STM JOURNALS

**Research & Reviews: Discrete Mathematical Structures**
ISSN: 2394-1979(online)
Volume 2, Issue 3
www.stmjournals.com

# Image Compression via Non-separable Discrete Fractional Fourier Transform

***Kritika Mittal****
Department of Electronics & Communication, Thapar University, Patiala, India

***Abstract***

*The main idea behind image compression is to reduce the bandwidth for transmission and required space for storage. Thus, image compression is of great importance. In this paper we present the image compression technique using Non-Separable Discrete Fractional Fourier Transform (NSDFrFT). Numerical simulation results suggested that image compression method using NSDFrFT as transform technique gives better performance for images when compression ratio is high in comparison to DFrFT and JPEG. Different image quality measurement methods such as Gradient Magnitude Similarity Deviation (GMSD), Mean Structural Similarity Index Measure (MSSIM), Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR) are used to determine the performance.*

***Keywords****: Image Compression, NSDFrFT, GMSD, MSSIM, MSE*

****Author for Correspondence*** E-mail: kritikamittal1991@gmail.com

## INTRODUCTION

Storage space required, the transmission bandwidth and the retention of quality of image are important issues to be dealt by image compression with reduced cost [1–3]. The pixels in image are correlated to each-other and the amount of correlated pixels in an image is termed as redundancy. Moreover, there is some data in image which cannot be noted by human eyes. Thus, such a data is irrelevant [4]. Both redundancy and irrelevancy made image compression possible and effective. A continuous image is sampled and quantized to get a digital image. But such a procedure results in big image which requires large storage space and samples to represent energy. Storage space is a limitation and adding extra storing device is not a solution to the storage problem. So, to counter such problems many image compression algorithms are given [5].

Modest compression is achieved when lossless image compression is done. As no information is lost in the process, thus result is a compressed image identical to the original image. However, in lossy image compression, significant compression can be achieved by discarding redundant data. But the quality of image is degraded due to blocking artifacts [6]. Apart from this image compression methods can be categorized as predictive coding and transform coding. Based on past known values, future values are predicted by predictive coding while transform coding converts the domain of image from spatial to frequency. JPEG2000 compression coding [7], Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) [8], DFrFT [9], Discrete Fractional Cosine Transform (DFrCT) [10], wavelet domain [11] are some of the image compression algorithms.

The paper discusses the usefulness of the image compression using NSDFrFT algorithm for grey-scaled images,. Preliminaries of NSDFrFT, GMSD, MSSIM, PSNR and MSE are discussed followed by the methodology implemented and discusses results with conclusion.

## PRELIMINARIES

The prelimies about NSDFrFT and various image quality parameters have been discussed in detail.

**Non-Separable Discrete Fractional Fourier Transform (NSDFrFT)**

Namias [12] first gave the definition of Fractional Fourier Transform (FrFT), a generalization of Fourier transform (FT). The definition of FrFT for different signals such as one-dimension and multi-dimension, aperiodic and periodic, discrete and continuous was given by Cariolario *et al.* [13].

Technology advancement motivated McClellan *et al.* [14] to give DFrFT, as usage of computer and other digital devices increased. The first definition of DFrFT available for calculation purposes was given by Pei *et al.* [15]. Utilizing non-separable linear canonical transform (NSLCT) definition [16, 17], definition for FrFT was given by Ozaktas *et al.* [18] and Almeida [19].

The definition of DFrFT is given as

$$G^{a_1,a_2}[v(x,y)](x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} S_{a_1,a_2}(x,y,x',y')\ v(x',y')dx'dy' \quad (1)$$

Where,

$$S_{a_1,a_2}(x,y,x',y') = S_{a_1}(x,x')S_{a_2}(y,y') \quad (2)$$

The NSDFrFT, a generalized case of DFrFT was given by Sahin *et al.* [20]. NSDFrFT uses the concept of interpolation for the purpose of rotation of grid so as to return to original rectangular grid. Sahin used bilinear interpolation for the rotation of x and y axis by $\varphi_1$ and $\varphi_2$ to obtain $x'$ and $y'$ respectively as shown in Figure 1. The definition of NSDFrFT has four parameters $a_1, a_2, \varphi_1$ and $\varphi_2$. When $\varphi_1$ and $\varphi_2$ are zero then NSDFrFT converts into DFrFT. Thus NSDFrFT is as follows

$$G_{\varphi_1,\varphi_2}^{a_1,a_2}[v(x,y)] = G^{a_1,a_2}[v(i,j)] \quad (3)$$

Where,

i=$(cos\varphi_1\ x + sin\varphi_1\ y)/\cos(\varphi_1\ - \varphi_2\ )$ and j= $(-sin\varphi_2\ x + cos\varphi_2\ y)/\cos(\varphi_1\ - \varphi_2\ )$

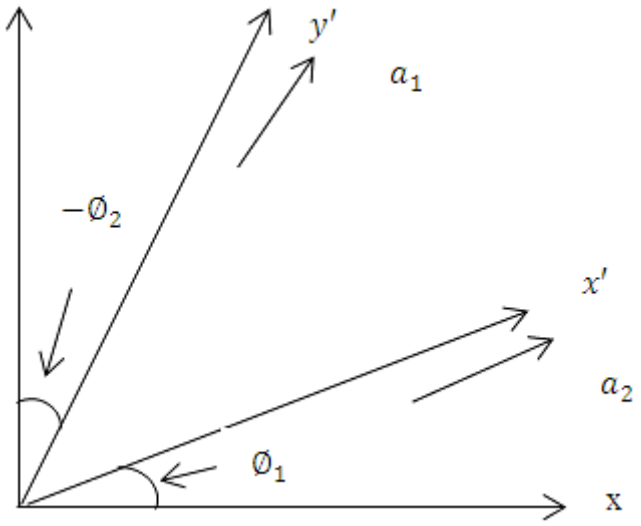



***Fig. 1:*** *Time-Frequency plane rotation for NSFrFT.*

Interpolation plays an important role in the definition of NSDFrFT. Thus, to study the effect of interpolation on the performance of NSDFrFT, Nearest-Neighbor interpolation (NN.), Bilinear interpolation (Bil.) and Bicubic interpolation (Bic.) has been used.

**Image Quality Parameters**

The comparison between the original image and the resultant image of a process based upon an mathematical algorithm is done by Objective Quality Metrics (OQM). OQM tells about extend of distortion present in the resultant image when compared with original image. Various OQM methods used for analysis of the quality of reconstructed image are PSNR, MSE [21], GMSD [22] and MSSIM [23].

The formula for MSE is given as

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \quad (4)$$

Where, I and K are the images to be compared.

The formula for PSNR is given as

$$PSNR = 10.log_{10}\left(\frac{MAX_I^2}{MSE}\right) \quad (5)$$

Where, $MAX_I$ is the maximum pixel value.
The standard deviation of the GMS map results into the final image quality score known as GMSD [22]. The formula to compute GMSD is given as

$$GMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(GMS(i) - GMSM)^2} \quad (6)$$

Where, $GMSM$ is the average value of GMS map forming the resultant final image quality score and $GMS$ is the GMS map at location $i$. The higher the GMSD score, the more is the distortion in the image.

The mathematical representation of SSIM index is as follows

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (7)$$

Where $\mu_x$ and $\mu_y$ are the mean intensity. $\sigma_x$ and $\sigma_y$ are the contrast. $C_1$ and $C_2$ are the constants required to balance the mathematical representation [23]. In-order to calculate SSIM for complete image MSSIM Index is computed using the following formula

$$MSSIM(x,y) = \frac{1}{M}\sum_{j=1}^{M} SSIM(x_j, y_j) \quad (8)$$

## IMAGE COMPRESSION ALGORITHM

The original image is rotated by $\emptyset_1$ in x direction and $\emptyset_2$ in y direction 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 $f(x,y)$ via interpolation. Mathematically, mapping via bilinear interpolation is given as

$$f\left[\frac{(cos\emptyset_1 x+sin\emptyset_1 y)}{\cos(\emptyset_1-\emptyset_2)}, \frac{(-sin\emptyset_2 x+cos\emptyset_2 y)}{\cos(\emptyset_1-\emptyset_2)}\right] = \frac{1}{(x_2-x_1)(y_2-y_1)} \times$$
$$\left[f(r_{11})\left(x_2 - \frac{(cos\emptyset_1 x+sin\emptyset_1 y)}{\cos(\emptyset_1-\emptyset_2)}\right)\left(y_2 - \frac{(-sin\emptyset_2 x+cos\emptyset_2 y)}{\cos(\emptyset_1-\emptyset_2)}\right) + \right.$$
$$f(r_{12})\left(x_2 - \frac{(cos\emptyset_1 x+sin\emptyset_1 y)}{\cos(\emptyset_1-\emptyset_2)}\right)\left(\frac{(-sin\emptyset_2 x+cos\emptyset_2 y)}{\cos(\emptyset_1-\emptyset_2)} - y_1\right) + f(r_{21})\left(\frac{(cos\emptyset_1 x+sin\emptyset_1 y)}{\cos(\emptyset_1-\emptyset_2)} - x_1\right)\left(y_2 -\right.$$
$$\left.\left.\frac{(-sin\emptyset_2 x+cos\emptyset_2 y)}{\cos(\emptyset_1-\emptyset_2)}\right) + f(r_{22})\left(\frac{(cos\emptyset_1 x+sin\emptyset_1 y)}{\cos(\emptyset_1-\emptyset_2)} - x_1\right)\left(\frac{(-sin\emptyset_2 x+cos\emptyset_2 y)}{\cos(\emptyset_1-\emptyset_2)} - y_1\right)\right] \quad (9)$$

Similarly, mathematical representation for mapping via bicubic interpolation can be obtained by substituting the values $x = (xcos\emptyset_1 + ysin\emptyset_1)/\cos(\emptyset_1 - \emptyset_2)$ and $y = (-xsin\emptyset_2 + ycos\emptyset_2)/\cos(\emptyset_1 - \emptyset_2)$ in the

$$f[i', j'] = \sum_{l=-1}^{2}\sum_{m=-1}^{2} f(i+l, j+m)K(l-dx)K(dy-m) \quad (10)$$

where,

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

and

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

we get bicubic interpolated image [24]. While for nearest neighbor interpolation

$$f(x_r) = \frac{x_{r-1}+x_r}{2} < x \leq \frac{x_r+x_{r+1}}{2} \quad (11)$$

gives the nearest neighbor interpolated image.

The interpolated image of $N \times N$ size is divided into sub-images known as blocks of $n \times n$ size. For simulation purposes, $N$ =512 and $n$ =8 has been used. Each block undergoes transformation independently. Mathematically,

$$F_{\emptyset_1,\emptyset_2}^{a_1,a_2} = F^{a_1,a_2}(f[q,r]) \quad (12)$$

where,

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

Resulting image is in transform domain rather than in spatial domain. Next quantization of obtained transform coefficients is done to remove the irrelevant information intact with an image. Compression percentage plays an important role in deciding the limit for quantization of transform coefficients.

$$CP = \frac{original\ image\ bits - compressed\ image\ bits}{original\ image\ bits} \times 100 \quad (13)$$

To achieve the image back into spatial domain, the complete procedure in inverse direction is implemented. The inverse NSDFrFT is applied on the compressed image and the blocks are put together to obtain the decompressed image (Figure 2). The quality of obtained image is evaluated by means of different image quality parameters such as MSE, PSNR, MSSIM and GMSD.

In block-based image compression algorithm blocking artifacts are the prominent visual impairment in the reconstructed decoded image. Blocking can be detected by measuring the MSE of the block boundaries in vertical and horizontal direction [25].

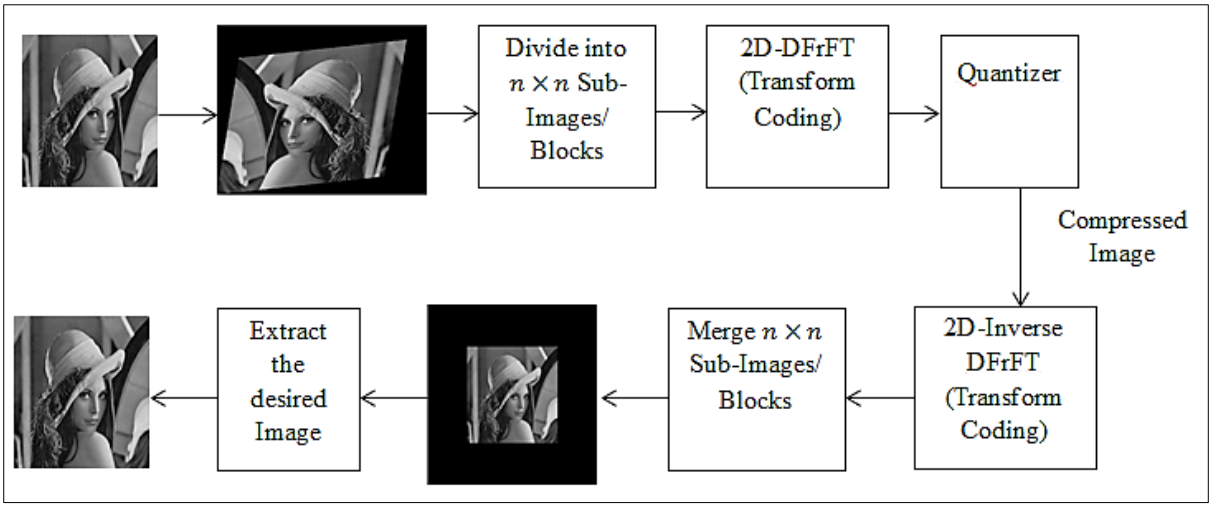



***Fig. 2:*** *Complete Procedure of Compression and Decompression Using NSDFrFT.*

Mathematically, the $MSE_{block}$ per pixel is given as

$$MSE_h + MSE_v = \sqrt{\frac{1}{MN}\sum_{i=8}^{9}\sum_{j=8}^{9}[u(i,j) - v(i,j)]^2} \quad (14)$$

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

## SIMULATION RESULTS

The optimized parameters for image compression algorithm using NSDFrFT with various interpolation techniques has been complied in Table. 1 for different compression percentages of 10%, 20%, 30%, 40%, 50%, 60% and 70% for boat image and comparison among the variations in definition of NSDFrFT is done. In-order to achieve computational simplicity, the $\emptyset_1$ and $\emptyset_2$ have been kept same and represented by $\emptyset_{op}$.

An important fact to be considered is that when the value of $\emptyset_1$ and $\emptyset_2$ is kept to be 0 than NSDFrFT converges to DFrFT. From the complied results of NSDFrFT variations, it can be concluded that NSDFrFT-Bic. performs better in comparison to NSDFrFT-NN. and NSDFrFT-Bil. All the image quality parameters, namely MSE, PSNR, MSSIM and GMSD has high optimized value for NSDFrFT-Bic. The blocked MSE denoted as $MSE_{block}$ is also less for NSDFrFT-Bic. in comparison to NSDFrFT-NN. and NSDFrFT-Bil. This clearly implies that amount of blocking at the block boundaries has been reduced significantly by NSDFrFT-Bic.

***Table 1:*** *Optimized Parameters for Boat at Different Compression Percentages Using NSDFrFT.*

| Compression Percentage | Transform Technique | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD | $MSE_{block}$ |
|---|---|---|---|---|---|---|---|---|---|
| 10 | NSDFrFT-NN. | 0.92 | -0.99 | 0.017 | 5.508 | 34.237 | 0.9951 | 0.0715 | 0.4346 |
| | NSDFrFT-Bil. | 0.92 | -0.96 | 0.017 | 4.964 | 41.171 | 0.9890 | 0.0748 | 0.3859 |
| | NSDFrFT-Bic. | 0.95 | -0.99 | 0.017 | **1.463** | **52.756** | **0.9968** | **0.0715** | **0.3465** |
| 20 | NSDFrFT-NN. | 0.95 | -0.97 | 0.017 | 5.689 | 34.205 | 0.9947 | 0.0721 | 0.4896 |
| | NSDFrFT-Bil. | 0.97 | -0.99 | 0.017 | 5.148 | 41.013 | 0.9887 | 0.0750 | 0.4064 |
| | NSDFrFT-Bic. | 0.95 | -0.99 | 0.017 | **1.673** | **50.592** | **0.9966** | **0.0718** | **0.3795** |
| 30 | NSDFrFT-NN. | 0.93 | -0.99 | 0.017 | 6.825 | 34.181 | 0.9938 | 0.0732 | 0.5567 |
| | NSDFrFT-Bil. | 0.96 | -0.99 | 0.017 | 5.599 | 40.649 | 0.9885 | 0.0752 | 0.4588 |
| | NSDFrFT-Bic. | 0.95 | -0.99 | 0.017 | **2.052** | **48.311** | **0.9961** | **0.0725** | **0.3982** |
| 40 | NSDFrFT-NN. | 0.93 | -0.98 | 0.017 | 7.519 | 34.062 | 0.9917 | 0.0762 | 0.5741 |
| | NSDFrFT-Bil. | 0.99 | -0.99 | 0.052 | 6.824 | 39.790 | 0.9876 | 0.0762 | 0.4882 |
| | NSDFrFT-Bic. | 0.95 | -0.99 | 0.052 | **4.517** | **45.370** | **0.9942** | **0.0745** | **0.4489** |
| 50 | NSDFrFT-NN. | 0.98 | -0.97 | 0.052 | 11.64 | 33.715 | 0.9876 | 0.0834 | 0.5903 |
| | NSDFrFT-Bil. | 0.95 | -0.99 | 0.052 | 9.329 | 38.432 | 0.9846 | 0.0790 | 0.5830 |
| | NSDFrFT-Bic. | 0.93 | -0.99 | 0.052 | **5.802** | **43.684** | **0.9924** | **0.0768** | **0.4901** |
| 60 | NSDFrFT-NN. | 0.95 | -0.99 | 0.052 | 15.03 | 33.355 | 0.9797 | 0.1016 | 0.5961 |
| | NSDFrFT-Bil. | 0.95 | -0.97 | 0.052 | 13.56 | 36.805 | 0.9809 | 0.0841 | 0.5831 |
| | NSDFrFT-Bic. | 0.95 | -0.99 | 0.052 | **11.60** | **39.484** | **0.9882** | **0.0838** | **0.5001** |
| 70 | NSDFrFT-NN. | 0.93 | -0.99 | 0.052 | 23.43 | 33.020 | 0.9667 | 0.1404 | 0.6312 |
| | NSDFrFT-Bil. | 0.95 | -0.97 | 0.052 | 20.16 | 35.083 | 0.9760 | 0.0933 | 0.5851 |
| | NSDFrFT-Bic. | 0.93 | -0.99 | 0.052 | **14.60** | **36.389** | **0.9804** | **0.1030** | **0.5064** |

The compressed images at 70% for boat has been shown in Figure 3 for all the variations of NSDFrFT, JPEG and DFrFT. The NSDFrFT-Bic. is compared to JPEG and DFrFT at 50% compression for Lena and Pepper image of size $256 \times 256$ in Table.2. An improvement of 1.21 dB for Lena and 4.80 dB for Pepper in PSNR has been achieved. However, the proposed algorithm lags from DFrFT in the computational time required to encode and decode the image by 1.31 sec and 0.79 sec for Lena and Pepper respectively. The number of parameters in DFrFT is only two $a_1$ and $a_2$ therefore, the required computation time for encoding and decoding is also less in comparison to NSDFrFT which has four parameters $a_1, a_2, \emptyset_1, \emptyset_2$ for computation.

***Table 2:*** *Comparison of Non-Separable Discrete Fractional Transform, Discrete Fractional Transform and JPEG at 50% Compression.*

| Algorithm | Image $(256 \times 256)$ | PSNR in dB | CPU Time | |
|---|---|---|---|---|
| | | | Encoding Time (sec) | Decoding Time (sec) |
| Non-Separable Discrete Fractional Transform | Lena | 45.76 | 6.64 | 6.64 |
| | Pepper | 47.08 | 4.27 | 4.27 |
| Discrete Fractional Transform [9] | Lena | 44.55 | 5.33 | 5.33 |
| | Pepper | 42.28 | 3.48 | 3.48 |
| JPEG [26] | Lena | 34.66 | 0.12 | 0.12 |
| | Pepper | 34.27 | 0.20 | 0.20 |

The Non-Separable Discrete Fractional Transform has been compared with Discrete Fractional Transform for Lena and Pepper of size $(512 \times 512)$ at 50% compression in Table.3. An improvement of 0.39 dB for Lena and 4.29 dB for Pepper in PSNR has been recorded. However, the computational lag in this case for Lena and Pepper are 2.81 sec and 2.70 sec respectively from DFrFT.

*(a). Original Image.*

*(b). Compressed Image using NSDFrFT-NN.*

*(c). Compressed Image using NSDFrFT-Bil.*

*(d). Compressed Image using NSDFrFT-Bic.*

*(e). Compressed Image using JPEG.*

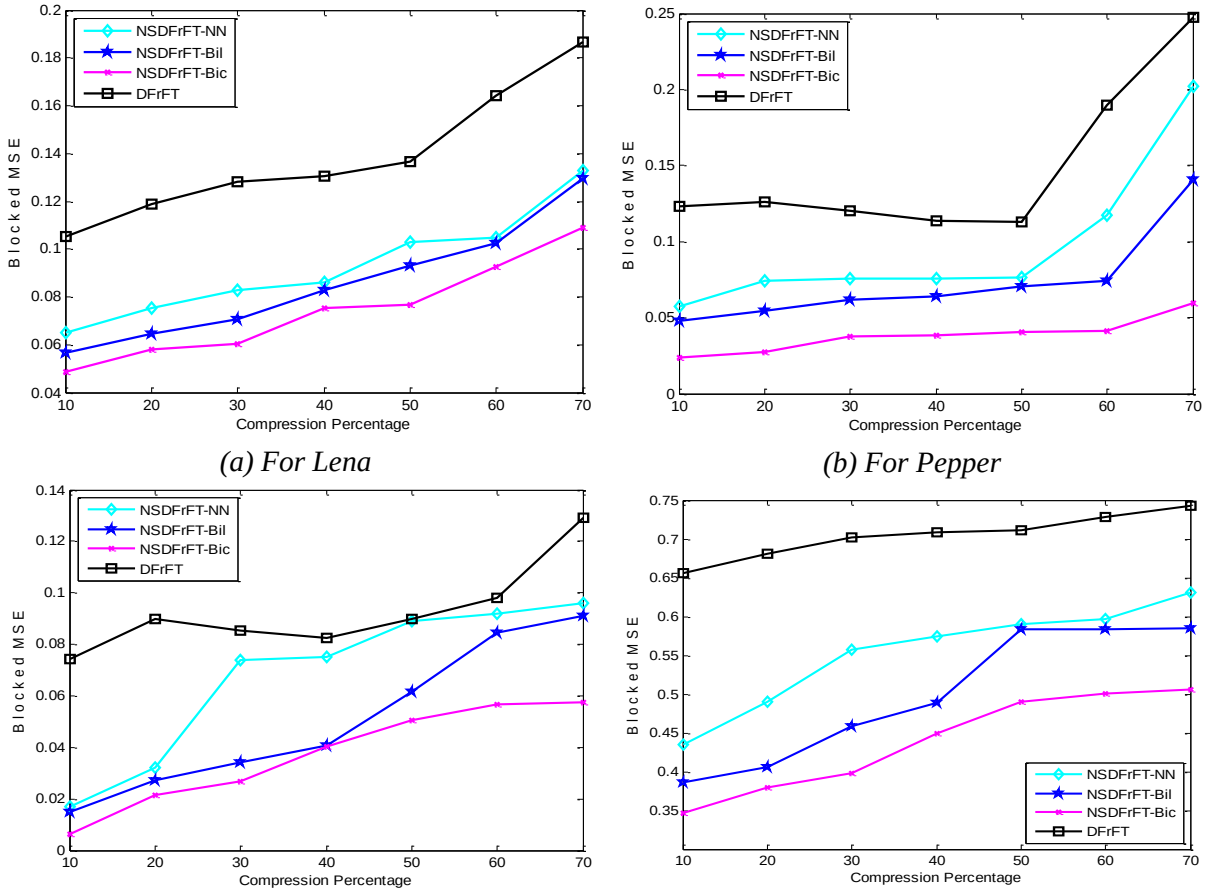
*(f). Compressed Image using DFrFT.*

***Fig. 3:*** *Compressed Images of Boat at 70% Compression for Different Variations of NSDFrFT, JPEG and DFrFT.*

***Table 3:*** *Comparison of Non-Separable Discrete Fractional Transform, Discrete Fractional Transform at 50% Compression.*

| Algorithm | Image ($512 \times 512$) | PSNR in dB | $MSE_{block}$ | CPU Time | |
|---|---|---|---|---|---|
| | | | | Encoding Time (sec) | Decoding Time (sec) |
| Non-Separable Discrete Fractional Transform | Lena | **45.59** | **0.0768** | 10.03 | 10.03 |
| | Pepper | **40.81** | **0.0405** | 8.56 | 8.56 |
| Discrete Fractional Transform | Lena | 45.20 | 0.1367 | **7.22** | **7.22** |
| | Pepper | 36.52 | 0.1128 | **5.86** | **5.86** |

The comparison of different transform techniques for different images concludes that all the definitions of NSDFrFT i.e. NSDFrFT-NN., NSDFrFT-Bil. and NSDFrFT-Bic. performs better and reduces blocking at all compression percentages. Blocking artifacts are significant for higher compression. The process of interpolation involved in the definition of NSDFrFT enables it to reduce the blocking in the reconstructed image. The process of interpolation involved in NSDFrFT definition performs the additional operation of low pass filter i.e. softening of edges or sharp transitions, enabling NSDFrFT performing better in terms of reduced blocking. However, for lower compression blocking artifacts are at minimal. While in case of DFrFT, separately post or pre-processing will be required to reduce or remove blocking artifacts in the reconstructed image. Comparison between all the definitions of NSDFrFT and DFrFT for Baboon, Boat, Lena and Pepper has been shown in Figure 4 for blocked MSE. In the comparison graph of image Baboon, Boat, Lena and Pepper, it can be clearly derived that all three definitions of NSDFrFT, viz., NSDFrFT-NN Intr., NSDFrFT-Bil Intr. and NSDFrFT-Bic Intr. have less value of $MSE_{block}$ with respect to DFrFT.




*(a) For Lena*

*(b) For Pepper*

*(c) For Baboon*

*(d) For Boat*

***Fig. 4:*** *Comparison of Fractional Transforms for Blocked MSE at Different Compression.*

## CONCLUSION

Image compression performs better in transform domain than in spatial domain, which implies when NSDFrFT and DFrFT are used as transform coding. The image has been compressed for different compression percentages and different image quality measures are used to estimate the quality of reconstructed image via NSDFrFT and DFrFT. The image quality parameters, namely, PSNR, MSE, MSSIM and GMSD suggested that the reconstructed image using NSDFrFT is of higher visual quality than the reconstructed image via DFrFT at different compression percentages. The images compressed using NSDFrFT comprises of less blocking at the boundaries of the block. All the variations of NSDFrFT, namely, NSDFrFT-Nearest Neighbor Interpolation, NSDFrFT-Bilinear Interpolation and NSDFrFT-Bicubic Interpolation resulted in less blocking artifacts in the compressed images in comparison to DFrFT. However, among all the variations of NSDFrFT implemented, NSDFrFT-Bicubic Interpolation resulted in minimum blocked MSE. An improvement in blocked MSE of about 53.34% for Lena and 74.71% for Pepper has been achieved for NSDFrFT-Bicubic Interpolation. However, the proposed algorithm comprises of computational time complexity required to encode and decode the image.

## REFERENCES


1. McIntyre KA. Dynamic Bandwidth Adaptive Image Compression/ Decompression Scheme. *U.S. Patent 7024045*. 2006.
2. Yng T L B, Lee BG, Yoo H. A Low Complexity and Lossless Frame Memory Compression for Display Devices. *IEEE Trans. Consumer Electronics*. 2008; 54 (3):1453–1458p.
3. Shukla J, Alwani M, Tiwari AK. A Survey on Lossless Image Compression Methods. *Proc. 2nd International Conference on Computer Engineering and Technology* (ICCET). 2010; 6: 136–141p.
4. Jayaraman S, Esakkirajan S, Veerakumar T. Digital Image Processing. *Tata McGraw-Hill Education*. 2011.
5. Gonzalez RC, Woods RE. Digital Image Processing. *3rd edition*. 2008.
6. Thyagarajan KS. Still Image and Video Compression with MATLAB. *John Wiley & Sons, Inc. Hoboken, New Jersey*. 2011.
7. Skodras A, Christopoulos C, Ebrahimi T. The JPEG 2000 still Image Compression Standard. *IEEE Trans. Signal Processing*. 2001; 18 (5): 36–58p.
8. Watson AB. Image Compression using the Discrete Cosine Transform. *Mathematica Journal*. 1994; 4 (1): 81–88p.
9. Jindal N, Singh K. Image and Video Processing using Discrete Fractional Transforms. *Signal, Image and Video Processing*. 2012; DOI: 10.1007/s11760-012-0391-4.
10. Singh K. Performance of Discrete Fractional Fourier Transform Classes in Signal Processing Applications. *Ph.D. Thesis, Department of Electronics and Communication Engineering, Thapar University, Patiala, India*. 2005.
11. Grgic S, Grgic M, Cihlar BZ. Performance Analysis of Image Compression using wavelets. *IEEE Trans. Industrial Electronics*. 2001; 48 (3): 682–695p.
12. Namias V. The Fractional Order Fourier Transform and its Applications to Quantum Mechanics. *IMA Journal of Applied Mathematics*. 1980; 25 (3): 241–265p.
13. Cariolario G, Ersrghe T, Kraniauskas P, Laurenti N. A unified framework for the fractional fourier transform. *IEEE Trans. Signal Processing*. 1998; 46 (12): 3206–3212p.
14. Santhanam B, McClellan JH. The DFRFT-A Rotation in Time Frequency Space. *Proc. 20th International Conference of Acoustics, Speech Signal Processing*. 1995; 2: 921–924p.
15. Pei SC, Yeh MH. Discrete Fractional Fourier Transform. *Proc. IEEE International Symposium on Circuits and Systems*. 1996; 2: 536–539p.
16. Koc A, Ozaktas HM, Hesselink L. Fast and accurate computation of two-dimensional non-separable quadratic-phase integrals. *Journal of the Optical Society of America A*. 2010; 27: 1288–1302p.
17. Ding JJ, Pei SC. Heisenberg's uncertainty principles for the 2-D nonseparable linear canonical transforms. *Signal Processing;* 2013; 93: 1027–1043p.

18. Ozaktas HM, Zalevsky Z, Kutay MA. The fractional Fourier Transform with Applications in Optics and Signal processing. *John Wiley & Sons Ltd.*, New York. 2000.
19. Almeida BL. The fractional Fourier transform and time–frequency representations. *IEEE Trans. Signal Processing*. 1994; 42: 3084–3091p.
20. Sahin A, Kutay MA, Ozaktus HM. Nonseparable two-dimensional fractional Fourier transform. *Journal of the Optical Society of America A*. 1998; 37 (23): 5444–5453p.
21. Richardson IEG. H.264and MPEG-4 Video Compression, John Wiley & Sons. *Inc. West Sussex, England*. 2003.
22. Xue W, Zhang L, Mou X, Bovik AC. Gradeint Magnitude Similarity Deviation: An Highly Efficient Perceptual Image Quality Index. *IEEE Trans. Image Processing*. 2014; 23 (2): 684–695p.
23. Whang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Processing*. 2004; 13 (4): 600–612p.
24. Parker JA, Kenyon RV, Troxel DE. Comparison of Interpolating Methods for Image Resampling. *IEEE Trans. Medical Imaging*. 1983; MI-2 (1): 31–39p.
25. Jindal N. Performance of Fractional Transforms in Image and Video Processing. *Ph.D Thesis, Department of Electronics and Communication Engineering, Thapar University, Patiala, India*. 2015.
26. Chen CC. On the selection of image compression algorithms. *Proc. IEEE 14th International Conference on Pattern Recognition* (ICPR'98). 1988: 1–5p.

**Cite this Article**
Kritika Mittal. Image Compression via Non-Separable Discrete Fractional Fourier Transform. Research & Reviews: Discrete Mathematical Structures. 2015; 2(3): 1–9p.