Digital Computation of Linear Canonical Transforms

Aykut Koç, Haldun M. Ozaktas, *Member, IEEE*, Cagatay Candan, and M. Alper Kutay

Abstract—We deal with the problem of efficient and accurate digital computation of the samples of the linear canonical transform (LCT) of a function, from the samples of the original function. Two approaches are presented and compared. The first is based on decomposition of the LCT into chirp multiplication, Fourier transformation, and scaling operations. The second is based on decomposition of the LCT into a fractional Fourier transform followed by scaling and chirp multiplication. Both algorithms take $\sim N \log N$ time, where N is the time-bandwidth product of the signals. The only essential deviation from exactness arises from the approximation of a continuous Fourier transform with the discrete Fourier transform. Thus, the algorithms compute LCTs with a performance similar to that of the fast Fourier transform algorithm in computing the Fourier transform, both in terms of speed and accuracy.

Index Terms—Diffraction integrals, fractional Fourier transform (FRT), linear canonical transform (LCT), time-frequency analysis, Wigner distributions.

I. INTRODUCTION

THE class of linear canonical transforms (LCTs) [1], [2] is a three-parameter class of linear integral transforms which includes among its many special cases, the one-parameter subclasses of fractional Fourier transforms (FRTs), scaling operations, and chirp multiplication (CM) and chirp convolution (CC) operations, the latter also known as Fresnel transforms. These integral transforms are of great importance in electromagnetic, acoustic, and other wave propagation problems since they represent the solution of the wave equation under a variety of circumstances. At optical frequencies, LCTs can model a broad class of optical systems including thin lenses, sections of free space in the Fresnel approximation, sections of quadratic graded-index media, and arbitrary concatenations of any number of these, sometimes referred to as first-order optical systems [2]–[6]. Therefore, the accurate and efficient digital computation of these transforms are of great interest for many applications.

LCTs have also been referred to by different names such as quadratic-phase integrals or quadratic-phase systems [3], generalized Huygens integrals [7], generalized Fresnel transforms [8], [9], special affine Fourier transforms [5], [6], extended frac-

tional Fourier transforms [10], and Moshinsky–Quesne transforms [1], among others.

In this paper, we discuss two approaches for the digital computation of LCTs. The first algorithm decomposes an LCT with arbitrary transform parameters into some combinations of three simpler operations: scaling, Fourier transformation, and chirp multiplication. The second method decomposes the LCT into fractional Fourier transformation, chirp multiplication, and scaling. Both are fast algorithms that take $\sim N \log N$ time, where N is the time–bandwidth product of the input signal. Despite the highly oscillatory nature of the integral kernel, special care is taken to carefully manage the sampling rate so as to ensure that the number of samples N is not chosen much larger than the time–bandwidth product of the input signal, so that the algorithms are as efficient as possible. A naive application of the Nyquist sampling theorem to determine the sampling rate, on the other hand, would result in an excessively large value of N and inefficient computation. The only deviation from exactness arises from the approximation of a continuous Fourier transform with the discrete Fourier transform (DFT). Thus, the algorithms compute LCT integrals with a performance similar to that of the fast Fourier transform (FFT) algorithm in digitally computing the continuous Fourier transform (FT), both in terms of speed and accuracy.

In an earlier paper [11], we had presented an algorithm which maps the samples of a given function to the samples of its ath-order fractional Fourier transform, which had the same efficiency and accuracy as the FFT in computing the FT. The present work represents the results of our efforts to obtain a much more general algorithm that can handle *all* members of the three-parameter family of LCTs with a similar performance.

In this paper, we discuss algorithms for numerically computing continuous LCTs with careful attention to sampling issues. Another avenue towards this end is to first define discrete linear canonical transforms (DLCTs), use them to approximate continuous LCTs much as the DFT is used to approximate the continuous FT, and then compute the DLCT using a fast algorithm [12]. There has been a certain amount of work on defining discrete/finite fractional Fourier transforms and, to a much lesser degree, discrete/finite linear canonical transforms [13]–[32]. While definitions of the discrete fractional Fourier transform (DFRT) may be considered satisfactory and well recognized [22], [30], [32], definition of the DLCT is far from being established. Further work on the definition and fast computation of discrete transforms, and their relationship to their continuous counterparts is desirable.

The paper is organized as follows. LCTs are reviewed in Section II, and Section III presents a systematic analysis of all possible decompositions of an LCT into the three basic operations of scaling, Fourier transformation and chirp multiplication. Based on this, the first algorithm is also presented

A. Koç is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: aykutkoc@stanford.edu).

H. M. Ozaktas is with the Department of Electrical Engineering, Bilkent University, TR-06800, Bilkent, Ankara, Turkey (e-mail: haldun@ee.bilkent.edu.tr).

C. Candan is with the Department of Electrical Engineering, Middle East Technical University, TR-06531, Ankara, Turkey (e-mail: ccandan@metu.edu.tr).

M. A. Kutay is with the TUBITAK-UEKAE National Research Institute of Electronics and Cryptology, TR-06100, Kavaklidere, Ankara, Turkey (e-mail: alper.kutay@iltaren.tubitak.gov.tr).

in Section III. The second approach involving the fractional Fourier transform is given in Section IV. Numerical examples and comparison of the two approaches are presented in Section V. In Section VI, the conclusion is given.

II. PRELIMINARIES

A. Linear Canonical Transforms

The linear canonical transform of $f(u)$ with parameter matrix $\mathbf{M}$ is denoted as $f_{\mathbf{M}}(u) = (\mathcal{C}_{\mathbf{M}}f)(u)$:

$$(\mathcal{C}_{\mathbf{M}}f)(u) = \sqrt{\beta}e^{-i\pi/4}\int_{-\infty}^{\infty}\exp\left[i\pi(\alpha u^2 - 2\beta uu' + \gamma u'^2)\right] \\ f(u')du' \quad (1)$$

where α, β, and γ are real parameters independent of u and u' and where $\mathcal{C}_{\mathbf{M}}$ is the LCT operator. The transform is unitary. The 2×2 matrix $\mathbf{M}$ whose elements are A, B, C, D, represents the same information as the three parameters α, β, and γ which uniquely define the LCT:

$$\mathbf{M} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \frac{\gamma}{\beta} & \frac{1}{\beta} \\ -\beta + \frac{\alpha\gamma}{\beta} & \frac{\alpha}{\beta} \end{bmatrix} = \begin{bmatrix} \frac{\alpha}{\beta} & \frac{-1}{\beta} \\ \beta - \frac{\alpha\gamma}{\beta} & \frac{\gamma}{\beta} \end{bmatrix}^{-1}. \quad (2)$$

The unit-determinant matrix $\mathbf{M}$ belongs to the class of unimodular matrices. More on the group-theoretical structure of LCTs may be found in [1] and [2].

The result of repeated application (concatenation) of LCTs can be handled easily with the above-defined matrix. When two or more LCTs are cascaded, the resulting transform is again an LCT whose matrix is given by multiplying the matrix of each LCT in the cascade structure. That is, if two LCTs with matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ operate in a successive manner, then the equivalent transform is an LCT with the matrix $\mathbf{M}_3 = \mathbf{M}_2\mathbf{M}_1$. LCTs are not commutative. The matrix of the inverse of an LCT is simply another LCT whose matrix is the inverse of the matrix of the original LCT [1], [2].

This paper studies different decompositions (or factorizations) of the given LCT into other LCTs with the purpose of fast and accurate calculation of the LCT integral. Different decompositions may be advantageous for LCTs with different parameters. The use of matrices will greatly facilitate our study of different decompositions, since dealing directly with the corresponding integral expressions is quite cumbersome.

Computation of the Fresnel diffraction integral, which is a special case of (1) with $\alpha = \beta = \gamma$, has received the greatest attention since it describes the propagation of light in free space (see [33] and [34] and the references therein). Since the input–output relationship represented by the Fresnel integral is time-invariant and takes the form of a convolution, it can be computed in $\sim N \log N$ time. The algorithms we present can compute (1) in $\sim N \log N$ time, despite the fact that the relationship represented by the more general (1) is not time-invariant and is not a convolution. It is important to underline that here N is chosen close to the time–bandwidth product of the set of input signals, which is usually the smallest possible value of N that can be chosen in terms of information-theoretic considerations. Therefore, the presented algorithms are highly efficient. Indeed, the distinguishing feature of the present approach is the care with which sampling and time–bandwidth product issues are handled. Straightforward use of conventional numerical methods

can result in inefficiencies either because their complexity is larger than $\sim N \log N$ and/or because the highly oscillatory quadratic-phase kernel in (1) forces N to be chosen much larger than the time–bandwidth product of the signals [35].

B. Relation of LCTs to the Wigner Distribution

Here we will review the relationship between LCTs and the Wigner distribution, which will aid us in understanding the effects of the elementary blocks used in our decompositions. The Wigner distribution $W_f(u, \mu)$ of a signal $f(u)$ can be defined as follows [36], [37]:

$$W_f(u,\mu) = \int_{-\infty}^{\infty} f\left(u + \frac{u'}{2}\right) f^*\left(u - \frac{u'}{2}\right) e^{-2\pi i\mu u'} du'. \quad (3)$$

Roughly speaking, $W(u, \mu)$ is a function which gives the distribution of signal energy over time and frequency. Its integral over time and frequency, $\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}W(u,\mu)dud\mu$, gives the signal energy.

Let f denote a signal and $f_{\mathbf{M}}$ be its LCT with parameter matrix $\mathbf{M}$. Then, the Wigner distribution (WD) of $f_{\mathbf{M}}$ can be expressed in terms of the WD of f as [2]

$$W_{f_{\mathbf{M}}}(u,\mu) = W_f(Du - B\mu, -Cu + A\mu). \quad (4)$$

This means that the WD of the transformed signal is a linearly distorted version of the original distribution. The Jacobian of this coordinate transformation is equal to the determinant of the matrix $\mathbf{M}$, which is unity. Therefore, this coordinate transformation does not change the support area of the Wigner distribution. (A precise definition of the support area is not necessary for the purpose of this paper; it may be defined as the area of the region where the values of the Wigner distribution are non-negligible, or the area of a region containing a certain high percentage of the total energy.) The invariance of support area means that LCTs do not concentrate or deconcentrate energy. The support area of the Wigner distribution can also be approximately interpreted as the number of degrees of freedom of the signal. Therefore, the number of samples needed to represent the signal does not change after an LCT operation.

It is well known that a nonzero function and its FT cannot both be confined to finite intervals. However, in practice we always work with samples of finite duration signals. We assume that a large percentage of the signal energy, as represented by the WD, is confined to an ellipse with diameters ΔT in the time dimension and ΔB in the frequency dimension, which can be ensured by choosing ΔT and ΔB suitably. This implies that the time-domain representation is approximately confined to the interval $[-\Delta T/2, \Delta T/2]$ and that the frequency-domain representation is approximately confined to $[-\Delta B/2, \Delta B/2]$. We then define the time–bandwidth product $\Delta T\Delta B$, which is always ≥ 1, because of the uncertainty relation. Let us now introduce the scaling parameter s and scaled coordinates, such that the time- and frequency-domain representations are confined to intervals of length $\Delta T/s$ and ΔBs. Let $s = \sqrt{\Delta T/\Delta B}$ so that the lengths of both intervals become equal to the dimensionless quantity $\sqrt{\Delta B\Delta T}$ which we denote by Δu. In the new coordinates, signals can be represented in both domains with Δu^2 samples spaced Δu^{-1} apart. We will assume that this dimensional normalization has been performed and that the coordinates u and μ are dimensionless.
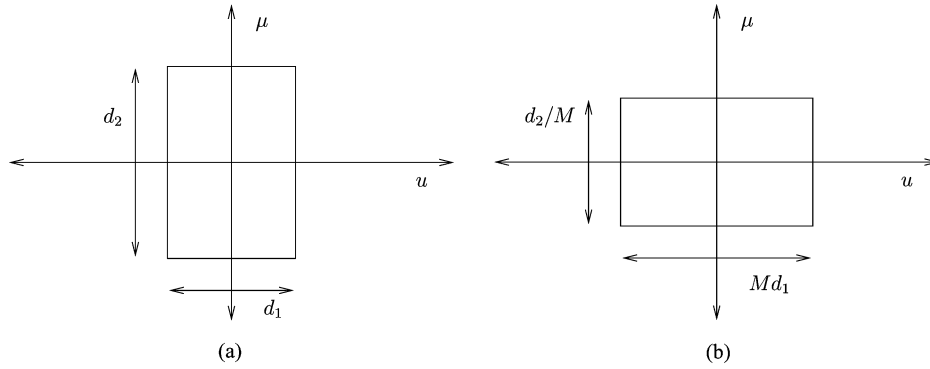
Fig. 1. Effect of scaling on the Wigner distribution. (a) Before scaling operation. (b) After scaling operation with parameter M.
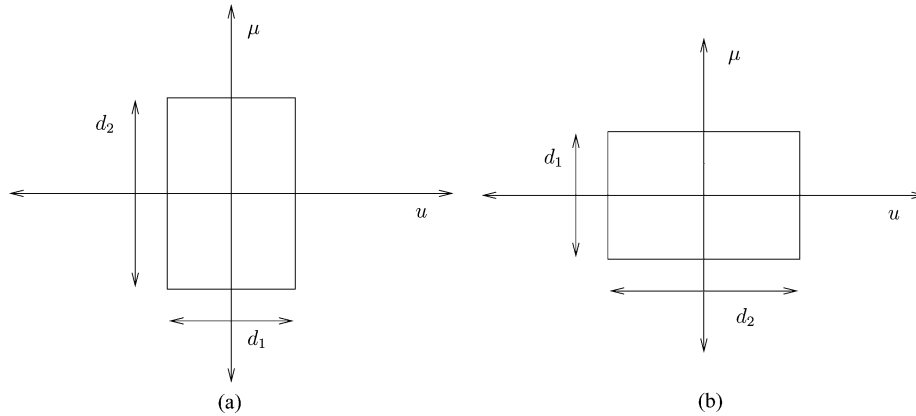

Fig. 2. Effect of Fourier transformation on the Wigner distribution. (a) Before Fourier transformation. (b) After Fourier transformation.

For a signal with rectangular time-frequency support, the time-bandwidth product is equal to the number of degrees of freedom. This is not true for signals with other support shapes. While we have observed that LCTs do not change the number of degrees of freedom of a signal, they may change its time–bandwidth product. This will be illustrated in what follows.

C. Effects of Special LCTs on the Wigner Distribution

Here we discuss the effects of certain operations, all special cases of LCTs, on the Wigner distribution of a signal. These are of interest since we will decompose general LCTs in terms of these operations.

1) Scaling: The scaling operation is a special case of the LCT defined as

$$\mathcal{C}_{\mathbf{M}_M} f(u) = \mathcal{M}_M f(u) = \sqrt{\frac{1}{M}} f\left(\frac{u}{M}\right) \tag{5}$$

$$\mathbf{M}_M = \begin{bmatrix} M & 0 \\ 0 & \frac{1}{M} \end{bmatrix} = \begin{bmatrix} \frac{1}{M} & 0 \\ 0 & M \end{bmatrix}^{-1}. \tag{6}$$

Its effect on the WD is given by

$$W_{\mathcal{M}_M f}(u, \mu) = W_f\left(\frac{u}{M}, M\mu\right) \tag{7}$$

where $M > 0$. The scaling operation does not change the support area, time-bandwidth product, or required number of samples (Fig. 1), but it changes the sampling intervals in both the time and frequency domains by factors of M and $1/M$ respectively.

2) Fourier Transformation: The ordinary Fourier transform operation is also a special case of the LCT

$$\mathcal{C}_{\mathbf{F}_{\mathrm{lc}}} f(u) = \mathcal{F}_{\mathrm{lc}} f(u) = e^{-i\pi/4} \int_{-\infty}^{\infty} f(u') e^{-i2\pi u u'} du' \tag{8}$$

$$\mathbf{F}_{\mathrm{lc}} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}^{-1}. \tag{9}$$

The subscript "lc" reminds us that the definition of the Fourier transform as a special case of LCTs differs from the conventional definition by the factor $e^{-i\pi/4}$. Readers wishing to understand the technical reason behind this inconsequential discrepancy may consult [1], [2]. The effect of Fourier transformation on the WD is given by

$$W_{\mathcal{F}_{\mathrm{lc}} f}(u, \mu) = W_f(-\mu, u) \tag{10}$$

which is a rotation of $\pi/2$ in the clockwise direction (Fig. 2), which again does not change the time–bandwidth product.

3) Chirp Multiplication: The chirp multiplication (CM) operation is another special case of the LCT

$$\mathcal{C}_{\mathbf{Q}_q} f(u) = \mathcal{Q}_q f(u) = e^{-i\pi q u^2} f(u) \tag{11}$$

$$\mathbf{Q}_q = \begin{bmatrix} 1 & 0 \\ -q & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ q & 1 \end{bmatrix}^{-1}. \tag{12}$$

Its effect on the WD is

$$W_{\mathcal{Q}_q f}(u, \mu) = W_f(u, \mu + qu). \tag{13}$$

Although the support area and therefore the number of degrees of freedom are preserved after chirp multiplication, the time–bandwidth product increases. This is due to the increase in signal bandwidth as a result of vertical shearing of the WD (Fig. 3). The new time–bandwidth product is $d_1(d_2 + |q|d_1)$. If we wish a signal with such a support to be recoverable from its samples in the conventional manner, this is the number of
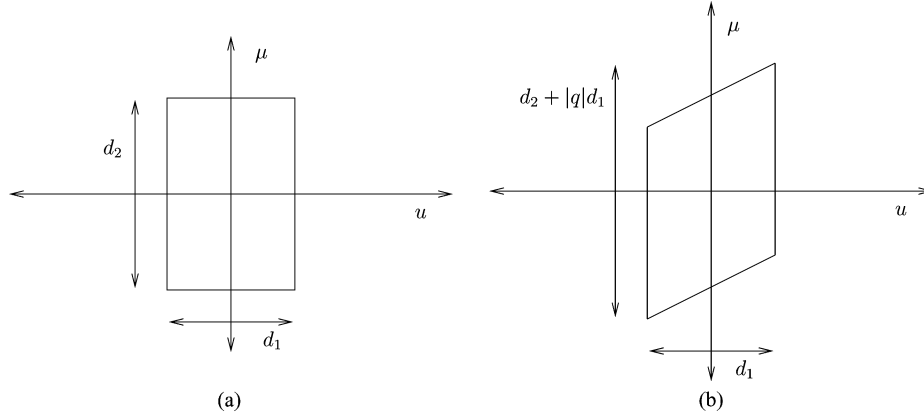
Fig. 3.   Effect of chirp multiplication on the Wigner distribution. (a) Before chirp multiplication. (b) After chirp multiplication.
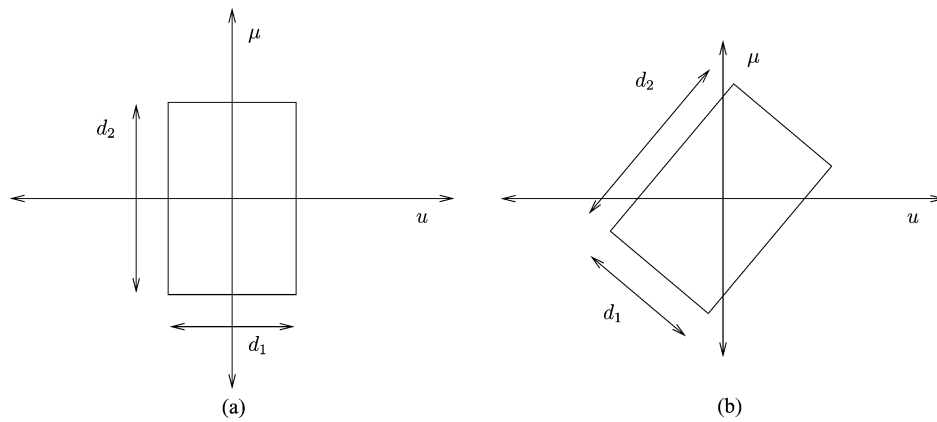

Fig. 4.   Effect of fractional Fourier transformation on the Wigner distribution. (a) Before FRT operation. (b) After FRT operation.

samples we need. (The sampling interval in the time domain is $1/(d_2 + |q|d_1)$ and that in the frequency domain is $1/d_1$.)

The chirp convolution (CC) operation is the dual of the chirp multiplication and corresponds to a horizontal, rather than a vertical shear in the time-frequency plane:

$$\mathcal{C}_{\mathbf{R}_r} f(u) = \mathcal{R}_r f(u) = f(u) * e^{-i\pi/4} \sqrt{\frac{1}{r}} \exp\left(\frac{i\pi u^2}{r}\right) \tag{14}$$

$$\mathbf{R}_r = \begin{bmatrix} 1 & r \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -r \\ 0 & 1 \end{bmatrix}^{-1}. \tag{15}$$

Its effect on the WD is

$$W_{\mathcal{R}_r f}(u, \mu) = W_f(u - r\mu, \mu). \tag{16}$$

4) Fractional Fourier Transformation: The ath-order fractional Fourier transform (FRT) [2], [38]–[44] of a function $f(u)$, denoted $f_a(u)$, is most commonly defined as

$$\mathcal{F}^a f(u) = f_a(u) = \int_{-\infty}^{\infty} K_a(u, u') f(u') du',$$
$$K_a(u, u') = A_\theta \exp\left[i\pi(\cot\theta u^2 - 2\csc\theta uu' + \cot\theta u'^2)\right],$$
$$A_\theta = \sqrt{1 - i\cot\theta}, \quad \theta = \frac{a\pi}{2} \tag{17}$$

when $a \neq 2j$ and $K_a(u, u') = \delta(u - u')$ when $a = 4j$ and $K_a(u, u') = \delta(u + u')$ when $a = 4j \pm 2$, where j is an integer. The square root is defined such that the argument of the result

lies in the interval $(-\pi/2, \pi/2]$. For $0 < |a| < 2\ (0 < |\theta| < \pi)$, A_θ can be rewritten without ambiguity as

$$A_\theta = \frac{e^{-i[\pi\mathrm{sgn}(\theta)/4 - \theta/2]}}{\sqrt{|\sin\theta|}} \tag{18}$$

where $\mathrm{sgn}(\cdot)$ is the sign function. When a is outside the interval $0 \leq |a| \leq 2$, we need simply replace a by its modulo 4 equivalent lying in this interval and use this value in (18).

The FRT is also a special case of the LCT with matrix

$$\mathbf{F}_{\mathrm{lc}}^a = \begin{bmatrix} \cos\left(\frac{a\pi}{2}\right) & \sin\left(\frac{a\pi}{2}\right) \\ -\sin\left(\frac{a\pi}{2}\right) & \cos\left(\frac{a\pi}{2}\right) \end{bmatrix}$$
$$= \begin{bmatrix} \cos\left(\frac{a\pi}{2}\right) & -\sin\left(\frac{a\pi}{2}\right) \\ \sin\left(\frac{a\pi}{2}\right) & \cos\left(\frac{a\pi}{2}\right) \end{bmatrix}^{-1} \tag{19}$$

differing only by the factor $e^{-ia\pi/4}$:

$$\mathcal{C}_{\mathbf{F}_{\mathrm{lc}}^a} f(u) = \mathcal{F}_{\mathrm{lc}}^a f(u) = e^{-ia\pi/4} \mathcal{F}^a f(u). \tag{20}$$

Again the subscript "lc" denotes this inconsequential discrepancy between the definition of the FRT given by (17) and the FRT defined as a special case of the LCT [1], [2]. The FRT rotates the WD in the clockwise direction with an angle of $\theta = a\pi/2$, as shown in Fig. 4:

$$W_{\mathcal{F}_{\mathrm{lc}}^a f} = W_f\left[\cos\left(\frac{a\pi}{2}\right)u - \sin\left(\frac{a\pi}{2}\right)\mu,\right.$$
$$\left. \sin\left(\frac{a\pi}{2}\right)u + \cos\left(\frac{a\pi}{2}\right)\mu\right]. \tag{21}$$

III. Systematic Analysis of Decompositions and Method I

The proposed algorithms use matrix factorizations to decompose LCTs into cascade combinations of the elementary LCT blocks discussed above. Since each stage can be computed in $\sim N \log N$ time, the overall LCT can also be. Numerous such decompositions are possible [2], but they are not equally suited for numerical purposes. For instance, direct naive application of the decomposition of chirp multiplication, Fourier transformation, scaling (magnification), and again chirp multiplication, which suggests itself upon inspection of (1) will in general lead to very high sampling rates. We have carried out a systematic exhaustive analysis of all possible decompositions of arbitrary LCTs into the three basic operations of scaling, chirp multiplication (CM), and Fourier transformation (FT). We have considered all possible decompositions with three, four, and five cascade blocks. Every permutation has been checked to see if that decomposition is capable of expressing an LCT with arbitrary parameters.

It is well known that arbitrary LCTs can be decomposed in either the form $\mathcal{Q}_{q_1} \mathcal{R}_r \mathcal{Q}_{q_2}$ or $\mathcal{R}_{r_1} \mathcal{Q}_q \mathcal{R}_{r_2}$, where $\mathcal{Q}_q$ and $\mathcal{R}_r$ are the chirp multiplication and convolution operations respectively [2]. Chirp convolutions can be realized as a Fourier transform followed by a chirp multiplication followed by an inverse Fourier transform (the Fourier transform of a chirp is also a chirp). Since we already consider all permutations involving chirp multiplication and Fourier transformation, approaches involving chirp convolution are also included in our development.

While generating all possible decompositions through permutations, duplicate decompositions arise. For example, two consecutive scaling or two consecutive chirp multiplication operations can both be collapsed into one. Or, for instance, in the case of five-stage decompositions with more free parameters than the three free parameters of LCTs, we have the freedom of choosing the additional parameters. When we select the scaling parameter as 1, this reduces to an equivalent four stage decomposition. In other words, some decompositions turn out to be equivalent to others, reducing the total number of possible decompositions.

Careful consideration shows that there are a total of 16 distinct decompositions. Four of these have four stages and 12 of them have five stages; decompositions with three stages are not flexible enough to match arbitrary LCTs. However, some decompositions among this 16 are equivalent in implementation. For example a scaling and FT cascade can be replaced with a FT and inverse scaling cascade, a trivial and inconsequential difference. When such trivial replacements are deducted from the set of 16 decompositions, we end up with 12 decompositions. We will immediately eliminate two of these decompositions. These two decompositions involve significantly more computational load than the others. As will be seen, the CM stages, especially when they occur early in the cascade, require us to increase the sampling rate and thus the complexity. Generally speaking it is desirable to have as few CM stages as possible and to have them appear as late in the cascade as possible. Therefore, we eliminate the two decompositions having three CM stages. Finally, our exhaustive consideration and careful sorting out of all decompositions with three, four, and five stages leaves us with ten distinct decompositions to consider: see equations (22)–(31) shown at the bottom of the page. Recall that the diagonal ma-

$$\mathbf{M} = \begin{bmatrix} \frac{1}{\beta} & 0 \\ 0 & \beta \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{\alpha}{\beta^2} & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma & 1 \end{bmatrix} \tag{22}$$

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{\gamma}{\beta^2} & 1 \end{bmatrix} \begin{bmatrix} \beta & 0 \\ 0 & \frac{1}{\beta} \end{bmatrix} \tag{23}$$

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\beta} & 0 \\ 0 & \beta \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma & 1 \end{bmatrix} \tag{24}$$

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ \alpha - \frac{\beta^2}{\gamma} & 1 \end{bmatrix} \begin{bmatrix} -\frac{\gamma}{\beta} & 0 \\ 0 & -\frac{\beta}{\gamma} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{\gamma} & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{25}$$

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ \alpha - \frac{\beta^2}{\gamma} & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{\gamma}{\beta^2} & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -\frac{\gamma}{\beta} & 0 \\ 0 & -\frac{\beta}{\gamma} \end{bmatrix} \tag{26}$$

$$\mathbf{M} = \begin{bmatrix} -\frac{\gamma}{\beta} & 0 \\ 0 & -\frac{\beta}{\gamma} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\gamma + \alpha\frac{\gamma^2}{\beta^2} & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{\gamma} & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{27}$$

$$\mathbf{M} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -\frac{\alpha}{\beta} & 0 \\ 0 & -\frac{\beta}{\alpha} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{\alpha}{\beta^2} & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma - \frac{\beta^2}{\alpha} & 1 \end{bmatrix} \tag{28}$$

$$\mathbf{M} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{\alpha} & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -\frac{\beta}{\alpha} & 0 \\ 0 & -\frac{\alpha}{\beta} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma - \frac{\beta^2}{\alpha} & 1 \end{bmatrix} \tag{29}$$

$$\mathbf{M} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{\alpha} & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\alpha + \gamma\frac{\alpha^2}{\beta^2} & 1 \end{bmatrix} \begin{bmatrix} -\frac{\beta}{\alpha} & 0 \\ 0 & -\frac{\alpha}{\beta} \end{bmatrix} \tag{30}$$

$$\mathbf{M} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{\alpha} & 1 \end{bmatrix} \begin{bmatrix} -\frac{\alpha}{\beta} & 0 \\ 0 & -\frac{\beta}{\alpha} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma - \frac{\beta^2}{\alpha} & 1 \end{bmatrix}. \tag{31}$$
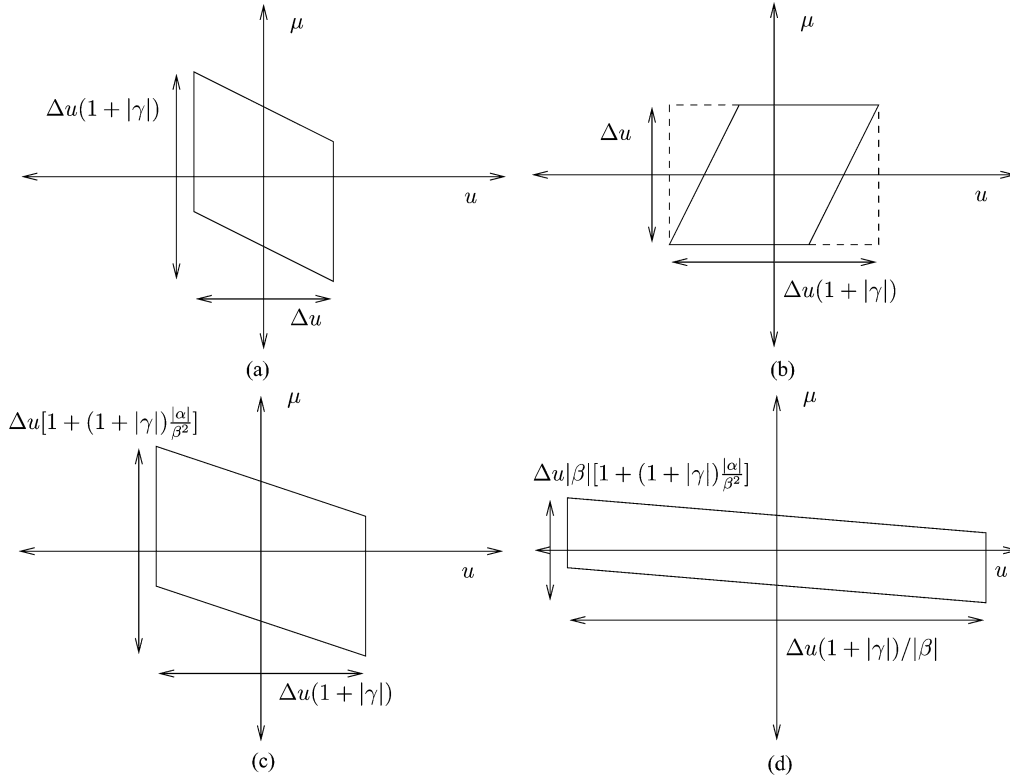
Fig. 5. Sequence of geometrical distortions for the decomposition in (22). The parallelogram in (c) is obtained by shearing the dashed rectangle in (b) in order to cover the worst case. (a) After the first stage: CM I. (b) After the second stage: FT. (c) After the third stage: CM II. (d) After the fourth stage: scaling.

trices correspond to scaling, the skew-diagonal ones to FT, and the lower-triangular ones to CM. The parameters α, β, γ of these operations are chosen in terms of the elements A, B, C, D of the matrix $\mathbf{M}$ so as to equate the left and right hands of the above equations [see (2)].

Scaling and FT do not require an increase in the number of samples. However, since CM will change the time-bandwidth product, we introduce $\times 2$ oversampling when confronted with the first CM operation, to allow us room to maneuver. We will however try to avoid oversampling beyond this, as much as possible. Specifically, we will impose any necessary restrictions so as to avoid further oversampling until the stage that involves the very last CM.

We will make sure that after each stage, the number of samples is sufficient (in the Nyquist sense) for recovery of the continuous signal. In order to illustrate how we deal with each decomposition, we consider the decomposition given in (22) as an example (Fig. 5). For graphical purposes, here we assumed that the initial time-frequency support is a square of edge length Δu rather than a circle. Since we require that $2N$ samples be sufficient, we must ensure that the time–bandwidth product does not exceed $2N$:

$$\Delta u \times \Delta u(1 + |\gamma|) \leq 2N = 2\Delta u^2 \Rightarrow |\gamma| \leq 1 \Rightarrow -1 \leq \gamma \leq 1.$$
$$(32)$$

Therefore, we obtain the restriction $-1 \leq \gamma \leq 1$ for the parameter γ appearing in the first chirp multiplication operation. This restriction ensures that oversampling by two is sufficient,

by ensuring that the bandwidth following the geometric distortion has not increased by more than a factor of two. After the FT, the second chirp multiplication operation, and the scaling operation, the time-bandwidth product becomes

$$\Delta u (1 + |\gamma|) \Delta u \left[1 + \frac{|\alpha|}{\beta^2} (1 + |\gamma|) \right]. \qquad (33)$$

Note that the parallelogram in Fig. 5(c) is obtained by shearing the dashed rectangle in Fig. 5(b) in order to cover the worst case. Is the above expression greater than $2N = 2\Delta u^2$ and if so, how greater is it? Let $k/2$ denote the additional oversampling factor required. Equating the above expression to the number of samples $k\Delta u^2$ corresponding to a total of k oversampling, leads us to the minimum value of k as

$$k \geq 1 + |\gamma| + \frac{|\alpha|(1 + |\gamma|)^2}{\beta^2}. \qquad (34)$$

If the right-hand side of this expression turns out to be ≤ 2, that means that we do not need any additional oversampling, in which case we simply take $k = 2$. Before continuing, we also note that the scaling operation merely changes the sampling interval in the sense of reinterpretation of the same samples with a scaled sampling interval, in a manner which corresponds to scaling of the underlying continuous signal.

Thus, by carefully following the evolution of the time-frequency support region through each stage of the decomposition, we have obtained i) any necessary restrictions on the parameters

TABLE I
RESTRICTIONS AND OVERSAMPLING FACTORS

Decomposition	Restriction	Oversampling factor
Eq. 22	$|\gamma| \leq 1$	$k \geq 1 + |\gamma| + \frac{|\alpha|(1+|\gamma|)^2}{\beta^2}$
Eq. 23	$|\gamma| \leq 1$	$k \geq 1 + |\gamma| + \frac{|\alpha|(1+|\gamma|)^2}{\beta^2}$
Eq. 24	$|\gamma| \leq 1$	$k \geq 1 + |\gamma| + \frac{|\alpha|(1+|\gamma|)^2}{\beta^2}$
Eq. 25	$|\gamma| \geq 1$	$k \geq 1 + \frac{1}{|\gamma|} + \frac{(1+|\gamma|)^2}{\beta^2}|\alpha - \frac{\beta^2}{\gamma}|$
Eq. 26	$|\gamma| \geq 1$	$k \geq 1 + \frac{1}{|\gamma|} + \frac{(1+|\gamma|)^2}{\beta^2}|\alpha - \frac{\beta^2}{\gamma}|$
Eq. 27	$|\gamma| \geq 1$	$k \geq 1 + \frac{1}{|\gamma|} + \frac{(1+|\gamma|)^2}{\beta^2}|\alpha - \frac{\beta^2}{\gamma}|$
Eq. 28	$|\gamma - \frac{\beta^2}{\alpha}| \leq 1$	$k \geq 1 + |\gamma - \frac{\beta^2}{\alpha}| + |\frac{\alpha}{\beta^2}|\left[1 + |\gamma - \frac{\beta^2}{\alpha}|\right]^2$
Eq. 29	$|\gamma - \frac{\beta^2}{\alpha}| \leq 1$	$k \geq 1 + |\gamma - \frac{\beta^2}{\alpha}| + |\frac{\alpha}{\beta^2}|\left[1 + |\gamma - \frac{\beta^2}{\alpha}|\right]^2$
Eq. 30	$|\gamma - \frac{\beta^2}{\alpha}| \leq 1$	$k \geq 1 + |\gamma - \frac{\beta^2}{\alpha}| + |\frac{\alpha}{\beta^2}|\left[1 + |\gamma - \frac{\beta^2}{\alpha}|\right]^2$
Eq. 31	$|\gamma - \frac{\beta^2}{\alpha}| \leq 1$	$k \geq 1 + |\gamma - \frac{\beta^2}{\alpha}| + |\frac{\alpha}{\beta^2}|\left[1 + |\gamma - \frac{\beta^2}{\alpha}|\right]^2$

TABLE II
COMPUTATIONAL COMPLEXITIES. $I(x, y)$ STANDS FOR THE COST
TO INTERPOLATE x SAMPLES BY A FACTOR OF y TO OBTAIN
xy SAMPLES AND $S(z)$ STANDS FOR THE COST OF THE SCALING OPERATION
ON z SAMPLES

Decomposition	Computational Complexity
Eq. 22	$2N + (2N)\log(2N) + Nk + I(N,2) + I(2N, k/2) + S(Nk)$
Eq. 23	$2N + (2N)\log(2N) + Nk + I(N,2) + I(2N, k/2) + S(N)$
Eq. 24	$2N + (2N)\log(2N) + Nk + I(N,2) + I(2N, k/2) + S(2N)$
Eq. 25	$2N + (2N)\log(2N) + N\log N + Nk + I(N,2) + I(2N, k/2) + S(2N)$
Eq. 26	$2N + (2N)\log(2N) + N\log N + Nk + I(N,2) + I(2N, k/2) + S(N)$
Eq. 27	$2N + (2N)\log(2N) + N\log N + Nk + I(N,2) + I(2N, k/2) + S(Nk)$
Eq. 28	$2N + (2N)\log(2N) + (Nk)\log(Nk) + Nk + I(N,2) + I(2N, k/2) + S(Nk)$
Eq. 29	$2N + (2N)\log(2N) + (Nk)\log(Nk) + Nk + I(N,2) + I(2N, k/2) + S(2N)$
Eq. 30	$2N + (2N)\log(2N) + (Nk)\log(Nk) + Nk + I(N,2) + I(2N, k/2) + S(N)$
Eq. 31	$2N + (2N)\log(2N) + (Nk)\log(Nk) + Nk + I(N,2) + I(2N, k/2) + S(2N)$

of the stages appearing in the decomposition, so that oversampling beyond $\times 2$ is not required until the last CM stage, ii) the additional oversampling factor $k/2$ which may be needed before the last CM stage to fully represent the continuous output signal. We underline that these considerations are guided by our goal to accommodate arbitrary input signals. In those cases where there exists some *a priori* knowledge of the input signal or the signal is restricted to a particular class, it may be possible to customize the approach here with benefit.

The same procedure has been repeated for the decompositions given in (23)–(31), and the results are given in Table I. It will be convenient to choose k as the smallest integer ≥ 2 satisfying the applicable inequality.

We have also added together the computational complexity of each stage to obtain the overall computational complexity of each decomposition and presented these in Table II. These complexity expressions will be reconsidered with a specific expression for $I(x, y)$ in Section V.

Observation of Table I reveals a natural grouping of the ten decompositions. The decompositions given in (22)–(24) exhibit the same restriction and oversampling factors. Likewise, the decompositions given in (25)–(27) share the same restriction and oversampling factors. Moreover, the restrictions of these two groups are complementary, the first three can be used for $|\gamma| \leq 1$ and the last three can be used for $|\gamma| \geq 1$, spanning the whole parameter space of LCTs.

TABLE III
PERCENTAGE ERRORS FOR DIFFERENT FUNCTIONS F, TRANSFORMS T, AND
ALGORITHMS A

	A1 T1	A1 T2	A2 T1	A2 T2	DFT
F1	3.2×10^{-22}	9.5×10^{-22}	2.7×10^{-17}	6.6×10^{-17}	2.0×10^{-21}
F2	7.8×10^{-4}	8.1×10^{-4}	11×10^{-4}	9.9×10^{-4}	6.2×10^{-4}
F3	1.5	1.6	1.4	1.5	1.2
F4	9.7×10^{-2}	11×10^{-2}	8.9×10^{-2}	9.9×10^{-2}	8.3×10^{-2}

Finally, we observe that the computational complexity of the third and last group of four decompositions is larger than the others. This group has a term with complexity $(Nk)\log(Nk)$ due to the second CM operation being followed by a DFT operation. Therefore, we discard the last four decompositions belonging to the third group. It should be noted that our elimination procedure is primarily based on complexity as opposed to error. However, as we will see in Table III, the algorithm obtained produces results with errors as small as can be reasonably expected; therefore we have not really lost anything by basing our procedure primarily on complexity.

Use of the information summarized in the two tables finally leads us to choose to work with two decompositions, one from the first and one from the second group of three. To ensure that oversampling by two is sufficient, we ended up using two complementary decompositions for different regions of the parameter space. Since they have a slightly lower complexity, we will prefer to work with the second decomposition in both groups, whose advantage arises primarily from the relative positions of the scaling and CM operations. The algorithm can now be outlined as follows.

• If $|\gamma| \leq 1$, use the decomposition

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{\gamma}{\beta^2} & 1 \end{bmatrix} \begin{bmatrix} \beta & 0 \\ 0 & \frac{1}{\beta} \end{bmatrix}. \quad (35)$$

In operator notation

$$\mathcal{C}_\mathbf{M} = \mathcal{Q}_{-\alpha} J_{k/2} \mathcal{F}_{\text{lc}} \mathcal{Q}_{-\gamma/\beta^2} J_2 \mathcal{M}_\beta \quad (36)$$

where J_x represents the $\times x$ oversampling operation. The minimum value of k is

$$k \geq 1 + |\gamma| + \frac{|\alpha|(1 + |\gamma|)^2}{\beta^2}. \quad (37)$$

• If $|\gamma| > 1$, use the decomposition

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ \alpha - \frac{\beta^2}{\gamma} & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{\gamma}{\beta^2} & 1 \end{bmatrix}$$
$$\times \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -\frac{\gamma}{\beta} & 0 \\ 0 & -\frac{\beta}{\gamma} \end{bmatrix}. \quad (38)$$

In operator notation

$$\mathcal{C}_\mathbf{M} = \mathcal{Q}_{-\alpha+\beta^2/\gamma} J_{k/2} \mathcal{F}_{\text{lc}} \mathcal{Q}_{\gamma/\beta^2} J_2 \mathcal{F}_{\text{lc}} \mathcal{M}_{-\gamma/\beta}. \quad (39)$$

The minimum value of k is:

$$k \geq 1 + \frac{1}{|\gamma|} + \frac{(1 + |\gamma|)^2}{\beta^2} \left| \alpha - \frac{\beta^2}{\gamma} \right|. \quad (40)$$

A recent work also discusses the use of decompositions to analyze algorithms for related transforms [45]. This work also em-

phasizes the importance of tracking the time-bandwidth product through successive stages of the decomposition and sets forth a systematic, uniform, and general approach to determining the overall increase in time–bandwidth product of the final transformed signal. We have chosen to avoid unnecessary increases in the time-bandwidth product in the early stages to avoid increasing the number of samples until the last CM stage, where the major and unavoidable increase in sampling rate occurs.

IV. METHOD II

The second approach we discuss is based on the following decomposition involving the FRT, scaling, and chirp multiplication:

$$\mathbf{M} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -q & 1 \end{bmatrix} \begin{bmatrix} M & 0 \\ 0 & \frac{1}{M} \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}. \tag{41}$$

Here $\theta = a\pi/2$, where a is the order of the FRT, q is the chirp multiplication parameter, and M is the scaling factor. As we will see, these three parameters are sufficient to satisfy the above equality for arbitrary ABCD matrices, so that this decomposition is capable of representing arbitrary LCTs. Since the fast method proposed in [11] can be used for the computation of the FRT, this decomposition leads to a fast algorithm for LCTs. This decomposition was inspired by the optical interpretation in [44] and is also a special case of the widely known Iwasawa decomposition [46]–[48]. It was also proposed later in [35] and [45]. Fig. 6 illustrates the sequence of geometrical distortions corresponding to this decomposition, where the initial time-frequency support is a circle of diameter Δu.

If we multiply out the right-hand side of (41) and replace the matrix entries A, B, C, D with α, β, γ, we obtain

$$\begin{bmatrix} \frac{\gamma}{\beta} & \frac{1}{\beta} \\ -\beta + \frac{\alpha\gamma}{\beta} & \frac{\alpha}{\beta} \end{bmatrix}$$
$$= \begin{bmatrix} M\cos\theta & M\sin\theta \\ -qM\cos\theta - \frac{\sin\theta}{M} & -qM\sin\theta + \frac{\cos\theta}{M} \end{bmatrix} \tag{42}$$

which is equivalent to four equations which we can solve for a, q, M:

$$a = \left(\frac{2}{\pi}\right)\text{arccot}\gamma \tag{43}$$

$$M = \begin{cases} \frac{\sqrt{1+\gamma^2}}{\beta}, & \gamma \geq 0 \\ -\frac{\sqrt{1+\gamma^2}}{\beta}, & \gamma < 0 \end{cases} \tag{44}$$

$$q = \frac{\gamma\beta^2}{(1+\gamma^2)} - \alpha. \tag{45}$$

The ranges of the square root and the arccotangent both lie in $(-\pi/2, \pi/2)$. Fig. 6 shows the geometric effect of the decomposition stages on the WD of a function, which is rotation, scaling, and shearing, respectively. In operator notation this algorithm can be expressed as

$$\mathcal{C}_{\mathbf{M}} = \mathcal{Q}_q J_k \mathcal{M}_M \mathcal{F}_{\text{lc}}^a. \tag{46}$$

In this method, the first operation is a FRT, whose fast computation in $\sim N\log N$ time is presented in [11] and [49]. (Other works dealing with fast computation of the FRT include [50]
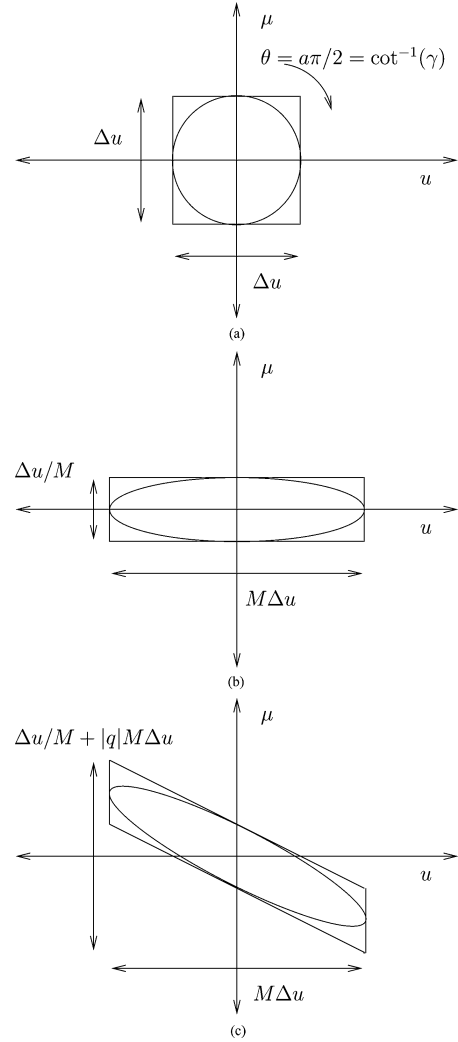

Fig. 6. Sequence of geometrical distortions for the decomposition in (41). (a) After the first stage: FRT. (b) After the second stage: scaling. (c) After the third stage: CM.

and [51].) The algorithm presented in [11] was based on decomposing the FRT into a CM followed by a CC followed by a final CM, and computed the samples of the continuous FRT in terms of the samples of the original signal. Just as in the present work, care was taken to ensure that the output samples represented the continuous FRT in the Nyquist–Shannon sense. The presently discussed algorithm employs the algorithm in [11] as a subroutine. The only approximation in this subroutine comes from the step involving chirp convolution in which a DFT/FFT is used to approximate the samples of the continuous FT. No other approximation need be made, either in this subroutine or in any of the other operations that we employ. Thus, the only source of approximation can be traced to the evaluation of a continuous FT by use of a DFT (implemented with a FFT), which is a consequence of the fundamental fact that the signal energy cannot be confined to finite intervals in both domains. The second operation in this method is scaling, which only involves a reinterpretation of the same samples with a scaled sampling interval. The final operation is CM which takes $\sim N$ time, leading to an overall complexity of $\sim N\log N$. (Detailed expressions will be given in Section V.) As in the first method, it is again necessary

to ensure that the output samples are sufficient to represent the transformed signal in the Nyquist–Shannon sense. Since LCTs distort the original time-frequency support, both the time and frequency extent of the signal, as well as its time–bandwidth product may increase, despite the fact that the area of the support remains the same. Therefore, a greater number of samples than Δu^2 may be needed to represent the transformed signal (unless we use some sophisticated basis to represent the signals) [35].

Delaying confrontation with the necessity to deal with this greater number of samples until the very last step is a significant advantage of this method. Since the FRT corresponds to rotation and scaling only to reinterpretation of the samples, these steps do not require us to increase the number of samples. At the last CM step, if we multiply the samples of the intermediate result with the samples of the chirp, the samples obtained will be good approximations of the true samples of the transformed signal at that sampling interval. If these samples are sufficient for our purposes, nothing further need be done. However, in general these samples will be below the Nyquist rate for the transformed signal and will not be sufficient for full recovery of the continuous function. To obtain a sufficient number of samples that will allow full recovery, we must interpolate the intermediate result at least by a factor k corresponding to the increase in time–bandwidth product

$$k \geq 1 + \left| \gamma - \frac{\alpha(1 + \gamma^2)}{\beta^2} \right|. \tag{47}$$

Again, for convenience we choose k to be the smallest integer satisfying this inequality.

V. NUMERICAL RESULTS AND COMPARISON OF METHODS

We have considered several examples to illustrate and compare the presented methods. We refer to the first algorithm involving Fourier transformation, scaling, and chirp multiplication as A1, and the second algorithm involving the fractional Fourier transform as A2. We consider the chirped pulse function $\exp(-\pi u^2 - i\pi u^2)$, denoted F1, and the trapezoidal function $1.5\mathrm{tri}(u/3) - 0.5\mathrm{tri}(u)$, denoted F2 ($\mathrm{tri}(u) = \mathrm{rect}(u) * \mathrm{rect}(u)$). Since these two functions are well confined to a circle with diameter $\Delta u = 8$ we take $N = 8^2$. We also consider the binary sequence 01101010 occupying $[-8, 8]$ with each bit 2 units in length, so that $N = 16^2$. This binary sequence is denoted by F3 and the function shown in Fig. 7 is denoted by F4, again with $N = 16^2$. These choices for Δu result in $\sim 0\%$, 0.0002%, 0.47%, 0.03% of the energies of F1, F2, F3, F4 respectively, to fall outside the chosen frequency extents. The chosen time extents include all of the energies of F2, F3, F4 and virtually all of the energy of F1. We consider two transforms, the first (T1) with parameters $(\alpha, \beta, \gamma) = (-3, -2, -1)$, and the second (T2) with parameters $(-4/5, 1, 2)$. The LCTs T1 and T2 of the functions F1, F2, F3, F4 have been computed both by the presented fast methods A1 and A2 and by a highly inefficient brute force numerical approach which is here taken as a reference.

The results for all functions (F1, F2, F3, F4) and both algorithms (A1, A2) are plotted in Fig. 8 for transform T1, and tabulated in Table III for both transforms (T1, T2). Also shown are the errors that arise when using the DFT in approximating the FT of the same functions, which serves as a reference. (The error
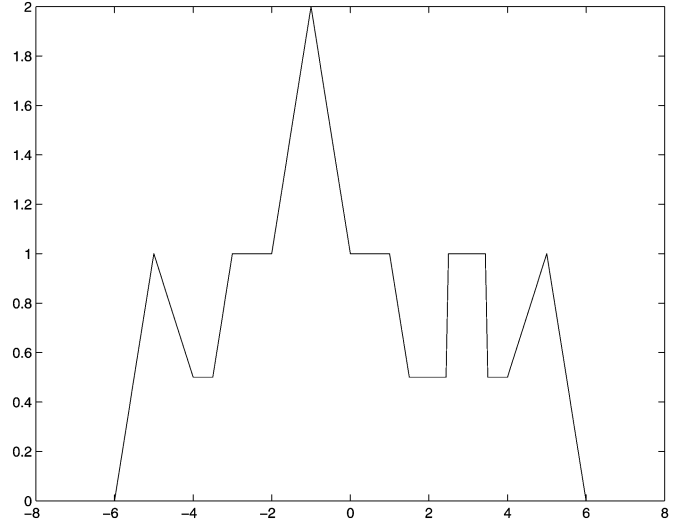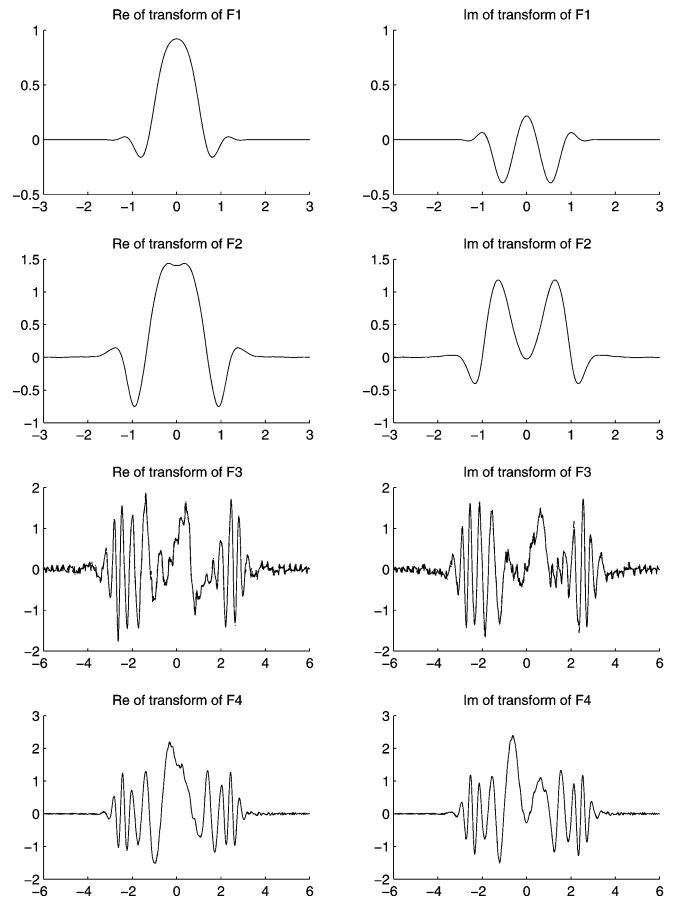

Fig. 7. Example function F4.

Fig. 8. Transforms (T1) of F1, F2, F3, F4. The results obtained with Methods I and II and the reference result have been plotted with dotted, dashed, and solid lines respectively. However, in most cases these lines are indistinguishable since the results are very close.

is defined as the energy of the difference normalized by the energy of the reference, expressed as a percentage.)

The key observations that can be made from this table are as follows. The errors obtained depend on the function, since different functions have different amounts of energy contained in their tails which fall outside the assumed time and frequency
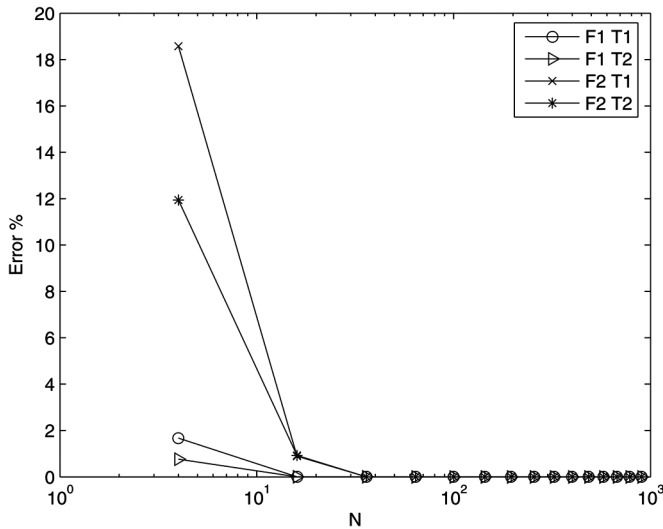
Fig. 9. Percentage errors versus N for selected functions and transforms [35].

extents (or assumed time-frequency region). For those cases in which the error is large, such as F3, this means that we have determined the time–bandwidth product less conservatively than the other examples, and the error can be reduced by increasing N. Generally speaking, the errors obtained depend very little on the transform parameters or which method we use, and are comparable to the error arising when we use the DFT to approximate the FT. Since a DFT lies at the heart of both methods, this is the smallest error we could have hoped to achieve to begin with.

Fig. 9 shows the error versus number of sample points N for selected functions and transforms. We observe that the error decreases steeply at first with increasing N as expected, but saturates when we approach and exceed the time-bandwidth product of the signals (here 64). This demonstrates that the number of samples N can be chosen comparable to the time–bandwidth product, which is the smallest number we can expect to work with, and need not be chosen larger. A2 was used to obtain this plot for illustration purposes but similar results can also be obtained when we use A1.

We now turn our attention to discussing the complexity (cost) of the algorithms as a function of N, the number of sample points. Based on the preceding paragraph, N can be chosen comparable to the time-bandwidth product so that the expressions given below can also be interpreted as functions of time-bandwidth product.

The computational complexity of the first method is given by either of the following expressions, depending on which decomposition is used (which was determined by whether $|\gamma| \le 1$ or not, respectively):

- $2N + 2N \log 2N + Nk + I(N, 2) + I(2N, k/2) + S(N)$;
- $2N + 2N \log 2N + N \log N + Nk + I(N, 2) + I(2N, k/2) + S(N)$.

On the other hand, for the second method, one of the following applies depending on the branch of the FRT algorithm used, which depends on whether $0.5 \le |a| \le 1$ or $|a| < 0.5$, respectively. Not surprisingly, this turns out to be the same as the condition $|\gamma| \le 1$ or $|\gamma| > 1$, respectively:

- $6N + 6N \log 2N + Nk + I(N, 2) + I(2N, k/2) + S(2N)$;
- $6N + 6N \log 2N + N \log N + Nk + I(N, 2) + I(2N, k/2) + S(2N)$.

The above expressions are derived by calculating the complexity of the FRT algorithm of [11] in its most efficient implementation $(6N + 6N \log 2N + I(N, 2) \quad [+N \log N])$, and adding the cost of the other operations.

Although included for completeness, the cost $S(\cdot)$ of the scaling operation is minimal and not of much consequence, since it amounts only to a reinterpretation of samples. In the above expressions, $I(x, y)$ stands for the cost to interpolate x samples by a factor of y to obtain xy samples. There are several efficient approximate interpolation methods which have complexities of order $\sim xy$[52]. We will write this cost as cxy where c is a constant.

Taking the difference of the costs of the two methods, Method II will have lower cost if

$$(1 + c)(k_I - k_{II}) > 4[1 + \log 2N] \qquad (48)$$

where k_I and k_{II} are associated with Method I and Method II respectively. If we go back to the steps of each method, we observe that it is usually possible to choose k_{II} much more tightly than k_I. Numerical simulations also confirm that k_{II} is usually smaller than k_I. Therefore, either method may turn out to have lower cost depending on the values of α, β, γ, N and it is not possible in general to declare one method superior over the other in this respect. If need be, both methods can be incorporated in the same code and the more efficient one invoked based on the parameters, but in many cases the difference may not be very significant. However, apart from its effect on cost of computation, having the lowest oversampling factor is desirable in itself since it produces an output represented with the least number of samples. This is a plus for Method II, which we also favor for its elegant construction.

Finally, we compare Method I and Method II, with the direct use of the well-known CM-CC-CM decomposition [2] without the kind of sampling rate management undertaken in this paper ("Method III"):

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ \alpha - \beta & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{\beta} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma - \beta & 1 \end{bmatrix} \qquad (49)$$

which is valid as long as $\beta \ne 0$. The CC stage has been implemented in the Fourier domain as a CM operation. At each step of the process, the sampling rate has been chosen as the minimum compatible with the shear-induced increases in the time and frequency extents. Considering T1, the minimum oversampling rate k is 5, 2, 8 for Methods I, II, and III, respectively. With the more demanding T2, the minimum value of k is 14, 7, 24, respectively. Larger values of k and thus larger total numbers of samples not only result in representational redundancy, but also they translate into greater computational time. For instance, for F4 and T2, the time of computation in seconds is 0.0910, 0.0396, 0.347 for Methods I, II, and III, respectively (obtained with MATLAB running on a personal computer), demonstrating that Method III is significantly slower. The corresponding percentage errors are comparable as expected: 0.11, 0.099, 0.10, respectively.

VI. CONCLUSION

In this paper, two algorithms for the computation of linear canonical transforms (LCTs) from the N samples of the input signal in $\sim N \log N$ time are discussed. Our approach is based on concepts from signal analysis and processing rather than conventional numerical analysis. With careful consideration of sampling issues, N can be chosen very close to the time-bandwidth product of the signals, and need not be much larger. The transform output may have a higher time-bandwidth product due to the nature of the transform family.

Both algorithms relate the samples of the input function to the samples of the continuous LCT of this function in the same sense that the fast Fourier transform (FFT) implementation of the discrete Fourier transform (DFT) computes the samples of the continuous FT of a function. Since the sampling rates are carefully controlled, the output samples obtained are accurate approximations to the true ones and the continuous LCT can be recovered via interpolation of these samples. The only inevitable source of deviation from exactness arises from the fundamental fact that a signal and its transform cannot both be of finite extent. This is the same source of deviation encountered when using the DFT/FFT to compute the continuous FT. Thus, the algorithms compute LCTs with a performance similar to the DFT/FFT in computing the Fourier transform, both in terms of speed and accuracy.

The fact that the two methods, although being arrived at from considerably different starting points, both exhibit similar limits in performance, strongly suggests that the performance achieved is close to the best achievable. Indeed, as already noted, it is difficult to expect an accuracy which is better than that of the DFT in approximating the FT, and a cost which is less than $\sim N \log N$ with N being close to the time–bandwidth product. Despite the different decompositions employed, an interesting structural similarity emerges between the two methods in their optimized forms. Both methods have two branches, the first one determined by whether $|\gamma| > 1$ or not, and the second one determined by whether $|a| < 0.5$ or not. If a is expressed in terms of the LCT parameters α, β, γ, we see that the two conditions are the same. Therefore, the separation of the LCT parameter space into two regions is most likely not a characteristic of the algorithm chosen, but an intrinsic structural property of the LCT parameter space.

Compared to earlier approaches, these algorithms not only handle a much more general family of integrals, but also effectively address certain difficulties, limitations, or tradeoffs that arise in other approaches to computing the Fresnel integral, which is of importance in the theory of diffraction (see [45] for a systematic comparison of several approaches). These algorithms can also be used for efficient realization of filtering in linear canonical transform domains [53].

ACKNOWLEDGMENT

REFERENCES

[1] K. B. Wolf, *Integral Transforms in Science and Engineering*. New York: Plenum, 1979, vol. Construction and Properties of Canonical Transforms, ch. 9.

[2] H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay, *The Fractional Fourier Transform with Applications in Optics and Signal Processing*. New York: Wiley, 2001.

[3] M. J. Bastiaans, "Wigner distribution function and its application to first-order optics," *J. Opt. Soc. Amer.*, vol. 69, pp. 1710–1716, 1979.

[4] M. J. Bastiaans, "Applications of the Wigner distribution function in optics," in *The Wigner Distribution: Theory and Applications in Signal Processing*, Amsterdam, The Netherlands, 1997, pp. 375–426.

[5] S. Abe and J. T. Sheridan, "Generalization of the fractional Fourier transformation to an arbitrary linear lossless transformation: An operator approach," *J. Phys. A*, vol. 27, pp. 4179–4187, 1994.

[6] S. Abe and J. T. Sheridan, "Optical operations on wavefunctions as the Abelian subgroups of the special affine Fourier transformation," *Optics Letters*, vol. 19, pp. 1801–1803, 1994.

[7] A. E. Siegman, *Lasers*. Mill Valley, CA: University Science Books, 1986.

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

[9] C. Palma and V. Bagini, "Extension of the Fresnel transform to $ABCD$ systems," *J. Opt. Soc. Amer. A*, vol. 14, pp. 1774–1779, 1997.

[10] J. Hua, L. Liu, and G. Li, "Extended fractional Fourier transforms," *J. Opt. Soc. Amer. A*, vol. 14, pp. 3316–3322, 1997.

[11] H. M. Ozaktas, O. Arikan, M. A. Kutay, and G. Bozdaği, "Digital computation of the fractional Fourier transform," *IEEE Trans. Signal Process.*, vol. 44, no. 9, pp. 2141–2150, Sep. 1996.

[12] B. M. Hennelly and J. T. Sheridan, "Fast numerical algorithm for the linear canonical transform," *J Opt. Soc. Amer. A*, vol. 22, pp. 928–937, 2005.

[13] N. M. Atakishiyev and K. B. Wolf, "Fractional Fourier-Kravchuk transform," *J. Opt. Soc. Amer. A*, vol. 14, pp. 1467–1477, 1997.

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

[15] N. M. Atakishiyev, S. M. Chumakov, and K. B. Wolf, "Wigner distribution function for finite systems," *J. Math. Phys.*, vol. 39, pp. 6247–6261, 1998.

[16] N. M. Atakishiyev, L. E. Vicent, and K. B. Wolf, "Continuous versus discrete fractional Fourier transforms," *J. Comput. Appl. Math.*, vol. 107, pp. 73–95, 1999.

[17] S. C. Pei, M. H. Yeh, and C. C. Tseng, "Discrete fractional Fourier transform based on orthogonal projections," *IEEE Trans. Signal Process.*, vol. 47, no. 5, pp. 1335–1348, May 1999.

[18] S.-C. Pei, M.-H. Yeh, and T.-L. Luo, "Fractional Fourier series expansion for finite signals and dual extension to discrete-time fractional Fourier transform," *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2883–2888, Oct. 1999.

[19] T. Erseghe, P. Kraniauskas, and G. Cariolaro, "Unified fractional Fourier transform and sampling theorem," *IEEE Trans. Signal Process.*, vol. 47, no. 12, pp. 3419–3423, Dec. 1999.

[20] M. A. Kutay, H. Özaktaş, H. M. Ozaktas, and O. Arikan, "The fractional Fourier domain decomposition," *Signal Process.*, vol. 77, pp. 105–109, 1999.

[21] A. I. Zayed and A. G. Garcia, "New sampling formulae for the fractional Fourier transform," *Signal Process.*, vol. 77, pp. 111–114, 1999.

[22] C. Candan, M. A. Kutay, and H. M. Ozaktas, "The discrete fractional Fourier transform," *IEEE Trans. Signal Process.*, vol. 48, no. 5, pp. 1329–1337, May 2000.

[23] I. Ş. Yetik, M. A. Kutay, H.Özaktaş, and H. M. Ozaktas, "Continuous and discrete fractional Fourier domain decomposition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 2000, pp. I:93–96.

[24] S. C. Pei and M. H. Yeh, "The discrete fractional cosine and sine transforms," *IEEE Trans. Signal Process.*, vol. 49, no. 6, pp. 1198–1207, Jun. 2001.

[25] G. Cariolaro, T. Erseghe, and P. Kraniauskas, "The fractional discrete cosine transform," *IEEE Transactions Signal Process.*, vol. 50, no. 4, pp. 902–911, Apr. 2002.

[26] L. Barker, "Continuum quantum systems as limits of discrete quantum systems. IV. Affine canonical transforms," *J. Math. Phys.*, vol. 44, pp. 1535–1553, 2003.

[27] C. Candan and H. M. Ozaktas, "Sampling and series expansion theorems for fractional Fourier and other transforms," *Signal Process.*, vol. 83, pp. 1455–1457, 2003.

[28] J. G. Vargas-Rubio and B. Santhanam, "On the multiangle centered discrete fractional Fourier transform," *IEEE Signal Process. Lett.*, vol. 12, no. 4, pp. 273–276, 2005.

[29] M. H. Yeh, "Angular decompositions for the discrete fractional signal transforms," *Signal Process.*, vol. 85, pp. 537–547, 2005.

[30] K. B. Wolf, "Finite systems, fractional Fourier transforms and their finite phase spaces," *Czech. J. Phys.*, vol. 55, pp. 1527–1534, 2005.

[31] K. B. Wolf, "Finite systems on phase space," *Int. J. Mod. Phys. B*, vol. 20, pt. 2 Sp. Iss, pp. 1956–1967, 2006.

[32] K. B. Wolf and G. Krotzsch, "Geometry and dynamics in the fractional discrete Fourier transform," *J. Opt. Soc. Amer. A*, vol. 24, pp. 651–658, 2007.

[33] D. Mendlovic, Z. Zalevsky, and N. Konforti, "Computation considerations and fast algorithms for calculating the diffraction integral," *J. Mod. Opt.*, vol. 44, pp. 407–414, 1997.

[34] D. Mas, J. Garcia, C. Ferreira, L. M. Bernardo, and F. Marinho, "Fast algorithms for free-space diffraction patterns calculation," *Opt. Commun.*, vol. 164, pp. 233–245, 1999.

[35] H. M. Ozaktas, A. Koç, I. Sari, and M. A. Kutay, "Efficient computation of quadratic-phase integrals in optics," *Opt. Lett.*, vol. 31, pp. 35–37, 2006.

[36] F. Hlawatsch and G. F. Boudreaux-Bartels, "Linear and quadratic time-frequency signal representations," *IEEE Signal Process. Mag.*, vol. 9, no. 2, pp. 21–67, Apr. 1992.

[37] L. Cohen, *Time-Frequency Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[38] D. Mendlovic and H. M. Ozaktas, "Fractional Fourier transforms and their optical implementation: I," *J. Opt. Soc. Amer. A*, vol. 10, pp. 1875–1881, 1993.

[39] H. M. Ozaktas and D. Mendlovic, "Fractional Fourier transforms and their optical implementation: II," *J. Opt. Soc. Amer. A*, vol. 10, pp. 2522–2531, 1993.

[40] H. M. Ozaktas and D. Mendlovic, "Fourier transforms of fractional order and their optical interpretation," *Opt. Commun.*, vol. 101, pp. 163–169, 1993.

[41] H. M. Ozaktas, B. Barshan, D. Mendlovic, and L. Onural, "Convolution, filtering, and multiplexing in fractional Fourier domains and their relation to chirp and wavelet transforms," *J. Opt. Soc. Amer. A*, vol. 11, pp. 547–559, 1994.

[42] L. B. Almeida, "The fractional Fourier transform and time-frequency representations," *IEEE Trans. Signal Process.*, vol. 42, no. 11, pp. 3084–3091, Nov. 1994.

[43] H. M. Ozaktas and D. Mendlovic, "Fractional Fourier optics," *J. Opt. Soc. Amer. A*, vol. 12, pp. 743–751, 1995.

[44] H. M. Ozaktas and M. F. Erden, "Relationships among ray optical, Gaussian beam, and fractional Fourier transform descriptions of first-order optical systems," *Opt. Commun.*, vol. 143, pp. 75–86, 1997.

[45] B. M. Hennelly and J. T. Sheridan, "Generalizing, optimizing, and inventing numerical algorithms for the fractional Fourier, Fresnel, and linear canonical transforms," *J. Opt. Soc. America. A*, vol. 22, pp. 917–927, 2005.

[46] T. Alieva and M. J. Bastiaans, "Alternative representation of the linear canonical integral transform," *Opt. Lett.*, vol. 30, pp. 3302–3304, 2005.

[47] M. J. Bastiaans and T. Alieva, "Synthesis of an arbitrary ABCD system with fixed lens positions," *Opt. Lett.*, vol. 31, pp. 2414–2416, 2006.

[48] J. A. Rodrigo, T. Alieva, and M. L. Calvo, "Optical system design for orthosymplectic transformations in phase space," *J. Opt. Soc. Amer. A*, vol. 23, pp. 2494–2500, 2006.

[49] X. Yang, Q. Tan, X. Wei, Y. Xiang, Y. Yan, and G. Jin, "Improved fast fractional-Fourier-transform algorithm," *J. Opt. Soc. Amer. A*, vol. 21, pp. 1677–1681, 2004.

[50] J. Garcia, D. Mas, and R. G. Dorsch, "Fractional Fourier transform calculation through the fast Fourier transform algorithm," *Appl. Opt.*, vol. 35, pp. 7013–7018, 1996.

[51] F. J. Marinho and L. M. Bernardo, "Numerical calculation of fractional Fourier transforms with a single fast Fourier transform algorithm," *J. Opt. Soc. Amer. A*, vol. 15, pp. 2111–2116, 1998.

[52] D. B. Turek, "Design of efficient digital interpolation filters for integer upsampling," M.E. thesis, MIT, Cambridge, MA, 2004.

[53] B. Barshan, M. A. Kutay, and H. M. Ozaktas, "Optimal filtering with linear canonical transformations," *Opt. Commun.*, vol. 135, pp. 32–36, 1997.

Aykut Koç was born in Eskisehir, Turkey, in 1982. He received the B.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2005 and the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 2007. He is currently working towards the Ph.D. degree under the supervision of Prof. L. Hesselink at the Solid State and Photonics Laboratory, Electrical Engineering Department, Stanford University.

As an undergraduate at Bilkent University, he contributed to the development of efficient algorithms for computing families of integrals with applications in diffraction and optical signal processing. His academic interests include Fourier optics, signal processing, the fractional Fourier transform and its applications, and nanooptics and nanostructured metallic surfaces, particularly plasmonic resonance effects.

Haldun M. Ozaktas (M'07) received the B.S. degree from Middle East Technical University, Ankara, Turkey, in 1987 and the Ph.D. degree from Stanford University, Stanford, CA, in 1991.

He joined Bilkent University, Ankara, Turkey, in 1991, where he is presently a Professor of electrical engineering. In 1992, he was at the University of Erlangen-Nurnberg, Bavaria as an Alexander von Humboldt Foundation Postdoctoral Fellow. During summer 1994, he worked as a Consultant at Bell Laboratories, Holmdel, NJ. He is the author of approximately 90 refereed journal articles, over 10 book chapters, and about 100 conference presentations and papers, over 35 of which have been invited. He is also author of the book *The Fractional Fourier Transform* (Wiley, 2001) and edited the book *Three-Dimensional Television* (Springer, 2008). His academic interests include signal and image processing, optical information processing, and optoelectronic and optically interconnected computing systems.

Dr. Ozaktas has a total of over 2500 citations to his work recorded in the Science Citation Index (ISI). He is the recipient of the 1998 ICO International Prize in Optics and one of the youngest recipients ever of the Scientific and Technical Research Council of Turkey (TUBITAK) Science Award (1999), among other awards and prizes. He is also one of the youngest members of the Turkish Academy of Sciences and a Fellow of the Optical Society of America (OSA).

Cagatay Candan received the B.S. degree from Middle East Technical University, Ankara, Turkey, in 1996, the M.S. degree from Bilkent University, Ankara, Turkey, in 1998, and the Ph.D. degree from the Georgia Institute of Technology, Atlanta in 2004.

He is working in Middle East Technical University as an Assistant Professor. His research interests are applications of fractional Fourier transform to time-frequency distributions, detection and estimation theory, statistical signal processing, and applications of statistical signal processing in communication systems and MIMO systems.

M. Alper Kutay was born in Konya, Turkey, in 1972. He received the B.S., M.S., and Ph.D. degrees, all in electrical and electronics engineering, from Bilkent University, Ankara, Turkey, in 1993, 1995, and 1999, respectively.

During his graduate studies, he was a teaching and research assistant. From March 1999 to June 2000, he was with the Communications and Signal Processing Laboratory at Drexel University, Philadelphia, PA. He is now a Senior Research Scientist at the Scientific and Technical Research Council of Turkey-National Research Institute of Electronics and Cryptology (TUBITAK-UEKAE). He is the coauthor of the book *The Fractional Fourier Transform with Applications in Optics and Signal Processing* (Wiley, 2001). His current research areas include time-frequency signal representations, fractional Fourier transform, time-varying filtering, signal detection, signal enhancement, and radar signal processing.

Dr. Kutay was awarded third place in the 1994 IEEE Region 8 student paper contest for his work titled "An Adaptive Speckle Suppression Filter for Medical Ultrasonic Imaging."