# Discrete Pseudo-Fractional Fourier Transform and Its Fast Algorithm 

Dorota Majorkowska-Mech *, ${ }^{\text {(D) }}$ and Aleksandr Cariow ${ }^{\text {+(D) }}$

Faculty of Computer Science and Information Technology, West Pomeranian University of Technology in Szczecin, Zolnierska 49, 71-210 Szczecin, Poland; acariow@wi.zut.edu.pl

* Correspondence: dmajorkowska@wi.zut.edu.pl
+ These authors contributed equally to this work

Citation: Majorkowska-Mech, D.; Cariow, A. Discrete Pseudo-Fractional Fourier Transform and Its Fast

Algorithm. Electronics 2021, 10, 2145. https:/ /doi.org/10.3390/electronics 10172145

Academic Editor: Bhanu Prakash KN

Received: 11 August 2021
Accepted: 1 September 2021
Published: 3 September 2021

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

Copyright: © 2021 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/).


#### Abstract

In this article, we introduce a new discrete fractional transform for data sequences whose size is a composite number. The main kernels of the introduced transform are small-size discrete fractional Fourier transforms. Since the introduced transformation is not, in the generally known sense, a classical discrete fractional transform, we call it discrete pseudo-fractional Fourier transform. We also provide a generalization of this new transform, which depends on many fractional parameters. A fast algorithm for computing the introduced transform is developed and described.


Keywords: signal and image processing; discrete fractional Fourier transform; Kronecker product

## 1. Introduction

Discrete fractional transforms are the generalizations of the conventional discrete transforms with an additional fractional parameter. They have been introduced in the 1990s. Noteworthy discrete fractional transforms, such as the discrete fractional Fourier transform (DFrFT) [1-3], discrete fractional Hartley transform [4], discrete fractional cosine and sine transforms [5], discrete fractional Hadamard transform (DFrHT) [6] have been defined. They are widely used in various fields of science and technology, including image representation and compression [7], image encryption [8], digital watermarking [9], adaptive filtering $[10,11]$ and others. Among these transforms, the DFrFT has been applied in most practical contexts. To date, several effective algorithms have been developed to implement this transform [5,12,13]. In [14], the DFrFT was generalized, so that it depends on many fractional parameters. It was named the multiple-parameter discrete fractional Fourier transform (MPDFrFT). In our previous work [12], we have proposed a fast algorithm for the DFrFT and MPDFrFT. However, even this algorithm was quite computationally intensive. Therefore, our goal was to further reduce the complexity of calculations while preserving the properties of fractionality. In this article, we present a new discrete pseudo-fractional Fourier transform and its multiple-parameter version as well as a fast algorithm to calculate them. A similar approach was proposed in our previous work [15] but concerned with the use of DFrHT kernels. As a result of this approach, a new orthogonal base was obtained and a fast algorithm for discrete transform in this base was synthesized. This transform can be implemented using small-size discrete fractional Fourier transforms, for which algorithms were described in [13]. The algorithm of the proposed transform has a parallel, simple and modular structure, which makes it convenient for hardware implementation.

## 2. Mathematical Background

The normalized discrete Fourier transform (DFT) matrix of order $N$ is defined as follows:

$$
\mathbf{F}_{N}=\frac{1}{\sqrt{N}}\left[\begin{array}{ccccc}
1 & 1 & \ldots & 1 & 1  \tag{1}\\
1 & w_{N}^{1} & \ldots & w_{N}^{N-2} & w_{N}^{N-1} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
1 & w_{N}^{N-2} & \ldots & w_{N}^{(N-2)^{2}} & w_{N}^{(N-2)(N-1)} \\
1 & w_{N}^{N-1} & \ldots & w_{N}^{(N-1)(N-2)} & w_{N}^{(N-1)^{2}}
\end{array}\right]
$$

where $w_{N}=e^{-j \frac{2 \pi}{N}}, j$ is the imaginary unit, and $1 / \sqrt{N}$ is the normalizing factor. Since $\mathbf{F}_{N}$ is a unitary and symmetric matrix, it can be diagonalized, i.e., represented as a product [3]

$$
\begin{equation*}
\mathbf{F}_{N}=\mathbf{Z}_{N} \boldsymbol{\Lambda}_{N} \mathbf{Z}_{N}^{-1} \tag{2}
\end{equation*}
$$

where $\boldsymbol{\Lambda}_{N}$ is an $N \times N$ diagonal matrix, whose diagonal entries are the eigenvalues of $\mathbf{F}_{N} . \mathbf{Z}_{N}$ is the matrix whose columns are normalized mutually orthogonal eigenvectors of the matrix $\mathbf{F}_{N}$ (hence $\mathbf{Z}_{N}^{-1}=\mathbf{Z}_{N}^{T}$ ). The eigenvector $\mathbf{z}_{N}^{(k)}$ corresponds to the eigenvalue $\lambda_{k}$. The matrix $\mathbf{F}_{N}$ has only four different eigenvalues: $1,-1, j,-j$ and, for $N \geq 4$, the eigenvalues are degenerated and the set of eigenvectors is not unique [3]. For this reason, a particular eigenvector set should be specified.

The discrete fractional Fourier transform (DFrFT) matrix is a power of the DFT matrix with one real parameter $a$ in the exponent. This parameter is usually a fraction, hence the term fractional. The DFrFT matrix could be calculated from the eigenvalue decomposition of DFT matrix [3]

$$
\begin{equation*}
\mathbf{F}_{N}^{a}=\mathbf{Z}_{N} \boldsymbol{\Lambda}_{N}^{a} \mathbf{Z}_{N}^{T} \tag{3}
\end{equation*}
$$

For $a=0$, the DFrFT matrix is equal to the identity matrix, and, for $a=1$, it becomes the ordinary DFT matrix. The definition (3) of DFrFT was first introduced by Pei and Yeh [1,2]. They defined the DFrFT in terms of a particular set of eigenvectors, namely the discrete counterpart of the set of Hermite-Gaussian functions. In our article, we assume that the set of eigenvectors of the DFT matrix and their ordering were determined according to the procedure described in [3]. These eigenvectors are not determined in an analytic form. They can be calculated using numerical methods.

## 3. Definition of a New Transform

To determine a new transform matrix, we firstly represent the order of the transform matrix $N$, which is assumed to be a composite number, as a product of positive integers $N=N_{1}, N_{2}, \ldots, N_{k}$, where $k \geq 2$. The numbers $N_{1}, N_{2}, \ldots, N_{K}$ may but need not to be prime numbers.

We define the fractional-like Fourier transform matrix in the following way:

$$
\begin{equation*}
\mathbf{F}_{N}^{\left(a, N_{1}, N_{2}, \ldots, N_{K}\right)}=\mathbf{F}_{N_{K}}^{a} \otimes \mathbf{F}_{N_{K-1}}^{a} \otimes \ldots \otimes \mathbf{F}_{N_{1}}^{a} \tag{4}
\end{equation*}
$$

where $\mathbf{F}_{N_{k}}^{a}, k=1,2, \ldots, K$, are the discrete fractional Fourier transform matrices with fractional parameter $a$, which are determined according to Equation (3), and $\otimes$ means the Kronecker product of matrices operation [16]. The definition (4) of the matrix $\mathbf{F}^{\left(a, N_{1}, N_{2}, \ldots, N_{K}\right)}$ depends on the way how $N$ is decomposed into the product of integers. Now, we will show that the matrix defined in this way satisfies most of the requirements imposed on the fractional transform matrix, i.e., the unitarity and index additivity. However, it is not reducible to the usual Fourier transform matrix when the fractional parameter $a$ is equal to one. At the beginning, we prove unitarity of the matrix $\mathbf{F}_{N}^{\left(a, N_{1}, N_{2}, \ldots, N_{K}\right)}$. The matrix $\mathbf{F}_{N_{k}}^{a}$ as a matrix of discrete fractional Fourier transform satisfies

$$
\begin{equation*}
\left(\mathbf{F}_{N_{k}}^{a}\right)^{-1}=\left(\mathbf{F}_{N_{k}}^{a}\right)^{*}=\mathbf{F}_{N_{k}}^{-a} \tag{5}
\end{equation*}
$$

where * means conjugate transpose of a matrix.

For the Kronecker product of any invertible matrices $\mathbf{A}$ and $\mathbf{B}$, the following equality is true [16]:

$$
\begin{equation*}
(\mathbf{A} \otimes \mathbf{B})^{-1}=\mathbf{A}^{-1} \otimes \mathbf{B}^{-1} \tag{6}
\end{equation*}
$$

Therefore, the matrix inverse to $\mathbf{F}_{N}^{\left(a, N_{1}, N_{2}, \ldots, N_{K}\right)}$ has the form

$$
\begin{equation*}
\left(\mathbf{F}_{N}^{\left(a, N_{1}, N_{2}, \ldots, N_{K}\right)}\right)^{-1}=\mathbf{F}_{N_{K}}^{-a} \otimes \mathbf{F}_{N_{K-1}}^{-a} \otimes \ldots \otimes \mathbf{F}_{N_{1}}^{-a}=\mathbf{F}_{N}^{\left(-a, N_{1}, N_{2}, \ldots, N_{K}\right)} \tag{7}
\end{equation*}
$$

In case of proving unitarity of the matrix $\mathbf{F}_{N}^{\left(a, N_{1}, N_{2}, \ldots, N_{K}\right)}$, the following property of the Kronecker product of any matrices $\mathbf{A}$ and $\mathbf{B}$ should be used [16]:

$$
\begin{equation*}
(\mathbf{A} \otimes \mathbf{B})^{*}=\mathbf{A}^{*} \otimes \mathbf{B}^{*} \tag{8}
\end{equation*}
$$

Using this property and Equations (5) and (7), we may write

$$
\begin{gathered}
\left(\mathbf{F}_{N}^{\left(a, N_{1}, N_{2}, \ldots, N_{K}\right)}\right)^{*}=\left(\mathbf{F}_{N_{K}}^{a} \otimes \mathbf{F}_{N_{K-1}}^{a} \otimes \ldots \otimes \mathbf{F}_{N_{1}}^{a}\right)^{*} \\
=\left(\mathbf{F}_{N_{K}}^{a}\right)^{*} \otimes\left(\mathbf{F}_{N_{K-1}}^{a}\right)^{*} \otimes \ldots \otimes\left(\mathbf{F}_{N_{1}}^{a}\right)^{*} \\
=\mathbf{F}_{N_{K}}^{-a} \otimes \mathbf{F}_{N_{K-1}}^{-a} \otimes \ldots \otimes \mathbf{F}_{N_{1}}^{-a}=\left(\mathbf{F}_{N}^{\left(a, N_{1}, N_{2}, \ldots, N_{K}\right)}\right)^{-1}
\end{gathered}
$$

Hence, the matrix $\mathbf{F}_{N}^{\left(a, N_{1}, N_{2}, \ldots, N_{K}\right)}$ is unitary.
The next requirement for a fractional transform is index additivity, which means that for any real fractional parameters $a$ and $b$, the result of two successive transforms with parameters $a$ and $b$ is the same as the result of one transform with parameter $a+b$. To demonstrate that the defined transform matrix (4) satisfies this property, we can use the appropriate property of the matrix $\mathbf{F}_{N_{k}}^{a}$ again. This matrix, being a matrix of discrete fractional transform, fulfills

$$
\begin{equation*}
\mathbf{F}_{N_{k}}^{b} \mathbf{F}_{N_{k}}^{a}=\mathbf{F}_{N_{k}}^{a+b} \tag{9}
\end{equation*}
$$

We will use this fact and the next property of the Kronecker product, which is true for matrices A, B, C and D of such sizes that the matrix products AC and BD exist [16]

$$
\begin{equation*}
(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D})=(\mathbf{A C}) \otimes(\mathbf{B D}) \tag{10}
\end{equation*}
$$

We obtain

$$
\begin{gathered}
\mathbf{F}_{N}^{\left(b, N_{1}, N_{2}, \ldots, N_{K}\right)} \mathbf{F}_{N}^{\left(a, N_{1}, N_{2}, \ldots, N_{K}\right)} \\
=\left(\mathbf{F}_{N_{K}}^{b} \otimes \mathbf{F}_{N_{K-1}}^{b} \otimes \ldots \otimes \mathbf{F}_{N_{1}}^{b}\right)\left(\mathbf{F}_{N_{K}}^{a} \otimes \mathbf{F}_{N_{K-1}}^{a} \otimes \ldots \otimes \mathbf{F}_{N_{1}}^{a}\right) \\
=\left(\mathbf{F}_{N_{K}}^{b} \mathbf{F}_{N_{K}}^{a}\right) \otimes\left(\mathbf{F}_{N_{K-1}}^{b} \mathbf{F}_{N_{K-1}}^{a}\right) \otimes \ldots \otimes\left(\mathbf{F}_{N_{1}}^{b} \mathbf{F}_{N_{1}}^{a}\right) \\
=\mathbf{F}_{N_{K}}^{a+b} \otimes \mathbf{F}_{N_{K-1}}^{a+b} \otimes \ldots \otimes \mathbf{F}_{N_{1}}^{a+b}=\mathbf{F}_{N}^{\left(a+b, N_{1}, N_{2}, \ldots, N_{K}\right)}
\end{gathered}
$$

Moreover, the defined matrix (4) is also symmetric, i.e.,

$$
\begin{equation*}
\mathbf{F}_{N}^{\left(a, N_{1}, N_{2}, \ldots, N_{K}\right)}=\left(\mathbf{F}_{N}^{\left(a, N_{1}, N_{2}, \ldots, N_{K}\right)}\right)^{T} \tag{11}
\end{equation*}
$$

Its symmetry is caused by symmetry of the matrix $\mathbf{F}_{N_{k}}^{a}$.
Taking into account the above considerations, we can conclude that the defined matrix has many properties in common with the DFrFT matrix. However, since this matrix is not equal to the DFrFT matrix and does not fulfill all the requirements for fractional transforms, we call it the discrete pseudo-fractional Fourier transform matrix and the transform determined by it-the discrete pseudo-fractional Fourier transform (DPFrFT).

## 4. Visualization of the DPFrFT

The DFrFT was specified for the input data vector, however, it is often used in image processing. We chose this type of input data because it is suitable for transform visualization. The two-dimensional transform of an image consists in determining one-dimensional transforms of all columns and all rows of the matrix representing this image. In matrix notation, it can be written as follows:

$$
\begin{equation*}
\mathbf{Y}_{N \times M}=\mathbf{F}_{N} \mathbf{X}_{N \times M} \mathbf{F}_{M}^{T} \tag{12}
\end{equation*}
$$

where $\mathbf{X}_{N \times M}$ is the matrix representing an input image, $\mathbf{F}_{N}$ and $\mathbf{F}_{M}$-matrices of onedimensional transforms, which operate on columns and rows of the input matrix, respectively, and $\mathbf{Y}_{N \times M}$ is the output matrix.

Since we chose the DFrFT as the starting point, we will compare the results of DFrFT and DPFrFT transforms obtained for some images. The calculations and visualizations have been made in the Matlab R2020b environment. Two well-known testing images"Lena" and "Boat", of size $512 \times 512$ pixels, were chosen to compare the DFrFT and DPFrFT transforms. In the case of DPFrFT, the number $N=512$ was decomposed into the product $N_{1} N_{2} N_{3}=16 \times 8 \times 4$, so according to Equation (4), $\mathbf{F}_{512}^{(a, 16,8,4)}=\mathbf{F}_{4}^{a} \otimes \mathbf{F}_{8}^{a} \otimes \ldots \otimes \mathbf{F}_{16}^{a}$. For the input $512 \times 512$ matrix $\mathbf{X}_{512}$, the output $\mathbf{Y}_{512}^{a}$ of DFrFT is calculated according to Equation (12)

$$
\begin{equation*}
\mathbf{Y}_{512}^{a}=\mathbf{F}_{512}^{a} \mathbf{X}_{512}\left(\mathbf{F}_{512}^{a}\right)^{T} \tag{13}
\end{equation*}
$$

Analogously, the output of DPFrFT is calculated as follows:

$$
\begin{equation*}
\mathbf{Y}_{512}^{(a, 16,8,4)}=\mathbf{F}_{512}^{(a, 16,8,4)} \mathbf{X}_{512}\left(\mathbf{F}_{512}^{(a, 16,8,4)}\right)^{T} \tag{14}
\end{equation*}
$$

Before calculating each transform, the grayscale image, represented by the matrix $\mathbf{X}_{512}$, is converted from the type uint8 into the type double, with values in the range $[0,1]$, using the Matlab function im2double. Then, the transforms were performed on the rows and columns of the input matrix, according to Equations (13) and (14), respectively. Since the results are complex matrices, the modulus, the real part and the imaginary part of each output will be presented. In order to visualize the outputs (the modulus, the real part and the imaginary part), they should also be scaled to the range [0, 1]. Logarithmic scaling was performed as follows:

$$
\begin{equation*}
\mathbf{Y}_{\text {scaled }}=\frac{\log _{2}(1+|\mathbf{Y}|)}{\log _{2}(1+\max (|\mathbf{Y}|))} \tag{15}
\end{equation*}
$$

where $\mathbf{Y}$ can be the modulus, the real part, or the imaginary part of the output matrix of each transform. $\mathbf{Y}_{\text {scaled }}$ is the matrix $\mathbf{Y}$ scaled to the range $[0,1] .|\cdot|$ denotes the absolute value and max is the largest value in the matrix. In the end, each scaled matrix is converted from the type double into the type uint8, using the Matlab function im2uint8, and visualized using the Matlab function imshow. Figures 1-8 show the obtained images for the four values of the fractional parameter $a$. In each figure, at the top, there is the original image. In the middle, from left to right, there are the images of the modulus, the real part and the imaginary part of its DFrFT, respectively. Analogously, in the bottom, there are the modulus, the real part and the imaginary part of DPFrFT. It is easy to notice that an increase in the fractional parameter $a$, both in the case of DFrFT and DPFrFT, makes the obtained images less and less similar to the original image. In other words, they are becoming more and more difficult to recognize. For the same value of the fractional parameter $a$ in DPFrFT images, the original image is more difficult to recognize than in DFrFT images, which may indicate that DPFrFT is more useful for hiding/encrypting images than DFrFT. Furthermore, it can be seen that in the DFrFT images, the large-scale elements of the original image are more visible, while in the case of DPFrFT, the details are more visible.


Figure 1. Comparison of DFrFT and DPFrFT of the grayscale image "Lena" for fractional parameter $a=0.1$.


$$
\text { fractional parameter } 0.1
$$


pseudo modulus

pseudo real

pseudo imag


Figure 2. Comparison of DFrFT and DPFrFT of the grayscale image "Boat" for fractional parameter $a=0.1$.

pseudo modulus

pseudo real


pseudo imag


Figure 3. Comparison of DFrFT and DPFrFT of the grayscale image "Lena" for fractional parameter $a=0.2$.


$$
\text { fractional parameter } 0.2
$$


pseudo modulus

pseudo real

pseudo imag


Figure 4. Comparison of DFrFT and DPFrFT of the grayscale image "Boat" for fractional parameter $a=0.2$.


$$
\text { fractional parameter } 0.3
$$



Figure 5. Comparison of DFrFT and DPFrFT of the grayscale image "Lena" for fractional parameter $a=0.3$.


$$
\text { fractional parameter } 0.3
$$


frac imaginary

pseudo modulus
pseudo real

pseudo imag


Figure 6. Comparison of DFrFT and DPFrFT of the grayscale image "Boat" for fractional parameter $a=0.3$.


## fractional parameter 0.4


pseudo modulus

pseudo real


pseudo imag


Figure 7. Comparison of DFrFT and DPFrFT of the grayscale image "Lena" for fractional parameter $a=0.4$.

fractional parameter 0.4

pseudo modulus

pseudo real


pseudo imag


Figure 8. Comparison of DFrFT and DPFrFT of the grayscale image "Boat" for fractional parameter $a=0.4$.

## 5. Multiple-Parameter Discrete Pseudo-Fractional Fourier Transform

There are applications of DFrFT, such as image encryption, in which it is more preferable when the transform has many fractional parameters [14]. Such a transform gives more degrees of freedom when the signal is represented in the MPDFrFT domain, which makes its correct decryption more difficult. It is also possible to generalize in this way the DPFrFT matrix (4). Let us define the matrix

$$
\begin{equation*}
\mathbf{F}_{N}^{\left(a_{1}, a_{2}, \ldots, a_{K}, N_{1}, N_{2}, \ldots, N_{K}\right)}=\mathbf{F}_{N_{K}}^{a_{K}} \otimes \mathbf{F}_{N_{K-1}}^{a_{K-1}} \otimes \ldots \otimes \mathbf{F}_{N_{1}}^{a_{1}} \tag{16}
\end{equation*}
$$

where $a_{1}, a_{2}, \ldots, a_{K}$ are some real fractional parameters. The transform to which the matrix $\mathbf{F}_{N}^{\left(a_{1}, a_{2}, \ldots, a_{K}, N_{1}, N_{2}, \ldots, N_{K}\right)}$ corresponds we call the multiple-parameter discrete pseudofractional Fourier transform (MPDPFrFT) and its output vector $\mathbf{y}_{N}^{\left(a_{1}, a_{2}, \ldots, a_{K}, N_{1}, N_{2}, \ldots, N_{K}\right)}$ for the input vector $\mathbf{x}_{N}$ is set out below

$$
\begin{equation*}
\mathbf{y}_{N}^{\left(a_{1}, a_{2}, \ldots, a_{K}, N_{1}, N_{2}, \ldots, N_{K}\right)}=\mathbf{F}_{N}^{\left(a_{1}, a_{2}, \ldots, a_{K}, N_{1}, N_{2}, \ldots, N_{K}\right)} \mathbf{x}_{N} \tag{17}
\end{equation*}
$$

The matrix $\mathbf{F}_{N}^{\left(a_{1}, a_{2}, \ldots, a_{K}, N_{1}, N_{2}, \ldots, N_{K}\right)}$ is invertible and its inverse matrix is equal to

$$
\begin{gather*}
\left(\mathbf{F}_{N}^{\left(a_{1}, a_{2}, \ldots, a_{K}, N_{1}, N_{2}, \ldots, N_{K}\right)}\right)^{-1}=\mathbf{F}_{N_{K}}^{-a_{K}} \otimes \mathbf{F}_{N_{K-1}}^{-a_{K-1}} \otimes \ldots \otimes \mathbf{F}_{N_{1}}^{-a_{1}} \\
=\mathbf{F}_{N}^{\left(-a_{1},-a_{2}, \ldots,-a_{K}, N_{1}, N_{2}, \ldots, N_{K}\right)} \tag{18}
\end{gather*}
$$

As in the case of the DPFrFT matrix, it can be easily shown that the MPDPFrFT matrix is unitary and satisfies index additivity. When all fractional parameters are equal to each other then the MPDPFrFT reduces to ordinary DPFrFT, that is,

$$
\begin{equation*}
\mathbf{F}_{N}^{\left(a_{1}, a_{2}, \ldots, a_{K}, N_{1}, N_{2}, \ldots, N_{K}\right)}=\mathbf{F}_{N}^{\left(a, N_{1}, N_{2}, \ldots, N_{K}\right)} \tag{19}
\end{equation*}
$$

for $a_{1}=a_{2}=\ldots=a_{K}=a$.

## 6. Fast Algorithm for the Multiple-Parameter Discrete Pseudo-Fractional Fourier Transform

Only the algorithm for MPDPFrFT will be presented because an algorithm for the ordinary DPFrFT is its special case in which all fractional parameters are equal to each other.

We use the Fundamental Tensor-Product Factorization Theorem, which was proven in [17]. It states that the tensor product of any square matrices $A_{n_{1}}, A_{n_{2}}, \ldots, A_{n_{t}}$, where $A_{n_{i}}$ is an $n_{i} \times n_{i}$ matrix, can be factorized as follows:

$$
\begin{equation*}
\mathbf{A}_{n_{1}} \otimes \mathbf{A}_{n_{2}} \otimes \ldots \otimes \mathbf{A}_{n_{t}}=\prod_{i=1}^{t}\left(\mathbf{I}_{N(i-1)} \otimes \mathbf{A}_{n_{i}} \otimes \mathbf{I}_{\frac{N}{N(i)}}\right) \tag{20}
\end{equation*}
$$

Moreover, the factorization is correct for any permutation of the factors $\left(\mathbf{I}_{N(i-1)} \otimes\right.$ $\left.\mathbf{A}_{n_{i}} \otimes \mathbf{I}_{\frac{N}{N(i)}}\right)$. In this equation, $I_{n}$ is the identity matrix of order $n, N(i)=n_{1} n_{2} \ldots n_{i}$, $N(0)=1$ and $N=N(t)$, where $t$ denotes the number of factors.

Using the mentioned theorem to the tensor product in Equation (16) and noticing that in this case $t=K$ and $N=N(K)$, we obtain

$$
\begin{equation*}
\mathbf{F}_{N_{K}}^{a_{K}} \otimes \mathbf{F}_{N_{K-1}}^{a_{K-1}} \otimes \ldots \otimes \mathbf{F}_{N_{1}}^{a_{1}}=\prod_{i=1}^{K}\left(\mathbf{I}_{N(i-1)} \otimes \mathbf{F}_{N_{K-i+1}}^{a_{K-i+1}} \otimes \mathbf{I}_{\frac{N}{N(i)}}\right) \tag{21}
\end{equation*}
$$

If we change the order of the factors from the last to the first, which we can do according to the mentioned theorem, and introduce a new index $j=K-i+1$, then we have

$$
\begin{equation*}
\mathbf{F}_{N_{K}}^{a_{K}} \otimes \mathbf{F}_{N_{K-1}}^{a_{K-1}} \otimes \ldots \otimes \mathbf{F}_{N_{1}}^{a_{1}}=\prod_{j=1}^{K}\left(\mathbf{I}_{\frac{N}{N(j)}} \otimes \mathbf{F}_{N_{j}}^{a_{j}} \otimes \mathbf{I}_{N(j-1)}\right) \tag{22}
\end{equation*}
$$

Since the left side of the above equality is equal to the matrix of MPDPFrFT, so when we enter the designation

$$
\begin{equation*}
\mathbf{R}_{N}^{\left(a_{k}, N_{k}, k\right)}=\mathbf{I}_{\frac{N}{N(k)}} \otimes \mathbf{F}_{N_{k}}^{a_{k}} \otimes \mathbf{I}_{N(k-1)} \tag{23}
\end{equation*}
$$

for $k=1,2, \ldots, K$, we may write

$$
\begin{equation*}
\mathbf{F}_{N}^{\left(a_{1}, a_{2}, \ldots, a_{K}, N_{1}, N_{2}, \ldots, N_{K}\right)}=\mathbf{R}_{N}^{\left(a_{K}, N_{K}, K\right)} \mathbf{R}_{N}^{\left(a_{K-1}, N_{K-1}, K-1\right)} \ldots \mathbf{R}_{N}^{\left(a_{1}, N_{1}, 1\right)} \tag{24}
\end{equation*}
$$

so, we have the following matrix-vector procedure to calculate the output of MPDPFrFT

$$
\begin{equation*}
\mathbf{y}_{N}^{\left(a_{1}, a_{2}, \ldots, a_{K}, N_{1}, N_{2}, \ldots, N_{K}\right)}=\mathbf{R}_{N}^{\left(a_{K}, N_{K}, K\right)} \mathbf{R}_{N}^{\left(a_{K-1}, N_{K-1}, K-1\right)} \ldots \mathbf{R}_{N}^{\left(a_{1}, N_{1}, 1\right)} \mathbf{x}_{N} \tag{25}
\end{equation*}
$$

For example, we consider the case of $N=12=2 \times 2 \times 3$. In this case $K=3, N_{1}=2$, $N_{2}=2$ and $N_{3}=3$. In this case, the algorithm for computing the output vector is represented by the following matrix-vector procedure:

$$
\begin{equation*}
\mathbf{y}_{12}^{\left(a_{1}, a_{2}, a_{3}, 2,2,3\right)}=\mathbf{R}_{12}^{\left(a_{3}, 3,3\right)} \mathbf{R}_{12}^{\left(a_{2}, 2,2\right)} \mathbf{R}_{12}^{\left(a_{1}, 2,1\right)} \mathbf{x}_{12} \tag{26}
\end{equation*}
$$

where the corresponding matrices take the forms

$$
\begin{gather*}
\mathbf{R}_{12}^{\left(a_{1}, 2,1\right)}=\mathbf{I}_{6} \otimes \mathbf{F}_{2}^{a_{1}} \otimes \mathbf{I}_{1}=\mathbf{I}_{6} \otimes \mathbf{F}_{2}^{a_{1}}  \tag{27}\\
\mathbf{R}_{12}^{\left(a_{2}, 2,2\right)}=\mathbf{I}_{3} \otimes \mathbf{F}_{2}^{a_{2}} \otimes \mathbf{I}_{2} \tag{28}
\end{gather*}
$$

and

$$
\begin{equation*}
\mathbf{R}_{12}^{\left(a_{3}, 3,3\right)}=\mathbf{I}_{1} \otimes \mathbf{F}_{3}^{a_{3}} \otimes \mathbf{I}_{4}=\mathbf{F}_{3}^{a_{3}} \otimes \mathbf{I}_{4} \tag{29}
\end{equation*}
$$

where $\mathbf{F}_{N_{k}}^{a_{k}}$ is the DFrFT matrix of order $N_{k}$ with fractional parameter $a_{k}$, so

$$
\mathbf{F}_{2}^{a_{1}}=\left[\begin{array}{ll}
f_{00}^{\left(a_{1}\right)} & f_{01}^{\left(a_{1}\right)}  \tag{30}\\
f_{10}^{\left(a_{1}\right)} & f_{11}^{\left(a_{1}\right)}
\end{array}\right], \mathbf{F}_{2}^{a_{2}}=\left[\begin{array}{ll}
f_{00}^{\left(a_{2}\right)} & f_{01}^{\left(a_{2}\right)} \\
f_{10}^{\left(a_{2}\right)} & f_{11}^{\left(a_{2}\right)}
\end{array}\right], \mathbf{F}_{3}^{a_{3}}=\left[\begin{array}{lll}
f_{00}^{\left(a_{3}\right)} & f_{01}^{\left(a_{3}\right)} & f_{02}^{\left(a_{3}\right)} \\
f_{10}^{\left(a_{3}\right)} & f_{11}^{\left(a_{3}\right)} & f_{12}^{\left(a_{3}\right)} \\
f_{20}^{\left(a_{3}\right)} & f_{21}^{\left(a_{3}\right)} & f_{22}^{\left(a_{3}\right)}
\end{array}\right]
$$

Figure 9 shows a data flow diagram of a fast algorithm for a 12-point MPDPFrFT. The data flow diagram is oriented from left to right and straight lines represent data transfer operations. In turn, the rectangles denote small-sized DFrFT kernels that implement the multiplications of the matrices $\mathbf{F}_{N_{k}}^{a_{k}}, k=1,2,3$, by the corresponding input vectors.


Figure 9. Data flow diagram of the algorithm of multiple-parameter discrete pseudo-fractional Fourier transform for $N=12=2 \times 2 \times 3$.

## 7. Computational Complexity

We will now estimate the total number of multiplications and additions of complex numbers needed to compute the output of MPDPFrFT (or DPFrFT if $a_{1}=a_{2}=\ldots=a_{K}=$ $a$ ), for the complex-valued input vector $\mathbf{x}_{N}$, according to Equation (25). We assume that the matrices $\mathbf{R}_{N}^{\left(a_{k}, N_{k}, k\right)}, k=1,2, \ldots, K$ are factorized as in Equation (23) and the matrices $\mathbf{F}_{N_{k}}^{a_{k}}$ have been prepared in advance for the selected fractional parameters $a_{k}, k=1,2, \ldots, K$. The numbers of arithmetic operations depend on the method of factorization of the number $N$ into the product $N_{1} N_{2} \ldots N_{K}$ and on the algorithm used for the calculation of the DFrFT for individual $k$. When performing the matrix-vector multiplication according to Equation (25), from the right side to the left, we have $K$ steps (corresponding to the multiplication of the matrices $\mathbf{R}_{N}^{\left(a_{k}, N_{k}, k\right)}, k=1,2, \ldots, K$ by a vector). At the step $k$, there are $N / N_{k}$ multiplications of the matrix $\mathbf{F}_{N_{k}}^{a_{k}}$ by the corresponding subvectors needed. If we carry out the multiplication of the latter matrix by the corresponding subvector in the classical way, we have $N_{k}^{2}$ multiplications and $N_{k}\left(N_{k}-1\right)$ additions of complex numbers. Hence, the total numbers of multiplications $L_{\times}$and additions $L_{+}$of complex numbers are equal to

$$
\begin{gather*}
L_{\times}=\sum_{k=1}^{K}=\frac{N}{N_{k}} N_{k}^{2}=N \sum_{k=1}^{K} N_{k}  \tag{31}\\
L_{+}=\sum_{k=1}^{K} \frac{N}{N_{k}} N_{k}\left(N_{k}-1\right)=N \sum_{k=1}^{K}\left(N_{k}-1\right) \tag{32}
\end{gather*}
$$

For the example shown in Figure 1, i.e., $N=12=2 \times 2 \times 3$, the numbers of additions and multiplications of complex numbers are equal to 84 and 48 , respectively.

It is necessary to emphasize that the matrices $\mathbf{F}_{N_{k}}^{a_{k}}, k=1,2, \ldots, K$ have a special structure; therefore, the numbers of arithmetic operations when calculating the matrixvector product of each of them by the vector can be further reduced by about a half $[12,13]$.

## 8. Conclusions

The paper describes a new type of discrete fractional-like transform called the discrete pseudo-fractional Fourier transform. We also present a generalization of this transform, called multiple-parameter discrete pseudo-fractional Fourier transform. Finally, we propose a fast algorithm for calculating this transform. A distinctive feature of this transform and its algorithm is that while maintaining all the advantages of fractionality and multiparametricity, a significant reduction in computational complexity is achieved compared to the true DFrFT [12]. In addition, the algorithm has a regular modular structure. Typical cores of the algorithm are standard small-size DFrFTs, which allows them to be used as unified building blocks in software or hardware implementation [13]. This will simplify and unify the process of developing programs and digital signal processors that implement the considered algorithm. Matrices of the proposed transform are defined by the smaller size DFrFT matrices. Taking into account the specific properties of the introduced transform, one can predict its interesting applications. However, these questions are beyond the scope of our article.

Author Contributions: Conceptualization, A.C.; methodology, D.M.-M.; validation, D.M.-M.; formal analysis, A.C. and D.M.-M.; investigation, D.M.-M.; writing-original draft preparation, D.M.-M.; writing-review and editing, A.C. and D.M.-M.; visualization, D.M.-M.; supervision, A.C. All authors have read and agreed to the published version of the manuscript.
Funding: This research received no external funding.
Data Availability Statement: All data included in this study are available upon request by contacting with the corresponding author.

Conflicts of Interest: Authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| DFrFT | discrete fractional Fourier transform |
| :--- | :--- |
| DFrHT | discrete fractional Hadamard transform |
| MPDFrFT | multiple-parameter discrete fractional Fourier transform |
| DFT | discrete Fourier transform |
| DPFrFT | discrete pseudo-fractional Fourier transform |
| MPDPFrFT | multiple-parameter discrete pseudo-fractional Fourier transform |

## References

1. Pei, S.C.; Yeh, M.H. Discrete fractional Fourier transform. In Proceedings of the 1996 IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World, ISCAS 96, Atlanta, GA, USA, 15 May 1996; Volume 2, pp. 536-539. [CrossRef]
2. Pei, S.C.; Yeh, M.H. Improved discrete fractional Fourier transform. Opt. Lett. 1997, 22, 1047-1049. [CrossRef] [PubMed]
3. Candan, C.; Alper Kutay, M.; Ozaktas, H.M. The discrete fractional Fourier transform. IEEE Trans. Sig. Proc. 2000, 48, 1329-1337. [CrossRef]
4. Pei, S.C.; Tseng, C.C.; Yeh, M.H.; Shyu, J.J. Discrete fractional Hartley and Fourier transforms. IEEE Trans. Circuits Syst. II Analog Digit. Signal Process. 1998, 45, 665-675. [CrossRef]
5. Pei, S.C.; Yeh, M.H. The discrete fractional cosine and sine transforms. IEEE Trans. Sig. Proc. 2001, 49, 1198-1207. [CrossRef]
6. Pei, S.C.; Yeh, M.H. Discrete fractional Hadamard transform. In Proceedings of the 1999 IEEE International Symposium on Circuits and Systems (ISCAS), Orlando, FL, USA, 30 May-2 June 1999; Volume 3, pp. 179-182. [CrossRef]
7. Samil Yetik, I.; Alper Kutay, M.; Ozaktas, H.M. Image representation and compression with the fractional Fourier transform. Opt. Commun. 2001, 197, 275-278. [CrossRef]
8. Hennelly, B.; Sheridan, J.T. Fractional Fourier transform-based image encryption: Phase retrieval algorithm. Opt. Commun. 2003, 226, 61-80. [CrossRef]
9. Djurovic, I.; Stankovic, S.; Pitas, I. Digital watermarking in the fractional Fourier transformation domain. J. Netw. Comput. Appl. 2001, 24, 167-173. [CrossRef]
10. Durak, L.; Aldirmaz, S. Adaptive fractional Fourier domain filtering. Signal Process. 2010, 90, 1188-1196. [CrossRef]
11. Lin, Q.; Yanhong, Z.; Ran, T.; Yue, W. Adaptive filtering in fractional Fourier domain. In Proceedings of the 2005 IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, Beijing, China, 8-12 August 2005; Volume 2, pp. 1033-1036. [CrossRef]
12. Majorkowska-Mech, D.; Cariow, A. A Low-Complexity Approach to Computation of the Discrete Fractional Fourier Transform. Circuits Syst. Signal Process. 2017, 36, 4118-4144. [CrossRef]
13. Cariow, A.; Papliński, J.; Majorkowska-Mech, D. Some Structures of Parallel VLSI-Oriented Processing Units for Implementation of Small Size Discrete Fractional Fourier Transforms. Electronics 2019, 8, 509. [CrossRef]
14. Pei, S.C.; Hsue, W.L. The multiple-parameter discrete fractional Fourier transform. IEEE Signal Process. Lett. 2006, 13, 329-332. [CrossRef]
15. Majorkowska-Mech, D.; Cariow, A. Discrete Pseudo-Fractional Hadamard Transform and its Fast Algorithm. IEEE Signal Process. Lett. 2020, 27, 1195-1199. [CrossRef]
16. Graham, A. Kronecer Products and Matrix Calculus: With Applications; Ellis Horwood Limited: Chichester, UK, 1981.
17. Johnson, J.R.; Johnson, R.W.; Rodriguez, D.; Tolimieri, R. A methodology for designing, modifying and implementing Fourier transform algorithms on various architecture. Circuits Syst. Signal Process. 1990, 9, 449-500. [CrossRef]
