Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 1946 – 1962

doi:10.3906/elk-1505-241

Image compression algorithm with reduced blocking artifacts

Kritika MITTAL, Kulbir SINGH, Neeru JINDAL*
Department of Electronics and Communication Engineering, Thapar University, Patiala, India

Received: 28.05.2015 • **Accepted/Published Online:** 17.07.2016 • **Final Version:** 29.05.2017

Abstract: The modern communication era has led to a proliferation of digital media contents. However, the large volume of data poses difficulties because of increased bandwidth and limited storage space. Hence, this has led to the need for compression techniques. Image compression with block processing allows the coder to adapt to local image statistics and exploit the correlation present among neighboring image pixels. The main degradation factor of block transform coding is blocking artifacts (visually undesirable patterns) at high compression ratios. The degradation occurs because of coarse quantization of the transform coefficients and the independent processing of the blocks. In this paper, the novelty of the algorithm is its ability to detect and reduce the blocking artifacts using nonseparable discrete fractional Fourier transform (NSDFrFT) at high compression ratios. Three transform techniques, namely nearest neighbor interpolation, bilinear interpolation, and bicubic interpolation, were implemented. The NSDFrFT-bicubic interpolation resulted in a structurally similar high subjective quality reconstructed image with reduced blocking (for low frequency images) at high compression ratios. Simulation results are calculated with many image quality metrics such as peak signal to noise ratio, mean square error, structural similarity index, and gradient magnitude similarity measure. Evaluations, such as comparisons between the proposed and existing algorithms (DFrFT, FFT), are presented with relevant tables, graphs, and figures.

Key words: Image compression, interpolation methods, discrete fractional Fourier transform, nonseparable discrete fractional Fourier transform, compression ratios

1. Introduction

Image processing has become an important area of research as nowadays a lot of data are represented in graphics. Digitized images require a large number of coefficients to measure the energy in the frequency domain. Thus, storage space availability, limited transmission bandwidth, and processing cost are some of the substantial issues that need to be handled by image processing. As a result, compression of the image is required to counter these problems [1–5] while preserving the visual quality of the image at reduced costs.

The image compression algorithms proposed in past decades [6] utilize spatial redundancy and irrelevant information found in the image file for compressing a picture with preserved visual quality [7]. Image compression is used in satellite processing, medical imaging, remote sensing, and the preservation of works of art, among other things. When an obtained compressed image is identical to the original image, compression is defined as lossless compression or reversible compression [8]. However, only a minimal amount of compression can be achieved. Thus, lossy or irreversible compression is often used to achieve a greater extent of compression. Lossy

*Correspondence: neeru.jindal@thapar.edu

image compression or irreversible compression discards irrelevant information but causes significant artifacts that hamper the quality of the image [9].

However, these artifacts, known as blocking artifacts [10], are quite prominent at higher compression ratios. Blocking occurs when an image undergoes transform coding. Correlations among the neighboring block boundaries in block-based transform techniques are not taken into consideration. As a result, the adjacent blocks' boundaries become visible, causing blocking artifacts while reconstructing the decoded image. Thus, the visual quality of an image can be increased by reducing these blocking artifacts. As a result, detecting and reducing blocking artifacts in reconstructed images during the compression process is important. Several image compression algorithms based on transform coding are available in the literature including JPEG2000 compression coding [11], the discrete Fourier transform (DFT), the discrete cosine transform (DCT) [12], discrete fractional Fourier transform (DFrFT) [13,14], discrete fractional cosine transform (DFrCT) [15], and many more. Many postprocessing algorithms have been developed for the reduction of blocking artifacts, but these algorithms are often computationally complex, include multiple iterations, or result in excessive smoothing of the image textures. Thus, in this paper, the nonseparable discrete fractional Fourier transform technique (NSDFrFT) is proposed as a way to detect and reduce blocking artifacts with less computational burden and better quality metrics.

The remaining portions of the paper are organized as follows: Section 2 will describe the review and mathematical analysis of the DFrFT, the NSDFrFT, interpolation, image quality metrics, and blocking artifacts. Section 3 will introduce the implementation of the image compression algorithm. Simulation results will be given in Section 4 in order to evaluate the performance of the proposed algorithm with quality metrics and to compare the algorithm with existing techniques. Finally, conclusions are drawn in Section 5.

2. Preliminaries

The Fourier transform (FT) has applications in almost every domain: signal analysis, optics, image, physics, statistics, acoustics, and antenna/array processing [12,16,17]. However, the FT proved to be inadequate for nonstationary signals and this has led to the emergence of the fractional Fourier transform (FrFT). Namias [18] introduced the definition of FrFT, which is the generalized definition for FT as a transform technique. Ozaktus and Mendlovic [19] presented numerous definitions of FrFT equivalent to each other. For various types of signals (one-dimensional, multidimensional, periodic, aperiodic, discrete, and continuous), Cariolario et al. [20] assigned a definition for FrFT. Pei et al. and Pei and Yeh [21,22] studied FrFT in detail by considering many of its different aspects. The expression for two-dimensional FrFT is given as [22]:

$$F^{a}\left(g\left(x,v\right)\right) = F^{a_x,a_v}\left(f\left(x,v\right)\right) \tag{1}$$

$$= \iint k^{\alpha_x,\alpha_v}(x.v;\, x^{'},\, v^{'}) f(x^{'},\, v^{'}) dx\, dv \tag{2}$$

$g\left(x,v\right) = xx^{\wedge} + vv^{\wedge}$ and $a = a_x x^{\wedge} + a_v v^{\wedge}$, where $x^{\wedge}$ and $v^{\wedge}$ are the unit vectors in the u and v directions, respectively. Figure 1 depicts the time–frequency plane representation of FrFT. The era of computers led to the development of DFT and its ability to digitally compute FT. Thus, working in the same direction, the digital version of FrFT was needed to process signals in discrete form. The two-dimensional definition of FrFT can be categorized into separable (DFrFT) and nonseparable (NSDFrFT) transform types. The separable transform is a special form of the nonseparable transform.

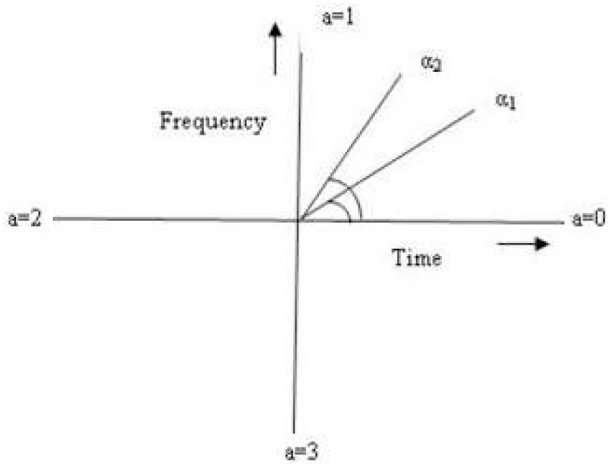

Figure 1. FrFT in the time–frequency plane.

2.1. Discrete fractional Fourier transform

The advent of the technological usage of computers and DSP processors increased and motivated Santhanam et al. [23] to define DFrFT as the linear combination of the Lagrange interpolation polynomial of degree 3. However, in time, many other definitions of DFrFT [24,25] were also created, but none of them was able to satisfy all of the properties of continuous FrFT [21].

The definition of DFrFT can be categorized into a sampling-based method, a linear combination method, an eigenvector-based method, and a weighted summation-based method [26]. The simplest way to achieve DFrFT is to sample continuous FrFT, but it will lose many more properties of FrFT [27]. The eigenvalues and eigenvectors of the DFT are computed for the eigen-based method. Subsequently, the Hermite function is used to calculate the fractional power of the DFT matrix [28]. The transform kernel used to calculate DFrFT is given as:

$$F^{2\alpha/\pi} = U^{\wedge} D^{2\alpha/\pi} U^{\wedge T} \tag{3}$$

where $U^{\wedge} = [u_0^{\wedge} | u_1^{\wedge}| \dots u_{N-1}^{\wedge}]$ for odd N and $U^{\wedge} = [u_0^{\wedge} | u_1^{\wedge}| \dots u_{N-2}^{\wedge} | u_N^{\wedge}]$ for even N. These are the normalized eigenvectors of the kth order discrete Hermite function. D is defined for odd and even N as shown in Eq. (4).

$$D^{2\alpha/\pi} = \begin{bmatrix} e^{-j0} & & & 0 \\ & e^{-j\alpha} & & \\ & & \ddots & \\ 0 & & & e^{-j\alpha(N-1)} \end{bmatrix},$$

$$D^{2\alpha/\pi} = \begin{bmatrix} e^{-j0} & & & 0 \\ & e^{-j\alpha(N-2)} & & \\ & & \ddots & \\ 0 & & & e^{-j\alpha N} \end{bmatrix} \tag{4}$$

Pei and Hsue [29] defined DFrFT using random DFT eigenvectors and eigenvalues. The weight-based method computed the DFrFT from the weighted summation of DFrFT at special angles with an odd point length [30].

DFrFT has applications in signal processing, image processing, tomography, cryptography, optical imaging, and computer modeling for image propagation [31,32].

2.2. Nonseparable discrete fractional Fourier transform

NSFrFT, a generalized definition of FrFT, can be obtained by substituting a 2×2 matrix (A, B, C, and D) in the definition of nonseparable linear canonical transform (NSLCT) [33].

$$A = \begin{bmatrix} \cos\alpha_1 & \cos\alpha_2 \\ \cos\alpha_3 & \cos\alpha_4 \end{bmatrix} B = \begin{bmatrix} \sin\alpha_1 & \sin\alpha_2 \\ \sin\alpha_3 & \cos\alpha_4 \end{bmatrix}, \tag{5}$$

$$C = \begin{bmatrix} -sin\alpha_1 & -\sin\alpha_2 \\ -\sin\alpha_3 & -\sin\alpha_4 \end{bmatrix}, \; D = \begin{bmatrix} \cos\alpha_1 & \cos\alpha_2 \\ \cos\alpha_3 & \cos\alpha_4 \end{bmatrix} \tag{6}$$

Consider that $\det(\mathrm{B}) \neq 0$ and

$$K^{(A,B,C,D)}(u,v) = (2\pi\sqrt{-\det(B)})^{-1} \times \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp[j(k_1u^2 + k_2uv + k_3v^2)/2\det(B)] \times \tag{7}$$

$$\exp[\frac{j((-b_{22}u + b_{12}v) + (-b_{21}u - b_{11}u)y)}{\det(B)}] \times$$

$$\exp[\frac{j(p_1x^2 + p_2xy + p_3y^2)}{2\det(B)}] g(x,y)\,dxdy$$

$K^{(A,B,C,D)}(u,v)$ is the output obtained and

$$k_1 = d_{11}b_{22} - d_{12}b_{21} \tag{8}$$

$$k_2 = 2(-d_{11}b_{12} + d_{12}b_{11}) \tag{9}$$

$$k_3 = -d_{21}b_{12} + d_{22}b_{11}$$

$$p_1 = a_{11}b_{22} - a_{21}b_{12}$$

$$p_2 = 2(a_{12}b_{22} - a_{22}b_{12}),$$

$$p_3 = -a_{12}b_{21} + a_{22}b_{11}$$

Eq. (9) should satisfy the following constraints:

$$A^TC = C^TA\,,\; B^TD = D^TB \,and\, A^TD - C^TB = 1 \tag{10}$$

Sahin et al. [34] suggested a definition for nonseparable FrFT and nonseparable DFrFT. This definition utilizes the concept of interpolation to obtain $f\left[\frac{(cos\emptyset_1 x + sin\emptyset_1 y)}{\cos(\emptyset_1 - \emptyset_2)}, \frac{(-sin\emptyset_2 x + cos\emptyset_2 y)}{\cos(\emptyset_1 - \emptyset_2)}\right]$ from $f(x,y)$. The mapping is done to attain the rotation of the x-axis and y-axis to arbitrary axis x' and y' by $\emptyset_1$ and $\emptyset_2$, respectively, with an order of a_1 and a_2. Nonseparable DFrFT has four deciding parameters: a_1, a_2, $\emptyset_1$, and $\emptyset_2$. Sahin et al. also suggested the use of bilinear interpolation to achieve the mapping. Thus, interpolation is an important aspect in this definition of nonseparable DFrFT. Figure 2 shows the time–frequency plane rotation for NSFrFT.

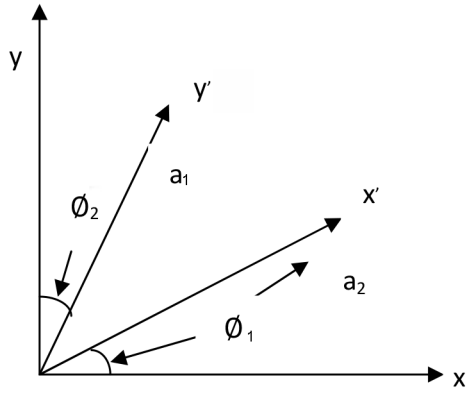

Figure 2. Time–frequency plane rotation for NSFrFT.

2.3. Interpolation

Interpolation is defined as the process of forming new intermediate data points within the range of known data points. Interpolation increases the sampling rate of the sampled digital signal. The basic interpolation methods are: bilinear interpolation (Bil Intr.), bicubic interpolation (Bic Intr.), and nearest neighbor interpolation (NN Intr.).

To calculate the interpolated point in 2D, four immediate neighbor points are used in Bil Intr. utilizing linear (1D) interpolation [35]. Suppose $\mathrm{P}(xy)$ is a point on the image to be interpolated, and $f(r_{11})$, $f(r_{12})$, $f(r_{21})$, and $f(r_{22})$ are the pixel values at the immediate neighbor pixel of the pixel/point to be interpolated, respectively. They are given as follows: $r_{11}(x_1y_1)$, $r_{12}(x_1y_2)$, $r_{21}(x_2y_1)$, and $r_{22}(x_2y_2)$.

The Bil Intr. pixel is given as:

$$\begin{aligned} f(x,y) &= \frac{f(r_{11})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y_2-y) + \frac{f(r_{12})}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y-y_1) \\ &+ \frac{f(r_{21})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y_2-y) + \frac{f(r_{22})}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y-y_1) \end{aligned} \tag{11}$$

NN Intr. is the simplest interpolation requiring the least complex calculations. It considers only consecutive neighboring points following a point shift algorithm [35] for calculation purposes. Mathematically, the point P to be interpolated can be computed for both horizontal and vertical directions as:

$$P(x,y) = Q(x+0.5, y+0.5) \tag{12}$$

where $Q(rj)$, $Q(r+1,j)$, $Q(rj+1)$, and $Q(r+1,j+1)$ are the input pixels of the image.

Bic Intr. uses 16 points to interpolate a point/pixel, giving smoother images with fewer interpolation artifacts [35]. Figure 3 shows $f(i'j')$ as the point to be interpolated and $f(ij)$ as the point in the original image. The point that needs to be interpolated by Bic Intr. is given as:

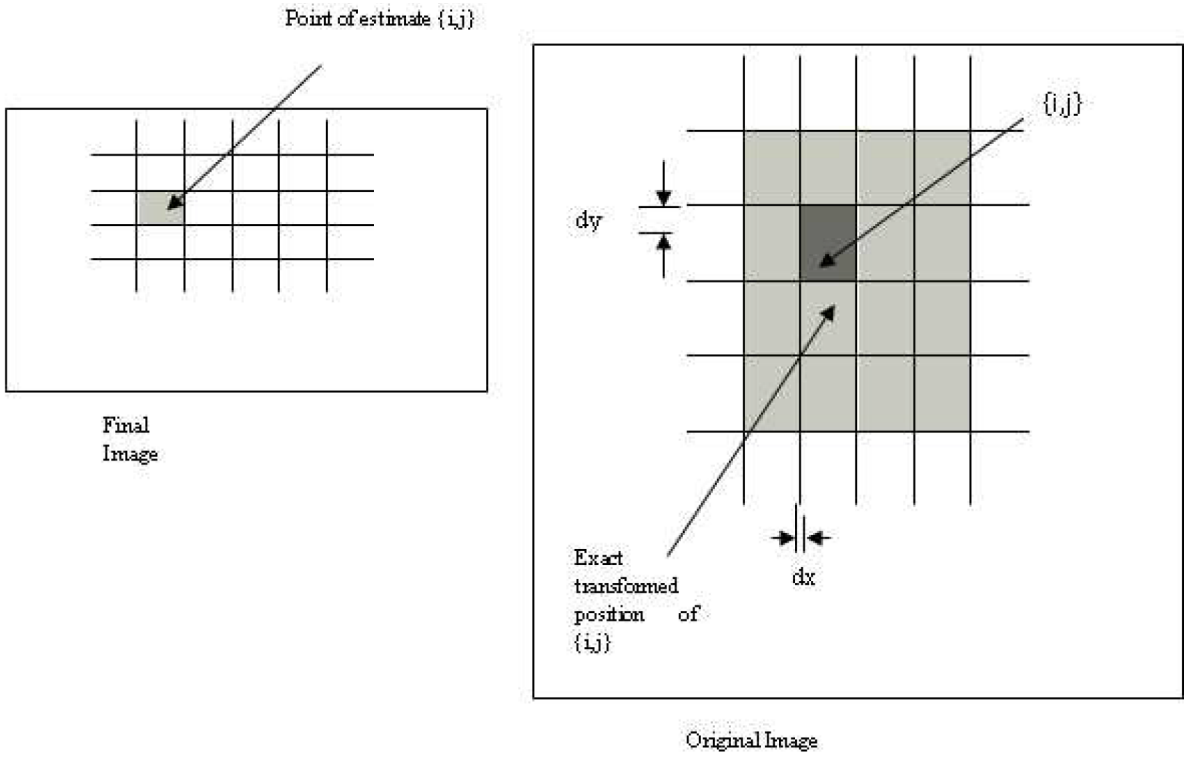

Figure 3. Bicubic interpolation of an image [35].

$$f\left[i^{'},j^{'}\right]=\sum_{l=-1}^{2}\sum_{m=-1}^{2}f\left(i+l,j+m\right)\times K\left(l-dx\right)K\left(dy-m\right) \tag{13}$$

where

$$K\left(x\right)=\frac{1}{6}[J\left(x+2\right)^{3}-4J\left(x+1\right)^{3}+6J\left(x\right)^{3}-4J\left(x-1\right)^{3}]$$

and

$$J\left(x\right)=\begin{cases} x & x>0 \\ 0 & x\leq 0 \end{cases}$$

2.4. Image quality metrics

Image quality measures (IQMs) are the parameters that determine the reconstructed image quality. IQMs can be classified in two categories: subjective quality measurements (SQMs) and objective quality measurements (OQMs). Visual scene perception, influenced by the viewing environment, spatial fidelity, the observer's state of mind, and the extent to which the observer interacts with the visual scene, is included in SQMs [36]. An algorithm-based comparison of the original and reconstructed image is included in OQMs. Between the two, OQMs are the more reliable and widely used IQMs to measure the quality of a reconstructed image. Various OQMs utilized are peak signal to noise ratio (PSNR), mean square error (MSE) [36], structural similarity index measure (SSIM), [37] and gradient magnitude similarity measure (GMSD) [38]. The formula for MSE is given

as [36]:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I\left(i,j\right) - K\left(i,j\right)\right]^2 \tag{14}$$

where I and K are the images to be compared. The formula for PSNR is given as [36]:

$$PSNR = 10.log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{15}$$

where MAX_I is the maximum pixel value. The SSIM index compares the luminance, contrast, and structure of the two images. However, these components are independent of each other. The mathematical representation of the SSIM is as follows:

$$SSIM\left(x,y\right) = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{16}$$

where μ_x and μ_y are the mean intensity, σ_x and σ_y are the contrast, and C_1 and C_2 are constants [37]. The overall image quality is calculated using a mean SSIM given as:

$$MSSIM\left(x,y\right) = \frac{1}{M}\sum_{j=1}^{M} SSIM(x_j, y_j) \tag{17}$$

The GMS map at location i is given as:

$$GMS\left(i\right) = \frac{2m_r\left(i\right)m_d\left(i\right) + c}{m_r^2\left(i\right) + m_d^2\left(i\right) + c} \tag{18}$$

The average value GMSM is given as:

$$GMSM = \frac{1}{N}\sum_{i=1}^{N} GMS(i) \tag{19}$$

The standard deviation of this GMS map results in the final image quality score known as GMSD [38]. The formula to compute GMSD is given as:

$$GMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(GMS\left(i\right) - GMSM\right)^2} \tag{20}$$

The higher the GMSD score, the more distorted is the image. The image gradients are sensitive to distortions occurring at different degrees for various local structures.

2.5. Blocking artifacts

In accordance with transform coding, an image is divided into subimages known as blocks. If an image is of $N \times N$ size, then blocks will be of $n \times n$ size, where n can equal 4, 8, 16, 32, and so on. Each block is processed independently [8]. To achieve the complete image again, these blocks are aggregated. This complete procedure results in visual impairment in the image as discontinuity at the block edges or boundaries becomes visible, as shown in Figure 4. This visual impairment is known as blocking artifacts.

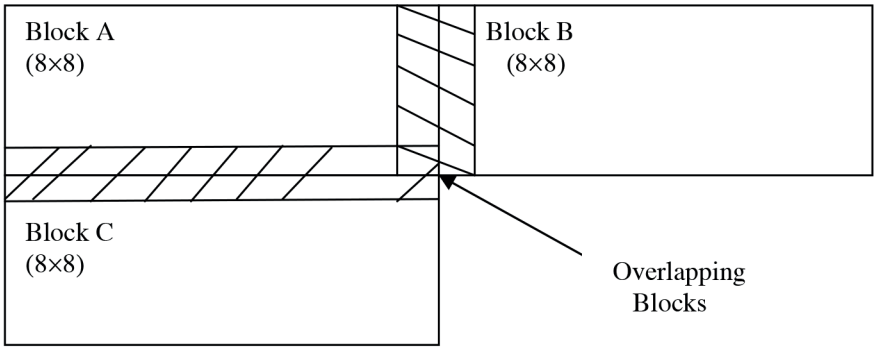

Figure 4. Horizontal and vertical boundaries.

The main cause of a blocking artifact is the loss in the accuracy of the transform coefficient resulting from the independent quantization of each block. During the process of quantization, a large number of transform coefficients are discarded. When an image is converted back to its spatial domain, a quantization error spreads all over the image as quantization is done in the transform domain. Blocking artifacts are prominent when coarse quantization is done. The visibility of blocking artifacts is high in plain areas or slowly varying portions of the image.

In an image compression procedure, there is always a trade-off between the coding bit rate and coded image quality. It is generally observed that with an increase in the coding bit rate, the quality of a reconstructed image improves. However, this improvement is restricted by limited transmission bandwidth and the storage space available. To improve the visual quality of an image, many blocking artifact removal algorithms have been proposed such as DCT filtering, the spatial averaging method, wavelet filtering, and reconstruction techniques for reconstructed images [39]. These algorithms can be classified into the preprocessed and postprocessed methods. Preprocessed methods are applied in the spatial domain before encoding. While the postprocessing technique can be applied in both the transform and the spatial domains, it is done on the reconstructed image at the decoding end.

3. Implementation of the image compression algorithm

The image compression process begins with transform coding. The original image is rotated by $\emptyset_1$ in the x direction and $\emptyset_2$ in the y direction to map $f(x,y)$ via interpolation of $\left[\frac{(cos\emptyset_1 x+sin\emptyset_1 y)}{\cos(\emptyset_1-\emptyset_2)}, \frac{(-sin\emptyset_2 x+cos\emptyset_2 y)}{\cos(\emptyset_1-\emptyset_2)}\right]$. Mathematically, mapping via Bil Intr. is given in Eq. (18). Similarly, substituting the values $i^{'} = (xcos\emptyset_1 + ysin\emptyset_1)/\cos(\emptyset_1 - \emptyset_2)$ and $j^{'} = (-xsin\emptyset_2 + ycos\emptyset_2)/\cos(\emptyset_1 - \emptyset_2)$ in Eq. (10), we are left with a Bic Intr. image. $f(x_r)$ gives the nearest neighbor interpolated image.

$$f(x_r) = \frac{x_{r-1} + x_r}{2} < x \leq \frac{x_r + x_{r+1}}{2} \tag{21}$$

The interpolated image of $N \times N$ size is divided into subimages known as blocks of $n \times n$ size. The proposed algorithm uses N = 512 and n = 8. Each block undergoes transformation independently. Mathematically,

$$F_{\emptyset_1,\emptyset_2}^{a_1,a_2} = F^{a_1,a_2}(f[q,r])$$

where

$$q = (cos\emptyset_1 x + sin\emptyset_1 y) / \cos(\emptyset_1 - \emptyset_2)$$

and

$$r = \left(-sin\emptyset_2 x + cos\emptyset_2 y\right) / \cos\left(\emptyset_1 - \emptyset_2\right) \tag{22}$$

The resulting image is in the transform domain rather than the spatial domain. The next quantization of obtained transform coefficients is done to remove the irrelevant information intact within an image. The compression ratio plays an important role in determining the limit for quantization of transform coefficients. The compression ratio is given as:

$$CR = \frac{\text{no. of compressed bits-no. of original bits}}{\text{no. of original bits}}$$

To move the image back into the spatial domain, the complete procedure in the inverse direction is implemented. Thus, the decompressed image is obtained. The quality of the obtained image is evaluated by means of different IQMs such as PSNR, MSE, MSSIM, and GMSD, with the addition of the blocked MSE. Figures 5 and 6 show the block diagram of the process. In block-based image compression, the algorithm-blocking artifacts are the prominent visual impairment in the reconstructed and decoded image. Blocking can be detected by measuring the MSE of the block boundaries in vertical and horizontal directions.

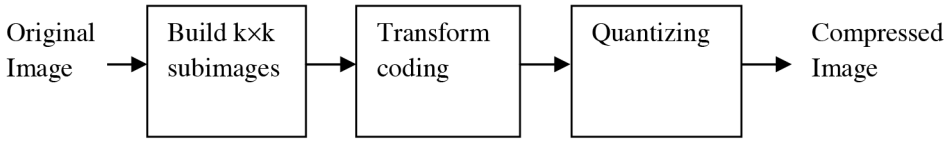

Figure 5. Encoding process of block-based image compression.

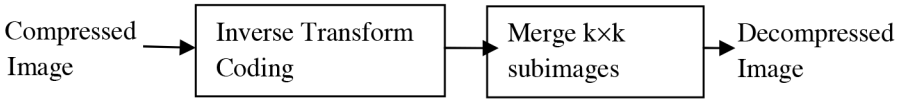

Figure 6. Decoding process of block-based image compression.

$$\begin{aligned} f\left[\frac{(cos\emptyset_1 x + sin\emptyset_1 y)}{\cos\left(\emptyset_1 - \emptyset_2\right)}, \frac{(-sin\emptyset_2 x + cos\emptyset_2 y)}{\cos\left(\emptyset_1 - \emptyset_2\right)}\right] = \frac{1}{\left(x_2 - x_1\right)\left(y_2 - y_1\right)} \times \\ \left[f\left(r_{11}\right)\left(x_2 - \frac{(cos\emptyset_1 x + sin\emptyset_1 y)}{\cos\left(\emptyset_1 - \emptyset_2\right)}\right)\left(y_2 - \frac{(-sin\emptyset_2 x + cos\emptyset_2 y)}{\cos\left(\emptyset_1 - \emptyset_2\right)}\right) + \right. \\ f\left(r_{12}\right)\left(x_2 - \frac{(cos\emptyset_1 x + sin\emptyset_1 y)}{\cos\left(\emptyset_1 - \emptyset_2\right)}\right)\left(\frac{(-sin\emptyset_2 x + cos\emptyset_2 y)}{\cos\left(\emptyset_1 - \emptyset_2\right)} - y_1\right) + \\ f\left(r_{21}\right)\left(\frac{(cos\emptyset_1 x + sin\emptyset_1 y)}{\cos\left(\emptyset_1 - \emptyset_2\right)} - x_1\right)\left(y_2 - \frac{(-sin\emptyset_2 x + cos\emptyset_2 y)}{\cos\left(\emptyset_1 - \emptyset_2\right)}\right) + \\ \left. f\left(r_{22}\right)\left(\frac{(cos\emptyset_1 x + sin\emptyset_1 y)}{\cos\left(\emptyset_1 - \emptyset_2\right)} - x_1\right)\left(\frac{(-sin\emptyset_2 x + cos\emptyset_2 y)}{\cos\left(\emptyset_1 - \emptyset_2\right)} - y_1\right)\right] \end{aligned} \tag{23}$$

Mathematically, the blocked MSE per pixel is given as:

$$MSE_h + MSEv = \sqrt{\frac{1}{MN}\sum_{i=8}^{9}\sum_{j=8}^{9}\left[u\left(i,j\right) - v(i,j)\right]^2} \tag{24}$$

Table 1. Optimized parameters for NSDFrFT using different interpolation techniques, DFrFT, and FFT.

Compression %	Method	a_1	a_2	$\emptyset_1\emptyset_2$	MSE	PSNR	MSSIM	GMSD	Blocked MSE
70	NSDFrFT-NN Intr.	0.93	–0.99	0.0524	21.4090	34.8248	0.9740	0.0952	0.1327
	NSDFrFT-Bil Intr.	0.95	-0.98	0.0524	6.4165	40.0578	0.9815	0.0937	0.1295
	NSDFrFT-Bic Intr.	0.99	–0.99	0.0524	4.7413	42.7402	0.9863	0.0306	0.1091
	Jindal [40]	0.99	0.99	0	9.2335	38.4771	0.9824	0.0827	0.1867
	Hu et al. [41]	1	1	0	4.8392	41.2831	0.9868	0.0090	0.2088
60	NSDFrFT-NN Intr.	0.95	–0.99	0.0524	19.3568	35.2625	0.9806	0.0859	0.1049
	NSDFrFT-Bil Intr.	0.93	–0.99	0.0524	5.8370	40.4689	0.9870	0.0786	0.1024
	NSDFrFT-Bic Intr.	0.95	–0.97	0.0524	3.2012	43.0777	0.9921	0.0292	0.0928
	Jindal [40]	0.98	0.98	0	3.8224	42.3075	0.9894	0.0658	0.1640
	Hu et al. [41]	1	1	0	2.1660	44.7741	0.9943	0.0054	0.1982
50	NSDFrFT-NN Intr.	0.98	–0.97	0.0524	12.5943	37.1291	0.9839	0.0767	0.1031
	NSDFrFT-Bil Intr.	0.95	–0.99	0.0524	5.0572	41.0917	0.9907	0.0658	0.0929
	NSDFrFT-Bic Intr.	0.93	–0.99	0.0524	1.7945	45.5913	0.9970	0.0168	0.0768
	Jindal [40]	0.97	0.97	0	1.9608	45.2064	0.9964	0.0646	0.1367
	Hu et al. [41]	1	1	0	1.0368	47.9740	0.9976	0.0034	0.1646
50	NSDFrFT-NN Intr.	0.97	–0.98	0.0175	11.8671	37.3874	0.9866	0.0759	0.0863
	NSDFrFT-Bil Intr.	0.91	–0.99	0.0524	4.7800	41.3365	0.9947	0.0652	0.0830
	NSDFrFT-Bic Intr.	0.98	–0.97	0.0524	1.0734	47.8230	0.9990	0.0117	0.0753
	Jindal [40]	0.97	0.97	0	0.8604	48.7836	0.9975	0.0620	0.1304
	Hu et al. [41]	1	1	0	0.3752	52.3879	0.9993	0.0031	0.1312
40	NSDFrFT-NN Intr.	0.97	–0.98	0.0175	11.8671	37.3874	0.9866	0.0759	0.0863
	NSDFrFT-Bil Intr.	0.91	–0.99	0.0524	4.7800	41.3365	0.9947	0.0652	0.0830
	NSDFrFT-Bic Intr.	0.98	–0.97	0.0524	1.0734	47.8230	0.9990	0.0117	0.0753
	Jindal [40]	0.97	0.97	0	0.8604	48.7836	0.9975	0.0620	0.1304
	Hu et al. [41]	1	1	0	0.3752	52.3879	0.9993	0.0031	0.1312
30	NSDFrFT-NN Intr.	0.93	–0.99	0.0175	11.3014	37.5995	0.9863	0.0710	0.0826
	NSDFrFT-Bil Intr.	0.93	–0.99	0.0175	4.5991	41.5041	0.9955	0.0650	0.0705
	NSDFrFT-Bic Intr.	0.95	–0.99	0.0175	0.9018	48.5798	0.9995	0.0109	0.0604
	Jindal [40]	0.98	0.98	0	0.2653	53.8928	0.9978	0.0568	0.1284
	Hu et al. [41]	1	1	0	0.1374	56.7501	0.9997	0.0025	0.1280
20	NSDFrFT-NN Intr.	0.94	–0.97	0.0175	11.1559	37.6558	0.9866	0.0666	0.0753
	NSDFrFT-Bil Intr.	0.97	–0.99	0.0175	3.5342	42.6479	0.9957	0.0567	0.0645
	NSDFrFT-Bic Intr.	0.95	–0.97	0.0175	0.6798	49.5615	0.9997	0.0106	0.0582
	Jindal [40]	0.99	0.99	0	0.0832	58.9274	0.9988	0.0532	0.1190
	Hu et al. [41]	1	1	0	0.0374	62.4069	0.9999	0.0029	0.1269
10	NSDFrFT-NN Intr.	0.92	–0.97	0.0175	4.2158	41.8820	0.9867	0.0647	0.0650
	NSDFrFT-Bil Intr.	0.93	–0.96	0.0175	2.5424	44.0784	0.9960	0.0545	0.0568
	NSDFrFT-Bic Intr.	0.95	–0.97	0.0175	0.2632	53.9284	0.9999	0.0023	0.0485
	Jindal [40]	0.98	0.98	0	0.0305	63.2888	1.0000	0.0094	0.1052
	Hu et al. [41]	1	1	0	0.0054	70.7679	1.0000	0.0039	0.1270

where MSE_h is the MSE in the horizontal direction and $MSEv$ is the MSE in the vertical direction. M and N are the number of rows and columns in a block. However, the MSE of overlapping pixels has to be subtracted.

4. Simulation results

The algorithm of NSDFrFT, with Bil Intr., Bic Intr., and NN Intr., has been implemented on several test images. The simulation results for Lena (as the test image) are given in Table 1 for compression ratios of 10%–70%.

Table 1 compares the three interpolation methods with the DFrFT [40] and FFT [41]. The optimized parameters $a_1, a_2, \emptyset_1, \emptyset_2$, MSE, PSNR, MSSIM, GMSD, and blocked MSE are considered as parameters of comparison. The values of $\emptyset_1$, $\emptyset_2$ in the case of NSDFrFT and $a_1 a_2$ in the case of DFrFT are considered similar for simplicity's sake. From Table 1, the main observations are that NSDFrFT-Bic Intr. performs better in comparison to NSDFrFT-Bil Intr. and NSDFrFT-NN Intr. The process of interpolation involved in the NSDFrFT definition performs an additional operation as a low pass filter (LPF), i.e. the softening of edges or sharp transitions, enabling NSDFrFT to perform better in terms of reduced blocking. NSDFrFT-Bic Intr. also performs better in terms of higher compression than DFrFT in every respect. Blocking artifacts are significant for higher compression ratios.

Table 2 shows the computational time of NSDFrFT-NN Intr., NSDFrFT-Bil Intr., NSDFrFT-Bic Intr., and DFrFT. The computational time includes both encoding and decoding time because the number of parameters in DFrFT is only two a_1 and a_2 in comparison to NSDFrFT, which has four parameters, namely $a_1, a_2, \emptyset_1, \emptyset_2$, for computation.

Table 2. CPU time for different definitions of NSDFrFT, DFrFT, and FFT.

Transform technique	CPU time (s)
NSDFrFT-NN Intr.	20.7043
NSDFrFT-Bil Intr.	20.1349
NSDFrFT-Bic Intr.	20.0719
Jindal et al. [40]	18.6351
Hu et al. [41]	09.7235

Therefore, the required computational time for encoding and decoding has also decreased. NSDFrFT-Bic Intr. takes less time as the number of calculations is reduced since it takes 16 points for interpolation. On the other hand, NSDFrFT-Bil Intr. takes 4 points for interpolation and NSDFrFT-NN Intr. takes only 2 points, resulting in an increment in the number of calculations. For further study, the images are categorized into three classes: high frequency images (Baboon, Grass), medium frequency images (Barbara, House), and low frequency images (Pepper, Boat) [42]. The required images are taken from http://sipi.usc.edu/database/ for simulation. Tables 3 and 4 outline the optimized IQMs and blocked MSE at their respective rates of 70% and 50% compression for all classes. PSNR and MSE are better for images of all classes in the case of NSDFrFT-Bic Intr. in comparison to DFrFT. The optimized values of MSSIM and GMSD show that the reconstructed image has high structural similarity to the original image along with high subjective image quality for low frequency images compressed via NSDFrFT-Bic Intr. However, reconstructed images of all classes have lower blocked MSE for NSDFrFT-Bic Intr. than with DFrFT.

The compressed images of Lena are compared using NSDFrFT-NN Intr., Bil Intr., Bic Intr., DFrFT, and FFT for compression ratios of 50% and are shown in Figure 7.

The plot of the GMSD score vs. compression ratio shown in Figure 8 suggests that subjective image quality is high for NSDFrFT-Bic Intr., which considers the local image distortions caused by local structure diversity. The plot of the MSSIM index vs. the compression ratio given in Figure 9 demonstrates that the reconstructed image from NSDFrFT-Bic Intr. has a high structural similarity to the original image in comparison to compared transform techniques. The plot of blocked MSE vs. compression ratio in Figure 10 suggests that the NSDFrFT-Bic Intr. reconstructed image has a lower degree of blocking compared to transform techniques.

Table 3. Optimized IQMs and blocked MSE for 70% compression.

Type of image		MSE		PSNR		MSSIM		GMSD		Blocked MSE	
		NSDFrFT (presented work)	Jindal [40]	NSDFrFT (presented work)	Jindal [40]	NSDFrFT (presented work)	Jindal [40]	NSDFrFT (presented work)	Jindal [40]	NSDFrFT (presented work)	Jindal
Low frequency image	Pepper	13.9341	20.1363	36.6900	35.0910	0.0764	1.3745	0.9827	0.9757	0.1654	0.2471
	Boat	14.9333	19.0190	36.3892	34.7561	0.1644	1.5900	0.9807	0.8919	0.6659	0.7789
Medium frequency image	Barbara	26.7606	28.1363	35.0811	33.4910	0.0801	0.9384	0.9800	0.9830	0.0614	0.1703
	House	2.4457	10.1346	44.2467	38.0727	0.1112	0.0272	0.9891	0.9930	0.2708	0.3222
High frequency image	Baboon	2.9424	5.0884	43.4437	41.0650	0.0488	0.0095	0.9871	0.9915	0.0287	0.0715
	Grass	77.6940	190.013	29.2280	26.2799	0.0629	0.0565	0.9674	0.9713	0.2297	0.2402

Table 4. Optimized IQMs and blocked MSE for 50% compression.

Type of image		MSE		PSNR		MSSIM		GMSD		Blocked MSE	
		NSDFrFT (presented work)	Jindal [40]	NSDFrFT (presented work)	Jindal [40]	NSDFrFT (presented work)	Jindal [40]	NSDFrFT (presented work)	Jindal [40]	NSDFrFT (presented work)	Jindal
Low frequency image	Pepper	7.7921	14.4629	39.2142	36.5283	0.9925	0.9925	0.0698	1.7222	0.0273	0.2471
	Boat	5.8015	94.9666	40.4954	28.3551	0.9901	0.9850	0.1516	2.9954	0.4636	0.7789
Medium frequency image	Barbara	14.4629	20.1822	40.1371	36.5283	0.9893	0.9941	0.0690	0.1656	0.1208	0.2325
	House	1.9454	4.0101	45.1699	40.7046	0.9967	0.9987	0.1033	0.2441	0.2521	0.3237
High frequency image	Baboon	2.1722	2.9424	44.7618	43.7051	0.9968	0.9990	0.0381	0.0809	0.0351	0.0810
	Grass	18.2413	24.7304	35.5202	28.3659	0.9885	0.9970	0.0529	0.0220	0.2182	0.2330

Figure 7. The compressed Lena image at 10% compression (a–e).

5. Discussion and conclusion

The practical effectiveness of NSDFrFT in the image compression algorithm for reducing the blocking artifacts was implemented. The summarized results of the analysis show that NSDFrFT with different interpolation methods resulted in higher image quality parameters than DFrFT with a relatively high GMSD score. We can observe that for an image divided into 8 $\times$ 8 blocks, the NSDFrFT definition utilizing Bic Intr. for mapping purposes performed better with higher compression percentages. NSDFrFT was compared with the discrete factional transform for Lena and Pepper of size 512 $\times$ 512 at 50% compression. An improvement of 0.39 dB for Lena and 4.29 dB for Pepper in PSNR was achieved. However, computational lag in this case for Lena and Pepper are 2.81 s and 2.70 s, respectively, from DFrFT. Among the different types of images used for analysis,

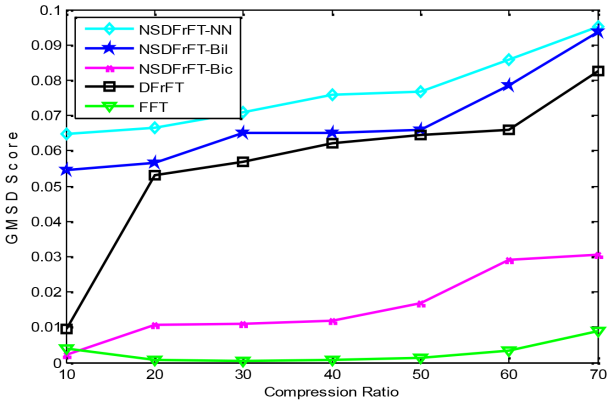

Figure 8. Compression ratio vs. GMSD score for different definitions of NSDFrFT, FrFT, and FFT.

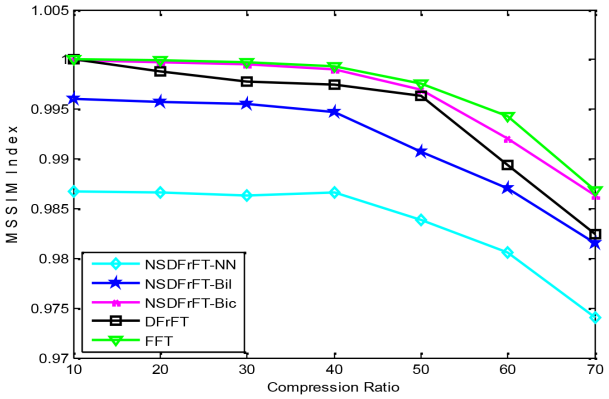

Figure 9. Compression ratio vs. MSSIM index for different definitions of NSDFrFT, DFrFT, and FFT.

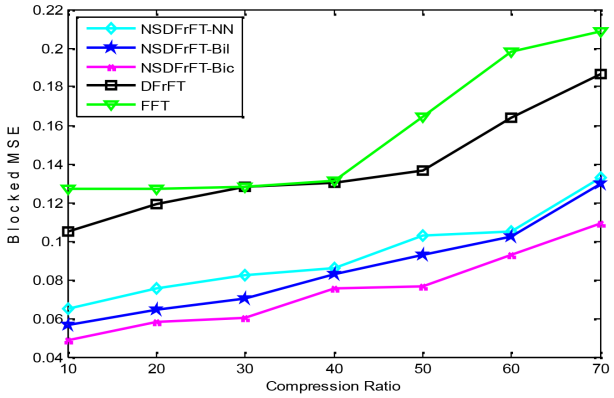

Figure 10. Compression ratio vs. blocked MSE for different definitions of NSDFrFT, DFrFT, and FFT.

low frequency images responded better for NSDFrFT-Bic Intr. than for DFrFT. The collective results of all image quality parameters suggest that NSDFrFT-Bic Intr. performs better for higher compression percentages.

The images compressed using NSDFrFT resulted in reduced blocking at the boundaries of the block. All the variations of NSDFrFT, namely NSDFrFT-NN Intr., NSDFrFT-Bil Intr., and NSDFrFT-Bic Intr., resulted in a reduced number of blocking artifacts in the compressed images in comparison to DFrFT. However, of all of the variations of NSDFrFT implemented, NSDFrFT-Bic Intr. resulted in minimal blocked MSE. An improvement in blocked MSE of about 53.34% for Lena and 74.71% for Pepper was achieved for NSDFrFT-Bic Intr.

Interpolation is a key aspect in the definition of NSDFrFT. As a result, improving the interpolation technique can be a way to achieve a highly improved performance. The computational time of the proposed method is longer than that of DFrFT and thus improvement in this regard needs to be considered for future work.

References

[1] McIntyre KA. Dynamic Bandwidth Adaptive Image Compression/Decompression Scheme. U.S. Patent 7 024 045, 2006.

[2] Yng TLB, Lee BG, Yoo H. A low complexity and lossless frame memory compression for display devices. IEEE T Consum Electr 2008; 54: 1453-1458.

[3] Shukla J, Alwani M, Tiwari AK. A survey on lossless image compression methods. In: IEEE 2010 International Conference on Computer Engineering and Technology; 16–18 April 2010; Chengdu, China. New York, NY, USA: IEEE. pp. V6-136-V6-141.

[4] Grgic S, Grgic M, Zovko-Cihlar B. Performance analysis of image compression using wavelets. IEEE T Ind Electron 2001; 48: 682-695.

[5] Wiener N. Hermitian polynomials and Fourier analysis. J Math Phys 1929; 8: 70-73.

[6] Gonzalez RC, Woods RE. Digital Image Processing. 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2008.

[7] Jayaraman S, Esakkirajan S, Veerakumar T. Digital Image Processing. New Delhi, India: Tata McGraw-Hill Education, 2011.

[8] Bandyopadhyay SK, Paul TU, Rajchoudhury A. Image compression using approximate matching and run length. Int J Adv Comp Sci Appl 2011; 2: 117-121.

[9] Murugan G, Kannan E, Arun S. Lossless image compression algorithm for transmitting over low bandwidth line. Int J Comp Sci Inf Secur 2011; 9: 140-145.

[10] Thyagarajan KS. Still Image and Video Compression with MATLAB. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2011.

[11] Skodras A, Christopoulos C, Ebrahimi T. The JPEG 2000 still image compression standard. IEEE T Signal Proces 2001; 18: 36-58.

[12] Watson AB. Image compression using the discrete cosine transform. Math J 1994; 4: 81-88.

[13] Jindal N, Singh K. Image and video processing using discrete fractional transforms. Sign Imag Vid Process 2014; 8: 1543-1553.

[14] Singh K. Performance of discrete fractional Fourier transform classes in signal processing applications. PhD, Thapar University, Patiala, India, 2005.

[15] Ozaktus HM, Zalevsky Z, Kutay MA. The fractional Fourier transform with applications in optics and signal processing. New York, NY, USA: John Wiley & Sons Ltd., 2000.

[16] Ludwig LF. Correction of Image Misfocus via Fractional Fourier Transform. U.S. Patent 6 687 418, 2004.

[17] Cusmario A. Cryptography Method using Modified Fractional Fourier Transform Kernel. U.S. Patent 6 718 038, 2004.

[18] Namias V. The fractional order Fourier transform and its applications to quantum mechanics. IMA J Appl Math 1980; 25: 241-265.

[19] Ozaktus HM, Mendlovic D. Fractional Fourier optics. J Opt Soc Am 1995; 12: 743-751.

[20] Cariolario G, Ersrghe T, Kraniauskas P, Laurenti N. A unified framework for the fractional Fourier transform. IEEE T Signal Proces 1998; 46: 3206-3212.

[21] Pei SC, Yeh MH, Luo TL. Fractional Fourier series expansion for finite signals and dual extension to discrete-time fractional Fourier transform. IEEE T Signal Proces 1999; 47: 2883-2888.

[22] Pei SC, Yeh MH. Discrete fractional Fourier transform. In: IEEE 1996 International Symposium on Connecting the World; 12–15 May 1996; Atlanta, GA. New York, NY, USA: IEEE. pp. 536-539.

[23] Santhanam B, McClellan JH. The DRFT-a rotation in time-frequency space. In: IEEE 1995 International Conference on Acoustics, Speech, and Signal Processing; 9–12 May 1995; Detroit, MI, USA. New York, NY, USA: IEEE. pp. 921-924.

[24] Pei SC, Ding JJ. Closed-form discrete fractional and affine Fourier transforms. IEEE T Signal Proces 2000; 48: 1338-1353.

[25] Candan C, Kutay MA, Ozaktus HM. The discrete fractional Fourier transform. IEEE T Signal Proces 2000; 48: 1329-1338.

[26] Singh AK, Saxena R. DFRFT: a classified review of recent methods with its application. J Eng 2013; 2013: 1-13.

[27] Ozaktas HM, Ankan O, Kutay MA, Bozdagi G. Digital computation of the fractional Fourier transform. IEEE T Signal Proces 1996; 44: 2141-2150.

[28] Pei SC, Yeh MH, Tseng CC. Discrete fractional Fourier transform based on orthogonal projections. IEEE T Signal Proces 1999; 47: 1335-1348.

[29] Pei SC, Hsue WL. Random discrete fractional Fourier transform. IEEE Signal Proc Let 2009; 16: 1015-1018.

[30] Yeh MH, Pei SC. A method for the discrete fractional Fourier transform computation. IEEE T Signal Proces 2003; 51: 889-891.

[31] Ludwig LF. High-Accuracy Centered Fractional Fourier Transform Kernel for Optical Imaging and Other Applications. U.S. Patent 8 712 185, 2014.

[32] Ludwig LF. Discrete Fractional Fourier Numerical Environments for Computer Modeling of Image Propagation Through a Physical Medium in Restoration and Other Applications. U.S. Patent 8 094 969, 2012.

[33] Ding JJ, Pei SC. Heisenberg's uncertainty principles for the 2-D nonseparable linear canonical transforms. Signal Process 2013; 93: 1027-1043.

[34] Sahin A, Kutay MA, Ozaktus HM. Nonseparable two-dimensional fractional Fourier transform. Appl Optics 1998; 37: 5444-5453.

[35] Prajapati A, Naik S, Mehta S. Evaluation of different image interpolation algorithms. Int J Comp Appl 2012; 58: 6-12.

[36] Richardson IEG. H.264 and MPEG-4 Video Compression. West Sussex, UK: John Wiley & Sons, Inc., 2003.

[37] Whang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE T Image Process 2004; 13: 600-612.

[38] Xue W, Zhang L, Mou X, Bovik AC. Gradient magnitude similarity deviation: a highly efficient perceptual image quality index. IEEE T Image Process 2014; 23: 684-695.

[39] Singh J, Singh S, Singh D, Uddin M. Detection method and filters for blocking effect reduction of highly compressed images. Signal Process Image Comm 2011; 26: 496-503.

[40] Jindal N. Performance of fractional transforms in image and video processing. PhD, Thapar University, Patiala, India, 2014.

[41] Hu J, Deng J, Wu J. Image compression based on improved FFT algorithm. J Netw 2011; 6: 1041-1048.

[42] Rout S. Orthogonal vs. biorthogonal wavelets for image compression. MSc, Virginia Tech, Blacksburg, VA, USA, 2003.