

# Digital Computation of Fractional Fourier and Linear Canonical Transforms and Sparse Image Representation

Aykut Koc\*, Haldun M. Ozaktas†, Burak Bartan‡, Erhan Gundogdu\* and Tolga Cukur†§,

\* ASELSAN Research Center, Ankara, Turkey

E-mail: aykutkoc@aselsan.com.tr Tel/Fax: +90-312-5926277

† Department of Electrical Engineering, Bilkent University, Ankara, Turkey

‡ Department of Electrical Engineering, Stanford University, CA, USA

§ National Magnetic Resonance Research Center (UMRAM), Bilkent University, Ankara, Turkey

**Abstract**—Fast and accurate digital computation of the fractional Fourier transform (FRT) and linear canonical transforms (LCT) are of utmost importance in order to deploy them in real world applications and systems. The algorithms in  $O(N \log N)$  to obtain the samples of the transform from the samples of the input function are presented for several different types of FRTs and LCTs, both in 1D and 2D forms. To apply them in image processing we consider the problem of obtaining sparse transform domains for images. Sparse recovery tries to reconstruct images that are sparse in a linear transform domain, from an underdetermined measurement set. The success of sparse recovery relies on the knowledge of domains in which compressible representations of the image can be obtained. In this work, we consider two- and three-dimensional images, and investigate the effects of the fractional Fourier (FRT) and linear canonical transforms (LCT) in obtaining sparser transform domains. For 2D images, we investigate direct transforming versus several patching strategies. For the 3D case, we consider biomedical images, and compare several different strategies such as taking 2D slices and optimizing for each slice and direct 3D transforming.

## I. INTRODUCTION

The fractional Fourier transform (FRT), [17], [14], [18], [19], is the generalization to the ordinary Fourier transform (FT). Linear canonical transforms (LCT), [24], [6], are linear integral transforms that generalizes the FRTs and several other transforms. Both are very important transforms in both optics and signal/image processing fields as well as quantum physics, [6]. As analogous to the fact that the fast Fourier transform (FFT) algorithm is very instrumental to use and make the well-known Fourier transform widespread, discretization and development of fast digital computation algorithms to deploy FRT and LCTs in real world applications effectively are needed. These fast algorithms begin with samples of the continuous input signal and compute samples of the continuous FRT or LCT output signal such that the continuous output can be interpolated from the computed output samples, [9], [7]. This is accomplished by decomposing the FRT or LCT operations into basic building blocks that already have fast algorithms. An application of FRT and LCTs in image processing is sparse image representation and compression. Sparse image representation and recovery needs the knowledge of transform domains

in which an image can be sparsely represented, [4]. This enables better compression of images. FRTs and LCTs can sparsify images to make them sparsely represented in FRT and LCT domains. Having fast digital algorithms is a requirement to use FRT and LCTs in sparse image representation since the procedure requires several computations of transforms. Both regular images and three dimensional complex-valued medical images are considered in presenting the use of FRT and LCTs in sparse image representation.

In this paper, first the preliminaries will be given, then fast digital computation of algorithms for FRTs, 1D LCTs and 2D LCTs will be presented. After presenting the application of these algorithms to the sparse image representation and compression, the paper will conclude.

## II. PRELIMINARIES

### A. The fractional Fourier transform

FRT is a generalization to the ordinary Fourier transform. The  $a$ th order fractional Fourier transform  $\{\mathcal{F}^a f\}(u)$  of the function  $f(u)$  is defined for  $0 < |a| < 2$  as

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

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.

### B. The linear canonical transform

The LCT of  $f(u)$  with parameter matrix  $\mathbf{M}$  is given by  $f_{\mathbf{M}}(u) = (\mathcal{C}_{\mathbf{M}} f)(u)$ :

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

where  $\alpha, \beta, \gamma$  are real parameters independent of  $u$  and  $u'$  and where  $\mathcal{C}_M$  is the LCT operator and  $\mathbf{M}$  is given by:

$$\mathbf{M} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \gamma/\beta & 1/\beta \\ -\beta + \alpha\gamma/\beta & \alpha/\beta \end{bmatrix}. \quad (3)$$

### III. DIGITAL COMPUTATION OF FRACTIONAL FOURIER TRANSFORM

The fractional Fourier transform can be decomposed into a cascade of chirp multiplication, chirp convolution, scaling, and ordinary Fourier transformation. The algorithm for efficient and accurate digital computation of FRTs depend on decomposing FRT into these simple building blocks, compute each of them in  $O(N \log N)$  time and obtain the result, [15].

To obtain the digital computation algorithm, Eqn. 1 is put in the form:

$$\{\mathcal{F}^a f\}(u) = A_\phi e^{i\pi\alpha u^2} \int_{-\infty}^{\infty} e^{-i2\pi\beta uu'} \left[ e^{i\pi\alpha u'^2} f(u') \right] du', \quad (4)$$

where  $\alpha = \cot \phi$  and  $\beta = \csc \phi$ . By using Shannon's interpolation formula and after some algebraic manipulations, from Eqn.4, the following form can be obtained, [15]:

$$\{\mathcal{F}^a f\} \left( \frac{m}{2\Delta u} \right) = \frac{A_\phi}{2\Delta u} e^{i\pi(\alpha-\beta)\left(\frac{m}{2\Delta u}\right)^2} \times \sum_{n=-N}^N e^{i\pi\beta\left(\frac{m-n}{2\Delta u}\right)^2} e^{i\pi(\alpha-\beta)\left(\frac{n}{2\Delta u}\right)^2} f \left( \frac{n}{2\Delta u} \right) \quad (5)$$

Observing Eqn.5, it can be recognized that the summation is the convolution of  $e^{i\pi\beta(n/2\Delta u)^2}$  and the chirp multiplied function  $f(\cdot)$ . The convolution can be computed in  $O(N \log N)$  time by using the standard FFT algorithm. The final output samples can then be obtained by a final chirp multiplication, resulting in the overall complexity of  $O(N \log N)$ .

### IV. DIGITAL COMPUTATION OF LINEAR CANONICAL TRANSFORMS

#### A. 1D Transforms

As in the case of FRT, we again employ the same approach for the digital computation of LCTs. The following decomposition involving the FRT, scaling operation, and chirp multiplication [16], [10]:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ -q & 1 \end{bmatrix} \begin{bmatrix} M & 0 \\ 0 & 1/M \end{bmatrix} \begin{bmatrix} \cos a\pi/2 & \sin a\pi/2 \\ -\sin a\pi/2 & \cos a\pi/2 \end{bmatrix}. \quad (6)$$

where  $a$  is the order of the FRT,  $q$  is the chirp multiplication parameter, and  $M$  is the scaling parameter. Since the fast method proposed in [15] can be used for fast computation of the FRT, this decomposition directly leads to a fast algorithm for an arbitrary LCTs if one recalls that scaling and chirp multiplication operations take only  $O(N)$  time.

However, it needs to be shown that the three parameters ( $a, q,$  and  $M$ ) are sufficient to satisfy the decomposition equality for an arbitrary ABCD matrix in order to have this decomposition is capable of representing an arbitrary LCT. To

do that, the right hand side of Eqn. 6 and the  $\mathbf{M}$  matrix entries in terms of  $\alpha, \beta, \gamma$  should be equal, which means:

$$\begin{bmatrix} \gamma/\beta & 1/\beta \\ -\beta + \alpha\gamma/\beta & \alpha/\beta \end{bmatrix} = \begin{bmatrix} M \cos \theta & M \sin \theta \\ -qM \cos \theta - \sin \theta/M & -qM \sin \theta + \cos \theta/M \end{bmatrix}. \quad (7)$$

After some algebra,  $a, q, M$  can be solved as:

$$a = (2/\pi) \cot^{-1} \gamma, \quad (8)$$

$$M = \begin{cases} \sqrt{1 + \gamma^2}/\beta, & \gamma \geq 0, \\ -\sqrt{1 + \gamma^2}/\beta, & \gamma < 0, \end{cases} \quad (9)$$

$$q = \gamma\beta^2/(1 + \gamma^2) - \alpha. \quad (10)$$

In this algorithm, the first operation is a FRT. The presently discussed fast LCT algorithm employs that fast FRT algorithm as a subroutine. The second operation in this method is scaling, which only involves a reinterpretation of the same samples with a scaled sampling interval. The last operation is CM which takes  $O(N)$  time, leading to an overall complexity of  $O(N \log N)$ .

#### B. 2D Transforms

The fast computation of 2D FRTs is straightforward since the separability of 2D FRT kernels. Given a fast algorithm for 1D FRT, one can easily obtain a 2D fast algorithm by first taking 1D FRTs along horizontal direction and then taking 1D FRTs along vertical direction over the result of the horizontal pass. On the case of LCTs, the above procedure can still be used if the 2D LCT is of separable form. However, the non-separable case needs special attention and deriving a fast algorithm for its digital computation is rather involved, [11]. Two-dimensional *non-separable* LCTs (2D-NS-LCT) with parameter matrix  $\mathbf{M}$ , of an input function  $f(\mathbf{u})$ , can be expressed as [5], [2]

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

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  $2 \times 2$  submatrices defining the transformation matrix  $\mathbf{M}$  of the system that represents the 2D-LCT. The matrix  $\mathbf{M} = [\mathbf{A} \ \mathbf{B}; \ \mathbf{C} \ \mathbf{D}]$  is real and symplectic. Further details on the definition can be found in [11], [9].

Derivation of a fast algorithm for digital computation of 2D-NS-LCTs starts with an alternative representation of LCTs based on Iwasawa decomposition, [2], [24]. In [2], the first matrix of the Iwasawa decomposition is decomposed further into a 2D separable fractional Fourier transform that is sandwiched between two coordinate rotators. By using the 2D Iwasawa-type decomposition of [2], it is possible to derive an efficient algorithm for the computation of 2D NS LCTs [11].

An arbitrary transformation matrix  $\mathbf{M}$  can be decomposed in the following Iwasawa form [24], [2]:

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{G} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ -\mathbf{Y} & \mathbf{X} \end{bmatrix}, \quad (11)$$

where

$$\mathbf{G} = -(\mathbf{CA}^T + \mathbf{DB}^T)(\mathbf{AA}^T + \mathbf{BB}^T)^{-1}, \quad (12)$$

$$\mathbf{S} = (\mathbf{AA}^T + \mathbf{BB}^T)^{1/2}, \quad (13)$$

$$\mathbf{X} = (\mathbf{AA}^T + \mathbf{BB}^T)^{-1/2}\mathbf{A}, \quad (14)$$

$$\mathbf{Y} = (\mathbf{AA}^T + \mathbf{BB}^T)^{-1/2}\mathbf{B}. \quad (15)$$

In this decomposition, the first operation is an orthosymplectic system, the second one is a scaling (magnification) system, and finally the last one is a two-dimensional chirp multiplication (2D CM). The first stage can be further decomposed into a two-dimensional *separable* fractional Fourier transform (2D-S-FRT) that is sandwiched between two coordinate rotators [2]:

$$\begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ -\mathbf{Y} & \mathbf{X} \end{bmatrix} = \mathbf{R}_{r_2} \mathbf{F}_{a_x, a_y} \mathbf{R}_{r_1}, \quad (16)$$

where the  $4 \times 4$  matrices  $\mathbf{R}_{r_1}$  and  $\mathbf{R}_{r_2}$  are rotation matrices of angles  $r_1$  and  $r_2$ , respectively.  $\mathbf{F}_{a_x, a_y}$  is a 2D separable FRT matrix. Since 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. In summary, the first stage of the 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  $O(N \log N)$  time.

The second stage is the scaling operation. The analogue of the 1D scalar scaling parameter in the 2D case is the matrix  $\mathbf{S}$ . The matrix  $\mathbf{S}$  can be used to determine the output samples by using the input-output relation of the scaling operation:

$$f_{sc}(\mathbf{u}) = \frac{1}{\sqrt{\det \mathbf{S}}} f(\mathbf{S}^{-1}\mathbf{u}) \quad (17)$$

where  $f$  is the function to be scaled and  $f_{sc}$  is the scaled function.

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, it does not impose a computational load bigger than  $O(N \log N)$ .

The last stage is the 2D CM operation whose parameters are given by the matrix  $\mathbf{G}$  as defined in Eqn. 12. The input-output relation of this 2D-CM is given as:

$$f_{ch}(\mathbf{u}) = e^{-i\pi(G_{11}u_x^2 + (G_{12} + G_{21})u_x u_y + G_{22}u_y^2)} f(\mathbf{u}) \quad (18)$$

where  $f_{ch}$  stands for the chirp-multiplied function. This CM operation can be non-separable or separable for the 2D-NS-LCTs under consideration. In both cases, it only requires one multiplication for each sample, resulting in  $O(N)$  computational complexity.

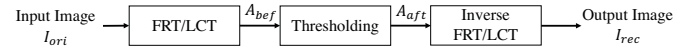


Fig. 1. The following stages are performed sequentially during the sparse recovery: FRT or LCT transformations, thresholding of varying percentages of transform-domain coefficients, inverse transformation, [8].

Since all the stages can be computed in  $O(N \log N)$  time, the overall transform can also be computed in  $O(N \log N)$  time where  $N$  stands for the total number of samples (pixel values) in a 2D image.

## V. APPLICATION TO SPARSE IMAGE REPRESENTATION

Digital image representation and compression receive tremendous attention due to storage and bandwidth concerns, use of which have been drastically increasing. Image compression, in a general context, is a very involved area with an enormous academic and industrial literature and a full-scale compression framework includes representation, quantization, encoding and decoding parts [20], [25], [22], [21], [1], [13]. The scope of this work does not cover quantization and coding of the compressed data, which usually exists in image compression standards. Here, we focus on representing the image signal in several FRT/LCT transform domains that would lead to sparser image representations. A transformation of the image does not by itself compress it. Rather it sparsifies the image, i.e., concentrates its energy in a more compact support in the transformed domain. This, in return, becomes a crucial preprocessing part since sparser representations lead to higher quality decompressed images at given compression ratios. It is, therefore, of great interest to study the transform that gives the sparsest representation of an image in some domain. Wavelet and Fourier related transforms are well-established in In [3] and [12], FRTs have been utilized to some extent for sparsification. However, these works have not systematically elaborated the effects of these transforms in sparsifying and increasing compression performance and are far from a thorough and complete analysis.

To study the sparsifying properties of FRTs and LCTs, the designed experiment is depicted in Figure 1. As shown in Figure 1, original images are first transformed by FRT or LCT. In the second stage, the subset of the transform-domain coefficients that are smaller than a predetermined threshold is set to zero. The inverse transform is performed on the thresholded coefficients to recover the image as a final step. The threshold selection reflects a desired rate of undersampling denoted here as CR:

$$CR = \frac{\|V_{bef}\|_0}{\|V_{aft}\|_0}, \quad (19)$$

where  $\|x\|_0$  denotes the  $l_0$ -norm of vector  $x$ , defined as the number of nonzero elements in  $x$ . Note that  $V_{bef} = [A_{bef,1}^T A_{bef,2}^T \dots A_{bef,N}^T]^T$  and  $V_{aft} = [A_{aft,1}^T A_{aft,2}^T \dots A_{aft,N}^T]^T$ , where  $A_{bef,i}$  and  $A_{aft,i}$  are the  $i^{th}$  columns of  $A_{bef}$  and  $A_{aft}$ , which correspond to the transform of the ideal reference image  $I_{ori} \in \mathbb{R}^{M \times N}$  before

and after the thresholding, respectively. So,  $A = \mathcal{T}\{I_{ori}\}$ , where  $\mathcal{T}$  is the transform that we are using. CR corresponds to the ratio of nonzero elements in the transformed data before and after the thresholding.

To study the behaviour of the proposed method and to compare it with other methods, a range of values for compression ratio (between 1 and 100) is spanned. Standard Discrete Cosine transform (DCT) and Wavelet based methods with 9/7 and 5/3 filters are used as baseline reference methods to compare against. Two comparative image quality metrics, the mean squared error (MSE) and structural similarity index (SSIM), are used. MSE is defined as:

$$MSE = \frac{\sum_{j=1}^M \sum_{i=1}^N (I_{rec}(i, j) - I_{ori}(i, j))^2}{\sum_{j=1}^M \sum_{i=1}^N (I_{ori}(i, j))^2} \times 100, \quad (20)$$

where  $I_{ori} \in \mathbb{R}^{M \times N}$  is the ideal reference image,  $I_{rec} \in \mathbb{R}^{M \times N}$  is the recovered image, and  $I_{ori}(i, j)$ ,  $I_{rec}(i, j)$  are the image pixels. An average SSIM metric was calculated across the whole image as follows [23]:

$$SSIM = \frac{1}{XY} \sum_{x=1}^X \sum_{y=1}^Y \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (21)$$

where  $x$  and  $y$  are image patches extracted from the recovered and ideal images, respectively.  $\mu$  and  $\sigma$  are the mean and covariance of pixel intensity within the given patches. The remaining parameters were selected as  $c_1 = 10^{-4}$ ,  $c_2 = 9 \times 10^{-4}$ , a Gaussian kernel of width 10, and a standard deviation of 5. The visual results of decompressed output images for compression ratio of 40 can also be seen in Figure 2 for the *lake* image. Figure 3 demonstrates how the SSIM index and the MSE change with respect to different compression ratios for our proposed transforms and for the baseline methods.

For fractional transforms and linear canonical transform, the optimal transform parameters (order  $a$  for FRT;  $\alpha$ ,  $\beta$ , and  $\gamma$  for LCTs) are determined by using simple parameter sweeping. The results for the parameter with the best output quality determined by using SSIM indices are shown in the plots. Here, the aim is to study the performance of the transforms in sparsifying the data without causing much distortion to the image. The parameter range is divided in 10 equal parts, the best performing parameter is found and then a finer sweep in the neighborhood of this quasi-optimal value with 10 more sub-steps is performed. Finally the optimal transform parameter is obtained. For the baseline methods of Haar wavelet transform, the block size is altered with the values of 8, 16, and 32 and the best one is used. For the baseline method of wavelet transforms with different levels, the level at which the transform reaches its maximum SSIM index is used.

Wavelet transform with filter 5/3 leads to higher SSIM indices for CR values up to 50. For CR values larger than

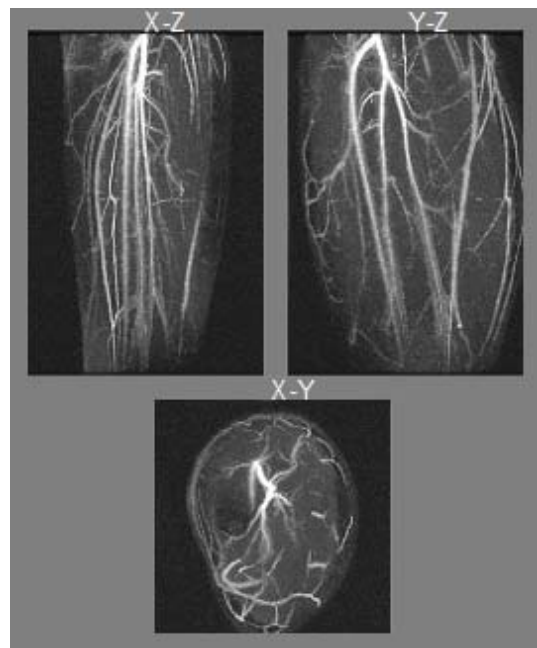


Fig. 4. Three-dimensional image of lower-leg along different cross sections, [8]

50, we see that the performances of FRT and LCT catch up with the wavelet transform with filter 5/3. Moreover, these three transforms seem to overperform DCT, in the sense of SSIM index, for a wide range of CRs when all of them are applied to the whole image.

A comparison between FRT and LCT shows that their performances are almost the same. Due to this reason, it would be preferable to use FRT instead of LCT since one needs to optimize three parameters for LCTs whereas only one parameter needs to be optimized for the FRT.

As another application, FRT and LCTs are applied as sparsifying transforms to three-dimensional (3D) medical data. The 3D images are frequently used in medical imaging and the one used in these experiments is shown in Figure 4. It is an image of the lower leg section of human body. The image is of dimension 128x128x192. The image is complex (pixel values are complex numbers), so it can be shown that the FRT and LCTs can also be used as sparsifying transforms for complex signals.

Two methods are used to study the three dimensional sample image. In the first method, the three-dimensional image is regarded as a concatenated set of two-dimensional images. Its each layers are considered separately in finding the optimal fractional order and compressed/reconstructed independently. The second method takes the three-dimensional image as a whole and treats it by taking a three-dimensional transform. In both of these methods, the fast algorithms for 2D FRTs and 2D LCTs as well as 1D versions, which are explained in the previous Sections are used. The performances of these two methods when FRT or LCT is deployed are shown in Figure 5. In Figure 5, FRT (A) and LCT (A) correspond to the cases



(a)



(b)



(c)



(d)



(e)



(f)

Fig. 2. Decompressed images after being compressed with CR = 40 using (a) Haar transform (16x16 blocks), (b) FRT (to whole image), (c) LCT (to whole image), (d) DCT (to whole image), (e) Wavelet 9/7 (to whole image), (f) Wavelet 5/3 (to whole image), [8]

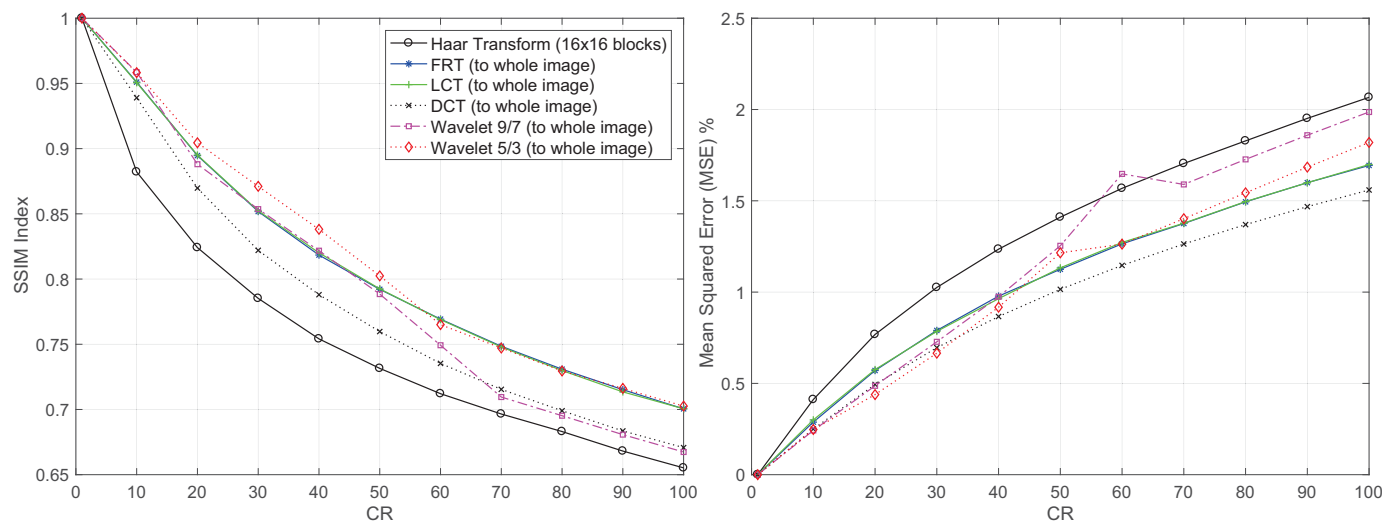


Fig. 3. SSIM index and MSE vs CR for lake image, [8]

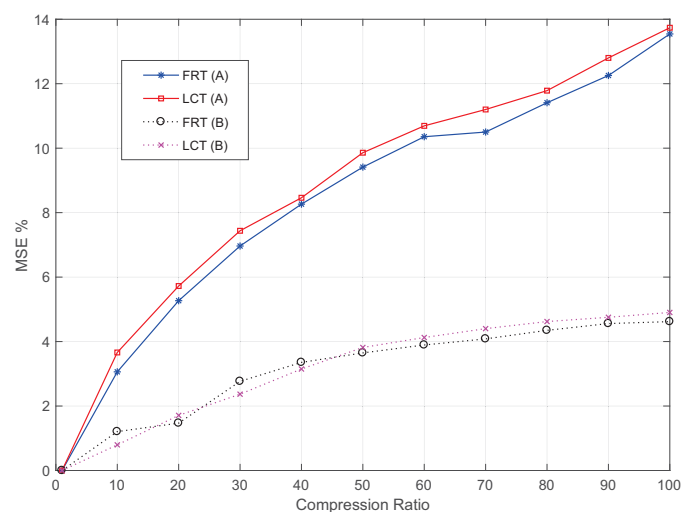


Fig. 5. MSE vs CR for the three-dimensional lower-leg image, [8]

where Method 1 is used. Likewise, FRT (B) and LCT (B) means that these transforms are applied to the image using the Method 2. Figure 5 presents that LCT and FRT perform with similar performance. However, Method 2 produces better quality outputs for three-dimensional images with considerable superiority to Method 1.

### VI. CONCLUSIONS

This work focuses on experimenting with FRTs and LCTs to achieve the sparsest representation of image data. To do this, one needs to have accurate and efficient digital computation algorithms for their computation. FRTs and LCTs have been shown to be useful in obtaining sparser representations for images. Simulations have shown that for higher compression ratios, the difference between the performances of the transforms becomes more noticeable. Fractional Fourier and linear canonical transforms are shown to yield better results. Regarding three-dimensional images, we found that it is much

better to treat all dimensions uniformly and work with three-dimensional transforms, as opposed to treating them as multiple two-dimensional layers. This was far from an obvious conclusion since the layered method allows the optimization of the parameters for each layer independently.

### ACKNOWLEDGMENT

The work of T. Çukur was supported in part by a Marie Curie Actions Career Integration Grant (PCIG13-GA-2013-618101), by a European Molecular Biology Organization Installation Grant (IG 3028), and by a TUBA GEBIP fellowship. H. M. Ozaktas acknowledges partial support of the Turkish Academy of Sciences.

### REFERENCES

- [1] *ISO/IEC 15444-1 — ITU-T Rec. T.800, Information Technology - JPEG 2000 Image Coding System: Core Coding System*. 2002.
- [2] T. Alieva and M. J. Bastiaans. Alternative representation of the linear canonical integral transform. *Opt. Lett.*, 30(24):3302–3304, 2005.
- [3] B. Divya and K. Prabha. 2d-discrete frac. fourier transform for facial image compression. *Int. J. of Invent. in Comp. Sci. and Eng.*, 2:2348–3431, 2015.
- [4] D. L. Donoho, M. Elad, and V. N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Trans. Inform. Theory*, 52(1):6–18, Jan. 2006.
- [5] G. B. Folland. *Harmonic Analysis in Phase Space*. Princeton University Press, 1989.
- [6] J. J. Healy, M. A. Kutay, H. M. Ozaktas, and J. T. S. eds. *Linear Canonical Transforms: Theory and Applications*. Springer New York, New York, NY, 2016.
- [7] A. Koc, H. M. Ozaktas, and L. Hesselink. Fast and accurate algorithms for quadratic phase integrals in optics and signal processing. *Three-Dimensional Imaging, Visualization, and Display: SPIE Proc Orlando, FL*, 8043:04–1–04–5, 2011.
- [8] A. Koç, B. Bartan, E. Gundogdu, T. Çukur, and H. M. Ozaktas. Sparse representation of two- and three-dimensional images with fractional fourier, hartley, linear canonical, and haar wavelet transforms. *Expert Systems with Applications*, 77:247 – 255, 2017.
- [9] A. Koç, F. S. Oktem, H. M. Ozaktas, and M. A. Kutay. *Linear Canonical Transforms: Theory and Applications*, chapter Fast Algorithms for Digital Computation of Linear Canonical Transforms, pages 293–327. Springer New York, New York, NY, 2016.
- [10] A. Koç, H. M. Ozaktas, C. Candan, and M. A. Kutay. Digital computation of linear canonical transforms. *IEEE Trans. Sig. Proc.*, 56:2383–2394, 2008.

- [11] A. Koç, H. M. Ozaktas, and L. Hesselink. Fast and accurate computation of two-dimensional non-separable quadratic-phase integrals. *J. Opt. Soc. Am. A*, 27(6):1288–1302, 2010.
- [12] R. Kumar, K. Singh, and R. Khanna. Satellite image compression using fractional fourier transform. *International Journal of Computer Applications*, 50(3):0975–8887, 2012.
- [13] R. Leung and D. Taubman. Transform and embedded coding techniques for maximum efficiency and random accessibility in 3-d scalable compression. *IEEE Trans. Image Process.*, 14:1632–1646, 2005.
- [14] D. Mendlovic and H. M. Ozaktas. Fractional fourier transforms and their optical implementation: I. *J. Opt. Soc. Am. A*, 10:1875–1881, 1993.
- [15] H. M. Ozaktas, O. Arikan, M. A. Kutay, and G. Bozdağı. Digital computation of the fractional fourier transform. *IEEE Transactions on Signal Processing*, 44:2141–2150, 1996.
- [16] H. M. Ozaktas, A. Koç, I. Sari, and M. A. Kutay. Efficient computation of quadratic-phase integrals in optics. *Opt. Lett.*, 31:35–37, 2006.
- [17] H. M. Ozaktas and D. Mendlovic. Fourier transforms of fractional order and their optical interpretation. *Optics Commun.*, 101(3-4):163–169, 1993.
- [18] H. M. Ozaktas and D. Mendlovic. Fractional fourier transforms and their optical implementation: II. *J. Opt. Soc. Am. A*, 10:2522–2531, 1993.
- [19] H. M. Ozaktas, Z. Zalevsky, and M. A. Kutay. *The Fractional Fourier Transform with Applications in Optics and Signal Processing*. New York: Wiley, 2001.
- [20] W. B. Pennebaker and J. L. Mitchell. *JPEG Still Image Data Compression Standard*. Kluwer, 1992.
- [21] X. Song, X. Peng, J. Xu, G. Shi, and F. Wu. Cloud-based distributed image coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 25(12):1926–1940, Dec 2015.
- [22] T. Suzuki and M. Ikehara. Integer dct based on direct-lifting of dct-idct for lossless-to-lossy image coding. *Image Processing, IEEE Transactions on*, 19(11):2958–2965, Nov 2010.
- [23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process*, 13(4):600–612, 2004.
- [24] K. B. Wolf. *Geometric Optics on Phase Space*. Springer, 2004.
- [25] S. Zhu and J. Chen. Performance comparison of several transform domain image coding algorithms. In *Computer Science and Information Processing (CSIP), 2012 Int. Conf. on*, pages 1030–1034, Aug 2012.