Fractional Cosine, Sine, and Hartley Transforms

Soo-Chang Pei, *Fellow, IEEE,* and Jian-Jiun Ding

***Abstract*—In previous papers, the Fourier transform (FT) has been generalized into the fractional Fourier transform (FRFT), the linear canonical transform (LCT), and the simplified fractional Fourier transform (SFRFT). Because the cosine, sine, and Hartley transforms are very similar to the FT, it is reasonable to think they can also be generalized by the similar way. In this paper, we will introduce several new transforms. They are all the generalization of the cosine, sine, or Hartley transform. We first derive the fractional cosine, sine, and Hartley transforms (FRCT/FRST/FRHT). They are analogous to the FRFT. Then, we derive the canonical cosine and sine transforms (CCT/CST). They are analogous to the LCT. We also derive the simplified fractional cosine, sine, and Hartley transforms (SFRCT/SFRST/SFRHT). They are analogous to the SFRFT and have the advantage of real-input–real-output. We also discuss the properties, digital implementation, and applications (e.g., the applications for filter design and space-variant pattern recognition) of these transforms. The transforms introduced in this paper are very efficient for digital implementation. We can just use one half or one fourth of the real multiplications required for the FRFT and LCT to implement them. When we want to process even, odd, or pure real/imaginary functions, we can use these transforms instead of the FRFT and LCT. Besides, we also show that the FRCT/FRST, CCT/CST, and SFRCT/SFRST are also useful for the one-sided ($t \in [0, \infty)$) signal processing.**

***Index Terms*—Fractional cosine transform, fractional Hartley transform, fractional sine transform.**

NOMENCLATURE

FT	Fourier transform.
FRFT	Fractional Fourier transform.
SFRFT	Simplified fractional Fourier transform.
LCT	Linear canonical transform.
FRCT	Fractional cosine transform.
FRST	Fractional sine transform.
FRHT	Fractional Hartley transform.
CCT	Canonical cosine transform.
CST	Canonical sine transform.
CHT	Canonical Hartley transform.
SFRCT	Simplified fractional cosine transform.
SFRST	Simplified fractional sine transform.
SFRHT	Simplified fractional Hartley transform.

The following are the notations and transform results used in this paper.

- Notations for FRFT, LCT, and SFRFT: $O_F^\alpha(\ )$, $O_F^{(a,b,c,d)}(\ )$, $O_{SF}^\alpha(\ )$.
- Notations for FRCT, FRST, and FRHT: $O_C^\alpha(\ )$, $O_S^\alpha(\ )$, $O_H^\alpha(\ )$.
- Notations for CCT, CST, and CHT: $O_C^{(a,b,c,d)}(\ )$, $O_S^{(a,b,c,d)}(\ )$, $O_H^{(a,b,c,d)}(\ )$.
- Notations for SFRCT, SFRST, and SFRHT of type 1 and 2: $O_{SC}^\alpha(\ )$, $O_{SS}^\alpha(\ )$, $O_{SH}^\alpha(\ )$, $O_{SC''}^\alpha(\ )$, $O_{SS''}^\alpha(\ )$, $O_{SH''}^\alpha(\ )$.
- Transform results of $g(t)$ for FRFT, LCT, and SFRFT: $G_F^\alpha(s)$, $G_F^{(a,b,c,d)}(s)$, $G_{SF}^\alpha(s)$.
- Transform results of $g(t)$ for FRCT, FRST, and FRHT: $G_C^\alpha(s)$, $G_S^\alpha(s)$, $G_H^\alpha(s)$.
- Transform results of $g(t)$ for CCT, CST, and CHT: $G_C^{(a,b,c,d)}(s)$, $G_S^{(a,b,c,d)}(s)$, $G_H^{(a,b,c,d)}(s)$.
- Transform results of $g(t)$ for SFRCT, SFRST, and SFRHT of types 1 and 2: $G_{SC}^\alpha(s)$, $G_{SS}^\alpha(s)$, $G_{SH}^\alpha(s)$, $G_{SC''}^\alpha(s)$, $G_{SS''}^\alpha(s)$, $G_{SH''}^\alpha(s)$.

I. INTRODUCTION

BEFORE discussing the fractional cosine, sine, and Hartley transforms (FRCT/FRST/FRHT), we first describe the concept of fractional operations. Suppose there is an operation $O(\ )$

$$O(g(x)) = G(k). \tag{1}$$

Then, its fractional operation (which is denoted by $O^a(\ )$, where a is some real number) is the operation satisfying the following properties.

1) Boundary properties:

$$O^0(g(x)) = g(x), \quad O^1(g(x)) = G(k). \tag{2}$$

2) Additivity property:

$$O^b(O^a(g(x))) = O^a(O^b(g(x))) = O^{a+b}(g(x)). \tag{3}$$

From the additivity property, the inverse of the fractional operation is just $O^{-a}(\ )$

$$O^{-a}(G_a(s)) = g(x) \text{ where } G_a(s) = O^a(g(x)). \tag{4}$$

Because it is free to choose the parameters a, using the fractional operation is more flexible than using the original operation, and some problems that cannot be solved by the original operation will be solved by the fractional operation.

The fractional operations for some operations are easy to derive. For example, for the multiplication operation $O(g(x)) = g(x) \cdot y(x)$, we can just define its fractional operation as

$$O^a(g(x)) = g(x) \cdot y^a(x). \tag{5}$$

Nevertheless, for most of the operations, the fractional operations are not so obvious, and we must use some special method

---

Manuscript received May 17, 1999; revised March 22, 2002. This work was supported by the National Science Council, R.O.C., under Contract NSC 90-2213-E-002-093. The associate editor coordinating the review of this paper and approving it for publication was Prof. Scott C. Douglas.

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

Publisher Item Identifier S 1053-587X(02)05652-0.

to find their fractional operations. We introduce a method as follows.

Suppose the operation O can be decomposed into O_1^{-1}, O_2, and O_1

$$G(k) = O(g(x)) = O_1^{-1}(O_2[O_1(g(x))]) \tag{6}$$

and fractional operation of O_2 has been known. Then, we can derive the fractional operation of O as

$$G_a(k') = O^a(g(x)) = O_1^{-1}(O_2^a[O_1(g(x))]). \tag{7}$$

We express this concept by Figs. 1 and 2. This is a useful method to derive the fractional operation. If O_2 is a multiplication operation, then the deriving process can be simplified even more.

For the Fourier transform (FT)

$$G(w) = FT(g(t)) = \sqrt{\frac{1}{2\pi}} \cdot \int_{-\infty}^{\infty} e^{-j\cdot w\cdot t} \cdot g(t)\cdot dt \tag{8}$$

its fractional operation, i.e., the *fractional Fourier transform (FRFT),* is defined as [3]

$$\begin{aligned} G_F^\alpha(s) =& O_F^\alpha(g(t)) = \sqrt{\frac{1-j\cot\phi}{2\pi}} \cdot e^{j/2\cdot\cot\phi\cdot s^2} \\ &\cdot \int_{-\infty}^{\infty} e^{-j\cdot\csc\phi\cdot s\cdot t} e^{j/2\cdot\cot\phi\cdot t^2} g(t)\cdot dt \end{aligned} \tag{9}$$

where $\phi = \alpha\pi/2$. The process of deriving the FRFT is shown in [3]. The original FT can be decomposed into the three steps as follows.

1)

$$g(t) \rightarrow \{a_m | m = 0, 1, 2, 3, \ldots\} \tag{10}$$

$$\text{where } a_m = C_m^{-1} \int_{-\infty}^{\infty} g(t) H_m(t) e^{-t^2/2} dt,$$

$$C_m = \int_{-\infty}^{\infty} H_m^2(t) e^{-t^2} dt \tag{11}$$

where $H_m(t)$ is the Hermite polynomial of order m [1].

2)

$$b_m = (-j)^m a_m. \tag{12}$$

3)

$$G(\omega) = \sum_{m=0}^{\infty} b_m \cdot e^{-w^2/2} \cdot H_m(\omega). \tag{13}$$

In [3], they derived FRFT by changing the second step as

2)

$$b_m = e^{-jm\cdot\alpha\cdot\pi/2} a_m \tag{14}$$

and keeping steps 1 and 3 unchanged [except that $G(\omega)$ in (13) is changed as $G_F^\alpha(s)$]. In fact, this method has used the concept of Figs. 1 and 2 [or (6) and (7)]. The first step (Hermite polynomials expansion) corresponds to the operation O_1 in Fig. 1, step 2 (multiplication operation) corresponds to the operation O_2, and the step 3 corresponds to the operation O_1^{-1}.

$$\begin{tikzcd} g(x) \arrow[rr, dotted, "O(g(x))"] \arrow[d, "O_1(g(x))"'] & & G(k) \\ p(v) \arrow[rr, "O_2(p(v))"] & & P(\omega) \arrow[u, "O_1^{-1}(P(\omega))"'] \end{tikzcd}$$

Fig. 1. Decomposing the operation O into O_1, O_2, and O_2^{-1}.

$$\begin{tikzcd} g(x) \arrow[rr, dotted, "O^a(g(x))"] \arrow[d, "O_1(g(x))"'] & & G_a(k') \\ p(v) \arrow[rr, "O_2^a(p(v))"] & & P_a(\omega') \arrow[u, "O_1^{-1}(P_a(\omega'))"'] \end{tikzcd}$$

Fig. 2. Deriving the fractional operation of O.

FRFT can be used for many applications, such as filter design, pattern recognition, optical system analysis, solving differentiation equation, phase retrieval, signal synthesis, etc.

The formulas of cosine, sine, and Hartley transforms [10], [11] are

$$\begin{aligned} G_C(w) =& CT(g(t)) \\ =& \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \cos(wt)\cdot g(t)\cdot dt \end{aligned} \tag{15}$$

$$\begin{aligned} G_S(w) =& ST(g(t)) \\ =& \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \sin(wt)\cdot g(t)\cdot dt \end{aligned} \tag{16}$$

$$\begin{aligned} G_H(w) =& HT(g(t)) \\ =& \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \text{cas}(wt)\cdot g(t)\cdot dt \end{aligned} \tag{17}$$

$$\text{where cas}(wt) = \cos(wt) + \sin(wt). \tag{18}$$

They are all very similar to the FT. Since the FT has fractional operation (FRFT), we expect that there are also fractional operations for the cosine, sine, and Hartley transforms.

Some types of the fractional cosine, sine, and Hartley transforms have been derived recently. In [9], they authors have derived the fractional cosine/sine transforms by taking the real/imaginary parts of the kernel of FRFT

$$\begin{aligned} O_C^\alpha(g(t)) &= \int_{-\infty}^{\infty} \text{Re}\,[K_F^\alpha(s,t)]\cdot g(t)\cdot dt \\ O_S^\alpha(g(t)) &= \int_{-\infty}^{\infty} \text{Im}\,[K_F^\alpha(s,t)]\cdot g(t)\cdot dt \end{aligned} \tag{19}$$

$$\begin{aligned} \text{where } K_F^\alpha(v,t) =& \sqrt{\frac{1-j\cot\phi}{2\pi}} \cdot e^{j/2\cot\phi\cdot s^2} \cdot e^{-j\csc\phi\cdot s\cdot t} \\ &\cdot e^{j/2\cot\phi\cdot t^2}, \quad \phi = \frac{\alpha\pi}{2}. \end{aligned} \tag{20}$$

In [9], they also derived the fractional Hartley transform by summing the fractional cosine and sine transforms defined as (19)

$$O_H^\alpha(g(t)) = \int_{-\infty}^{\infty} \{\text{Re}\,[K_F^\alpha(s,t)] + \text{Im}\,[K_F^\alpha(s,t)]\}\, g(t)\cdot dt. \tag{21}$$

It has also been derived in [7]. The fractional cosine, sine, and Hartley transforms defined as (19)–(21) satisfy the boundary property as (2) and have the main advantages that the transform results are real for real input, but they do not have the additivity property as (3). Moreover, they do not have simpler a inverse transform. For example, for the fractional cosine transform defined as (19), if

$$G_C^{\alpha}(s) = O_C^{\alpha}(g(t)) \tag{22}$$

then we cannot recover $g(t)$ directly from $O_C^{-\alpha}(G_C^{\alpha}(s))$ as (4). Instead, we must follow the process as follows.

1)

$$\begin{aligned}&G_C^{\alpha}(s) \to \{b_m | m = 0, 1, 2, 3, \ldots\}\\ &\text{where } b_m = C_m^{-1}\int_{-\infty}^{\infty} G_C^{\alpha}(s) H_m(s) e^{-s^2/2} ds\\ &C_m \text{ is defined as (11).}\end{aligned} \tag{23}$$

2)

$$a_m = \frac{b_m}{\cos\left(\frac{m\alpha\pi}{2}\right)}. \tag{24}$$

3)

$$g(t) = \sum_{m=0}^{\infty} a_m \cdot e^{-t^2/2} \cdot H_m(t). \tag{25}$$

Because the fractional cosine, sine, and Hartley transforms defined as (19)–(21) have no simpler inverse transforms, it is hard to use them for the applications where we must implement the inverse transform, such as filter design and signal analysis.

The goal of this paper is to find the FRCT, FRST, and FRHTs that have more elegant mathematical properties and are more suitable for practical applications. We will discuss the topics in the following.

1) In Section II, we will derive the FRCT, FRST, and FRHTs that satisfy the boundary property (2) and additivity property (3). Since they have additivity property, they all have simpler inverse transforms.
2) Then, as the FRFT can be further generalized into the linear canonical transform (LCT), in Section III, we will further generalize the FRCT/FRST into the canonical cosine/sine transforms (CCT/CST).
3) In Section IV, we will derive the simplified fractional cosine, sine, and Hartley transforms (SFRCT, SFRST, and SFRHTs). They are similar to the FRCT, FRST, and FRHT, but there are fewer complexities of computation. Besides, for real input, the output is also real. They are very useful for processing pure real/imaginary functions.
4) In Section V, we will derive the properties of the generalized cosine, sine, and Hartley transforms introduced in Sections II–IV.
5) In Section VI, we will introduce some digital implementation methods.
6) In Section VII, we will discuss the applications of the generalized cosine, sine, and Hartley transforms derived in Sections II–IV. They can be substituted for the FRFT and LCT when we deal with the pure real/imaginary functions or even/odd functions and are suitable for one-sided signal processing.

II. DERIVATION OF FRACTIONAL COSINE, SINE, AND HARTLEY TRANSFORMS

In [3], the authors had discussed that $\{\exp(-t^2/2) \cdot H_m(t) | m = 0, 1, 2, 3, \ldots, H_m(t)$ is the Hermite polynomial of degree $m\}$ is the eigenfunctions set of the FT, and the corresponding eigenvalues are $\exp(-jm\pi/2)$

$$O_F\left(e^{-jt^2/2} \cdot H_m(t)\right) = e^{-j\cdot m\pi/2} \cdot e^{-j\omega^2/2} \cdot H_m(\omega). \tag{26}$$

Since cosine, sine, and Hartley transforms have the following relations with Fourier transform:

$$\begin{aligned}O_{\cos}(g(t)) =& \frac{(G(\omega) + G(-\omega))}{2}\\ O_{\sin}(g(t)) =& j\frac{(G(\omega) - G(-\omega))}{2}\end{aligned} \tag{27}$$

$$\begin{aligned}O_{\text{Hartley}}(g(t)) =& O_{\cos}(g(t)) + O_{\sin}(g(t))\\ =& \frac{1+j}{2} \cdot G(\omega) + \frac{1-j}{2} \cdot G(-\omega)\end{aligned} \tag{28}$$

where $G(w) = FT(g(t))$, and from the fact that $H_m(t)$ is an even function if m is even and $H_m(t)$ is an odd function if m odd, we can conclude the following.

1) When m is even, then $\exp(-t^2/2) \cdot H_m(t)$ is the eigenfunction of cosine transform.
2) When m is odd, then $\exp(-t^2/2) \cdot H_m(t)$ is the eigenfunction of sine transform.
3) For all non-negative integer m, $\exp(-t^2/2) \cdot H_m(t)$ is the eigenfunction of the Hartley transform.

Their corresponding eigenvalues (which are denoted by $\lambda_{C,S,H}(m)$) are shown in Table I.

We can derive fractional cosine, sine, and Hartley transforms from the method of (6) and (7). Similar to the derivation process of the fractional Fourier transform (FRFT) introduced in [3] (which is listed in (10)–(14)), we can derive these fractional transforms from the process as follows.

1)

$$\begin{aligned}g(t) \to& \{a_m | m = 0, 1, 2, 3, \ldots\}\\ \text{where } a_m =& C_m^{-1}\int_{-\infty}^{\infty} g(t) H_m(t) e^{-t^2/2} \cdot dt\\ &\cdot C_m \text{ is defined as (11).}\end{aligned} \tag{29}$$

2)

$$d_m = \lambda_{C,S,H}^{\alpha}(m) \cdot a_m. \tag{30}$$

3)

$$G_{C,S,H}^{\alpha}(s) = \sum_{m=0}^{\infty} d_m \cdot e^{-s^2/2} \cdot H_m(s) \tag{31}$$

where $\lambda\alpha_{C,S,H}(m)$ is the fractional power of eigenvalues of cosine, sine, and Hartley transforms, as shown in Table II.

From (13) and (14), we can express the FRFT as follows ($G(\omega)$ in (13) is changed as $G_F^{\alpha}(s)$).

TABLE I
EIGENVALUES OF $\exp(-t^2/2) \cdot H_m(t)$ FOR COSINE, SINE, AND HARTLEY TRANSFORMS

Eigenvalues:	m is even	m is odd
$\lambda_C(m)$	$(-j)^m$	0
$\lambda_S(m)$	0	$(-j)^{m-1}$
$\lambda_H(m)$	$(-j)^m$	$(-j)^{m-1}$

TABLE II
EIGENVALUES OF $\exp(-t^2/2) \cdot H_m(t)$ FOR FRACTIONAL COSINE, SINE, AND HARTLEY TRANSFORMS

	m is even	m is odd
$\lambda_C^\alpha(m)$	$\exp(-jm\alpha\pi/2)$	0
$\lambda_S^\alpha(m)$	0	$\exp(-j(m-1)\alpha\pi/2)$
$\lambda_H^\alpha(m)$	$\exp(-jm\alpha\pi/2)$	$\exp(-j(m-1)\alpha\pi/2)$

- fractional Fourier transform:

$$G_F^\alpha(s) = \sum_{m=0}^{\infty} e^{-jm\alpha\pi/2} a_m \cdot e^{-s^2/2} \cdot H_m(s). \tag{32}$$

  From (30), (31), and Table II, we find that the fractional cosine/sine transforms can be expressed as follows.
- fractional cosine transform:

$$G_C^\alpha(s) = \sum_{m=0}^{\infty} e^{-jm\alpha\pi} a_{2m} \cdot e^{-s^2/2} H_{2m}(s). \tag{33}$$

- fractional sine transform:

$$G_S^\alpha(s) = \sum_{m=0}^{\infty} e^{-jm\alpha\pi} a_{2m+1} \cdot e^{-s^2/2} H_{2m+1}(s). \tag{34}$$

Then, because

$$\begin{aligned} \frac{(G_F^\alpha(s) + G_F^\alpha(-s))}{2} =& \sum_{m=0}^{\infty} e^{-jm\alpha\pi/2} a_m \cdot e^{-s^2/2} \\ & \cdot \frac{(H_m(s) + H_m(-s))}{2} \end{aligned} \tag{35}$$

$$\begin{aligned} \frac{(G_F^\alpha(s) - G_F^\alpha(-s))}{2} =& \sum_{m=0}^{\infty} e^{-jm\alpha\pi/2} a_m \cdot e^{-s^2/2} \\ & \cdot \frac{(H_m(s) - H_m(-s))}{2} \end{aligned} \tag{36}$$

and $H_m(t)$ is even when m is even, $H_m(t)$ is odd when m is odd; therefore, we find the fractional cosine/sine transforms have the following relations with the FRFT:

$$\begin{aligned} G_C^\alpha(s) =& \frac{(G_F^\alpha(s) + G_F^\alpha(-s))}{2} \\ G_S^\alpha(s) =& e^{j\alpha\pi/2} \frac{(G_F^\alpha(s) - G_F^\alpha(-s))}{2}. \end{aligned} \tag{37}$$

Therefore, we obtain the fractional cosine transform (FRCT) and fractional sine transform (FRST) as follows:

- *Fractional Cosine Transform (FRCT)*:

$$\begin{aligned} G_C^\alpha(s) = O_C^\alpha(g(t)) = \sqrt{\frac{1 - j\cot\phi}{2\pi}}\, e^{j(s^2/2)\cot\phi} \\ \cdot \int_{-\infty}^{\infty} \cos(\csc\phi \cdot st)\, e^{j(t^2/2)\cot\phi} g(t) \cdot dt. \end{aligned} \tag{38}$$

- *Fractional Sine Transform (FRST)*:

$$\begin{aligned} G_S^\alpha(s) =& O_S^\alpha(g(t)) = \sqrt{\frac{1 - j\cot\phi}{2\pi}} e^{j(\phi - (\pi/2))} e^{j(s^2/2)\cot\phi} \\ & \cdot \int_{-\infty}^{\infty} \sin(\csc\phi \cdot st) e^{j(t^2/2)\cot\phi} g(t) \cdot dt \end{aligned} \tag{39}$$

  where $\phi = \alpha\pi/2$. Then, we can derive the fractional Hartley transform (FRHT) from

$$O_H^\alpha(f(t)) = O_C^\alpha(f(t)) + O_S^\alpha(f(t)). \tag{40}$$

  Therefore, we obtain
- *Fractional Hartley Transform (FRHT)*: We have (41), shown at the bottom of the page, where $\phi = \alpha\pi/2$. It has relations with the FRCT, FRST, and FRFT as in

$$\begin{aligned} G_H^\alpha(s) =& G_C^\alpha(s) + G_S^\alpha(s) \\ =& \frac{1 + e^{(j\cdot\alpha\pi/2)}}{2} \cdot G_F^\alpha(s) \\ & + \frac{1 - e^{(j\cdot\alpha\pi/2)}}{2} \cdot G_F^\alpha(-s). \end{aligned} \tag{42}$$

Therefore, we have obtained the formulas of the fractional cosine, sine, and Hartley transforms. We must notice that the FRCT, FRST, and FRHT all have the periods of 2

$$\begin{aligned} O_C^\alpha(g(t)) =& O_C^{\alpha+2}(g(t)) \\ O_S^\alpha(g(t)) =& O_S^{\alpha+2}(g(t)) \\ O_H^\alpha(g(t)) =& O_H^{\alpha+2}(g(t)). \end{aligned} \tag{43}$$

It is different with the case of FRFT (FRFT has the period of 4).

Because the FRCT, FRST, and FRHT defined as (38), (39), and (41) are derived from the method of (6) and (7), they all satisfy the additivity property and boundary property. Since these

$$\begin{aligned} G_H^\alpha(s) =& O_H^\alpha(g(t)) = \sqrt{\frac{1 - j\cot\phi}{2\pi}} \cdot e^{j\cdot(s^2/2)\cot\phi} \cdot \int_{-\infty}^{\infty} e^{j\cdot(t^2/2)\cot\phi} \\ & \cdot \frac{\{(1 - je^{j\phi})\mathrm{cas}(\csc\phi \cdot st) + (1 + je^{j\phi}) \cdot \mathrm{cas}(-\csc\phi \cdot st)\}}{2} \cdot g(t)dt \end{aligned} \tag{41}$$

fractional transforms have the additivity property, they are all reversible, and their inverse transforms are

$$\begin{aligned} O_C^{-\alpha}\left(G_C^{\alpha}(s)\right) =& g(t) \text{ if } g(t) \text{ is even} \\ O_S^{-\alpha}\left(G_S^{\alpha}(s)\right) =& g(t) \text{ if } g(t) \text{ is odd} \qquad (44) \\ O_H^{-\alpha}\left(G_H^{\alpha}(s)\right) =& g(t) \text{ no matter} \\ & \text{whether } g(t) \text{ is even or odd.} \qquad (45) \end{aligned}$$

The FRCT has no odd eigenfunctions, and the FRST has no even eigenfunctions. Thus, as the original cosine and sine transforms, the odd part of input function will be lost after doing the FRCT, and the even part of input will be lost after doing the FRST. Therefore, it is better to use the FRCT to process even functions and use the FRST to process odd functions. Therefore, we can constrain the input function of FRCT to be even and define the one-sided FRCT as follows.

- *One-sided fractional cosine transform (One-sided FRCT)*:

$$G_C^{\alpha}(s) = O_C^{\alpha}(g(t)) = \sqrt{\frac{2-j2\cot\phi}{\pi}}\, e^{j(s^2/2)\cot\phi} \cdot \int_0^{\infty} e^{j(t^2/2)\cot\phi}\cos(\csc\phi\cdot st)g(t)\cdot dt. \qquad (46)$$

We can constrain the input function of FRST to be odd and define the one-sided FRST as follows.

- *One-side fractional sine transform (One-sided FRST)*:

$$G_S^{\alpha}(s) = O_S^{\alpha}(g(t)) = \sqrt{\frac{2-j2\cot\phi}{\pi}}\, e^{j(\phi-(\pi/2))} e^{j(s^2/2)\cot\phi} \cdot \int_0^{\infty} e^{j(t^2/2)\cot\phi}\sin(\csc\phi\cdot st)g(t)dt. \qquad (47)$$

We notice that the transform results of the one-sided FRCT/FRST are the same as the transform results of FRFT for even/odd functions

$$\begin{aligned} G_C^{\alpha}(s) =& G_F^{\alpha}(s) \text{ when } g(t) \text{ is even} \\ G_S^{\alpha}(s) =& e^{j\phi}G_F^{\alpha}(s) \text{ when } g(t) \text{ is odd.} \qquad (48) \end{aligned}$$

When we use the one-sided FRCT/FRST to process the even/odd functions, the complexity of computation is half of the complexity of the FRFT (because the range of integration for the one-sided FRCT, FRST is $[0,\ \infty)$). Therefore, the one-sided FRCT/FRSTs are more efficient than the FRFT when processing the even/odd functions.

III. CANONICAL COSINE AND SINE TRANSFORMS

A. *Linear Canonical Transform and its Eigenfunctions*

Canonical cosine and sine transforms are the further generalization of FRCT and FRST. They are analogous to the *linear canonical transform (LCT)* [2], [13], [14]. The LCT is the generalization of the FRFT [see (9)]. It is defined as

$$\begin{aligned} G_F^{(a,b,c,d)}(s) =& O_F^{(a,b,c,d)}(g(t)) \\ =& \sqrt{\frac{1}{j2\pi b}} \cdot e^{(j/2)(d/b)\,s^2} \\ & \cdot \int_{-\infty}^{\infty} e^{-j(s/b)t} e^{(j/2)\cdot(a/b)t^2} g(t)dt \\ \text{for } b \neq& 0 \qquad (49) \end{aligned}$$

TABLE III
FORMS OF THE EIGENFUNCTIONS OF LCT FOR ALL THE CASES

Condition	Forms of the eigenfunctions
$\lvert a+d\rvert < 2$	Hermite functions with scaling and chirp multiplication
$a+d=2,\ b\neq 0$	Periodic (or almost periodic) functions with chirp multiplication
$a+d=-2,\ b\neq 0$	Symmetric periodic (or almost periodic) functions with chirp multiplication
$a+d=2,\ b=0$	Impulse train
$a+d=-2,\ b=0$	Symmetric impulse train
$a+d>2$	Chirp convolution, chirp multiplication of scaling invariant functions
$a+d<-2$	Chirp convolution, chirp multi. of symmetric scaling invariant functions

$$\begin{aligned} G_F^{(a,b,c,d)}(s) =& O_F^{(a,b,c,d)}(g(t)) \\ =& \sqrt{d}\cdot e^{(j/2)\,cd\,s^2} g(d\cdot s) \text{ for } b=0. \qquad (50) \end{aligned}$$

The LCT satisfies the additivity property as

$$\begin{aligned} & O_F^{(a_2,b_2,c_2,d_2)}\left(O_F^{(a_1,b_1,c_1,d_1)}(g(t))\right) \\ & = O_F^{(a_3,b_3,c_3,d_3)}(g(t)) \qquad (51) \\ & \text{where } \begin{bmatrix} a_3 & b_3 \\ c_3 & d_3 \end{bmatrix} = \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix}\begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix}. \qquad (52) \end{aligned}$$

The FRFT is the special case of the LCT wherein $\{a,b,c,d\} = \{\cos\phi, \sin\phi, -\sin\phi, \cos\phi\}$:

$$\begin{aligned} O_F^{\alpha}(g(t)) =& \sqrt{e^{j\cdot\phi}} \cdot O_F^{(\cos\phi,\sin\phi,-\sin\phi,\cos\phi)}(g(t)) \\ \text{where } \phi =& \frac{\alpha\pi}{2}. \qquad (53) \end{aligned}$$

When $\{a,b,c,d\} = \{1, \lambda z/2\pi, 0, 1\}$, the LCT becomes the Fresnel transform [15]. When $\{a,b,c,d\} = \{\sigma^{-1}, 0, 0, \sigma\}$, the LCT becomes a scaling operation. Since the LCT can be viewed as the generalization of the FRFT, Fresnel transform, and scaling operation, it can extend their utilities and applications and is a useful tool for the signal process and optical system analysis.

The eigenfunctions of the LCT have been found in recent years [16], [17]. Since listing the explicit formulas of the eigenfunctions of the LCT for each case is too complicated, we just list their forms in Table III. The details about the eigenfunctions of the LCT are described in [17].

For different cases, the forms of the eigenfunctions are also different. Nevertheless, in all cases, the eigenfunctions of the LCT all have the properties of orthogonality, completion, and symmetry properties of the eigenfunctions of LCT.

There is at least one subset of the eigenfunctions of the LCT with parameters $\{a,b,c,d\}$ that forms an *orthogonal, complete, and symmetric* function set. That is, we can always find an eigenfunctions set $\{\phi_0(t), \phi_1(t), \phi_2(t), \ldots\}$ of the LCT with parameters $\{a,b,c,d\}$ that satisfies the following.

1)

$$\int_{-\infty}^{\infty} \phi_m(t)\cdot\overline{\phi_n(t)}\cdot dt = C_m\cdot\delta_{m,n} \cdot C_m \text{ is some constant.} \qquad (54)$$

2) Any function $g(t)$ can be expressed as the linear combination of $\phi_0(t), \phi_1(t), \phi_2(t), \ldots$

$$\begin{aligned} g(t) &= \sum_{n=0}^{\infty} a_n \cdot \phi_n(t) \\ \text{where } a_n &= C_m^{-1} \cdot \int_{-\infty}^{\infty} g(t) \cdot \overline{\phi_n(t)} \cdot dt. \end{aligned} \tag{55}$$

3) $\phi_0(t), \phi_1(t), \phi_2(t), \phi_3(t), \ldots$ are either symmetric or asymmetric

$$\phi_m(t) = \phi_m(-t) \text{ or } \phi_m(t) = -\phi_m(-t). \tag{56}$$

That is, $\phi_m(t)$ is either even or odd.

This property is very helpful for us to derive canonical cosine and sine transforms from the LCT.

B. Deriving the Canonical Cosine and Sine Transforms

From the above property, we know that for the LCT with any parameters set $\{a, b, c, d\}$, we can always find an orthogonal, complete, and symmetric eigenfunctions set $\{\phi_0(t), \phi_1(t), \phi_2(t), \phi_3(t), \ldots\}$. Since the eigenfunctions set is complete and symmetric, we can sort the eigenfunctions set properly so that $\phi_{2m}(t)$ is even and $\phi_{2m+1}(t)$ is odd

$$\phi_{2m}(t) = \phi_{2m}(-t), \quad \phi_{2m+1}(t) = -\phi_{2m+1}(-t). \tag{57}$$

Since the eigenfunctions set is orthogonal and complete, as is the case of the FRFT, we can express the LCT as the following process instead of (50) and (51).

1)

$$\begin{aligned} g(t) &\to \{a_m | m = 0, 1, 2, 3, \ldots\} \\ a_m &= C_m^{-1} \cdot \int_{-\infty}^{\infty} g(t) \cdot \overline{\phi_m(t)} \cdot dt \\ C_m &= \int_{-\infty}^{\infty} |\phi_m(t)|^2 dt. \end{aligned} \tag{58}$$

2)

$$\begin{aligned} b_m &= \lambda_m a_m \\ &\text{where } \lambda_m \text{ is the eigenvalue of the LCT for } \phi_m(t) \\ &\cdot O_F^{(a,b,c,d)}(\phi_m(t)) = \lambda_m \phi_m(t). \end{aligned} \tag{59}$$

3)

$$G_F^{(a,b,c,d)}(s) = \sum_{m=0}^{\infty} b_m \cdot \phi_m(s). \tag{60}$$

Then, we try to define the canonical cosine transform (CCT) with parameters $\{a, b, c, d\}$. As with the case of the fractional cosine transform (FRCT), we choose the eigenfunctions of the CCT with parameters $\{a, b, c, d\}$ as the even eigenfunctions of the LCT with parameters $\{a, b, c, d\}$, and the corresponding eigenvalues are the same. We also set the odd eigenfunctions of the LCT with parameters $\{a, b, c, d\}$ to belong to the null space of the LCT with parameters $\{a, b, c, d\}$:

$$\begin{aligned} O_C^{(a,b,c,d)}(\phi_{2m}(t)) &= \lambda_{2m} \phi_{2m}(t) \\ O_C^{(a,b,c,d)}(\phi_{2m+1}(t)) &= 0. \end{aligned} \tag{61}$$

Similarly, for the canonical sine transform (CST) with parameters $\{a, b, c, d\}$, we choose

$$\begin{aligned} O_S^{(a,b,c,d)}(\phi_{2m}(t)) &= 0 \\ O_S^{(a,b,c,d)}(\phi_{2m+1}(t)) &= \lambda_{2m+1} \phi_{2m+1}(t). \end{aligned} \tag{62}$$

The CCT and CST can be expressed as the following process.

1)

$$\begin{aligned} &g(t) \to \{a_m | m = 0, 1, 2, 3, \ldots\} \\ &\qquad \text{where } a_m, \; C_m \text{ are defined as (58).} \end{aligned} \tag{63}$$

2)

$$\begin{aligned} &b_m = \lambda_m a_m, \\ &\qquad \text{where } \lambda_m \text{ is the eigenvalue of the LCT for } \phi_m(t). \end{aligned} \tag{64}$$

3)

$$\begin{aligned} G_C^{(a,b,c,d)}(s) &= \sum_{m=0}^{\infty} b_{2m} \cdot \phi_{2m}(s) \\ G_S^{(a,b,c,d)}(s) &= \sum_{m=0}^{\infty} b_{2m+1} \cdot \phi_{2m+1}(s). \end{aligned} \tag{65}$$

The only difference among the CCT, CST, and LCT is the last step. Since

$$\begin{aligned} G_C^{(a,b,c,d)}(s) &= \sum_{m=0}^{\infty} b_{2m} \cdot \phi_{2m}(s) \\ &= \frac{\left[\sum_{m=0}^{\infty} b_m \cdot \phi_m(s) + \sum_{m=0}^{\infty} b_m \cdot \phi_m(-s)\right]}{2} \\ &= \frac{\left[G_F^{(a,b,c,d)}(s) + G_F^{(a,b,c,d)}(-s)\right]}{2} \end{aligned} \tag{66}$$

and

$$\begin{aligned} G_S^{(a,b,c,d)}(s) &= \frac{1}{2}\left[\sum_{m=0}^{\infty} b_m \phi_m(s) - \sum_{m=0}^{\infty} b_m \phi_m(-s)\right] \\ &= \frac{1}{2}\left[G_F^{(a,b,c,d)}(s) - G_F^{(a,b,c,d)}(-s)\right]. \end{aligned} \tag{67}$$

The formulas of the canonical cosine and sine transforms we derived are as follows.

- *Canonical Cosine Transform (CCT):*

$$\begin{aligned} G_C^{(a,b,c,d)}(s) &= O_C^{(a,b,c,d)}(g(t)) \\ &= \sqrt{\frac{1}{j2\pi b}}\, e^{(j/2)\,(d/b)s^2} \\ &\quad \cdot \int_{-\infty}^{\infty} \cos(\frac{s}{b}t) e^{(j/2)(a/b)\,t^2}\, g(t) dt \\ &\text{for } b \neq 0 \end{aligned} \tag{68}$$

$$\begin{aligned} G_C^{(a,b,c,d)}(s) &= O_C^{(a,b,c,d)}(g(t)) \\ &= \sqrt{d} \cdot e^{(j/2)\,cd\,s^2} g(d \cdot s) \text{ for } b = 0. \end{aligned} \tag{69}$$

- *Canonical Sine Transform (CST):*

$$\begin{aligned} G_S^{(a,b,c,d)}(s) =& O_S^{(a,b,c,d)}(g(t)) \\ =& \sqrt{\frac{1}{j2\pi b}} \cdot e^{(j/2)(d/b)s^2} \\ & \cdot \int_{-\infty}^{\infty} -j\sin(\frac{s}{b}\,t) \cdot e^{(j/2)\cdot(a/b)t^2} \\ & \cdot g(t) \cdot dt \text{for } b \neq 0 \end{aligned} \tag{70}$$

$$\begin{aligned} G_S^{(a,b,c,d)}(s) =& O_S^{(a,b,c,d)}(g(t)) \\ =& \sqrt{d} \cdot e^{(j/2)\,cd\,s^2} g(d \cdot s) \\ \text{for} b =& 0. \end{aligned} \tag{71}$$

The FRCT/FRST derived in Section II are the special cases of the CCT/CST where $\{a, b, c, d\} = \{\cos\phi, \sin\phi, -\sin\phi, \cos\phi\}$ (with the difference of some constant phase)

$$G_C^{\alpha}(s) = \sqrt{e^{j\phi}} \cdot G_C^{(\cos\phi, \sin\phi, -\sin\phi, \cos\phi)}(s) \tag{72}$$

$$\begin{aligned} G_S^{\alpha}(s) =& \sqrt{e^{j\phi}} \cdot e^{j\phi} \cdot G_S^{(\cos\phi, \sin\phi, -\sin\phi, \cos\phi)}(s) \\ \text{where } \phi =& \frac{\alpha\pi}{2}. \end{aligned} \tag{73}$$

The CCT and CST have additivity property as follows.

$$O_C^{(a_2,b_2,c_2,d_2)}\left(O_C^{(a_1,b_1,c_1,d_1)}(g(t))\right) = O_C^{(a_3,b_3,c_3,d_3)}(g(t)) \quad \text{if } g(t) \text{ is even} \tag{74}$$

$$O_S^{(a_2,b_2,c_2,d_2)}\left(O_S^{(a_1,b_1,c_1,d_1)}(g(t))\right) = O_S^{(a_3,b_3,c_3,d_3)}(g(t)) \quad \text{if } g(t) \text{ is odd} \tag{75}$$

$$\text{where } \begin{bmatrix} a_2 & b_2 \\ c_2 & d_2 \end{bmatrix} \cdot \begin{bmatrix} a_1 & b_1 \\ c_1 & d_1 \end{bmatrix} = \begin{bmatrix} a_3 & b_3 \\ c_3 & d_3 \end{bmatrix}. \tag{76}$$

We can now prove them.

(Proof of the additivity properties of CCT and CST): Since

$$G_C^{(a,b,c,d)}(-s) = \sqrt{-1} \cdot G_F^{(-a,-b,-c,-d)}(s) \tag{77}$$

from (66), we obtain

$$\begin{aligned} & O_C^{(a,b,c,d)}(g(t)) \\ & = \frac{\left[O_F^{(a,b,c,d)}(g(t)) + \sqrt{-1} \cdot O_F^{(-a,-b,-c,-d)}(g(t))\right]}{2} \end{aligned} \tag{78}$$

where we have (78a), shown at the bottom of the page. Then, we use the additivity property of LCT and obtain (79), shown at the bottom of the page. Therefore, we have proved the additivity property of CCT (74). We can use similar process to prove the additivity property of CST (75).

Since CCT and CST are all additive, they are all reversible, and the inverse operations of the CCT and CST with parameters $\{a, b, c, d\}$ is just the CCT and CST with parameters $\{d, -b, -c, a\}$

$$O_C^{(d,-b,-c,-d)}\left(G_C^{(a,b,c,d)}(s)\right) = g(t) \text{ if } g(t) \text{ is even} \tag{80}$$

$$O_S^{(d,-b,-c,a)}\left(G_S^{(a,b,c,d)}(s)\right) = g(t) \text{ if } g(t) \text{ is odd.} \tag{81}$$

Although we can define the CCT and CST, the canonical Hartley transform (CHT) is very hard to define. We cannot define it as the summation of the CCT and the CST because the result has no additivity property. It is also hard to apply the method as Figs. 1 and 2. Nevertheless, since the original Hartley transform has the relation with the FT as

$$G_H(\omega) = (1 + j) \cdot \frac{[G(\omega) - jG(-\omega)]}{2} \tag{82}$$

we suggest that we can define the CHT as

$$G_H^{(a,b,c,d)}(s) = \frac{\left[G_F^{(a,b,c,d)}(s) \pm \sqrt{-1} \cdot G_F^{(a,b,c,d)}(-s)\right]}{2}. \tag{83}$$

We can prove that the CHT defined above has an additivity property, but the FRHT defined in Section II is not its special case.

$$\begin{aligned} & O_C^{(a_2,b_2,c_2,d_2)}\left(O_C^{(a_1,b_1,c_1,d_1)}(g(t))\right) \\ & = \frac{\left[\left(O_F^{(a_2,b_2,c_2,d_2)} + \sqrt{-1} \cdot O_F^{(-a_2,-b_2,-c_2,-d_2)}\right)\left(O_F^{(a_1,b_1,c_1,d_1)} + \sqrt{-1} \cdot O_F^{(-a_1,-b_1,-c_1,-d_1)}(g(t))\right)\right]}{4}. \end{aligned} \tag{78a}$$

$$\begin{aligned} & O_C^{(a_2,b_2,c_2,d_2)}\left(O_C^{(a_1,b_1,c_1,d_1)}(g(t))\right) \\ & = \frac{\left[O_F^{(a_3,b_3,c_3,d_3)}(g(t)) + \sqrt{-1} \cdot O_F^{(-a_3,-b_3,-c_3,-d_3)}(g(t)) + \sqrt{-1} \cdot O_F^{(-a_3,-b_3,-c_3,-d_3)}(g(t)) + O_F^{(a_3,b_3,c_3,d_3)}(g(t))\right]}{4} \\ & O_C^{(a_2,b_2,c_2,d_2)}\left(O_C^{(a_1,b_1,c_1,d_1)}(g(t))\right) \\ & = \frac{\left[O_F^{(a_3,b_3,c_3,d_3)}(g(t)) + \sqrt{-1} \cdot O_F^{(-a_3,-b_3,-c_3,-d_3)}(g(t))\right]}{2} \\ & = O_C^{(a_3,b_3,c_3,d_3)}(g(t)) \end{aligned} \tag{79}$$

C. *One-Sided Canonical Cosine, Sine Transforms*

We can prove that for even/odd inputs, the transform results of the CCT/CST are the same as the transform results of the linear canonical transform (LCT)

$$O_C^{(a,b,c,d)}(g(t)) = O_F^{(a,b,c,d)}(g(t)) \text{ if } g(t) \text{ is even} \tag{84}$$

$$O_S^{(a,b,c,d)}(g(t)) = O_F^{(a,b,c,d)}(g(t)) \text{ if } g(t) \text{ is odd .} \tag{85}$$

As with the case of the FRCT and FRST, the CCT is suitable to deal with the even functions, and the CST is suitable to deal with the odd functions. We can constrain the input function of the CCT to be even and constrain the input function of the CST to be odd and define the one-sided CCT and the one-sided CST as follows.

- *One-sided canonical cosine transform (one-sided CCT)*

$$\begin{aligned} G_C^{(a,b,c,d)}(s) =& O_C^{(a,b,c,d)}(g(t)) \\ =& \sqrt{\frac{2}{j\pi b}} e^{(j/2)\,(d/b)\,s^2} \\ & \cdot \int_0^{\infty} \cos\left(\frac{s}{b}\,t\right)\, e^{(j/2)\cdot(a/b)\,t^2}\, g(t)\cdot dt \\ & \text{for } b \neq 0 \end{aligned} \tag{86}$$

$$\begin{aligned} G_C^{(a,b,c,d)}(s) =& O_C^{(a,b,c,d)}(g(t)) \\ =& \sqrt{d} \cdot e^{(j/2)\,cd\,s^2} g(d \cdot s) \text{ for } b = 0. \end{aligned} \tag{87}$$

- *One-sided canonical sine transform (one-sided CST)*

$$\begin{aligned} G_S^{(a,b,c,d)}(s) =& O_S^{(a,b,c,d)}(g(t)) \\ =& \sqrt{\frac{2}{j\pi b}}\, e^{(j/2)\,(d/b)\,s^2} \\ & \cdot \int_0^{\infty} -j\sin\left(\frac{s}{b}\,t\right) e^{(j/2)(a/b)\,t^2}\, g(t)dt \\ & \text{for } b \neq 0 \end{aligned} \tag{88}$$

$$\begin{aligned} G_S^{(a,b,c,d)}(s) =& O_S^{(a,b,c,d)}(g(t)) \\ =& \sqrt{d} \cdot e^{(j/2)\,cd\,s^2} g(d \cdot s) \text{ for } b = 0. \end{aligned} \tag{89}$$

Since the range of integration for the one-sided CCT/CST is just $[0,\ \infty]$ and not $(-\infty,\ \infty)$, using the one-sided CCT/CST is more efficient than using the LCT to deal with the even/odd functions.

IV. SIMPLIFIED FRCT, FRST, AND FRHTS

The FRCT, FRST, FRHT, CCT, and CST derived in Sections II and III all have the additivity property and are suitable to deal with the even and odd functions. Nevertheless, for real input, the transform results of these transforms are not real. The original FRCT, FRST, and FRHTs are suitable to deal with the real functions because for real input, the output is also real. This advantage does not exist for the transforms derived in Sections II and III.

In this section, we will derive the SFRCT, SFRST, and SFRHTs. Their performances are very similar to those of the FRCT, FRST, and FRHT and have the advantage of real-input–real-output.

A. *Review of SFRFT*

Before discussing the SFRCT, SFRST, and SFRHT, we first describe the SFRFT. It was introduced in [19] recently. In [19], we had illustrated that although the LCT has four parameters $\{a, b, c, d\}$, in many applications, only the value of a/b really effects the performance. Therefore, if $a_1/b_1 = a_2/b_2$, then the LCT with parameters $\{a_1, b_1, c_1, d_1\}$ will have the same effects as the LCT with parameters $\{a_2, b_2, c_2, d_2\}$ for many applications (such as filter design).

We remember that the FRFT is a special case of the LCT where $\{a, b, c, d\} = \{\cos\phi, \sin\phi, -\sin\phi, \cos\phi\}(\phi = \alpha\pi/2)$. Therefore, for the FRFT, the value of a/b is $\cot\phi$. Thus, we can define the SFRFT as the special case of the LCT where $a/b = \cot\phi$; then, it will have the same effects as the FRFT. Then, we choose other parameters properly such that the SFRFT will be simpler than the FRFT for digital implementation, optical system implementation, gradient-index medium implementation, etc. In [19], we defined several types of SFRFT. The most important one is the following.

- *SFRFT:*

$$\begin{aligned} G_{SF}^{\alpha}(s) =& O_{SF}^{\alpha}(g(t)) \\ =& \sqrt{\frac{1}{j2\pi}} \int_{-\infty}^{\infty} e^{-j\cdot st} e^{(j/2)\cdot\cot\phi\, t^2} g(t)dt \\ & \text{where } \phi = \frac{\alpha\pi}{2}. \end{aligned} \tag{90}$$

It is the special case of the LCT with parameters $\{a, b, c, d\} = \{\cot\phi, 1, -1, 0\}(\phi = \alpha\pi/2)$. It has the same effects as the FRFT and has two important advantages.

1) Compared with the original FRFT [see (9)], for the SFRFT defined as (90), the outer chirp term is removed, and the complexity of design and computation is reduced.
2) Suppose that Δ_t and Δ_s are the sampling intervals in time and frequency domains. For the case of FRFT, if we try to make the term $\exp(-j\csc\phi \cdot st)$ in (9) become the kernel of the DFT, then

$$\Delta_t \cdot \Delta_s = \frac{2\pi\sin\phi}{P}. \tag{91}$$

Since $\Delta_t \cdot \Delta_s$ varies with ϕ, we must adjust the value of Δ_t or Δ_s when ϕ is changed, but for the SFRFT, to make the term $\exp(-j \cdot st)$ in (90) become the kernel of the DFT, we just have to set

$$\Delta_t \cdot \Delta_s = \frac{2\pi}{P}. \tag{92}$$

This does not vary with ϕ, and we do not have to adjust the value of Δ_t and Δ_s when ϕ is changed.

Therefore, for digital implementation, the SFRFT defined as (90) is simpler than the original FRFT. It can substitute the FRFT for many applications.

B. *SFRCT*

We will derive the SFRCT. The SFRCT we derive should have at least the two advantages: i) It is efficient for digital implementation, and ii) for real input, the output is also real.

We will try to achieve these two goals. The SFRFT with parameter α is the special case of the LCT where $\{a, b, c, d\} = \{\cot\phi, 1, -1, 0\}(\phi = \alpha\pi/2)$. For the CCT [see (68)], when $\{a, b, c, d\} = \{\cot\phi, 1, -1, 0\}$, then

$$G_C^{(\cot\phi,1,-1,0)}(s) = \sqrt{\frac{1}{j2\pi}} \int_{-\infty}^{\infty} \cos(st) \cdot e^{(j/2)\cot\phi \cdot t^2} \cdot g(t) \cdot dt. \quad (93)$$

We can define the SFRCT as (we suppose that the input $g(t)$ is a real function)

$$\begin{aligned} G_{SC}^{\alpha}(s) =& \mathrm{Re}\left(\sqrt{j} \cdot G_C^{(\cot\phi,1,-1,0)}(s)\right) \\ =& \sqrt{\frac{1}{2\pi}} \int_{-\infty}^{\infty} \cos(st) \cos\left(\frac{1}{2}\cot\phi \cdot t^2\right) \\ & \cdot g(t) \cdot dt. \end{aligned} \quad (94)$$

Since, as with the FRCT and CCT we will usually use the SFRCT to process even function, in (94), it is convenient to change the range of integration from $(-\infty, \infty)$ into $[0, \infty]$. Therefore, we obtain the following.

- *SFRCT of type 1:*

$$\begin{aligned} G_{SC}^{\alpha}(s) =& O_{SC}^{\alpha}(g(t)) \\ =& \sqrt{\frac{2}{\pi}} \int_{0}^{\infty} \cos(st) \cos\left(\frac{1}{2}\cot\phi \cdot t^2\right) g(t) dt \\ \text{where } \phi =& \frac{\alpha\pi}{2}. \end{aligned} \quad (95)$$

The SFRCT has very close relation with the SFRFT, especially when the input is a real, even function. We now prove this statement. Suppose $g(t)$ is a real, even function. From the even-input–even-output property of the SFRFT:

$$G_{SF}^{\alpha}(s) = G_{SF}^{\alpha}(-s) \text{ if } g(t) = g(-t) \quad (96)$$

we can conclude that $G_{SF}^{\alpha}(s)$, and hence $\sqrt{j}G_{SF}^{\alpha}(s)$, are all even functions. Therefore

$$\begin{aligned} \mathrm{Re}\left(\sqrt{j} \cdot G_{SF}^{\alpha}(s)\right) =& \mathrm{Re}\left(\sqrt{j} \cdot G_{SF}^{\alpha}(-s)\right) \\ 2\mathrm{Re}\left(\sqrt{j} \cdot G_{SF}^{\alpha}(s)\right) =& \mathrm{Re}\left(\sqrt{j} \cdot G_{SF}^{\alpha}(s)\right) \\ & + \mathrm{Re}\left(\sqrt{j} \cdot G_{SF}^{\alpha}(-s)\right) \\ =& \sqrt{\frac{1}{2\pi}} \int_{-\infty}^{\infty} \left[\cos\left(-st + \frac{1}{2}\cot\phi t^2\right) \right. \\ & \left. + \cos\left(st + \frac{1}{2}\cot\phi t^2\right)\right] g(t) \cdot dt \\ =& 2\sqrt{\frac{1}{2\pi}} \int_{-\infty}^{\infty} \cos(st) \cos\left(\frac{\cot\phi t^2}{2}\right) \\ & \cdot g(t) \cdot dt \\ \mathrm{Re}\left(\sqrt{j} \cdot G_{SF}^{\alpha}(s)\right) =& \sqrt{\frac{2}{\pi}} \int_{0}^{\infty} \cos(st) \cos\left(\frac{\cot\phi t^2}{2}\right) \\ & \cdot g(t) \cdot dt = G_{SC}^{\alpha}(s). \end{aligned} \quad (97)$$

Thus, if the input is a real, even function, then the transform result of the SFRCT is the same as the real part of $\sqrt{j} \cdot G_{SF}^{\alpha}(s)$, where $G_{SF}^{\alpha}(s)$ is the transform result of the SFRFT. Therefore, the SFRCT has a very close relation with the SFRFT, especially when the input is real and even.

We remember that if the input function is a real, even function, then the transform result of the original cosine transform is the same as the transform result of the FT. This means that the original cosine transform is suitable to deal with the real, even functions. Similarly, from the relation of (97), we can conclude that the SFRCT is also suitable to deal with the real, even functions. Suppose, for digital implementation, that the sampling points of the input function $g(t)$ are

$$g(n\Delta_t), \quad n = -N, -N+1, \ldots, 0, 1, 2, \ldots, N-1, N. \quad (98)$$

Then, in the case where $g(t)$ is real and even, we just have to store the values of

$$\mathrm{Re}(g(n\Delta_t)), \quad n = 0, 1, 2, \ldots, N-1, N \quad (99)$$

and there are only $N + 1$ real values to store. For the SFRFT of $g(t)$, because the transform result is usually a complex, even function, we must store the values of

$$\mathrm{Re}\left(G_{SF}^{\alpha}(n\Delta_s)\right), \quad \mathrm{Im}\left(G_{SF}^{\alpha}(n\Delta_s)\right), \quad n = 0, 1, 2, \ldots, N-1, N \quad (100)$$

and there are $2N + 2$ real values to store (twice of the input). It is overdetermined and wastes the storage and computation time. In contrast, for the case of SFRCT, we just have to store

$$\mathrm{Re}\left(G_{SC}^{\alpha}(n\Delta_s)\right), \quad n = 0, 1, 2, \ldots, N-1, N \quad (101)$$

and there are totally $N + 1$ real values to store (the same as the input). It is very reasonable that the storage required for the transform result is the same as the storage required for the input because it means no information loss and no redundancy. Since the degree of freedom is halved, using the SFRCT to deal with the real, even functions is more efficient than using the SFRFT.

In Section IV-A, we have stated for digital implementation that the SFRFT is more efficient than the FRFT. Here, we have proved that the SFRCT is even more efficient than the SFRFT if the input is real and even. Using the SFRCT to process the real, even functions is better than using the SFRFT, and hence, it is better than using the FRFT.

The SFRCT defined as (95) has no additivity property. In mathematics, it does not do as well as the FRCT introduced in Section II. We derived the SFRCT mainly from the consideration of practical applications. Although the SFRCT has no additivity property, it is still easy to recover the original function from the transform result.

- *Inverse SFRCT of type 1:*

$$\begin{aligned} g(t) =& O_{ISC}^{\alpha}\left(G_{SC}^{\alpha}(s)\right) \\ =& \sqrt{\frac{2}{\pi}} \sec\left(\frac{\cot\phi t^2}{2}\right) \int_{0}^{\infty} \cos(st) G_{SC}^{\alpha}(s) ds \\ \text{where } \phi =& \frac{\alpha\pi}{2}. \end{aligned} \quad (102)$$

In (102), it is possible that $\sec(\cot\phi \cdot t^2/2) \to \infty$ for some t, but this problem is not important for the digital implementation because we can choose the sampling interval Δ_t carefully such that $\cos(\cot\phi \cdot n^2\Delta_t^2/2) \neq 0$ for all n.

The SFRCT we defined in (95) is similar to the FRCT defined by [9] [which is shown in (19)]. They all have the property of real-input–real-output, and all have no additivity property. Nevertheless, there are two key differences between them. The first difference is that it is easy to recover the original function from the transform result of SFRCT [just using (102)], but for the FRCT defined as (19), we must follow the complicated process from (23) to (25) to recover the original function.

The second difference is about the digital computation. For the FRCT defined as (19), we must implement it by the FRFT. For the SFRCT defined as (95), we can implement it without resorting to the FRFT. We can use a more efficient algorithm, such as applying the fast algorithm of the DCT, and the complexity of computation can be much less (see Section VI).

In fact, there are also other types of the SFRCT. In (94), we define the SFRCT as the real part of $\sqrt{j}\cdot G_C(\cot\phi,1,-1,0)(s)$. We can also take its imaginary part and define the SFRCT of type 2 as follows.

- *SFRCT of type 2:*

$$\begin{aligned} G^{\alpha}_{SC''}(s) =& O^{\alpha}_{SC''}(g(t)) \\ =& \sqrt{\frac{2}{\pi}}\int_0^{\infty}\cos(st)\sin(\frac{1}{2}\cot\phi\cdot t^2)g(t)dt \\ \text{where } \phi =& \frac{\alpha\pi}{2}. \end{aligned} \tag{103}$$

Its inverse transform is

$$\begin{aligned} g(t) =& O^{\alpha}_{ISC''}\left(G^{\alpha}_{SC''}(s)\right) \\ =& \sqrt{\frac{2}{\pi}}\csc\left(\frac{\cot\phi\cdot t^2}{2}\right) \\ &\cdot\int_0^{\infty}\cos(st)G^{\alpha}_{SC''}(s)ds. \end{aligned} \tag{104}$$

We can prove

$$\text{Im}\left(\sqrt{j}\cdot G^{\alpha}_{SF}(s)\right) = G^{\alpha}_{SC''}(s) \quad \text{if the input } g(t) \text{ is real and even.} \tag{105}$$

As in the case of the SFRCT of type 1, the SFRCT of type 2 also has a very close relation with the SFRFT when the input is a real, even function. The SFRCT of type 1 preserves the real part of $\sqrt{j}\cdot G^{\alpha}_{SF}(s)$, and the SFRCT of type 2 preserves its imaginary part. They all treat the other part as the redundancy and make the degree of freedom of the transform result the same as that of the input when the input is a real, even function. They can deal with real, even functions more efficiently than the SFRFT and the FRFT.

C. SFRST and SFRHTs

As with the SFRCT, we can define the SFRST from the CST with parameters $\{\cot\phi,1,-1,0\}$ and take the real/imaginary parts

$$G^{\alpha}_{SS}(s) = \text{Re}\left(\sqrt{j}\cdot G_S^{(\cot\phi,1,-1,0)}(s)\right). \tag{106}$$

We also suppose the input $g(t)$ is a real, odd function. Then together, with (70), we obtain the following.

- *SFRST of type 1:*

$$\begin{aligned} G^{\alpha}_{SS}(s) =& O^{\alpha}_{SS}(g(t)) \\ =& \sqrt{\frac{2}{\pi}}\int_0^{\infty}\sin(st)\sin\left(\frac{1}{2}\cot\phi\cdot t^2\right)g(t)dt \\ \text{where } \phi =& \frac{\alpha\pi}{2}. \end{aligned} \tag{107}$$

- *SFRST of type 2:*

$$\begin{aligned} G^{\alpha}_{SS''}(s) =& O^{\alpha}_{SS''}(g(t)) \\ =& \sqrt{\frac{2}{\pi}}\int_0^{\infty}\sin(st)\cos\left(\frac{1}{2}\cot\phi\cdot t^2\right)g(t)dt \\ \text{where } \phi =& \frac{\alpha\pi}{2}. \end{aligned} \tag{108}$$

In addition, their inverse transforms are

$$\begin{aligned} g(t) =& O^{\alpha}_{ISS}\left(G^{\alpha}_{SS}(s)\right) \\ =& \sqrt{\frac{2}{\pi}}\csc\left(\frac{\cot\phi\cdot t^2}{2}\right) \\ &\cdot\int_0^{\infty}\sin(st)\,G^{\alpha}_{SS}(s)ds \end{aligned} \tag{109}$$

$$\begin{aligned} g(t) =& O^{\alpha}_{ISS''}\left(G^{\alpha}_{SS''}(s)\right) \\ =& \sqrt{\frac{2}{\pi}}\sec\left(\frac{\cot\phi\cdot t^2}{2}\right) \\ &\cdot\int_0^{\infty}\sin(st)\,G^{\alpha}_{SS''}(s)ds. \end{aligned} \tag{110}$$

Thus, although the SFRST has no additivity property, from (109) and (110), it is still easy to recover the original function from the transform result of the SFRST.

We can prove that if $g(t)$ is a real, odd function, then the transform results of the SFRFT of type 1 and type 2 and the transform result of the SFRST will have the relation

$$\begin{aligned} G^{\alpha}_{SS}(s) =& \text{Re}\left(\sqrt{j}\cdot G^{\alpha}_{SF}(s)\right) \\ G^{\alpha}_{SS''}(s) =& -\text{Im}\left(\sqrt{j}\cdot G^{\alpha}_{SF}(s)\right) \end{aligned} \tag{111}$$

when $g(t)$ is a real, odd function. The transform result of the SFRFT is real if the input is a real function. In contrast, the transform results of the SFRFT and FRFT are complex even when the input is real. Therefore, when processing the real, odd functions, we can use the SFRST instead of the SFRFT and FRFT to improve the efficiency.

Then, we try to define the SFRHT. Since the original Hartley transform is the summation of the original cosine and sine transforms, we can also define the SFRHT by a similar way. We define the SFRHT of type 1 as the summation of the SFRCT of type 1 and the SFRST of type 2 (since they all have the term of $\cos(\cot\phi\cdot t^2/2)$) and define the SFRHT of type 2 as the summation of the SFRCT of type 2 and the SFRST of type 1

$$\begin{aligned} G^{\alpha}_{SH}(s) =& G^{\alpha}_{SC}(s) + G^{\alpha}_{SS''}(s) \\ G^{\alpha}_{SH''}(s) =& G^{\alpha}_{SC''}(s) + G^{\alpha}_{SS}(s). \end{aligned} \tag{112}$$

We thus have the following.

- *SFRHT of type 1:*

$$\begin{aligned} G_{SH}^{\alpha}(s) =& O_{SH}^{\alpha}(g(t)) \\ =& \sqrt{\frac{1}{2\pi}} \int_{-\infty}^{\infty} \operatorname{cas}(st) \cos\left(\frac{1}{2}\cot\phi \cdot t^2\right) g(t)dt \\ \text{where } \phi =& \frac{\alpha\pi}{2}. \end{aligned} \tag{113}$$

- *SFRHT of type 2:*

$$\begin{aligned} G_{SH''}^{\alpha}(s) =& O_{SH''}^{\alpha}(g(t)) \\ =& \sqrt{\frac{1}{2\pi}} \int_{-\infty}^{\infty} \operatorname{cas}(st) \sin\left(\frac{1}{2}\cot\phi \cdot t^2\right) g(t)dt \\ \text{where } \phi =& \frac{\alpha\pi}{2}. \end{aligned} \tag{114}$$

Here, we change the range of integration from $[0, \infty)$ back into $(-\infty, \infty)$ since we no longer constrain the input $g(t)$ to be even or odd. The inverse transforms of the SFRHT of types 1 and 2 are

$$\begin{aligned} g(t) =& O_{SH}^{\alpha}\left(G_{SH}^{\alpha}(s)\right) \\ =& \sqrt{\frac{1}{2\pi}} \sec\left(\frac{\cot\phi \cdot t^2}{2}\right) \\ & \cdot \int_{-\infty}^{\infty} \operatorname{cas}(st)\, G_{SH}^{\alpha}(s)ds \end{aligned} \tag{115}$$

$$\begin{aligned} g(t) =& O_{SH''}^{\alpha}\left(G_{SH''}^{\alpha}(s)\right) \\ =& \sqrt{\frac{1}{2\pi}} \csc\left(\frac{\cot\phi \cdot t^2}{2}\right) \\ & \cdot \int_{-\infty}^{\infty} \operatorname{cas}(st)\, G_{SH''}^{\alpha}(s)ds. \end{aligned} \tag{116}$$

If the input function $g(t)$ is real, then the transform results of the SFRHT of types 1 and 2 and the transform result of the SFRFT will have the relations such as

$$\begin{aligned} G_{SH}^{\alpha}(s) =& \operatorname{Re}\left(\sqrt{j} \cdot \operatorname{Even}\left(G_{SF}^{\alpha}(s)\right)\right) \\ & - \operatorname{Im}\left(\sqrt{j} \cdot \operatorname{Odd}\left(G_{SF}^{\alpha}(s)\right)\right) \\ & \text{when } g(t) \text{ is real} \\ G_{SH''}^{\alpha}(s) =& \operatorname{Im}\left(\sqrt{j} \cdot \operatorname{Even}\left(G_{SF}^{\alpha}(s)\right)\right) \\ & + \operatorname{Re}\left(\sqrt{j} \cdot \operatorname{Odd}\left(G_{SF}^{\alpha}(s)\right)\right) \\ & \text{when } g(t) \text{ is real.} \end{aligned} \tag{117}$$

Therefore, the SFRHT has very close relation with the SFRFT. The SFRHT has the advantage of real-input–real-output. In contrast, the SFRFT and FRFT do not have this advantage and require more storage and more computation time to process the real functions. Therefore, when dealing with the real functions, it is more efficient to use the SFRHT instead of the SFRFT and FRFT.

In summary, the advantages of the SFRCT, SFRST, and SFRHT are the following.

1) *For the real input, the output is also a real function.* Similarly, for the pure imaginary input, the output is also a pure imaginary function.
2) The amount of required storage can be saved. When we use the SFRCT, SFRST, and SFRHT to deal with pure real/imaginary functions, the amount of storage required is only half of the storage required for SFRFT and FRFT.
3) *It is easy to recover the original function from the transform results of the SFRCT, SFRST, and SFRHT.* This advantage is very important for the applications; we must do both the forward and the inverse transforms (such as filter design).
4) *They are very efficient for digital implementation.* In Section VI, we will show, with the aid of the fast algorithm of the DCT, DST, and DHT, that the complexities of digital implementation of the SFRCT, SFRST, and SFRHT are much less than those of the FRCT, FRST, FRHT, which are defined as (19) and (21). They are also more efficient than all the fractional and canonical transforms introduced in Sections II and III for digital implementation, especially for real input functions.

V. PROPERTIES

In this section, we will discuss the properties of the CCT and SFRCT. Most properties of the FRCT can be obtained from the properties of the CCT by substituting $\{a, b, c, d\}$ as $\{\cot\phi, 1, -1, 0\}$, where $\phi = \alpha\pi/2$. The properties of the FRST, FRHT, and CST are similar to the properties of the FRCT and CCT. The properties of the SFRST and SFRHT are similar to the properties of the SFRCT.

We first list the properties of the CCT in Table IV.

Although the properties in Table IV are derived for the two-sided CCT, if we constrain the input function of the one-sided CCT to be even, then these properties can also be applied to the one-sided CCT defined as (86). We notice that for some properties of the CCT, such as the differentiation property, we cannot just use the CCT to describe it. We must describe them by the CST. In Table IV, Properties 5 and 6 can be combined as

$$\begin{aligned} & O_C^{(a,b,c,d)}(h \cdot g'(t) + k \cdot (-jt) \cdot g(t)) \\ & \quad = h_1 \cdot G_S'^{(a,b,c,d)}(s) + k_1 \cdot (-js) \cdot G_S^{(a,b,c,d)}(s) \end{aligned} \tag{118}$$

where

$$\begin{bmatrix} h_1 \\ k_1 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} h \\ k \end{bmatrix}. \tag{119}$$

Therefore, the differentiation-multiplication operations pair has a very close relation with the CCT.

Then, we list the properties of the SFRCT in Table V (O_{sc}^{α} means the SFRCT of type 1, and $O_{sc'}^{\alpha}$ means the SFRCT of type 2). In Properties 1, 5, and 6, the input function $g(t)$ is constrained to be real and even, and in Properties 2–4, the input $g(t)$ is constrained to be real and odd.

We notice that the modulation property (Properties 1 and 2) of the SFRCT is independent of α. If we multiply the input function by $\cos(\eta t)$ or $\sin(\eta t)$, then whatever the value of α is, in frequency domain, it will always cause the displacement of η. For the case of the CCT and FRCT, from Properties 3 and 4 in

TABLE IV
PROPERTIES OF THE CCT

(1) Conjugation	$\overline{O_C^{(a,b,c,d)}(g(t))} = O_C^{(a,-b,-c,d)}\left(\overline{g(t)}\right)$
(2) Time shift property for $g(t-\eta)+g(t+\eta)$	$O_C^{(a,b,c,d)}(g(t-\tau)+g(t+\tau))$ $= e^{-jac\cdot\tau^2/2}\left[G_C^{(a,b,c,d)}(s-a\tau)\cdot e^{-jc\cdot\tau\cdot s} + G_C^{(a,b,c,d)}(s+a\tau)\cdot e^{-jc\cdot\tau\cdot s}\right]$
(3) Modulation by $\cos(\eta\cdot t)$	$O_C^{(a,b,c,d)}(\cos(\eta t)\cdot g(t))$ $= e^{-jbd\cdot\eta^2/2}\left[G_C^{(a,b,c,d)}(s-b\eta)e^{jd\cdot\eta\cdot s} + G_C^{(a,b,c,d)}(s+b\eta)e^{-jd\cdot\eta\cdot s}\right]/2$
(4) Modulation by $\sin(\eta\cdot t)$	$O_C^{(a,b,c,d)}(\sin(\eta t)\cdot g(t))$ $= -je^{-jbd\cdot\eta^2/2}\left[G_S^{(a,b,c,d)}(s-b\eta)e^{jd\cdot\eta\cdot s} + G_S^{(a,b,c,d)}(s+b\eta)e^{-jd\cdot\eta\cdot s}\right]/2$
(5) Differentiation	$O_C^{(a,b,c,d)}(g'(t)) = a\cdot G_S'^{(a,b,c,d)}(s) - c\cdot js\cdot G_S^{(a,b,c,d)}(s)$
(5a) n^{th} order (n is odd) differentiation	$O_C^{(a,b,c,d)}\left(g^{(n)}(t)\right) = \left[a^2\frac{d^2}{ds^2} - 2ac\cdot js\cdot\frac{d}{ds} - ac - c^2s^2\right]^{\frac{n-1}{2}}$ $\cdot\left(a\cdot G_S'^{(a,b,c,d)}(s) - c\cdot js\cdot G_S^{(a,b,c,d)}(s)\right)$
(5b) n^{th} order (n is even) differentiation	$O_C^{(a,b,c,d)}\left(g^{(n)}(t)\right) = \left[a^2\frac{d^2}{ds^2} - 2ac\cdot js\cdot\frac{d}{ds} - jac - c^2s^2\right]^{\frac{n}{2}}\cdot G_C^{(a,b,c,d)}(s)$
(6) Multiplication by t	$O_C^{(a,b,c,d)}(t\cdot g(t)) = d\cdot s\cdot G_S^{(a,b,c,d)}(s) + jb\cdot G_S'^{(a,b,c,d)}(s)$
(6a) Multiplication by t^n (n is odd)	$O_C^{(a,b,c,d)}(t^n g(t)) = \left[-b^2\frac{d^2}{ds^2} + 2bd\cdot js\cdot\frac{d}{ds} + jbd + d^2s^2\right]^{\frac{n-1}{2}}$ $\cdot\left(jb\cdot G_S'^{(a,b,c,d)}(s) + d\cdot s\cdot G_S^{(a,b,c,d)}(s)\right)$
(6b) Multiplication by t^n (n is even)	$O_C^{(a,b,c,d)}(t^n g(t)) = \left[-b^2\frac{d^2}{ds^2} + 2bd\cdot js\frac{d}{ds} + jbd + d^2s^2\right]^{\frac{n}{2}}\cdot G_C^{(a,b,c,d)}(s)$
(7) Time reverse	$O_C^{(a,b,c,d)}(g(-t)) = O_C^{(a,b,c,d)}(g(t)) = G_C^{(a,b,c,d)}(s)$
(8) Parseval's theorem	$\int_{-\infty}^{\infty}\left\lvert G_C^{(a,b,c,d)}(s)\right\rvert^2 ds = \int_{-\infty}^{\infty}\lvert Even(g(t))\rvert^2 dt.$
(9) Generalized Parseval's theorem	$\int_{-\infty}^{\infty} G_C^{(a,b,c,d)}(s)\cdot\overline{H_C^{(a,b,c,d)}(s)}\cdot ds = \int_{-\infty}^{\infty} Even(g(t))\cdot Even(h(t))\,dt$
(10) Transform result of CCT for $g(t) = \exp[-j(pt^2+qt)]$	$G_C^{(a,b,c,d)}(s) = \sqrt{\frac{1}{a-2pb}}\cdot e^{\frac{j}{2}\cdot\frac{c-2pd}{a-2pb}\cdot s^2}\cdot e^{-j\frac{q^2b}{2(a-2pb)}}\cdot\cos\left(\frac{q\cdot s}{ab-2pb^2}\right).$ In fact, the transform result for $g(t) = \exp(-jpt^2)\cos(qt)$, and the transform result for $g(t) = 2\exp(-jpt^2)\cos(qt)\cdot u(t)$ are also the same as above.
(11) Transform result of CCT for $g(t) = 1$	$G_C^{(a,b,c,d)}(s) = \sqrt{\frac{1}{a}}\cdot e^{\frac{j}{2}\cdot\frac{c}{a}\cdot s^2}$ . .

TABLE V
PROPERTIES OF THE SFRCT

(1) Modulation by $\cos(\eta\cdot t)$	$O_{SC}^{\alpha}(\cos(\eta t)\cdot g(t)) = \left[G_{SC}^{\alpha}(s-\eta) + G_{SC}^{\alpha}(s+\eta)\right]/2,$ $O_{SC^*}^{\alpha}(\cos(\eta t)\cdot g(t)) = \left[G_{SC^*}^{\alpha}(s-\eta) + G_{SC^*}^{\alpha}(s+\eta)\right]/2.$
(2) Modulation by $\sin(\eta\cdot t)$	$O_{SC}^{\alpha}(\sin(\eta t)\cdot g(t)) = -\left[G_{SS^*}^{\alpha}(s-\eta) + G_{SS^*}^{\alpha}(s+\eta)\right]/2,$ $O_{SC^*}^{\alpha}(\sin(\eta t)\cdot g(t)) = -\left[G_{SS}^{\alpha}(s-\eta) + G_{SS}^{\alpha}(s+\eta)\right]/2.$
(3) Differentiation	$O_{SC}^{\alpha}(g'(t)) = a\cdot G_{SS}'^{\alpha}(s) - c\cdot s\cdot G_{SS^*}^{\alpha}(s)$
(4) Multiplication by t	$O_{SC}^{\alpha}(t\cdot g(t)) = d\cdot s\cdot G_{SS}^{\alpha}(s) + b\cdot G_{SS^*}'^{\alpha}(s)$
(5) Parseval's theorem	$\int_0^{\infty}\left(G_{SC}^{\alpha}(s)\right)^2 ds = \int_0^{\infty}\left(\cos\left(\cot\phi\cdot t^2/2\right)\cdot g(t)\right)^2 dt,\qquad \phi = \alpha\pi/2.$
(6) Generalized Parseval's theorem	$\int_0^{\infty} G_{SC}^{\alpha}(s)H_{SC}^{\alpha}(s)\cdot ds = \int_0^{\infty}\cos^2\left(\cot\phi\cdot t^2/2\right)\cdot g(t)\cdot h(t)\cdot dt,\qquad \phi = \alpha\pi/2.$
(7) Transform result of SFRCT for $g(t) = 1$	$G_{SC}^{\alpha}(s) = \sqrt{\frac{1}{2\cot\phi}}\cdot\left(\cos\left(\frac{s^2}{2\cot\phi}\right) + \sin\left(\frac{s^2}{2\cot\phi}\right)\right),\qquad \phi = \alpha\pi/2.$

Table IV, the amount of displacement depends on their parameters. The fact that the modulation property is independent of α is also an advantage of the SFRCT.

VI. DIGITAL IMPLEMENTATION

A. Review of the Digital Implementation of the FRFT, LCT, and SFRFT

In this section, we will discuss the digital implementation methods of all the fractional, canonical, and simplified fractional transforms introduced in Section II–IV. For the convenience of comparison, we first review the digital implementation of the FRFT, LCT, and SFRFT.

In [12] and [29], the authors stated that we require two P-point multiplications (P is the total number of sampling points) and one P-points convolution to implement the FRFT or LCT. Then, in our paper [28], we have stated that we can constrain the sampling intervals in the time and frequency domains to satisfy

$$\Delta_t\cdot\Delta_S = \frac{2\pi\sin\alpha}{P}\text{for FRFT},\quad \Delta_t\cdot\Delta_S = \frac{2\pi b}{P}\text{ for LCT.}\tag{120}$$

In this case, we just require two chirp multiplications and one DFT to implement the FRFT or LCT. Since the DFT requires $(P/3)\cdot\log_2 P$ complex multiplications (using the split-radix algorithm), we totally require $2P+(P/3)\cdot\log_2 P$ complex multiplications to implement the FRFT or LCT if (120) is satisfied. Because each complex number multiplication requires three real number multiplications, the amount of real number multiplications required for the FRFT or LCT is

$$6P + P\cdot\log_2 P.\tag{121}$$

The SFRFT is the special case of the LCT where $\{a,b,c,d\} = (\cot\phi, 1-1, 0)$, and therefore, its complexity is similar to the complexity of the LCT, except that one chirp multiplication can be saved (since $d=0$). Therefore, the amount of real number multiplications required for the SFRFT is

$$3P + P\cdot\log_2 P$$
$$\text{(if the sampling intervals satisfy}\Delta_t\cdot\Delta_S = \frac{2\pi}{P}).\tag{122}$$

B. Using the Fast Algorithm of FRFT or LCT

We now discuss the digital implementation of all the transforms introduced in Sections II–IV. We can apply the fast algorithms of the DCT [20]–[23], [25], [27], DST [22], [23], and DHT [10], [24], [26] to implement these transforms.

We first discuss how to use DCT to implement the FRCT. We can first sample the t-axis and s-axis as

$$\begin{aligned} t =& (n+n_0)\Delta_t,\quad s = (m+m_0)\Delta_s\\ n,m =& ,1,2,\ldots,N-1.\end{aligned}\tag{123}$$

Then, together with the formula of the one-sided FRCT ((46)), we can implement the FRCT as

$$\begin{aligned} &G_C^{\alpha}((m+m_0)\Delta_s) \\ &= \sqrt{\frac{2-j2\cot\phi}{\pi}} e^{j((m+m_0)^2\Delta_s^2/2)\cot\phi}\cdot\Delta_t \\ &\quad\cdot\left\{\sum_{n=1}^{N-1}\cos(\csc\phi\cdot(m+m_0)(n+n_0)\Delta_s\Delta_t)\right. \\ &\quad\cdot e^{j((n+n_0)^2\Delta_t^2/2)\cot\phi} g((n+n_0)\Delta_t) \\ &\quad+\cos(\csc\phi\cdot(m+m_0)n_0\Delta_s\Delta_t) \\ &\quad\left.\cdot\left(n_0+\frac{1}{2}\right)\cdot e^{j(n_0^2\Delta_t^2/2)\cot\phi} g(n_0\Delta_t)\right\}. \qquad (124) \end{aligned}$$

Then, we choose the values of Δ_s, Δ_t, n_0, and m_0 properly to make the cosine term in (124) becomes the kernel of the DCT, and we can use the fast algorithm of the DCT [20]–[23], [25], [27] together with the chirp multiplications to implement the FRCT. The amount of real number multiplications required for the DCT, when the input is a real function, is [20], [27] as follows.

- For the first-type DCT

$$1-N+\left(\frac{N}{2}\right)\cdot\log_2 N. \qquad (125)$$

- For the second- and the third-type DCT

$$\left(\frac{N}{2}\right)\cdot\log_2 N. \qquad (126)$$

- For the fourth-type DCT

$$N+\left(\frac{N}{2}\right)\cdot\log_2 N. \qquad (127)$$

They are much less than the amount of real number multiplications required for the DFT (about $N\cdot\log_2 N$). Therefore, if we implement the FRCT by the DCT, then the complexity of the FRCT will be much less than the complexity of the FRFT (because the FRFT is implemented by the DFT).

For example, if we choose Δ_s, Δ_t, n_0, and m_0 as

$$\Delta_s\cdot\Delta_t=\frac{\pi\cdot\sin\phi}{(N-1)},\quad n_0=0,\quad m_0=0 \qquad (128)$$

then (124) becomes

$$\begin{aligned} G_C^{\alpha}(m\Delta_s) =&\sqrt{\frac{2-j2\cot\phi}{\pi}}\, e^{j(m^2\Delta_s^2/2)\cot\phi} \\ &\cdot\Delta_t\cdot\sum_{n=1}^{N-1}\cos\left(\frac{\pi\, m\, n}{N-1}\right)\cdot\tilde{g}(n\Delta_t) \qquad (129) \end{aligned}$$

$$\begin{aligned} \text{where } \tilde{g}(0) =&\frac{g(0)}{2} \\ \tilde{g}(n) =&e^{j(n^2\Delta_t^2/2)\cot\phi} g(n\Delta_t) \text{ for } n\neq 0. \qquad (130) \end{aligned}$$

We notice that $\cos(\pi mn/(N-1))$ is just the kernel of the first-type DCT. Therefore, in this case, we can use the fast algorithm of the first-type DCT to implement the FRCT. We can follow the process as follows.

1) Do the chirp multiplication as (130) to obtain $\tilde{g}(m)$.
2) Then, do the first-type DCT for $\tilde{g}(m)$.
3) Then, multiply the above transform result by the outside chirp term, as (129).

The first and the third steps both require N-point complex multiplication (i.e., $3N$ real multiplications). Since, for the second step, the input is a complex function, we do the first-type DCT with real input function twice, and the real number multiplications required for the second step is twice (125) Thus, the total amount of real number multiplications required is

$$4N+2+N\cdot\log_2 N. \qquad (131)$$

Since, from (123), N is just the number of sampling points for $t\geq 0$, if P is the total number of sampling points (including the sampling points for $t\geq 0$ and $t<0$), then $N=(P+1)/2$. Therefore, (131) can be rewritten as follows.

- *Amount of the real multiplications required for the FRCT implemented by the first-type DCT:*

$$\begin{aligned} 2P+4+\left(\frac{(P+1)}{2}\right)\cdot\log_2\left(\frac{(P+1)}{2}\right) \\ \approx 2P+\left(\frac{P}{2}\right)\cdot\log_2\left(\frac{P}{2}\right). \qquad (132) \end{aligned}$$

We can compare this with the complexity of the FRFT [see (121)]. The complexity of the FRCT is just about half of the complexity of the FRFT. When we deal with the even functions, it is much more efficient to use the FRCT than to use the original FRFT. The FRCT is really an efficient tool to process the even functions.

Except for the first-type DCT, we can also use the second-, the third-, and the fourth-type DCTs to implement the FRCT. In these cases, we choose the values of Δ_s, Δ_t, n_0, and m_0 in (124) as

$$\begin{aligned} \Delta_s\cdot\Delta_t =&\pi\cdot\frac{\sin\phi}{N} \\ \{n_0,m_0\} =&\left\{\frac{1}{2},0\right\} \text{ for the second-type DCT} \\ \{n_0,m_0\} =&\left\{0,\frac{1}{2}\right\} \text{ for the third-type DCT} \\ \{n_0,m_0\} =&\left\{\frac{1}{2},\frac{1}{2}\right\} \text{ for the fourth-type DCT.} \end{aligned}$$

In these cases, the amount of real multiplications required for the FRCT are also about half of the amount of real number multiplications required for the FRFT.

Similarly, we can also implement the CCT by the DCT, and the number of multiplications required is also the same as or near (132). It is about half of the amount of real number multiplications required for the LCT.

C. Using DST and DHT to Implement the FRST, FRHT, and CST

Similarly, we can use the fast algorithm of the DST [22], [23] to implement the FRST and CST and use the fast algorithm of the DHT [10], [24], [26] to implement the FRHT.

We can implement the FRST and CST by two chirp multiplications and one DST. Since the amount of real number multiplications required for the DST is about

$$\left(\frac{N}{2}\right)\log_2 N \text{ (when the input of the DST is a real function)} \tag{133}$$

the total amounts of real multiplications required for the FRST and CST are about

$$6N + N\cdot\log_2 N \approx 3P + \left(\frac{P}{2}\right)\cdot\log_2\left(\frac{P}{2}\right) \tag{134}$$

where P is the total number of sampling points, and $N \approx P/2$ is the number of sampling points for $t \geq 0$. (It is also the length of the input function of DST.)

We can also use two chirp multiplications and one DHT to implement the FRHT. The amount of real number multiplications required for the DHT is about

$$\left(\frac{N}{2}\right)\cdot\log_2\left(\frac{N}{2}\right). \tag{135}$$

For the case of the DHT, since the input function is neither even nor odd, the number of points of the input function of the DHT must equal the total number of sampling points, i.e., $N = P$. The total amount of real number multiplications required for the FRHT is

$$2\cdot 3N + 2\cdot\left(\frac{N}{2}\right)\cdot\log_2\left(\frac{N}{2}\right) = 6P + P\cdot\log_2\left(\frac{P}{2}\right). \tag{136}$$

Comparing (134) with (121), we find the complexities of the FRST and CST to be only half of the complexities of the FRFT and LCT. It is quite efficient to use the FRST/CST instead of the FRFT/LCT to process odd functions. Besides, the FRHT is also an efficient signal processing tool.

D. Implementation of the SFRCT, SFRST, and SFRHT

For the SFRCT of type 1, we can sample the t and s axes as (123). Then, (95) becomes

$$\begin{aligned} G^{\alpha}_{SC}&((m+m_0)\Delta_s) \\ &= \sqrt{\frac{2}{\pi}}\Delta_t \sum_{n=1}^{N-1} \cos\left((m+m_0)(n+n_0)\Delta_s\Delta_t\right) \\ &\quad\cdot \cos\left(\frac{1}{2}\cot\phi (n+n_0)^2\Delta_t^2\right) g((n+n_0)\Delta_t) \\ &\quad+ \left(n_0+\frac{1}{2}\right)\cos((m+m_0)n_0\Delta_s\Delta_t) \\ &\quad\cdot \cos\left(\frac{\cot\phi\, n_0^2\Delta_t^2}{2}\right) g(n_0\Delta_t). \end{aligned} \tag{137}$$

When we choose Δ_s, Δ_t, n_0, and m_0 properly, then the cosine term of the above equation becomes the kernel of the DCT of type 1, 2, 3, or 4, and we can use the fast algorithm of the DCT of type 1, 2, 3, or 4 to implement it. For example, if we choose Δ_s, Δ_t, n_0, and m_0 as

$$\Delta_s\cdot\Delta_t = \frac{\pi}{(N-1)}, \quad n_0 = 0, \quad m_0 = 0 \tag{138}$$

then (138) becomes

$$\begin{aligned} G^{\alpha}_{SC}((m+m_0)\Delta_s) &= \sqrt{\frac{2}{\pi}}\Delta_t \sum_{n=0}^{N-1}\cos\left(\frac{\pi m n}{N-1}\right)\cdot\tilde{g}(n\Delta_t) \\ \text{where } \tilde{g}(0) &= \frac{g(n\Delta_t)}{2} && (139) \\ \tilde{g}(n\Delta_t) &= \cos\left(\frac{1}{2}\cot\phi n^2\Delta_t^2\right) g(n\Delta_t) \\ \text{for } n &\neq 0. && (140) \end{aligned}$$

Then, we can implement the SFRCT of type 1 by the first-type DCT together with the multiplication of $\cos(\cot\phi\cdot n^2\Delta_t^2/2)$. Since the input function $g(n\Delta_t)$ of the SFRCT is constrained to be a real function, (140) is the multiplication operation between two real functions. It requires N real number multiplications because if $g(n\Delta_t)$ is real, then the input $\tilde{g}(n\Delta_t)$ of the first-type DCT is also a real function, and the amount of real multiplications required for the first-type DCT with real input is $1 - N + (N/2)\cdot\log_2 N$. Therefore, he total amount of real number multiplications required for the SFRCT implemented by the first-type DCT is as follows.

- *Amount of real multiplications required for the SFRCT implemented by the first-type DCT*

$$1 + \left(\frac{N}{2}\right)\cdot\log_2 N \approx 1 + \left(\frac{P}{4}\right)\cdot\log_2\left(\frac{P}{2}\right). \tag{141}$$

We also use the fact that N (the length of the input for DCT, i.e., the number of sampling points for $t \geq 0$) is about half of P (the total number of sampling points) for the case of the SFRCT. Equation (141) is not only much less than the complexities of the FRFT and LCT [see (121)] (about one fourth) but is also less than the complexities of the FRCT and CCT [see (132)] (about one half). When the input function is both real and even, it is much more efficient to use the SFRCT to process it than to use the FRFT, LCT, FRCT, and CCT.

In (138), the product of sampling intervals Δ_t, Δ_s is fixed to $\pi/(N-1)$. It is independent of the parameter ϕ of SFRCT. We do not have to adjust Δ_t, Δ_s when ϕ is changed. This advantage does not exist for the FRCT and CCT.

We can use a similar algorithm to implement the SFRCT of type 2.

By the same reasons, we can use the fast algorithms of the DST and DHT to implement the SFRST and SFRHT of types 1 and 2. We require one multiplication of $\cos(\cot\phi\cdot n^2\Delta_t^2/2)$ [or $\sin(\cot\phi\cdot n^2\Delta_t^2/2)$] and one DST (or DHT) to implement these transforms. Since $g(n\Delta_t)\times\cos(\cot\phi\cdot n^2\Delta_t^2/2)$ or $g(n\Delta_t)\times\sin(\cot\phi\cdot n^2\Delta_t^2/2)$ is the multiplication operation between two real functions, and the input of DST is a real function, the total amount of real multiplications required for the SFRST (no matter for type 1 or type 2) implemented by the DST is

$$N+\left(\frac{N}{2}\right)\cdot\log_2 N\approx\frac{P}{2}+\left(\frac{P}{4}\right)\cdot\log_2\left(\frac{P}{2}\right). \tag{142}$$

Similarly, the total amount of real multiplications required for the SFRHT (no matter for type 1 or type 2) implemented by the DHT is

$$N+\left(\frac{N}{2}\right)\cdot\log_2\left(\frac{P}{2}\right)=P+\left(\frac{P}{2}\right)\cdot\log_2\left(\frac{P}{2}\right). \tag{143}$$

We remember that for the case of the SFRST, $N\approx P/2$, and for the case of the SFRHT, $N=P$.

Comparing (142) and (143) with (121), (134), and (136), we find that the complexity of the SFRST is one fourth of the complexities of the FRFT or LCT and is one half of the complexity of FRST or CST. The complexity of SFRHT is one half of the complexities of FRFT or LCT, and FRHT. Therefore, it is much more efficient to use the SFRST to deal with the real, odd functions than using other transforms. It is also more efficient to use the SFRHT to deal with pure real/imaginary functions than to use other transforms.

VII. APPLICATIONS

A. *Substituting the FRFT and LCT*

The transforms introduced in this paper have an important advantage. That is, they can replace the FRFT and LCT when the input function is even, odd, pure real, or pure imaginary. From Sections II–IV, we have known that the FRCT and FRST have very close relations with the the FRFT. The CCT and CST have very close relations with the LCT. The SFRCT, SFRST, and SFRHT have very close relations with the SFRST and, hence, the FRFT. Therefore, the performances of these transforms are very similar to the performances of the FRFT and LCT.

In Section VI, we have stated the following.

1) When we use the FRCT and CCT to process *even functions*, then the amount of the required real number multiplications is only one half of those of the FRFT and LCT.
2) When we use the FRST and CST to process *odd functions*, then the amount of the required real number multiplications is only one half of those of the FRFT and LCT.
3) When we use the SFRHT to process *pure real/imaginary functions*, then the amount of the required real number multiplications is only one half of those of the FRFT and LCT.
4) When we use the SFRCT to process *pure real/imaginary, even functions*, then the amount of the required real number multiplications is only one fourth of those of the FRFT and LCT.
5) When we use the SFRST to process *pure real/imaginary, odd functions*, then the amount of the required real number multiplications is only one fourth of those of the FRFT and LCT.

Since the performances of the FRCT, FRST, CCT, CST, SFRCT, SFRST, and SFRHT are similar to the performances of the FRFT, LCT, and SFRFT and their complexities are much less, we can use these transforms to replace the the FRFT and LCT if the input is even, odd, pure real, or pure imaginary.

Thus, *all the applications of the FRFT and LCT are also the applications of the transforms introduced in this paper*. The applications of the FRFT and LCT are filter design, optical system analysis, radar system analysis, solving differential equations, phase retrieval, multiplexing, space-variant pattern recognition, localized edge detection, etc. They will also be the applications of the transforms introduced in this paper. In Sections VII-A1–3, we give three examples:

1) optical system analysis,
2) filter design,
3) pattern recognition.

They are all the applications of the FRFT and LCT and, hence, the applications of the transforms introduced in this paper.

1) Optical System Analysis: For the optical system as Fig. 3, the relation between the input and output can be expressed as the LCT with parameters $\{1-d_2/f_0,(d_1+d_2-d_1d_2/f_0)/k,-k/f_0,1-d_1/f_0\}$, where $k=2\pi/\lambda$ [19]

$$\begin{aligned} f_o(p,q)=&e^{j\varphi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}K_F^{(a,b,c,d)}(p,x)K_F^{(a,b,c,d)}\\ &\cdot(q,y)f_i(x,y)dxdy \end{aligned} \tag{144}$$

$$\text{where}\begin{bmatrix}a & b\\ c & d\end{bmatrix}=\begin{bmatrix}1-\frac{d_2}{f_0} & \frac{1}{k}\left(d_1+d_2-\frac{d_1d_2}{f_0}\right)\\ -\frac{k}{f_0} & 1-\frac{d_1}{f_0}\end{bmatrix}$$

φ is some constant phase

$$\begin{aligned} K_F^{(a,b,c,d)}(p,x)=&\sqrt{\frac{1}{j2\pi b}}\cdot e^{j(d/2b)p^2}\,e^{j(1/b)px}\,e^{j(a/2b)x^2}\\ &\text{(the kernel of LCT).} \end{aligned} \tag{145}$$

In the case that the input is even for both the x and y axes

$$f_i(x,y)=f_i(-x,y),\quad f_i(x,y)=f_i(x,-y) \tag{146}$$

then we can use the CCT with the same parameters to substitute the LCT for both x and y-axes, and change (144) to be

$$\begin{aligned} f_o(p,q)=e^{j\varphi}\int_0^{\infty}\int_0^{\infty}&K_C^{(a,b,c,d)}(p,x)\\ &\cdot K_C^{(a,b,c,d)}(q,y)f_i(x,y)\cdot dxdy \end{aligned} \tag{147}$$

where the values of $\{a,b,c,d\}$ are the same as (145), φ is some constant phase, and $K_C^{(a,b,c,d)}(p,x)$ is the kernel of the one-sided CCT

$$K_C^{(a,b,c,d)}(p,x)=\sqrt{\frac{2}{j\pi b}}\cdot e^{j(d/2b)p^2}\cdot\cos\left(\frac{px}{b}\right)\cdot e^{j(a/2b)x^2}. \tag{148}$$

The value of $f_o(p,q)$ calculated from (147) is the same as the value of $f_o(p,q)$ calculated from (144), but since the complexity of the CCT is about one half of the complexity of the LCT, the complexity of (147) is just about one fourth of the complexity of (144).

Similarly, if the input is even for the x axis and odd for the y axis

$$f_i(x,y)=f_i(-x,y),\quad f_i(x,y)=-f_i(x,-y) \tag{149}$$

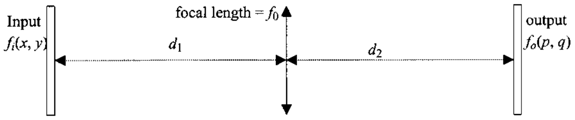


Fig. 3. Optical system with one spherical lens and two free spaces.

then in (144), we can use the CCT instead of the LCT for the x axis and use the CST instead of the LCT for the y axis. When the input is odd for the x axis and even for the y axis, we can use the CST for the x axis and use the CCT for the y axis. When the input is odd for both the x axis and y axis, we can also use the CST for both the x axis and the y axis. For all of these cases, the complexities of computation are just about one fourth of the complexity when we use the LCT directly.

For the application of optical system analysis, we can use CCT and CST instead of the LCT to improve the efficiency when the input is even or odd.

2) Filter Design: The filter designed by the FRFT [18] can be expressed as

$$g_0(t) = O_F^{-\alpha}\left(O_F^{\alpha}(g_i(t)) \cdot H(s)\right) \tag{150}$$

where $g_i(t)$ is the received signal. We try to design the proper values of α and $H(s)$ such that the output $g_o(t)$ is near the original signal. This type of filter is more flexible and can do more things than the filter designed by the FT.

Since, in practice, the received signal $g_i(t)$ is usually a real function, in (150), it may be convenient to use the SFRHT instead of the FRFT

$$g_o(t) = O_{SH}^{-\alpha}\left(O_{SH}^{\alpha}(g_i(t)) \cdot H(s)\right). \tag{151}$$

From Section VI-D, the complexity of the SFRHT is just about one half of the complexity of the FRFT; for the real signal, using the SFRHT for filter design is more efficient than using the FRFT.

3) Application of the SFRCT for Space-Variant Pattern Recognition: In [6], the authors found, with the aid of fractional correlation, that the FRFT can be used for space variant pattern recognition. Fractional correlation [30] is defined as

$$z(t) = O_{\mathrm{corr}}^{a}(x(t), y(t)) = FT\left(O_F^{a}(x(t)) \cdot \overline{O_F^{a}(y(y))}\right). \tag{152}$$

For the application of space-variant pattern recognition, we treat $x(t)$ as the reference pattern and treat $y(t)$ as the input object. Then, we calculate $z(t)$ and find the peak value of $|z(t)|$. If

$$\mathrm{Max}(|z(t)|) > \text{threshold} \tag{153}$$

then we conclude that the input object matches the reference pattern and that the location difference between the input object and the reference pattern is within some certain region. That is

$$y(t) = x(t - t_0), \text{ and } |t_0| < R. \tag{154}$$

If the input object does not match the reference pattern or the location difference is too large, then (153) will not be satisfied, and the input object will not be recognized as matching the reference pattern. Therefore, it is called space-variant pattern recognition. The value of R in (154) is affected by a. If, in (152), we set $a = 1$, i.e., we do the original FTs for $x(t)$ and $y(t)$, then the value of R will be infinite. If we choose the value of a between 0 and 1, then the value of R will be finite. When the value of $|\tan(a\pi/2)|$ is smaller, then the value of R is also smaller.

Because, for the application of space-variant pattern recognition, the reference pattern and the input object are always real functions, we can use the SFRCT, SFRST, and SFRHT instead of the the FRFTs in (152) to improve efficiency.

For example, we can use the SFRCTs instead of the FRFTs in (152) when the reference pattern is a real, even function. Due to the displacement, the input object is always not an even function, but from our experiments, it does not cause any problem when we replace all the FRFTs with SFRCTs. We just require the reference pattern to be a real, even function. If we use SFRCTs instead of FRFTs in (152), then (152) is changed as

$$z(t) = O_{\mathrm{CCR}}^{a}(x(t), y(t)) = CT\left(O_{SC}^{a}(x(t)) \cdot O_{SC}^{a}(y(t))\right) \tag{155}$$

where $CT(\ )$ means the original cosine transform. We call (155) the FRCCR of $x(t)$ and $y(t)$. We notice that (155) can be rewritten as

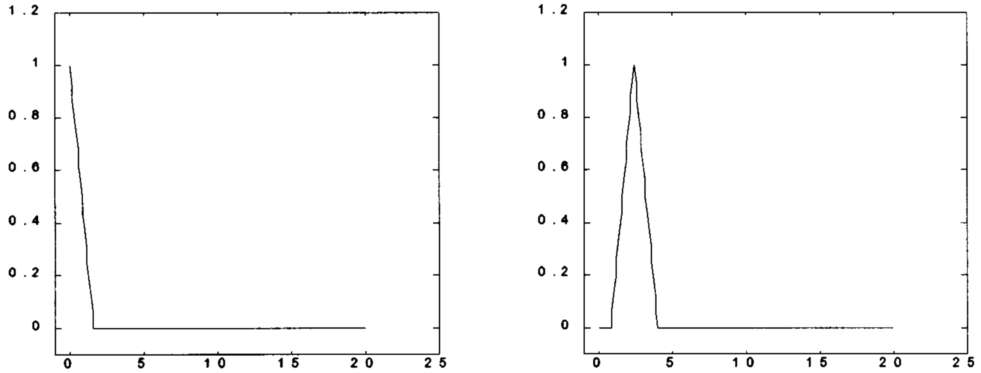
$$\begin{aligned} z(t) = &\sqrt{\frac{8}{\pi^3}} \int_0^\infty \int_0^\infty \left[\int_0^\infty \cos(ts)\cos(sp)\cos(sq) \cdot ds\right] \\ &\cdot \cos\left(\cot\phi \cdot \frac{p^2}{2}\right) \cos\left(\cot\phi \cdot \frac{q^2}{2}\right) \\ &\cdot x(p)y(q) \cdot dpdq, \text{ where } \phi = \frac{\alpha\pi}{2}. \end{aligned}$$

If $y(t)$ is the displacement of $x(t)$, as (157), then after some computation, we can prove that the maximum of $|z(t)|$ usually occurs at the location of $t = t_0$, and its value is

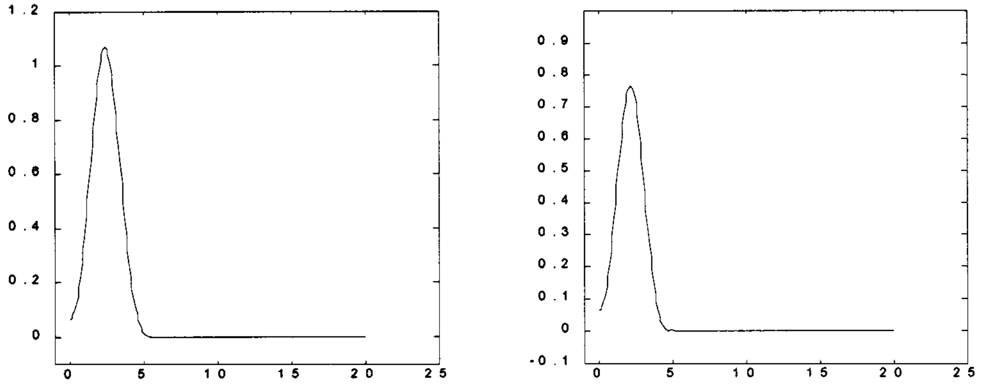
$$\begin{aligned} \mathrm{Max}(|z(t)|) \approx &\sqrt{\frac{1}{2\pi}} \left| \int_0^\infty \cos\left(\cot\phi \cdot \frac{p^2}{2}\right) \right. \\ &\cdot \cos\left(\cot\phi \cdot \frac{(t_0 + p)^2}{2}\right) x^2(p) \cdot dp \\ &+ \int_0^{t_0} \cos\left(\cot\phi \cdot \frac{p^2}{2}\right) \\ &\left. \cdot \cos\left(\cot\phi \cdot \frac{(t_0 - p)^2}{2}\right) x^2(p) \cdot dp \right| \end{aligned} \tag{156}$$

where $\phi = a\pi/2$. Therefore, we conclude that the value of $\mathrm{Max}|z(t)|$ is affected by whether the input object $y(t)$ matches the reference pattern $x(t)$ and whether the displacement t_0 is sufficient small. For a fixed value of ϕ, $\mathrm{Max}|z(t)|$ is large when t_0 is small, and $\mathrm{Max}|z(t)|$ is small when t_0 is large. The value of ϕ can control how fast $\mathrm{Max}|z(t)|$ varies with t_0. If $|\cot\phi|$ is larger, then $\mathrm{Max}|z(t)|$ will decay faster with t_0. Therefore, if we want (153) to be satisfied, then the input object $y(t)$ must match the reference pattern $x(t)$, and their location difference should not be very large:

$$y(t) = x(t - t_0), \text{ and } |t_0| < R. \tag{157}$$



(a) reference pattern **(b) object ($t_0 = 2.5$)**



(c) correlation for $a = 1$ **(d) correlation for $a = 0.92$**

Fig. 4. Space-variant pattern recognition. (a) Reference $x(t)$. (b) Input $y(t) = x(t-3)$. (c) Result (absolute value) of FRCCR with $a = 1$. (d) Result (absolute value) of FRCCR with $a = 0.9$.

The value of R is affected by $\phi(\phi = a\pi/2)$. If $|\tan\phi|$ is smaller, then R is also smaller. Therefore, we can use the SFRCT for space-variant pattern recognition when the reference pattern is real] and even.

We give an example in the following. Here, we choose the reference pattern as a triangular function

$$x(t) = \Lambda\left(\frac{t}{1.6}\right). \tag{158}$$

We plot $x(t)$ for $t \geq 0$ in Figs. 4(a) and 5(a). Then, we choose the input object as

$$\begin{aligned}\text{for Fig4: } y(t) =& \Lambda\left(\frac{t-2.5}{1.6}\right)\\ \text{for Fig5: } y(t) =& \Lambda\left(\frac{t-14}{1.6}\right).\end{aligned} \tag{159}$$

They are plotted in Figs. 4(b) and 5(b). They all match the reference pattern but have different amount of displacement. Then, we do the FRCCR with $a = 1$ [in this case, the SFRCTs in (155) all become the original cosine transforms] for $x(t)$ and $y(t)$ and plot the results in Figs. 4(c) and 5(c). Finally, we do the FRCCR with $a = 0.92$ for $x(t)$ and $y(t)$ and plot the results in Figs. 4(d) and 5(d).

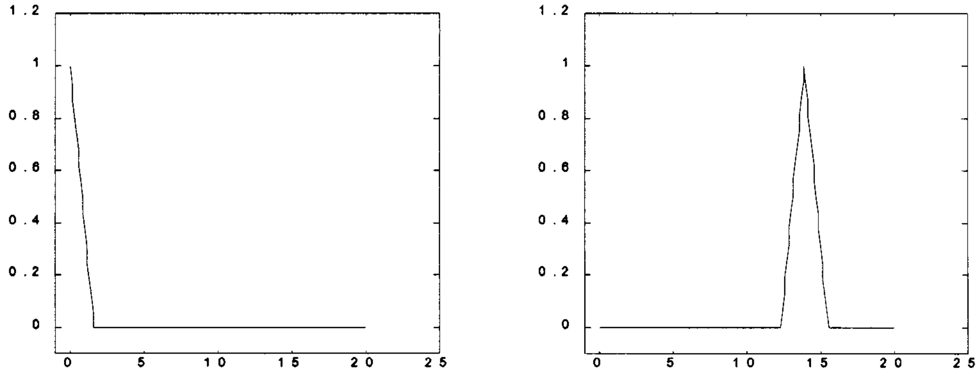
From the above experiments, we find, for the case that $a = 1$, that the value of $\text{Max}|z(t)|$ ($z(t)$ is the output of FRCCR) is almost independent of t_0. This is because in this case, the value of $\cot\phi(\phi = \alpha\pi/2)$ in (156) equals to 0; therefore, t_0 has no effect on the value of $\text{Max}|z(t)|$. For the case where $a = 0.92$, from Figs. 4(d) and 5(d), we find that the value of $\text{Max}|z(t)|$ decreases with t_0. In this case, we can choose the threshold properly such that only when t_0 is within some region, the value of $\text{Max}|z(t)|$ will be above the threshold. Therefore, we can use the SFRCT for space-variant pattern recognition when $0 < a < 1$ and when the reference pattern is real and even.

B. One-Sided Signal Processing and Other Potential Applications

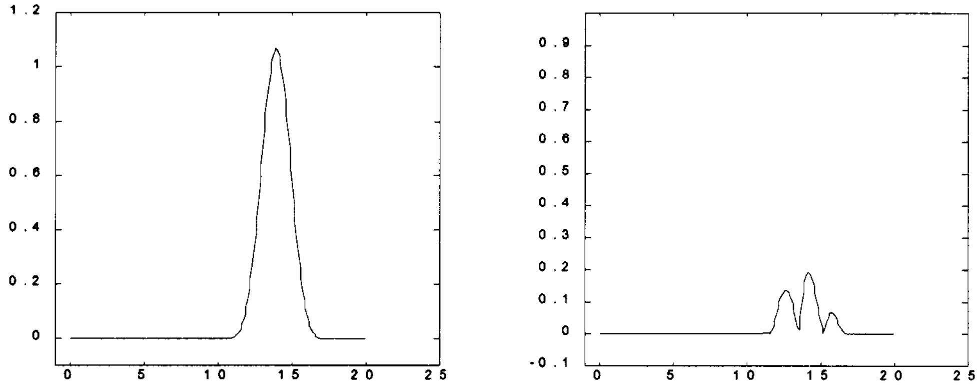
In Section VII-A, we have stated that we can use the transforms introduced in this paper to substitute the FRFT and LCT in many conditions. In fact, except for substituting the FRFT and LCT, there are also other potential applications for these transforms.

We can use the FRCT, FTST, CCT, CST, FRSCT, and FRSST for one-sided signal processing. For one-sided signal processing, the time interval we considered is

$$t \in [0, \infty). \tag{160}$$



(a) reference pattern (b) object ($t_0 = 14$)



(c) correlation for $a = 1$ (d) correlation for $a = 0.92$

Fig. 5. Space-variant pattern recognition. (a) Reference $x(t)$. (b) Input $y(t) = x(t-14)$. (c) Result (absolute value) of FRCCR with $a = 1$. (d) Result (absolute value) of FRCCR with $a = 0.9$.

In this case, the constraint of the one-sided FRCT to be reversible is

$$g'(0) = 0 \text{ where } g(t) \text{ is the input of the one-sided FRCT.} \tag{161}$$

This is also the constraint that we can use the FRCT for one-sided signal processing.

Proof of (161): If $g(t)$ is reversible after doing the one-sided FRCT, then it can be expressed as

$$\begin{aligned} g(t) = &\sqrt{\frac{2+j2\cot\phi}{\pi}} e^{-j(s^2/2)\cot\phi} \\ &\cdot \int_0^\infty \cos(\csc\phi \cdot st) e^{-j(t^2/2)\cot\phi} G_C^\alpha(t) \cdot dt. \end{aligned} \tag{162}$$

Then, because

$$\begin{aligned} g'(t) =& \sqrt{\frac{2+j2\cot\phi}{\pi}}(-jt\cot\phi) \cdot e^{-j(t^2/2)\cot\phi} \\ &\cdot \int_0^\infty \cos(\csc\phi \cdot st) e^{-j(s^2/2)\cot\phi} G_C^\alpha(s) \cdot ds \\ &- \sqrt{\frac{2+j2\cot\phi}{\pi}} \cdot, e^{-j(t^2/2)\cot\phi} \\ &\cdot \int_0^\infty \csc\phi \cdot s \cdot \sin(\csc\phi \cdot st) e^{-j(s^2/2)\cot\phi} G_C^\alpha(s) \cdot ds \\ g'(0) =& 0 \end{aligned}$$

if $g(t)$ is reversible after doing the one-sided FRCT, then $g'(0) = 0$ must be satisfied.

Similarly, $g'(0) = 0$ is also the constraint for the one-sided CCT [see (86) and (87)] and the SFRCT [(95) and (103)] to be reversible

$$\begin{aligned} O_C^{(d,-b,-c,a)}\left(O_C^{(a,b,c,d)}(g(t))\right) &= g(t) \\ O_{ISC}^a\left(O_{SC}^a(g(t))\right) &= g(t) \\ O_{ISC''}^a\left(O_{SC''}^a(g(t))\right) &= g(t) \text{ if } g'(0) = 0. \end{aligned} \tag{163}$$

If the input function has zero differential value at $t = 0$, then we can use the one-sided FRCT, one-sided CCT, and the SFRCT for one-sided signal processing.

Similarly, we can also conclude that for one-sided signal processing, if the input satisfies

$$g(0) = 0 \tag{164}$$

then it is reversible for one-sided FRST, one-sided CST, and the SFRST. We can use these transforms for one-sided signal processing when the initial value of the input is zero.

If the input function satisfies $g'(0) = 0$ or $g(0) = 0$, then we can use the generalized cosine and sine transforms introduced in this paper for one-sided signal processing. We can use them to solve the differential equations with zero initial value or zero initial differential value. We can also use them for one-sided filter

TABLE VI
SUMMARY OF THE FRACTIONAL, CANONICAL, AND SIMPLIFIED COSINE, SINE, AND HARTLEY TRANSFORMS INTRODUCED IN THIS PAPER

Transform	Formula	Amount of real multiplications	Analogous to	Suitable for which kind of functions
FRCT	Eq. (38)	$2P+(P/2)\cdot\log_2(P/2)$	FRFT	Even
FRST	Eq. (39)	$3P+(P/2)\cdot\log_2(P/2)$	FRFT	Odd
FRHT	Eq. (41)	$6P+P\cdot\log_2(P/2)$	FRFT	
CCT	Eqs. (68), (69)	$2P+(P/2)\cdot\log_2(P/2)$	LCT	Even
CST	Eqs. (70), (71)	$3P+(P/2)\cdot\log_2(P/2)$	LCT	Odd
SFRCT	type 1: Eq. (95) type 2: Eq. (103)	$1+(P/4)\cdot\log_2(P/2)$	SFRFT	Pure real / imaginary, even
SFRST	type 1: Eq. (107) type 2: Eq. (108)	$P/2+(P/4)\cdot\log_2(P/2)$	SFRFT	Pure real / imaginary, odd
SFRHT	type 1: Eq. (113) type 2: Eq. (114)	$P+(P/2)\cdot\log_2(P/2)$	SFRFT	Pure real or imaginary

design and other applications. Although the conventional cosine and sine transforms can also be used for one-sided signal processing, using the family of generalized cosine and sine transforms can solve more problems. For example, for the noise (we notice that it has zero differential value at $t = 0$)

$$n(t) = A \cdot \exp(j\tau t^2) \text{ for } t \in [0\ \infty) \tag{165}$$

it is hard to use the conventional one-sided cosine transform to remove it. In contrast, if we use the one-sided FRCT or CCT, then we can remove the above noise easily.

In fact, if $g'(0) \neq 0$ and $g(0) \neq 0$, then we can still use the FRCT, FRST, CCT, CST, SFRCT, and SFRST for one-sided signal processing, but in this case, there are some errors around $t = 0$.

Although the family of generalized Hartley transforms cannot be used for one-sided signal processing, they also have some potential applications. For example, since the performance and the efficiency of FRHT is near those of the FRFT, for the consideration of encryption, we can use the FRHT instead of the FRFT. Besides, for the SFRHT introduced in Section IV-C, except for substituting the FRFT and LCT when processing a real signal, it is also useful to deal with the signal that is the linear combination of

$$\{\mathrm{cas}(st) \cdot \sec(\tau \cdot t^2) \mid s \in (-\infty, \infty)\} \quad \tau \text{ is some fixed constant} \tag{166}$$

or

$$\{\mathrm{cas}(st) \cdot \csc(\tau \cdot t^2) \mid s \in (-\infty, \infty)\} \quad \tau \text{ is some fixed constant.} \tag{167}$$

VIII. CONCLUSION

In this paper, we have derived the fractional, canonical, and simplified fractional cosine, sine, and Hartley transforms. They are the generalization of the original cosine, sine, and Hartley transforms. We also discuss their properties, digital implementation, and applications. We summarize all the transforms derived in this paper in Table VI.

The original cosine, sine, and Hartley transforms can replace the FT when we deal with the even, odd, and pure real/imaginary functions. Similarly, among the transforms derived in paper, the FRCT and CCT can replace the FRFT and LCT when we deal with even functions. The FRST and CST can replace the FRFT and LCT when we deal with odd functions. The SFRHT can replace the FRFT and LCT when we deal with pure real/imaginary functions. The SFRCT is suitable for processing pure real/imaginary, even functions, and the SFRST is suitable for processing pure real/imaginary, odd functions. From the discussion in Section VI, the complexities of these transforms are much less than the complexities of the FRFT and LCT. We can just use one half or one fourth of the real number multiplications required for the FRFT and LCT to implement these transforms.

The FRFT and LCT are useful signal processing tools. Since the transforms derived in this paper can be substituted for the FRFT and LCT under many conditions, we believe they will also become useful signal processing tools in the future.

REFERENCES

[1] M. R. Spiegel, *Mathematical Handbook of Formulas and Tables*. New York: McGraw-Hill, 1990.

[2] N. Wiener, "Hermitian polynomials and fourier analysis," *MIT J. Math. Phys.*, vol. 18, pp. 70–73, 1929.

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

[4] P. M. Morse and H. Feschbach, *Methods of Theoretical Physics*. London, U.K.: McGraw-Hill, 1953, p. 781.

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

[6] A. W. Lohmann, Z. Zalevsky, and D. Mendlovic, "Synthesis of pattern recognition filters for fractional Fourier processing," *Opt. Commun.*, vol. 128, pp. 199–204, Jul. 1996.

[7] D. Dragoman, "Fractional Fourier-related functions," *Opt. Commun.*, vol. 128, pp. 91–98, July 1996.

[8] D. Dragoman and M. Dragoman, "Temporal implementation of Fourier-related transforms," *Opt. Commun.*, vol. 145, pp. 33–37, Jan. 1998.

[9] A. W. Lohmann, D. Mendlovic, Z. Zalevsky, and R. G. Dorsch, "Some important fractional transforms for signal processing," *Opt. Commun.*, vol. 125, pp. 18–20, Apr. 1996.

[10] R. N. Bracewell, *The Hartley Transform*. Oxford, U.K.: Oxford Univ. Press, 1986.

[11] ——, *The Fourier Integral and Its Applications*, 2nd ed. NewYork: McGraw-Hill, 1978.

[12] 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, Sep 1996.

[13] K. B. Wolf, "Canonical transforms," in *Integral Transforms in Science and Engineering*. New York: Plenum, 1979, ch. 9.

[14] S. Abe and J. T. Sheridan, "Optical operations on wave functions as the Abelian subgroups of the special affine Fourier transformation," *Opt. Lett.*, vol. 19, no. 22, pp. 1801–1803, 1994.

[15] J. W. Goodman, *Introduction to Fourier Optics*, 2nd ed. New York: McGraw Hill, 1988.

[16] D. F. V. James and G. S. Agarwal, "The generalized Fresnel transform and its applications to optics," *Opt. Commun.*, vol. 126, pp. 207–212, May 1996.

[17] S. C. Pei and J. J. Ding, "Eigenfunctions of the canonical transform and self-imaging problems in optical system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Istanbul, Turkey, June 2000, pp. 73–76.

[18] D. Mendlovic, H. M. Zalevsky, and A. W. Lohmann, "Fractional correlation," *Appl. Opt.*, vol. 34, no. 2, pp. 303–309, Jan. 1995.

[19] S. C. Pei and J. J. Ding, "Simplified fractional Fourier transform," *J. Opt. Soc. Amer. A*, vol. 17, no. 13, pp. 2355–2367, 2000.

[20] B. G. Lee, "A new algorithm for computing the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 1243–1245, Dec. 1984.

[21] K. R. Rao and P. Yip, *Discrete Cosine Transform, Algorithms, Advantage, Applications*. New York: Academic, 1990.

[22] P. Z. Lee and F. Y. Huang, "Restructured recursive DCT and DST algorithms," *IEEE Trans. Signal Processing*, vol. 42, pp. 1600–1609, July 1994.

[23] Z. Cvetkovic and M. V. Popovic, "New fast recursive algorithms for the computation of discrete cosine and sine transforms," *IEEE Trans. Signal Processing*, vol. 40, pp. 2083–2086, Aug. 1992.

[24] H. V. Sorensen, "On computing the discrete hartley transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1231–1238, Oct. 1985.

[25] S. C. Chan and K. L. Ho, "Fast algorithm for computing the discrete cosine transform," *IEEE Trans. Circuits Syst. II*, vol. 44, pp. 185–190, Mar. 1993.

[26] ——, "Prime factor real-valued Fourier, cosine and hartley transform," *Proc. Signal Processing VI*, pp. 1045–1048, 1992.

[27] C. W. Kok, "Fast algorithm for computing discrete cosine transform," *IEEE Trans. Signal Processing*, vol. 45, pp. 757–760, Mar. 1997.

[28] 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.

[29] X. Deng, B. Bihari, J. Gan, F. Zhou, and R. T. Chen, "Fast algorithm for chirp transforms with zooming-in ability and its applications," *J. Opt. Soc. Amer. A*, vol. 17, no. 4, pp. 762–771, Apr. 2000.

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

Soo-Chang Pei (F'00) was born in Soo-Auo, Taiwan, R.O.C., in 1949. He received the B.S.E.E. degree from National Taiwan University (NTU), Taipei, in 1970 and the M.S.E.E. and Ph.D. degree from the University of California, Santa Barbara, in 1972 and 1975, respectively.

He was an engineering officer in the Chinese Navy Shipyard from 1970 to 1971. From 1971 to 1975, he was a Research Assistant at the University of California, Santa Barbara. He was Professor and Chairman of the Electrical Engineering Department of Tatung Institute of Technology and NTU from 1981 to 1983 and 1995 to 1998, respectively. Presently, he is a Professor with the Electrical Engineering Department at NTU. His research interests include digital signal processing, image processing, optical information processing, and laser holography.

Dr. Pei received the National Sun Yet-Sen Academic Achievement Award in Engineering in 1984, the Distinguished Research Award from the National Science Council from 1990 to 1998, the Outstanding Electrical Engineering Professor Award from the Chinese Institute of Electrical Engineering in 1998, and the Academic Achievement Award in Engineering from the Ministry of Education in 1998. He has been President of the Chinese Image Processing and Pattern Recognition Society in Taiwan from 1996 to 1998. He is a member of Eta Kappa Nu and the Optical Society of America.

Jian-Jiun Ding was born in 1973 in Pingdong, Taiwan, R.O.C. He received the B.S. degree in 1995, the M.S. degree in 1997, and the Ph.D. degree in 2001, all in electrical engineering from National Taiwan University (NTU), Taipei.

He is currently a Postdoctoral Researcher with the Department of Electrical Engineering, NTU. His current research areas include the fractional Fourier transform, linear canonical transforms, orthogonal polynomials, fast algorithms, quaternion algebra, pattern recognition, and filter design.