A Method for the Discrete Fractional Fourier Transform Computation

Min-Hung Yeh and Soo-Chang Pei

***Abstract*—A new method for the discrete fractional Fourier transform (DFRFT) computation is given in this paper. With the help of this method, the DFRFT of any angle can be computed by a weighted summation of the DFRFTs with the special angles.**

***Index Terms*—Discrete Fourier transform, discrete fractional Fourier transform, fractional Fourier transform.**

I. Introduction

The fractional Fourier transform (FRFT) indicates a rotation of signal in the time–frequency plane, and it has been widely investigated with many applications [1]–[3]. Because of the importance of FRFT, the discrete fractional Fourier transform (DFRFT) has become an important issue in recent years [4]–[11]. In the development of DFRFT, the DFRFT has been considered to be the combination of four parts [4]:

1) the original signal;
2) its DFT;
3) circular flipped of signal;
4) circular flipped of its DFT.

Unfortunately, it cannot have similar results as the continuous case [6]. In 1996, we found that the DFRFT with DFT Hermite eigenvectors can have similar outputs as those of the continuous case [7], [8]. These DFRFTs use the DFT Hermite eigenvectors as their eigenvectors and have similar a eigendecomposition form as the continuous FRFT kernel. The eigendecomposition method has been proved and justified in [9], and it has been successfully used in many applications [3].

The DFRFT discussed in this paper is for the eigendecomposition method [7], [8]. Although the eigendecomposition method can have similar results as in the continuous case, its computation cost is very large. The goal of this paper is to introduce a new computation method for the DFRFT.

II. Review of the Discrete Fractional Fourier Transform

A. The Definition of DFRFT

The DFRFT is developed based on the eigendecomposition, and its transform kernel is written as [7]–[9]

$$\mathbf{F}^{2\alpha/\pi} = \mathbf{V}\mathbf{D}^{2\alpha/\pi}\mathbf{V}^T \tag{1}$$

where α indicates the rotation angle of DFRFT. $\mathbf{V} = [\mathbf{v}_0|\mathbf{v}_1|\cdots|\mathbf{v}_{N-2}|\mathbf{v}_{N-1}]$ for N is odd, $\mathbf{V} = [\mathbf{v}_0|\mathbf{v}_1|\cdots|\mathbf{v}_{N-2}|\mathbf{v}_N]$ for N is even, and $\mathbf{v}_k$ is the kth-order DFT Hermite eigenvector. $\mathbf{D}^{2\alpha/\pi}$ is a diagonal matrix with eigenvalues of DFRFT in the diagonal entries. The methods for finding the DFT Hermite eigenvectors $\mathbf{v}_k$ are presented in [7] and [8]. In Table I, there exists a jump in the last eigenvalues for the two even-length cases. Therefore, there are some differences in computing the DFRFT kernels between even- and odd-length cases. For the odd- and even- length cases, (1) can be written as follows:

TABLE I
EIGENVALUES ASSIGNMENT RULE OF DFRFT KERNEL MATRIX

N	the eigenvalues
$4m$	$e^{-jk\alpha}$, $k = 0, 1, 2, \cdots, (4m-2), 4m$
$4m+1$	$e^{-jk\alpha}$, $k = 0, 1, 2, \cdots, (4m-1), 4m$
$4m+2$	$e^{-jk\alpha}$, $k = 0, 1, 2, \cdots, 4m, (4m+2)$
$4m+3$	$e^{-jk\alpha}$, $k = 0, 1, 2, \cdots, (4m+1), (4m+2)$

$$\mathbf{F}^{2\alpha/\pi} = \sum_{k=0}^{N-1} e^{-jk\alpha}\mathbf{v}_k\mathbf{v}_k^T \ (N \text{ is odd}) \tag{2}$$

$$\mathbf{F}^{2\alpha/\pi} = \sum_{k=0}^{N-2} e^{-jk\alpha}\mathbf{v}_k\mathbf{v}_k^T + e^{-jN\alpha}\mathbf{v}_N\mathbf{v}_N^T \ (N \text{ is even}). \tag{3}$$

The DFRFT output signal is computed as

$$\mathbf{X}_\alpha = \sum_{k=0}^{N-1} e^{-jk\alpha}\mathbf{v}_k\mathbf{v}_k^T\mathbf{x} \ (N \text{ is odd}) \tag{4}$$

$$\mathbf{X}_\alpha = \sum_{k=0}^{N-2} e^{-jk\alpha}\mathbf{v}_k\mathbf{v}_k^T\mathbf{x} + e^{-jN\alpha}\mathbf{v}_N\mathbf{v}_N^T\mathbf{x} \ (N \text{ is even}). \tag{5}$$

B. Current Implementation Method of DFRFT

The DFRFT is based on the eigendecomposition of DFT kernel matrix, and the DFT Hermite eigenvectors are used for the DFRFT kernel construction. For a fixed number of point, the DFT Hermite eigenvectors and the transform kernel of DFRFT can be computed *a priori*, but regardless of the cases in (4) and (5), an innerproduct operation for the DFRFT transform kernel and the input signal is still required to compute DFRFT. Unfortunately, the matrix-vector products in (4) and (5) take $\mathcal{O}(N^2)$ time.

III. Method for DFRFT Computation

A. Development of the New Method

In this section, we will develop a new method for DFRFT computation. The ideal of the developed method is to compute the DFRFT of any angle by a weighted summation of the DFRFTs in special angles.

Proposition 1: It is assumed that $\mathbf{x}$ is a discrete signal with odd length N. The DFRFT of $\mathbf{x}$ for rotation angle α can be computed as

$$\mathbf{X}_\alpha = \sum_{n=0}^{N-1} B_{n,\alpha}\mathbf{X}_{n\beta} \tag{6}$$

where $\beta = 2\pi/N$. The weighting coefficients $B_{n,\alpha}$ are computed as

$$B_{n,\alpha} = IDFT\{e^{-jk\alpha}\}_{k=0,1,2,\ldots,N-1} = \frac{1}{N}\sum_{n=0}^{N-1} e^{-jk\alpha}e^{j(2\pi/N)nk}. \tag{7}$$

Manuscript received October 24, 2001; revised October 28, 2002. This work was supported by the National Science Council, R.O.C., under Contract NSC91-2219-E-002-044 and by the Ministry of Education under Contract 89-E-FA06-2-4. The associate editor coordinating the review of this paper and approving it for publication was Dr. Anamitra Makur.

M.-H. Yeh is with the Department of Electronic Engineering, National I-Lan Institute of Technology, I-Lan, Taiwan, R.O.C. (e-mail: mhyeh@ilantech.edu.tw)

S.-C. Pei is with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. (e-mail: pei@cc.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TSP.2002.808113

Proof: Because $B_{n,\alpha}$ is equal to the inverse DFT of $\{e^{-jk\alpha}\}_{k=0,1,\ldots,N-1}$, the DFT of $B_{n,\alpha}$ will reach the sequence $\{e^{-jk\alpha}\}_{k=0,1,\ldots,N-1}$

$$e^{-jk\alpha} = \sum_{n=0}^{N-1} B_{n,\alpha} e^{-j\beta nk} = \sum_{n=0}^{N-1} B_{n,\alpha} e^{-j(2\pi/N)nk}. \quad (8)$$

Then, the definition of DFRFT in (4) will become

$$\begin{aligned}\mathbf{X}_\alpha &= \sum_{k=0}^{N-1}\left(\sum_{n=0}^{N-1} B_{n,\alpha} e^{-j(2\pi/N)kn}\right)\mathbf{v}_k\mathbf{v}_k^T\mathbf{x}\\ &= \sum_{n=0}^{N-1} B_{n,\alpha}\left(\sum_{k=0}^{N-1} e^{-j(2\pi/N)kn}\mathbf{v}_k\mathbf{v}_k^T\mathbf{x}\right)\\ &= \sum_{n=0}^{N-1} B_{n,\alpha}\mathbf{X}_{n\beta} \quad (9)\end{aligned}$$

□

Equation (7) indicates that $B_{n,\alpha}$ can be computed from the inverse discrete Fourier transform (IDFT) of the values $\{e^{-jk\alpha}\}_{k=0,1,\ldots,N-1}$. The weighting coefficients $B_{n,\alpha}$ have a closed-form solution.

$$B_{n,\alpha} = \begin{cases}\frac{1}{N}\frac{1-e^{-j(N-1)(\alpha-n\beta)}}{1-e^{-j(\alpha-n\beta)}}, & \alpha \neq k\beta\\ \delta(n-k), & \alpha = k\beta\end{cases}. \quad (10)$$

Proposition 1 tells us the DFRFT computation with odd point of length can be realized by the weighted summation of the DFRFTs in special angles. The special angles are multiples of $2\pi/N$, and the weighting coefficients are computed from an IDFT operation.

In Table I, there exists a jump in the eigenvalue assignment when the length of signal is even. Therefore, the above computation method for DFRFT with odd point cannot work for the even length. It is necessary to design a new different method for the even point of DFRFT computation.

Proposition 2: It is assumed that $\mathbf{x}$ is a discrete signal with even length N. The DFRFT of $\mathbf{x}$ with rotation angle α can be computed by the following equation:

$$\mathbf{X}_\alpha = \sum_{n=0}^{N} B_{n,\alpha}\mathbf{X}_{n\beta} \quad (11)$$

where $\beta = 2\pi/(N+1)$, and the weighting coefficients $B_{n,\alpha}$ are computed as

$$\begin{aligned}B_{n,\alpha} =& IDFT\{e^{-jk\alpha}\}_{k=0,1,2,\ldots,N}\\ =& \frac{1}{N+1}\sum_{n=0}^{N} e^{-jk\alpha} e^{j(2\pi/(N+1))nk}. \quad (12)\end{aligned}$$

Proof: There exist two different points between (11) and (6). In (11), $(N+1)$ terms are summed, and the special angles are multiples of $2\pi/(N+1)$. Based on the same idea as the odd case, the DFT of the sequence $B_{n,\alpha}$ is used to express the sequence $\{e^{-jk\alpha}\}_{k=0,1,\ldots,N}$.

$$e^{-jk\alpha} = \sum_{n=0}^{N} B_{n,\alpha} e^{-j(2\pi/(N+1))nk}. \quad (13)$$

Thus, (5) can be derived as in (14), shown at the bottom of the page. □

Similar to the odd length case, the weighting coefficients $B_{n,\alpha}$ also have a closed form. The closed form of $B_{n,\alpha}$ is shown as follows:

$$B_{n,\alpha} = \begin{cases}\frac{1}{N+1}\frac{1-e^{-jN(\alpha-n\beta)}}{1-e^{-j(\alpha-n\beta)}}, & \alpha \neq k\beta\\ \delta(n-k), & \alpha = k\beta\end{cases}. \quad (15)$$

By Proposition 1 and 2, we can conclude that the DFRFT of any angle can be computed by the weighted summation of the DFRFTs with special angles. The special angles are multiples of $2\pi/N$ for the odd case and are multiples of $2\pi/(N+1)$ for even length. Regardless of even- or odd-length cases, the weighting coefficients are obtained from an IDFT operation. The N-point IDFT is needed for odd length, and the $(N+1)$-point is computed for even length. Both in odd- and even- length cases, an odd-point IDFT is computed to get the weighting coefficients. The popular FFT algorithm can only be applied while the length is power of 2; therefore, it cannot help us get the weighted coefficients. It is lucky that the DFT or IDFT with a prime point of length can have the Winograd Fourier transform algorithm [12]. Thus, we can use it with the DFRFT with prime length.

B. Discussion

The original method in (4) and (5) requires N^2 storages to store the transform kernel. If the computing angle is changed, the transform kernel is also changed and needs to be recomputed. In our new algorithm, only the DFRFT of the specified angle is stored. Any angle of the DFRFT can be obtained by a weighted summation of these specified DFRFTs. The weighted computation in this new algorithm still takes $\mathcal{O}(N^2)$ multiplications; therefore, its computation load is still $\mathcal{O}(N^2)$.

IV. Two Implementation Methods for the New Algorithm

In this section, we will introduce two implementation methods for the DFRFT computation algorithm shown in the previous section. One is called the *parallel method*, and the other is the *cascade method*. The principle of parallel method is straightforward, and it uses the results in Propositions 1 and 2 directly. It must be noted that the special angles and numbers of terms are different for the even- and odd-length cases.

Because the angle additivity is existed in the DFRFT, a DFRFT with angle α performed by a DFRFT with angle β will be a DFRFT with angle $(\alpha+\beta)$. Using this angle additivity in the DFRFT, the DFRFT in (6) can be rewritten in the following form:

$$\begin{aligned}\mathbf{X}_\alpha =& B_{N-1,\alpha}\mathbf{F}^{(N-1)2\beta/\pi}\mathbf{x} + B_{N-2,\alpha}\mathbf{F}^{(N-2)2\beta/\pi}\mathbf{x}\\ &+ \cdots + B_{1,\alpha}\mathbf{F}^{2\beta/\pi}\mathbf{x} + B_{0,\alpha}\mathbf{x} \quad (16)\end{aligned}$$

$$\begin{aligned}\mathbf{X}_\alpha &= \sum_{k=0}^{N-2}\sum_{n=0}^{N} B_{n,\alpha} e^{-j(2\pi/(N+1))kn}\mathbf{v}_k\mathbf{v}_k^T\mathbf{x} + \sum_{n=0}^{N} B_{n,\alpha} e^{-j(2\pi/(N+1))Nn}\mathbf{v}_N\mathbf{v}_N^T\mathbf{x}\\ &= \sum_{n=0}^{N} B_{n,\alpha}\left(\sum_{k=0}^{N-2} e^{-j(2\pi/(N+1))kn}\mathbf{v}_k\mathbf{v}_k^T\mathbf{x} + e^{-j(2\pi/(N+1))Nn}\mathbf{v}_N\mathbf{v}_N^T\mathbf{x}\right)\\ &= \sum_{n=0}^{N} B_{n,\alpha}\mathbf{X}_{n\beta} \quad (14)\end{aligned}$$

$$= \mathbf{F}^{2\beta/\pi}(\cdots(\mathbf{F}^{2\beta/\pi}(\mathbf{F}^{2\beta/\pi}(B_{N-1,\alpha}\mathbf{X}_\beta + B_{N-2,\alpha}\mathbf{x}) + B_{N-3,\alpha}\mathbf{x}) + B_{N-4,\alpha}\mathbf{x}) + \cdots) + B_{0,\alpha}\mathbf{x}. \quad (17)$$

Equation (17) shows us that the DFRFT with any angle can be realized by the DFRFT with *only one* special angle. Similar to the parallel method, the special angle and numbers of terms of cascade forms are also different for the even and odd cases. The cascade form of the DFRFT in even case is written as

$$\mathbf{X}_\alpha = \mathbf{F}^{2\beta/\pi}(\cdots(\mathbf{F}^{2\beta/\pi}(\mathbf{F}^{2\beta/\pi}(B_{N,\alpha}\mathbf{X}_\beta + B_{N-1,\alpha}\mathbf{x}) + B_{N-2,\alpha}\mathbf{x}) + B_{N-3,\alpha}\mathbf{x}) + \cdots) + B_{0,\alpha}\mathbf{x}. \quad (18)$$

Both the implementation methods can have the advantages for the DFRFT computation. The parallel method is suitable for the signal whose DFRFTs in special angles are already known. The computation of the DFRFT will become only a linear combination of the DFRFTs in special angles. It has been shown in [2] and [8] that a chirp signal can have an impulse output for an FRFT or DFRFT with a appropriate angular parameter. Therefore, the FRFT and DFRFT can be used for chirp signal detection and optimal filtering in fractional Fourier domains [13]. Using this proposed method, the DFRFTs of special angles of the desired signal can be computed *a priori*. The chirp detection and optimal filtering in the fractional Fourier domain can be resolved by adjusting the rotation angle α.

In other way, the cascade method means that the computation of the DFRFT can be realized from the DFRFT with only one specified angle. If the DFRFT with the specified angle can be computed efficiently, the computation of the DFRFT will become efficient. Such an architecture is very suitable for VLSI implementations.

V. Conclusions

In this paper, we develop a new method for DFRFT computation. By this new method, the DFRFT of any angle can be computed by a linear combination of the DFRFTs with special angles. The weighting coefficients of the linear combination are obtained from an IDFT computation. Moreover, this new method can be realized by two methods: parallel and cascade. The parallel method is suitable for the signal whose DFRFT in special angles are already known. The chirp signal detection is a common one, and the cascade method has a regular structure; therefore, it is very suitable for VLSI implementation.

References

[1] V. Namias, "The fractional order Fourier transform and its application to quantum mechanics," *J. Inst. Math. Applicat.*, vol. 25, pp. 241–265, 1980.

[2] L. B. Almeida, "The fractional Fourier transform and time-frequency representation," *IEEE Trans. Signal Processing*, vol. 42, pp. 3084–3091, Nov. 1994.

[3] H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, *The Fractional Fourier Transform with Applications in Optics and Signal Processing*. New York: Wiley, 2000.

[4] B. Santhanam and J. H. McClellan, "The discrete rotational Fourier transform," *IEEE Trans. Signal Processing*, vol. 42, pp. 994–998, Apr. 1996.

[5] C. C. Shih, "Fractionalization of Fourier transform," *Opt. Commun.*, vol. 118, pp. 495–498, Aug. 1995.

[6] H. M. Ozaktas, O. Arikan, M. A. Kutay, and G. Bozdagi, "Digital computation of the fractional Fourier transform," *IEEE Trans. Signal Processing*, vol. 44, pp. 2141–2150, Sept. 1996.

[7] S. C. Pei and M. H. Yeh, "Improved discrete fractional Fourier transform," *Opt. Lett.*, vol. 22, pp. 1047–1049, July 1997.

[8] S. C. Pei, M. H. Yeh, and C. C. Tseng, "Discrete fractional Fourier transform based on orthogonal projections," *IEEE Trans. Signal Processing*, vol. 47, pp. 1335–1348, May 1999.

[9] C. Candan, M. A. Kutay, and H. M. Ozaktas, "The discrete fractional Fourier transform," *IEEE Trans. Signal Processing*, vol. 48, pp. 1329–1337, May 2000.

[10] S. C. Pei and J. J. Ding, "Closed-form discrete fractional and affine Fourier transforms," *IEEE Trans. Signal Processing*, vol. 48, pp. 1338–1353, May 2000.

[11] S. Oraintara, "The generalized discrete fractional Fourier transforms," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 1185–1188, May 2002.

[12] D. F. Elliott and K. R. Rao, *Fast Fourier Transform and Convolution Algorithms*. New York: Springer-Verlag, 1982.

[13] M. A. Kutay, H. M. Ozaktas, O. Arikan, and L. Onural, "Optimal filtering in fractional Fourier domains," *IEEE Trans. Signal Processing*, vol. 45, pp. 1129–1143, May 1997.