$\text{If } \omega_1 = 0, \text{ then } M_g(0) = 1 + \lambda[1 - M(0)] = 1 + \lambda \left[1 - \left| \frac{\sin c(0)}{\sin c(0)} \right| \right] = 1. \tag{60}$$

$$\text{If } \omega_1 = 2\pi/A_1, \text{ then } M_g(2\pi/A_1) = 1 + \lambda \left[1 - \left| \frac{\sin c(\pi)}{\sin c(\pi/A_1)} \right| \right] = 1 + \lambda \tag{61}$$

Hence, the amplitudes of the high frequencies $\omega_1 = 2\pi n/A_1$, divisible to $n = 1, 2, \dots$, are accelerated $(1 + \lambda)$ times towards those of the low frequencies, $\omega_1 = 0$. Similar properties have the adaptive unsharp masking 1D filters, whose AFRs are $M_g(\omega_2)$ and $M_g(\omega_3)$, correspondingly.

4.4. Implementation of the Recursive Unsharp Masking 3D Filter through Cascade Recursive Unsharp Masking 1D Filters

For further simplification, it is possible, instead of applying one 3D filter, to use three 1D filters: the first one in the horizontal direction, the second in the vertical direction, and the third in the lateral direction, represented by the relations below:

- for the first line of the image in directions x/y/z respectively, we use the relations:

$$y(i, r, k) = x(i, r, k) + \lambda_1 [x(i, r, k) - (1/A_1) \sum_{s=-a}^a x(i + s, r, k)], \quad (62)$$

$$z(i, r, k) = y(i, r, k) + \lambda_2 [y(i, r, k) - (1/A_2) \sum_{m=-b}^b y(i, r + m, k)], \quad (63)$$

$$g(i, r, k) = z(i, r, k) + \lambda_3 [z(i, r, k) - (1/A_3) \sum_{n=-h}^h z(i, r, k + n)], \quad (64)$$

where $\lambda = \lambda_1 \times \lambda_2 \times \lambda_3$, $y(i, r, k)$ and $q(i, r, k)$ represent the voxel (i, r, k) of the intermediate images, obtained as a result of the first, second and third filtration. By analogy with Equation (4), for the remaining lines of the image, Equations (62)–(64) are presented in a recursive form:

$$\begin{aligned} y(i, r, k) &= y(i-1, r, k) + (1 + \lambda_1) [x(i, r, k) - x(i-1, r, k)] + \\ &\quad + (\lambda_1/A_1) [x(i+a, r, k) - x(i-a-1, r, k)]; \\ q(i, r, k) &= q(i, r-1, k) + (1 + \lambda_2) [y(i, r, k) - y(i, r-1, k)] + \\ &\quad + (\lambda_2/A_2) [y(i, r+b, k) - y(i, r-b-1, k)]; \\ g(i, r, k) &= g(i, r, k-1) + (1 + \lambda_3) [q(i, r, k) - q(i, r, k-1)] + \\ &\quad + (\lambda_3/A_3) [q(i, r, k+h) - q(i, r, k-h-1)]. \end{aligned} \quad (65)$$

If $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_0 = \eta = \sqrt[3]{\lambda}$, then the system of Equation (65) turns into:

$$\begin{aligned} y(i, r, k) &= y(i-1, r, k) + (1 + \eta) [x(i, r, k) - x(i-1, r, k)] + \frac{\eta}{A_1} [x(i+a, r, k) - x(i-a-1, r, k)]; \\ q(i, r, k) &= q(i, r-1, k) + (1 + \eta) [y(i, r, k) - y(i, r-1, k)] + \frac{\eta}{A_2} [y(i, r+b, k) - y(i, r-b-1, k)]; \\ g(i, r, k) &= g(i, r, k-1) + (1 + \eta) [q(i, r, k) - q(i, r, k-1)] + \frac{\eta}{A_3} [q(i, r, k+h) - q(i, r, k-h-1)]. \end{aligned} \quad (66)$$

The system of Equation (66) replaces the recursive relations in Equation (57).

4.5. Computational Complexity of the Recursive Sliding Adaptive Unsharp Masking 3D Filter

For the CC evaluation of the recursive sliding adaptive unsharp masking 3D filter, the number of executed arithmetic operations should be defined.

The number of terms in Equation (57) is always 25, and does not depend on the filter window size $(2a+1) \times (2b+1) \times (2h+1)$. The reduced number of arithmetic operations is defined by the following relation:

$$R_6 = (1/24) [(2a+1)(2b+1)(2h+1) + 2] \quad (67)$$

Example 4: if $a = b = h = 5$ (i.e., for a 3D window of size $11 \times 11 \times 11$), then the reduction coefficient is $R_6 = 55.54$, i.e., the number of operations, executed by the recursive filter, is more than 55 times lower. In Figure 5 we show the relation between R_6 and the cubic window size, for $a = b = h = c$.

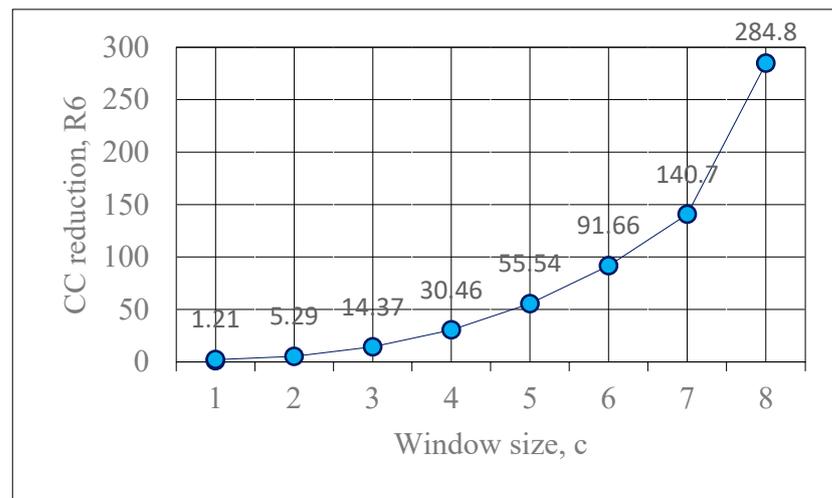


Figure 5. CC reduction of the recursive sliding adaptive unsharp masking 3D filter, as a function of the cubic window size.

For the case when three cascade recursive unsharp masking 1D filters are used, the reduced number of arithmetic operations R_7 for each voxel, in correspondence with Equation (66), is:

$$R_7 = \frac{2a + 4}{6} + \frac{2b + 4}{6} + \frac{2h + 4}{6} = (1/3)(a + b + h) + 2. \quad (68)$$

Example 5: for $a = b = h = 5$ (3D cubic window of size $11 \times 11 \times 11$) we obtain $R_7 = 7$, i.e., as a result of the recursive approach, the number of operations is 7 times lower. The algorithm efficiency is additionally enhanced after triple use of the 1D filter.

The comparison between the CC of the recursive filtration through one 3D filter and that calculated for the case when three cascade 1D filters of the same parameters ($a = b = h = c$) are used, gives the result below:

$$R_6/R_7 = [(2c + 1)^3 + 2]/24(c + 2). \quad (69)$$

In this example, for $c = 5$ we obtain $R_6/R_7 = 7.93$.

5. Discussion and Conclusions

In this work, two basic algorithms for the locally-adaptive processing of tensor 3D images are presented, based on the use of the recursive SM3DF with a central symmetric 3D kernel: for adaptive noise filtration, and for sharpness enhancement. The main characteristics (spatial and frequency) of the algorithms are analyzed for when a progressive linear image scan is used. The CC of each of the analyzed algorithms is also evaluated.

As it is known, the famous recursive and non-recursive filters have their advantages and shortcomings. For example, recursive filters have higher computational efficiency, and offer easier usage, but the software implementation is more complicated and additional buffer is needed to maintain the recursive coefficients' values. In addition to this, the non-recursive approach does not use feedback, has lower sensitivity, and the delay in providing a response is greater and (depending on the application) needs more processing power [21]. In brief, depending on the objectives, recursive or non-recursive filters are preferred. Their use for multidimensional visual information processing implies a more sophisticated approach. As a result of the analysis presented in this research, it follows that:

- The obtained results confirm that the recursive and the non-recursive locally-adaptive filtering presented in our approach are practically equivalent in respect of the calculated mean output voxel value;
- The SM3DF filter does not introduce phase distortions;

- The recursive 3D filter could be replaced by three identical recursive 1D filters, sequentially bonded together, which use simplified algorithms for the processing in horizontal, vertical and lateral directions. In this case, however, three scans of the input tensor are needed, instead of one;
- The comparison between the recursive and the non-recursive filtration shows that the recursive filtration, implemented as three sequential 1D filters, reduces the CC significantly. Moreover, in the case of non-recursive processing, the filter window size is limited by the horizontal size of the tensor, which is an advantage, when compared to the recursive filtration;
- Some considerations exist about the filter window size. The minimum possible size is $3 \times 3 \times 3$, and it offers a small CC reduction, but the algorithm is really efficient for a larger window size. As it is shown in the examples, the CC reduction is very high for windows of size 5 or more. The relation is shown in Example 3, where the CC is reduced by approximately 50 times, for a window of size $11 \times 11 \times 11$. Naturally, the window size should be selected in accordance with the image dimensions, and for very large pictures (for example, satellite images of size $10,000 \times 10,000$ or more) the window could be increased correspondingly, which will give better results;
- The recursive approach needs additional memory to save the recursive coefficients, but it is incomparably smaller than the buffer, which is needed for image processing;
- The use of the analyzed recursive 3D algorithms for locally-adaptive processing of tensor images results in significant CC reduction, when compared to similar non-recursive algorithms' performance;
- The filtration CC in the tensor frequency space, evaluated through the number of basic operations needed, is twice as low as that needed for the voxel space.
- The trends of future work will be aimed at the development of new recursive algorithms, with application to various areas, mainly focused on:
 - Adaptation of the created new algorithms to the casual area shape, by using other kinds of 3D scan for the tensor voxels, such as, for example, the recursive scans of Hilbert, Peano, Morton and Sierpinski [22], etc. Compared to the linear, the recursive scans in the three spatial orthogonal directions retain better the existing correlation between neighboring voxels, which enhances the tensor processing efficiency;
 - Application of the recursive SM3DF in the 3D CNN, where locally-adaptive 3D filtering is performed in the network layers. As a result of the CC reduction in the 3D filtration, new capabilities are available for the efficiency enhancement of the tensor CNN;
 - Creation of recursive algorithms for locally-adaptive non-linear 3D filtration of tensor images;
 - Software implementation of the algorithm;
 - Creation of recursive algorithms for parallel processing of multidimensional signals, appropriate for hardware implementation, etc.

Author Contributions: Conceptualization, R.K. (Roumen Kountchev) and R.M.; methodology, R.K. (Roumen Kountchev); software, R.M.; validation, R.K. (Roumen Kountchev), R.M. and R.K. (Roumiana Kountcheva); formal analysis, R.K. (Roumen Kountchev) and R.M.; investigation, R.M.; resources, R.K. (Roumen Kountchev), R.M. and R.K. (Roumiana Kountcheva); data curation, R.K. (Roumen Kountchev) and R.M.; writing—original draft preparation, R.K. (Roumen Kountchev); writing—review and editing, R.M. and R.K. (Roumiana Kountcheva); visualization, R.K. (Roumen Kountchev), R.M. and R.K. (Roumiana Kountcheva); supervision, R.M.; project administration, R.M.; funding acquisition, R.M. 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.

References

1. Gonzalez, R.; Woods, R. *Digital Image Processing*, 4th ed.; Pearson: New York, NY, USA, 2018.
2. Pratt, W. *Introduction to Digital Image Processing*; CRC Press: Boca Raton, FL, USA, 2014.
3. Reed, T. *Digital Image Sequence Processing, Compression, and Analysis*; CRC Press: Boca Raton, FL, USA, 2004. [[CrossRef](#)]
4. Ali, R.; Hardie, R. Recursive Non-Local Means Filter for Video Denoising. *EURASIP J. Image Video Process.* **2017**, *2017*, 29. [[CrossRef](#)]
5. Erkan, U.; Enginoglu, S.; Thanh, D. A Recursive Mean Filter for Image Denoising, Intern. In Proceedings of the Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 21–22 September 2019. [[CrossRef](#)]
6. Tran, M. 3D Image Analysis with Variational Methods and Wavelets: Applications to Medical Image Processing. Ph.D. Thesis, Université d'Orléans, Orléans, France, 2012.
7. Woods, J. *Multidimensional Signal, Image, and Video Processing and Coding*, 2nd ed.; Academic Press: Cambridge, MA, USA; Elsevier: Amsterdam, The Netherlands, 2012.
8. Jin, F.; Fieguth, P.; Winger, L.; Jernigan, E. Adaptive Wiener Filtering of Noisy Images and Image Sequences. In Proceedings of the International Conference on Image Processing, Barcelona, Spain, 14–17 September 2003. [[CrossRef](#)]
9. 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](#)]
10. Kamnitsas, K.; Ledig, C.; Newcombe, V.; Simpson, J.; Kane, A.; Menon, D.; Rueckert, D.; Glocker, B. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Med. Image Anal.* **2017**, *36*, 61–78. [[CrossRef](#)] [[PubMed](#)]
11. Subhojit, S.; Chowdhury, S.; Laha, S.; Dey, D. Use of Non-local Means Filter to Denoise Image Corrupted by Salt and Pepper Noise. *Signal Image Process. Int. J. SIPIJ* **2012**, *3*, 223–235.
12. Coupe, P.; Manjon, J.; Robles, M.; Collins, L. Adaptive Multiresolution Non-Local Means Filter for 3D MR Image Denoising, IET Image Processing. *Inst. Eng. Technol.* 2011, preprint.
13. Buades, A.; Coll, B.; Morel, J.M. A non-local algorithm for image denoising. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005. [[CrossRef](#)]
14. Westin, C.F.; Knutsson, H.; Kikinis, R. Adaptive Image Filtering. In *Handbook of Medical Imaging*; Bankman, I., Ed.; Academic Press: Cambridge, MA, USA, 2000; 22p.
15. Mirmozaffari, M. Filtering in image processing. *ENG Trans.* **2020**, *1*, 1–5.
16. Li, Z.; Yu, L.; Trzasko, J.D.; Lake, D.; Blezek, D.; Fletcher, J.; McCollough, C.; Manduca, A. Adaptive nonlocal means filtering based on local noise level for CT denoising. *Med. Phys.* **2014**, *41*, 011908. [[CrossRef](#)] [[PubMed](#)]
17. Golilarz, N.; Addeh, A.; Gao, H.; Ali, L.; Roshandeh, A.M.; Munir, H.M.; Khan, R. A new automatic method for control chart patterns recognition based on ConvNet and Harris Hawks meta heuristic optimization algorithm. *IEEE Access* **2019**, *7*, 149398–149405. [[CrossRef](#)]
18. Rakshit, S.; Ghosh, A.; Shankar, B.U. Fast mean filtering technique (FMFT). *Pattern Recognit.* **2007**, *40*, 890–897. [[CrossRef](#)]
19. Nakariyakul, S. Fast spatial averaging: An efficient algorithm for 2D mean filtering. *J. Supercomput.* **2013**, *65*, 262–273. [[CrossRef](#)]
20. Palani, S. The z-Transform Analysis of Discrete Time Signals and Systems. In *Signals and Systems*; Palani, S., Ed.; Springer: Cham, Switzerland, 2021; pp. 921–1055. [[CrossRef](#)]
21. Abbas, M.; Gustafsson, O.; Wanhammar, L. Power Estimation of Recursive and Non-Recursive CIC Filters Implemented in Deep-Submicron Technology. In Proceedings of the 2010 International Conference on Green Circuits and Systems, Shanghai, China, 21–23 June 2010; pp. 221–225.
22. Jin, G.; Crummey, J. SFCCGen: A framework for efficient generation of multi-dimensional space filling curves by recursion. *ACM Trans. Math. Softw.* **2005**, *31*, 120–148. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.