

Chapter 10

Fast Algorithms for Digital Computation of Linear Canonical Transforms

Aykut Koç, Figen S. Oktem, Haldun M. Ozaktas, and M. Alper Kutay

Abstract Fast and accurate algorithms for digital computation of linear canonical transforms (LCTs) are discussed. Direct numerical integration takes $O(N^2)$ time, where N is the number of samples. Designing fast and accurate algorithms that take $O(N \log N)$ time is of importance for practical utilization of LCTs. There are several approaches to designing fast algorithms. One approach is to decompose an arbitrary LCT into blocks, all of which have fast implementations, thus obtaining an overall fast algorithm. Another approach is to define a discrete LCT (DLCT), based on which a fast LCT (FLCT) is derived to efficiently compute LCTs. This strategy is similar to that employed for the Fourier transform, where one defines the discrete Fourier transform (DFT), which is then computed with the fast Fourier transform (FFT). A third, hybrid approach involves a DLCT but employs a decomposition-based method to compute it. Algorithms for two-dimensional and complex parametered LCTs are also discussed.

10.1 Introduction

Linear canonical transforms (LCTs) are commonly referred to as quadratic-phase integrals or quadratic-phase systems in optics [1]. They have also been referred to

A. Koç, (✉)

Intelligent Data Analytics Research Program Department, ASELSAN Research Center, ASELSAN Inc., Ankara, Turkey
e-mail: aykutkoc@aselsan.com.tr

F.S. Oktem

Department of Electrical and Electronics Engineering, Middle East Technical University, 06800 Çankaya, Ankara, Turkey
e-mail: figeno@metu.edu.tr

H.M. Ozaktas

Department of Electrical Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey
e-mail: haldun@ee.bilkent.edu.tr

M.A. Kutay

The Scientific and Technological Research Council of Turkey, 06100 Kavaklıdere, Ankara, Turkey
e-mail: alper.kutay@tubitak.gov.tr

by names such as generalized Huygens integrals [2], generalized Fresnel transforms [3, 4], special affine Fourier transforms [5, 6], extended fractional Fourier transforms (FRTs) [7], and Moshinsky–Quesne transforms [8], among other names. The so-called $ABCD$ systems widely used in optics [9] are also represented by LCTs.

One-dimensional (1D) LCTs [8, 10] constitute a three-parameter class of linear integral transforms [1, 11, 12] which include among its special cases, the one-parameter subclasses of FRTs, scaling operations, and chirp multiplication (CM) and chirp convolution (CC) operations, the latter also known as Fresnel transforms.

LCTs appear widely in optics [2, 10, 11], electromagnetics, classical and quantum mechanics [8, 13, 14], as well as in computational and applied mathematics [15]. The application areas of LCTs include, among others, the study of scattering from periodic potentials [16–18], laser cavities [2, 19, 20], and multilayered structures in optics and electromagnetics [21]. They can also be used for fast and efficient realization of filtering in LCT domains [22].

Generalizations to two-dimensional (2D) transforms and complex-parametered transforms are also present in the literature. Classification of first-order optical systems and their representation through LCTs are studied in [23–27] for 1D and 2D cases, respectively. Bilateral Laplace transforms, Bargmann transforms, Gauss–Weierstrass transforms [8, 28, 29], fractional Laplace transforms [30, 31], and complex-ordered fractional Fourier transformations (CFRTs) [32–35] are all special cases of complex linear canonical transforms (CLCTs).

The LCTs 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 [1, 5, 6, 10, 12].

Given its ubiquitous nature and numerous applications, the discretization, sampling, and efficient digital computation of LCTs are of considerable interest.

The 1D LCT of $f(u)$ with parameter matrix \mathbf{M} is denoted as $f_{\mathbf{M}}(u) = (\mathcal{C}_{\mathbf{M}}f)(u)$:

$$(\mathcal{C}_{\mathbf{M}}f)(u) = \sqrt{\beta} e^{-i\pi/4} \int_{-\infty}^{\infty} \exp[i\pi(\alpha u^2 - 2\beta uu' + \gamma u'^2)] f(u') du', \quad (10.1)$$

where α, β, 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, γ which uniquely define the LCT:

$$\mathbf{M} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \gamma/\beta & 1/\beta \\ -\beta + \alpha\gamma/\beta & \alpha/\beta \end{bmatrix} = \begin{bmatrix} \alpha/\beta & -1/\beta \\ \beta - \alpha\gamma/\beta & \gamma/\beta \end{bmatrix}^{-1}. \quad (10.2)$$

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 [8, 10].

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 the product of the matrices of the cascaded LCTs. For instance, if two LCTs with matrices \mathbf{M}_1 and \mathbf{M}_2 operate successively, then the equivalent transform is an LCT with matrix $\mathbf{M}_3 = \mathbf{M}_2\mathbf{M}_1$. LCTs are not commutative. The matrix of the inverse of an LCT is simply the inverse of the matrix of the original LCT [8, 10].

There has been considerable work on defining discrete/finite FRTs and, to a lesser degree, discrete/finite LCTs [36–55]. Definitions of the discrete FRT (DFRT) [45, 53, 55] are more established and recognized than definitions of the discrete LCT (DLCT). In this chapter the primary emphasis is not on sampling and the definition of discrete transforms. We concentrate on fast and accurate algorithms for digitally computing continuous LCTs, with careful attention to sampling issues, so as to produce results that are nearly as accurate and fast as is theoretically possible. Some approaches do involve the definition of a discrete transform, others do not.

Historically, computation of the Fresnel diffraction integral, which is a special case of LCTs, has received the greatest attention since it describes the propagation of light in free space (see [56, 57] and the references therein). Since the Fresnel integral is space-invariant and takes the form of a convolution, it can be computed in $O(N \log N)$ time. It is important to note that despite the fact that general LCTs are not space-invariant (not in convolution form), so that a standard Fourier domain approach cannot be used to obtain an $O(N \log N)$ -time algorithm, the algorithms presented in this chapter are $O(N \log N)$ -time algorithms.

The Fourier transform (FT) is the most prominent special case of LCTs. Most often, the continuous FT is approximated by the discrete Fourier transform (DFT) and the DFT is computed with the fast Fourier transform (FFT) algorithm [58] in $O(N \log N)$ time. The FRT, another important special case of LCTs, is a generalization of the FT. A fast algorithm for digital computation of the continuous FRT was first developed in [59]. This fast FRT algorithm paved the way for the development of fast algorithms for more general transforms, leading to the algorithms for arbitrary LCTs that are discussed in Sects. 10.4, 10.7, and 10.8. The algorithm in [59] serves as a basic building block within these fast linear canonical transform (FLCT) algorithms.

In the next section, we present some preliminary material. In Sect. 10.3, we discuss fast computation of the FRT. In Sect. 10.4, we turn our attention to decomposition-based approaches to LCT computation. Next, in Sect. 10.5, we present DLCT based methods. This is followed by Sect. 10.6, where we discuss hybrid algorithms that involve the DLCT but employ a decomposition-based method to compute it. Finally, we discuss extensions to two-dimensional (2D) and complex transforms.

10.2 Preliminaries

We begin by reviewing the concepts of compactness and the relationship of LCTs to the Wigner distribution.

10.2.1 Compactness in Space, Frequency, and Phase Space

A function will be referred to as compact if its support is so. The support of a function is the subset of the real axis in which the function is not equal to zero. In other words, a function is compact if and only if its nonzero values are confined to a finite interval. It is well known that a function and its Fourier transform cannot both be compact (unless they are identically zero). In practice however, it seems that we are always working with a finite space interval and a finite bandwidth. This discrepancy between our mathematical idealizations and the real world is usually not a problem when we work with signals of large space-bandwidth product. The space-bandwidth product can be crudely defined as the product of the spatial extent of the signal and its (double-sided) bandwidth. It is equal to the number of degrees of freedom, the number of complex numbers required to uniquely characterize the signal among others of the same space-bandwidth product.

We will assume that the space-domain representation of our signal is approximately confined to the interval $[-\Delta x/2, \Delta x/2]$ and that its frequency-domain representation is confined to the interval $[-\Delta \sigma/2, \Delta \sigma/2]$. With this statement we mean that a sufficiently large percentage of the signal energy is confined to these intervals. For a given class of functions, this can be ensured by choosing Δx and $\Delta \sigma$ sufficiently large. We then define the space-bandwidth product $N \equiv \Delta x \Delta \sigma$, which is always greater than unity, because of the uncertainty relation.

Let us now introduce the scaling parameter s with the dimension of space and introduce scaled coordinates $u = x/s$ and $\mu = \sigma s$. With these new coordinates, the space and frequency domain representations will be confined to intervals of length $\Delta x/s$ and $\Delta \sigma s$. Let us choose $s = \sqrt{\Delta x/\Delta \sigma}$ so that the lengths of both intervals are now equal to the dimensionless quantity $\sqrt{\Delta \sigma \Delta x}$ which we will denote by Δu . In the newly defined coordinates, our signal can be represented in both domains with $N = \Delta u^2$ samples spaced $\Delta u^{-1} = 1/\sqrt{N}$ apart.

From now on we will assume that this dimensional normalization has been performed and that the coordinates appearing in the definition of the FRT, Wigner distribution, etc. are all dimensionless quantities.

For a signal with rectangular space-frequency support, the space-bandwidth product is equal to the number of degrees of freedom. This is not true for signals with other support shapes. While LCTs do not change the number of degrees of freedom of a signal, they may change its space-bandwidth product.

10.2.2 Relationship of LCTs to the Wigner Distribution

The relationship between first-order optical systems (quadratic-phase systems or LCTs) and the Wigner distribution has been extensively studied [1, 11, 12, 60, 61]. The Wigner distribution $W_f(u, \mu)$ of a signal $f(u)$ can be defined as [62, 63]

$$W_f(u, \mu) = \int_{-\infty}^{\infty} f(u + u'/2) f^*(u - u'/2) e^{-2\pi i \mu u'} du'. \quad (10.3)$$

Roughly speaking, $W_f(u, \mu)$ is a function which gives the distribution of signal energy over space and frequency. Its infinite integral over space and frequency, expressed as $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_f(u, \mu) du d\mu$, 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 [10]

$$W_{f_{\mathbf{M}}}(u, \mu) = W_f(Du - B\mu, -Cu + A\mu). \quad (10.4)$$

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; that is, they do not carry energy in or out of the defined support area, keeping the total energy within the support area constant. 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.

10.3 Fast Computation of Fractional Fourier Transforms

Here we review the fast algorithm for computing the continuous FRT presented in [59], both for historical reasons and since it constitutes an important building block of the LCT algorithms we will later present in Sects. 10.4 and 10.7.

Let $\{\mathcal{F}f\}(u)$ denote the Fourier transform of $f(u)$. Integral powers \mathcal{F}^j of the operator $\mathcal{F} \equiv \mathcal{F}^1$ may be defined as its successive applications. Then we have $\{\mathcal{F}^2f\}(u) = f(-u)$ and $\{\mathcal{F}^4f\}(u) = f(u)$. The a th order FRT $\{\mathcal{F}^a f\}(u)$ of the function $f(u)$ may be defined for $0 < |a| < 2$ as

$$\begin{aligned} \mathcal{F}^a[f(u)] &\equiv \{\mathcal{F}^a f\}(u) \equiv \int_{-\infty}^{\infty} K_a(u, u')f(u') du', \\ K_a(u, u') &\equiv A_\phi \exp [i\pi(u^2 \cot \phi - 2uu' \csc \phi + u'^2 \cot \phi)], \\ A_\phi &\equiv \frac{\exp(-i\pi \operatorname{sgn}(\sin \phi)/4 + i\phi/2)}{|\sin \phi|^{1/2}}, \end{aligned} \tag{10.5}$$

where $\phi \equiv a\pi/2$ and i is the imaginary unit. The kernel approaches $K_0(u, u') \equiv \delta(u - u')$ and $K_{\pm 2}(u, u') \equiv \delta(u + u')$ for $a = 0$ and $a = \pm 2$, respectively. The definition is easily extended outside the interval $[-2, 2]$ by remembering that \mathcal{F}^{4j} is the identity operator for any integer j and that the FRT operator is additive in index, that is, $\mathcal{F}^{a_1} \mathcal{F}^{a_2} = \mathcal{F}^{a_1+a_2}$.

The FRT, like all LCTs, can be broken down into a succession of simpler operations, such as chirp multiplication, chirp convolution, scaling, and ordinary Fourier transformation. Here we will concentrate on two particular decompositions which lead to two distinct algorithms. By ensuring that the sampling interval satisfies the Nyquist criterion at each stage of the decomposition, it becomes possible to use the output samples to reconstruct good approximations of the continuous FRT.

First, we consider decomposing the FRT into a chirp multiplication followed by a chirp convolution followed by another chirp multiplication [59]. We assume $a \in [-1, 1]$. Manipulating Eq. (10.5), we can write

$$f_a(u) = \exp[-i\pi u^2 \tan(\phi/2)]g'(u), \tag{10.6}$$

$$g'(u) = A_\phi \int_{-\infty}^{\infty} \exp[i\pi\beta(u - u')^2]g(u') du', \tag{10.7}$$

$$g(u) = \exp[-i\pi u^2 \tan(\phi/2)]f(u), \tag{10.8}$$

where $g(u)$ and $g'(u)$ represent intermediate results and $\beta = \csc \phi$. In the first step [Eq. (10.8)] we multiply the function $f(u)$ by a chirp function. As shown in [59], the bandwidth and space-bandwidth product of $g(u)$ can be as large as twice that of $f(u)$. Thus, we require samples of $g(u)$ at intervals of $1/2\Delta u$. If the samples of $f(u)$ spaced at $1/\Delta u$ are given to begin with, we can interpolate these by a factor of two and then multiply by the samples of the chirp function to obtain the desired samples of $g(u)$. The next step is to convolve $g(u)$ with a chirp function, as given in Eq. (10.7). To perform this convolution, we note that since $g(u)$ is bandlimited, the chirp function can also be replaced with its bandlimited version without any effect. That is,

$$g'(u) = A_\phi \int_{-\infty}^{\infty} \exp[i\pi\beta(u - u')^2]g(u') du' = A_\phi \int_{-\infty}^{\infty} h(u - u')g(u') du', \tag{10.9}$$

where

$$h(u) = \int_{-\Delta u}^{\Delta u} H(\mu) \exp(i2\pi\mu u) d\mu, \tag{10.10}$$

where

$$H(\mu) = \frac{1}{\sqrt{\beta}} e^{i\pi/4} \exp(-i\pi\mu^2/\beta), \quad (10.11)$$

is the Fourier transform of $\exp[i\pi\beta u^2]$. It is possible to express $h(u)$ explicitly in terms of the Fresnel integral defined as $F(z) = \int_0^z \exp(\pi z^2/2) dz$. Now, Eq. (10.7) can be sampled, giving

$$g' \left(\frac{m}{2\Delta u} \right) = \sum_{n=-N}^N h \left(\frac{m-n}{2\Delta u} \right) g \left(\frac{n}{2\Delta u} \right). \quad (10.12)$$

This convolution can be evaluated using a FFT. Then, after performing the last step [Eq. (10.6)], we obtain the samples of $f_a(u)$ spaced at $1/2\Delta u$. Since we assumed that all transforms of $f(u)$ are bandlimited to the interval $[-\Delta u/2, \Delta u/2]$, we finally decimate these samples by a factor of 2 to obtain samples of $f_a(u)$ spaced at $1/\Delta u$. Then the continuous function $f_a(u)$ can be reconstructed from these samples.

The second method does not require Fresnel integrals [59]. Equation (10.5) can be alternatively put in the form:

$$\{\mathcal{F}^a f\}(u) = A_\phi e^{i\pi\alpha u^2} \int_{-\infty}^{\infty} e^{-i2\pi\beta uu'} \left[e^{i\pi\alpha u'^2} f(u') \right] du', \quad (10.13)$$

where $\alpha = \cot \phi$ and $\beta = \csc \phi$. We again assume that the Wigner distribution of $f(\cdot)$ is zero outside a circle of diameter Δu centered around the origin. Under this assumption, and by limiting the order a to the interval $0.5 \leq |a| \leq 1.5$, the amount of vertical shear in Wigner space resulting from the chirp modulation is bounded by $\Delta u/2$. Then the modulated function $e^{i\pi\alpha u'^2} f(u')$ is band-limited to Δu in the frequency domain. Thus $e^{i\pi\alpha u'^2} f(u')$ can be represented by Shannon's interpolation formula:

$$e^{i\pi\alpha u'^2} f(u') = \sum_{n=-N}^N e^{i\pi\alpha \left(\frac{n}{2\Delta u}\right)^2} f \left(\frac{n}{2\Delta u} \right) \operatorname{sinc} \left(2\Delta u \left(u' - \frac{n}{2\Delta u} \right) \right), \quad (10.14)$$

where $N = (\Delta u)^2$. The summation goes from $-N$ to N since $f(u')$ is assumed to be zero outside $[-\Delta u/2, \Delta u/2]$. By using Eqs. (10.14) and (10.13), and changing the order of integration and summation we obtain

$$\begin{aligned} \{\mathcal{F}^a f\}(u) &= A_\phi e^{i\pi\alpha u^2} \sum_{n=-N}^N e^{i\pi\alpha \left(\frac{n}{2\Delta u}\right)^2} f \left(\frac{n}{2\Delta u} \right) \\ &\quad \times \int_{-\infty}^{\infty} e^{-i2\pi\beta uu'} \operatorname{sinc} \left(2\Delta u \left(u' - \frac{n}{2\Delta u} \right) \right) du'. \end{aligned} \quad (10.15)$$

The integral is equal to $e^{-i2\pi\beta u \frac{n}{2\Delta u}} (1/2\Delta u) \text{rect}(\beta u/2\Delta u)$. For the range of $0.5 \leq |a| \leq 1.5$, $\text{rect}(\beta u/2\Delta u)$ will always be equal to unity on the support $|u| \leq \Delta u/2$ of the transformed function. Hence we can write

$$\{\mathcal{F}^a f\}(u) = \frac{A_\phi}{2\Delta u} \sum_{n=-N}^N e^{i\pi\alpha u^2} e^{-i2\pi\beta u \frac{n}{2\Delta u}} e^{i\pi\alpha(\frac{n}{2\Delta u})^2} f\left(\frac{n}{2\Delta u}\right). \quad (10.16)$$

Then, the samples of the transformed function are obtained as

$$\{\mathcal{F}^a f\}\left(\frac{m}{2\Delta u}\right) = \frac{A_\phi}{2\Delta u} \sum_{n=-N}^N e^{i\pi\left(\alpha(\frac{m}{2\Delta u})^2 - 2\beta\frac{mn}{(2\Delta u)^2} + \alpha(\frac{n}{2\Delta u})^2\right)} f\left(\frac{n}{2\Delta u}\right) \quad (10.17)$$

which is a finite summation allowing us to obtain the samples of the fractional transform in terms of the samples of the original function. Direct computation of this form would require $O(N^2)$ multiplications. An $O(N \log N)$ algorithm can be obtained as follows. We put Eq. (10.17) into the following form after some algebraic manipulations:

$$\{\mathcal{F}^a f\}\left(\frac{m}{2\Delta u}\right) = \frac{A_\phi}{2\Delta u} e^{i\pi(\alpha-\beta)(\frac{m}{2\Delta u})^2} \sum_{n=-N}^N e^{i\pi\beta(\frac{m-n}{2\Delta u})^2} e^{i\pi(\alpha-\beta)(\frac{n}{2\Delta u})^2} f\left(\frac{n}{2\Delta u}\right). \quad (10.18)$$

It can be recognized that the summation is the convolution of $e^{i\pi\beta(n/2\Delta u)^2}$ and the chirp modulated function $f(\cdot)$. The convolution can be computed in $O(N \log N)$ time by using the FFT. The output samples are then obtained by a final chirp modulation. Hence the overall complexity is $O(N \log N)$.

10.4 Decomposition-Based LCT Algorithms

This section discusses decomposition-based approaches to fast and accurate digital computation of continuous LCTs. These approaches begin with samples of the continuous input signal and compute samples of the continuous LCT output signal such that the continuous output can be interpolated from the computed output samples. This is accomplished by decomposing the LCT operation into basic building blocks that already have fast algorithms. The main approach we discuss is based on the following decomposition involving the FRT, scaling, and chirp multiplication [64, 65]:

$$\mathbf{M} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -q & 1 \end{bmatrix} \begin{bmatrix} M & 0 \\ 0 & 1/M \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}. \quad (10.19)$$

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

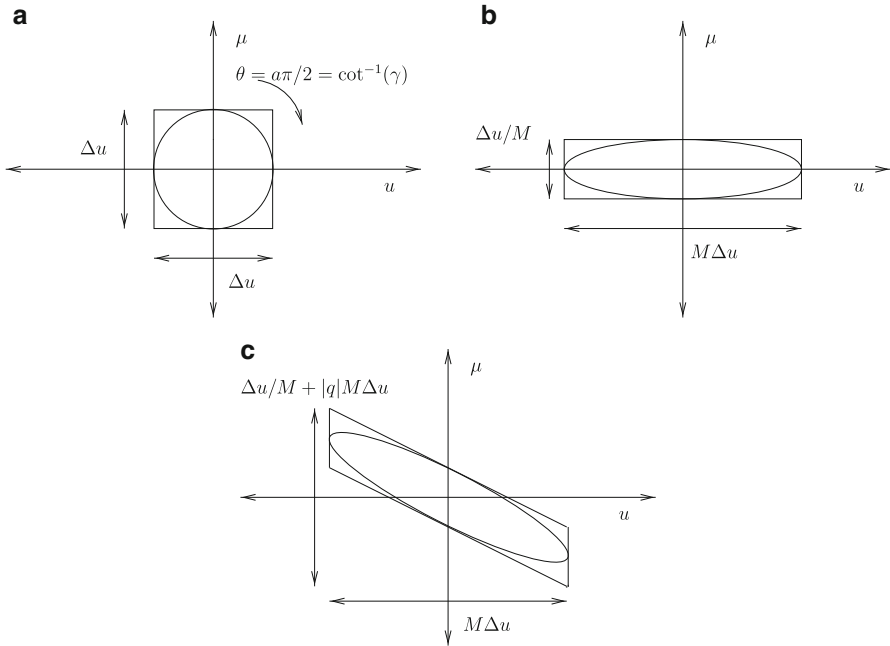


Fig. 10.1 Sequence of geometrical distortions for the decomposition in Eq. (10.19) [65]. (a) After the first stage: FRT; (b) after the second stage: scaling; (c) after the third stage: CM

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 [59] and reviewed in the previous section can be used for fast computation of the FRT, this decomposition directly leads to a fast algorithm for arbitrary LCTs. This decomposition was inspired by the optical interpretation in [66] and is also a special case of the widely known Iwasawa decomposition [26, 67, 68]. It was also proposed later in [64, 69]. Figure 10.1 illustrates the sequence of geometrical distortions in phase space corresponding to this decomposition, which is rotation, scaling, and shearing, respectively. The initial space-frequency support is a circle of diameter Δu .

To obtain the decomposition parameters in terms of the LCT parameters, we multiply out the right-hand side of Eq. (10.19) and replace the matrix entries A, B, C, D with α, β, γ , we obtain:

$$\begin{bmatrix} \gamma/\beta & 1/\beta \\ -\beta + \alpha\gamma/\beta & \alpha/\beta \end{bmatrix} = \begin{bmatrix} M \cos \theta & M \sin \theta \\ -qM \cos \theta - \sin \theta/M & -qM \sin \theta + \cos \theta/M \end{bmatrix}, \tag{10.20}$$

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

$$a = (2/\pi)\cot^{-1}\gamma, \quad (10.21)$$

$$M = \begin{cases} \sqrt{1 + \gamma^2}/\beta, & \gamma \geq 0, \\ -\sqrt{1 + \gamma^2}/\beta, & \gamma < 0, \end{cases} \quad (10.22)$$

$$q = \gamma\beta^2/(1 + \gamma^2) - \alpha. \quad (10.23)$$

The ranges of the square root and the \cot^{-1} both lie in $(-\pi/2, \pi/2]$. In operator notation this algorithm can be expressed as

$$\mathcal{C}_M = \mathcal{Q}_q J_k \mathcal{M}_M \mathcal{F}_{lc}^a. \quad (10.24)$$

In this method, the first operation is an FRT, whose fast computation in $O(N \log N)$ time is presented in [59, 70]. Other works dealing with fast computation of the FRT include [71, 72]. The first algorithm presented in [59] and reviewed above 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. Care was taken to ensure that the output samples uniquely represented the continuous FRT in the Nyquist–Shannon sense. The presently discussed LCT algorithm employs that algorithm 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 is 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 an 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 $O(N)$ time, leading to an overall complexity of $O(N \log N)$. As in the first method, it is again necessary to ensure that the final output samples are sufficient to represent the transformed signal in the Nyquist–Shannon sense. Since LCTs distort the original space-frequency support, both the space and frequency extent of the signal, as well as its space-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 in the Nyquist–Shannon sense (unless we use some specialized basis to represent the signals) [64].

Delaying confrontation with the necessity to deal with this greater number of samples until the very last step is a significant advantage of the present 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 before chirp multiplication at least by a factor k corresponding to the increase in space-bandwidth product [65]:

$$k \geq 1 + |\gamma - \alpha(1 + \gamma^2)/\beta^2|. \quad (10.25)$$

For convenience we choose k to be the smallest integer satisfying this inequality.

We have considered several examples to illustrate and compare the presented methods. We consider the chirped pulse function $\exp(-\pi u^2 - i\pi u^2)$, denoted F1, and the trapezoidal function $1.5\text{tri}(u/3) - 0.5\text{tri}(u)$, denoted F2 ($\text{tri}(u) = \text{rect}(u) * \text{rect}(u)$). 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. 10.2 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 space 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 by the presented fast method (referred to as A2), another fast method which will be presented shortly (referred to as A1), and by a highly inefficient brute force numerical approach based on composite Simpson's rule, which is here taken as a reference.

The results are tabulated in Table 10.1 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 is defined as the energy of the difference normalized by the energy of the reference, expressed as a percentage.) Results are shown for two algorithms denoted as A1 and A2, both of which appear in [65] as Method I and Method II, respectively. The algorithm outlined above is A2. The algorithm A1 will be summarized below after we discuss the results.

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 space and frequency extents (or assumed space-frequency region). For those cases in which the error is large, such as F3, this means that we have determined the space-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 one could hope for to begin with.

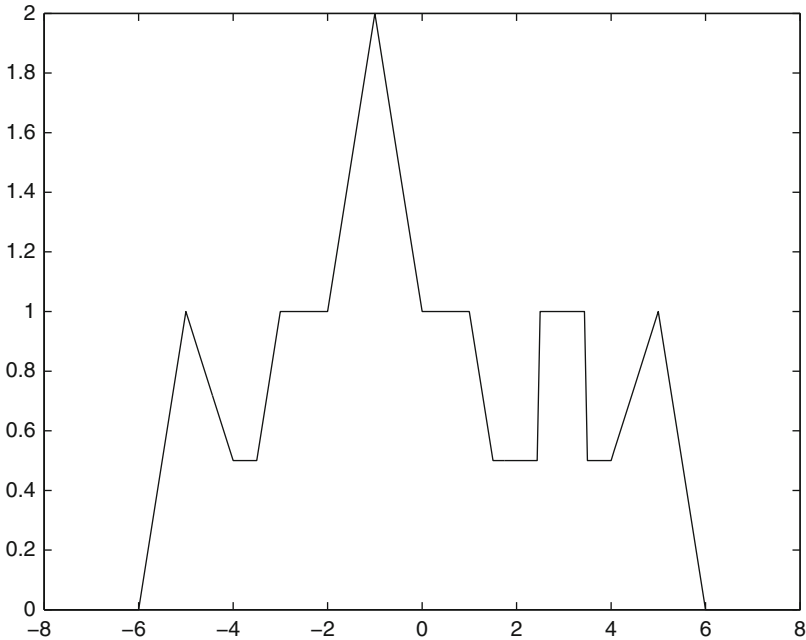


Fig. 10.2 Example function F4 [65]

Table 10.1 Percentage errors for different functions F, transforms T, and algorithms A [65]

	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}

Figure 10.3 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 space-bandwidth product of the signals (here 64). This demonstrates that the number of samples N can be chosen comparable to the space-bandwidth product, which is the smallest number we can expect to work with, and need not be chosen larger. Algorithm A2 was used to obtain this plot for illustration purposes but similar results can also be obtained when we use Algorithm A1.

We now briefly summarize Algorithm A1, appearing in [65], where the use of matrix factorizations to decompose LCTs into cascade combinations of elementary LCT blocks has been studied exhaustively. Since each stage in such a decomposition

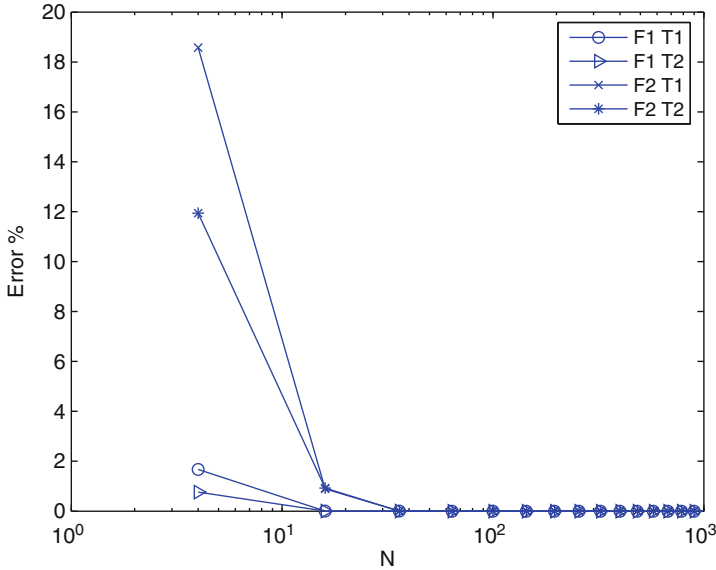


Fig. 10.3 Percentage errors versus N for selected functions and transforms [65]

can be computed in at most $O(N \log N)$ time, the overall LCT can also be. Numerous such decompositions are possible [10, 73], 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 Eq. (10.1), will in general lead to very high sampling rates if conventional Shannon–Nyquist sampling is employed. 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). All possible decompositions with three, four, and five cascade blocks have been considered and every permutation has been checked to see if that decomposition is capable of expressing an LCT with arbitrary parameters. The resulting algorithm can be summarized as follows:

- If $|\gamma| \leq 1$, use the decomposition:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma/\beta^2 & 1 \end{bmatrix} \begin{bmatrix} \beta & 0 \\ 0 & 1/\beta \end{bmatrix},$$

$$\mathcal{C}_{\mathbf{M}} = \mathcal{Q}_{-\alpha} J_{k/2} \mathcal{F}_{1c} \mathcal{Q}_{-\gamma/\beta^2} J_2 \mathcal{M}_{\beta}, \tag{10.26}$$

where J_z represents the $\times z$ oversampling operation. The minimum value of k is $k \geq 1 + |\gamma| + |\alpha|(1 + |\gamma|)^2/\beta^2$.

- If $|\gamma| > 1$, use the decomposition:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ \alpha - \beta^2/\gamma & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\gamma/\beta^2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} -\gamma/\beta & 0 \\ 0 & -\beta/\gamma \end{bmatrix},$$

$$\mathcal{C}_{\mathbf{M}} = \mathcal{Q}_{-\alpha+\beta^2/\gamma} J_{k/2} \mathcal{F}_{1c} \mathcal{Q}_{\gamma/\beta^2} J_2 \mathcal{F}_{1c} \mathcal{M}_{-\gamma/\beta}. \tag{10.27}$$

The minimum value of k is $k \geq 1 + 1/|\gamma| + (1 + |\gamma|)^2|\alpha - \beta^2/\gamma|/\beta^2$.

The above algorithm, A1, has been compared with the earlier presented FRT-based algorithm A2, both in terms of computational complexity and accuracy, but no significant difference has been found [65].

Another work dealing with decomposition-based methods to digitally compute the fractional Fourier, Fresnel, and general LCTs is [69]. This work is of significance because of two reasons. First, it reviews previous fast algorithms presented in the literature for computing the several important special cases of LCTs such as FRT and Fresnel transformation (FST), and distills them into a single decomposition-based approach covering the general case. Second, it emphasizes the importance of tracking the space-bandwidth product through successive stages of the decomposition as also stressed in [65]. It also sets forth a systematic, elegant, uniform, and general approach to determining the overall increase in space-bandwidth product of the final transformed signal and hence the number of samples needed for Nyquist–Shannon interpolation. It is assumed that the input signal energy is initially contained within some arbitrary four-sided shape in phase space that is defined by the coordinates of the four corners, $u_1, \mu_1, u_2, \mu_2, u_3, \mu_3, u_4, \mu_4$. The spatial and frequency extents of the signal are denoted by W_0 and B_0 , respectively. The resulting number of regularly placed samples required to represent the signal in the Nyquist–Shannon sense is given by $N_0 = W_0B_0$. Given the four corners of the initial support as

$$\mathbf{S} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \\ \mu_1 & \mu_2 & \mu_3 & \mu_4 \end{bmatrix}, \tag{10.28}$$

the new coordinates of the four corners of the region after the LCT is given by

$$\mathbf{S}' = \begin{bmatrix} Au_1 + B\mu_1 & Au_2 + B\mu_2 & Au_3 + B\mu_3 & Au_4 + B\mu_4 \\ Cu_1 + D\mu_1 & Cu_2 + D\mu_2 & Cu_3 + D\mu_3 & Cu_4 + D\mu_4 \end{bmatrix}, \tag{10.29}$$

where A, B, C, D are the LCT parameters.

Then, the spatial and frequency extents are the maximum distance between any two of the u and any two of the μ coordinates, respectively. To represent this, the $\text{Max}(\dots)$ notation is defined to denote the maximum element on each row of the matrix it operates on Hennelly and Sheridan [69]. Finally, the resultant spatial, W_{LCT} , and frequency, B_{LCT} , extents after the LCT are given by

$$\begin{aligned}
\begin{bmatrix} W_{\text{LCT}} \\ B_{\text{LCT}} \end{bmatrix} &= \text{Max} \begin{bmatrix} |A(u_1 - u_2) + B(\mu_1 - \mu_2)| & |A(u_1 - u_3) + B(\mu_1 - \mu_3)| \\ |C(u_1 - u_2) + D(\mu_1 - \mu_2)| & |C(u_1 - u_3) + D(\mu_1 - \mu_3)| \\ |A(u_1 - u_4) + B(\mu_1 - \mu_4)| & |A(u_2 - u_3) + B(\mu_2 - \mu_3)| \\ |C(u_1 - u_4) + D(\mu_1 - \mu_4)| & |C(u_2 - u_3) + D(\mu_2 - \mu_3)| \\ |A(u_2 - u_4) + B(\mu_2 - \mu_4)| & |A(u_3 - u_4) + B(\mu_3 - \mu_4)| \\ |C(u_2 - u_4) + D(\mu_2 - \mu_4)| & |C(u_3 - u_4) + D(\mu_3 - \mu_4)| \end{bmatrix}. \\
\end{aligned} \tag{10.30}$$

By using the above procedure, the space-bandwidth of the signal can be tracked through the intermediate stages of the LCT decomposition and the number of samples required to represent the signal in the Nyquist–Shannon sense can be determined.

The Fresnel transform has received a lot of attention, since it models free-space propagation of waves under the Fresnel approximation. Hennelly and Sheridan [69] also studies the problem of fast computation of the Fresnel transform and reviews restrictions of several earlier approaches. Based on these observations, two decompositions are proposed that work for arbitrary LCTs. One of them is based on the Iwasawa decomposition that we have presented above [see Eq. (10.19)]. The second is given by the following matrix equation:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ C/A & 1 \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & 1/A \end{bmatrix} \begin{bmatrix} 1 & B/A \\ 0 & 1 \end{bmatrix}. \tag{10.31}$$

This is a chirp convolution, scaling, chirp multiplication decomposition. Although it can indeed realize arbitrary LCTs, it uses both a chirp convolution and a chirp multiplication that requires an unnecessary increase in the sampling rate. This is partly avoidable by using more suitable decompositions [65].

10.5 DLCT Based Algorithms

In the previous section, we focused on algorithms for fast computation of continuous LCTs that are based on decomposition of the LCT operation. In this section, we discuss the approach involving the definition of a DLCT, but in which computation is not based on decomposition into basic operations as in the previous section. In this approach, fast digital algorithms for LCTs, which are often referred to as FLCT algorithms, are derived by first defining a DLCT, which is to the continuous LCT what the DFT is to the continuous Fourier transform, and then developing a fast algorithm to compute the DLCT similar to the FFT algorithm.

The key points of the derivation of [74] are given here. We first review the derivation of the DLCT, which was first given in [75], and discuss the shifting properties of this DLCT, which is the key component in the development of the FLCT [74]. We start with the sampled version of a continuous signal $f(u)$ as

$$f^{\delta u}(u) = f(u)\delta_{\delta u}(u) = \sum_{n=-\infty}^{\infty} f(n\delta u)\delta(u - n\delta u) \quad (10.32)$$

or in Fourier series form

$$f^{\delta u}(u) = \frac{1}{\delta u} f(u) \sum_{n=-\infty}^{\infty} \exp(i2\pi kf_s u), \quad (10.33)$$

where $\delta(u)$ is Dirac's impulse, δu is the sampling interval, and f_s is the sampling frequency. Let us recall the definition of the LCT with parameters α , β , γ , of a function $f(u)$ as

$$(\mathcal{C}_M f)(u) = \int_{-\infty}^{\infty} A \exp[i\pi(\alpha u^2 - 2\beta uu' + \gamma u'^2)] f(u') du'. \quad (10.34)$$

\mathcal{C}_M is the LCT operator. We note that in [74] and later in [76], $\alpha \rightarrow -\gamma$, $\gamma \rightarrow -\alpha$, $\beta \rightarrow -\beta$, since the authors work with the inverse of the LCT matrix as defined here. We omit the complex constant A in what follows. The key properties to be utilized are the shifting theorems [74]:

$$\begin{aligned} \mathcal{C}_M[\exp(i2\pi \xi u') f(u')](u) &= \exp(-i\pi \alpha \xi^2 / \beta^2) \exp(i2\pi u \xi \alpha / \beta) \\ &\quad \times \mathcal{C}_M[f(u')](u - \xi / \beta), \end{aligned} \quad (10.35)$$

$$\begin{aligned} \mathcal{C}_M[f(u' - \xi)](u) &= \exp[i\pi \xi^2 (\gamma - \alpha \gamma^2 / \beta^2)] \exp[i2\pi u \xi (\alpha \gamma / \beta - \beta)] \\ &\quad \times \mathcal{C}_M[f(u')](u - \xi \gamma / \beta). \end{aligned} \quad (10.36)$$

When the LCT operator operates on the sampled signal in Eq. (10.32) we get:

$$\begin{aligned} \mathcal{C}_M[f^{\delta u}(u')](u) &= \int_{-\infty}^{\infty} \left[\sum_{n=-\infty}^{\infty} f(n\delta u) \delta(u' - n\delta u) \right] \exp(-i2\pi \beta uu') \\ &\quad \times \exp[-i\pi(\gamma u'^2 + \alpha u^2)] du' \\ &= \exp(i\pi \alpha u^2) \sum_{n=-\infty}^{\infty} f(n\delta u) \exp[i\pi \gamma (n\delta u)^2] \exp(-i2\pi \beta u n \delta u). \end{aligned} \quad (10.37)$$

Equation (10.37) is the discrete time (or space) LCT (DTLCT) which is analogous to the discrete time (or space) Fourier transform (DTFT). We can also apply the LCT operator to the alternative expression of a sampled function as given in Eq. (10.33), to obtain:

$$\begin{aligned} \mathcal{C}_{\mathbf{M}}[f^{\delta u}(u')](u) &= \frac{1}{\delta u} \sum_{n=-\infty}^{\infty} \exp[-i\pi(k\delta u)^2\alpha/\beta^2] \\ &\quad \times \exp[i2\pi(k/\delta u)\alpha u/\beta] \mathcal{C}_{\mathbf{M}}[f(u')](u - k/\delta u\beta). \end{aligned} \quad (10.38)$$

The above expression is important since it reveals the periodicity of the LCT of the sampled signal, which is a crucial property in deriving the FLCT. The magnitude of $\mathcal{C}_{\mathbf{M}}[f^{\delta u}(u')](u)$ is equal to the magnitude of $\mathcal{C}_{\mathbf{M}}[f(u')](u)$ repeated periodically with the period $1/\delta u\beta$ and the phase of $\mathcal{C}_{\mathbf{M}}[f^{\delta u}(u')](u)$ is also equal to that of $\mathcal{C}_{\mathbf{M}}[f(u')](u)$ repeated periodically with the same period $1/\delta u\beta$.

Denoting the DTLCT operator by \mathcal{DTLCT} , analysis yields the following shift theorem for the DTLCT [74]:

$$\begin{aligned} \mathcal{DTLCT}_{\mathbf{M}}[f((n-l)\delta u)](u) \\ &= \exp[i\pi l^2\delta u^2(\gamma - \alpha\gamma^2/\beta^2)] \\ &\quad \times \exp[i2\pi ul\delta u(\alpha\gamma/\beta - \beta)] \mathcal{DTLCT}_{\mathbf{M}}[f(n\delta u)](u - l\delta u\gamma/\beta), \end{aligned} \quad (10.39)$$

which implies that when the input function is shifted and then a DTLCT is applied to the result, the output will be the same as the DTLCT being directly applied to the original input function, except that the result is shifted by an amount proportional to the shift in the input function and has a linear phase factor and a constant phase factor dependent on the shift amount.

For the DTLCT in Eq.(10.37), the summation is still infinite and the output variable u is continuous. In order to achieve a true discrete transform definition which maps discrete signals to discrete signals, we replace the infinite summation with a finite summation over N samples, but we assume that N is chosen sufficiently large so that the summation covers the interval over which the signal is non-negligible:

$$\exp(i\pi\alpha u^2) \sum_{n=-N/2}^{N/2-1} f(n\delta u) \exp[i\pi\gamma(n\delta u)^2] \exp(-i2\pi\beta un\delta u). \quad (10.40)$$

Now, we discretize the u variable by taking N samples in the following range:

$$-\frac{1}{2\delta u\beta} \leq u \leq \frac{1}{2\delta u\beta} - \frac{1}{N\delta u\beta} \quad (10.41)$$

in steps of $\delta u_M = 1/N\delta u\beta$ [74]. The discrete transform thus obtained repeats itself outside of this range. Thus we finally obtain the DLCT as a function of the discrete output variable m as:

$$\begin{aligned} & \mathcal{DLCT}_M[f(n\delta u)](m\delta u_M) \\ &= \exp\left[i\pi\alpha(m\delta u_M)^2\right] \sum_{n=-N/2}^{N/2-1} f(n\delta u) \exp[i\pi\gamma(n\delta u)^2] \exp[-i2\pi\beta(n\delta u)(m\delta u_M)] \\ &= \exp\left[i\pi\alpha\left(\frac{m}{N\delta u\beta}\right)^2\right] \sum_{n=-N/2}^{N/2-1} f(n\delta u) \exp[i\pi\gamma(n\delta u)^2] \exp\left(-\frac{i2\pi nm}{N}\right), \end{aligned} \tag{10.42}$$

where m covers the range $-N/2 \leq m \leq N/2 - 1$. A few remarks are in order at this point. First, note that this DLCT definition [75] contains a finite sum which arises from the sampling of the continuous input function and the continuous transform kernel. Second, the only string attached to the definition is $\delta u_M = 1/(N\delta u|\beta|)$; hence, there are many ways to choose the parameters N , δu , and δu_M , which correspond to the number of samples, and the sampling intervals in the input and output domains. In order to use this DLCT definition in practice to approximately compute the continuous LCT, it is necessary to know how to choose the number of samples N , and the sampling intervals δu and δu_M based on some prior information about the signal. The relationship between the DLCT and the continuous LCT is also needed to provide a foundation for how to use the DLCT to approximately compute the samples of the LCT of a continuous signal. The answers to these issues will be provided in the next section based on [77, 78]. The resulting LCT computation method allows us to work with the same number of samples in both the input and output domains without requiring any oversampling. On the other hand, the use of the Shannon–Nyquist sampling theorem instead of the LCT sampling theorem leads to problems such as the need to use a greater number of samples, requiring different sampling rates at intermediate stages of the computation, or different numbers of samples at the input and output domains [79, 80].

Now we go back to our discussion of the derivation of the FLCT algorithm. In [74], two important shifting properties of (10.42) have been derived. These two properties are essential in deriving the FLCT from the DLCT, much like the derivation of the FFT algorithm. These are:

$$\begin{aligned} & \exp\left[i\pi\alpha\left(\frac{m}{N\delta u\beta}\right)^2\right] \sum_{n=-\infty}^{\infty} [f(n\delta u) \exp(i2\pi\xi n\delta u)] \\ & \exp[i\pi\gamma(n\delta u)^2] \exp\left(-\frac{i2\pi nm}{N}\right) \end{aligned}$$

$$\begin{aligned}
&= \exp \left[-i\pi\alpha \left(\frac{\xi}{\beta} \right)^2 \right] \exp \left(\frac{i2\pi\alpha m\xi}{N\delta u\beta^2} \right) \exp \left[i\pi\alpha \left(\frac{m - \xi N\delta u}{N\delta u\beta} \right)^2 \right] \\
&\quad \times \sum_{n=-\infty}^{\infty} f(n\delta u) \exp[i\pi\gamma(n\delta u)^2] \exp \left[-\frac{i2\pi n(m - \xi N\delta u)}{N} \right], \tag{10.43}
\end{aligned}$$

$$\begin{aligned}
&\exp \left(\frac{i2\pi\xi m}{N\delta u\beta} \right) \exp \left[i\pi\alpha \left(\frac{m}{N\delta u\beta} \right)^2 \right] \\
&\quad \sum_{n=-\infty}^{\infty} f(n\delta u) \exp[i\pi\gamma(n\delta u)^2] \exp \left(-\frac{i2\pi nm}{N} \right) \\
&= \exp \left[i\pi\alpha \left(\frac{m}{N\delta u\beta} \right)^2 \right] \sum_{n=-\infty}^{\infty} f(n\delta u) \exp \left(\frac{i2\pi\gamma n\xi\delta u}{\beta} \right) \exp \left(-\frac{i\pi\gamma\xi^2}{\beta^2} \right) \\
&\quad \exp \left\{ i\pi\gamma \left[\left(n - \frac{\xi}{\delta u\beta} \right) \delta u \right]^2 \right\} \exp \left[-\frac{i2\pi m(n - \xi/\delta u\beta)}{N} \right]. \tag{10.44}
\end{aligned}$$

The properties given in Eqs.(10.43) and (10.44), in conjunction with the chirp periodicity of the DLCT, enable us to use the time (or space) decomposition and frequency (or spatial frequency) decomposition to derive the FLCT. The rest of the derivation is quite long and may be found in [74]. The resulting algorithm converts an N -point DLCT into four $N/4$ -point DLCTs. Recursively applying this and by properly choosing $N = 2^n$ for some integer n , we finally obtain 2×2 DLCTs to calculate. This gives an $O(N \log N)$ method to compute the DLCT. The procedure is similar to that for the conventional FFT.

Another significant work on fast computation of LCTs is [76]. In this paper, the DLCT and its fast algorithm, the FLCT, have been further studied by deriving an advanced FLCT that can work with an input sample vector of almost any length. The FLCT given in [74] is a radix-2 algorithm, so there is a restriction on the length of the input vector such that $N = 2^n$. For other lengths of the input vector, [76] proposes the following DLCT that can be implemented in any radix:

$$\mathcal{DLCT}_{\mathbf{M}}[f(n\delta u)](m\delta u_{\mathbf{M}}) = \sqrt{\frac{\beta}{2\pi}} \exp \left(\frac{-i\pi}{4} \right) \sum_{n=-N/2}^{N/2-1} f(n\delta u) W_N^{n,m}, \tag{10.45}$$

where

$$W_N^{n,m} = \exp\{i\pi[\alpha(m/N\delta u\beta)^2 - 2nm/N + \gamma(n\delta u)^2]\} \tag{10.46}$$

and \mathcal{DLCT}_M stands for DLCT operator with LCT parameters (α, β, γ) , δu is the sampling interval at the input, δu_M is the sampling interval at the output, and N is the length of the input vector. The details of the derivation can be found in [76]. The important point about this FLCT is that it can be implemented to work with any radix, so any input vector of any length can be DLCT-transformed with any desired radix in a way that the computation cost is minimized. An arbitrary vector length can be decomposed into its prime dividers and the corresponding radices can be chosen to optimize the computation. Note, however, that this does not work with vectors of prime length. No LCT algorithm exists, as yet, that can work on prime-length input vectors [76].

10.6 Hybrid LCT Algorithms

In Sect. 10.4, fast algorithms for computing continuous LCTs are discussed, which are obtained by decomposing continuous LCTs. These algorithms produce output vectors which are good approximations to the samples of the continuous transform, limited only by the fundamental fact that a signal cannot have finite extent in more than one domain. Since the sampling interval is ensured to satisfy the Nyquist criterion at each stage of the decomposition, the output samples can be used to reconstruct good approximations of the continuous LCT. While this approach is nearly optimal in terms of speed and accuracy, it has two disadvantages. Although it implicitly defines a discrete mapping from the vector of input samples to the vector of transform samples, this discrete mapping does not constitute an analytically elegant DLCT. More importantly, this type of approach sometimes requires an increase in the sampling rate, either during computation at intermediate stages, or in representation of the output. This is contrary to the fact that the area of the time- or space-frequency support, and hence the total number of samples required to represent a signal, remains unchanged when a signal undergoes linear canonical transformation.

In Sect. 10.5, we reviewed works that first propose a definition of the DLCT and then develop a fast algorithm for computing it. However, the issue of how to relate this DLCT to the continuous LCT we are trying to approximate with it has not been addressed in these works. In order to use these fast computation approaches in practice to approximately compute the continuous LCT, it is necessary to know how to choose the number of samples N , and the sampling intervals δu and δu_M . Note that the DLCT algorithm described in the last section has been developed with the following condition only: $\delta u_M = 1/(N\delta u|\beta|)$, leaving many ways to choose the parameters N , δu , and δu_M . Works that use the Shannon–Nyquist sampling theorem instead of the LCT sampling theorem lead to problems such as the need to use a greater number of samples or different numbers of samples at the input and output domains [79, 80].

In this section, we will present a FLCT computation method [77, 78] based on the DLCT defined in [75], which overcomes many of the limitations of

previous algorithms. This approach works with the minimum number of samples as determined by the LCT sampling theorem without requiring any sampling rate change at the intermediate stages. Moreover, it provides a good approximation to the continuous LCT as ensured by an exact relation between the DLCT and the continuous LCT [78].

We first review the exact relation between the DLCT and the continuous LCT given in [78], which provides a solid basis for how to use the DLCT to approximately compute the samples of the LCT of a continuous signal. This exact relation helps us use the DLCT to obtain a good and efficient approximation of the continuous LCT by properly choosing the number of samples. We want to obtain as good an approximation as possible, limited only by the fundamental fact that a signal cannot have strictly finite extent in more than one domain. The answer to this problem first appeared in [77], but became better established when it was formulated as an exact relation between the DLCT and the continuous LCT [78] in a manner similar to the classical theorem relating the DFT to the continuous Fourier transform [81].

The DLCT of $f(k\delta u)$ is defined as follows for $m = -N/2, \dots, N/2 - 1$ [75]:

$$\mathcal{DLCT}_{\mathbf{M}}[f(k\delta u)](m\delta u_{\mathbf{M}}) \equiv \delta u \sum_{k=-N/2}^{N/2-1} f(k\delta u) K_{\mathbf{M}}(m\delta u_{\mathbf{M}}, k\delta u),$$

$$K_{\mathbf{M}}(m\delta u_{\mathbf{M}}, k\delta u) = \sqrt{\beta} e^{-\frac{i\pi}{4}} e^{\frac{i\pi}{N|\beta|}(\alpha \frac{\delta u_{\mathbf{M}}}{\delta u} m^2 - 2\beta km + \gamma \frac{\delta u}{\delta u_{\mathbf{M}}} k^2)}, \quad (10.47)$$

where $\delta u_{\mathbf{M}} = (|\beta|N\delta u)^{-1}$. Here δu and $\delta u_{\mathbf{M}}$ are the sampling intervals in the time (or space) and LCT domains, respectively. N is the number of samples. This definition of DLCT can be made unitary by including an additional factor $\sqrt{\delta u_{\mathbf{M}}/\delta u}$ [78]. Let $f(u)$ and $f_{\mathbf{M}}(u)$ be a continuous-time signal and its LCT with parameters α, β, γ . Define the following periodically replicated functions where each period has been modulated with varying phase terms:

$$\bar{f}(u)_{(\mathbf{M}^{-1}, \Delta u)} \equiv \sum_{n=-\infty}^{\infty} f(u - n\Delta u) e^{-i\pi\gamma n\Delta u(2u - n\Delta u)}, \quad (10.48)$$

$$\bar{f}_{\mathbf{M}}(u)_{(\mathbf{M}, \Delta u_{\mathbf{M}})} \equiv \sum_{n=-\infty}^{\infty} f_{\mathbf{M}}(u - n\Delta u_{\mathbf{M}}) e^{i\pi\alpha n\Delta u_{\mathbf{M}}(2u - n\Delta u_{\mathbf{M}})}, \quad (10.49)$$

where Δu and $\Delta u_{\mathbf{M}}$ are arbitrary. These functions are chirp-periodic in the sense of [42]. Then, the exact relation between continuous and discrete LCTs can be stated as follows: *The samples of the chirp periodic functions defined in Eqs. (10.48) and (10.49) are exactly related to each other through the samples of the continuous kernel [the DLCT matrix in Eq. (10.47)]:*

$$\bar{f}_{\mathbf{M}}(m\delta u_{\mathbf{M}})_{(\mathbf{M}, \Delta u_{\mathbf{M}})} = \delta u \sum_{k \in \langle N \rangle} \bar{f}(k\delta u)_{(\mathbf{M}^{-1}, \Delta u)} K_{\mathbf{M}}(m\delta u_{\mathbf{M}}, k\delta u), \quad (10.50)$$

for any m and any interval of length N denoted by $\langle N \rangle$, and where the sampling intervals and the number of samples depends on the periods Δu and Δu_M as follows:

$$\delta u = \frac{1}{|\beta| \Delta u_M}, \quad \delta u_M = \frac{1}{|\beta| \Delta u}, \quad N = \Delta u \Delta u_M |\beta|. \quad (10.51)$$

This exact relation provides the underlying foundation for approximately computing the samples of the LCT of a continuous signal by replacing the transform integral with a finite sum. Because this exact relation generalizes the corresponding relation for Fourier transforms, which has been regarded as a fundamental theorem by Papoulis [81], this DLCT approximates the continuous LCT in the same sense that the DFT approximates the continuous FT. We can use this exact relation to see how this DLCT provides a good approximation of the continuous LCT. Consider an arbitrary signal $f(u)$ that is *not* chirp-periodic. Let us assume that a large percentage of the total energy of the signal is concentrated in the intervals $[-\Delta u/2, \Delta u/2]$ and $[-\Delta u_M/2, \Delta u_M/2]$, in the time (or space) and LCT domains, respectively. Then, $\tilde{f}(u)_{(M^{-1}, \Delta u)} \approx f(u)$ and $\tilde{f}_M(u)_{(M, \Delta u_M)} \approx f_M(u)$ in the respective intervals, and from (10.50) the DLCT of the samples of the function are the approximate samples of the continuous LCT of that function:

$$f_M(m \delta u_M) \approx \delta u \sum_{k=-N/2}^{N/2-1} f(k \delta u) K_M(m \delta u_M, k \delta u), \quad (10.52)$$

where δu , δu_M , and N are as in (10.51). If both the functions $f(u)$ and $f_M(u)$ could be identically zero outside of the given intervals, the mapping between the samples of these functions would be exact. But, since the extent of a function and its LCT cannot both be finite for $\beta \neq \infty$ [61, 82], there will be overlaps between the periodically replicated and phase modulated functions, and the DLCT will be an approximation between the samples of the continuous signals. This approximation for the LCT and FRT is similar to that for the FT, where in the FT case the limitation is that the extent of the signal and its Fourier transform cannot both be finite. The functions (10.48) and (10.49) reveal the precise nature of overlap and aliasing that occurs, which is somewhat different than the Fourier case which is a pure periodic replication. As with the DFT, the approximation improves with increasing N since this decreases the overlap between the replicas.

Note that we need chirp-periodic functions in order to have an exact equivalence between continuous and discrete LCTs, just as we need periodic functions to have an exact equivalence between continuous and discrete FTs. However, our interest in using this DLCT is mostly for *non*-chirp-periodic functions (just as we use the DFT for non-periodic functions), in which case the DLCT provides a good approximation to the continuous LCT, limited only by the fact that the extent of a function and its LCT cannot both be finite for $\beta \neq \infty$ [61, 82]. Moreover, the proper choice of the sampling intervals and the number of samples for a good approximation is given by the relations in (10.51).

This elegant and natural, fast and accurate LCT computation method provides a very different approach than that given in Sect. 10.4 [78]. In Sect. 10.4, we assumed that the time (or space) and frequency extents of the signals are specified in the conventional manner, defining an initial rectangular time- or space-frequency support. The number of samples were determined from the standard Nyquist–Shannon sampling theorem [59, 64, 65, 69, 79, 80, 83]. In this section, we assume the extents are specified in the input and output LCT domains; that is, in the original time or space domain, and the target LCT domain. This leads to an initial parallelogram-shaped time- or space-frequency support [84, 85]. The number of samples is determined from the LCT sampling theorem [78]. The minimum number of samples required for computation is given by the so-called bicanonical width product [78], which is also equal to the area of the parallelogram support. The DLCT defined in [75] works with this minimum number of samples without requiring any interpolation or oversampling at the intermediate stages of the computation, in contrast to previously given approaches [79, 80] for the same DLCT. Also recall that, sampling the input and output by using the Nyquist–Shannon sampling theorem, as in these works and Sect. 10.4, usually leads to a greater number of samples and sometimes requires different numbers of samples for the input and output signals.

So far, we discussed how this DLCT can be used to accurately compute the samples of a continuous DLCT. Now, let us discuss how this DLCT can be implemented in a fast way. As discussed previously, the DLCT in (10.47) can be evaluated with either of two fast methods:

1. A direct approach by successively performing a chirp multiplication, a FFT and a second chirp multiplication (by taking advantage of the simple form of the DLCT) [42, 75, 78].
2. A radix-type approach that generalizes the FFT for the Fourier transform to the LCT (by taking advantage of the shifting properties of the DLCT) [74, 76].

Both approaches yield efficient computation in $O(N \log N)$ time. The first of these is simpler, since by employing the FFT as a building block, it does not require us to get into the complications of a radix-type approach [78]. This also has the advantage of relying on widely available, highly optimized FFT implementations, whereas such optimized implementations do not exist for the radix-type approach to fast LCT computation.

The DLCT definition given in Eq. (10.47) has many desirable properties when used with the parameters satisfying Eq. (10.51) [78]. It has a simple analytic expression and is unitary. It can be efficiently computed in $O(N \log N)$ time by successively performing a chirp multiplication, a FFT, and a second chirp multiplication. It has a well-defined relationship to the continuous LCT [77, 78]. Its accuracy is only limited by the fundamental fact that a signal cannot have strictly finite extent in more than one domain. Therefore it is an important candidate for being a widely accepted definition of the discrete version of the LCT.

We refer to this type of approach to LCT computation as a hybrid algorithm. It involves both an analytically desirable definition of the DLCT and fast

decomposition-based computation by successively performing a chirp multiplication, a FFT, and a second chirp multiplication. Thus, it combines the best features of DLCT based algorithms and decomposition based algorithms. It conserves computational resources by eliminating the interpolation/decimation steps, as well as keeping the number of samples at the minimum possible [78].

This approach has been revisited in [85], where an interpretation of the method has been given through phase-space diagrams, revealing the decomposition-based nature of the algorithm (similar to the decompositions of the continuous LCT discussed in Sect. 10.4). The decomposition consists of a chirp multiplication, magnified FT, and a second chirp multiplication [63]. It first shears the parallelogram-shaped initial support into a rectangle, then rotates it by 90° , and then again shears it back to a parallelogram with the same area as the initial parallelogram. During these stages, no overlap is introduced between the replicated supports (arising from sampling). Hence this phase-space picture allows us to see from yet another perspective how this elegant and accurate LCT computation method [77, 78] works with the minimum number of samples, without requiring interpolation.

Finally, before ending this section, we also mention another fast algorithm for LCT computation which is based on an alternative form for the unitary DLCT [86]. In this approach, a convergent quadrature definition for the continuous Fourier transform is used to decompose arbitrary LCTs as a scaling, chirp, DFT, chirp, scaling decomposition. The decomposition and the DFT definition developed in this paper results in the following algorithm for fast digital computation of LCTs: Let the vector \mathbf{g} , of length N , stand for the samples of the LCT of $f(u)$ with parameters A, B, C, D , evaluated at the points $v_j = 4Bu_j/\pi$, where $u_j = (\frac{2^{j-N-1}}{\sqrt{2N}})\pi/2$. Now,

1. Set up the vector \mathbf{p} with $p_k = e^{-i\pi \frac{(k-1)(N-1)}{N}} e^{iAu_k^2/2B} f(u_k)$, where $k = 1, 2, \dots, N$. This step corresponds to first scaling the input samples and then multiplying with the chirp in the decomposition.
2. Take the conventional DFT of the vector \mathbf{p} using the FFT algorithm and denote the result as \mathbf{q} .
3. Construct the diagonal matrix \mathbf{S} as $\mathbf{S}_{jk} = \frac{\pi e^{i\frac{\pi}{2} \frac{(N-1)^2}{N}} e^{\frac{iDv_j^2}{2B}} e^{-i\pi \frac{N-1}{N}(j-1)}}{\sqrt{2N} \sqrt{2\pi iB}} \delta_{jk}$ where $j, k = 1, 2, \dots, N$. This matrix serves to implement the second chirp multiplication. δ_{jk} stands for the Kronecker delta.
4. Lastly, obtain the result vector \mathbf{g} by multiplying the vector \mathbf{q} with the diagonal matrix \mathbf{S} . The elements of the vector \mathbf{g} correspond to samples of the continuous LCT of $f(u)$ at the points $v_j = 4Bu_j/\pi$.

None of the above steps takes more than $O(N \log N)$ time, so that the overall complexity is $O(N \log N)$.

10.7 Computation of Two-Dimensional LCTs

Two-dimensional *separable* LCTs are addressed in [8, 10, 87–91]. The most special case is the isotropic 2D LCT in which the system is fully symmetric, orthogonal, and the parameters for both dimensions are identical. This case can be represented by only three parameters as in a 1D LCT [24]. When the system is still orthogonal but the parameters for the orthogonal dimensions differ, the system becomes a 2D separable LCT, which is represented by six parameters [24]. Separable 2D transforms do not pose much difficulty because the separable transform is essentially two independent 1D transforms along the two dimensions and the dimensions can be treated independently. However, the *non-separable* transform (2D NS LCT) is significantly more general. The two dimensions are coupled to each other by four additional cross-parameters, increasing the total number of parameters to ten. This general case is non-separable, non-axially symmetric, non-orthogonal, and anamorphic/astigmatic [2, 24, 27, 68, 73]. 2D NS LCTs are able to represent not only systems involving anamorphic/astigmatic components and reference surfaces, but also other interesting systems such as optical mode convertors and resonators since they can represent the coupling between the dimensions [24, 92–94]. Another prominent feature of 2D NS LCTs is their ability to represent systems with rotations between any arbitrary planes in phase-space, like rotations and gyrations [24, 27]. These systems are collected under the general name of gyrators and are useful in two-dimensional image processing, signal processing, mode transformation, etc. [27, 95–98].

Given an algorithm for efficiently computing 1D LCTs [64, 65, 74], the efficient computation of separable 2D transforms is straightforward because the kernel can be separated and the 2D transform can be reduced to two successive 1D LCTs. On the other hand, in the non-separable case, the two dimensions are coupled and handling this case requires special attention. An alternative representation of LCTs is presented and studied in [67]. This decomposition is based on the well-known Iwasawa decomposition [99]. In [67], the authors further decompose the first matrix of the Iwasawa decomposition into a two-dimensional *separable* fractional Fourier transform (2D S FRT) that is sandwiched between two coordinate rotators. Earlier in this chapter, we had mentioned the deployment of the 1D version of the Iwasawa decomposition to develop a fast and efficient algorithm for 1D LCTs [64, 65]. By using the 2D Iwasawa-type decomposition of [67], it becomes likewise possible to derive an efficient algorithm for the computation of 2D NS LCTs [83].

The 2D NS LCT with parameter matrix \mathbf{M} , of an input function $f(\mathbf{u})$, can be expressed as [67, 100]

$$\begin{aligned} f_{\mathbf{M}}(\mathbf{u}) &= (\mathcal{C}_{\mathbf{M}}f)(\mathbf{u}) \\ &= \frac{1}{\sqrt{\det i\mathbf{B}}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp[i\pi(\mathbf{u}'^T \mathbf{B}^{-1} \mathbf{A} \mathbf{u}' \\ &\quad - 2\mathbf{u}'^T \mathbf{B}^{-1} \mathbf{u} + \mathbf{u}'^T \mathbf{D} \mathbf{B}^{-1} \mathbf{u})] f(\mathbf{u}') \, d\mathbf{u}', \end{aligned} \quad (10.53)$$

where $\mathbf{u} = [u_x \ u_y]^T$, $\mathbf{u}' = [u'_x \ u'_y]^T$ with T denoting the transpose operation. \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} are 2×2 submatrices defining the transformation matrix \mathbf{M} of the system that represents the 2D-LCT, \mathbf{B} being nonsingular. The matrix $\mathbf{M} = [\mathbf{A} \ \mathbf{B}; \ \mathbf{C} \ \mathbf{D}]$ is real and symplectic. From a group-theoretical point of view, 2D NS LCTs form the ten-parameter symplectic group $Sp(4, \mathbb{R})$. (\mathbf{M} has 16 parameters with six constraints leaving ten independent parameters.) More on group-theoretical properties of LCTs can be found in [8].

The Iwasawa decomposition is the core of our algorithm. After the dimensional normalization explained in Sect. 10.2.1, any transformation matrix \mathbf{M} can be written in the following Iwasawa form [67, 99]:

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{G} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ -\mathbf{Y} & \mathbf{X} \end{bmatrix}, \quad (10.54)$$

where

$$\mathbf{G} = -(\mathbf{C}\mathbf{A}^T + \mathbf{D}\mathbf{B}^T)(\mathbf{A}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T)^{-1}, \quad (10.55)$$

$$\mathbf{S} = (\mathbf{A}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T)^{1/2}, \quad (10.56)$$

$$\mathbf{X} = (\mathbf{A}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T)^{-1/2}\mathbf{A}, \quad (10.57)$$

$$\mathbf{Y} = (\mathbf{A}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T)^{-1/2}\mathbf{B}. \quad (10.58)$$

Given the 4×4 matrix \mathbf{M} , we can determine 2×2 matrices \mathbf{G} , \mathbf{S} , \mathbf{X} , \mathbf{Y} by using Eqs. (10.55)–(10.58). If we are able to develop a fast algorithm to compute the three stages in $O(\tilde{N} \log \tilde{N})$ time, the overall transform can also be calculated in $O(\tilde{N} \log \tilde{N})$ time where \tilde{N} stands for the total number of samples in a 2D signal. In this decomposition, the first operation is an orthosymplectic system, followed by a scaling (magnification) system, finally followed by a two-dimensional chirp multiplication (2D CM). (Note that each of the stages of the algorithm are special cases of 2D NS LCTs.)

The first and the most sophisticated stage of the decomposition is the orthosymplectic system. This stage of the decomposition can be further decomposed into a 2D S FRT that is sandwiched between two coordinate rotators [67]:

$$\begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ -\mathbf{Y} & \mathbf{X} \end{bmatrix} = \mathbf{R}_{r_2} \mathbf{F}_{a_x, a_y} \mathbf{R}_{r_1}, \quad (10.59)$$

where the 4×4 matrices \mathbf{R}_{r_1} , \mathbf{F}_{a_x, a_y} , \mathbf{R}_{r_2} are defined as:

$$\mathbf{R}_{r_1} = \begin{bmatrix} \cos(r_1) & \sin(r_1) & 0 & 0 \\ -\sin(r_1) & \cos(r_1) & 0 & 0 \\ 0 & 0 & \cos(r_1) & \sin(r_1) \\ 0 & 0 & -\sin(r_1) & \cos(r_1) \end{bmatrix}, \quad (10.60)$$

$$\mathbf{R}_{r_2} = \begin{bmatrix} \cos(r_2) & \sin(r_2) & 0 & 0 \\ -\sin(r_2) & \cos(r_2) & 0 & 0 \\ 0 & 0 & \cos(r_2) & \sin(r_2) \\ 0 & 0 & -\sin(r_2) & \cos(r_2) \end{bmatrix}, \quad (10.61)$$

$$\mathbf{F}_{a_x, a_y} = \begin{bmatrix} \cos(a_x\pi/2) & 0 & \sin(a_x\pi/2) & 0 \\ 0 & \cos(a_y\pi/2) & 0 & \sin(a_y\pi/2) \\ -\sin(a_x\pi/2) & 0 & \cos(a_x\pi/2) & 0 \\ 0 & -\sin(a_y\pi/2) & 0 & \cos(a_y\pi/2) \end{bmatrix}. \quad (10.62)$$

\mathbf{R}_{r_1} and \mathbf{R}_{r_2} are rotation matrices that impose rotations of angles r_1 and r_2 , respectively, through the spatial variables (u_x, u_y) and through their frequency variables (μ_x, μ_y) . Unlike these traditional rotators which rotate within space and spatial frequency separately, the FRT rotates within the space-frequency planes of each dimension. \mathbf{F}_{a_x, a_y} stands for the 2D S FRT that makes separable rotations of angle $a_x\pi/2$ in the (u_x, μ_x) plane and of angle $a_y\pi/2$ in the (u_y, μ_y) plane. Since this 2D FRT operation is separable, it corresponds to two 1D FRT operations performed over each of the dimensions. This amounts to first performing 1D FRTs with the fractional order a_x for each of the rows (or columns) and then performing 1D FRTs with the fractional order a_y for each of the columns (or rows) of the sampling grid. It is this observation that enables us to implement this stage of the decomposition efficiently in $O(\tilde{N} \log \tilde{N})$ time.

The interpretation of the coordinate rotators requires care. When we are working with sampled functions, we know the value and coordinates (the location where the particular sample is taken) of all the samples we have. A coordinate rotation can be interpreted in this situation as a rotation of the locations of the samples resulting in a new sampling grid, rather than a change in the sample values. If we assume we start with a regular rectangular grid, after the coordinate rotation, the grid would no longer coincide with the original grid unless the rotation is an integer multiple of $\pi/2$. Unfortunately, in order to perform FRT operations along the horizontal and vertical directions, we need the samples to be on a regular rectangular grid in order to employ available fast algorithms. Therefore, we must carry out an interpolation operation to determine the values of the function on a regular rectangular grid. In summary, the first stage of our algorithm involves determining the angle parameters for the first coordinate rotation, followed by two 1D FRTs over each of the dimensions, and then followed by the second coordinate rotation. These angles can be computed by equating the LCT matrix with the decomposition matrix and solving for the angles. All these steps can be calculated in $O(\tilde{N} \log \tilde{N})$ time.

The second stage is the scaling operation and it seems to be the simplest of the three stages. It is not, however, as trivial as in the 1D case [65]. In 1D, it corresponds to only a reinterpretation of the spacing between the samples. The sampling interval scales with the scaling parameter. Intuitively, it squeezes in or stretches out the total number of samples as the word scaling implies. This means there is no change in the total number of samples and thus no need to oversample

the input samples. The analogue of the 1D scalar scaling parameter in the 2D case is the matrix \mathbf{S} . When \mathbf{S} is diagonal, which means there is no coupling between the two dimensions of the function for scaling purposes, the scaling is separable. Due to this separability, this situation does not impose an increase in the space-bandwidth products and thus does not require oversampling, just as in the 1D case. But when the off-diagonal elements of \mathbf{S} are non-zero, the scaling operation is no longer so trivial. Computationally, such a scaling operation amounts to modifying the information that tells us which coordinates the samples belong to. Nevertheless, since it requires only the reinterpretation of the coordinates of the samples plus a possible oversampling, it does not impose much computational load. The matrix \mathbf{S} can be easily used to determine the output samples by using the input–output relation of the scaling operation:

$$f_{sc}(\mathbf{u}) = \frac{1}{\sqrt{\det \mathbf{S}}} f(\mathbf{S}^{-1} \mathbf{u}), \quad (10.63)$$

where f is the function to be scaled and f_{sc} is the scaled function, and $\mathbf{u} = [u_x \ u_y]^T$.

The last stage of the main Iwasawa decomposition is the 2D CM operation whose parameters are given by the matrix \mathbf{G} as defined in Eq. (10.55). The input–output relation of this 2D-CM is given as:

$$f_{ch}(\mathbf{u}) = e^{-i\pi(G_{11}u_x^2 + (G_{12} + G_{21})u_x u_y + G_{22}u_y^2)} f(\mathbf{u}), \quad (10.64)$$

where f_{ch} stands for the chirp-multiplied function. The 2D CM operation is the stage that is mainly burdened with any shears inherent in the 2D NS LCT to be computed. Such shears may considerably increase the space-bandwidth products of the function. Thus, before the 2D CM operation, the space-bandwidth products of the function should be calculated carefully and any necessary oversampling should be performed. This CM operation may turn out to be non-separable or separable for particular 2D NS LCTs but regardless, it requires only one multiplication for each sample, resulting in $O(\tilde{N})$ time computation.

As in the 1D case, this algorithm also has the ability to track the space-bandwidth product of the function through each step in order to control the sampling rate of the function with the goal of having enough samples to be able to reconstruct the continuous function without information loss, and at the same time without needlessly increasing the number of samples to maintain efficiency. However, the sampling rate control mechanism is quite involved so we refer the reader to [83] for details.

Above, we presented the 2D-NS-LCT fast computation algorithm based on the Iwasawa decomposition, which is a generalization of the algorithm denoted as A2 in Sect. 10.4. In the same section, we also mentioned another algorithm, denoted A1, which is also based on basic decompositions. In [101], generalization of the A1 algorithm to 2D-NS-LCT computation has been considered. The results of both approaches are comparable to each other both in 1D and 2D.

10.8 Computation of Complex-Parametered Linear Canonical Transforms

Extension of real-parametered LCTs to complex-parametered LCTs (CLCTs) is rather involved. Bilateral Laplace transforms, Bargmann transforms, Gauss–Weierstrass transforms, [8, 28, 29], fractional Laplace transforms [30, 31], and CFRTs [32–35] are all special cases of complex LCTs. More on the mathematical foundations and theory of CLCTs can be found in [8, 28, 102, 103].

Complex-parametered LCTs allow several kinds of optical systems to be represented, including lossy as well as lossless ones. Magnification (scaling), Fourier transformation (FT), real fractional Fourier transformation (RFRT), real chirp multiplication (CM), complex chirp multiplication (CCM), Gauss–Weierstrass Transform, CFRT are all special cases of CLCTs that have optical realizations. The CFRT is the generalization of the FRT where the order of the transformation is allowed to be a complex number, and consequently the $ABCD$ matrix elements are in general complex. The optical interpretation of the CFRT, its properties and optical realizations can be found in [32–35, 104].

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

$$(\mathcal{C}_{\mathbf{M}_C}f)(u) = \int_{-\infty}^{\infty} K_C(u, u')f(u') du',$$

$$K_C(u, u') = e^{-i\pi/4} \sqrt{\bar{\beta}} \exp \left[i\pi(\bar{\alpha}u^2 - 2\bar{\beta}uu' + \bar{\gamma}u'^2) \right], \quad (10.65)$$

where $\bar{\alpha}, \bar{\beta}, \bar{\gamma}$ are complex parameters independent of u and u' and where $\mathcal{C}_{\mathbf{M}_C}$ is the CLCT operator. \mathbf{M}_C again has unit-determinant and is given by

$$\mathbf{M}_C = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} A_r + iA_c & B_r + iB_c \\ C_r + iC_c & D_r + iD_c \end{bmatrix} = \begin{bmatrix} \bar{\gamma}/\bar{\beta} & 1/\bar{\beta} \\ -\bar{\beta} + \bar{\alpha}\bar{\gamma}/\bar{\beta} & \bar{\alpha}/\bar{\beta} \end{bmatrix}, \quad (10.66)$$

where $A_r, A_c, B_r, B_c, C_r, C_c, D_r, D_c$ are real numbers.

Fast digital computation algorithms based on decomposition approaches have been developed. Here we will only show how given $ABCD$ matrices can be decomposed in a manner that leads to a fast algorithm for computation of CLCTs. For further details, we refer to [105]. In the most general case, the matrix \mathbf{M}_C is composed of the four complex parameters A, B, C, D , whose real and imaginary parts add up to a total of eight parameters. These eight parameters are restricted by the unimodularity condition on \mathbf{M}_C , which requires the real part of the determinant to be 1 and the imaginary part to be 0. Because of these two equations, the total number of independent parameters of a general CLCT is 6. The following decomposition will be the basis of the fast algorithm for CLCTs:

$$\mathbf{M}_C = \begin{bmatrix} 1 & 0 \\ -q_{3r} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -iq_{3c} & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -q_{2r} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -iq_{2c} & 1 \end{bmatrix} \\ \times \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -q_{1r} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -iq_{1c} & 1 \end{bmatrix}. \quad (10.67)$$

The above algorithm can efficiently compute any arbitrary CLCT with complex parameters but the output plane should be real, i.e. the variable at the transform output is not complex. This is indeed the case for most of the real-world cases, where one is interested in the output field on a particular plane. In [106], the above algorithm has been further generalized to cover any complex variable at the output. The sampling issues of CLCTs have also been studied in detail in [106].

Before ending this section, we mention that LCTs are a special case of the more general family of oscillatory integrals of the form $F(w) = \int f(u)e^{i\text{wg}(u)}du$, $w \gg 1$. Generally speaking, computation of such more general integrals is time consuming. In [107], a method to convert any general oscillatory integral to a canonical form $F(w) = \int_{-1}^1 f(u)e^{i\text{w}u}du$ and then compute it in $O(N^2)$ time is presented.

10.9 Conclusion

In this chapter, we discussed algorithms for computation of LCTs from the N samples of the input signal in $O(N \log N)$ time. 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 theoretical minimum required to represent the signals. The transform output may have a higher time- or space-bandwidth product due to the nature of the transform family, though it has the same bicanonical width product with the input.

We considered three groups of algorithms. In the first, the LCT operation is decomposed into more elementary operations. Most elegant among these is based on the Iwasawa decomposition, which involves the FRT. In the second, one first defines a DLCT, and then obtains a FLCT algorithm for this DLCT, much like the FFT algorithm for the DFT. In the third group, a DLCT is defined, but the computation procedure is based on decompositions.

The algorithms can relate the samples of the input function to the samples of the continuous LCT of this function in the same sense that the FFT implementation of the 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. This limitation affects not only LCT algorithms, but also the computation of Fourier transforms using the DFT. Thus this is a source of error we cannot hope to overcome.

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.

Acknowledgement H.M. Ozaktas acknowledges partial support of the Turkish Academy of Sciences.

References

1. M.J. Bastiaans, Wigner distribution function and its application to first-order optics. *J. Opt. Soc. Am.* **69**, 1710–1716 (1979)
2. A.E. Siegman, *Lasers* (University Science Books, Mill Valley, 1986)
3. D.F.V. James, G.S. Agarwal, The generalized Fresnel transform and its application to optics. *Opt. Commun.* **126**(4–6), 207–212 (1996)
4. C. Palma, V. Bagini, Extension of the Fresnel transform to ABCD systems. *J. Opt. Soc. Am. A* **14**(8), 1774–1779 (1997)
5. S. Abe, J.T. Sheridan, Generalization of the fractional Fourier transformation to an arbitrary linear lossless transformation an operator approach. *J. Phys. A Math. Gen.* **27**(12), 4179–4187 (1994)
6. S. Abe, J.T. Sheridan, Optical operations on wavefunctions as the Abelian subgroups of the special affine Fourier transformation. *Opt. Lett.* **19**, 1801–1803 (1994)
7. J. Hua, L. Liu, G. Li, Extended fractional Fourier transforms. *J. Opt. Soc. Am. A* **14**(12), 3316–3322 (1997)
8. K.B. Wolf, Construction and properties of canonical transforms, Chap. 9, in *Integral Transforms in Science and Engineering* (Plenum Press, New York, 1979)
9. E. Hecht, *Optics*, 4th edn. (Addison Wesley, Reading, 2001)
10. H.M. Ozaktas, Z. Zalevsky, M.A. Kutay, *The Fractional Fourier Transform with Applications in Optics and Signal Processing* (Wiley, New York, 2001)
11. M.J. Bastiaans, The Wigner distribution function applied to optical signals and systems. *Opt. Commun.* **25**(1), 26–30 (1978)
12. M.J. Bastiaans, Applications of the Wigner distribution function in optics, in *The Wigner Distribution: Theory and Applications in Signal Processing*, ed. by W. Mecklenbräuer, F. Hlawatsch (Elsevier, Amsterdam, 1997), pp. 375–426
13. M. Moshinsky, Canonical transformations and quantum mechanics. *SIAM J. Appl. Math.* **25**(2), 193–212 (1973)
14. C. Jung, H. Kruger, Representation of quantum mechanical wavefunctions by complex valued extensions of classical canonical transformation generators. *J. Phys. A Math. Gen.* **15**, 3509–3523 (1982)
15. B. Davies, *Integral Transforms and Their Applications* (Springer, New York, 1978)
16. D.J. Griffiths, C.A. Steinke, Waves in locally periodic media. *Am. J. Phys.* **69**(2), 137–154 (2001)
17. D.W.L. Sprung, H. Wu, J. Martorell, Scattering by a finite periodic potential. *Am. J. Phys.* **61**(12), 1118–1124 (1993)
18. L.L. Sanchez-Soto, J.F. Carinena, A.G. Barriuso, J.J. Monzon, Vector-like representation of one-dimensional scattering. *Eur. J. Phys.* **26**(3), 469–480 (2005)

19. S. Baskal, Y.S. Kim, Lens optics as an optical computer for group contractions. *Phys. Rev. E* **67**(5), 056601 (2003)
20. S. Baskal, Y.S. Kim, ABCD matrices as similarity transformations of Wigner matrices and periodic systems in optics. *J. Opt. Soc. Am. A* **26**(9), 2049–2054 (2009)
21. E. Georgieva, Y.S. Kim, Slide-rule-like property of Wigner's little groups and cyclic S matrices for multilayer optics. *Phys. Rev. E* **68**(2), 026606 (2003)
22. B. Barshan, M.A. Kutay, H.M. Ozaktas, Optimal filtering with linear canonical transformations. *Opt. Commun.* **135**(1–3), 32–36 (1997)
23. S.C. Pei, J.J. Ding, Eigenfunction of linear canonical transform. *IEEE Trans. Signal Process.* **50**, 11–26 (2002)
24. T. Alieva, M.J. Bastiaans, Properties of the canonical integral transformation. *J. Opt. Soc. Am. A* **24**, 3658–3665 (2007)
25. M.J. Bastiaans, T. Alieva, Classification of lossless first-order optical systems and the linear canonical transformation. *J. Opt. Soc. Am. A* **24**, 1053–1062 (2007)
26. J. Rodrigo, T. Alieva, M. Luisa Calvo, Optical system design for orthosymplectic transformations in phase space. *J. Opt. Soc. Am. A* **23**, 2494–2500 (2006)
27. R. Simon, K.B. Wolf, Structure of the set of paraxial optical systems. *J. Opt. Soc. Am. A* **17**(2), 342–355 (2000)
28. K.B. Wolf, Canonical transformations I. Complex linear transforms. *J. Math. Phys.* **15**(8), 1295–1301 (1974)
29. K.B. Wolf, On self-reciprocal functions under a class of integral transforms. *J. Math. Phys.* **18**(5), 1046–1051 (1977)
30. A. Torre, Linear and radial canonical transforms of fractional order. *J. Comput. Appl. Math.* **153**, 477–486 (2003)
31. K.K. Sharma, Fractional Laplace Transform. *Signal Image Video Process.* **4**(3), 377–379 (2009)
32. C.C. Shih, Optical interpretation of a complex-order Fourier transform. *Opt. Lett.* **20**(10), 1178–1180 (1995)
33. L.M. Bernardo, O.D.D. Soares, Optical fractional Fourier transforms with complex orders. *Appl. Opt.* **35**(17), 3163–3166 (1996)
34. C. Wang, B. Lu, Implementation of complex-order Fourier transforms in complex ABCD optical systems. *Opt. Commun.* **203**(1–2), 61–66 (2002)
35. L.M. Bernardo, Talbot self-imaging in fractional Fourier planes of real and complex orders. *Opt. Commun.* **140**, 195–198 (1997)
36. N.M. Atakishiyev, K.B. Wolf, Fractional Fourier–Kravchuk transform. *J. Opt. Soc. Am. A* **14**(7), 1467–1477 (1997)
37. S.C. Pei, M.H. Yeh, Improved discrete fractional Fourier transform. *Opt. Lett.* **22**(14), 1047–1049 (1997)
38. N.M. Atakishiyev, S.M. Chumakov, K.B. Wolf, Wigner distribution function for finite systems. *J. Math. Phys.* **39**(12), 6247–6261 (1998)
39. N.M. Atakishiyev, L.E. Vicent, K.B. Wolf, Continuous vs. discrete fractional Fourier transforms. *J. Comput. Appl. Math.* **107**(1), 73–95 (1999)
40. S.C. Pei, M.H. Yeh, C.C. Tseng, Discrete fractional Fourier transform based on orthogonal projections. *IEEE Trans. Signal Process.* **47**(5), 1335–1348 (1999)
41. S.C. Pei, M.H. Yeh, T.L. Luo, Fractional Fourier series expansion for finite signals and dual extension to discrete-time fractional Fourier transform. *IEEE Trans. Signal Process.* **47**(10), 2883–2888 (1999)
42. T. Erseghe, P. Kraniuskas, G. Carioraro, Unified fractional Fourier transform and sampling theorem. *IEEE Trans. Signal Process.* **47**(12), 3419–3423 (1999)
43. M.A. Kutay, H. Ozaktas, H.M. Ozaktas, O. Arikan, The fractional Fourier domain decomposition. *Signal Process.* **77**(1), 105–109 (1999)
44. A.I. Zayed, A.G. Garçça, New sampling formulae for the fractional Fourier transform. *Signal Process.* **77**(1), 111–114 (1999)

45. C. Candan, M.A. Kutay, H.M. Ozaktas, The discrete fractional Fourier transform. *IEEE Trans. Signal Process.* **48**(5), 1329–1337 (2000)
46. I.S. Yetik, M.A. Kutay, H. Ozaktas, H.M. Ozaktas, Continuous and discrete fractional Fourier domain decomposition, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'00)*, vol. 1 (2000), pp. 93–96
47. S.C. Pei, M.H. Yeh, The discrete fractional cosine and sine transforms. *IEEE Trans. Signal Process.* **49**(6), 1198–1207 (2001)
48. G. Cariolaro, T. Erseghe, P. Kraniuskas, The fractional discrete cosine transform. *IEEE Trans. Signal Process.* **50**(4), 902–911 (2002)
49. L. Barker, Continuum quantum systems as limits of discrete quantum systems, IV. Affine canonical transforms. *J. Math. Phys.* **44**(4), 1535–1553 (2003)
50. C. Candan, H.M. Ozaktas, Sampling and series expansion theorems for fractional Fourier and other transforms. *Signal Process.* **83**(11), 2455–2457 (2003)
51. J.G. Vargas-Rubio, B. Santhanam, On the multiangle centered discrete fractional Fourier transform. *IEEE Signal Process. Lett.* **12**(4), 273–276 (2005)
52. M.H. Yeh, Angular decompositions for the discrete fractional signal transforms. *Signal Process.* **85**(3), 537–547 (2005)
53. K.B. Wolf, Finite systems, fractional Fourier transforms and their finite phase spaces. *Czech. J. Phys.* **55**, 1527–1534 (2005)
54. K.B. Wolf, Finite systems on phase space. *J. Mod. Phys. B* **20**(11), 1956–1967 (2006)
55. K.B. Wolf, G. Krötzsch, Geometry and dynamics in the fractional discrete Fourier transform. *J. Opt. Soc. Am. A* **24**(3), 651–658 (2007)
56. D. Mendlovic, Z. Zalevsky, N. Konforti, Computation considerations and fast algorithms for calculating the diffraction integral. *J. Mod. Opt.* **44**(2), 407–414 (1997)
57. D. Mas, J. Garcia, C. Ferreira, L.M. Bernardo, F. Marinho, Fast algorithms for free-space diffraction patterns calculation. *Opt. Commun.* **164**(4–6), 233–245 (1999)
58. J.W. Cooley, J.W. Tukey, An algorithm for the machine calculation of complex Fourier series. *Math. Comput.* **19**, 297–301 (1965)
59. H.M. Ozaktas, O. Arikan, M.A. Kutay, G. Bozdağı, Digital computation of the fractional Fourier transform. *IEEE Trans. Signal Process.* **44**, 2141–2150 (1996)
60. M.J. Bastiaans, The Wigner distribution function and Hamilton's characteristics of a geometric-optical system. *Opt. Commun.* **30**(3), 321–326 (1979)
61. F.S. Oktem, H.M. Ozaktas, Equivalence of linear canonical transform domains to fractional Fourier domains and the bicanonical width product: a generalization of the space-bandwidth product. *J. Opt. Soc. Am. A* **27**(8), 1885–1895 (2010)
62. F. Hlawatsch, G.F. Boudreaux-Bartels, Linear and quadratic time-frequency signal representations. *IEEE Signal Process. Mag.* **9**(2), 21–67 (1992)
63. L. Cohen, *Time-Frequency Analysis* (Prentice Hall, Englewood Cliffs, 1995)
64. H.M. Ozaktas, A. Koç, I. Sari, M.A. Kutay, Efficient computation of quadratic-phase integrals in optics. *Opt. Lett.* **31**, 35–37 (2006)
65. A. Koç, H.M. Ozaktas, C. Candan, M.A. Kutay, Digital computation of linear canonical transforms. *IEEE Trans. Signal Process.* **56**(6), 2383–2394 (2008)
66. H.M. Ozaktas, M.F. Erden, Relationships among ray optical, Gaussian beam, and fractional Fourier transform descriptions of first-order optical systems. *Opt. Commun.* **143**, 75–86 (1997)
67. T. Alieva, M.J. Bastiaans, Alternative representation of the linear canonical integral transform. *Opt. Lett.* **30**(24), 3302–3304 (2005)
68. M.J. Bastiaans, T. Alieva, Synthesis of an arbitrary ABCD system with fixed lens positions. *Opt. Lett.* **31**, 2414–2416 (2006)
69. B.M. Hennelly, J.T. Sheridan, Generalizing, optimizing, and inventing numerical algorithms for the fractional Fourier, Fresnel, and linear canonical transforms. *J. Opt. Soc. Am. A* **22**, 917–927 (2005)

70. X. Yang, Q. Tan, X. Wei, Y. Xiang, Y. Yan, G. Jin, Improved fast fractional-Fourier-transform algorithm. *J. Opt. Soc. Am. A* **21**(9), 1677–1681 (2004)
71. J. García, D. Mas, R.G. Dorsch, Fractional-Fourier-transform calculation through the fast-Fourier-transform algorithm. *Appl. Opt.* **35**(35), 7013–7018 (1996)
72. F.J. Marinho, L.M. Bernardo, Numerical calculation of fractional Fourier transforms with a single fast-Fourier-transform algorithm. *J. Opt. Soc. Am. A* **15**(8), 2111–2116 (1998)
73. X. Liu, K.H. Brenner, Minimal optical decomposition of ray transfer matrices. *Appl. Opt.* **47**(22), E88–E98 (2008)
74. B.M. Hennelly, J.T. Sheridan, Fast numerical algorithm for the linear canonical transform. *J. Opt. Soc. Am. A* **22**, 928–937 (2005)
75. S.C. Pei, J.J. Ding, Closed-form discrete fractional and affine Fourier transforms. *IEEE Trans. Signal Process.* **48**, 1338–1353 (2000)
76. J.J. Healy, J.T. Sheridan, Fast linear canonical transforms. *J. Opt. Soc. Am. A* **27**(1), 21–30 (2010)
77. A. Stern, Why is the linear canonical transform so little known?, in *AIP Conference Proceedings*, 2006, pp. 225–234
78. F.S. Oktem, H.M. Ozaktas, Exact relation between continuous and discrete linear canonical transforms. *IEEE Signal Process. Lett.* **16**(8), 727–730 (2009)
79. J.J. Healy, B.M. Hennelly, J.T. Sheridan, Additional sampling criterion for the linear canonical transform. *Opt. Lett.* **33**(22), 2599–2601 (2008)
80. J.J. Healy, J.T. Sheridan, Sampling and discretization of the linear canonical transform. *Signal Process.* **89**(4), 641–648 (2009)
81. A. Papoulis, *Signal Analysis* (McGraw-Hill, New York, 1977)
82. J.J. Healy, J.T. Sheridan, Cases where the linear canonical transform of a signal has compact support or is band-limited. *Opt. Lett.* **33**(3), 228–230 (2008)
83. A. Koç, H.M. Ozaktas, L. Hesselink, Fast and accurate computation of two-dimensional non-separable quadratic-phase integrals. *J. Opt. Soc. Am. A* **27**(6), 1288–1302 (2010)
84. F.S. Oktem, Signal representation and recovery under partial information, redundancy, and generalized finite extent constraints. M.S. thesis, Bilkent University, Turkey, 2009
85. J.J. Healy, J.T. Sheridan, Reevaluation of the direct method of calculating Fresnel and other linear canonical transforms. *Opt. Lett.* **35**(7), 947–949 (2010)
86. R.G. Campos, J. Figueroa, A fast algorithm for the linear canonical transform. *Signal Process.* **91**(6), 1444–1447 (2011)
87. A. Sahin, H.M. Ozaktas, D. Mendlovic, Optical implementations of two-dimensional fractional Fourier transforms and linear canonical transforms with arbitrary parameters. *Appl. Opt.* **37**, 2130–2141 (1998)
88. M. Moshinsky, C. Quesne, Linear canonical transformations and their unitary representations. *J. Math. Phys.* **12**(8), 1772–1780 (1971)
89. M. Nazarathy, J. Shamir, First-order optics—a canonical operator representation: lossless systems. *J. Opt. Soc. Am.* **72**(3), 356–364 (1982)
90. A. Sahin, H.M. Ozaktas, D. Mendlovic, Optical implementation of the two-dimensional fractional Fourier transform with different orders in the two dimensions. *Opt. Commun.* **120**, 134–138 (1995)
91. M.F. Erden, H.M. Ozaktas, A. Sahin, D. Mendlovic, Design of dynamically adjustable anamorphic fractional Fourier transformer. *Opt. Commun.* **136**(1–2), 52–60 (1997)
92. E.G. Abramochkin, V.G. Volostnikov, Generalized Gaussian beams. *J. Opt. A Pure Appl. Opt.* **6**, S157–S161 (2004)
93. L. Allen, M.W. Beijersbergen, R.K.C. Spreeuw, J.P. Woerdman, Orbital angular momentum of light and the transformation of Laguerre-Gaussian laser modes. *Phys. Rev. A* **45**(11), 8185–8189 (1992)
94. R. Pratesi, L. Ronchi, Generalized Gaussian beams in free space. *J. Opt. Soc. Am.* **67**(9), 1274–1276 (1977)
95. J.A. Rodrigo, T. Alieva, M.L. Calvo, Experimental implementation of the gyrator transform. *J. Opt. Soc. Am. A* **24**(10), 3135–3139 (2007)

96. J.A. Rodrigo, T. Alieva, M.L. Calvo, Gyration transform: properties and applications. *Opt. Express* **15**(5), 2190–2203 (2007)
97. J.A. Rodrigo, T. Alieva, M.L. Calvo, Applications of gyration transform for image processing. *Opt. Commun.* **278**(2), 279–284 (2007)
98. K.B. Wolf, T. Alieva, Rotation and gyration of finite two-dimensional modes. *J. Opt. Soc. Am. A* **25**(2), 365–370 (2008)
99. K.B. Wolf, *Geometric Optics on Phase Space* (Springer, Berlin, 2004)
100. G.B. Folland, *Harmonic Analysis in Phase Space* (Princeton University Press, Princeton, 1989)
101. J. Ding, S. Pei, C. Liu, Improved implementation algorithms of the two-dimensional nonseparable linear canonical transform. *J. Opt. Soc. Am. A* **29**(8), 1615–1624 (2012)
102. K.B. Wolf, Canonical transformations II. Complex radial transforms. *J. Math. Phys.* **15**(12), 2102–2111 (1974)
103. P. Kramer, M. Moshinsky, T.H. Seligman, Complex extensions of canonical transformations and quantum mechanics, in *Group Theory and Its Applications*, vol. 3, ed. by E.M. Loeb (Academic, New York, 1975), pp. 249–332
104. A.A. Malyutin, Complex-order fractional Fourier transforms in optical schemes with Gaussian apertures. *Quantum Electron.* **34**(10), 960–964 (2004)
105. A. Koç, H.M. Ozaktas, L. Hesselink, Fast and accurate algorithm for the computation of complex linear canonical transforms. *J. Opt. Soc. Am. A* **27**(9), 1896–1908 (2010)
106. C. Liu, D. Wang, J.J. Healy, B.M. Hennelly, J.T. Sheridan, M.K. Kim, Digital computation of the complex linear canonical transform. *J. Opt. Soc. Am. A* **28**(7), 1379–1386 (2011)
107. Y. Liu, Fast evaluation of canonical oscillatory integrals. *Appl. Math. Inf. Sci.* **6**(2), 245–251 (2012)