# Reduction of Blocking Artifacts in Compressed Images using Non-Separable Fractional Fourier Transform

Dissertation submitted towards the partial fulfillment of the requirement for the award of the degree of

Master of Engineering

in

**Wireless Communication**


Submitted by:

**Kritika Mittal**

**Roll No: 801363018**

Under the guidance of:

**Dr. Kulbir Singh**

**Associate Professor, ECED**



**ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT**

**THAPAR UNIVERSITY**

**PATIALA – 147004 (PUNJAB)**


**JULY 2015**

# DECLARATION

I, Kritika Mittal, hereby certify that the work which is being presented in this thesis entitled **"Reduction of Blocking Artifacts in Compressed Images using Non-Separable Fractional Fourier Transform"** by me in partial fulfillment of the requirements for the award of degree of Master of Engineering in Wireless Communication from Thapar University, Patiala, is an authentic record of my own work carried under the supervision of **Dr. Kulbir Singh** (Associate Professor), **ECED.**

The matter presented in this thesis has not been submitted in any University/Institute for the award of Master of Engineering.

Date: 07/07/2015

Kritika Mittal

Roll No. 801363018

It is certified that the above statement made by the student is correct to the best of my knowledge and belief.

Date: 07/07/2015

Dr. Kulbir Singh

Associate Professor, ECED

Thapar University, Patiala

Countersigned by:

Dr. Sanjay Sharma

Professor and Head ECED

Thapar University, Patiala

Date:

Dr. S.S. Bhatia

Dean of Academic Affairs

Thapar University, Patiala

Date:

# ACKNOWLEDGEMENT

First of all, I would like to thank GOD Almighty, for showering his blessings upon me during the complete work duration and providing me with courage, patience and ability to work against all odds.

Another thanks from the bottom of my heart to my supervisor Dr. Kulbir Singh, Associate Professor, Department of Electronics and Communication, Thapar University, Patiala for all his untiring efforts as a teacher to support me in bringing the work in the present shape. His healthy criticism always encouraged me to work hard and made me learn my lessons. I feel honored to be working under his supervision.

I am also very thankful to the Head of the Department, Dr. Sanjay Sharma and P.G. Coordinator, Dr. Amit Kumar Kohli for providing the supporting facilities for the completion of the work. I also extend my thanks to Navdeep Goel, Assistant Professor, Yadavindra College of Engineering, Talwandi Sabo and Dr. Neeru Jindal, Assistant Professor, Department of Electronics and Communication, Thapar University, Patiala for helping me in my work. I am also thankful to the staff and faculty members of Electronics and Communication Department for their encouragements.

I thank all my friends for their support and encouragement that provided me with the strength to complete the work with full dedication.

I extend my heartiest thanks to my maternal uncle, aunt and my parents for their immense love, support, encouragement and selfless help in the complete tenure of my work. They stood behind me with their blessing in every hardship of my work. At last my appreciations for helping, encouraging and standing by my side during the course of my work to my siblings, Amulya Agarwal and Ashish Mittal.

(Kritika Mittal)

# ABSTRACT

Image compression has become an important aspect for many multimedia applications to fulfill the need of processing image for storage space, transmission bandwidth and representation with reduced cost. Image compression enables the autonomous machine to represent image utilizing less bits. Block-based compression algorithm with different transform techniques had been long used to compress image in a lossy manner. However, the reconstructed images from compression produces annoying blocking artifacts near the block boundaries, particularly in highly compressed images, as each block is transformed and quantized independently. Several techniques or algorithms have been proposed by researchers, both in spatial and frequency domains, for reduction of these blocking artifacts with varied degree of success.

An image compression algorithm using Non-Separable Discrete Fractional Fourier Transform (NSDFrFT) employing any one of the suggested Bicubic interpolation and Nearest Neighbor interpolation apart from Bilinear interpolation as a transform technique has been proposed. The algorithm divide image into sub-images known as blocks, processing each independently. The NSDFrFT resulted in less MSE in blocked region than image compression using DFrFT and JPEG. However, if the different variations of NSDFrFT given as NSDFrFT-Nearest Neighbor Interpolation, NSDFrFT-Bilinear Interpolation and NSDFrFT-Bicubic Interpolation are compared then NSDFrFT-Bicubic Interpolation performs better than the other two variations. The performance of the transform techniques has been analyzed using various image quality metrics (IQM) among which GMSD is faster to calculate and provides high predictive accuracy. Thus, relying on accuracy of simulations, NSDFrFT results in structurally similar high subjective quality reconstructed image with reduced blocking for low frequency images at high compression percentages.

# CONTENTS

# LIST OF ABBREVIATIONS

| | |
|---|---|
| APCM | Adaptive Pulse Code Modulation |
| Bic Intr. | Bicubic Interpolation |
| Bil Intr. | Bilinear Interpolation |
| CP | Compression Percentage |
| DCM | Differential Code Modulation |
| DCT | Discrete Cosine Transform |
| DFT | Discrete Fourier Transform |
| DFrCT | Discrete Fractional Cosine Transform |
| DFrFT | Discrete Fractional Fourier Transform |
| DPCM | Differential Pulse Code Modulation |
| DWT | Discrete Wavelet Transform |
| FT | Fourier Transform |
| FFT | Fast Fourier Transform |
| FrFT | Fractional Fourier Transform |
| GMSD | Gradient Magnitude Similarity Deviation |
| JPEG2000 | Joint Photographic Experts Group2000 |
| MSDS | Mean Squared Difference of Slope |
| MSE | Mean Square Error |
| MSSIM | Mean Structural Similarity Index Measure |
| NN Intr. | Nearest Neighbor Interpolation |
| NSDFrFT | Non-Separable Discrete Fractional Fourier Transform |
| NSLCT | Non-Separable Linear Canonical Transform |
| PSNR | Peak Signal to Noise Ratio |

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

This chapter includes the exordium of fractional transform, image compression techniques and the blocking artifact knot.

## 1.1 PROLOGUE

An image is a two dimensional (2D) grid-form signal representation of physical things that are continuous in time and space. Mathematically, an image can be represented as $i(p, q)$ where, $p$ and $q$ locates the position of light intensity falling on a plane, popularly known as pixel [1]. An image can be of various types extending from black & white image to RGB plane image, X-ray image to grey image, ultrasound image to infrared image. These various types of images can be of nature, heat maps, satellite, human, ocean, microscopic, telescopic, sports, etc.

Image processing has become an important matter as now-a-days lots of data are represented in graphics. Considerable amount of storage space and transmission bandwidth is needed by any uncompressed multimedia (graphics, audio and video) [2]. The capabilities of available technologies faces challenges in-terms of storage space and bandwidth despite rapid progress in the area of storage space, speed of processors, performance of digital communication systems and density of mass storage. The recent growth of web applications consisting intensive multimedia, have insisted on encoding and decoding of signals and images in an efficient manner. Thus, to handle such a large amount of data, practically efficient and planned methods are needed. Image compression is an apprehensive solution to this challenge [3].Image compression is a science that reduces the number of bits or amount of data required for image representation.

Researchers strive hard to search for various techniques which can make image compression a competing process, yielding images of small size with good visual quality. People using digital camera, internet for uploading and downloading pictures, DVDs for watching movies and listening to MP3 songs are benefitted by the

algorithms of image compression [4]. Image compression finds application in satellite processing, medical imaging, remote sensing, preserving of art works, etc [5].

## 1.2 ESSENTIALS OF COMPRESSION

Compression means reduction in amount of data required to represent image. The amount of data required to represent same amount of information can vary, as information (image) comprises of some data which repeats itself or is irrelevant in nature. Such data is known as redundant information or irrelevant data [6]. If $k_1$ and $k_2$ are taken into consideration, as two different number of bits for representation of same information then compression is said be given as

$$C = k_1/k_2 \tag{1.1}$$

And relative redundancy of $k_1$ data bits is given as

$$R = 1 - 1/C \tag{1.2}$$

The dependence of redundancy on compression can be analyzed for three different cases as-

$CASE\ 1: IF\ k_1 = k_2\ THEN\ C = 1\ AND\ R = 0$

$CASE\ 2: IF\ k_1 \gg k_2\ THEN\ C \to \infty\ AND\ R \to 1$

$CASE\ 3: IF\ k_1 \ll k_2\ THEN\ C \to 0\ AND\ R \to -\infty$

According to $CASE\ 1$ when both representations of information are identical, there is no relative redundancy among the data. In $CASE\ 2$ the number of bits in first representation is more than the other which tends compression to increase towards infinity and contributes towards high relative redundancy. $CASE\ 3$ considers that the representation of second data has more number of bits, resulting in reduced compression and least amount of relative redundancy.

In case of images, pixels are correlated to neighboring pixels, thus this redundant data of image can be eliminated without affecting much of image quality [5]. Moreover, image comprises of some pixel values which are not visible to the naked human eye, thus are irrelevant data and can be deducted without loosing much on the image visual

quality. Any image suffers from major three kinds of redundancies, namely, Coding Redundancy, Spatial Redundancy and Irrelevant Information.

## 1.3 IMAGE COMPRESSION GENRE

Image compression can be classified in many ways. Two such classifications have been discussed here.

### 1.3.1 LOSSLESS VS LOSSY COMPRESSION

This categorization is based on the amount of information required for the reconstructed image. When the image compression algorithm results in compressed image identical to the original image, then such an algorithm is said to be lossless image compression algorithm. Lossless image compression is also known as reversible compression as all information remains intact with the reconstructed image similar to that of the original image [7]. Therefore, it is generally used in applications where any loss of information is not acceptable such as medical imaging, technical drawings, preserved art-works, etc But only minimal amount of compression is achieved and thus, lossy compression is popular to attain higher extend of compression [6].

Lossy image compression discards significant amount of irrelevant information, thus, degrading the quality of reconstructed image in comparison to original image. This type of image compression is also known as irreversible compression because useless information is removed during the compression process, thus, some information of image is lost. These algorithms find application where minor loss of information is affordable such as photographic images, internet uploads, etc So, it can be assumed that high compression is achieved with no visual impairments at normal viewing conditions.

### 1.3.2 PREDICTIVE VS TRANSFORM CODING

This categorization is based on the application of compression algorithm on a space or region or area of interest. Predictive coding, compresses the information based upon the prediction of future values utilizing previously known values. It is generally applied in the spatial domain or upon pixels [8]. So, it has minimum computational

complexity and local image characteristics adapt it readily. Differential Pulse Code Modulation (DPCM), Adaptive Pulse Code Modulation (APCM), Differential Code Modulation (DCM), etc are some examples of predictive coding.

On the other hand, transform coding converts the image from spatial domain to transform domain using various transform techniques. This method achieves high compression in comparison to predictive coding but at the cost of large computational complexity. Different transform techniques are Discrete Fourier transform (DFT), Discrete Cosine transform (DCT), Discrete Fractional Fourier Transform (DFrFT), Discrete Fractional Cosine Transform (DFrCT), Non-Separable Fractional Fourier Transform (NSFrFT) etc.

## 1.4 BLOCKING ARTIFACTS

A compression artifact is a detectable distortion of the multimedia including images, audio and video. As discussed earlier, lossy image compression results in high compression but some visual impairment are caused. These compression artifacts may occur at any media such as MP3 or MP4 or JPEG or MPEG files. The detection, reduction or removal of the compression artifacts is the main goal behind implementing an artifact removal or reduction algorithm [5].

Blocking occurs when image undergoes transform coding where image is converted from spatial to transform domain. The neighboring block boundaries having correlation among them are not taken into consideration in block based transform techniques. As a result, the adjacent blocks boundaries become visible causing blocking artifacts on reconstructing the decoded image [9]. Thus, visual quality of an image can be increased by reducing these blocking artifacts. So, detecting and reducing blocking artifacts in reconstructed images of compression process is an critical issue.

## 1.5 FRACTIONAL TRANSFORM IN IMAGE COMPRESSION

Among various techniques used in transform coding, fractional transform proved to be quite efficient. Fourier transform (FT) on generalization results into Fractional Fourier Transform (FrFT). FrFT has come up as an important tool in the field of signal processing and image processing.

The well-known and widely used FT was first given by Jean-Baptiste-Joseph Fourier [10]. FT finds application in signal analysis, optics, image, physics, statistics, acoustics, antenna and array processing. But FT proved to be inadequate for non-stationary signals, thus leading to the emergence of FrFT. The first introduction of fraction into the definition of FT was done back in year 1929 and lead to development of FrFT [11]. Without any knowledge of work done in 1929, Namias [12] gave the definition of FrFT, the generalized definition for FT as a transform technique. Ozaktas and Mendlovic [13] gave numerous definition of FrFT equivalent to each-other. For various types of signals (1D, multi-D, periodic, aperiodic, discrete and continuous) Cariolario *et al.* [14] gave FrFT. Pei *et al.* [15]-[16] studied FrFT in detail considering many different aspects.

An era of computers leads to development of DFT to compute FT digitally. Thus, working on the same line of thought, digital version of FrFT was needed to process the signals in discrete form. Use of digital devices like computers and DSP processors increased with advancement in technology. Keeping in mind the advancement in technology, the definition of DFrFT was given by McClellan *et al.* [17]. However, many other definitions of DFrFT [18]-[19] also came up but none of them was able to satisfy all the properties of continuous FrFT [15]. Non-Separable definition of FrFT is a generalized case of Separable FrFT. In non-separable definition both the axes are rotated simultaneously to achieve rotation in both directions [20]. NSDFrFT can also be derived from the Non-Separable Linear Canonical Transform (NSLCT) definition.

In earlier times most of the applications of FrFT were found in the field of optics and signal processing. FrFT has inspired the study of fraction introduction in various transforms used in signal and image processing. But with advancements in research, it has found application in solving differential equations, optical beam propagation, quantum mechanics, statistical optical, watermarking, compression, encryption, signal detectors, space variant filters, correlation and pattern recognition, restoration, etc [21]. However, NSDFrFT have been used for the purpose of filtering the noise intact in the images and optics so far [20].

## 1.6 ORGANIZATION OF THESIS

This work comprises of following chapters:

CHAPTER 1: INTRODUCTION, discusses about need, principle and elements of image compression with an introduction to images, types of image compression, fractional transform, blocking artifacts and aim behind research.

CHAPTER 2: LITERATURE SURVEY, briefs about the developments made in image compression algorithms, blocking artifacts reduction and removal techniques and types of fractional transform.

CHAPTER 3: INTERPOLATION, discusses about non-adaptive and adaptive techniques and algorithms for various applications.

CHAPTER 4: IMAGE COMPRESSION USING VARIOUS TRANSFORM TECHNIQUES, briefs about the techniques used to compress an image along with the simulation results and comparison are done among various transform techniques.

CHAPTER 5: REDUCTION OF BLOCKING ARTIFACTS, briefs about the extent of compression artifact known as blocking artifact found in the proposed algorithm along with the comparison with other techniques.

CHAPTER 6: CONCLUSION AND FUTURE SCOPE, sums up the results concluded from the proposed algorithm and implementation of other algorithms and discusses the possibility of future work.

# CHAPTER 2

# LITERATURE SURVEY

This chapter discusses about the precis of the research work, regarding image compression, fractional transform in image compression and blocking artifacts.

## 2.1 IMAGE COMPRESSION

JPEG in 1987 studied the subjective quality of image using blind assessment based selection process. This study suggested that best image quality is achieved by $8 \times 8$ DCT, concluding that statistical form of data compression gives better results [22]-[23]. Thus, at the encoder end, the input image is divided into $8 \times 8$ blocks and processed by Forward DCT. Similarly, at decoder end, the image is processed using Inverse DCT and $8 \times 8$ blocks are aggregated to achieve the reconstructed image.

ISO/CCIT group indulged in developing an international image compression algorithm which is known as JPEG, whose overall structure was given by Wallace in 1990 [23]. In 2001 [24], JPEG2000 standard was studied using the wavelet coding techniques. The main technique in focus was wavelet transform and not the JPEG standard itself. Wavelet coders use bi-orthogonal filters which do not preserve energy rather than orthogonal filters which preserve the energy. In 2001 Yetik *et al.* [25] presented a FDD filtering representation of images along with image compression algorithm. In fractional Fourier filtering initially, input image is processed using FrFT then multiplied with the filter function and the resultant is processed by Inverse Fractional Fourier Transform (IFrFT). Multi-channel and multi-stage filtering has two unknowns namely, fractional Fourier transform orders and filter coefficients. These unknowns are estimated by the use of minimum mean square approach. The implementation cost of FrFT and FT is same, thus, FrFT provides faster implementation of system in space [25]. As the compression extend increases so does the error.

In 2006, DFrFT combining with Set Partitioning In Hierarchical Tree (SPIHT) scheme was used for signal compression [26]. High compression was achieved with

better results than DCT when tested for different types of signals. In 2008, Cebrail *et al.* [27] analyzed fidelity criteria, coding repetitions for different compression approaches. Meanwhile, different techniques to implement image compression came up such as using neural networks, wavelet transform and phase information [28]. Image compression was performed using Artificial Neural Network (ANN) considering visual features and information embedded in the image [29]. This algorithm works in a lossy way causing high compression but with preserved characteristic data with or without quantization and median filter. Lossy image compression utilizing phase information with reduced storage requirements can regain an original image adaptively using spectral phase information only. In fact, the number of quantization bits had been varied adaptively based on spectral phase information to achieve high compression.

In 2012, Li *et al.* [30] discussed the wavelet, a multi-resolution based image compression which overcomes the DCT drawbacks. In 2013, Jindal *et al.* [31] used discrete fractional transforms to compress as well as encrypt the image. In the same year, Jindal *et al.* [32] gave an image retrieval algorithm using discrete fractional transforms.

## 2.2 BLOCKING ARTIFACTS

In order to reduce the computational complexity of the image compression algorithm, an image is divided into $n \times n$ sub-images known as blocks, where $n = 2^l$ but results in image impairment known as blocking artifacts [5].The algorithms for blocking artifacts reduction can be classified into pre-processed and post-processed methods. Pre-processed methods are applied in the spatial domain before encoding. While in post-processing, the technique can be applied in both transform domain and spatial domain as it is done on reconstructed image at decoding end.

### 2.2.1 DETECTION AND REDUCTION ALGORITHMS

In 1983, Reeve *et al*. [33] proposed two methods to reduce the effect of blocking artifacts. Firstly, method in which an image is segmented into sections containing some pixels known as an overlap method. When merging these regions to recreate the image slight overlap around the perimeter can reduce blocking artifacts. Secondly, the

filtering method, a $3 \times 3$ Gaussian spatial filter can be applied at pixels adjacent to sub-image boundaries which reduces the blocking artifacts with least damage to the content of the image. Zakhor [34] in 1992, gave an iterative block reduction technique which starts with low pass or band limit filtering of the image which has horizontal and vertical high frequency components. Then $N \times N$ image is divided into sub-images and DCT of each sub-image is taken independently, followed by quantization. The proposed algorithm uses $3 \times 3$ low pass filters which reduce blocking artifacts and excessive blurring.

Later in 2000, Paek *et al.* [35] studied high frequency components at two stages as blocking artifacts are the result of the high frequency component in the image. At first stage, two adjacent homogenous blocks from the boundary of block are found. Then at the second stage, the local frequency characteristics in the homogeneous block are studied through DCT. For two homogeneous blocks of different sizes the relation between DCT coefficients is derived, through which identification of high frequency components is done. An algorithm to detect and reduce the effect of blocking in JPEG compressed images was given by Singh *et al.* [36], in 2007. Visual perception along the block boundary provides assistance in detection of blocking artifacts modeled as 2D step function. The regions between block boundaries are either smooth or non-smooth regions. The blocking artifacts in the smooth region are removed by modifying DCT coefficients while the blocking artifacts in the non-smooth region are removed by applying an edge preserving smoothing filter. Liew *et al.* [37] designed the algorithm which uses the fact that the block discontinuities are constrained by the DC quantization interval of the quantization table. Since artifacts mainly occur at the strong edges, which can be found at block boundaries, thus suppression of block discontinuities do not always reduce blocking artifacts. The algorithm reduces blocking artifacts and ringing effects while preserving edges and texture information.

Lee *et al.* [38] proposed an algorithm which was used to detect the corner outlines and then applying thresholding on the coefficients blocking is reduced. A $3 \times 3$ median filter was applied on the reconstructed image and the image quality metrics used for image quality determination viz PSNR, MSE, MSSIM were also improved. Singh *et al.* [39] proposed a detection and reduction algorithm for blocking artifacts using a de-blocking filter in the four modes of attack. These modes include a corner mode and

three frequency based modes namely, smooth, non-smooth, intermediate. The main feature of this algorithm came out to be its reduced computational complexity with preserved and enhanced image quality. In later 2012, Purushothaman *et al.* [40] gave an adaptive non-linear algorithm based on smooth and edgy regions detected by binary edge map. This algorithm reduces blocking artifacts along with preserving the fine details and edges of the image at low-bit rates. Pandey *et al.* [41] in 2015, studied block wise DCT based image compression and the quantization matrix values was modified to achieve reduced blocking artifacts in the reconstructed image.

### 2.2.2 REMOVAL ALGORITHMS

Apart from various detection and reduction techniques, several algorithms to eradicate blocking from the compressed images were also given. Hsu *et al.* [42] proposed an adaptive separable median filter which adaptively transforms from a traditional median filter to a low-pass filter progressively when the filter is close to the position of blocking. This filter gives the advantage of edge preserving and noise reduction. In the conventional transform coding such as JPEG standardized coding, Minami *et al.* [43] suggested a new approach without introducing extra information or blurring to be applied. This technique exploits the correlation between the pixels intensity values at the block boundary. Due to the quantization of DCT coefficients, the expected value of Mean Squared Difference of Slope (MSDS) between the slope across the adjacent blocks and average between the boundary slopes of each of the two blocks is increased.

Meier *et al.* [44] proposed an algorithm which segments the degraded image and at each segment low pass filter is applied. This prevents blurring of edges as low pass filter is never applied at the segmented region boundary. The algorithm result depends greatly upon the segmentation method and the type of filter used. Chou *et al.* [45] estimated the maximum likelihood of quantization noise to be formed to distinguish possible artificial edges from natural edges. Then by applying one time non-linear smoothing artificial edges are removed. Luo *et al.* [46] has taken two properties into consideration. First, for smooth region within the block the original pixel level provides continuity. And second, the correlation between the neighboring blocks is utilized to reduce the discontinuity of pixel across the boundaries. Utilizing these two properties, the blocking removal algorithm enhances the visual quality of image.

Singh *et al.* [47] gave a new adaptive post-processing algorithm for removing blocking artifacts in the block based DCT compressed images. The blocking artifacts in the smooth and non-smooth region were removed by modifying DCT coefficients and an edge preserving smoothing filter is applied at the intermediate region. In 2013, Dandiwal *et al.* [48] summarized and analyzed the performance of various post-processing blocking removal algorithms given till date in spatial domain. The analysis suggested that the algorithm given by Chou [45] produced good quantative results.

## 2.3 FRACTIONAL TRANSFORM

In $17^{th}$ century, Bernoulli questioned the meaning of non-integer order derivative, giving rise to the concept of fractional orders. Later in 1929, the concept of fractions was introduced using FT [11]. Initially, fractional calculus concept was applied in the field of mathematics and optics only. In 1930 H. Weyl worked upon FrFT definition who was followed by others namely, in 1937 by E. U. Condon, in 1939 by H. Kober, in 1956 by A. P. Guinanad, in 1959 by A. L. Patterson, in 1961 by V. Bargmann, in 1973 by De Brujin and lastly in 1974 by R. S. Khare [49].

### 2.3.1 SEPARABLE FRACTIONAL FOURIER TRANSFORM

The first mathematical concept of fractions in the definition of FT, with ordinary FT at an order of 1 was introduced by Namias [12] completely unaware of the work done earlier in 1929. McBride *et al.* [50] considered the application of modified fractional operators and their proved theorems on ordinary differential equation. Almeida [51] discussed FrFT as a signal which changes from time to frequency while changing fractional order from 0 to 1. FrFT when defined as a linear integral transform kernel in the one-dimensional (1D) then FrFT of $f(x)$ is defined as [51]

$$F^a\big(f(x)\big) = \int_{-\infty}^{\infty} K^a(x, x')f(x)dx = f^a(x') \tag{2.1}$$

According to this definition $F^a\big(f(x)\big)$ transforms $f(x)$ into $x'$ domain such that $\alpha = a\pi/2$. The kernel of FrFT is given as

$$F^a\big(f(x)\big) = \sum_{n=0}^{\infty} e^{-jn\alpha} H_n(x)H_n(x') \tag{2.2}$$

$H_n(x)$ is the normalized $n$ th-order Hermite function with $\sigma$ as unit variance. Mathematically,

$$H_{\sigma,n}(x) = \frac{1}{\left(2^n n!\,\sigma\sqrt{\pi}\right)^{\frac{1}{2}}} h_n\left(\frac{x}{\sigma}\right) e^{-\left(\frac{t^2}{2\sigma^2}\right)} \tag{2.3}$$

where, $h_n(.)$ is the Hermite polynomial of $n$ th-order. The Hermite-Gaussian function has the same eigen-function with similar eigenvalues and eigenvectors as that of Fourier Transform. In accordance with this reason, the Hermite-Gaussian function is considered to be the FrFT decomposition kernel. The inverse of the FrFT can be obtained simply by replacing $-a$ which is actually a backward angle. Mathematically, it can be given as

$$F^{-a}\big(f(x')\big) = \int_{-\infty}^{\infty} K^a(x,x') f(x') dx = f^{-a}(x) \tag{2.4}$$

For different values of $a$, FrFT of a signal is considered to be rotated at arbitrary angles on the time-frequency plane.

**Table 2.1: Convergence of FrFT**

| $a$ | $\alpha$ | Operator | Operation performed on Signal |
|---|---|---|---|
| 0 and 4 | 0 and $2\pi$ | $F_0 = I$ | Identity Operator |
| 1 | $\pi/2$ | $F_1 = F^1$ | Fourier Transform |
| 2 | $\pi$ | $F_2 = P$ | Inverse Identity Operator |
| 3 | $3\pi/2$ | $F_3 = F^{-1}$ | Inverse Fourier Transform |

For the analysis of 2 dimensional (2D) signals, the expression for FrFT is given as [52]

$$F^a\big(g(x,v)\big) = F^{a_x,a_v}\big(f(x,v)\big) = \iint K^{a_x,a_v}(x,v;x',v') f(x',v') dx dv \tag{2.5}$$

where,

$$K^{a_x,a_v}(x,v;x',v') = K^{a_x}(x,x') K^{a_v}(v,v') \tag{2.6}$$

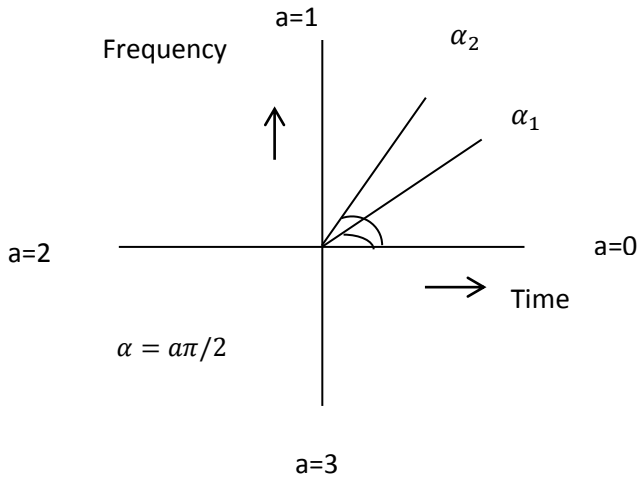



**Figure 2.1: FrFT in time-frequency plane**

In 1997 Kutay *et al.* [53] gave the digital computation of FrFT which disobeyed additivity property. While developing Discrete Fractional Fourier Transform (DFrFT), it was essential that it closely follows the property of continuous FrFT. Initial definitions of DFrFT were made by linearly combining the four signals namely, original signal, its circularly flipped signal, its DFT and circularly flipped signal of DFT [52]. The results of these definitions were not close to continuous FrFT. In 1996 Ozaktas [13] gave an accurate and efficient FrFT computation simply by factor 2 oversampling without involving any additional cost as in case of ordinary transform. The discrete counterpart was also defined. In the same year, McClellan *et al.* [17] defined DFrFT by digitally computing FrFT which follows rotational property but not angle addition property of FrFT. Later Cariolario [14] gave definition for different types of signals namely, one dimensional and multi-dimensional signals, periodic and aperiodic signals and continuous and discrete signals.

The simplest manner to define DFrFT is to sample FrFT but this result in DFrFT which do not follow properties of FrFT at all. Candan *et al*. [19] proposed generalized DFT as DFrFT based on the particular set of eigenvectors of DFT, which comprises of discrete counterpart of Hermite-Gaussian functions. $X(k)$, the DFT of the sequence of data $x(n)$ is given as [54]

$$X(k) = \sum_{n=0}^{N} x(n) e^{\frac{i2\pi kn}{N}} \tag{2.7}$$

And Inverse Discrete Fourier Transform (IDFT) is given as

$$x(n) = \frac{1}{N}\sum_{k=0}^{N} X(k)e^{-\frac{i2\pi kn}{N}} \tag{2.8}$$

The DFrFT obtained satisfactorily follows unitary, index additive and for unit order reduces to DFT. Utilizing the Hermite-Gaussian function the transform kernel to calculate DFrFT is given as

$$F^{2\alpha/\pi} = U^{\wedge} D^{2\alpha/\pi} U^{\wedge T} \tag{2.9}$$

$$= \begin{cases} \sum_{k=0}^{N-1} e^{-jk\alpha} u_k^{\wedge} u_k^{\wedge T}, & \forall N = 4m+1 \\ \left(\sum_{k=0}^{N-2} e^{-jk\alpha} u_k^{\wedge} u_k^{\wedge T}\right) + e^{-jk\alpha} u_k^{\wedge} u_k^{\wedge T}, & \forall N = 4m \end{cases}$$

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 eigenvector of the k-th order discrete Hermite function and D is defined as

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

and (2.10)

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

In 2000 Pei *et al.* [18] proposed a new DFrFT which has DFT eigenvalues and eigenvectors. The eigenvectors were obtained by two orthogonal projection method. The obtained eigenvectors of the S matrix was made orthogonal using Gram Schmidt procedure in accordance to the first orthogonal method. The least Frobenius norm between the samples of Hermite function and the orthogonal DFT Hermite eigenvectors was obtained by using Orthogonal Procrustes Method (OPM). Hanna *et al.* [55] formulated a technique for generating orthonormal eigenvectors from F

(twiddle matrix of DFT) using SVD method of orthogonal projection matrices with Gram Schmidt Orthogonalization (GSO), Orthogonal Procrustes Method (OPM) and Sequential OPA (SOPA). Santhanam *et al.* [56] defines DFrFT based on the centered version of DFT as the eigenvectors are derived using the Grunbaum's tri-diagonal commutator which approximates the Hermite-Gauss eigenvectors very closely. Serbes *et al*. [57] proposed a new DFrFT using centered DFT twiddle matrix. This CDFT is used to determine the projection matrices which are Hermite-Gauss like eigenvectors of the CDFT corresponding to each eigenvalues of CDFT. Singh *et al.* [58] reviewed the existing methods of DFrFT classifying them into classes based upon the methodology used. Comparison of various classes concluded that eigen-decomposition was the best class of definition for DFrFT. However, none definition of DFrFT satisfies all the properties of FrFT till date, creating scope for further improvements in the definition.

### 2.3.2 NON-SEPARABLE FRACTIONAL FOURIER TRANSFORM

Non-Separable Fractional Fourier Transform (NSFrFT) is a generalization of FrFT and a special case of Non-Separable Linear Canonical Transform (NSLCT) [59]. The class of non-separable transforms can represent a large number of systems such as non-orthogonal systems, anamorphic systems and astigmatic systems. The non-separable case represents the two dimensions coupled to each other with the help of additional four cross-parameters. A methodology to represent NSFrFT from NSLCT has been discussed by Ding *et al.* [21] by working upon the values of the ABCD parameters of the NSLCT definition. The definition of Non-Separable Linear Canonical Transform (NSLCT) [21] considering that $\det(B) \neq 0$ is given as

$$K(u,v) = \left(2\pi\sqrt{-\det(B)}\right)^{-1} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp[j(k_1u^2 + k_2uv + k_3v^2)/2\det(B)]$$
$$\exp\left[\frac{j\big((-b_{22}u + b_{12}v)x + (b_{21}u - b_{11}u)y\big)}{\det(B)}\right] \exp\left[\frac{j(p_1x^2 + p_2xy + p_3y^2)}{2\det(B)}\right] g(x,y)dxdy \quad (2.11)$$

where, A, B, C, D are $2 \times 2$ matrix representing 16 parameters of NSLCT

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},\ B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix},$$

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \text{ and } D = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \tag{2.12}$$

$K^{(A,B,C,D)}(u,v)$ is the output obtained [21] and

$$k_1 = d_{11}b_{22} - d_{12}b_{21} ,$$

$$k_2 = 2(-d_{11}b_{12} + d_{12}b_{11}), \tag{2.13}$$

$$k_3 = -d_{21}b_{12} + d_{22}b_{11} ,\ p_1 = a_{11}b_{22} - a_{21}b_{12}$$

$$p_2 = 2(a_{12}b_{22} - a_{22}b_{12}), p_3 = -a_{12}b_{21} + a_{22}b_{11}$$

The equation (2.11) should satisfy the following constraints-

$$A^T C = C^T A , B^T D = D^T B , A^T D - C^T B = 1 \tag{2.14}$$

NSFrFT, a generalized definition of FrFT, can be obtained by substituting

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

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

in the eq. (2.11), the definition of NSLCT.

In 1998, Sahin *et al.* [20] proposed a definition of NSFrFT along with its discrete version depicting its usefulness by optical and image-noise removal implementation. When the linear distortion is applied on the function, the 2D FrFT cannot be represented as the linear distorted signal of the original signal. If $f(x,y)$ has a FT given by $F(x,y)$ then $f(ax+by, cx+dy)$ has FT given as

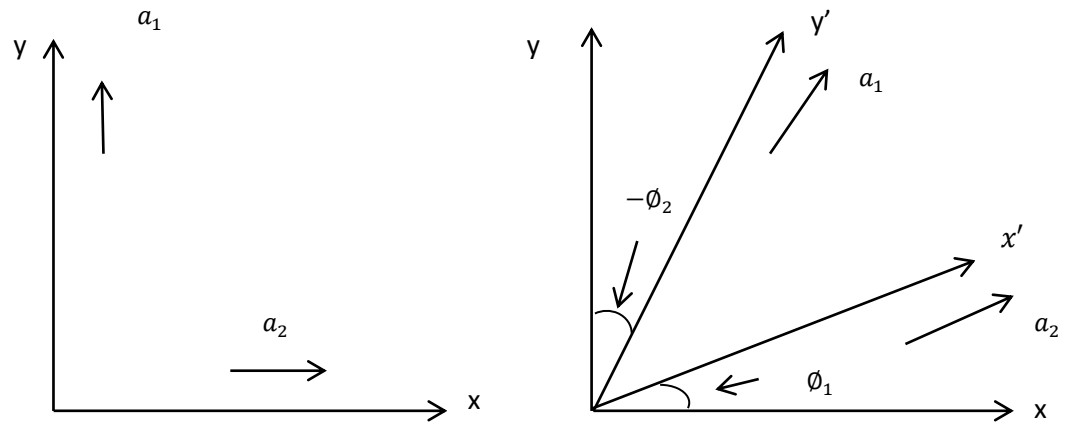
$$G(x,y) = \frac{1}{\partial} F\left(\frac{dx - cy}{\partial}, \frac{-bx + ay}{\partial}\right) \tag{2.15}$$

where,

$$\partial = ad - bc$$

Utilizing this fact, a non-separable definition was given which can specify transform orders in arbitrary directions without any hindrance from orthogonal x and y axes.

This process of rotation is equivalent to mapping $f(x,y)$to $f[(cos\emptyset_1 x + sin\emptyset_1 y)/\cos(\emptyset_1 - \emptyset_2), (-sin\emptyset_2 x + cos\emptyset_2 y)/\cos(\emptyset_1 - \emptyset_2)]$. This mapping is achieved by the use of Bilinear Interpolation (Bil Intr.). Thus, NSDFrFT has four main parameters of vital importance $a_1$, $a_2$, $\emptyset_1$and $\emptyset_2$. In 2015, Ran *et al.* [60] implemented NSFrFT along with chaotic maps to achieve image encryption. Highly secure and valid image cryptosystem has been achieved using this technique.




**(a)** **(b)**

**Figure 2.2: Time-Frequency plane rotation for (a) separable FrFT (b) NSFrFT**

The non-separable transforms have an important necessity for interpolation to reverse rotated grid samples to the original grid which is in the shape of rectangle. However, an important fact to be noted is that no such substantial impact in separable case has been observed [59]. Thus, the errors occurred greatly rely upon the accuracy of the interpolation algorithm used. In some cases, customized interpolation algorithms for non-rectangular grids are required to further enhance the accuracy of the non-separable definition.

## 2.4 MOTIVATION

The literature of image compression is full of techniques and methodologies to achieve a high quality reconstructed image at the receiver's end. The thrilling history of Fractional Transforms created interest for its implementation. Though now-a-days Fractional Transforms have gained popularity owing to its versatile features but still applications utilizing NSDFrFT are limited to optics and filtering. Working upon the same line of thought, the usefulness of NSDFrFT as a transform technique in the JPEG based image compression algorithm has to be explored. Thus, the main driving

force behind this research is the target to achieve visually superior, highly compressed reconstructed image.

## 2.5 OBJECTIVES OF THESIS

This research accomplishes following objectives:

- To achieve image compression using Non-Separable Discrete Fractional Fourier Transform (NSDFrFT).
- To compare NSDFrFT definition comprising of different interpolation techniques and suggest the better performer for compression purposes.
- To analyze the blocking in the reconstructed image for different fractional transform techniques.

# CHAPTER 3

# INTERPOLATION

## 3.1 INTRODUCTION

The lossless process of implementing digital samples to approximate the continuous function's values is known as interpolation. Interpolation performs the transformation of function from one resolution to another resolution without any degradation in the quality of transformed function. In the process of interpolation, actually the empty or unknown spaces are created in the source image and are filled up with the determined appropriate pixel values. An image interpolation can be done to achieve image enhancement, image rotation, image resizing, image zooming and many more [61].

## 3.2 NON-ADAPTIVE INTERPOLATION ALGORITHMS

The algorithms which are applied on all the pixels of the image irrespective of features and quality are known as non-adaptive interpolation algorithms. These algorithms can process all the pixels of the image without any constraint. The non-adaptive interpolation algorithms are the simplest interpolation techniques available.

### 3.2.1 NEAREST NEIGHBOR INTERPOLATION

The simplest interpolation method and requires minimum calculations to compute the point to be interpolated. The Nearest Neighbor Interpolation (NN Intr.) follows the point shift algorithm [62]. However, it can also be achieved by convolving the image with one-pixel rectangle in the spatial domain. The kernel for NN Intr. is given for both horizontal and vertical directions as

$$g(x) = \begin{cases} 0 & |x| > 0.5 \\ 1 & |x| < 0.5 \end{cases} \tag{3.1}$$

Mathematically, the interpolation point P based upon the kernel can be given as

$$P(x, y) = r(x + 0.5, y + 0.5) \tag{3.2}$$

where, $r_{11}(x_1, y_1)$, $r_{12}(x_1, y_2)$, $r_{21}(x_2, y_1)$ and $r_{22}(x_2, y_2)$ are pixels of an image.

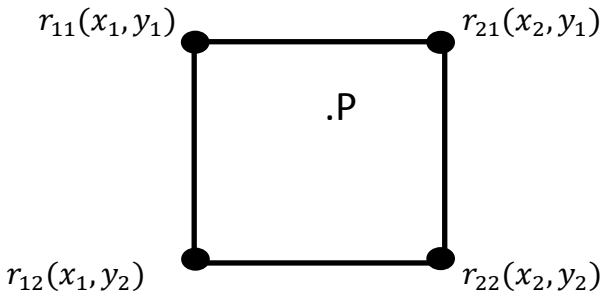



**Figure 3.1: Interpolation of a point P**

The effect of this process is that image pixel becomes larger with jagged heavy edges. To overcome the problem of jaggy edges, bilinear interpolation algorithm was used.

### 3.2.2 BILINEAR INTERPOLATION

Bil Intr. takes four immediate neighbor points to calculate the point to be interpolated. It is an extension of linear interpolation which is performed on a 2D grid. The main concept is to apply linear interpolation first in horizontal direction and then in vertical direction to achieve the Bil Intr. However, the directional sequence has no significance [63]. Suppose P$(x, y)$ is the point in the image to be interpolated and $f(r_{11})$ , $f(r_{12})$ , $f(r_{21})$ and $f(r_{22})$ are the pixel value at $r_{11}(x_1, y_1)$ , $r_{12}(x_1, y_2)$,$r_{21}(x_2, y_1)$ and $r_{22}(x_2, y_2)$ the immediate neighbor pixel of the point to be interpolated respectively as shown in Figure 3.1.

The mathematical representation of linear interpolation in the x-direction is given as

$$f(x, y_1) = \frac{x_2 - x}{x_2 - x_1} f(r_{11}) + \frac{x - x_1}{x_2 - x_1} f(r_{21}) \tag{3.3}$$

$$f(x, y_2) = \frac{x_2 - x}{x_2 - x_1} f(r_{12}) + \frac{x - x_1}{x_2 - x_1} f(r_{22}) \tag{3.4}$$

Similarly, the linear interpolation in the y-direction is obtained. Substituting the values of $f(x, y_1)$and $f(x, y_2)$ from eq. (3.3) and eq. (3.4) respectively, the Bil Intr. pixel can be given as

$$f(x,y) = \frac{f(r_{11})}{(x_2 - x_1)(y_2 - y_1)}(x_2 - x)(y_2 - y) + \frac{f(r_{12})}{(x_2 - x_1)(y_2 - y_1)}(x_2 - x)(y - y_1) + \frac{f(r_{21})}{(x_2 - x_1)(y_2 - y_1)}(x - x_1)(y_2 - y) + \frac{f(r_{22})}{(x_2 - x_1)(y_2 - y_1)}(x - x_1)(y - y_1) \quad (3.5)$$

The images interpolated by Bil Intr. have softer edges overcoming the shortcoming of NN Intr. with a better tradeoff between image visual quality and cost of computation. But the interpolated images are blur, reducing the visual quality of image. Thus to counter the problem of blurring, the interpolation algorithms using the concept of convolution were suggested, giving way to Bicubic Interpolation (Bic Intr.).

### 3.2.3 BICUBIC INTERPOLATION

Bic Intr. is an extension of cubic interpolation. Bic Intr. considers 16 points or coefficients to estimate the interpolation point of the image. The point to be interpolated by Bic Intr. is given as

$$f[x,y] = \sum_{l=-1}^{2} \sum_{m=-1}^{2} f(x_1 + l, y_1 + m) K(l - dx) K(dy - m) \quad (3.6)$$

where,

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

and

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

Bic Intr. results in smoother images and extend of interpolation artifacts in the interpolated images are also very less. However, speed and complexity of computation is where Bic Intr. lags [61].

## 3.3 APPLICATIONS

Interpolation being capable of predicting points at unknown locations finds numerous applications. In one such application of image zooming, while trying to excessively zoom an image, it becomes blur and blocky. So zooming an image only increases the size of a pixel, rather than imploring more pixels which effectively reduces the number of pixel contained in the same area. By putting the same number of pixels in the area after zooming as it was before zooming [63], a better resolved image without significantly affecting the size of image is attained as a result. This is termed as image zooming via interpolation. This process of image editing helps us focus at particular area in the image when the details in the image are more important than intricacy of detail. The process of image resizing helps us increase (or decrease) the size of image which are not perceived in their original size. This type of editing is necessary when total number of pixels in an image requires change [61].

Image rotation via interpolation is done when the viewing angle is not appropriately adjusted. In this method the pixels are simply copied and pasted on their new location [64], hence producing a rotated image. The new location of pixel can be calculated using rotation of axis about a fixed point.

**(a)** **(b)** **(c)**

**Figure 3.4: (a) Original image, (b) Image rotated at an angle of $15^0$, (c) Image rotated at an angle of $45^0$**

## 3.4 SUMMARY

The different definitions of non-adaptive interpolation techniques have been discussed in this chapter. The mathematical analysis of different non-adaptive interpolation techniques such as Nearest Neighbor Interpolation, Bilinear Interpolation and Bicubic Interpolation has been done along with their applications in the area of images. These different non-adaptive interpolation techniques have been used in the definition of NSDFrFT respectively, so that the effect of interpolation on the performance can be analyzed.

# CHAPTER 4

# IMAGE COMPRESSION USING VARIOUS TRANSFORM TECHNIQUES

Image compression deals with the scarcity of storage space and transmission bandwidth with less processing cost and computational time. Image compression is a procedure to reduce the number of bits required to represent an image without any significant loss in image visual quality. This chapter discusses and compares different transform techniques in image compression algorithm following JPEG technique.

## 4.1 INTRODUCTION

Digital imaging has gained importance, giving way to many applications [65]. These applications thrive for greater need of image compression to reduce the amount of digital data available globally. Redundancy and irrelevancy in image are the reasons which lead to the requirement of image compression. Redundancy is the amount of correlated pixels available in the image and irrelevancy of data is generally not noted by human eyes [1]. A digital image is obtained by sampling and quantizing a continuous image, but such images requires large amount of storage and samples to represent the energy. So, limitation of storage space will always be there and addition of extra storage devices does not solve the problem. Thus, to counter the storage problem, various image compression algorithms had been given [5]. In the past decades various image compression algorithms had been given which are JPEG2000 compression coding [65], using DFT, using DCT [66], using DFrFT [67], using DFrCT [67] and wavelet domain [68].

## 4.2 IMAGE COMPRESSION CHARACTERISTICS

Image compression can be characterized by three vital parameters: image compression percentage, compression speed and compressed image quality. Any image compression algorithm can undergo inspection considering these vital characteristics. These vital characteristics have been discussed in detail.

### 4.2.1 COMPRESSION PERCENTAGE

Compression percentage can be given as the ratio between the compressed image size and the original image size, multiplied with 100. Mathematically, compression percentage is given as

$$CP = \frac{no\ of\ original\ bits - no\ of\ compressed\ bits}{no\ of\ original\ bits} \times 100 \qquad (4.1)$$

The quality of image greatly depends upon the compression percentage. Higher the value of compression percentage, the visual quality of reconstructed image is inferior [10]. Highly image data dependent compression percentages are also considered by some image compression algorithms.

### 4.2.2 COMPRESSION SPEED

Compression speed is inversely proportional to the total computational time which includes both the compression time and the decompression time of an image [10]. Specifically, compression time is the time taken by the algorithm to compress an image and decompression time is the time taken to decompress an image. Thus, it can summarized that an image compression algorithm is considered to be best if it takes least compression and decompression time which enables the system to process quickly and depict the processed images in an efficient and interactive manner.

### 4.2.3 IMAGE QUALITY PARAMETERS

To analyze the quality of reconstructed image, various mathematical and statistical parameters are required. Among which subjective quality measurements includes visual scene perception influenced by spatial fidelity, viewing environment, observer's state of mind and extend of observer interacts with the visual scene [9], whereas, objective quality measurements includes algorithms which compares the processed images with original image. For example- Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Mean Structural Similarity Index Measure (MSSIM), Gradient Magnitude Similarity Deviation (GMSD) etc. The image quality metrics used for the analysis have been discussed in detail.

MSE is the average squared difference between a reference image and a distorted image. It is computed pixel by pixel, adding the squared differences of all pixels and dividing by the total count of the pixels [9]. Considering that $I(i,j)$ represents a pixel of reference image $I$ and $K(i,j)$ represents a pixel of distorted image $K$. Mathematically

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{4.2}$$

An important objective quality measure, PSNR is generally computed on the logarithmic scale and its values inversely depend upon MSE in the ratio with squared of maximum value of pixel, i.e. $(2^n - 1)^2$.The formula for PSNR is given as

$$PSNR = 10.log_{10}\left(\frac{MAX_I^2}{MSE}\right) \tag{4.3}$$

where, $MAX_I$ is the maximum pixel value [9]. In case of 8-bit image $MAX_I = 255$.

In 2004, Wang *et al.* [69] gave a Structural Similarity Index Measure (SSIM) with the idea that human eyes are adaptive to the structural information of the test object, image or scene. It compares the luminance firstly then contrast and at last the structure of the reference image and the distorted image. However, these components are independent of each other. The mathematical representation of SSIM index is as follows

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4.4}$$

where, $\mu_x$ and $\mu_y$ are the mean intensity and the contrast estimation is done by standard deviation $\sigma_x$ and $\sigma_y$. $C_1$ and $C_2$ are the constants which are included to avoid instability when $\mu_x^2 + \mu_y^2$ is near to zero [69]. The overall image quality is calculated using mean SSIM index given as

$$MSSIM(x,y) = \frac{1}{M}\sum_{j=1}^{M} SSIM(x_j, y_j) \tag{4.5}$$

In 2014, Xue *et al.* [70] gave an image quality assessment model known as GMSD. The image gradients are sensitive to distortions occurring at different degrees for various local structures. The GMSD algorithm used is highly accurate and faster in calculation. The standard deviation of the GMS map results into the final image quality score known as GMSD. The GMS map at location $i$ is given as

$$GMS(i) = \frac{2m_r(i)m_d(i) + c}{m_r^2(i) + m_d^2(i) + c} \tag{4.6}$$

whose average value GMSM is given as

$$GMSM = \frac{1}{N}\sum_{i=1}^{N} GMS(i) \tag{4.7}$$

Utilizing above mentioned $GMS(i)$ and $GMSM$, the GMSD is given as

$$GMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(GMS(i) - GMSM)^2} \tag{4.8}$$

The higher the GMSD score, the more is the distortion in the image. For the good quality of the reconstructed image among the image quality metrics used, MSE and GMSD must be minimum while the metrics PSNR and MSSIM should be maximum.

## 4.3 IMAGE COMPRESSION USING DISCRETE FRACTIONAL FOURIER TRANSFORM

To compress image using fractional transform, following methodology has been adopted.

1. The original image of $N \times N$ size is divided into sub-images known as blocks of $n \times n$ size. In the presented simulations $N$ =512 and $n$ =8 has been used.
2. Each block undergoes transformation independently. That means each sub-image is processed using 2D fractional transform technique.
3. The obtained image is in transform domain rather than in spatial domain. Obtained transform coefficients undergo quantization to remove the irrelevant information indulged with an image. Compression percentage plays an important role in deciding the limit for quantization of transform coefficients.

4. To achieve the image back into spatial domain, the complete procedure in inverse order is implemented. Implying that inverse fractional transform of the compressed image is done and then the blocks are combined and thus the decompressed image is obtained. The quality of obtained image is evaluated by means of different image quality parameters such as PSNR, MSE, MSSIM and GMSD. Figure 4.1 and Figure 4.2 shows the block diagram of compression and decompression process.

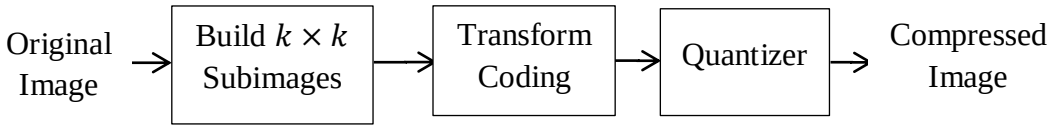



**Figure 4.1: Encoding process of block based image compression**

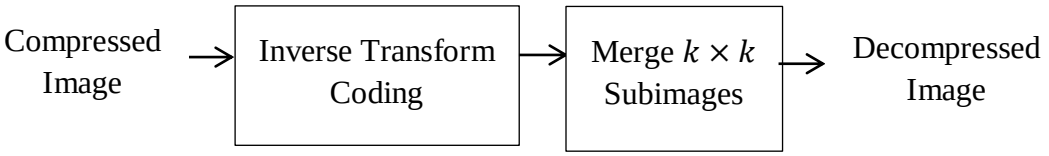
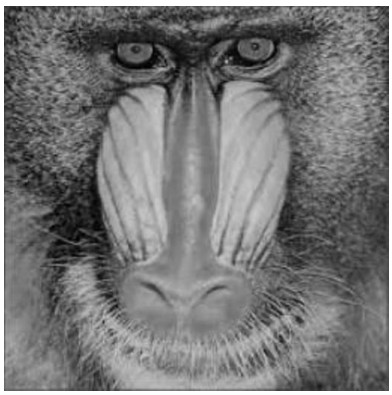



**Figure 4.2: Decoding process of block based image compression**

The algorithm of image compression has been applied upon the grey scale images of $512 \times 512$ shown in Figure 4.3 (Baboon, Barbara, Boat, Lighthouse, House, Lena, Pepper [71] and Grass [72]) for different compression percentages viz 10%-70%. The optimum value of $a_1$ and $a_2$ varies from 0 to 1. For the sake of simplicity $a_1$ and $a_2$ have been kept same, represented by $a_{op}$. The compressed image by DFrFT is shown in Figure 4.4 and Figure 4.5 with optimum values complied in Table 4.1.

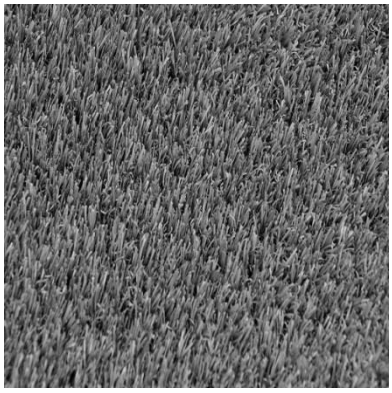


**(a) Baboon**

**(b) Barbara**

**(c) Boat**

**(d) Lighthouse**



**(e) Grass**

**(f) House**

**(g) Lena**

**(h) Pepper**

**Figure 4.3: Different images for the purpose of image compression [71], [72]**

**(a) Compressed Image at 10%**

**(b) Compressed Image at 20%**

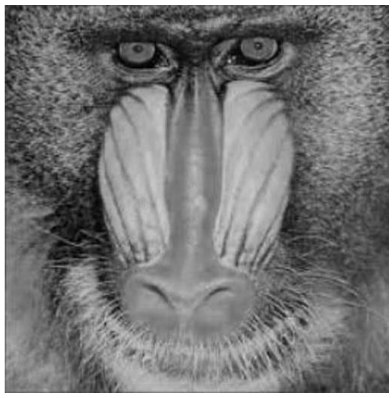
**(c) Compressed Image at 30%**

**(d) Compressed Image at 40%**

**(e) Compressed Image at 50%**

**(f) Compressed Image at 70%**

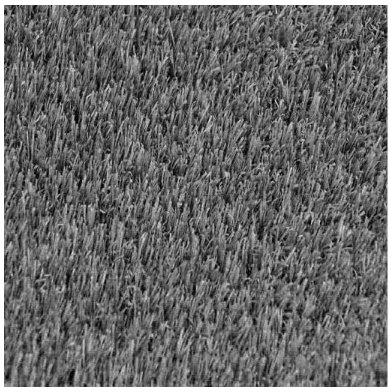
**Figure 4.4: Simulation results of image Lena at different compression percentages using DFrFT as transform technique**



**(a) Compressed Image of Baboon**

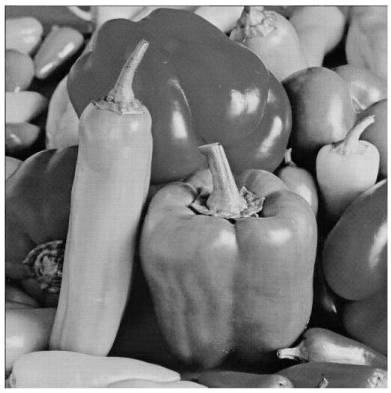
**(b) Compressed Image of Barbara**

**(c) Compressed Image of Boat**



**(d) Compressed Image of Grass**

**(e) Compressed Image of House**



**(f) Compressed Image of Pepper**

**Figure 4.5: Simulation results of different images at 70% compression using DFrFT as transform technique**

**Table 4.1: MSE, PSNR, MSSIM and GMSD at optimized values for different images using DFrFT**

| Compression Percentage | Baboon | | | | | Barbara | | | | | Boat | | | | | Pepper | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.99 | 0.246 | 62.64 | 1.0000 | 0.0016 | 0.93 | 0.206 | 54.64 | 1.0000 | 0.0096 | 0.91 | 0.496 | 57.71 | 0.9102 | 0.0715 | 0.93 | 2.006 | 52.64 | 0.9998 | 0.0129 |
| 20 | 0.98 | 0.516 | 51.00 | 0.9999 | 0.0077 | 0.94 | 0.792 | 49.13 | 0.9998 | 0.0125 | 0.93 | 1.470 | 51.39 | 0.9767 | 0.6733 | 0.91 | 6.753 | 39.83 | 0.9996 | 0.0149 |
| 30 | 0.95 | 0.572 | 50.55 | 0.9998 | 0.0105 | 0.95 | 3.046 | 43.29 | 0.9995 | 0.0127 | 0.95 | 3.860 | 46.45 | 0.9850 | 0.5980 | 0.95 | 2.346 | 44.42 | 0.9977 | 0.0357 |
| 40 | 0.96 | 2.942 | 43.70 | 0.9994 | 0.0195 | 0.97 | 5.874 | 40.44 | 0.9982 | 0.0348 | 0.98 | 7.582 | 43.28 | 0.9866 | 0.7656 | 0.96 | 10.122 | 38.07 | 0.9955 | 0.0798 |
| 50 | 0.98 | 5.088 | 41.06 | 0.9990 | 0.0419 | 0.98 | 14.46 | 36.52 | 0.9952 | 0.1626 | 0.92 | 9.801 | 41.25 | 0.9876 | 0.9954 | 0.98 | 14.462 | 36.52 | 0.9912 | 0.1673 |
| 60 | 0.97 | 5.807 | 40.49 | 0.9968 | 0.0572 | 0.98 | 20.51 | 35.00 | 0.9883 | 0.6308 | 0.91 | 13.01 | 38.28 | 0.9873 | 0.9697 | 0.97 | 20.136 | 35.09 | 0.9852 | 0.3745 |
| 70 | 0.99 | 9.780 | 38.22 | 0.9915 | 0.0809 | 0.99 | 26.13 | 33.09 | 0.9792 | 0.9384 | 0.93 | 19.01 | 34.82 | 0.9206 | 0.3665 | 0.99 | 53.579 | 30.84 | 0.9757 | 0.2901 |

| Compression Percentage | Grass | | | | | Lena | | | | | House | | | | | Lighthouse | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.91 | 0.306 | 53.26 | 0.9999 | 0.0006 | 0.98 | 0.030 | 63.28 | 1.0000 | 0.0094 | 0.99 | 0.948 | 48.35 | 1.0000 | 0.0023 | 0.92 | 0.631 | 50.12 | 0.9999 | 0.0021 |
| 20 | 0.93 | 3.205 | 47.32 | 0.9997 | 0.0032 | 0.99 | 0.083 | 58.92 | 0.9988 | 0.0532 | 0.99 | 0.967 | 48.27 | 0.9998 | 0.0047 | 0.93 | 1.431 | 46.57 | 0.9998 | 0.0058 |
| 30 | 0.95 | 7.631 | 39.30 | 0.9992 | 0.0045 | 0.98 | 0.265 | 53.89 | 0.9978 | 0.0568 | 0.92 | 2.445 | 44.24 | 0.9995 | 0.0218 | 0.99 | 4.971 | 41.16 | 0.9993 | 0.0251 |
| 40 | 0.97 | 16.554 | 35.94 | 0.9979 | 0.0122 | 0.97 | 0.860 | 48.78 | 0.9975 | 0.0620 | 0.97 | 3.767 | 42.37 | 0.9989 | 0.0272 | 0.95 | 6.083 | 40.28 | 0.9990 | 0.0278 |
| 50 | 0.98 | 33.65 | 32.86 | 0.9949 | 0.0276 | 0.97 | 1.960 | 45.20 | 0.9964 | 0.0646 | 0.99 | 13.875 | 36.70 | 0.9983 | 0.1232 | 0.96 | 10.092 | 38.09 | 0.9986 | 0.0550 |
| 60 | 0.98 | 51.22 | 31.03 | 0.9886 | 0.0960 | 0.98 | 3.822 | 42.30 | 0.9894 | 0.0658 | 0.98 | 22.010 | 34.70 | 0.9963 | 0.2363 | 0.97 | 16.192 | 36.03 | 0.9971 | 0.1046 |
| 70 | 0.99 | 190.62 | 26.30 | 0.9766 | 0.3202 | 0.99 | 9.233 | 38.47 | 0.9824 | 0.0827 | 0.99 | 22.245 | 34.65 | 0.9930 | 0.2506 | 0.99 | 23.253 | 34.46 | 0.9945 | 0.2447 |

## 4.4 IMAGE COMPRESSION USING NON-SEPARABLE DISCRETE FRACTIONAL FOURIER TRANSFORM

To compress an image using 2D NSDFrFT, following methodology has been adopted.

1. 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 Bil Intr. is given as

$$f\left[\frac{(cos\emptyset_1 x + sin\emptyset_1 y)}{\cos(\emptyset_1 - \emptyset_2)}, \frac{(-sin\emptyset_2 x + cos\emptyset_2 y)}{\cos(\emptyset_1 - \emptyset_2)}\right] = \frac{1}{(x_2 - x_1)(y_2 - y_1)} \times$$

$$\left[f(r_{11})\left(x_2 - \frac{(cos\emptyset_1 x + sin\emptyset_1 y)}{\cos(\emptyset_1 - \emptyset_2)}\right)\left(y_2 - \frac{(-sin\emptyset_2 x + cos\emptyset_2 y)}{\cos(\emptyset_1 - \emptyset_2)}\right)\right.$$
$$+ f(r_{12})\left(x_2 - \frac{(cos\emptyset_1 x + sin\emptyset_1 y)}{\cos(\emptyset_1 - \emptyset_2)}\right)\left(\frac{(-sin\emptyset_2 x + cos\emptyset_2 y)}{\cos(\emptyset_1 - \emptyset_2)} - y_1\right)$$
$$+ f(r_{21})\left(\frac{(cos\emptyset_1 x + sin\emptyset_1 y)}{\cos(\emptyset_1 - \emptyset_2)} - x_1\right)\left(y_2 - \frac{(-sin\emptyset_2 x + cos\emptyset_2 y)}{\cos(\emptyset_1 - \emptyset_2)}\right)$$
$$\left. + f(r_{22})\left(\frac{(cos\emptyset_1 x + sin\emptyset_1 y)}{\cos(\emptyset_1 - \emptyset_2)} - x_1\right)\left(\frac{(-sin\emptyset_2 x + cos\emptyset_2 y)}{\cos(\emptyset_1 - \emptyset_2)} - y_1\right)\right] \quad (4.9)$$

Similarly, mathematical representation for mapping via Bic Intr. 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 eq. (3.6) we get Bic Intr. image. While for NN Intr.

$$f(x_r) = \frac{x_{r-1} + x_r}{2} < x \leq \frac{x_r + x_{r+1}}{2} \quad (4.10)$$

gives the NN Intr. image.

2. The interpolated image of $N \times N$ size is divided into sub-images known as blocks of $n \times n$ size. In our simulations we have used $N$=512 and $n$=8. Each block undergoes transformation independently. Mathematically,

$$F_{\emptyset_1,\emptyset_2}^{a_1,a_2} = F^{a_1,a_2}(f[q, r]) \quad (4.11)$$

where,

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

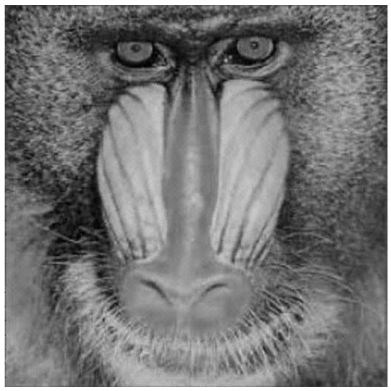
and

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

3. 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.
4. 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. The quality of obtained image is evaluated by means of different IQMs such as MSE, PSNR, MSSIM and GMSD.

The compressed images using NSDFrFT-NN Intr., for different compression percentages of 10%, 20%, 30%, 40%, 50%, 60% and 70% for different images have been shown in Figure 4.6 and for NSDFrFT-Bil Intr. in Figure 4.7 and finally for NSDFrFT-Bic Intr. in Figure 4.8 and Figure 4.9 respectively. The optimum fractional order and the rotation angle for different compression percentages along with image quality parameters have been compiled in Table 4.2 for NSDFrFT-NN Intr., in Table 4.3 for NSDFrFT-Bil Intr. and in Table 4.4 for NSDFrFT-Bic Intr. respectively.

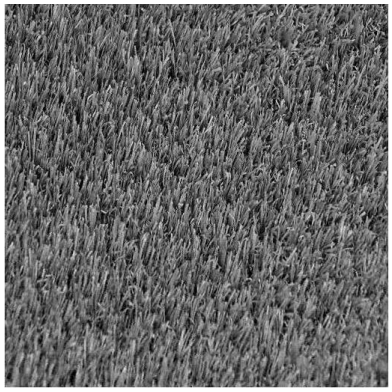
For sake of simplicity $\emptyset_1$ and $\emptyset_2$ have been kept same and represented by $\emptyset_{op}$. The definition of NSDFrFT converges to DFrFT by substituting 0 to both rotational angles $\emptyset_1$ and $\emptyset_2$.



**(a) Compressed Image of Baboon**

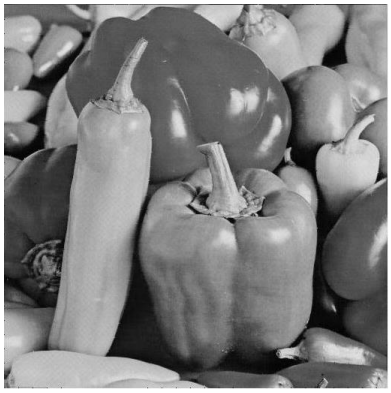
**(b) Compressed Image of Barbara**

**(c) Compressed Image of Boat**



**(d) Compressed Image of Grass**

**(e) Compressed Image of House**



**(f) Compressed Image of Pepper**

**Figure 4.6: Simulation results of different images at 70% compression using NSDFrFT-NN Intr. as transform technique**

**Table 4.2: MSE, PSNR, MSSIM and GMSD at optimized values for different images using NSDFrFT-NN Intr.**

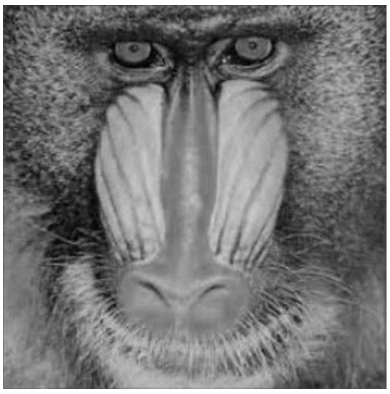
| Compression Percentage | Baboon | | | | | | | Barbara | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.91 | -0.97 | 0.0175 | 1.9731 | 46.1110 | 0.9974 | 0.0364 | 0.99 | -0.97 | 0.0175 | 2.2622 | 40.7890 | 0.9919 | 0.0822 |
| 20 | 0.93 | -0.95 | 0.0175 | 2.0017 | 45.1793 | 0.9973 | 0.0423 | 0.94 | -0.97 | 0.0175 | 3.3592 | 38.7829 | 0.9917 | 0.0828 |
| 30 | 0.94 | -0.99 | 0.0175 | 2.1297 | 45.1167 | 0.9971 | 0.0435 | 0.93 | -0.99 | 0.0175 | 7.4977 | 37.7740 | 0.9911 | 0.0838 |
| 40 | 0.97 | -0.98 | 0.0175 | 3.4610 | 44.7618 | 0.9966 | 0.0449 | 0.98 | -0.98 | 0.0175 | 11.9402 | 35.7461 | 0.9902 | 0.0853 |
| 50 | 0.92 | -0.97 | 0.0524 | 3.6882 | 43.4437 | 0.9947 | 0.0460 | 0.98 | -0.97 | 0.0524 | 20.3531 | 33.6580 | 0.9864 | 0.0912 |
| 60 | 0.95 | -0.99 | 0.0524 | 4.6119 | 42.8711 | 0.9904 | 0.0558 | 0.95 | -0.99 | 0.0524 | 23.3033 | 30.5392 | 0.9829 | 0.1000 |
| 70 | 0.93 | -0.99 | 0.0524 | 7.6250 | 39.5534 | 0.9863 | 0.0644 | 0.93 | -0.99 | 0.0524 | 27.6630 | 29.3998 | 0.9709 | 0.1413 |

| Compression Percentage | Boat | | | | | | | Grass | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.92 | -0.99 | 0.0175 | 5.508 | 34.237 | 0.9951 | 0.0715 | 0.95 | -0.98 | 0.0175 | 11.8101 | 39.708 | 0.9919 | 0.0288 |
| 20 | 0.95 | -0.97 | 0.0175 | 5.689 | 34.205 | 0.9947 | 0.0721 | 0.94 | -0.97 | 0.0175 | 15.0469 | 34.675 | 0.9917 | 0.0293 |
| 30 | 0.93 | -0.99 | 0.0175 | 6.825 | 34.181 | 0.9938 | 0.0732 | 0.93 | -0.95 | 0.0175 | 20.980 | 32.602 | 0.9912 | 0.0303 |
| 40 | 0.93 | -0.98 | 0.0175 | 7.519 | 34.062 | 0.9917 | 0.0762 | 0.97 | -0.98 | 0.0175 | 42.361 | 30.048 | 0.9900 | 0.0320 |
| 50 | 0.98 | -0.97 | 0.0524 | 11.641 | 33.715 | 0.9876 | 0.0834 | 0.92 | -0.97 | 0.0524 | 65.445 | 28.717 | 0.9872 | 0.0367 |
| 60 | 0.95 | -0.99 | 0.0524 | 15.030 | 33.355 | 0.9797 | 0.1016 | 0.94 | -0.99 | 0.0524 | 186.819 | 25.416 | 0.9819 | 0.0456 |
| 70 | 0.93 | -0.99 | 0.0524 | 23.435 | 33.020 | 0.9667 | 0.1404 | 0.95 | -0.99 | 0.0524 | 237.054 | 24.382 | 0.9761 | 0.0577 |

**Table 4.2: MSE, PSNR, MSSIM and GMSD at optimized values for different images using NSDFrFT-NN Intr. (contd.)**

| Compression Percentage | Lena | | | | | | | House | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.92 | -0.97 | 0.0175 | 4.2158 | 41.8820 | 0.9867 | 0.0647 | 0.96 | -0.99 | 0.0175 | 3.8820 | 42.2402 | 0.9967 | 0.1238 |
| 20 | 0.94 | -0.97 | 0.0175 | 11.1559 | 37.6558 | 0.9866 | 0.0666 | 0.99 | -0.97 | 0.0175 | 3.9558 | 42.1584 | 0.9966 | 0.1239 |
| 30 | 0.93 | -0.99 | 0.0175 | 11.3014 | 37.5995 | 0.9863 | 0.0710 | 0.93 | -0.98 | 0.0175 | 4.1586 | 41. 9413 | 0.9965 | 0.1242 |
| 40 | 0.97 | -0.98 | 0.0175 | 11.8671 | 37.3874 | 0.9866 | 0.0759 | 0.98 | -0.95 | 0.0175 | 4.3286 | 41. 7674 | 0.9959 | 0.1272 |
| 50 | 0.98 | -0.97 | 0.0524 | 12.5943 | 37.1291 | 0.9839 | 0.0767 | 0.94 | -0.96 | 0.0524 | 4.3436 | 41.7523 | 0.9940 | 0.1305 |
| 60 | 0.95 | -0.99 | 0.0524 | 19.3568 | 35.2625 | 0.9806 | 0.0859 | 0.99 | -0.99 | 0.0524 | 4.3568 | 41.7391 | 0.9485 | 0.3064 |
| 70 | 0.93 | -0.99 | 0.0524 | 21.4090 | 34.8248 | 0.9740 | 0.0952 | 0.95 | -0.99 | 0.0524 | 17.8974 | 35.6029 | 0.9355 | 0.4561 |

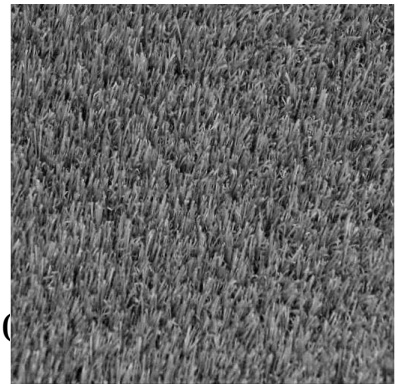
| Compression Percentage | Pepper | | | | | | | Lighthouse | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.94 | -0.97 | 0.0175 | 16.6171 | 35.9252 | 0.9958 | 0.0813 | 0.93 | -0.99 | 0.0175 | 7.8218 | 39.1978 | 0.9911 | 0.1307 |
| 20 | 0.90 | -0.98 | 0.0175 | 17.0137 | 35.8228 | 0.9952 | 0.0824 | 0.94 | -0.99 | 0.0175 | 8.6211 | 38.7752 | 0.9910 | 0.0534 |
| 30 | 0.93 | -0.99 | 0.0175 | 17.7938 | 35.6281 | 0.9939 | 0.0841 | 0.93 | -0.99 | 0.0175 | 8.6464 | 38.7624 | 0.9909 | 0.0535 |
| 40 | 0.97 | -0.98 | 0.0175 | 18.4953 | 35.4602 | 0.9920 | 0.0860 | 0.95 | -0.99 | 0.0175 | 8.6716 | 38.7498 | 0.9900 | 0.0535 |
| 50 | 0.95 | -0.97 | 0.0524 | 20.4170 | 35.0309 | 0.9896 | 0.0884 | 0.95 | -0.99 | 0.0524 | 8.9101 | 38.6320 | 0.9886 | 0.0542 |
| 60 | 0.95 | -0.99 | 0.0524 | 22.8815 | 34.5360 | 0.9840 | 0.0962 | 0.95 | -0.99 | 0.0524 | 9.3188 | 38.4372 | 0.9838 | 0.0551 |
| 70 | 0.93 | -0.99 | 0.0524 | 24.9232 | 34.1648 | 0.9772 | 0.1107 | 0.93 | -0.99 | 0.0524 | 10.528 | 37.9071 | 0.9789 | 0.0710 |



**(a) Compressed Image of Baboon**

**(b) Compressed Image of Barbara**

**(c) Compressed Image of Boat**



**(d) Compressed Image of Grass**

**(e) Compressed Image of House**

**(f) Compressed Image of Pepper**

**Figure 4.7: Simulation results of different image at 70% compression using NSDFrFT-Bil Intr. as transform technique**

**Table 4.3: MSE, PSNR, MSSIM and GMSD at optimized values for different images using NSDFrFT-Bil Intr.**

| Compression Percentage | Baboon | | | | | | | Barbara | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.95 | -0.97 | 0.0175 | 1.2459 | 47.1759 | 0.9953 | 0.0223 | 0.93 | -0.95 | 0.0175 | 1.1509 | 46.7879 | 0.9703 | 0.0854 |
| 20 | 0.97 | -0.99 | 0.0175 | 1.6502 | 46.6815 | 0.9952 | 0.0424 | 0.97 | -0.98 | 0.0175 | 2.2366 | 42.7663 | 0.9701 | 0.0857 |
| 30 | 0.93 | -0.98 | 0.0175 | 1.7538 | 45.6890 | 0.9949 | 0.0432 | 0.93 | -0.99 | 0.0175 | 3.4635 | 40.7095 | 0.9699 | 0.0861 |
| 40 | 0.97 | -0.99 | 0.0524 | 2.9421 | 45.1110 | 0.9942 | 0.0443 | 0.91 | -0.99 | 0.0524 | 9.9163 | 38.5983 | 0.9691 | 0.0871 |
| 50 | 0.95 | -0.99 | 0.0524 | 3.3571 | 44.8197 | 0.9921 | 0.0473 | 0.95 | -0.99 | 0.0524 | 17.2622 | 35.8515 | 0.9670 | 0.0894 |
| 60 | 0.93 | -0.96 | 0.0524 | 3.6882 | 42.2627 | 0.9893 | 0.0513 | 0.93 | -0.99 | 0.0524 | 21.2782 | 35.1755 | 0.9643 | 0.0924 |
| 70 | 0.95 | -0.98 | 0.0524 | 5.1927 | 40.9769 | 0.9855 | 0.0574 | 0.95 | -0.98 | 0.0524 | 24.8464 | 34.2575 | 0.9607 | 0.0986 |

| Compression Percentage | Boat | | | | | | | Grass | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.92 | -0.96 | 0.0175 | 4.9647 | 41.1719 | 0.9890 | 0.0748 | 0.93 | -0.96 | 0.0175 | 9.669 | 40.8245 | 0.9729 | 0.0373 |
| 20 | 0.97 | -0.99 | 0.0175 | 5.1489 | 41.0136 | 0.9887 | 0.0750 | 0.97 | -0.99 | 0.0175 | 13.000 | 36.9108 | 0.9727 | 0.0375 |
| 30 | 0.96 | -0.99 | 0.0175 | 5.5990 | 40.6497 | 0.9885 | 0.0752 | 0.93 | -0.99 | 0.0175 | 17.980 | 33.4188 | 0.9722 | 0.0380 |
| 40 | 0.99 | -0.99 | 0.0524 | 6.8242 | 39.7903 | 0.9876 | 0.0762 | 0.91 | -0.99 | 0.0524 | 29.819 | 30.2434 | 0.9706 | 0.0395 |
| 50 | 0.95 | -0.99 | 0.0524 | 9.3298 | 38.4321 | 0.9846 | 0.0790 | 0.95 | -0.99 | 0.0524 | 37.361 | 29.0354 | 0.9668 | 0.0433 |
| 60 | 0.95 | -0.97 | 0.0524 | 13.568 | 36.8056 | 0.9809 | 0.0841 | 0.93 | -0.99 | 0.0524 | 81.445 | 27.9018 | 0.9627 | 0.0470 |
| 70 | 0.95 | -0.97 | 0.0524 | 20.169 | 35.0838 | 0.9760 | 0.0933 | 0.95 | -0.98 | 0.0524 | 130.054 | 26.9837 | 0.9579 | 0.0523 |

**Table 4.3: MSE, PSNR, MSSIM and GMSD at optimized values for different images using NSDFrFT-Bil Intr. (contd.)**

| Compression Percentage | Lena | | | | | | | House | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.93 | -0.96 | 0.0175 | 2.5424 | 44.0784 | 0.9960 | 0.0545 | 0.97 | -0.99 | 0.0175 | 3.7199 | 42.4255 | 0.9953 | 0.1228 |
| 20 | 0.97 | -0.99 | 0.0175 | 3.5342 | 42.6479 | 0.9957 | 0.0567 | 0.97 | -0.99 | 0.0175 | 3.9454 | 42.1699 | 0.9952 | 0.1230 |
| 30 | 0.93 | -0.99 | 0.0175 | 4.5991 | 41.5041 | 0.9955 | 0.0650 | 0.95 | -0.99 | 0.0175 | 4.4211 | 41.6755 | 0.9951 | 0.1232 |
| 40 | 0.91 | -0.99 | 0.0524 | 4.7800 | 41.3365 | 0.9947 | 0.0652 | 0.93 | -0.99 | 0.0524 | 4.4781 | 41.6199 | 0.9943 | 0.1257 |
| 50 | 0.95 | -0.99 | 0.0524 | 5.0572 | 41.0917 | 0.9907 | 0.0658 | 0.95 | -0.99 | 0.0524 | 4.5096 | 41.5895 | 0.9894 | 0.1306 |
| 60 | 0.93 | -0.99 | 0.0524 | 5.8370 | 40.4689 | 0.9870 | 0.0786 | 0.93 | -0.99 | 0.0524 | 4.5200 | 41.5794 | 0.9862 | 0.1338 |
| 70 | 0.95 | -0.98 | 0.0524 | 6.4165 | 40.0578 | 0.9815 | 0.0937 | 0.95 | -0.99 | 0.0524 | 10.134 | 38.0727 | 0.9726 | 0.1661 |

| Compression Percentage | Pepper | | | | | | | Lighthouse | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.93 | -0.97 | 0.0175 | 4.9233 | 41.2082 | 0.9928 | 0.0772 | 0.95 | -0.97 | 0.0175 | 2.0831 | 44.9436 | 0.9964 | 0.0608 |
| 20 | 0.93 | -0.97 | 0.0175 | 5.3663 | 40.8340 | 0.9924 | 0.0780 | 0.95 | -0.97 | 0.0175 | 2.0929 | 44.9234 | 0.9964 | 0.0609 |
| 30 | 0.90 | -0.95 | 0.0175 | 5.7341 | 40.5462 | 0.9913 | 0.0792 | 0.93 | -0.97 | 0.0175 | 3.0309 | 43.3151 | 0.9959 | 0.0486 |
| 40 | 0.93 | -0.99 | 0.0524 | 6.3004 | 40.1371 | 0.9902 | 0.0801 | 0.95 | -0.99 | 0.0524 | 3.1342 | 43.1695 | 0.9954 | 0.0491 |
| 50 | 0.95 | -0.98 | 0.0524 | 7.7921 | 39.2142 | 0.9871 | 0.0824 | 0.95 | -0.99 | 0.0524 | 3.4266 | 42.7822 | 0.9941 | 0.0502 |
| 60 | 0.93 | -0.99 | 0.0524 | 8.6539 | 38.7587 | 0.9827 | 0.0867 | 0.93 | -0.95 | 0.0524 | 3.9531 | 42.1614 | 0.9890 | 0.1309 |
| 70 | 0.93 | -0.99 | 0.0524 | 13.9341 | 36.6900 | 0.9776 | 0.0956 | 0.95 | -0.99 | 0.0524 | 3.6580 | 42.4984 | 0.9845 | 0.1392 |

**(a) Compressed Image at 10%**

**(b) Compressed Image at 20%**

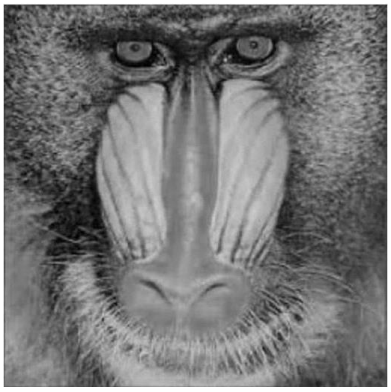
**(c) Compressed Image at 30%**

**(d) Compressed Image at 40%**

**(e) Compressed Image at 50%**

**(f) Compressed Image at 70%**

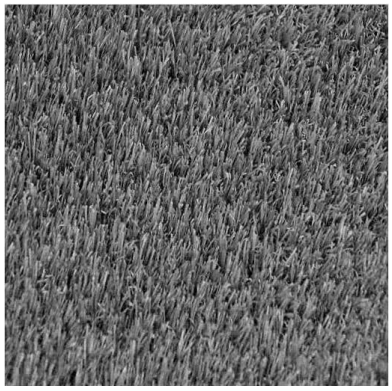
**Figure 4.8 : Simulation results of image Lena at different compression percentages using NSDFrFT-Bic Intr. as transform technique**



**(a) Compressed Image of Baboon**

**(b) Compressed Image of Barbara**

**(c) Compressed Image of Boat**



**(d) Compressed Image of Grass**

**(e) Compressed Image of House**

**(f) Compressed Image of Pepper**

**Figure 4.9: Simulation results of different images at 70% compression using NSDFrFT-Bic Intr. as transform technique**

**Table 4.4 : MSE, PSNR, MSSIM and GMSD at optimized values for different images using NSDFrFT-Bic Intr.**

| Compression Percentage | Baboon | | | | | | | Barbara | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.95 | -0.97 | 0.0175 | 1.1266 | 47.6130 | 0.9990 | 0.0124 | 0.95 | -0.97 | 0.0175 | 0.3105 | 48.8340 | 0.9928 | 0.0168 |
| 20 | 0.95 | -0.97 | 0.0175 | 1.2502 | 47.1609 | 0.9989 | 0.0254 | 0.95 | -0.97 | 0.0175 | 1.2630 | 46.3971 | 0.9923 | 0.0290 |
| 30 | 0.95 | -0.99 | 0.0175 | 1.3538 | 46.8154 | 0.9988 | 0.0361 | 0.95 | -0.99 | 0.0175 | 3.0040 | 42.5115 | 0.9913 | 0.0322 |
| 40 | 0.98 | -0.97 | 0.0524 | 1.5921 | 46.5377 | 0.9984 | 0.0406 | 0.98 | -0.97 | 0.0524 | 5.8749 | 40.4408 | 0.9916 | 0.0601 |
| 50 | 0.93 | -0.99 | 0.0524 | 2.1391 | 45.9354 | 0.9968 | 0.0437 | 0.93 | -0.99 | 0.0524 | 9.5521 | 37.4575 | 0.9893 | 0.0706 |
| 60 | 0.95 | -0.97 | 0.0524 | 2.1722 | 44.8285 | 0.9944 | 0.0441 | 0.95 | -0.97 | 0.0524 | 20.182 | 35.7663 | 0.9864 | 0.0822 |
| 70 | 0.99 | -0.99 | 0.0524 | 3.4138 | 42.7984 | 0.9871 | 0.0488 | 0.99 | -0.99 | 0.0524 | 23.010 | 35.4187 | 0.9800 | 0.0824 |

| Compression Percentage | Boat | | | | | | | Grass | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.95 | -0.99 | 0.0175 | 1.4633 | 52.7562 | 0.9968 | 0.0715 | 0.95 | -0.97 | 0.0175 | 1.2698 | 46.2479 | 0.9955 | 0.0213 |
| 20 | 0.95 | -0.99 | 0.0175 | 1.6733 | 50.5925 | 0.9966 | 0.0718 | 0.95 | -0.97 | 0.0175 | 6.2698 | 41.9736 | 0.9953 | 0.0218 |
| 30 | 0.95 | -0.99 | 0.0175 | 2.0525 | 48.3115 | 0.9961 | 0.0725 | 0.95 | -0.97 | 0.0175 | 11.3192 | 39.0760 | 0.9948 | 0.0227 |
| 40 | 0.95 | -0.99 | 0.0524 | 4.5175 | 45.3701 | 0.9942 | 0.0745 | 0.95 | -0.97 | 0.0524 | 14.4212 | 38.3406 | 0.9929 | 0.0259 |
| 50 | 0.93 | -0.99 | 0.0524 | 5.8029 | 43.6845 | 0.9924 | 0.0768 | 0.93 | -0.97 | 0.0524 | 18.2413 | 35.5202 | 0.9885 | 0.0329 |
| 60 | 0.95 | -0.99 | 0.0524 | 11.6056 | 39.4841 | 0.9882 | 0.0838 | 0.95 | -0.97 | 0.0524 | 62.0318 | 30.2047 | 0.9808 | 0.0418 |
| 70 | 0.93 | -0.99 | 0.0524 | 14.6000 | 36.3892 | 0.9804 | 0.1030 | 0.95 | -0.97 | 0.0524 | 77.6940 | 29.2280 | 0.9769 | 0.0503 |

**Table 4.4 : MSE, PSNR, MSSIM and GMSD at optimized values for different images using NSDFrFT-Bic Intr. (contd.)**

| Compression Percentage | Lena | | | | | | | House | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.95 | -0.97 | 0.0175 | 0.2632 | 53.9284 | 0.9999 | 0.0023 | 0.95 | -0.97 | 0.0175 | 2.0610 | 44.9900 | 0.9976 | 0.1251 |
| 20 | 0.95 | -0.97 | 0.0175 | 0.6798 | 49.5615 | 0.9997 | 0.0106 | 0.97 | -0.98 | 0.0175 | 2.0721 | 44.9667 | 0.9976 | 0.1252 |
| 30 | 0.95 | -0.99 | 0.0175 | 0.9018 | 48.5798 | 0.9995 | 0.0109 | 0.98 | -0.99 | 0.0175 | 2.0823 | 44.9453 | 0.9974 | 0.1261 |
| 40 | 0.98 | -0.97 | 0.0524 | 1.0734 | 47.8230 | 0.9990 | 0.0117 | 0.94 | -0.95 | 0.0524 | 2.1887 | 44.7290 | 0.9971 | 0.1276 |
| 50 | 0.93 | -0.99 | 0.0524 | 1.7945 | 45.5913 | 0.9970 | 0.0168 | 0.93 | -0.99 | 0.0524 | 2.7976 | 43.6629 | 0.9967 | 0.1033 |
| 60 | 0.95 | -0.97 | 0.0524 | 3.2012 | 43.0777 | 0.9921 | 0.0292 | 0.92 | -0.97 | 0.0524 | 3.3081 | 42.9350 | 0.9952 | 0.1047 |
| 70 | 0.99 | -0.99 | 0.0524 | 4.7413 | 42.7402 | 0.9863 | 0.0306 | 0.99 | -0.97 | 0.0524 | 4.2405 | 41.8566 | 0.9851 | 0.2871 |

| Compression Percentage | Pepper | | | | | | | Lighthouse | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD | $a_1$ | $a_2$ | $\emptyset_{op}$ | MSE | PSNR | MSSIM | GMSD |
| 10 | 0.95 | -0.97 | 0.0175 | 2.5383 | 44.0853 | 0.9973 | 0.0544 | 0.95 | -0.97 | 0.0175 | 1.4612 | 46.4838 | 0.9981 | 0.0547 |
| 20 | 0.95 | -0.97 | 0.0175 | 2.7181 | 43.7882 | 0.9972 | 0.0661 | 0.95 | -0.97 | 0.0175 | 1.7906 | 45.6008 | 0.9980 | 0.0513 |
| 30 | 0.95 | -0.99 | 0.0175 | 3.1950 | 43.0862 | 0.9966 | 0.0672 | 0.95 | -0.99 | 0.0175 | 2.1364 | 44.8339 | 0.9977 | 0.0589 |
| 40 | 0.98 | -0.97 | 0.0524 | 4.0851 | 42.0188 | 0.9958 | 0.0698 | 0.98 | -0.97 | 0.0524 | 2.1835 | 44.7392 | 0.9973 | 0.0496 |
| 50 | 0.93 | -0.99 | 0.0524 | 5.3871 | 40.8172 | 0.9925 | 0.0764 | 0.93 | -0.99 | 0.0524 | 2.2763 | 44.5584 | 0.9954 | 0.0493 |
| 60 | 0.95 | -0.97 | 0.0524 | 8.7148 | 38.7282 | 0.9911 | 0.0787 | 0.95 | -0.97 | 0.0524 | 2.3403 | 44.4380 | 0.9917 | 0.0490 |
| 70 | 0.99 | -0.99 | 0.0524 | 12.1635 | 37.2802 | 0.9827 | 0.0797 | 0.99 | -0.99 | 0.0524 | 2.3687 | 44.3858 | 0.9871 | 0.0488 |

## 4.5 COMPARISON OF COMPRESSION ALGORITHMS USING IMAGE QUALITY PARAMETERS

The Non-Separable Discrete Fractional Transform has been compared with Discrete Fractional Transform and JPEG for Lena and Pepper of size $(256 \times 256)$ at 50% compression in Table 4.5. 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 4.5 : 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 [67] | Lena | 44.55 | 5.33 | 5.33 |
| | Pepper | 42.28 | 3.48 | 3.48 |
| JPEG [74] | Lena | 34.66 | 0.12 | 0.12 |
| | Pepper | 34.27 | 0.20 | 0.20 |

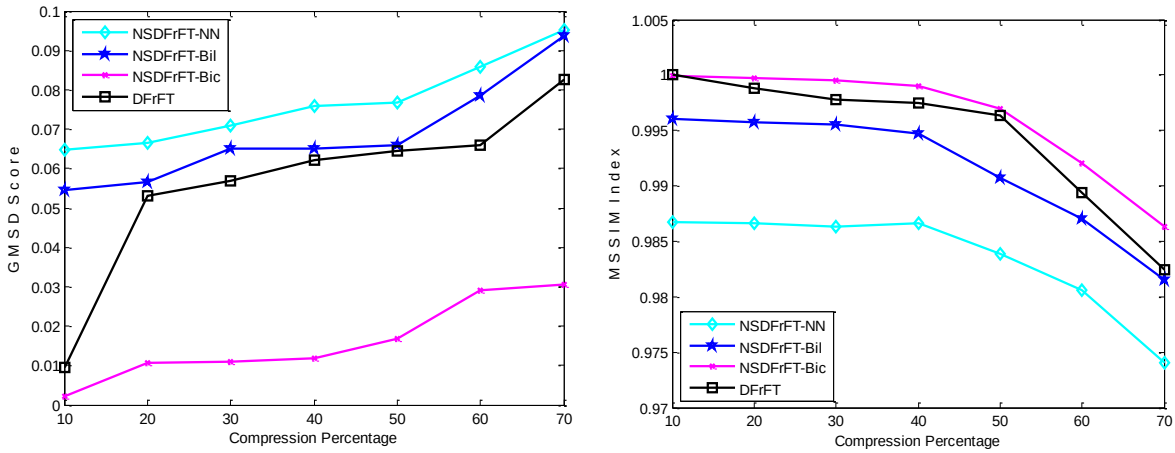
**Table 4.6 : Comparison of Non-Separable Discrete Fractional Transform, Discrete Fractional Transform at 50% compression**

| Algorithm | Image $(512 \times 512)$ | PSNR in dB | CPU Time | |
|---|---|---|---|---|
| | | | Encoding Time (sec) | Decoding Time (sec) |
| Non-Separable Discrete Fractional Transform | Lena | 45.59 | 10.03 | 10.03 |
| | Pepper | 40.81 | 8.56 | 8.56 |
| Discrete Fractional Transform | Lena | 45.20 | 7.22 | 7.22 |
| | Pepper | 36.52 | 5.86 | 5.86 |

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 4.6. 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.

The different variations of Non-Separable Discrete Fractional Transform, when compared suggest that NSDFrFT-Bic Intr. perform better in comparison to NSDFrFT-Bil Intr. and NSDFrFT-NN Intr. In terms of image quality parameters, PSNR suggest that NSDFrFT-Bic Intr. performs better in comparison to DFrFT at higher compression percentages. In accordance to GMSD score and MSSIM index, NSDFrFT gives structurally and predictively similar reconstructed image as that of original image. Figure 4.10 shows the comparison of different transform techniques for the image Lena at different compression percentages.




**(a) Compression vs GMSD Score for Lena**

**(b) Compression vs MSSIM Index for Lena**

**Figure 4.10: Comparison of different fractional transforms for image Lena at different compression percentages**

The plot of GMSD Score vs compression percentage suggested that subjective image quality is high for NSDFrFT-Bic Intr. which considers the local image distortions caused by local structure diversity. The plot of MSSIM index vs compression percentage concludes that the reconstructed image from NSDFrFT-Bic Intr. have high structural similarity to the original image in comparison to compared fractional transform techniques. In order to analyze the techniques properly images have been categorized into three classes [72]:

1. High Frequency Images- Baboon, Grass
2. Medium Frequency Images- Barbara, House, Lighthouse
3. Low Frequency Images- Boat, Lena, Pepper

PSNR and MSE are better for images of all classes in case of NSDFrFT-Bic Intr. in comparison to DFrFT for higher compression percentages. The optimized values of MSSIM and GMSD shows that the reconstructed image comprises of high structural similarity to original image along with high subjective image quality for low frequency images compressed via NSDFrFT-Bic Intr.

## 4.6 SUMMARY

An algorithm to compress an image using NSDFrFT has been proposed. In this work, the effect of interpolation on the performance of NSDFrFT has been studied and it was found that for an image divided into a block of $8 \times 8$, the NSDFrFT definition utilizing Bicubic Interpolation for mapping purposes performed better for higher compression percentages. The compressed image obtained comprises of blocking artifacts. Thus, the effect of fractional transform techniques on the extent of blocking artifacts has to be analyzed.

# CHAPTER 5

# REDUCTION OF BLOCKING ARTIFACTS

## 5.1 INTRODUCTION

In accordance to transform coding, an image is divided into sub-images known as blocks. If an image is of $N \times N$ size then blocks will be of $n \times n$ size where $n$ can be 4, 8, 16, 32 and so on. Each block is processed independently [5]. 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 become visible. This visual impairment is known as blocking artifacts.

The main cause of blocking artifact is the loss in the accuracy of transform coefficient resulted from the independent quantization of each block. During the process of quantization a large number of transform coefficients are discarded. And when image is converted back into spatial domain, quantization error gets spread 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 the plain areas or slowly varying portions of the image.

In image compression procedure, there is always a trade-off between the coding bit rate and coded image quality. It is generally observed that with increase in coding bit rate, the quality of reconstructed image improves. But, this improvement is restricted by limited transmission bandwidth and storage space available. To improve the visual quality of image, many blocking artifact removal algorithms has been proposed such as DCT filtering, spatial averaging method, wavelet filtering and reconstruction technique for reconstructed image [9]. These algorithms can be categorized into pre-processed methods which are applied in the spatial domain before encoding and post-processing methods which comprises of technique that can be applied in both transform domain and spatial domain as it is done on reconstructed image at decoding end.

## 5.2 CALCULATION OF BLOCKED MEAN SQUARE ERROR

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 [73].

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

$$MSE_h + MSE_v = \sqrt{\frac{1}{MN}\sum_{i=8}^{9}\sum_{j=8}^{9}[u(i,j) - v(i,j)]^2} \tag{5.1}$$

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.

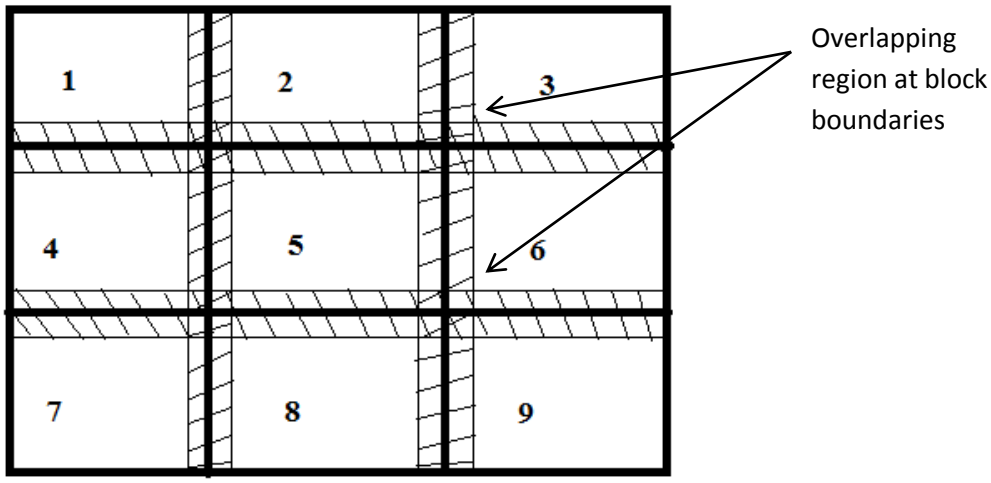



**Figure 5.1: Horizontal and vertical boundaries of nine blocks**

## 5.3 COMPARISON OF VARIOUS TRANSFORM TECHNIQUES

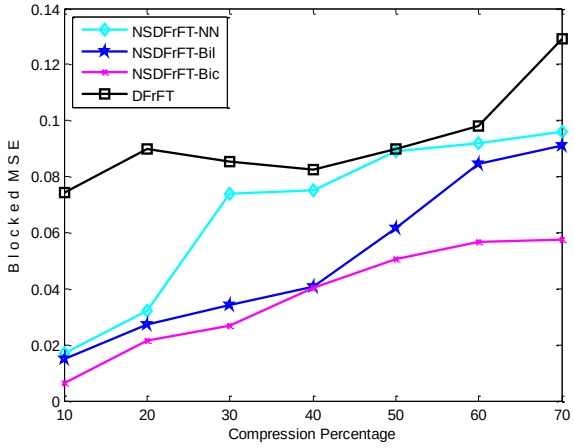
The $MSE_{block}$ for different images at different compression percentages using DFrFT, NSDFrFT-NN Intr., NSDFrFT-Bil Intr. and NSDFrFT-Bic Intr. has been complied in the Table 5.1. The comparison of different transform techniques is shown in Figure. 5.2.

**Table 5.1: Blocked MSE ($MSE_{block}$) of different images at different compression percentages with different transform techniques**

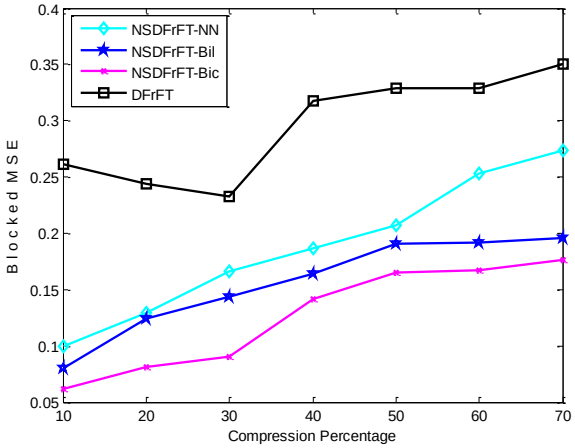
| Compression Percentage | Transform Techniques | $MSE_{block}$ of different images | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Baboon | Barbara | Boat | Lighthouse | Grass | House | Lena | Pepper |
| 10 | DFrFT | 0.0745 | 0.2609 | 0.6555 | 0.1329 | 0.2124 | 0.3141 | 0.1052 | 0.1233 |
| | NSDFrFT-NN Intr. | 0.0169 | 0.0996 | 0.4346 | 0.0578 | 0.1300 | 0.2732 | 0.0650 | 0.0574 |
| | NSDFrFT-Bil Intr. | 0.0150 | 0.0806 | 0.3859 | 0.0450 | 0.1329 | 0.1361 | 0.0568 | 0.0477 |
| | NSDFrFT-Bic Intr. | **0.0136** | **0.0614** | **0.3465** | **0.0243** | **0.1256** | **0.1316** | **0.0485** | **0.0239** |
| 20 | DFrFT | 0.0897 | 0.2444 | 0.6805 | 0.1479 | 0.2330 | 0.3203 | 0.1190 | 0.1262 |
| | NSDFrFT-NN Intr. | 0.0321 | 0.1296 | 0.4896 | 0.0988 | 0.2228 | 0.2763 | 0.0753 | 0.0740 |
| | NSDFrFT-Bil Intr. | 0.0272 | 0.1239 | 0.4064 | 0.0590 | 0.1566 | 0.1815 | 0.0645 | 0.0545 |
| | NSDFrFT-Bic Intr. | **0.0216** | **0.0810** | **0.3795** | **0.0526** | **0.1476** | **0.1741** | **0.0582** | **0.0273** |
| 30 | DFrFT | 0.0855 | 0.2325 | 0.7016 | 0.1810 | 0.2402 | 0.3214 | 0.1284 | 0.1206 |
| | NSDFrFT-NN Intr. | 0.0737 | 0.1661 | 0.5567 | 0.1074 | 0.2296 | 0.2771 | 0.0826 | 0.0758 |
| | NSDFrFT-Bil Intr. | 0.0341 | 0.1438 | 0.4588 | 0.0688 | 0.1749 | 0.2645 | 0.0705 | 0.0615 |
| | NSDFrFT-Bic Intr. | **0.0268** | **0.0908** | **0.3982** | **0.0605** | **0.1521** | **0.1783** | **0.0604** | **0.0374** |
| 40 | DFrFT | 0.0825 | 0.3175 | 0.7091 | 0.1836 | 0.2611 | 0.3219 | 0.1304 | 0.1140 |
| | NSDFrFT-NN Intr. | 0.0752 | 0.1870 | 0.5741 | 0.1253 | 0.2592 | 0.2958 | 0.0863 | 0.0758 |
| | NSDFrFT-Bil Intr. | 0.0408 | 0.1642 | 0.4882 | 0.1135 | 0.2492 | 0.2777 | 0.0830 | 0.0640 |
| | NSDFrFT-Bic Intr. | **0.0405** | **0.1421** | **0.4489** | **0.0725** | **0.1872** | **0.2141** | **0.0753** | **0.0385** |

**Table 5.1: Blocked MSE ($MSE_{block}$) of different images at different compression percentages with different transform techniques (contd.)**

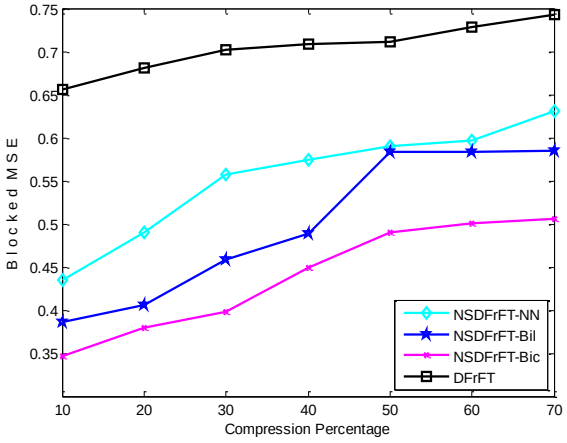
| Compression Percentage | Transform Techniques | $MSE_{block}$ of different images | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Baboon | Barbara | Boat | Lighthouse | Grass | House | Lena | Pepper |
| 50 | DFrFT | 0.0897 | 0.3291 | 0.7109 | 0.1854 | 0.2833 | 0.3401 | 0.1367 | 0.1128 |
| | NSDFrFT-NN Intr. | 0.0891 | 0.2071 | 0.5903 | 0.1434 | 0.2581 | 0.3104 | 0.1031 | 0.0762 |
| | NSDFrFT-Bil Intr. | 0.0618 | 0.1908 | 0.5830 | 0.1310 | 0.2554 | 0.2822 | 0.0929 | 0.0705 |
| | NSDFrFT-Bic Intr. | **0.0505** | **0.1652** | **0.4901** | **0.0840** | **0.2278** | **0.2159** | **0.0768** | **0.0405** |
| 60 | DFrFT | 0.0980 | 0.3292 | 0.7279 | 0.1876 | 0.2899 | 0.3440 | 0.1640 | 0.1895 |
| | NSDFrFT-NN Intr. | 0.0920 | 0.2528 | 0.5961 | 0.1497 | 0.2889 | 0.3108 | 0.1049 | 0.1170 |
| | NSDFrFT-Bil Intr. | 0.0845 | 0.1919 | 0.5831 | 0.1347 | 0.2694 | 0.2838 | 0.1024 | 0.0744 |
| | NSDFrFT-Bic Intr. | **0.0567** | **0.1672** | **0.5001** | **0.0847** | **0.2309** | **0.2161** | **0.0928** | **0.0412** |
| 70 | DFrFT | 0.1290 | 0.3499 | 0.7427 | 0.1905 | 0.3068 | 0.3442 | 0.1867 | 0.2471 |
| | NSDFrFT-NN Intr. | 0.0960 | 0.2738 | 0.6312 | 0.1549 | 0.2925 | 0.3117 | 0.1327 | 0.2021 |
| | NSDFrFT-Bil Intr. | 0.0910 | 0.1954 | 0.5851 | 0.1406 | 0.2921 | 0.2901 | 0.1295 | 0.1406 |
| | NSDFrFT-Bic Intr. | **0.0576** | **0.1762** | **0.5064** | **0.0858** | **0.2329** | **0.2182** | **0.1091** | **0.0593** |



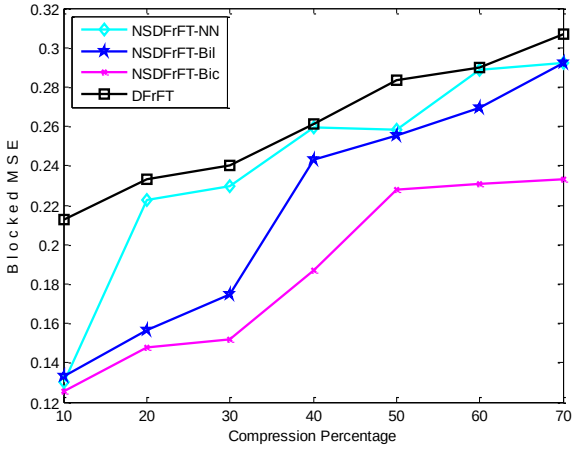

**(a) For Baboon Image**



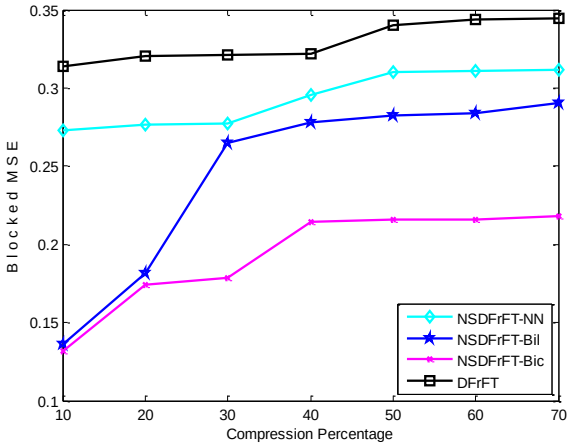

**(b) For Barbara Image**



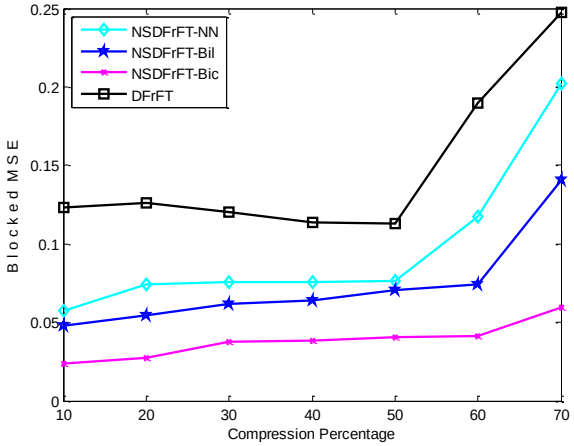

**(c) For Boat Image**



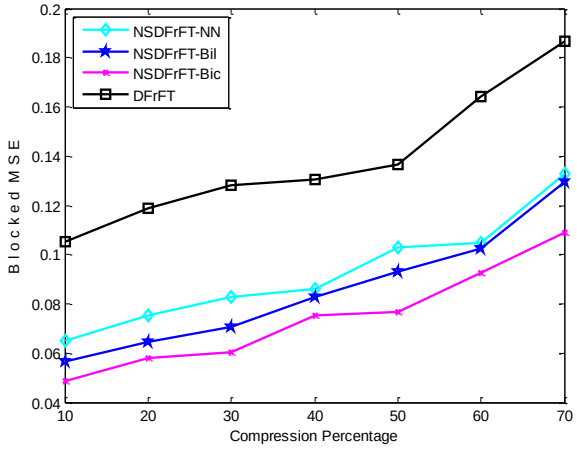

**(d) For Grass Image**



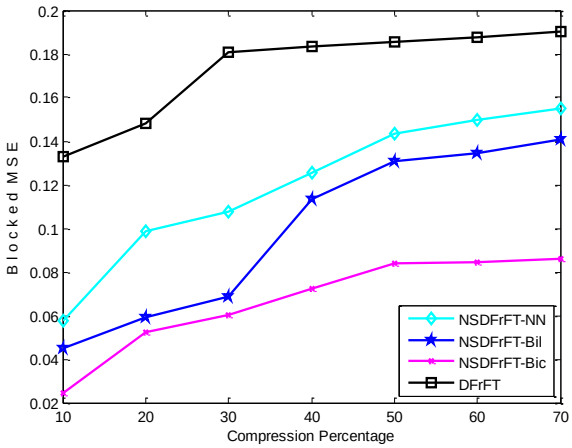

**(e) For House Image**




**(f) For Pepper Image**




**(g) For Lena Image**




**(h) For Lighthouse Image**

**Figure 5.2: Compression vs Blocked MSE ($MSE_{block}$) graph for images comparing the techniques**

In Table 5.1 $MSE_{block}$ has been calculated for eight different images for five different transform techniques at various compression percentages varying from 10% to 70%. It was observed that as the compression percentage increases from 10% to 70%, the $MSE_{block}$ also increases for a particular image. Since the large value of MSE points towards degradation in the quality of image thus, increasing value of $MSE_{block}$ implies that as the compression increases the degree of blockiness increases.

The comparison of different transform techniques for different images concludes that all the definitions of NSDFrFT i.e. NSDFrFT-NN Intr., NSDFrFT-Bil Intr. and NSDFrFT-Bic Intr. 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.

The graphical representation of comparison of different transform techniques for each of the image has been complied in Figure 5.2. The graph of a particular image depicts

the difference in $MSE_{block}$ value computed for different fractional transform techniques. For example: In the comparison graph of image Lena 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.

## 5.4 SUMMARY

The procedure of block processing results in the visual impairment known as blocking artifact in the reconstructed image. The non-separable definition of FrFT with different interpolation techniques reduces blocking in the compressed image much more efficiently than any other transform technique. However, among these three definitions of NSDFrFT, it was concluded that NSDFrFT-Bicubic Interpolation reduces blocking in the reconstructed image more profoundly than other two definitions of NSDFrFT.

# CHAPTER 6

# CONCLUSION AND FUTURE SCOPE

## 6.1 CONCLUSION

This thesis presents the performance analysis of the NSDFrFT in the compression of an image to justify the practical utility and effectiveness of the technique in reduction of blocking artifacts. The summarized results of the analysis states that NSDFrFT with different interpolation methods resulted in high image quality parameters than DFrFT with relatively high GMSD score. For Lena and Pepper, an improvement of 0.39 dB and 4.29 dB in PSNR respectively has been achieved. Among different types of images used for analysis, low frequency images responded well for NSDFrFT-Bicubic Interpolation than DFrFT. The collective results of all image quality parameters suggested that NSDFrFT-Bicubic Interpolation performs better for higher 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.

Thus, it can be concluded that NSDFrFT-Bicubic Interpolation has better performance at higher compression percentages for low frequency images resulting in high visual quality images with reduced blocking artifacts occupying less storage space and transmission bandwidth at reduced cost.

## 6.2 FUTURE SCOPE

Evolution of ideas is a continuous process and thus, some aspects in the proposed algorithm can be improved to achieve an enhanced compression algorithm. As discussed in previous chapters, interpolation is a key aspect in the definition of NSDFrFT. Thus, improving the interpolation technique will improve the performance of NSDFrFT. Adaptive interpolation technique can be a way to achieve highly improved performance. Another aspect that will need improvement is the required computational time for the encoding and decoding of the image while performing the compression.

# REFERENCES


[1] K. A. McIntyre, "Dynamic Bandwidth Adaptive Image Compression/ Decompression Scheme," U.S. Patent 7024045, 2006.

[2] T. L. B. Yng, B. G. Lee and H. Yoo, "A Low Complexity and Lossless Frame Memory Compression for Display Devices," *IEEE Trans. Consumer Electronics*, vol. 54, no. 3, pp. 1453-1458, 2008.

[3] J. Shukla, M. Alwani and A. K. Tiwari, "A Survey on Lossless Image Compression Methods," in *Proc. 2nd International Conference on Computer Engineering and Technology* (ICCET), vol. 6, pp. 136-141, 2010.

[4] S. Jayaraman, S. Esakkirajan and T. Veerakumar, *Digital Image Processing*, Tata McGraw-Hill Education, 2011.

[5] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd edition, 2008.

[6] K. S. Thyagarajan, *Still Image and Video Compression with MATLAB*, John Wiley & Sons, Inc. Hoboken, New Jersey, 2011.

[7] Dr. E. Kannan and G. Murugan, "Lossless Image Compression Algorithm for Transmitting Over Low Bandwidth Line," *International Journal of Advanced Research in Computer Science and Software Engineering* (IJARCSSE), vol. 2, no. 2, pp. 1-6, 2012.

[8] S. K. Bandyopadhyay, T. U. Paul and A. Rajchoudhury, "Image Compression using Approximate Matching and Run Length," *International Journal of Advanced Computer Science Applications* (IJACSA), vol. 2, no. 6, pp. 117-121, 2011.

[9] I. E. G. Richardson, *H.264and MPEG-4 Video Compression*, John Wiley & Sons, Inc. West Sussex, England, 2003.

[10] K. Singh, "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] N. Wiener, "Hermitian Polynomials and Fourier Analysis," *Journal of Mathematics and Physics*, vol. 8, pp. 70-73, 1929.

[12] V. Namias, "The Fractional Order Fourier Transform and its Applications to Quantum Mechanics," *IMA Journal of Applied Mathematics*, vol. 25, no. 3, pp. 241-265, 1980.

[13] H. M. Ozaktus and D. Mendlovic, “Fractional Fourier Optics,” *Journal of the Optical Society of America A*, vol. 12, no. 4, pp. 743-751, 1995.

[14] G. Cariolario, T. Ersrghe, P. Kraniauskas, and N. Laurenti, “A unified framework for the fractional Fourier transform,” *IEEE Trans. Signal Processing*, vol. 46, no. 12, pp. 3206-3212, 1998.

[15] S. C. Pei, M. H. Yeh and T. L. Luo, “Fractional Fourier Series Expansion for Finite Signals and Dual Extension to Discrete-Time Fractional Fourier Transform,” *IEEE Trans. Signal Processing*, vol. 47, no. 10, pp. 2883, 1999.

[16] S. C. Pei and M. H. Yeh, “Discrete Fractional Fourier Transform,” in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 536-539, 1996.

[17] B. Santhanam, and J. H. McClellan, “DFRFT-a rotation in time-frequency space,” in *Proc. 20th International Conference of Acoustics, Speech Signal Processing*, vol. 2 , pp. 921–924, 1995.

[18] S. C. Pei and J. J. Ding, “Closed-Form Discrete fractional and Affine Fourier Transforms,” *IEEE Trans. Signal Processing*, vol. 48, no. 5, pp. 1338-1353, 2000.

[19] C. Candan, M. A. Kutay and H. M. Ozaktus, “The Discrete Fractional Fourier Transform,” *IEEE Trans. Signal Processing*, vol. 48, no. 5, pp. 1329-1338, 2000.

[20] A. Sahin, M. A. Kutay and H. M. Ozaktus, “Nonseparable two-dimensional fractional Fourier transform,” *Journal of the Optical Society of America A* , vol. 37, no. 23, pp. 5444- 5453, 1998.

[21] J. J. Ding and S. C. Pei, “Heisenberg’s uncertainty principles for the 2-D non-separable linear canonical transforms,” *Signal Processing*, vol. 93, pp. 1027-1043, 2013.

[22] G. K. Wallace, “Overview of the JPEG (ISO/CCITT) Still Image Compression Standard,” in *Proc. SPIE Image Processing Algorithm Techniques*, vol. 1244, pp. 220-233, 1990.

[23] G. K. Wallace, “The JPEG Still Picture Compression Standard,” *IEEE Trans. Consumer Electronics*, vol. 38, no. 1, pp. 30-44, 1991.

[24] B. Usevitch, "A Tutorial on Modern Lossy Wavelet Image Compression: Foundations of JPEG 2000," *IEEE Trans. Signal Processing*, vol. 18, no. 5, pp. 22-35, 2001.

[25] I. S. Yetik, M. A. Kutay and H. M. Ozaktas, "Image Representation and Compression with the Fractional Fourier Transform," *Optics Communications*, vol. 197, pp. 275-278, 2001.

[26] C. Vijaya and J. S. Bhat, "Signal Compression using Discrete Fractional Fourier Transform and Set Partitioning in Hierarchical Tree," *Signal Processing*, vol. 86, no. 8, pp. 1976-1983, 2006.

[27] T. Cebrail and S. Serder, "An Overview of Image Compression Approaches," in *Proc.* 3r*d International Conference on Digital Telecommunications*, vol. 4, pp. 174-179, 2008.

[28] A. Alfalou, M. Elbouz, A. Mansour and G. Keryer, "New Spectral Image Compression Method based on an Optimal Phase Coding and the RMS duration Principle," *Journal of Optics*, vol. 12, no. 11, pp. 1-12, 2010.

[29] D. P. Dutta, S. D. Choudhury, Md. A. Hussain and S. Majumder, "Digital Image Compression using Neural Networks," in *Proc. International Conference on Advances in Computing, Control, Telecommunication Technology* (ACT'09), pp. 116-120, 2009.

[30] Y. Li, Z. Zhang and Y. Li, "Recovery of the Optimal Approximation from Samples in Wavelet Subspace," *Digital Signal Processing*, vol. 22, no. 5, pp. 795-807, 2012.

[31] N. Jindal and K. Singh, "Joint Image Compression-Encryption Using Discrete Fractional Transforms," *The Image Science Journal*, vol. 62, no. 5, pp. 265-272, 2013.

[32] N. Jindal and K. Singh, "Image Retrieval Algorithm based on Discrete Fractional Transforms," *Journal of Electrical Engineering*, vol. 64, no. 4, pp. 250-255, 2013.

[33] H. Reeve and J. Lim, "Reduction of Blocking Effect in Image Coding," in *Proc. IEEE International Conference on Acoustics, Speech, Signal Processing* (ICASSP), vol. 8, pp. 1212–1215, 1983.

[34] A. Zakhor, "Iterative Procedure for Reduction of Blocking Effects in Transform Image Coding," *IEEE Trans. Circuits and Systems on Video Technology*, vol. 2, no. 1, pp. 91–95, 1992.

[35] H. Paek, R. C. Kim and S. U. Lee, "A DCT-Based Spatially Adaptive Post Processing Technique to Reduce the Blocking Artifacts in Transform Coded Images," *IEEE Trans. Circuits and Systems on Video Technology*, vol. 10, no. 1, pp. 36–41, 2000.

[36] S. Singh, V. Kumar and H. K. Verma, "Reduction of Blocking Artifacts in JPEG Compressed Images," *Digital Signal Processing*, vol. 17, pp. 225–243, 2007.

[37] A. W. C. Liew and H. Yan, "Blocking Artifacts Suppression in Block-Coded Images Using Over Complete Wavelet Representation," *IEEE Trans. Circuits and System on Video Technology*, vol. 14, no. 4, pp. 450-461, 2004.

[38] Y. L. Lee, H. C. Kim and H. W. Park, "Blocking Effect Reduction of JPEG Images by Signal Adaptive Filtering," *IEEE Trans. Image Processing*, vol. 7, no. 2, pp. 229–234, 1998.

[39] S. Singh, "Reduction of Blocking Artifacts in JPEG Compressed Images," in *Proc. National Level Technical Symposium on Emerging Trends in Technology* (TECHNOVISION'10), pp. 234-243, 2010.

[40] A. Purushothaman, Dr. K. R. S. Kumar, Dr. R. Rangarajan and Dr. A. Kandasawamy, "Block Artifact Reduction in Low Bit Rate Images and Videos," *International Journal of Computer Science and Network Security* (IJCSNS), vol. 9, no. 12, pp. 96-101, 2012.

[41] Dr. S. S. Pandey, M. P. Singh and V. Pandey, "Block wise image compression & Reduced Blocks Artifacts Using Discrete Cosine Transform," *International Journal of Science Research Publications*, vol. 5, no. 3, pp. 1-10, 2015.

[42] Y. F. Hsu and Y. C. Chen, "A New Adaptive Separable Median Filter for Removing Blocking Effects," *IEEE International Conference on Consumer Electronics*, vol. 39, pp. 510-513, 1993.

[43] S. Minami and A. Zakhor, "An Optimization Approach for Removing Blocking Effects in Transform Coding," *IEEE Trans. Circuits and Systems on Video Technology*, vol. 5, pp. 74–82, 1995.

[44] T. Meier, K. N. Ngan and G. Crebbin, "A Region-Based Algorithm for Enhancement of Images Degraded by Blocking Effects," *IEEE TENCON. Digital Signal Processing Applications*, vol. 1, pp. 405–408, 1996.

[45] J. Chou, M. Crouse and K. Ramchadran, "A Simple Algorithm for Removing Blocking Artifacts in Block Transform Coded Images," *IEEE Signal Processing Letters*, vol. 5, no. 2, pp. 33–35, 1998.

[46] Y. Luo and R. K. Ward, "Removing The Blocking Artifacts of Block Based DCT Compressed Images," *IEEE Trans., Image Processing*, vol. 12, no. 7, pp. 838–842, 2003.

[47] J. Singh, S. Singh, D. Singh and M. Uddin, "Efficient DCT-Domain Blind Measurement of Blocking Artifacts," *The Journal of Information and Computing Science*, vol. 5, pp. 47-54, 2010.

[48] Y. Dandiwal and K. Sachdeva, "The Study of Various Approaches for Removal of Blocking Artifacts in Spatial Domain," *International Journal of Computer Science and Mobile Computing* (IJCSMC), vol. 2, no. 12, pp. 350-358, 2013.

[49] G. Sansone, *Orthogonal Functions*, InderScience Publishers, New York, 1959.

[50] A. C. McBride and F. H. Kerr, "On Namias's Fractional Fourier Transform," *IMA Journal of Applied Mathematics*, vol. 39, pp. 159-175, 1987.

[51] B. L. Almeida, "The Fractional Fourier Transform and Time-frequency Representations," *IEEE Trans. Signal Processing*, vol. 42, no. 11, pp. 3084-3091, 1994.

[52] H. M. Ozaktas, M. A. Kutay and D. Mendlovic, "Introduction to the Fractional Fourier Transform and its Applications," *Advances in Imaging and Electron Physics*, vol. 106, pp. 239-291, 1999.

[53] M. A. Kutay, H. M. Ozaktas, O. Arikan and L. Onural, "Optimal filtering in Fractional Fourier Domain," *IEEE Trans. Signal Processing*, vol. 45, no. 3, pp. 1129-1143, 1997.

[54] J. Hu, J. Deng and J. Wu, "Image Compression Based on Improved FFT Algorithm," *Journal of Networks*, vol. 6, no. 7, pp. 1041-1048, 2011.

[55] M. T. Hanna, N. P. A. Seifand and W. A. E. M. Ahmed, "Hermite-Gauss like Eigenvectors of the Discrete Fourier Transform Matrix based on Singular Value Decomposition of its Orthogonal Projection Matrices," *IEEE Trans.*

*Circuits and Systems I-Regular Papers*, vol. 51, no. 11, pp. 2245-2254, 2004.

[56] J. G. V. Rubio and B. Santhanam, "On the Multiangle Centered Discrete Fractional Fourier Transform," *IEEE Trans. Signal Processing*, vol. 12, no. 4, pp. 273-276, 2005.

[57] A. Serbes and L. D. Ata, "The discrete fractional Fourier transform based on the DFT matrix," *Signal Processing*, vol. 91, pp. 571-581, 2011.

[58] A. K. Singh and R. Saxena, "DFRFT: A Classified Review of Recent Methods with Its Applications," *Journal of Engineering*, vol. 13, pp. 1-13, 2013.

[59] A. Koc, H. M. Ozaktas and L. Hesselink, "Fast and accurate computation of two-dimensional non-separable quadratic-phase integrals," *Journal of the Optical Society of America A*, vol. 27, pp. 1288–1302, 2010.

[60] Q. Ran, L. Yuan and T. Zhao, "Image encryption based on nonseparable fractional Fourier transform and chaotic map," *Optical Commun*ications, vol. 348, pp. 43-49, 2015.

[61] A. Prajapati, S. Naik and S. Mehta, "Evaluation of Different Image Interpolation Algorithms," *International Journal of Computer Applications* (IJCA), vol. 58, no. 12, pp. 6-12, 2012.

[62] R. Olivier and C. Hanqiang, "Nearest Neighbor Value Interpolation," *International Journal of Advanced Computer Science and Applications* (IJACSA), vol. 3, no. 4, pp. 1-6, 2012.

[63] J. A. Parker, R. V. Kenyon and D. E. Troxel, "Comparison of Interpolating Methods for Image Resampling," *IEEE Trans. Medical Imaging*, vol. MI-2, no. 1, pp. 31-39, 1983.

[64] M. Unser, P. Thevenaz and L. Yaroslavsky, "Convolution based Interpolation for Fast, High Quality Rotation of Images," *IEEE Trans. Image Processing*, vol. 4, no. 10, pp. 1371-1381, 1995.

[65] A. Skodras, C. Christopoulos and T. Ebrahimi, "The JPEG 2000 still Image Compression Standard," *IEEE Trans. Signal Processing*, vol. 18, no. 5, pp. 36-58, 2001.

[66] A. B. Watson, "Image Compression using the Discrete Cosine Transform," *Mathematica Journal*, vol. 4, no. 1, pp. 81-88, 1994.

[67] N. Jindal and K. Singh, "Image and Video Processing using Discrete Fractional Transforms," *Signal Image and Video Processing*, DOI: 10.1007/s11760-012-0391-4, 2012.

[68] S. Grgic, M. Grgic and B. Z. Cihlar, "Performance Analysis of Image Compression using wavelets," *IEEE Trans. Industrial Electronics*, vol. 48, no. 3, pp. 682-695, 2001.

[69] Z. Whang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600-612, 2004.

[70] W. Xue, L. Zhang, X. Mou and A. C. Bovik, "Gradeint Magnitude Similarity Deviation: An Highly Efficient Perceptual Image Quality Index," *IEEE Trans. Image Processing*, vol. 23, no. 2, pp. 684-695, 2014.

[71] "USC-SIPI Image Database", *http://sipi.usc.edu/database/*, 2015.

[72] S. Rout, "Orthogonal vs. Biorthogonal Wavelets for Image Compression," M.S. Thesis, Dept. of Elect. Eng., Virginia Tech, Blacksburg, Virginia, 2003.

[73] N. Jindal, "Performance of Fractional Transforms in Image and Video Processing," Ph.D Thesis, Department of Electronics and Communication Engineering, Thapar University, Patiala, India, 2015.

[74] C. C. Chen, "On the selection of image compression algorithms," in *Proc. IEEE 14$^{th}$ International Conference on Pattern Recognition* (ICPR'98), pp. 1-5, 1988.

# LIST OF PUBLICATIONS

1. Mittal K. and Singh K., Low Frequency Image Compression via Non-separable Discrete Fractional Fourier Transform, in Proceedings of IEEE Fourth International Conference on Advances in Computing, Communications and Informatics (ICACCI-2015)- *Accepted*, 2015.

2. Mittal K. and Singh K., Image Compression using 2D Non-Separable Discrete Fractional Fourier Transform, Multimedia Tools and Applications- *Communicated,* 2015. *(SCI Indexed)*

# Turnitin Originality Report

ME Thesis by Kritika Mittal

From Quick Submit (Quick Submit)

Processed on 07-Jul-2015 12:42 IST
ID: 554458754
Word Count: 11346

| Similarity Index | Similarity by Source | |
|---|---|---|
| 7% | Internet Sources: | 3% |
| | Publications: | 6% |
| | Student Papers: | 1% |

**sources:**

1 < 1% match (publications)
Singh, Ashutosh Kumar, and Rajiv Saxena. "DFRFT: A Classified Review of Recent Methods with Its Application", Journal of Engineering, 2013.

2 < 1% match (publications)
Aysegul Sahin. "Nonseparable Two-Dimensional Fractional Fourier Transform", Applied Optics, 08/10/1998

3 < 1% match (Internet from 21-Apr-2014)
http://www.worldacademicunion.com/journal/1746-7659JIC/jicvol8no2paper03.pdf

4 < 1% match (student papers from 26-Apr-2011)
Submitted to Higher Education Commission Pakistan on 2011-04-26

5 < 1% match (publications)
Xu, Y.. "Stress analysis of multi-phase and multi-layer plain weave composite structure using global/local approach", Composite Structures, 201004

6 < 1% match (student papers from 10-Sep-2013)
Submitted to Institute of Technology, Nirma University on 2013-09-10

7 < 1% match (Internet from 30-May-2012)
http://istrides.in/2004.html

8 < 1% match (publications)
Singh, S.. "Reduction of blocking artifacts in JPEG compressed images", Digital Signal Processing, 200701

9 < 1% match (Internet from 10-Jan-2008)
http://www.ece.umassd.edu/Faculty/acosta/ICASSP/Icassp_1999/PDF/AUTHOR/IC991219.PDF

10 < 1% match (Internet from 17-Apr-2015)