BLOCKING ARTIFACTS REDUCTION IN JPEG COMPRESSED IMAGES USING OVERLAPPED BLOCK PROCESSING

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

Master of Engineering

in

Wireless Communication

Submitted by

SIMRANJIT SINGH

Roll No: 801263025

Under the guidance of

Dr. KULBIR SINGH

Associate Professor

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

(Established under the section 3 of UGC Act, 1956)

PATIALA – 147004 (PUNJAB)

DECLARATION

I hereby declare that the work which is being presented in the dissertation entitled, **"Blocking artifacts reduction in JPEG compressed images using overlapped block processing"** in partial fulfillment of the requirement for the award of degree of Master of Engineering in Wireless Communication submitted in Electronics and Communication Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Kulbir Singh, Associate Professor, ECED and refers other researcher's work which are duly listed in the reference section.

The matter presented in this dissertation has not been submitted in any other University/Institute for the award of degree.

Date: June 26, 2014

(SIMRANJIT SINGH)
Roll No: 801263025

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

(Dr. KULBIR SINGH)
Associate Professor
ECED, Thapar University

Countersigned by:

26/6/2014

Head
ECED, Thapar University
Patiala-147004

Dean of Academic Affairs
Thapar University
Patiala- 147004

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to **Dr. KULBIR SINGH**, **Associate Professor**, Electronics & Communication Engineering Department, Thapar University, Patiala for his guidance and support throughout this thesis work. I am really very fortunate to have the opportunity to work with him. I found his guidance to be extremely valuable.

I am thankful to the **Head of Department, Professor (Dr.) SANJAY SHARMA** of Electronics & Communication Engineering Department for their encouragement and inspiration for the execution of this thesis work.

I am also thankful to the entire faculty and staff of Electronics & Communication Engineering Department for the help and moral support which went along the way for the successful completion of this thesis work.

My greatest thanks are to all who wished me success especially my parents. Above all I render my gratitude to the Almighty who bestowed self-confidence, ability and strength in me to complete this work for not letting me down at the time of crisis and showing me the silver lining in the dark clouds. I do not find enough words with which I can express my feelings of thanks to my dear friends for their help, inspiration and moral support which went a long way in successful competition of the present study.

Simranjit Singh
801263025

ABSTRACT

In recent years, there has been a huge interest in compression of digital image data due to the fact that a large number of communication applications contain pictures or video. Thus, image compression is considered very useful in multimedia applications and in distributed information systems that operate in network environments. One of the most commonly used image format, JPEG, uses the concept of transform coding or more specifically Block based Discrete Cosine Transform (BDCT). Despite their widespread popularity BDCT compression scheme leads to noticeable blocking artifacts at high compression levels due to discontinuities between sub-images. Prior research shows that incorporating systems that reduce blocking artifacts into codecs is useful because visual quality is improved. Existing methods reduce blocking artifacts by applying various pre and post processing techniques to the image.

This dissertation examines a framework where blocking artifacts are reduced by means of overlapped block processing. In the proposed method, the blocking artifacts problem is dealt with at the source end, namely, the segmentation process. In a typical block processing, an image is divided into mutually exclusive regions. Instead of constraining the regions to be exclusive of each other, it is rational that a slim overlap around the boundary of each region could reduce the blocking artifacts.

Objective quality of post processing images is measured using human visual system based peak signal to noise ratio (PSNR-HVS-M) instead of PSNR due to ineptness of the latter for real analysis purpose. An increment of 1.917 dB of PSNR-HVS-M at 90% compression level is observed for 256×256 Lena image by applying this algorithm using DCT with 7×7 inside block. However, it also increases the number of real arithmetic computations by 26.5%. An attempt is also made to apply the algorithm by means of Discrete Fractional Cosine Transform (DFrCT) as a transform coding technique which is generalised form of DCT (Type-I) and considered to play a vital role in the area of image processing in future because of its fractional order acting as an extra degree of freedom. However, DCT (Type-II) comes out to be better than DFrCT in case of image compression. The subjective and objective results for various images indicate that the proposed method performs better in terms of both visual quality and PSNR-HVS-M for all ranges of compression level.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

1-D	One Dimensional
2-D	Two Dimensional
BBM	Block Boundary Measure
BDCT	Block based Discrete Cosine Transform
CLS	Constrained Least Square
CM	Contrast Masking
CSF	Contrast Sensitivity Function
CTC	Combined Transform Coding
DAC	Divide And Conquer
DCT	Discrete Cosine Transform
DFOVS	Deblocking Frames Of Variable Size
DFrCT	Discrete Fractional Cosine Transform
DFrFT	Discrete Fractional Fourier Transform
DFT	Discrete Fourier Transform
DPCM	Differential Pulse Code Modulation
DST	Discrete Sine Transform
DTFT	Discrete Time Fourier Transform
DWT	Discrete Wavelet Transform
FFT	Fast Fourier Transform
FrFT	Fractional Fourier Transform
FT	Fourier Transform
HLBT	Hierarchical Lapped Bi-orthogonal Transform
HVS	Human Visual System
IDCT	Inverse Discrete Cosine Transform
JPEG	Joint Pictures Expert Group
LBT	Lapped Bi-orthogonal Transform
LIS	Lower Image Set
LOT	Lapped Orthogonal Transform
MAP	Maximum A Posteriori

MLP	Multilayer Perceptron
MPEG	Moving Picture Expert Group
MRF	Markov Random Field
MSDS	Mean Squared Difference of Slope
MSE	Mean Square Error
MSSIM	Mean Structure Similarity
NAI	Numerical Artifacts Indicator
NMSE	Normalized Mean Square Error
POCS	Projection On Convex Set
PSNR	Peak Signal to Noise Ratio
PSNR-HVS-M	Human Visual System based Peak Signal to Noise Ratio (Modified)
SF	Similarity Factor
UIS	Upper Image Set
VM	Verification Model
WABG	Weight Adaptation By Grading

INTRODUCTION | CHAPTER 1

Digital data communication plays a vital role in modern commerce, education and communication and efficient representation of data is crucial to meaningful use of the scarce resources available for communication. Different techniques for efficient data representation have been explored widely for a long time, under different headings such as signal compression, source coding and transform coding. Data compression also helps reduce the costs associated with storage of data. Digital image compression is a subset of general data compression and has seen sustained attention in the signal and image processing research community.

1.1 IMAGE COMPRESSION AND ITS NEED

An image is a positive function on a plane. Digital images are sampled versions of such functions, where the value is specified only at discrete locations. An analog image in a two dimensional continuous space can be converted into digital image in a discrete space through sampling. During digitization, it is divided into M rows and N columns. The intersection of a row and a column is termed as pixel. In fact this physical signal, which impinges on the face of a two dimensional sensor is actually a function of many variables including depth, luminance, color, and time. Each pixel specifies the luminance/ brightness of the picture at that point and to be represented with a pre-defined precision. Eight bits of precision for luminance is common [35] and is motivated by both the existing computer memory structures as well as by the dynamic range of human eye. The memory required for storage and network capability for transmission are high for digital images; thus, image compression is essential in many applications.

Image compression is concerned with minimizing the number of bits required to represent an image i.e. minimizing the redundancies. Most applications require high compression ratios with low bit rates. But this requirement in general is, in conjunction with the desire for high quality of the resulting content due to artifacts. Quality improvement can be achieved by eliminating the compression artifacts of the reconstructed image. Blocking artifacts of Joint Photographic Expert Group (JPEG) images [15] is one of the important artifact to be considered. The other issues are the preservation of edges of the image, preferably through high compression ratio and with low computations.

1.1.1 REDUNDANCIES

Image compression is largely possible by exploiting various kinds of redundancies [34] which are typically present in an image. The extent of redundancies may vary from image to image.

Redundancies in images may be categorized as follows –

(a) Coding redundancy: A code is a system of symbols (i.e. bytes, bits) that represents information. Each piece of information is represented by a set of code symbols. The gray level histogram of an image can be used in construction of codes to reduce the data used to represent it. The shortest code words are assigned to the most frequent (high probability) gray levels. The longest code words are assigned to the least frequent (low probability) gray levels. Data compression is achieved by assigning fewer bits to more probable gray levels than the less probable gray levels. In general, coding redundancy is present when the codes assigned to set of events (such as gray level values) have not been selected to take full advantage of the probabilities of the events,

(b) Inter-pixel redundancy: This type of redundancy is related with the inter-pixel correlations within an image. Much of the visual contribution of a single pixel is redundant and can be guessed from the values of its neighbours.

(c) Psychovisual Redundancy: Psychovisual redundancy arises due to the problem of perception. Human eyes are more responsive to slow and gradual changes of illumination than perceiving finer details and rapid changes of intensities. Hence, to what extent the details for human perception should be preserved and to what extent one can compromise on the quality of reconstructed image that human eye perceive is essentially carried out by exploiting the psychovisual redundancy.

Minimizing the redundancies leads to compression of an image. Lesser the redundancies, more will be the compression. One needs to exploit them quite sagely so as to get good compression without much loss of quality of the image.

1.1.2 CLASSIFICATION OF IMAGE COMPRESSION

In general, image compression schemes can be broadly classified into two categories, namely Lossless image compression & Lossy image compression. Both of them are briefly explained as below:

1.1.2.1 Lossless image compression

As the name implies, lossless image compression schemes exploit redundancies without incurring any loss of data. Thus, the data stream prior to encoding and after decoding is exactly the same and no distortion in the reconstruction quality is observed. Lossless image compression is therefore exactly reversible.

1.1.2.2 Lossy image compression

Contrary to lossless image compression, lossy image compression schemes incur loss of data and hence suffer a loss of quality in reconstruction. Like lossless image compression, the image is first transformed into a string of symbols, which are quantized to a discrete set of allowable levels. In this scheme, it is possible to achieve significant data compression, but quantization being a many-to-one mapping is irreversible and exact reconstruction is never possible.

1.2 ELEMENTS OF IMAGE COMPRESSION SYSTEM

Due to the high degree of redundancies present in a natural image, it can be compressed without significant degradation of the visual quality. A typical image compression system consists of the following elements:

(a) Transformer: This block transforms the original input data into a form that is more amenable to compression. The very basic purpose of a transformer (also known as mapper) is to remove or minimize the inter-pixel redundancy. The transformation can be local, involving pixels in the neighbourhood; or global, involving the full image or a block of pixels. The example of local transformation is linear predictive coding followed by Differential Pulse Code Modulation (DPCM). Global transformation techniques [37] use Discrete Fourier Transforms (DFT), Discrete Cosine Transforms (DCT), Discrete Wavelet Transforms (DWT) etc.

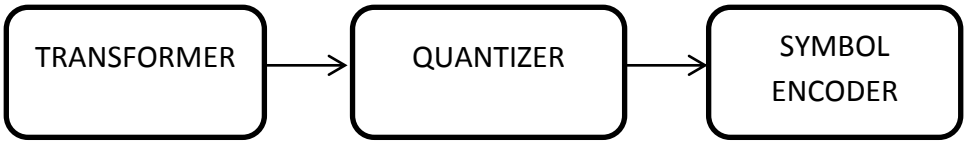


Figure 1.1 Image compression block diagram [35]

The transformer block transforms the original spatial domain signal into another spatial domain signal of reduced dynamic range, as is done in DPCM or into the

transform domain, where only a few coefficients contain bulk of the energy and efficient compression is possible. Please note that this block in itself does not perform any compression and is lossless.

(b) Quantizer: The quantizer follows the transformer block in image compression systems and generates a limited number of symbols that can be used in the representation of the transformed signal. It is a many-to-one mapping which is irreversible. Quantizers are of two basic types:

- Scalar quantization – refers to element-by-element quantization of data
- Vector quantization - refers to quantization of a block at a time.

Quantization exploits psychovisual redundancy and achieves significant bit reduction. It is the only block in image compression system, which is lossy.

(c) Symbol encoder: Coders assign a code word, a binary bit-stream, to each symbol at the output of the quantizer. A symbol encoder should be chosen such that coding redundancy is minimum.

The output of an image compression system is a bit-stream in an encoded form and hence cannot be displayed. In an image communication system, where the image is transmitted from one place to another, the encoded bit-stream is sent through a communication channel, which is ideally lossless but is noisy and hence lossy for all practical channels. At the receiver end, the encoded bit stream received through the communication channel has to be decoded before it can be displayed. The image de-compression system, also known as image decoding system should do exact reversal of the processes adopted during encoding.

An image de-compression system consists of the following elements –

- Image decoder – Performs exact reversal of the coder in image compression system. This block extracts the quantized coefficients.
- Inverse Transformer – Performs exact reversal of the transformation operation carried out in the corresponding image compression system. The output of this block can be used for display.

Inverse of quantizer is not possible, so there is no block of De-quantizer. In terms of compression, performance is seen to be better in transform-domain approaches, in which the pixel intensities are first mapped into a set of linear, reversible transform coefficients, which are subsequently quantized and encoded. The transform coefficients are de-

correlated and tend to pack most of the energy within few coefficients only. Thus, it is possible to achieve significant compression by either discarding the coefficients which do not carry much of the energy or, at least coarsely quantizing them.

1.3 TRANSFORM CODING

The basic principle of transform coding is to map the pixel values into a set of linear transform coefficients, which are subsequently quantized and encoded. By applying an inverse transformation on the decoded transform coefficients, it is possible to reconstruct the image with some loss. The basic block diagrams have been shown in Fig 1.2. The very first step in image compression with the technique of transform coding is segmentation or blocking. An $N \times N$ image is segmented using $n \times n$ block size. The explanation for this concept will be discussed shortly. After segmentation, the image is transformed to some domain other than spatial domain. Then after quantization and symbol encoder, compressed image is obtained as explained earlier. For reconstruction of image, the reverse of this procedure is followed as can be seen in Figure 1.2 (b).

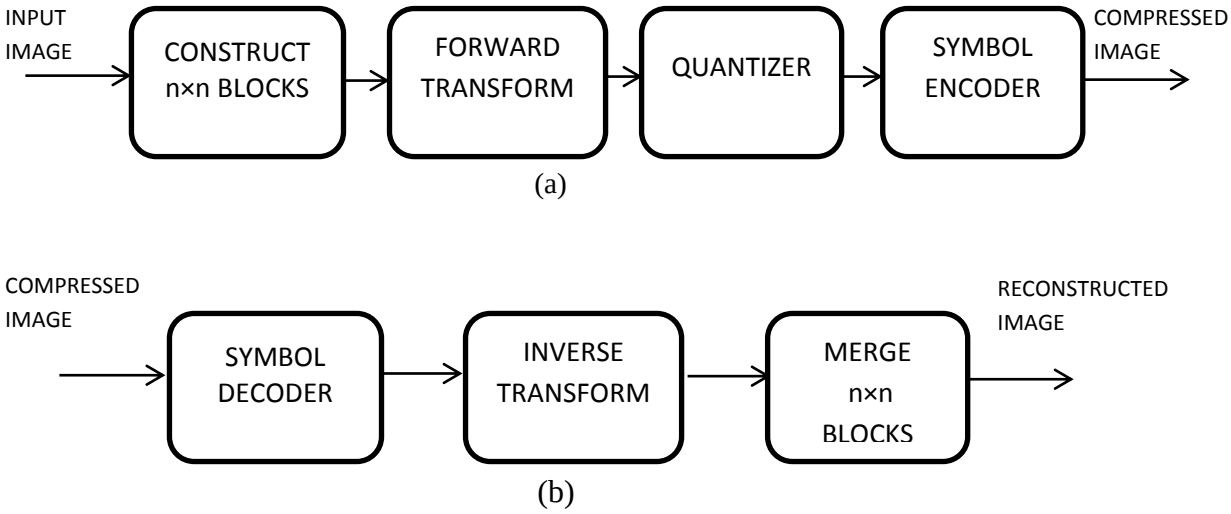


Figure 1.2 Transform coding system (a) encoder & (b) decoder [35]

It must be noted that the loss is not due to the process of transformation and inverse transformation, but due to quantization alone. A transformation must necessarily fulfil the following properties –

- The coefficients in the transformed space should be de-correlated.

- Only a limited number of transform coefficients should carry most of the signal energy (in other words, the transformation should possess energy compaction capabilities) and most of the coefficients should carry insignificant energy. Only then the quantization process can coarsely quantize those coefficients to achieve compression, without much of perceptible degradation.

A number of transformation techniques, such as Discrete Fourier Transforms (DFT), Discrete Cosine Transforms (DCT), Discrete Wavelet Transforms (DWT), Discrete Haar Transforms, and Discrete Hadamard Transforms etc. exist that fulfil the above properties, although their energy packing capabilities vary. Ever since the JPEG adopts DCT as its standard [21], it has been widely used for the compression of general images.

1.3.1 NEED OF BLOCK PROCESSING

For transform coding, the first and foremost step is to subdivide the image into non-overlapping blocks of fixed size. Since the details of an image and hence it's spatial frequency content vary from one local region to the other, it leads to a better coding efficiency if the transformation on local areas of the image is applied, rather than applying global transformation on the entire image. Such local transformations require manageable size of the hardware, which can be replicated for parallel processing.

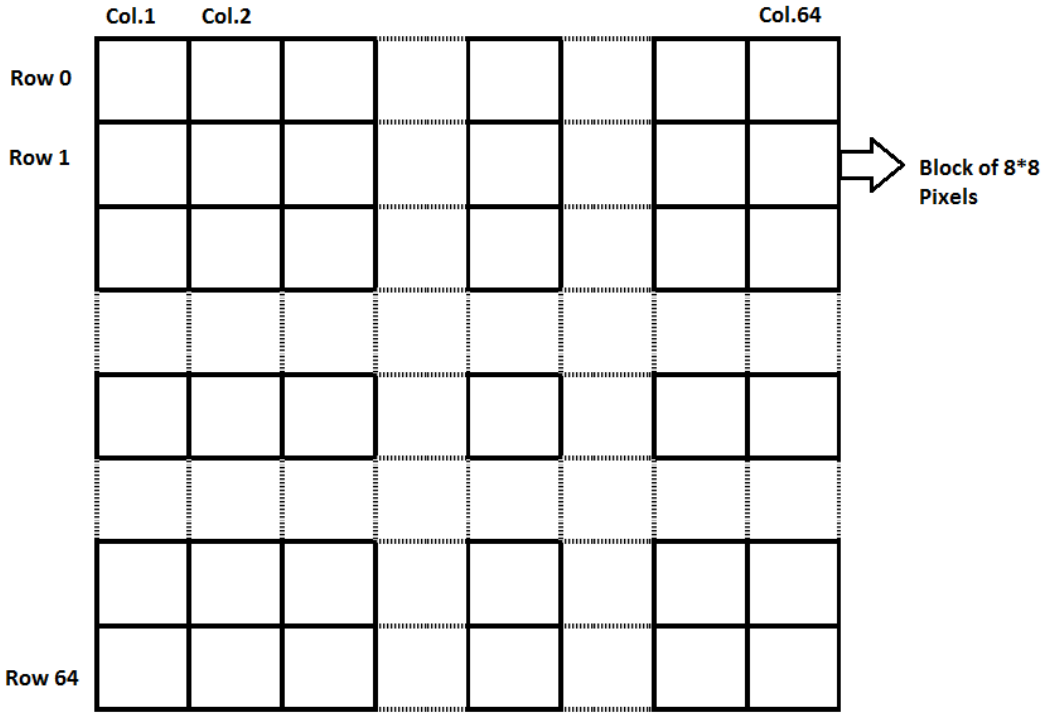


Figure 1.3 Dividing the 512×512 image into 64×64 blocks using 8×8 block size [42]

Computational complexity is another huge parameter which encourages the block processing. Block processing requires a lot lesser number of computations as compared to the usual one. Obviously, less the number of computations implies more will be the speed of system, hence block processing is the preferred one because of all above reasons.

Typically for a transform coding e.g. DFT or DCT, if fast algorithms are applied, the number of computations necessary are $N \log N$, where N is number of data values in a sequence of array [23]. For an $N \times N$ image, the required number of computations are $N^2 \log N$. Now, suppose an $N \times N$ image is divided into n^2 number of blocks each block size being $(N/n) \times (N/n)$ where $n << N$ and is a factor of N. For this kind of processing, the number of computations are $n^2 \left(\left(\frac{N}{n}\right)^2 \log \frac{N}{n}\right)$ which is very much lesser than $N^2 \log N$. To make things more clear let us assume a 512×512 image divided into 64×64 using block size of 8×8 as shown in the Figure1.3 The number of computations in this case and some other image sizes using blocks of size 8×8 is shown in Table 1.1

Table 1.1 Computation count for different image sizes using 8×8 block size

Image Size	Computations count	
	Without Block processing	With Block Processing
512×512	710218	118400
256×256	157810	59185
128×128	34525	29600

From the above discussions, It can be concluded that images use block based transform coding & it is largely based on the following observations [49]:

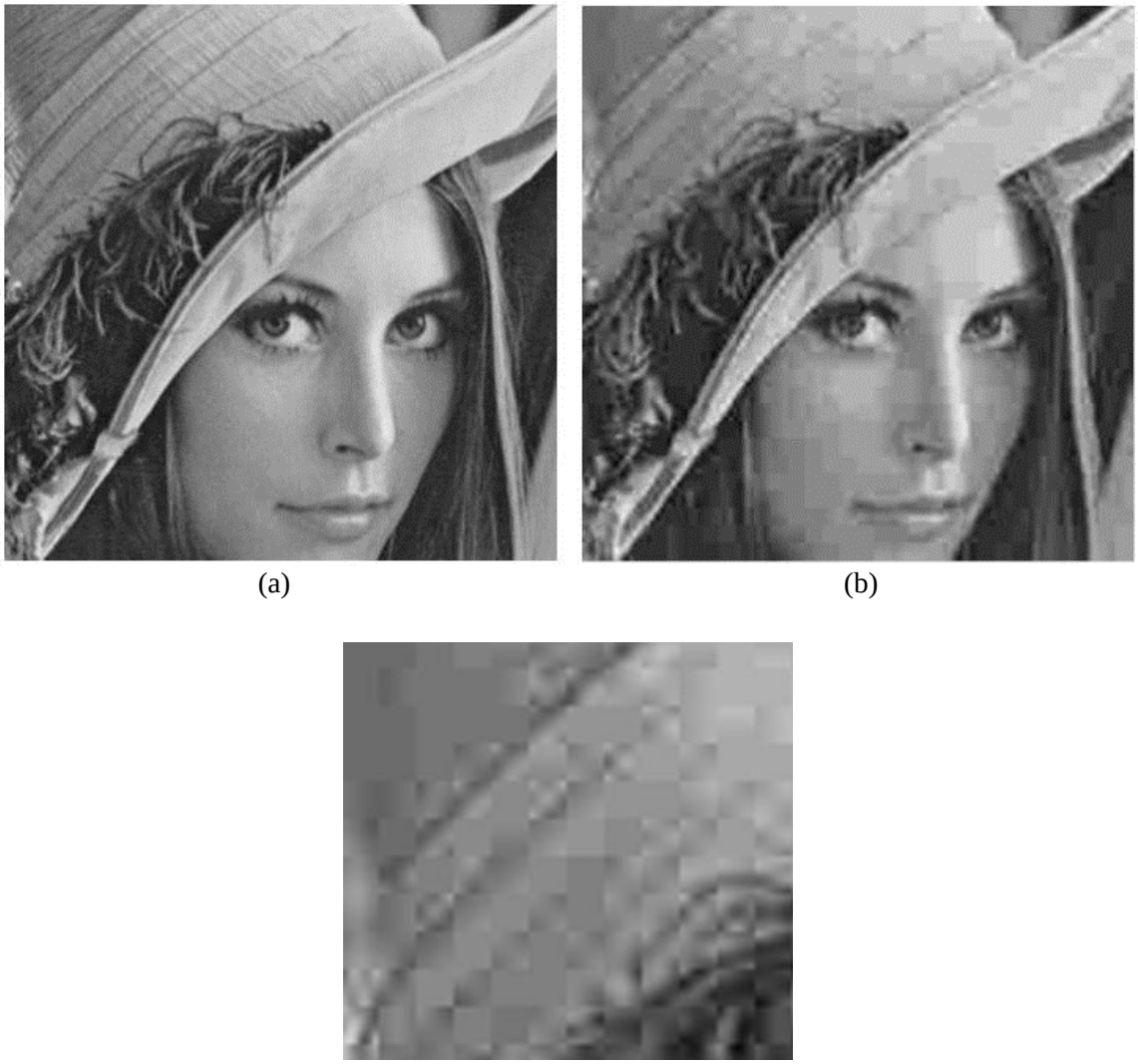
- Observation 1: A large majority of useful image contents change relatively slowly across images, i.e., it is unusual for intensity values to alter up and down several times in a small area, for example, within an 8 × 8 image block. Translate this into the frequency domain, it says that, generally, lower frequency components contain more information than the high frequency components which often correspond to less useful details and noises.

- Observation 2: Psychophysical experiments suggest that humans are more receptive to the loss of higher spatial frequency components than the loss of lower frequency components.

However, apart from all the above benefits of block based transform coding, there are some certain limitations at coarse quantization like blurring of image, graininess and most importantly blocking artifacts. Serious blocking artifacts are introduced at the block boundaries, since each block is independently encoded, often with a different encoding strategy and the extent of quantization. Of all the listed problems blocking artifact is the most serious and objectionable one at low bit rates.

1.4 BLOCKING ARTIFACTS

Transform coding, one of the most popular image compression techniques, generally divides the original image into sub-images called blocks. Each block is transformed and the selected large coefficients are quantized and then stored/transmitted.



(a) (b)

(c)

Figure 1.4 (a) original Image, (b) decompressed Image (affected by blocking artifacts) and (c) zoomed part of the image.

One of these techniques' drawbacks is that the discontinuities at the block boundaries are visible. Such a discontinuity implies the appearance of high frequency components. Here in Figure 1.4, two images are shown, one is Original image (a) & the other (b) is affected by Blocking Artifacts. Figure 1.4(c) shows the part of the image zoomed in, just to show the artifacts more clearly.

The basic cause behind blocking artifacts is block processing itself. As in Lossy compression, Image is divided into blocks to apply transform coding & the quantization is done. They are caused by accuracy loss of transform coefficients in the process of independent quantization of each block. The artifacts are particularly annoying when transform coefficients are subjected to a coarse quantization. Since the quantization takes place in the transform domain, the effect of the quantization error is spread over all of the spatial locations within the block. The artifacts are noticeable particularly in the plain, or slowly varying parts of an image. During quantization many of the transform coefficients are lost. They are visible due to spatially high-frequency components produced by the block discontinuity. This concept is further explained by Gibbs phenomenon, which is the root cause of artifacts.

1.4.1 GIBBS PHENOMENON AND BLOCK PROCESSING

The main cause of blocking artifacts is due to a signal being bandlimited (specifically, not having high frequencies) or passed through a low-pass filter; this is the frequency domain description. In terms of the time domain, the cause of this type of artifacts is the ripples in the sinc function, which is the impulse response (time domain representation) of a perfect low-pass filter. In mathematics, the Gibbs phenomenon is the peculiar manner in which the Fourier series of a piecewise continuously differentiable periodic function behaves at a jump discontinuity: the nth partial sum of the Fourier series has large oscillations near the jump, which might increase the maximum of the partial sum above that of the function itself. The overshoot does not die out as the frequency increases, but approaches a finite limit. The Gibbs phenomenon involves both the fact that Fourier sums overshoot at a jump discontinuity, and that this overshoot does not die out as the frequency increases. The picture here demonstrate the phenomenon for a square wave (of height $\pi/4$) whose Fourier expansion is $\sin x + \frac{1}{3}\sin 3x + \frac{1}{5}\sin 5x$.

More precisely, this is the function f which equals $\pi/4$ between $2n\pi$ and $(2n+1)\pi$ and $-\pi/4$ between $(2n+1)\pi$ and $(2n+2)\pi$ for every integer n.

Informally, it reflects the difficulty inherent in approximating a discontinuous function by a finite series of continuous sine and cosine waves. It is important to put emphasis on the word finite because even though every partial sum of the Fourier series overshoots the function it is approximating, the limit of the partial sums does not. The value of x where the maximum overshoot is achieved moves closer and closer to the discontinuity as the number of terms summed increases so, again informally, once the overshoot has passed by a particular x, convergence at the value of x is possible.

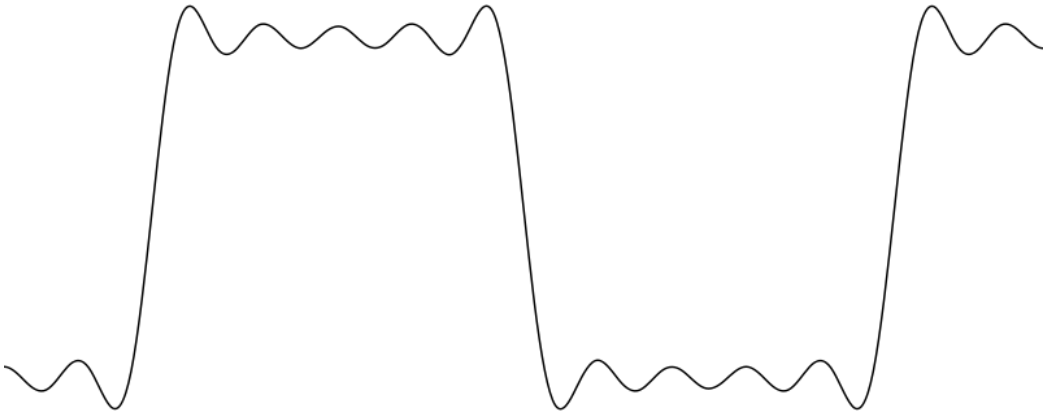


Figure 1.5 Functional approximation of square wave

From the above discussion, it can be concluded that the effect of Gibbs phenomenon is largely on the edges of a signal, which in the case of an image is at the block boundaries, creating blocking artifacts. The Transform coding techniques like DFT, DCT, DWT etc. are Fourier-related transform, and blocking artifacts occurs because of loss of high frequency components or loss of precision in high frequency components.

1.5 ORGANISATION OF DISSERTATION

Chapter 2 is the review on some major transforms including fractional transforms along with Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT). The mathematics of the two fractional transforms: Discrete Fractional Fourier Transform (DFrFT) and Discrete Fractional Cosine Transform (DFrCT) have been discussed in details as the latter is proposed as transform coding in this dissertation. Then another review has been provided on various blocking artifacts reduction schemes comprising methods like Overlapped block processing, Maximum a Posteriori (MAP) technique, Projection on Convex Set (POCS) based algorithms, Spatial filters deblocking methods,

Deblocking based on Human Visual System (HVS) etc. The literature has been deeply studied and gaps in study are mentioned at the end of literature. Finally, the chapter closes with the objectives of this dissertation.

Chapter 3 provides an in-depth discussion of the proposed method for the reduction of blocking artifacts based on overlapped block processing. The method is explained and interpreted using flow chart and other block diagrams.

Experimental results are thoroughly discussed in chapter 4. Evaluation of the results are made using various images and results are inferred using proper tables and figures.

To end with, Chapter 5 summarizes the main conclusions of this dissertation and points to future research directions.

LITERATURE SURVEY | CHAPTER 2

The literature survey summarizes, interprets, and evaluates existing "literature" (or published material) in order to establish current knowledge of a subject. The purpose for doing so relates to on-going research to develop that knowledge.

One goal of image compression is to represent an image with as few bits as possible. The reduction in bit rate is achieved by exploiting the redundancy and irrelevancy present in the image. A transform that decorrelates pixel intensities in this small spatial region (e.g. 8 × 8 blocks) and packs the energy in as few coefficients as possible would be ideally suited for this task, because in this case, a good approximation to the original data would be possible by coding only the few high energy coefficients. The transform idea is the heart of compression systems based on Transform Coding.

2.1 TRANSFORMS

Transform coding is the core part of several standards for image and video compression for example JPEG [15] and MPEG [25]. As already stated, the purpose of Transform coder is to transform the original spatial domain signal into some another domain signal. From compression point of view, the transformed domain should be of reduced dynamic range, where only a few coefficients contain bulk of the energy and thus efficient compression is possible. Discrete Cosine Transform (DCT) is widely used in practice. For most natural images, the DCT is also shown to be very effective at decorrelation and energy compaction. In particular, the DCT results in high energetic low spatial frequency components for most natural images since they possess significant low frequency content. Furthermore, fast algorithms for the computation of the DCT are available. In general transform based image compression codecs can be modeled by:

$$Y = T^{-1}QTX \tag{2.1}$$

where X represents the input image, T represents the forward Transformation, T^{-1} represents the inverse transformation, Q represents the quantisation and Y represents the compressed image.

2.1.1 DISCRETE FOURIER TRANSFORM

Fourier Transform (FT) characterizes the spectral behavior of signals by transforming them from the spatial domain into frequency domain globally. These transforms that

operate on a function over a finite domain, can be thought of as implicitly defining an extension of that function outside the domain. Any arbitrary function $f(x)$ can be written as a sum of sinusoids. One can evaluate that sum at any x, for all x where the original $f(x)$ was not specified [55]. Due to the global property of FT, it is quite inadequate in analysing the signals with non-stationary properties. It leads to the concept of introducing, a window function, which should slide in the whole spatial domain such that the transformation can be extracted locally. This in turn show the way to different versions like Discrete Fourier Transform (DFT), Discrete Time Fourier Transform (DTFT), Discrete Cosine Transform (DCT), and Discrete Sine transform (DST). Mathematically, for a sequence of length N, one dimensional forward Discrete Fourier transform kernel can be written as:

$$F = \frac{1}{\sqrt{N}} e^{-i2\pi nk/N} \; for \; k, n = 0,1,2 \dots \dots . . N - 1 \tag{2.2}$$

Inverse DFT kernel is nothing but the complex conjugate of F.

A Two dimensional data set $x(k, l)$ can be fourier transformed to $y(m, n)$ using 2-D forward Discrete Fourier transform kernel which can be defined as:

$$F = \frac{1}{N} e^{-i2\pi(kl+mn)/N} \; for \; k, l = 0,1,2 \dots \dots . . N - 1 \tag{2.3}$$

2.1.2 DISCRETE COSINE TRANSFORM

The discovery of DCT, in 1974, is an important achievement for the research community working on image compression. It expresses a function or a signal in terms of a sum of sinusoids with different frequencies and amplitudes. Its principle, properties and the way of computations are similar to that of DFT**.** DCT is equivalent to a DFT of roughly twice the length, operating on real data with even symmetry. It can be computed with a Fast Fourier Transform (FFT) in O ($N \log N$) operations [23].

Mathematically, DCT is defined by four different variants DCT-I to DCT-IV. The definitions of DCT kernel matrices have been well reviewed in [56]. We will quote them later for further discussion. Among these, DCT-II is probably the most commonly used form because of its property of better energy compaction than other forms. Hence, it is often simply referred to as "The DCT". So whenever the word DCT is used, it simply implies DCT-II.

2.1.2.1 DCT versus DFT: An Example

The obvious distinction between a DCT and a DFT is that the former uses only cosine functions, while the latter uses both cosines and sines, in the form of complex exponentials. However, DCT and DFT produce different boundary conditions. Unlike DFT, DCT provides a better approximation of a signal with fewer coefficients. DCT works on separating images into parts of differing. There is an inherent limitation of Fourier methods. Non-periodic signals and signals with discontinuities require many coefficients to represent accurately the shape of the signal. This limitation can be overcome by DCT.

DCT is chosen over DFT for compressing the jpeg images due to property of the former that it can approximate linear signals well with few coefficients. To understand this concept more clearly, let's consider a linearly increasing 8-point sequence and apply both the transforms separately.

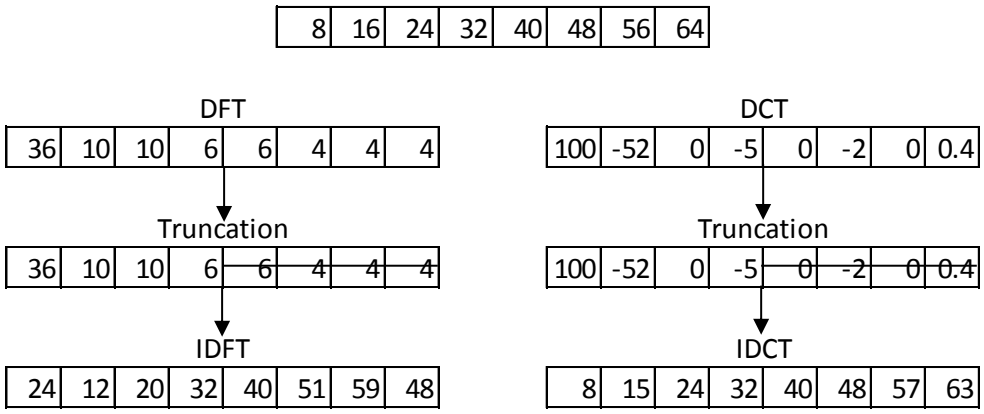


Figure 2.1 Comparison of DFT and DCT for an array [49]

Then truncate some of the data from the transformed sequence & apply inverse transform as shown in Figure above. It is shown here that even after removing some of the coefficients from DCT transformed sequence, the original data sequence can easily be recovered because most of the information of the data contains in a few coefficients only. However, in case of DFT, removing same number of coefficients produces a sequence way different from the original one.

2.1.2.2 One Dimensional Signal Compression

Let the one-dimensional signal to be transformed can be represented as column vector z and the corresponding DCT vector as h. The convolution of z with h produces DCT

coefficients. DCT vector elements can be computed using the kernel of the transform. One dimensional DCT is mathematically defined as:

$$y(k) = w(k) \sum_{n=0}^{N-1} x(n) \cos\left(\frac{\pi(2n+1)(k+1)}{2N}\right) \tag{2.4}$$

$for\ k = 0,1,2,3 \ldots \ldots N-1$

$$w(k) = \begin{cases} \frac{1}{\sqrt{N}}\ for\ k = 0 \\ \sqrt{\frac{2}{N}}\ for\ 1 \leq k \leq N-1 \end{cases}$$

where y is the DCT of x, and N is the length of x.

As an illustration, both DCT and FFT are applied in the encoder to a one dimensional luminance waveform made up of eight points linearly. Running the waveform through a transform, trimming off few high frequency components, and converting it back to an original with an inverse transformation contribute the output. DCT over FFT yields something closer to the original data sequence [2].

2.1.2.3 Two dimensional Signal Compression

The two-dimensional DCT is simply a separable extension of the DCT in one dimension. The DCT kernel of a two dimensional data set (matrix form) with N number of rows and columns is given as:

$$C_N = \sqrt{2/N}\left[k_m \cos(m(n+\frac{1}{2})\pi/N)\right] \tag{2.5}$$

$for\ m, n = 0,1, \ldots \ldots N-1$

k_m and k_n are defined as:

$$k_m = \begin{Bmatrix} \frac{1}{\sqrt{2}}\ for\ m = 0\ and\ m = N \\ 1 \quad otherwise \end{Bmatrix}$$

Assuming $N = 8$, the kernel matrix becomes:

$$C_8 = \sqrt{2/8}\left[k_m \cos(m(n+\frac{1}{2})\pi/8)\right] \tag{2.7}$$

Image is special case of 2-D signal where each data value is known as picture element or briefly to say pixel. DCT is most commonly used for the compression of general images since JPEG approves DCT as its standard [21] for transform coding. The following is the general overview of the jpeg image compression:

- The image is broken into 8×8 blocks of pixels.
- Working from left to right, top to bottom, the DCT is applied to each block. Each block is compressed through quantization.
- The array of compressed blocks that constitute the image is stored in a drastically reduced amount of space.
- When desired, the image is reconstructed through decompression, a process that uses the Inverse Discrete Cosine Transform (IDCT).

2.1.2.4 Zigzag Scanning

Zigzag scanning describes about the order of the DCT coefficients from low frequency to higher frequencies. The zigzag scanning is started at the upper-left corner and end up at the bottom-right corner as shown in Figure 2.2 The DCT coefficient at the upper-left corner is known as DC coefficient and rest of them are AC coefficients. The DC coefficient is a measure of the average value of the original 64 image samples, thus contains significant fraction of the total image energy. Zigzag scanning helps in the process of quantization and run length encoding of a transformed image.

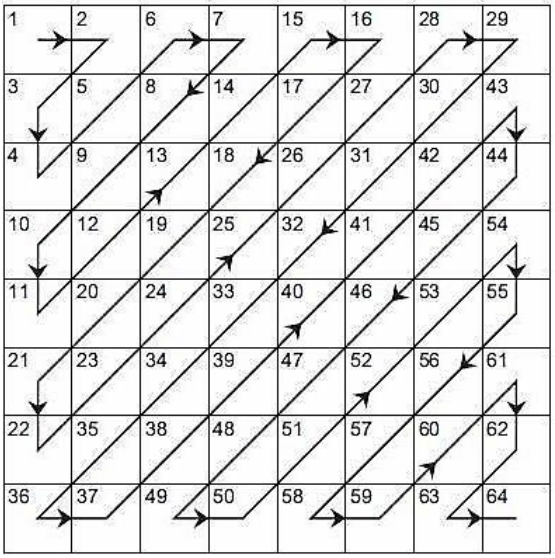


Figure 2.2 Zigzag scanning order for 8×8 block size [50]

2.1.2.5 Quantization

Quantization is the step where data is thrown away in actual. The DCT is a lossless procedure. The data can be precisely recovered through the IDCT (this isn't entirely true because in reality no physical implementation can compute with perfect accuracy). During Quantization every coefficients in the 8×8 DCT matrix is divided by a

corresponding quantization value. The quantized coefficient is defined in (2.6), and the reverse the process can be achieved by the (2.7)

$$F(u,v)_Q = round\left(\frac{F(u,v)}{Q(u,v)}\right) \tag{2.8}$$

$$F(u,v)_{deQ} = F(u,v)_Q \times Q(u,v) \tag{2.9}$$

The goal of quantization is to reduce most of the less important high frequency DCT coefficients to zero, the more zeros generation means the better the image will compress. The matrix Q generally has lower numbers in the upper left direction and large numbers in the lower right direction. Though the high-frequency components are removed, the IDCT still can obtain an approximate matrix which is close to the original 8×8 block matrix.

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{2.10}$$

The JPEG committee has recommended certain Q matrix (2.5) that work well and the performance is close to the optimal condition. However, a user can create its own kind of quantizer as required by the system keeping in mind about the zigzag scanning order which conveys that which coefficients are more important as compared to the others. After quantization, the image is encoded using run length coding keeping the coefficients in the zigzag scanning order.

With this, the process of image compression with DCT is over. Now, the approach will be towards the generalization of DCT, known as Discrete Fractional Cosine Transform (DFrCT), starting from the very basics of fractional transform.

2.1.3 FRACTIONAL TRANSFORMS

During the last two decades, the process of going from the whole of an entity to fractions of it underlies several interesting applications such as fractal objects, fuzzy logic and fractional signal processing [28]. The fourth power of 8 may be defined as $8^4 = 8 \times 8 \times 8 \times 8$, but it is not apparent to define $8^{3.5}$ in a similar way. It must have taken sometime before the common definition $8^{3.5} = 8^{7/2} = \sqrt{8^7}$emerged.

The first and second derivatives of the function $f(x)$ are commonly denoted by:

$$\frac{df(x)}{dx} \quad \text{and} \quad \frac{d^2 f(x)}{dx} = \frac{d}{dx}\left[\frac{df(x)}{dx}\right] = \frac{d[df(x)/dx]}{dx} = \left(\frac{d}{dx}\right)^2 f(x) \quad \text{respectively.}$$

And higher order derivatives are defined similarly. But it is not obvious that what will be 0.9^{th} order derivative using above definition. We know that by using differentiation property of Fourier Transform, the a^{th} derivative of $f(x)$ i.e. $\frac{d^a f(x)}{dx^a}$ is equivalent to the inverse Fourier transform of $(i2\pi\mu_a)^a F(\mu_a)$ where F (μ_a) is the Fourier Transform(FT) (of real order 'a') of the function $f(x)$. FT is widely used in many areas of science and engineering like optics, physics, acoustics, statistics, heat conduction and diffusion, electrical engineering, antenna and array processing etc. FT is a linear transform used to solve linear system problems. However, the FT is unable to solve certain classes of ordinary and partial differential equations of optics, signal processing and quantum mechanics [36]. Looking into the applicability of FT the concept of fraction was introduced in the FT in the year 1929 [33] and lead to the development of fractional Fourier transform (FrFT). If map the history of fractional thought is followed, it is found that in 17^{th} century, Bernoulli (1695) formulated a question about the meaning of a non-integer order derivative. This was the start of the fractional calculus which is the base of the continuous time fractional systems described by the fractional differential equations. Since then, the concept of fractional calculus has evolved in pure mathematics and developed by famous mathematicians [28]. In spite of the advancement in pure mathematics this concept had been applied in applied sciences only in 1920's. Furthermore, it is only in the last three decades that the applications of fractional calculus have emerged in engineering field which lead to a significant impact in several areas and attracted the scientific and technical community to the fractional objects [28].

The Fourier Transform (FT) is undoubtedly one of the most appreciated and frequently used tools in signal processing and analysis. Little need be said of the prominence and ubiquity of the ordinary Fourier transform in many areas of science and engineering. The FrFT, which is a generalization of the simple Fourier transform, was introduced 75 years ago but only in the last two decades, it has gained prominence in signal processing, optics and quantum mechanics. FrFT was introduced in 1980 by V. Namias [48] and it was established in the same year that the other transforms could also be fractionalized [47].

McBride and Keer explored the refinement and mathematical definition in 1987 [8]. In a very short span of time, FrFT has established itself as a commanding tool for the analysis of time varying signals. Mathematically the a^{th} order Fractional Fourier transform operator is the a^{th} power of the conventional Fourier transform operator. The fractional Fourier transform with parameter $a = 1$ corresponds to the orthodox Fourier transform. It is noticeable to bring in mind that the ordinary Fourier transform is a special case of a continuum of fractional Fourier domains. Essentially, the a^{th} order fractional Fourier transform is an intermediate between any function x(t) and its Fourier transform X(f). The 0^{th} order transform is merely the function itself, whereas the 1^{st} order transform is its Fourier transform. In all the time–frequency representations, one typically uses a plane with two orthogonal axes that usually are time and frequency. Consider a signal x(t) to be represented along the time axis and its ordinary Fourier transform X(f) to be represented along the frequency axis, then the Fourier transform operator (denoted by F) can be visualized as a change in representation of the signal corresponding to a counter clockwise rotation of the axis by an angle $\pi/2$ [47]. This is reliable with some of the pragmatic properties of the Fourier transform (FT). For example, two successive rotations of the signal through $\pi/2$ will outcome in an inversion of the time axis. Moreover, four consecutive rotations will leave the signal unchanged since a rotation through 2π of the signal should leave the signal same.

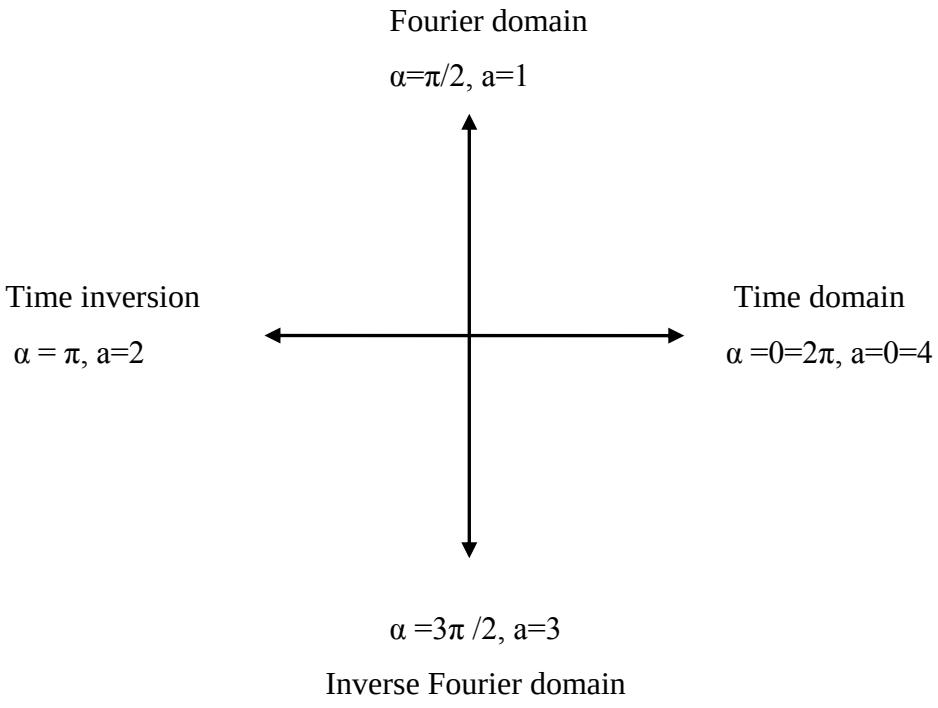


Figure 2.3 FRFT domain in time-frequency plane

The Fig 2.3 shows the FRFT domain in Time-frequency plain. The FRFT is a linear operator that corresponds to the rotation of the signal through an angle which is not a multiple of π/2, i.e. it is the representation of the signal along the axis making an angle α with the time axis.

Some important cases of the FRFT operator for $\alpha = a\frac{\pi}{2}$ are listed below:

- For a=0 or 4; i.e. $a = 0$ or 2π, one acquires the identify operator: $F^0 = F^4 = I$ (Identity Operator)
- For $\alpha = \frac{\pi}{2}$; i.e. a=1, one gets the Fourier operator: $F^1 = F$ (Fourier Operator)
- For $\alpha = \pi$; i.e. a=2 one achieves the reflection operator: $F^2 = FF = I -$ (Reflection Operator shown by time inversion).
- For $\alpha = \frac{3\pi}{2}$; i.e. a=3, one gets the inverse Fourier operator $F^3 = FF^2 = F^{-1}$ (Inverse Fourier Operator)

So for an angle from 0 to 2π, there are values of 'a' from 0 to 4 and it can be shown that the transform kernel is periodic with a period 4.

The FRFT of a signal $x(t)$ as can be computed by four steps process [30]:

1. Multiplying the function with a chirp,
2. Taking its Fourier transform,
3. Again multiplying with a chirp, and
4. Then multiplication with an amplitude factor.

It is found that the FRFT of a signal $x(t)$ exists under the same conditions in which its Fourier transform exists.

The FrFT have a number of applications in the areas of signal and image processing applications as signal detectors, correlation, pattern recognition, time variant filtering, multiplexing, image encryption, and signal and image recovery, restoration.

2.1.3.1 Discrete Fractional Fourier Transform

The continuous FRFT performs a rotation of signal in the time–frequency plane, and the conventional Fourier transform is a π/2 rotation of signal [30]. Similar to the continuous notation, the DFT can be regarded as a π/2 rotation for discrete signals [40]. The N-point DFT kernel is defined in the following way for energy preservation.

$$F_N = \sqrt{1/N}\begin{bmatrix} 1 & 1 & \dots & 1 & 1 \\ 1 & W_N^1 & \dots & W_N^{N-2} & W_N^{N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & W_N^{N-2} & \dots & W_N^{(N-2)^2} & W_N^{(N-1)(N-2)} \\ 1 & W_N^{N-1} & \dots & W_N^{(N-1)(N-2)} & W_N^{(N-1)^2} \end{bmatrix} \quad (2.11)$$

where $W_N = e^{-i(2\pi/N)}$. The DFrFT performs any angle rotation for discrete signals [56], [40]. The methods in [40] and [38] use the DFT Hermite eigenvectors to construct the DFrFT kernel matrix. In [40] and [38], the N-point DFrFT kernel is computed as

$$\mathbf{F}_{N,\alpha} = \mathbf{V}_N \mathbf{D}_N^{2\alpha/\pi} \mathbf{V}_N^T \quad (2.12)$$

where,
$$\mathbf{D}_N^{2\alpha/\pi} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & e^{-j\alpha} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & e^{-j(N-1)\alpha} \end{bmatrix} \quad (2.13)$$

and,
$$\mathbf{V}_N = [v_0|v_1|\dots|v_{N-1}] \quad (2.14)$$

v_k is the k^{th} order DFT Hermite eigenvector and α indicates the rotation angle of transform in the time–frequency plane. When $\alpha = 0$, $F_{N,\alpha}$ is an identity operator. If $\alpha = \pi/2$, the DFrFT becomes the conventional DFT. This concept is further used in derivation of Discrete cosine fractional transform.

2.1.3.2 Discrete Fractional Cosine Transform

DFrCT basically involves rotation of a discrete signal by an angle α in the time-frequency plane. But for analysis of 2-D signals such as images, a two dimensional version of DFrCT is required. For an $M \times N$ matrix, the 2-D DFrCT is computed in an unpretentious way. The 1-D DFrCT is applied to each row of given matrix and then same is applied to each column of the result matrix. Thus, the generalization of the DFrCT to 2-D is given by taking the DFrCT of the rows of the matrix i.e. image in a fractional domain and then taking the DFrCT of the subsequent column wise. In case of 2-D DFrCT, two angles of rotation α=π/2 and β= π/2 have to be taken. It has been recently observed that DFrCT can be used in the field of image processing .The vital feature of Discrete Fractional Cosine domain Image compression aids from its extra degree of freedom that is provided by its fractional orders. DCT-I is chosen & used in developing DFrCT [39]. The four types of DCT [56] are as follows:

DCT-I
$$C_{N+1}^I = \sqrt{2/N}[k_m k_n \cos(mn\pi/N)] \quad (2.15)$$
for m,n = 0,1,....N

DCT-II
$$C_N^{II} = \sqrt{2/N}\left[k_m \cos(m(n+\frac{1}{2})\pi/N)\right] \quad (2.16)$$
for m,n = 0,1,....N-1

DCT-III
$$C_N^{III} = \sqrt{2/N}\left[k_n \cos((m+\frac{1}{2})n\pi/N)\right] \quad (2.17)$$
for m,n = 0,1,....N-1

DCT-IV
$$C_N^{IV} = \sqrt{2/N}\left[k_m \cos((m+\frac{1}{2})(n+\frac{1}{2})\pi/N)\right] \quad (2.18)$$
for m,n = 0,1,....N-1

k_m and k_n are defined as:

$$k_m = \begin{Bmatrix} \frac{1}{\sqrt{2}} \; for\; m = 0 \; and \; m = N \\ 1 \quad otherwise \end{Bmatrix}$$

The DCT-I kernel has symmetric structure and is periodic with period 2. The periodicity means that repeated application of DCT-I would give the original sequence. DCT-IV is the same as DCT-I for symmetry and periodicity, but DCT-II and DCT-III operators are the forward and inverse transform pair of each other and are non-periodic.

$$C_N^I = \sqrt{\frac{2}{N-1}}\begin{bmatrix} \frac{1}{2} & \frac{1}{\sqrt{2}} & \dots & \frac{1}{\sqrt{2}} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & \cos\frac{\pi}{N-1} & \dots & \cos\frac{(N-2)\pi}{N-1} & \frac{1}{\sqrt{2}}\cos\frac{(N-1)\pi}{N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{\sqrt{2}} & \cos\frac{(N-2)\pi}{N-1} & \dots & \cos\frac{(N-2)^2\pi}{N-1} & \frac{1}{\sqrt{2}}\cos\frac{(N-2)(N-1)\pi}{N-1} \\ \frac{1}{2} & \frac{1}{\sqrt{2}}\cos\frac{(N-1)\pi}{N-1} & \dots & \frac{1}{\sqrt{2}}\cos\frac{(N-2)(N-1)\pi}{N-1} & \cos\frac{(N-1)^2\pi}{N-1} \end{bmatrix} \quad (2.19)$$

To develop the DFrCT, the eigenvalues and eigenvectors of DCT-I kernel matrix are needed to be well studied [10], [13] and [24]. In [39], S.C Pei made five propositions which leads to the development of DFrCT.

1. The DFT kernel matrix has only four distinct eigenvalues: $\{1, -1, j, -j\}$ and its multiplicities are summarized in Table 2.1. Because the DFT has only four distinct eigenvalues, the DFT eigenvectors will constitute four eigenspaces. It is trivial to find that any vector spanned by the DFT eigenvectors corresponding to the same eigenvalue is still a DFT eigenvector. Therefore, there exist infinite eigenvectors for the DFT kernel matrix. The multiplicities of DFT eigenvalues are just the dimensions of eigenspaces.

Table 2.1 Eigenvalue multiplicities of the DFT kernel matrices [39]

N	Multiplicity of 1	Multiplicity of $-j$	Multiplicity of -1	Multiplicity of j
$4m$	m+1	m	m	m-1
$4m$+1	m+1	m	m	m
$4m$+2	m+1	m	m+1	m
$4m$+3	m+1	m+1	m+1	m

2. All the DFT eigenvectors are even or odd. The even eigenvectors are with the eigenvalues 1 or -1; in addition, the odd eigenvectors correspond to the eigenvalues j or $-j$. This proposition is very important for the development of DFrCT. In the following discussion, the eigenvalues and eigenvectors for the DCT are explained and the relationships with the conventional DFT is established.
3. The DCT-I eigenvectors can be attained from the DFT eigenvectors. If $\mathbf{v} = [v_0, v_1, \cdots\cdots, v_{N-2}, v_{N-1}, v_{N-2}, \cdots\cdots, v_1]^{\boldsymbol{T}}$is an even eigenvector of the $(2N-2)$ point DFT kernel matrix $\mathbf{F_{2N-2}v} = \lambda\mathbf{v}$ **,** where $(\lambda = 1, -1)$ [39]. Then

$$\hat{\mathbf{v}} = \left[v_0, \sqrt{2}v_1, \cdots\cdots, \sqrt{2}v_{N-2}, v_{N-1}\right]^{\boldsymbol{T}} \tag{2.20}$$

will be an eigenvector of the N-point DCT-I kernel matrix, where λ is the corresponding eigenvalue.

$$\mathbf{C_N^I}\hat{\mathbf{v}} = \lambda\hat{\mathbf{v}} \tag{2.21}$$

4. The eigenvalues of DCT-I kernel matrices are only 1 and -1. Their multiplicities are shown in Table 2.2

Table 2.2 Eigenvalue multiplicities of the DCT-I kernel matrices [39]

N	Multiplicity of 1	Multiplicity of -1
Even	$N/2$	$N/2$
Odd	$(N+1)/2$	$(N-1)/2$

Regardless of N is even or odd, the sum of multiplicity of 1 and multiplicity of -1 is equal to N. Thus, the DCT-I eigenvectors obtained from the DFT even eigenvectors can result in a full-rank DCT-I kernel matrix. All the DCT-I eigenvectors can be obtained from the DFT even eigenvectors.

5. The orthogonality in DCT-I and DST-I eigenvectors can be inherited from that in DFT eigenvectors. If $\mathbf{v}_m$ and $\mathbf{v}_n$ $(m \neq n)$ are both the even and orthogonal DFT eigenvectors, the DCT-I eigenvectors $\hat{\mathbf{v}}_{\boldsymbol{m}}$ and $\hat{\mathbf{v}}_{\boldsymbol{n}}$ will also be orthogonal. Thus,

$$\hat{\mathbf{v}}_m^T \hat{\mathbf{v}}_n = \mathbf{v}_m^T \mathbf{v}_n = 0 \tag{2.22}$$

This tells us that the DFT orthogonal eigenvectors can be used to generate the DCT-I orthogonal eigenvectors.

From the previous discussions, it can be said that all the DFT and DCT transform kernels have infinite eigenvectors. Because the DFrFT defined with these DFT Hermite eigenvectors can have similar output as continuous FrFT and have the properties of unitarity, additivity, and reversibility, the DFT Hermite eigenvectors can be used in developing DFrCT as reasonable choices. The eigenvector $\hat{\mathbf{v}}_k$ will have the eigenvalue $e^{-jk\alpha}$ (k is even) for the DFrCT kernel matrix. Such an assignment rule will result in a DCT kernel for $\alpha = \pi/2$. Similar to the DFrFT, the N-point DFrCT kernel [39] can be defined as

$$\mathbf{C}_{N,\alpha} = \hat{\mathbf{V}}_N \, \hat{\mathbf{D}}_N^{2\alpha/\pi} \, \hat{\mathbf{V}}_N^T \tag{2.23}$$

where,
$$\hat{\boldsymbol{D}}_N^{2\alpha/\pi} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & e^{-j2\alpha} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & e^{-j2(N-1)\alpha} \end{bmatrix} \tag{2.24}$$

and,
$$\hat{\mathbf{V}}_N = [\hat{\mathbf{v}}_0 | \hat{\mathbf{v}}_1 | \dots | \hat{\mathbf{v}}_{N-1}] \tag{2.25}$$

$\hat{\mathbf{v}}_k$ is the DCT-I eigenvector obtained from the k^{th}-order DFT Hermite eigenvector. While, $\alpha = \pi/2$ the DFrCT will become the conventional DCT-I. When $\alpha = 0$, $\mathbf{C}_{N,\alpha}$ is an identity matrix. The steps for constructing the N-point DFrCT kernel with angular parameter α are summarized as follows [39].

- Step-1: Compute the M-point DFT Hermite even eigenvectors. $M = 2(N-1)$
- Step-2: Use (2.18) to compute the DCT-I eigenvectors from the DFT Hermite even eigenvectors.
- Step-3: Determine the DFrCT transform kernel from (2.21).

2.2 PERFORMANCE EVALUATION PARAMETERS

The Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are the two error metrics most commonly used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower is the error. Mathematically, PSNR is defined in terms of Mean square error (MSE) as follows [35]:

$$\mathbf{PSNR} = \mathbf{10}\log_{\mathbf{10}}\left(\frac{\boldsymbol{R}^{\mathbf{2}}}{\mathbf{MSE}}\right) \tag{2.26}$$

where R is the maximum fluctuation in the input image data type and MSE is defined as:

$$\mathbf{MSE} = \frac{\mathbf{1}}{\boldsymbol{MN}}\sum_{\boldsymbol{M,N}}\left[\boldsymbol{I}(\boldsymbol{m},\boldsymbol{n}) - \hat{\boldsymbol{I}}(\boldsymbol{m},\boldsymbol{n})\right]^{\mathbf{2}} \tag{2.27}$$

where M and N are number of rows and columns in the input images. I and $\hat{I}$ are the original image and the distorted image respectively.

It is well known that the PSNR is not always a good measure to reflect the subjective image quality, even though it is one of the most popular criteria employed in image processing. Neither image content information nor HVS characteristics were taken into account by PSNR and MSE when they are used to assess image quality. Consequently PSNR and MSE can't achieve good results when compared to subjective quality scores. So none of these are considered to be a good parameter for evaluating the quality of images, because PSNR and MSE may have the same value for two totally different quality images, which is well explained by Zhou Wang and Alan C. Bovik in [57]. So, human visual system based peak signal to noise ratio (PSNR-HVS-M) [32] is chosen to measure the quality of images.

2.2.1 HVS BASED PSNR

PSNR-HVS-M is a model which has been designed to improve the performance of PSNR and MSE. The PSNR-HVS-M divides the image into 8×8 non-overlapping blocks. Then the $\delta(i,j)$ difference between the original and the distorted blocks is weighted for every 8×8 block by the coefficients of the Contrast Sensitivity Function (CSF) and contrast masking metric (CM) [1,26].

$$\delta_{hvsm}(i,j) = \delta(i,j).CM(i,j).CSF(i,j) \tag{2.28}$$

Here, $\delta(i,j)$ is calculated using DCT coefficients.

$$CSF = \begin{bmatrix} 1.608 & 2.340 & 2.574 & 1.608 & 1.072 & 0.643 & 0.505 & 0.422 \\ 2.145 & 2.145 & 1.838 & 1.354 & 0.990 & 0.444 & 0.429 & 0.468 \\ 1.838 & 1.980 & 1.608 & 1.072 & 0.643 & 0.451 & 0.373 & 0.460 \\ 1.838 & 1.514 & 1.170 & 0.887 & 0.505 & 0.296 & 0.322 & 0.415 \\ 1.430 & 1.170 & 0.696 & 0.460 & 0.378 & 0.236 & 0.250 & 0.334 \\ 1.072 & 0.735 & 0.468 & 0.402 & 0.318 & 0.247 & 0.228 & 0.280 \\ 0.525 & 0.402 & 0.330 & 0.296 & 0.250 & 0.213 & 0.214 & 0.255 \\ 0.357 & 0.280 & 0.271 & 0.263 & 0.230 & 0.257 & 0.250 & 0.260 \end{bmatrix} \tag{2.29}$$

$$CM = \begin{bmatrix} 0.391 & 0.826 & 1.000 & 0.391 & 0.174 & 0.063 & 0.038 & 0.027 \\ 0.694 & 0.694 & 0.510 & 0.277 & 0.148 & 0.030 & 0.028 & 0.033 \\ 0.510 & 0.592 & 0.391 & 0.174 & 0.063 & 0.031 & 0.021 & 0.032 \\ 0.510 & 0.346 & 0.207 & 0.119 & 0.038 & 0.013 & 0.016 & 0.026 \\ 0.309 & 0.207 & 0.073 & 0.032 & 0.022 & 0.008 & 0.009 & 0.017 \\ 0.174 & 0.082 & 0.033 & 0.024 & 0.015 & 0.009 & 0.008 & 0.012 \\ 0.042 & 0.024 & 0.016 & 0.013 & 0.009 & 0.007 & 0.007 & 0.010 \\ 0.019 & 0.012 & 0.011 & 0.010 & 0.008 & 0.010 & 0.009 & 0.010 \end{bmatrix} \tag{2.30}$$

$$\mathrm{MSE}_{hvsm}(i,j,I,J) = \frac{1}{MN} \sum_{I=1}^{M/m} \sum_{J=1}^{N/n} \left[\sum_{i=1}^{m} \sum_{j=1}^{n} ((\delta_{hvsm}(i,j))^2) \right] \tag{2.31}$$

where *(I,J)* is the position of an $m \times n$ block in the image and *(i,j)* is the position of a pixel in the $m \times n$ block, however in this case $m = n$. Then PSNR-HVS-M can be computed by replacing the MSE in equation (5.2) with MSE_{hvsm}.

$$\mathrm{PSNR}_{hvsm} = 10 \log_{10} \left(\frac{R^2}{\mathrm{MSE}_{hvsm}} \right) \tag{2.32}$$

where where *R* is the maximum fluctuation in the input image data type.

2.3 BLOCKING ARTIFACTS REDUCTION

JPEG introduces blocking artifacts at medium and low bit rates because of its short and non-overlapping basis functions. Moreover, the cancellation of cosine coefficients in the vicinity of sharp edges generates Gibbs oscillations, which are emphasized by the image structure. Compression effects of the transformed images are categorized like:

- Stair case noise along the image edges
- Grid noise in the monotone areas
- Corner outliers in the corner points of the 8×8 DCT blocks
- False contours visible in the area of slowly varying intensity
- Disturbing blocking effect removed from textured area

As block-based coders became increasingly more popular, the need to diminish the effects of blocking artifacts grew stronger. This in turn drives new ideas and implementations in this field. A great deal of effort has been invested in attempts to reduce these artifacts, while preserving the information content of the image. Subjective picture quality can be significantly improved by decreasing the blocking artifacts.

Over the past several years, many techniques have been applied to reduce the blocking artifacts in block DCT-coded images. Two approaches are generally adopted. In the first

approach, the reduction of blocking artifacts is carried out at the encoding side [18,19]. In the second approach, the reconstructed image is post-processed aimed at improving its visual quality without any modification in the encoding or decoding mechanisms. Post-processing of the decoded image may be carried out in spatial domain or in frequency domain.

Many algorithms are proposed for reduction of blocking artifacts for block based transform coding schemes. Among them are:

- Overlapped block processing
- Maximum a Posteriori (MAP) technique
- Projection on Convex Set (POCS) based algorithms
- Lapped orthogonal transform (LOT) based deblocking
- Wavelet Based Deblocking Algorithms
- Spatial filters deblocking
- Deblocking based on Human Visual System (HVS)

Apart from these methods there are some other techniques also which are discussed under the heading of 'Other Techniques'.

2.3.1 OVERLAPPED BLOCK PROCESSING

An early work on artifact suppression was done by H. C. Reeves *et al.* [16] using the technique of overlapped block processing, wherein the image is sub-divided into overlapping blocks instead of mutually exclusive blocks, which is typically the case with ordinary block based techniques. During reconstruction, the pixels in the overlapping regions are reconstructed as the average of decoded values. For explanation purpose, the author considered an image of size 5×5, and segmented it into blocks of 3×3 and used the concept of overlapping as shown in Figure

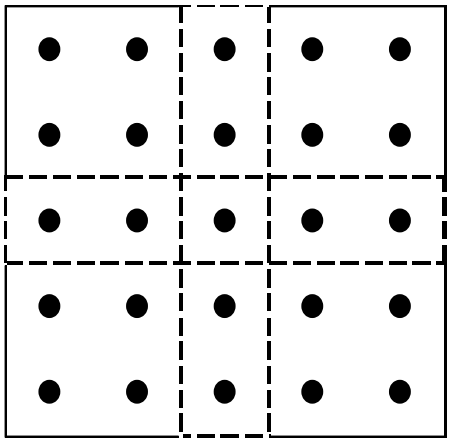


Figure 2.4 5×5 image segmented into blocks of 3×3

This method alleviates blocking artifacts without degrading the image, but causes an increase in computational complexity. The author used this method by segmenting the image with block size of 16×16. The performance of the method was evaluated in terms of Normalised Mean Square Error (NMSE) defined as:

$$\text{NMSE} = \frac{\sum_{M,N}\left[I(m,n) - \hat{I}(m,n)\right]^2}{\sum_{M,N}[I(m,n)]^2} \tag{2.33}$$

2.3.2 MAXIMUM A POSTERIORI (MAP) TECHNIQUE

MAP based technique is based on stochastic model of image data [46]. It selects the best image from a set of better images. T. P. O'Rourke *et al*. proposed a method in which quantization step partitions the transform coefficients and maps all points in a partition cell to a reconstruction point, taken as centroid of cell. The technique selects the reconstruction point within quantization partition cell which results in reconstructed image that best fits a non-Gaussian Markov random field (MRF) image model. A Gibbs distribution is used to explicitly write the distribution of MRF's. The gradient projection method is used to update the estimate based on image model iteratively. A decompression algorithm is then outlined based on a previously proposed image model. From experimental results, reconstructed image sequence shows a reduction in many of the most noticeable artifacts.

2.3.3 PROJECTION ON CONVEX SET (POCS) BASED ALGORITHMS

An iterative block reduction technique based on the theory of projection onto convex sets is proposed by A. Zakhor [7]. A number of constraints on coded image are used for restoration into original form. For example, one constraint can be devised from the information that blocking artifact image has high frequency components across boundary of neighboring blocks. These high frequency components are omitted from original image, so projection of artifact image onto original image is performed by iterative procedure. These iterations are repeated until artifact free image is obtained.

In POCS based algorithm proposed by Y. Yang *et al*. [52], based on line processes modelling of the image edge structure, a new family of directional smoothness constraint sets is described. Because of the fact that visibility of artifacts in an image is spatially varying, the authors have also taken definition of smoothness sets. The numerical difficulty of computing the projections onto these sets is overcome by a divide-and-

conquer (DAC) strategy. The algorithm can remove blocking artifacts from compressed image and video.

The highly correlated images are assumed by H. Paek *et al*. [17] to reduce blocking artifacts based on POCS. As assumed images are highly correlated, the global frequency characteristics in two adjacent blocks are similar to the local ones in each block. The high frequency components in global characteristics of a decoded image, which are not found in local ones, results from blocking artifacts are considered. N-point DCT to obtain the local characteristics, and 2N-point DCT to obtain the global ones, and then relation between N-point and 2N-point DCT coefficients are employed. The undesired high frequency components caused by blocking artifacts are detected by comparison of N-point with 2N-point DCT coefficients. Then novel convex sets and their projection operators in the DCT domain are proposed by authors and they claim that it yields significantly better performance than the conventional techniques in terms of objective quality, subjective quality, and convergence behaviour.

2.3.4 LAPPED ORTHOGONAL TRANSFORM (LOT) BASED DEBLOCKING

Lapped orthogonal transform (LOT) can reduce blocking artifacts to very low levels. It is tool with basis functions that overlap adjacent blocks. H. S. Malvar *et al.* [19] proposed an optimal LOT that is related to the DCT in such a way that a fast algorithm for a nearly optimal LOT is derived. The LOT is distinguished by the fact that each block of size N is mapped into a set of N basis functions, each one being longer than N samples. As coding noise is mainly a function of quantization process, therefore, it is virtually unaffected by LOT. However, it requires about 20-30 percent more computations, mostly additions in comparison with DCT [19]. In another research by H. S. Malvar [18], the lapped bi-orthogonal transform (LBT) and hierarchical lapped bi-orthogonal transform (HLBT) are used for image coding. The HLBT has a significantly lower computational complexity than the lapped orthogonal transform (LOT), with almost no blocking artifacts in comparison with DCT. Experimental results performed by author show better performance of the LBT and HLBT and they have fewer ringing artifacts.

2.3.5 WAVELET BASED DEBLOCKING ALGORITHMS

Wavelet based deblocking algorithm [43] computes soft threshold values based on difference between wavelet transform coefficients of image blocks and coefficients of

entire image to threshold high-frequency wavelet coefficients in different sub-bands using different values and strategies. An adaptive threshold value is employed for different images and characteristics of blocking effects. The filtered image is obtained by thresholding of different sub bands by three-level decomposition. A.W.-C. Liew *et al.* [5] proposed a non-iterative wavelet-based deblocking algorithm. The algorithm exploits the fact that block discontinuities are constrained by the dc quantization interval of the quantization table, as well as the behavior of wavelet modulus maxima evolution across wavelet scales to derive appropriate threshold maps at different wavelet scales. The algorithm can suppress blocking artifacts as well as ringing artifacts effectively while preserving true edges and textural information.

2.3.6 SPATIAL FILTERS DEBLOCKING

T. Meier *et al.* [45] proposed an algorithm in which the degraded image is segmented and each region obtained by the segmentation process is then low-pass filtered separately. This prevents blurring of edges because the low-pass filter is never applied across region boundaries. The quality of the enhanced images depends on the segmentation algorithm and the filter used and further improvements are possible. It is likely that there are better filters, for example adaptive filters that change their mask close to region boundaries.

An algorithm based on non-linear smoothing of pixels for deblocking is proposed by J. Chou *et al*. [22]. The deblocking is performed in two steps. In step 1, difference between actual image edges and artificial discontinuities produced by quantization noise at block boundaries is taken into account. A probabilistic framework is used to derive estimates for the reconstructed DCT coefficients and for the quantization error of each image coefficient. While removal of blockiness by reducing discontinuities at block edges is done in step 2. The principal used is to reduce discontinuities of artificial edges at block boundaries to a level that is imperceptible to the eye. First, the discontinuities are computed by differencing the pixels across each block boundary and then, authors attempted to reduce these discontinuities below visibility threshold. Experimental results show significant improvement in visual quality of images.

A deblocking algorithm by adaptively using spatial frequency and temporal information extracted from the compressed data is proposed by H. W. Park *et al.* [20]. The authors investigated the distribution of the inverse quantized coefficients and the motion vectors for extraction of semaphores of the blocking artifacts and ringing noise in each 8×8 block.

For reduction of blocking artifacts, a 1-D low pass filter (LPF) and 2-D signal adaptive filter are applied adaptively to every 8×8 block by using blocking and ringing semaphores. Computer simulations performed by authors on several images show that proposed method's better performance over MPEG-4 VM (verification model). An adaptive algorithm based on the filtering of block boundaries for reduction of blocking artifacts in compressed images and video is proposed by N. I. Cho *et al.* [31]. The authors consider a scan line of an image with blocking artifacts as continuous-time step function and output also as the continuous-time step response of the low pass filter. The continuous-time response can be found out with given boundary difference and duration of blocky region without computation. Then, deblocking is just sampling of the output at the pixel locations. As a consequence, an appropriate filtered output is obtained without actual filtering. Simulations carried out by authors show that objective and subjective quality of proposed method is comparable to other conventional algorithms and POCS.

2.3.7 DEBLOCKING BASED ON HUMAN VISUAL SYSTEM (HVS)

B. Macq *et al.* [9] proposed a criterion based on visual model for reduction of blocking artifacts. The target is to decompose the corrupted image into perceptual channels and to cancel the channels where the noise is above the visibility threshold. Then image is reconstructed with the only channels where the estimated noise is below the visibility threshold. More specifically, the noisy picture is first split up into several perceptual channels by means of filters tuned to specific spatial frequencies and orientations. Each resulting filtered picture is then weighted by a masking function in order to cancel the visible noise. The masking is a function of the perceptual component contrast of the original picture. Difference of noisy picture and noise estimation is used for carrying out the contrast. The addition of each masked pictures provides at last the restored picture.

T. Chen *et al.* [44] proposed an approach that works in transform domain for reduction of quantization noise. The adaptive weighting mechanism is integrated by considering the masking effect of human visual system. The proposed approach makes use of transform coefficients of shifted blocks, rather than those of the neighboring blocks, in order to obtain a close correlation between the DCT coefficients at the same frequency. The filtering is operated location-variantly based on the local activity of blocks to achieve the artifacts reduction and detail preservation simultaneously. More exactly, an adaptively weighted low-pass filtering technique is activated to image blocks of different activities,

which represent the inherent masking abilities for artifacts. Human visual system sensitivity at different frequencies is used to characterize the block activity. Blocking artifacts are more noticeable for low-activity blocks and post-filtering of the transform coefficients is applied within a large neighbourhood to smooth out the artifacts. For high activity blocks, a small window and a large central weight are used to preserve the image details since the eye has difficulty discerning small intensity variations in portions of an image where strong edges and other abrupt intensity changes occur. Finally, the quantization constraint is also applied to the filtered DCT coefficients prior to the reconstruction of the image from coefficients.

S. Singh *et al.* [42] put forth a method and an algorithm, working in frequency domain, for the detection and reduction of such blocking artifacts. These artifacts are modelled here as 2-D step functions between two neighboring blocks. Presence of the blocking artifacts is detected by using block activity based on human visual system (HVS) and block statistics. The boundary regions between blocks are identified as either smooth or non-smooth regions. The blocking artifacts in smooth regions are removed by modifying a few DCT coefficients appropriately, whilst an edge-preserving smoothing filter is applied to the non-smooth regions, i.e., genuine edges. The reduction in the blocking artifacts for each image have been evaluated using three indices, namely peak signal-to-noise ratio (PSNR), mean structure similarity (MSSIM) index based on human visual perception, and a new index, called here block boundary measure (BBM).

J. Singh *et al.* [27] proposed a new adaptive post-filtering algorithm to remove coding artifacts in block-based DCT compressed images. Presence of blocking artifacts is detected using block activity based on human visual system and block statistics. The boundary regions between the blocks are identified as smooth, non-smooth or intermediate regions. The blocking artifact in the smooth and non-smooth regions are removed by modifying a few DCT coefficients while an edge preserving smoothing filter is applied to the intermediate region. Experimental results illustrating the performance of proposed method on the basis of PSNR, Mean SSIM, SF (Similarity Factor), and BBM indices are presented and evaluated.

A. Chetouani *et al.* [3] proposed a method of deblocking which aims to reduce the blocking artifacts in the compressed image by analyzing their visibility. A perceptual map is obtained using some Human Visual System (HVS) characteristics. This perceptual map is used as input to a recursive filter to reduce the blocking effect.

2.3.8 OTHER TECHNIQUES

For smooth regions, Y. Luo *et al.* [51] proposed a method which takes advantage of the fact that the original pixel levels in the same block provide continuity and this property and the correlation between the neighboring blocks is utilized to reduce the discontinuity of the pixels across the boundaries.

Y. Yang *et al.* [53] describes reconstruction of images from incomplete block discrete cosine transform (BDCT) data. In it, prior knowledge about the smoothness of the original image is transmitted along with the image data. The decoder reconstructs the image by using both of them. Two methods are proposed in this paper based on POCS and CLS respectively. In Constrained Least Square (CLS), the proposed objective function captures the smoothness properties of original image. The recovered image is obtained by minimizing an objective function, which is the weighted sum of two functions that impose conflicting requirements on the recovered image. Thus, if one of these functions penalizes deviation from the available data the other must penalize the undesired effects if an image is reconstructed only from the available data. In this sense, the second function introduces prior knowledge that complements the available data or, in other words, constrains the behavior of the reconstructed image. Iterative algorithms are introduced for its minimization. The authors claim with the help of experimental results that blocking artifacts can be reduced drastically.

In another paper based on adaptive constrained least squares restoration by A. Kaup [4], a numerically simple post-processing scheme is proposed. The spatial adaptation of post processing to local image structure preserves high frequency details of image. The authors claim that proposed technique almost completely removes blocking artifacts.

In Combined Transform Coding (CTC) scheme, as adopted by Y. Zhang *et al.* [54], image is divided into two sets that contain different correlation properties, i.e., the upper image set (UIS) and lower image set (LIS). The UIS contains the most significant information and tends to be highly correlated whereas; LIS contains the less significant information and carries less correlation. Then the UIS is compressed noiselessly without dividing into blocks and LIS is coded by conventional block transform coding. This results in suppression of blocking effects in image due to the fact that correlation in UIS is reduced without distortion and thus as a result the inter-block correlation is significantly reduced .The additional advantage of the CTC scheme is removal of ringing effects.

The research by C. Kim [11] proposes an AC prediction based blocking artifact reduction method. For each block, its DC value and DC values of the surrounding eight neighbor blocks are exploited to predict low frequency AC coefficients. Each block is categorized into low activity or high activity block by use of these predicted AC coefficients. Then two types of low pass filters are adaptively applied based on the categorized result of each block. A strong low pass filter is applied in low activity region, where blocking artifacts are most noticeable. High activity regions are filtered by weak low pass filter. Computer simulations performed by author show that proposed algorithm is effective in reducing blocking artifacts as well as ringing artifacts. Hadamard transform is used by K. Veeraswamy *et al.* [29] for AC coefficients prediction to reduce the blocking artifacts. In proposed method, Hadamard transform DC values are transmitted. AC restoration method is used for image reconstruction. The proposed method improves the peak signal to noise ratio and reduces the blocking effects significantly.

Multilayer Perceptron (MLP) neural network deblocking is based on adaptive learning by examples concept. In this scheme [14], relevant information from the image is extracted and given as input to neural network. The MLP neural network tries to learn to reconstruct the original image. On the encoder side, the image is compressed and decompressed by image compression algorithms. By the decompressed image, features representing the occurrence of blocking effects, the numerical artifacts indicators (NAIs), are taken out and as an input given to the MLP network. The MLP will try to produce an output approximating the difference between the original image and the decompressed image. To train the MLP network, a suitable supervised learning algorithm and difference between the original and the decompressed image as a desired output is used. After the completion of training, the weights of the MLP network together with the compressed image data are transmitted or stored. When compressed data is received at the decoder, decompression and extraction of blocking effect features is done and given as input to MLP network. The output of MLP network is added in the decompressed image for final decoded image formation.

In deblocking filtering using sum of symmetrically aligned pixels [6], a new class of deblocking algorithms for reduction of blocking artifacts in images and video is proposed. A symmetrically aligned weighted sum of pixel quartets with respect to block boundaries is employed for image deblocking. The basic weights are obtained from a function which obeys predefined constraints. A deblocked image is produced using these weights which contain blurred edges near real edges. The authors refer these blurred edges as ghosting

phenomenon. To prevent this, non-monotone area weights of pixels is modified by dividing each pixel's weight by predefined factor called a grade. This scheme is referred as weight adaptation by grading (WABG). Better deblocking of monotone areas is done by doing three iterations of WABG. The fourth iteration is done on rest of image to deblock the detailed blocks. The authors call this as deblocking frames of variable size i.e., DFOVS. The WABG and the DFOVS approaches automatically adapt themselves to different bit rates. It produces very good results for decompressed images ranging from extremely low to medium bit rates as claimed by authors.

The gradient projection based method [41] exploits the correlation between the intensity values of boundary pixels of two neighboring blocks. It is based on the theoretical and empirical observation that under mild assumptions, quantization of the DCT coefficients of two neighboring blocks increases the expected value of the Mean Squared Difference of slope (MSDS) between the slope across two adjacent blocks, and the average between the boundary slopes of each of the two blocks. This increase in expected value of MSDS is dependent on the width of quantization intervals of transform coefficients. Consequently, amongst all permitted inverse quantized coefficients, the set which reduces the expected value of this MSDS by a suitable amount is most likely to decrease the blocking artifacts. In order to estimate the set of unquantized coefficients, a constrained quadratic programming problem in which the quantization decision intervals provide upper and lower bound constraints on the coefficients is solved. The authors with the help of simulations claim that from a subjective viewpoint, the blocking effect is less noticeable in processed images than in the ones using existing filtering techniques.

A new metric is first given to evaluate the blocking artifacts; and then non-local means filter is applied to remove quantization noise on the blocks by C. Wang *et al.* [12]. During the process, nonlocal means filters with different variances are used to do deblocking, and their efficiencies are recorded as the references. The deblocked image is finally the one combined with all blocks filtered with the optimal parameters.

2.4 GAPS IN STUDY

As per the Literature review, various techniques have been studied and following gaps are found that are needed to be worked on:

- In overlapped block processing method, used to produce blocking artifacts free image, segmentation of the image can be done with using some other block sizes.

Also inside block size may be varied while keeping a check on computational complexity.

- Instead of dividing the image of $N \times N$ into blocks $n \times n$, the image may be divided into $m \times n$. This might produce some better results. Most commonly in almost every method of deblocking 8×8 block size is chosen to work on. Though this may be the best size for DCT based compression producing good results, as recommended by JPEG committee, yet any novel idea of working on some other block size cannot be denied.
- Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE) are the two most common parameters being used to measure the quality of the images. However, none of them ensures the correctness of results. Their values may be sometimes misleading, so need to use some other parameter.
- Now a days, a lot of work is being done on Human Visual Based (HVS) based systems to analyse and process of images. So, the performance measuring methods should also take HVS into consideration, so that accurate quality of images can be checked.
- Quantizer designs may be further improved, as coarse quantization adds a lot of artifacts. So, an improved Quantizer will obviously reduce the artifacts.
- Some research analysts also tried their hands on other transforms like DWT, LOT, LBT etc. replacing DCT as a tool for compression purpose, but not any significant work has been done using Discrete Fractional Cosine Transform (DFrCT) which is generalisation of DCT and has got fractional order as an extra degree of freedom.

2.5 OBJECTIVES

Dissertation has following objectives:

1. To implement existing techniques of overlapped block processing using Discrete Cosine Transform (DCT) for the purpose of blocking artifacts reduction.
2. To vary the size of inside block during overlapped block processing and compare the results.
3. To use Discrete Fractional Cosine Transform (DFrCT) in the field of image compression and implement overlapped block processing using this as a transform coding technique.

4. To compare the above technique with existing method of overlapped block processing.

OVERLAPPED BLOCK PROCESSING FOR BLOCKING ARTIFACTS REDUCTION

CHAPTER 3

3.1 OVERLAPPED BLOCK PROCESSING

Due to ever increasing amounts of data transferred and stored, image compression is now more important than ever. The purpose of the image compression is to reduce the storage and transmission costs while maintaining image quality. Block based discrete cosine transform (BDCT) is amongst the most widely used transform for compression of images due to its energy compacting property and relative ease of implementation. In a typical BDCT compression scheme, the input image is segmented into non overlapping blocks of size $n \times n$, each block being transformed independently, quantized, coded and transmitted. However, at high compression ratios this block-based approach leads to noticeable blocking artifacts, artificial discontinuities that appear between the boundaries of the blocks resulting from the independent processing of the blocks without taking into account the inter-block pixel correlations. There is an obvious need of removing these blocking artifacts in the low-bit-rate transform compressed images. As it is well known that blocking artifacts have maximum effect at the edges of these blocks, it is reasonable that a slight overlap around the perimeter of each block could reduce the blocking effect. The pixels at the perimeter would then be transformed in two or more blocks. Thus, abrupt boundary discontinuities caused by transforming and quantizing are reduced. In this chapter overlapped block processing based technique is explained for the reduction of blocking artifacts. This technique is implemented using two types of transform coding techniques and results are compared using human visual system based peak signal to noise ratio (PSNR-HVS-M) as parameter of performance evaluation. Two transform coding techniques are:

- Discrete cosine transform (DCT)
- Discrete fractional cosine transform (DFrCT)

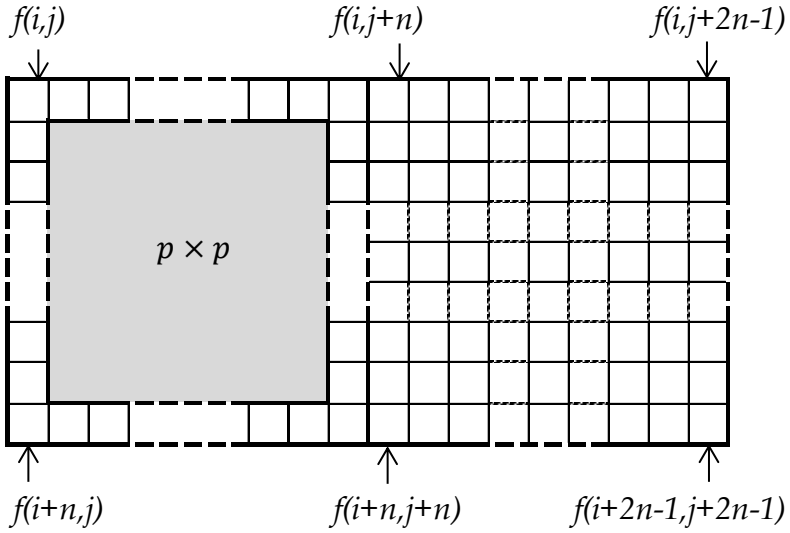
The details of DCT and DFrCT have already been explained in previous chapter

3.2 METHODOLOGY

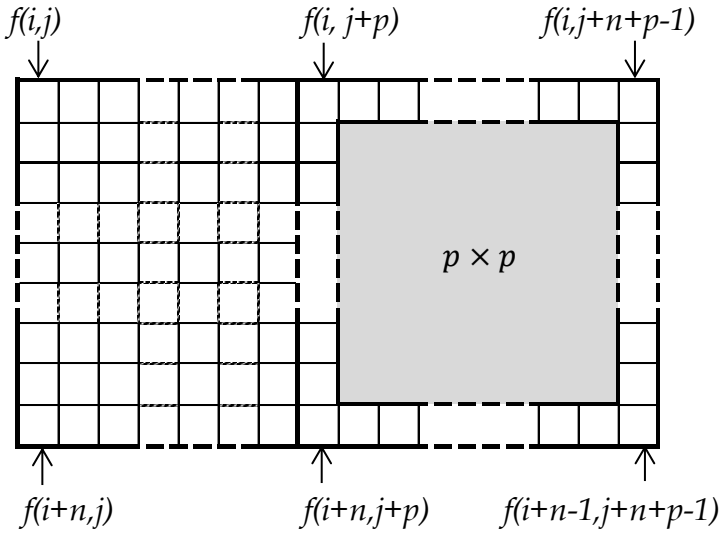
Consider an $N \times N$ image segmented into blocks of size $n \times n$. During overlapped block processing, an $n \times n$ block is transformed, quantized, inverse transformed and inside $p \times p$ block is saved.

- For non-overlapped block processing $p = n$.
- For overlapped block processing $p < n$.

First of all, we will explain how two horizontally adjacent blocks are being processed in the proposed method.



(a)



(b)

Figure 3.1 Overlapped block processing for horizontally adjacent blocks (a) first processing (b) next processing

- Consider an $n \times n$ block extracted from an $N \times N$ image with $f(i,j)$ being the very first pixel i.e. at top left corner, where (i,j) indexes the pixel co-ordinates for $N \times N$ image and $f(i+n-1\,,\ j+n-1)$ being the last pixel i.e. bottom right coordinate as shown in Figure 3.1

- This $n \times n$ block is transformed, quantized, inverse transformed and inside $p \times p$ block is saved.
- Now, the next $n \times n$ block, which also contains the last column of previous $n \times n$ block for the purpose of overlapping, having $f(i, j+p)$ and $f(i+n-1, j+n+p-1)$ as its first and last pixel coordinate, is transformed, quantized, inverse transformed and inside $p \times p$ block is saved.

The above three steps represents the procedure of overlapped block processing for two horizontally adjacent $n \times n$ blocks. Same criteria is to be followed for two vertical blocks as shown in Figure 3.2. For a whole image of size $N \times N$, it is scanned from left to right and from top to bottom.

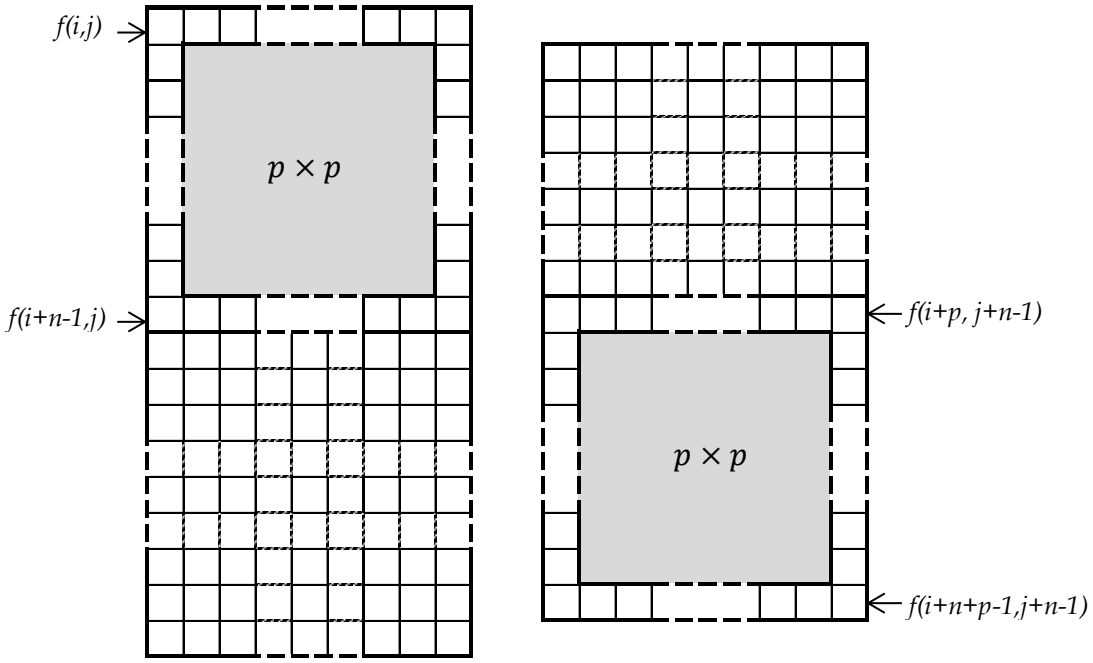
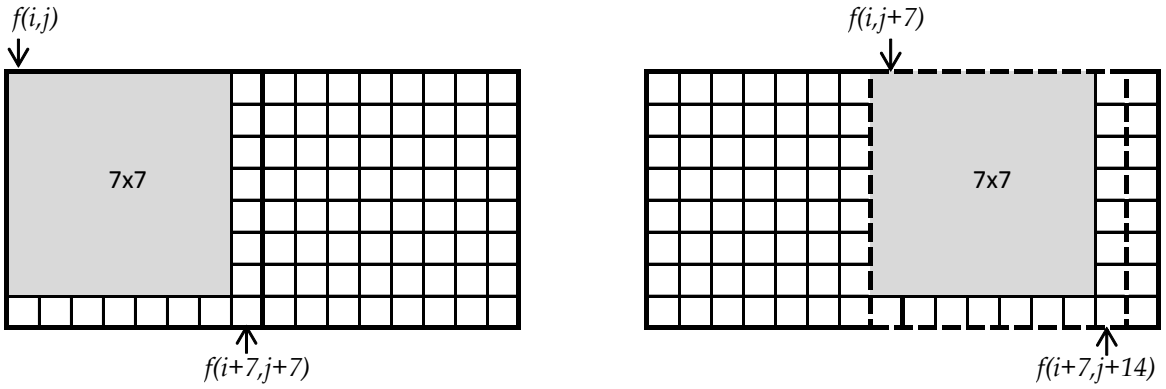


Figure 3.2 Overlapped block processing for vertically adjacent blocks

3.3 CASE OF 7×7 AND 6×6 INSIDE BLOCK SIZE

Image compression using block by block transform produces best results using block size of 8×8 i.e. taking $n = 8$ and thus $p < 8$ for overlapped block processing. To find out that which size of inside block ($p \times p$) yields better reduction in blocking artifacts and hence better quality of image, we have compared here different types of block sizes and it turns out to be that for $n = 8$, the best results are achieved when $p = 7$. For analysis purpose cases of $p = 6$ and $p = 7$ are explained here assuming $n = 8$.



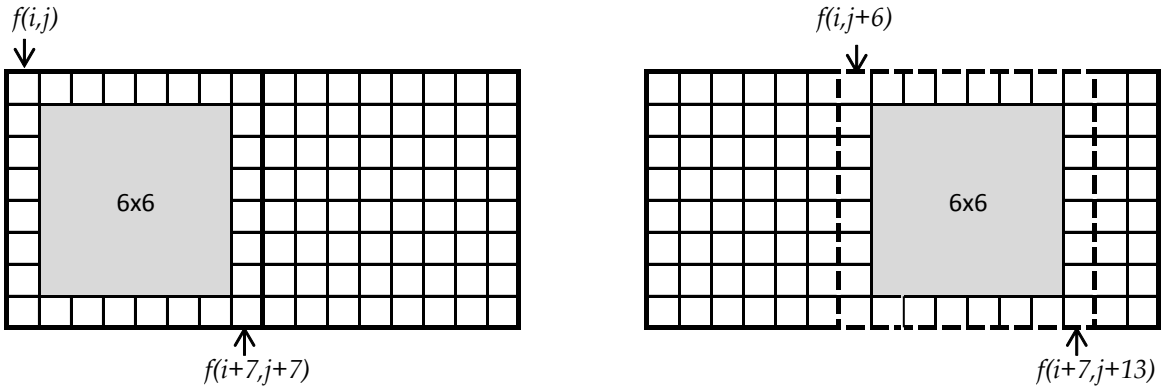
(a) (b)

Figure 3.3 Overlapped Block processing for the case of $n = 8$ and $p = 7$

Consider two horizontal adjacent 8 × 8 blocks having pixel values denoted by $f(i,j)$, where (i,j) indexes the pixel co-ordinates for $N \times N$ image, however 2nd 8 × 8 block is not utilized as such because of the overlapping. During a block processing, transformation of 8 × 8 block i.e. from pixels $f(i,j)$ to $f(i+7,j+7)$ is taken and inside 7 × 7 block i.e. pixels $f(i,j)$ to $f(i+6,j+6)$ are saved after quantization and inverse transformation as shown in Figure 3.3. Note that here we have considered inside 7 × 7 block leaving behind the last row and last column of 8 × 8 block. However it is not necessary to do so which makes it optional that we could have considered inside 7 × 7 block leaving behind the last row and first column of 8 × 8 block and results achieved will be almost similar.

During next process, 8 × 8 block containing pixels from $f(i,j+7)$ to $f(i+7,j+14)$ is processed and inside 7 × 7 block containing pixels $f(i,j+7)$ to $f(i+6,j+13)$ is saved as shown in Figure 3.3 and this process goes on until whole image is processed.

Similar procedure is followed for the case of $p = 6$.



(a) (b)

Figure 3.4 Overlapped Block processing for the case of $n = 8$ and $p = 6$

During a block processing, transformation of 8 × 8 block i.e. from pixels $f(i,j)$ to $f(i+7,j+7)$ is taken and inside 6 × 6 block are saved after quantization and inverse transformation as shown in Figure 3.4. During next process, 8 × 8 block containing pixels from $f(i,j+6)$ to $f(i+7,j+13)$ is processed and inside 6 × 6 block is saved. The performance of the this method is evaluated using various inside block sizes and it turns out to be that 7 × 7 produces better results than 6 × 6 and others also. The detailed results are discussed in next chapter.

3.4 8×8 KERNEL MATRICES

To implement this process using block size of 8 × 8, we need the kernel matrices of the same size. Two dimensional 8-point DCT and DFrCT kernel matrices are needed to be used for this purpose. 8-point DCT (Type-II) kernel is already defined in (2.6). For DFrCT, Let us take the 8-point, type I forward DCT as an example. The corresponding matrix [39] can be expressed as follows:

$$C = \sqrt{\frac{2}{8-1}} * \left[k_m k_n \cos(\frac{mn\pi}{8-1})\right] \tag{3.1}$$

where, $m = 0,1,2,\ldots.7 \; and \; n = 0,1,2,\ldots.7$

$$k_m = \begin{Bmatrix} \frac{1}{\sqrt{2}} \; for \; m = 0 \\ 1 \; otherwise \end{Bmatrix}$$

$$k_n = \begin{Bmatrix} \frac{1}{\sqrt{2}} \; for \; n = 0 \\ 1 \; otherwise \end{Bmatrix}$$

It can be diagonalized by

$$C = UVU^T \tag{3.2}$$

where,

— U is an orthonormal matrix obtained from the eigenvectors of C,

— V is a diagonal matrix composed of the corresponding eigenvalues,

— and U^T is the transpose matrix of U.

The square of the DCT matrix can be written as

$$C^2 = C * C = UV^2U^T \tag{3.3}$$

Similarly, we have

$$C^\alpha = C * C = UV^\alpha U^T \tag{3.4}$$

V^α and V are given as follows

$$V^{\alpha} = \begin{bmatrix} \lambda_1^{\alpha} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_2^{\alpha} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \lambda_3^{\alpha} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_4^{\alpha} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_5^{\alpha} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \lambda_6^{\alpha} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda_7^{\alpha} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_8^{\alpha} \end{bmatrix} \quad (3.5)$$

$$V = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad (3.6)$$

where α is a real fraction and λ_n^{α} are eigen values.

at α=1, U can be calculated as [39]:

$$U = \begin{bmatrix} -0.6104 & 0.4296 & 0.3887 & 0.3853 & -0.3300 & -0.1825 & -0.0308 & 0.0013 \\ -0.6972 & 0.0559 & -0.2235 & -0.4170 & 0.4481 & 0.2865 & 0.0617 & -0.0036 \\ -0.3587 & -0.6502 & -0.3998 & 0.0618 & -0.3679 & -0.3643 & -0.1289 & 0.0120 \\ -0.1102 & -0.5819 & 0.4675 & 0.3661 & 0.1558 & 0.4384 & 0.2799 & -0.0410 \\ -0.0174 & -0.2217 & 0.5941 & -0.3994 & 0.2080 & -0.3287 & -0.5196 & 0.1288 \\ -0.0003 & -0.0415 & 0.2536 & -0.5333 & -0.3312 & -0.1909 & 0.6207 & -0.3432 \\ 0.0004 & -0.0018 & 0.0557 & -0.2846 & -0.5054 & 0.4351 & -0.0241 & 0.6859 \\ 0.0001 & 0.0012 & 0.0081 & -0.1209 & -0.3494 & 0.4751 & -0.4941 & -0.6270 \end{bmatrix} \quad (3.7)$$

Thus, we have got every parameter required to calculate DFrCT of any fractional order (α). By varying the value of α between 0.1 and 1, one can compare different results at different values of α and find out its optimized value which achieves the best results.

3.5 STEP BY STEP ALGORITHM

The proposed method is elaborated through a generalised flow chart diagram for the whole image as shown in Figure 3.5 The step by step details explain the working of the flow chart. Assume an $n \times n$ block represented by $B_{a,b}(x, y)$, where a, b represents block co-ordinates and (x, y) indexes the pixel co-ordinates for $N \times N$ image. The step by step details explain the working of the flow chart. Assume an $n \times n$ block represented by $B_{a,b}(x, y)$, where a, b represents block co-ordinates and (x, y) indexes the pixel. co-ordinates for $N \times N$ image. The flowchart here explains the complete process with input being the original image produces compressed artifacts free image.

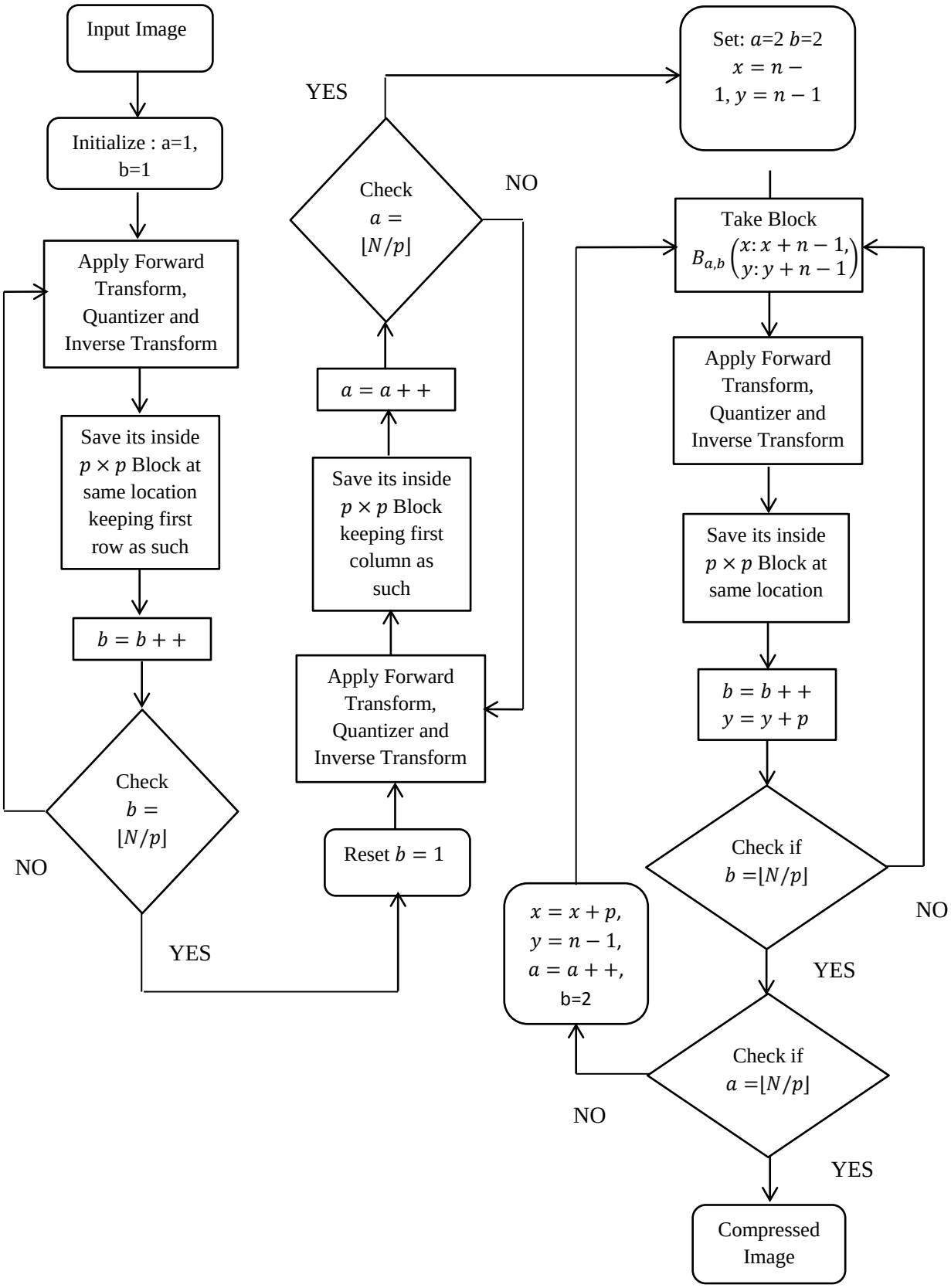


Figure 3.5 Flow chart diagram for overlapped block processing

There are basically three kinds of process explained in the flow chart. The first part explains for the case of a=1 , second part is for b=1 and the last one is for all the other values of a and b. This is done because a=1 represents the first row of blocks, for which we don't have any row of blocks above it to get overlapped with and similarly b=1 represents the first column of blocks, for which we don't have any column of blocks behind it to get overlapped with. Assume an $n \times n$ block represented by $B_{a,b}(x, y)$, where a, b represents block co-ordinates and (x, y) indexes the pixel. co-ordinates for $N \times N$ image. The flowchart here explains the complete process with input being the original image produces compressed artifacts free image. There are basically three kinds of process explained in the flow chart. The first part explains for the case of a=1 , second part is for b=1 and the last one is for all the other values of a and b. This is done because a=1 represents the first row of blocks, for which we don't have any row of blocks above it to get overlapped with and similarly b=1 represents the first column of blocks, for which we don't have any column of blocks behind it to get overlapped with. A simple solution to this problem is to retain some of the pixels as such. A simple solution to this problem is to retain some of the pixels as such. Same kind of problem will also arise for the last row and column of blocks and solution is also same which is to retain some pixels as such.

- If $a = 1$, first row has to be retained as such and same is the case for first column when $b = 1$.
- If $a = b = 1$, first row and column both are retained.

The step by step algorithm is explained as follows:

- To start with, the very first block i.e. the top left block is processed for which $a = b = 1$.
- Apply forward transform, quantizer and inverse transform to the block.
- Save its inside $p \times p$ block at the same location from where the original block was picked keeping its first row as such
- Increment b by 1 and thus moving forward from left to right and process block by block. Repeat the process until we reach at the end of the blocks of first row.
- Then reset the value of b to 1 and process the first column of blocks in the same fashion for which a=1, keeping first row as such for each block until we reach $a = \lfloor N/p \rfloor$, where $\lfloor . \rfloor$ represents the floor function

- set $a = b = 2$ i.e. select first block of 2nd row and 2nd column of blocks which is $B_{2,2}(x, y)$. Also set $x = n - 1$ and $y = n - 1$ where $n \times n$ is the block size of image.
- Consider $B_{a,b}(x: x + n - 1, y + n - 1)$ block of pixels to process. Note that initially $a = b = 2$, hence corresponding block is processed to start with.
- Apply forward transform, quantizer and inverse transform to the block mentioned above.
- Save its inside $p \times p$ block at the same location from where the original block was picked.
- Increment b by 1 and y by p i.e. $b = b + 1$ and $y = y + p$. This selects the next block to process moving from left to right.
- At this stage, we check if $b = \lfloor N/p \rfloor$,. If it comes out to be true, then it means we have reached at the end of row of blocks and needed to jump on to next row of blocks by setting $x = x + p$, $y = n - 1$, $a = a + +$ and $b = 2$ after confirming that if $a = \lfloor N/p \rfloor$ which implies that it is end of last row of blocks i.e. image has been completely scanned. However if $b \neq \lfloor N/p \rfloor$, then step-2 is executed again and this is repeated until we get $b = \lfloor N/p \rfloor$.

4.1 COMPARATIVE ANALYSIS OF DIFFERENT SIZES OF INSIDE BLOCKS

In order to evaluate the performance of our algorithm for the case of $n = 8$ and subcase of $p = 5, 6$ and 7, it is applied to the Lena image (256×256) at various percentages of compression. The compression percentage is defined as:

$$\text{Compression percentage} = \frac{n_1}{n_2} \times 100 \tag{4.1}$$

where n_1 and n_2 are the number of information units in two data sets.

The method is implemented using two transform coding techniques - DFrCT and DCT (Type-II) to find out the best inside block size which produces the greatest value of PSNR-HVS-M. Table 4.1 shows the comparison results of normal block processing and proposed method, where angle (α) denotes the fractional order of DFrCT. For the analysis purpose, the results for some selected values of α at different compression levels are shown here. The proposed method is further evaluated using three inside block sizes : 5 × 5, 6 × 6 and 7 × 7. It can be easily proved from Table 4.1 that 7 × 7 inside block size produces better results than the other two in case of DFrCT.

Table 4.1 PSNR-HVS-M(dB) results for Lena image (256×256) using DFrCT

Compression Percentage	Angle		α = 0.93	α = 0.95	α = 0.98	α = 0.99	α = 1
70%	Normal Block processing		38.0536	40.0873	43.4659	44.3080	43.6803
	Proposed Method	5 × 5	38.8723	42.1098	46.2987	46.1212	45.5432
		6 × 6	39.3237	42.4817	46.5025	46.7508	45.9465
		7 × 7	40.4024	42.6064	46.7218	46.7218	46.1214
75%	Normal Block processing		34.8046	36.9233	40.1921	40.9660	39.9723
	Proposed Method	5 × 5	35.6502	38.3452	42.1793	43.1129	42.8709
		6 × 6	36.3807	39.1168	42.8272	43.3664	42.9022
		7 × 7	37.6998	39.6795	42.9837	43.4220	42.9225
80%	Normal Block processing		31.2534	33.2541	36.0537	36.3936	35.9952
	Proposed Method	5 × 5	32.0934	34.7862	38.4532	38.5782	38.6826
		6 × 6	32.6337	35.5602	38.6509	38.8154	38.7213
		7 × 7	34.0357	36.1671	38.6524	38.7906	38.8521

Figure 4.1 Lena image(256×256) at 70% compression using DFrCT (a) and (c) are images using normal block processing at α=0.95 and α=0.99 respectively (b) and (d) are images using proposed method at α=0.95 and α=0.99 respectively. (e) and (f) are zoomed parts of (a) and (b).

Figure 4.2 Lena image (256×256) at two different compression levels using DCT (a) and (c) are images using normal block processing at 90% and 95% compression respectively, (b) and (d) are images using proposed method at 90% and 95% compression respectively. (e) is zoomed part of (c) and (f) is zoomed part of (d).

The subjective improvement in the quality of images by applying proposed method using DFrCT can be seen in Figure 4.1, which shows the comparative results of Lena image at 70% compression levels for α=0.95 and α=0.99. Similar comparison is made using DCT at different compression levels and 7 × 7 proved out to be better than 5 × 5 and 6 × 6 here too as shown in Table 4.2.

Table 4.2 : PSNR-HVS-M(dB) results for Lena image(256×256) using DCT (Type-II)

Compression percentage	70%	80%	85%	90%	95%
JPEG	83.4302	45.5194	39.8700	33.4890	27.5275
Proposed Method 5 × 5	87.5637	45.7834	40.0935	34.2675	29.1917
Proposed Method 6 × 6	88.2702	45.9881	41.3675	34.9915	29.1740
Proposed Method 7 × 7	89.0042	46.5650	41.3718	35.4060	29.0154

Figure 4.2 shows the comparative results of Lena image at 90% and 95% compression levels for DCT (Type-II). Comparing results of Table 4.1 and Table 4.2, one can conclude that DCT (Type-II) produces better results than DFrCT and hence DCT (Type-I). So, further any kind of analysis of this method is done using DCT (Type-II) only.

4.2.1 COMPUTATIONAL COMPLEXITY

In order to prove the further benefits of 7 × 7 inside block size over others, computational complexity is counted out for different image sizes using various inside block sizes.

Table 4.3 Computational cost count for different sizes of image using 8×8 block size

Image size	Inside Block Size	Number of real arithmetic computations
256×256	7 × 7	74906
	6 × 6	101955
	5 × 5	150332
512×512	7 × 7	307962
	6 × 6	2227106
	5 × 5	3207033

Although the direct application of formulae, N-point DCT would require N^2 real arithmetic operations. It is possible to compute the same thing with only $N \log N$ complexity by factorizing the computation similarly to the fast Fourier transform (FFT). However, for 2-D DCT applied on an $N \times N$ image using block size of $n \times n$, the number of operations becomes $\left(\frac{N}{n}\right)^2 n^2 \log n$. Now, assuming an $N \times N$

image segmented using block size of $n \times n$ and $p \times p$ inside block size, the number of blocks processed for normal processing is $\left[\lfloor \frac{N}{p} \rfloor\right]^2$, where $\lfloor . \rfloor$ represents the floor function. Thus Total number of computations become equal to $\left[\lfloor \frac{N}{p} \rfloor\right]^2 n^2 \log n$. The number of computations is counted for different sizes of inside blocks and 7 × 7 proved out to be the best among them as shown in Table 4.3

4.3 COMPARATIVE ANALYSIS WITH REEVE's METHOD

In order to evaluate the performance of the proposed algorithm, the results are compared with overlap method [16] in DCT domain proposed by H. C. Reeve *et al*. The experiment is performed on Lena image at different compression levels due to unavailability of the image used in Reeve's method. To demonstrate the performance of both the techniques, the first experiment is conducted at five different compression levels ranging from 70% to 95%. The author used Normalised Mean Square Error (NMSE) in method [16] to evaluate the performance, hence the same is used here. Table 4.1 shows that our technique gives better results than method [16] in terms of NMSE, for Lena image of size 256×256 at various compression levels.

Table 4.4 NMSE (%) comparison of proposed method and Reeve's Method [16]

	Compression percentage					
	70%	80%	85%	90%	92%	95%
Proposed Method 7 × 7	3.26×10^{-6}	0.00834	0.02124	0.05457	0.06649	0.09980
Reeve's Method [16]	3.14×10^{-6}	0.00911	0.02237	0.05777	0.06929	0.10516

One of the drawback of this method is that it reduces artifacts on the expense of computational cost, however ours is lesser than method [16] as shown in Table

Table 4.5 Computational cost comparison of two methods for different sizes of images

Image size	Method	Number of real arithmetic computations
256×256	Non-overlapped Block processing	59185
	Proposed method (7×7)	74906
	Reeve's Method [16]	99873
512×512	Non-overlapped Block processing	118400
	Proposed method (7×7)	307962
	Reeve's Method [16]	399492

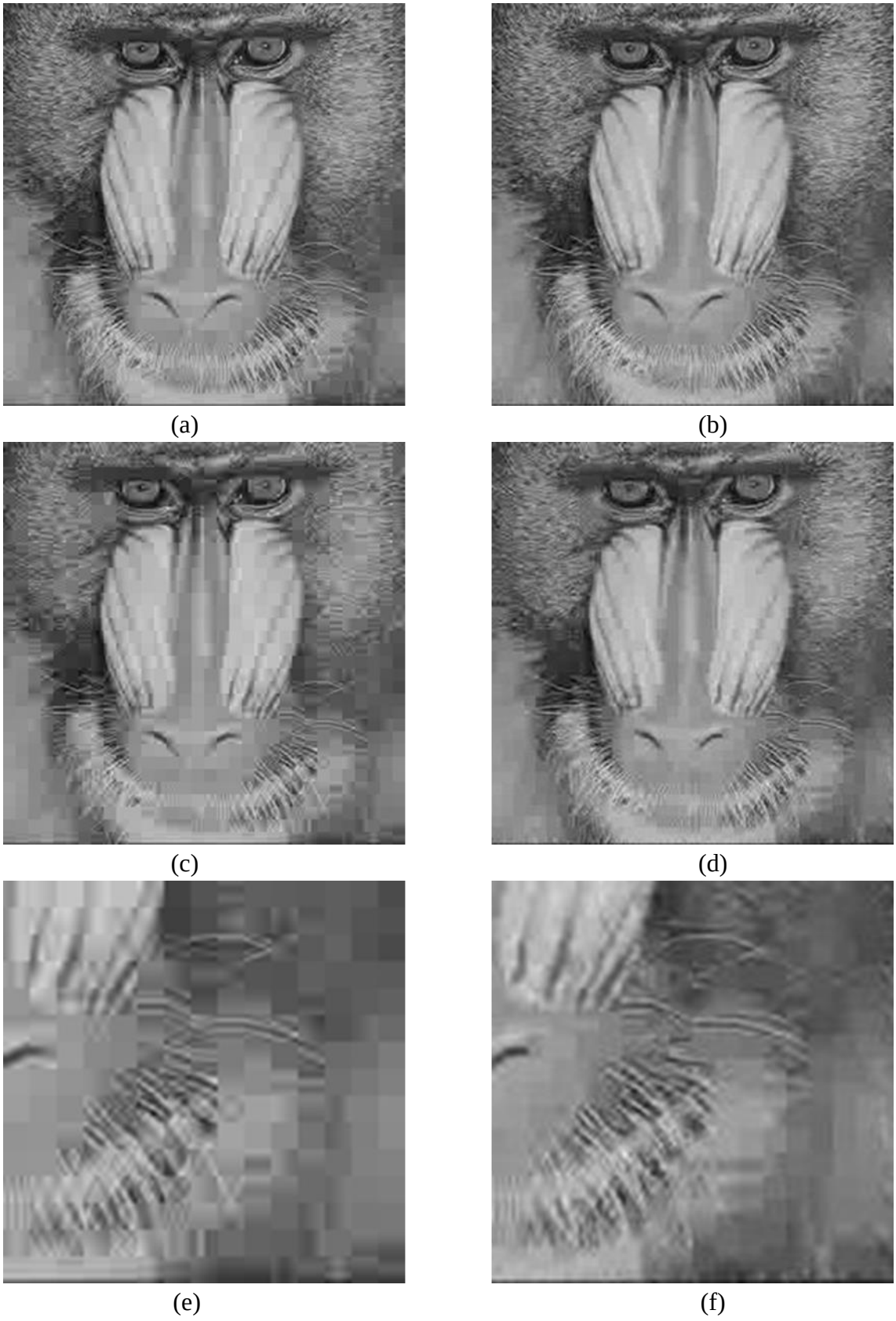


Figure 4.3 Baboon image (256×256) at two different compression levels using DCT (a) and (c) are images using normal block processing at 90% and 95% compression respectively, (b) and (d) are images using proposed method at 90% and 95% compression respectively. (e) is zoomed part of (c) and (f) is zoomed part of (d).

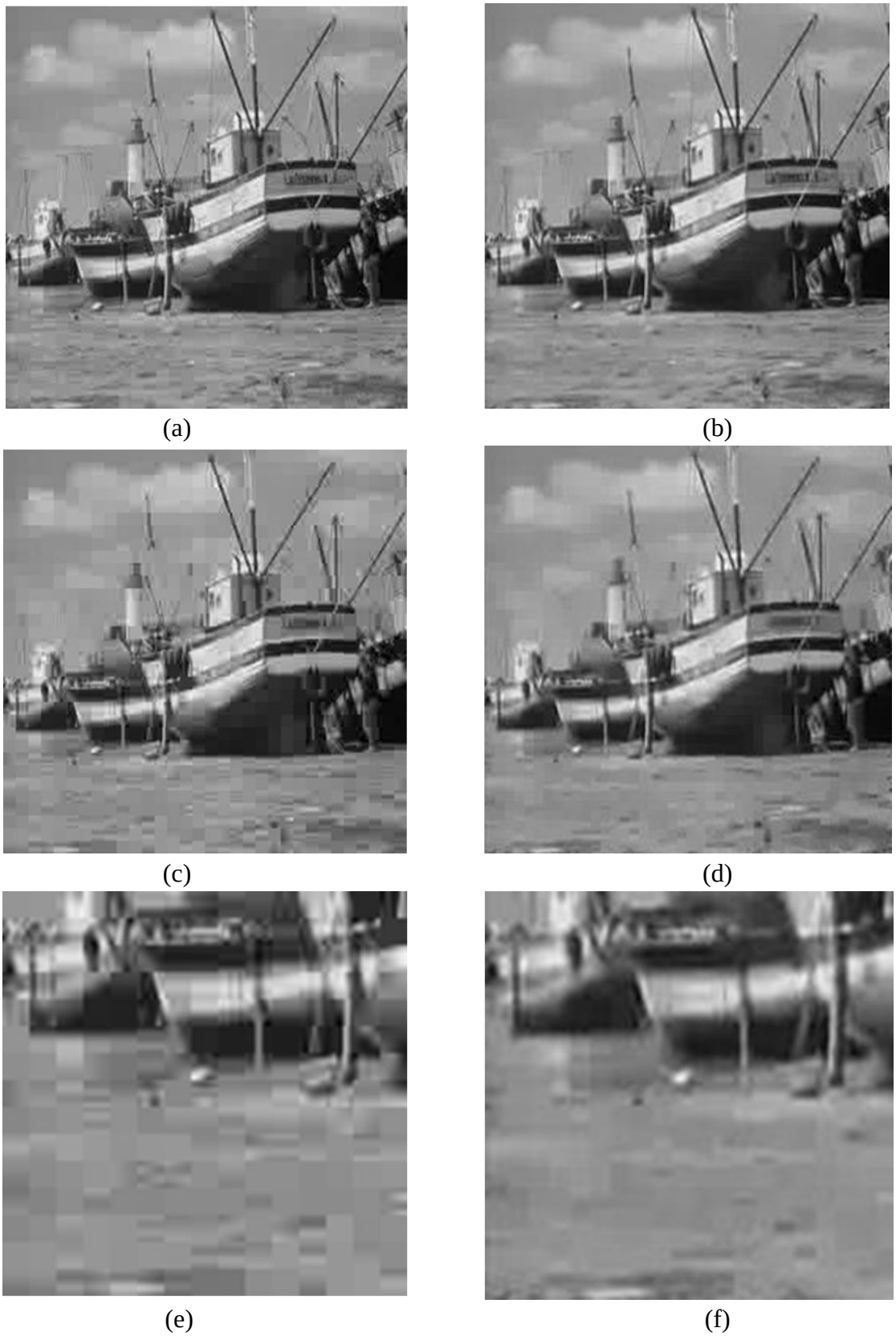


Figure 4.4 Boat image (256×256) at two different compression levels using DCT (a) and (c) are images using normal block processing at 90% and 95% compression respectively, (b) and (d) are images using proposed method at 90% and 95% compression respectively. (e) is zoomed part of (c) and (f) is zoomed part of (d).

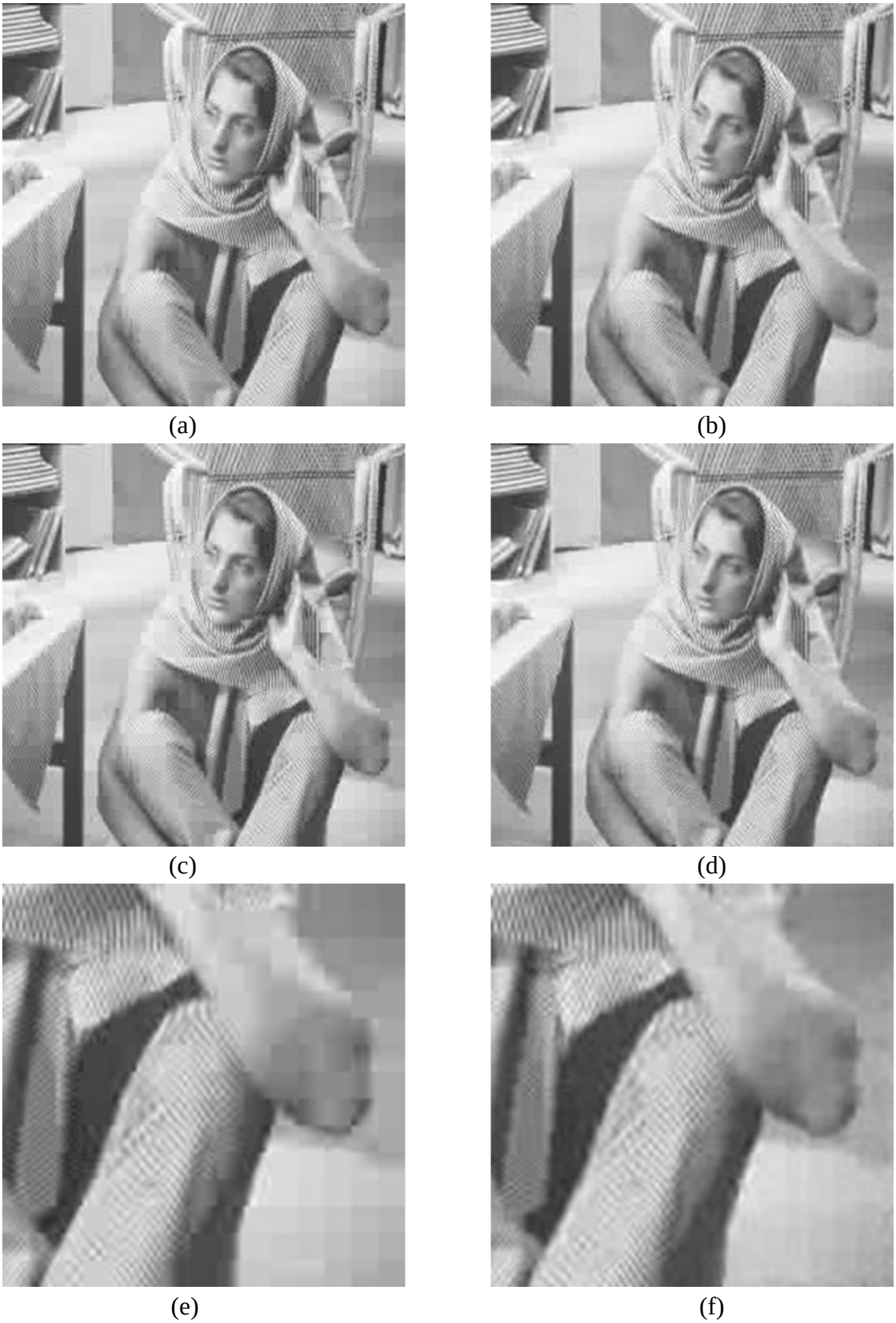


Figure 4.5 Barbara image (256×256) at two different compression levels using DCT (a) and (c) are images using normal block processing at 90% and 95% compression respectively, (b) and (d) are images using proposed method at 90% and 95% compression respectively. (e) is zoomed part of (c) and (f) is zoomed part of (d).

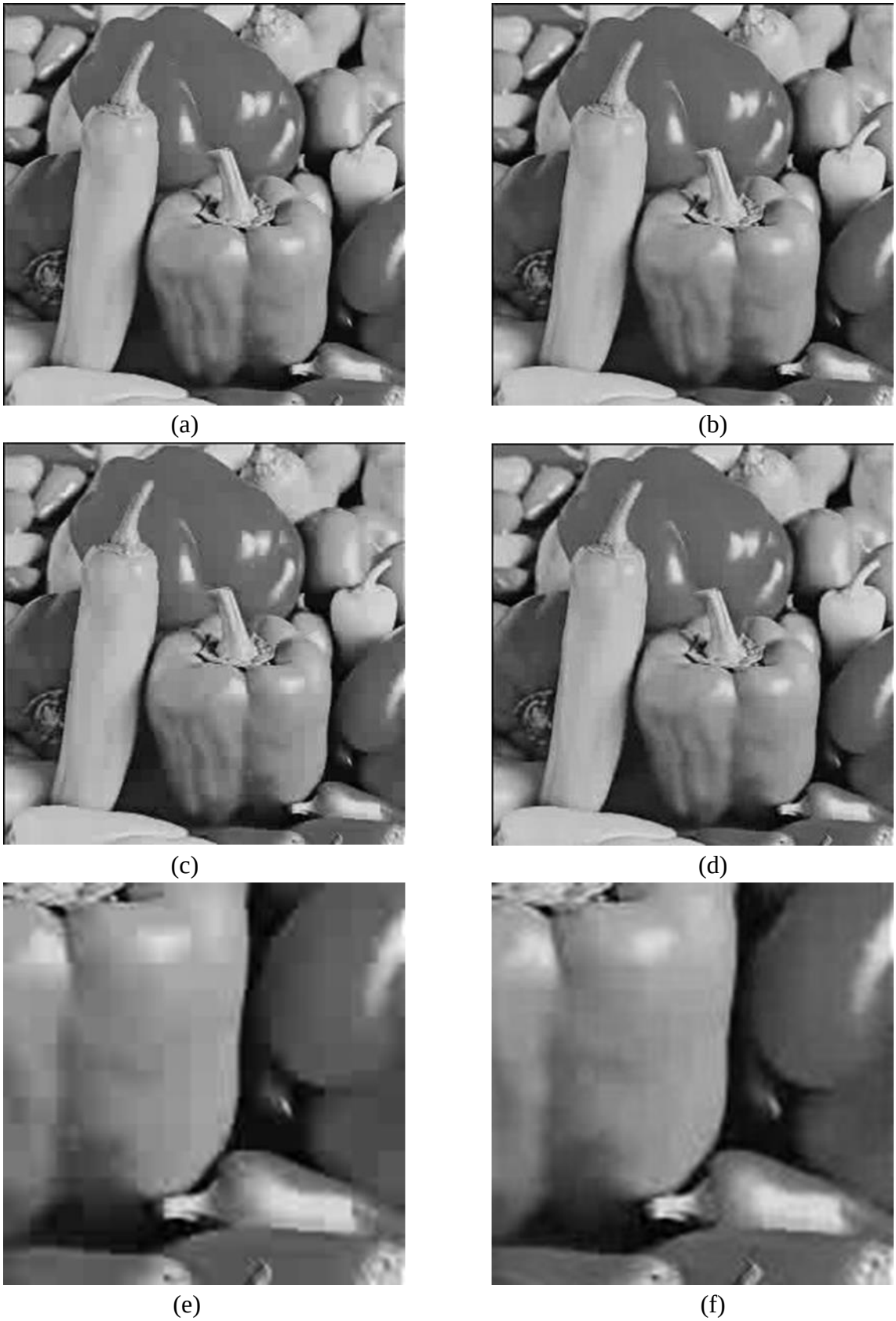


Figure 4.6 Peppers image (256×256) at two different compression levels using DCT (a) and (c) are images using normal block processing at 90% and 95% compression respectively, (b) and (d) are images using proposed method at 90% and 95% compression respectively. (e) is zoomed part of (c) and (f) is zoomed part of (d).

4.4 DIFFERENT IMAGES RESULTS USING DCT (TYPE-II)

Further, to evaluate the performance of this algorithm in terms of PSNR-HVS-M, it has been applied to variety of images of size 256×256 using DCT (Type-II) with inside block size of 7 × 7 and the results are shown in Table 5.3.

Table 4.6 PSNR-HVS-M(dB) results for various images (256×256) using DCT (Type-II)

Compression percentage \ Image		Baboon	Boat	Barbara	Peppers
70%	JPEG Method	35.3848	80.5315	79.9703	47.1049
	Proposed Method	38.4852	81.9132	83.5778	47.7357
80%	JPEG Method	33.2722	41.9704	41.0939	42.2562
	Proposed Method	33.4349	42.0970	41.8874	43.5782
85%	JPEG Method	30.4740	35.9220	35.7248	39.6206
	Proposed Method	30.2634	37.1189	36.7905	40.8671
90%	JPEG Method	27.5760	31.7741	32.1140	35.2152
	Proposed Method	29.2348	32.6985	33.2734	37.2391
95%	JPEG Method	24.3687	26.5741	26.8813	27.7274
	Proposed Method	25.8087	27.6501	28.1245	29.5815

Figure 4.2 – 4.6 demonstrate the five types of images to show improvement in subjective quality. In each figure there are six images (a)-(e), where (a) and (c) are the images obtained using normal block processing, at 90% and 95% compression respectively, (b) and (d) are the images using proposed method at same compression levels. To show the details of improvement in subjective quality, zoomed part of each image is shown in (e) and (f). The proposed method yields good results, achieving favourably both aspects of reducing blocking artifacts and preserving image details. Each decoded image is visually more pleasing than the images produced by normal block processing method. As well, the best values of PSNR-HVS-M have been obtained using the proposed technique for transform coded images at various compression levels.

CONCLUSION AND FUTURE SCOPE | CHAPTER 5

5.1 CONCLUSION

A simple yet effective algorithm has been developed in this dissertation for removing blocking artifacts in JPEG compressed images. The method of overlapped block processing is proposed here to reduce these artifacts. Several images were tested and fruitful results were obtained for various kinds of images. Quantitative and qualitative measures have been evaluated for all the images. Human visual system based peak signal to noise ratio (PSNR-HVS-M) is used to measure objective quality of image instead of PSNR because of the lack of ability of PSNR to successfully evaluate any image. This algorithm when applied using DCT successfully reduces the artifacts by significant improve in PSNR-HVS-M of 1.917 dB at 90% compression level for 256×256 Lena image with 7×7 inside block, but at the expense of 26.5% increase in real arithmetic computations. Also, this method proves out to be better than Reeve's method [16] in terms of Normalised Mean Square Error (NMSE) and computational complexity. An attempt is also made to apply the same algorithm using DFrCT which also results in improvement in PSNR-HVS-M at higher values of fractional order. However, DCT (Type-II) is proved out to be a better transform than DFrCT and DCT (Type-I) for the purpose of image compression.

5.2 FUTURE SCOPE

One future research direction is to improve the complementary parts of the proposed system. Some different kind of approach to implement overlapping can be thought of to reduce blocking artifacts keeping computational cost on check. The performance of Discrete Fractional Cosine Transform (DFrCT) may be improved by suitably choosing eigen values and eigen vectors required for its calculation. Further exploration may also be done to calculate DFrCT from the Type-II form of DCT, which may produce better results in the area of image processing. However, it is worth noting again that the purpose of this dissertation is not to show the optimality of the particular system that is proposed. The purpose of this dissertation is to demonstrate that the approach to use overlapped block processing and DFrCT for deblocking has significant benefits and should be explored further in the future research for reduction of blocking artifacts.

REFERENCES

[1] A. B. Watson, and J. A. Solomon, "Model of visual contrast gain control and pattern masking," *Journal of the Optical Society of America a-Optics Image Science and Vision*, vol. 14 no. 9, pp. 2379–2391, 1997.

[2] A. B. Watson, "Image compression using the discrete cosine transform," *Mathematica Journal*, vol.4, no.1, pp. 81-88, 1994.

[3] A. Chetouani , G. Mostafaoui and A. Beghdadi, "Deblocking filtering method using a perceptual map," *Signal Processing: Image Communication*, vol. 25, pp. 527–534, 2010.

[4] A. Kaup, "Adaptive constrained least squares restoration for removal of blocking artifacts in low bit rate video coding," *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 97, pp. 2913-2916, 1997.

[5] A. W. C. Liew, and H. Yan, "Blocking artifacts suppression in block-coded images using overcomplete wavelet representation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 4, pp. 450-461, 2004.

[6] A. Z. Averbuch, , A. Schclar, and D. L. Donoho, "Deblocking of block- transform compressed images using weighted sums of symmetrically aligned pixels," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 200-212, 2005.

[7] A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, no. 1, pp. 91-95, 1992.

[8] A.C. McBride, and F. H. Keer, "On Namia's fractional Fourier transform," *IMA Journal of Applied Mathematics*, vol. 239, pp. 159-175, 1987.

[9] B. Macq, M. Mattavelli, O. V. Calster, E. V. D. Plancke, S. Comes, and W. Li, "Image visual quality restoration by cancellation of the unmasked noise," *In Proc. IEEE Conference on Acoustics, Speech, and Signal Processing*, pp. 53-56, 1994.

[10] B. W. Dickinson, and K. Steiglitz, "Eigenvectors and functions of the discrete Fourier transform," *IEEE Transactions on Acoustic, Speech, Signal Processing*, vol. 30, pp. 25–31, 1982.

[11] C. Kim, "Adaptive post-.filtering for reducing blocking and ringing artifacts in low bit-rate video coding," *Signal Processing: Image Communication*, vol. 17, pp. 525-535, 2002.

[12] C. Wang , J. Zhou, and S. Liu, "Adaptive non-local means filter for image

deblocking," *Signal Processing: Image Communication*, vol. 28, pp. 522–530, 2013.

[13] G. Cincotti, F. Gori, and M. Santarsiero, "Generalized self-Fourier functions," *Journal of Physics*, vol. 25, pp. 1191–1194, 1992.

[14] G. Qiu, "MLP for adaptive postprocessing block-coded images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 8, pp. 1450-1454, 2000.

[15] G.K. Wallace, "The JPEG still-picture compression standard," *Communications of the ACM*, vol. 34, pp. 30-44, 1991

[16] H. C. Reeve, and J. S. Lim, "Reduction of blocking effect in image coding," *In Proc. IEEE Conference on Acoustics, Speech, and Signal Processing*, pp. 1212-1215, 1983.

[17] H. Paek, R. C. Kim, and S. Lee, "On the POCS-based postprocessing technique to reduce the blocking artifacts in transform coded images," *IEEE Transactions on Circuits and Systems for Video Technology***,** vol. 8, no. 3, pp. 358-367, 1998.

[18] H. S. Malvar, "Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts," *IEEE Transactions on Signal Processing*, vol. 46, no.4, pp. 1043–1053, 1998.

[19] H. S. Malvar, and D. H. Staelin, "The LOT: Transform coding without blocking effects," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 4, pp. 553-559, 1989.

[20] H. W. Park, and Y. L. Lee, "A postprocessing method for reducing quantization effects in low bit-rate moving picture coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 161-171, 1999.

[21] International Organization for standardization and International Electrotechnical Commission, "Digital compression and coding of continuous-tone still images:part 1: requirements and guidelines" *ISO/IEC IS 10918-1 ITU-T T.81*, 1994

[22] J. Chou, M. Crouse, and K. Ramchandran, "A simple algorithm for removing blocking artifacts in block-transform coded images," *IEEE Signal Processing Letters*, vol. 5, no. 2, pp. 33-35, 1998.

[23] J. G. Proakis and D. G. Manolakis, "Digital signal processing: principles, algorithms and applications," Prentice Hall Inc., Third Edition, New Jersey, 1996.

[24] J. H. McClellan and T. W. Parks, "Eigenvalue and eigenvector decomposition of the discrete Fourier transform," *IEEE Transactions on Audio Electroacoustics*, vol. 20, pp. 66–74, 1972.

[25] J. L. Mitchell, W.B. Pennebaker, C.E. Fogg, and D.J. LeGall, "MPEG Video

Compression Standard," Chapman & Hall, First Edition, New York, 1997.

[26] J. M. Foley "Human luminance pattern-vision mechanisms- masking experiments require a new model," *Journal of the Optical Society of America a-Optics Image Science and Vision*, vol. 11 no. 6, pp. 1710-1719, 1994.

[27] J. Singh, S. Singh, D. Singh, and M. Uddin, "Efficient DCT domain blind measurement of blocking artifacts," *Journal of Information and Computing Science,* vol. 5, pp. 47-54, 2010.

[28] K. Singh, "Performance of discrete fractional Fourier transform classes in signal processing applications," Thesis, Ph.D, Thapar University, Patiala, 2006

[29] K. Veeraswamy, B. Chandra Mohan, and T. Jyothirmayi, "An image compression scheme using AC predictions," *In Proc. IEEE International Conference on Computational Intelligence and Multimedia Applications*, pp. 25-30, 2007.

[30] L. B. Almeida, "The fractional Fourier transform and time-frequency representation," *IEEE Transaction on Signal Processing*, vol. 42, pp. 3084-3091, 1994.

[31] N. I. Cho, B. G. Roh, and S. Lee, "Reduction of blocking artifacts by a modeled lowpass filter output," *In Proc. IEEE International Symposium on Circuits and Systems*, pp. 673-676, 2000.

[32] N. Ponomarenko, F. Battisti, K. Egiazarian, J. Astola, and V. Lukin "Metrics performance comparison for color image database," *In Proc. Fourth International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, pp. 14-16, 2009.

[33] N. Wiener, "Hermitian polynomials and Fourier analysis," *Journal of Mathematical Physics*, vol. 8, pp. 70-73, 1929.

[34] R. C. Reininger, and Jerry D. Gibson, "Distributions of two-dimensional DCT coefficients for images," *IEEE Transactions on Communications*, vol. 31, no. 6, pp. 835-839, 1983.

[35] R. Gonzalez and R. Woods, "Digital image processing," Prentice Hall Inc., Second Edition, New Jersey 2002.

[36] R. N. Bracewell, "The Fourier transforms and its applications," McGraw-Hill, Second Edition, New York, 1986.

[37] R. Pajarola and P. Widmayer, "An image compression method for spatial search," *IEEE Transaction on Image Processing*, vol. 9, no. 3, pp. 357-365, 2000.

[38] S. C. Pei and M. H. Yeh, "Improved discrete fractional Fourier transform," *Optics*

Letters, vol. 22, pp. 1047–1049, 1997.

[39] S. C. Pei and M. H. Yeh, "The discrete fractional cosine and sine transforms," *IEEE Transactions on Signal Processing*, vol. 49, no. 6, pp. 1198–1207, 2001.

[40] S. C. Pei, M. H. Yeh, and C. C. Tseng, "Discrete fractional Fourier transform based on orthogonal projection," *IEEE Transactions on Signal Processing*, vol. 47, pp. 1335–1348, 1999.

[41] S. Minami, and A. Zakhor, "An optimization approach for removing blocking effects in transform coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 2, pp. 74-82, 1995.

[42] 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.

[43] S. Wu, H. Yan, and Z. Tan, "An efficient wavelet-based deblocking algorithm for highly compressed images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 11, pp. 1193-1198, 2001.

[44] T. Chen, H. R. Wu, and B. Qiu, "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 5, pp. 594-602, 2001.

[45] T. Meier, K. N. Ngan, and G. Crebbin, "A region-based algorithm for enhancement of images degraded by blocking effects," *IEEE Transactions on Digital Signal Processing Applications*, vol. 1, pp. 405–408, 1996.

[46] T. P. O'Rourke, and R. L. Stevenson, "Improved image decompression for reduced transform coding artifacts," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 6, pp. 490-499, 1995.

[47] V. Namias, "Fractionalization of Hankel transforms," *Journal of the Institute of Math Applications*, vol. 26, pp. 187-197, 1980.

[48] V. Namias, "The fractional order Fourier transform and its application to quantum Mechanics," *Journal of Institute of Mathematics and its Applications*, vol. 25, pp. 241-265, 1980.

[49] www.cs.cf.ac.uk

[50] www.judge.u-aizu.ac.jp

[51] Y. Luo, R. K. Ward, "Removing the blocking artifacts of block-based dct compressed images," *IEEE Transactions on Image Processing*, vol. 12, no.7, pp. 838–842, 2003.

[52] Y. Yang, and N. P. Galatsanos, "Removal of Compression Artifacts Using Projections

onto Convex Sets and Line Process Modeling," *IEEE Transactions on Image Processing*, vol. 6, no. 10, pp. 1345-1357, 1997.

[53] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Regularized Reconstruction to Reduce Blocking Artifacts of Block Discrete Cosine Transform Compressed Images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 6, pp. 421-431, 1993.

[54] Y. Q. Zhang, L. R. Pickholtz, and H. M. Loew, "A new approach to reduce the blocking effects of transform coding," *IEEE Transactions on Communications*, vol. 41, no. 2, pp. 299-302, 1993.

[55] M. Yan, Z. Lin, and C. Zhang, "A new Fourier transform approach for protein coding measure based on the format of the Z curve," *Journal of Bio informatics*, vol.14, no. 8, pp. 685-690, 1995.

[56] Z. Wang, "Fast algorithm for the discrete W transform and for the discrete Fourier transform," *IEEE Transactions on Acoustic Speech, and Signal Processing*, vol. 32, pp. 803–816, 1984.

[57] Z. Wang, and A. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, pp. 81–84, 2002.

LIST OF PUBLICATION(S)

S. Singh, and K. Singh, "Blocking Artifacts Reduction By Overlapped Block Processing Using Discrete Fractional Cosine Transform," – Communicated to *Taylor and Francis International Journal of Electronics (SCI Indexed).*