Discrete Tchebichef Transform and its Application to Image/ Video Compression

Sujata Ishwar

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

June 2008

Your file Votre référence
ISBN: 978-0-494-42493-3
Our file Notre référence
ISBN: 978-0-494-42493-3

Abstract

Discrete Tchebichef Transform and its Application to Image/ Video Compression

Sujata Ishwar

The discrete Tchebichef transform (DTT) is a novel polynomial-based orthogonal transform. It exhibits interesting properties, such as high energy compaction, optimal decorrelation and direct orthogonality, and hence is expected to produce good transform coding results. Advances in the areas of image and video coding have generated a growing interest in discrete transforms. The demand for high quality with a limited use of computational resources and improved cost benefits has lead to experimentation with novel transform coding methods. One such experiment is undertaken in this thesis with the DTT.

We propose the integer Tchebichef transform (ITT) for 4x4 and 8x8 DTTs. Using the proposed ITT, we also design fast multiplier-free algorithms for 4-point and 8-point DTTs that are superior to the existing algorithms.

We perform image compression using 4×4 and 8×8 DTT. In order to analyze the performance of DTT, we compare the image compression results of DTT, discrete cosine transform (DCT) and integer cosine transform (ICT). Image quality measures that span both the subjective and objective evaluation techniques are computed for the compressed images and the results analyzed taking into account the statistical properties of the images for a better understanding of the behavioral trends. Substantial improvement is observed in the quality of DTT-compressed images.

The appealing characteristics of DTT motivate us to take a step further to evaluate the computational benefits of ITT over ICT, which is currently being used in the H.264/ AVC standard. The merits of DTT as demonstrated in this thesis are its simplicity, good image compression potential and computational efficiency, further enhanced by its low precision requirements.

Acknowledgements

I express my deep gratitude to my research supervisor, Dr. M. N. Srikanta Swamy for his encouragement, financial support and his quick responses in all the phases of this thesis. I sincerely thank him for his incredible patience, invaluable suggestions, keen observations and remarks that were indispensable to this submission. His distinguished personality and his attitude have been an inspiration to me throughout this thesis work. I am especially thankful to him for giving me an opportunity to work on this thesis, which has been a truly enriching and an enjoyable experience.

I thank Dr. Pramod Kumar Meher for his insight, suggestions and encouraging words during the progress of this thesis. Our collaboration was indeed a fruitful one.

I am very grateful to my parents for having believed in me always. Their love and support in this step of my life, as well as in other career decisions have been my main source of motivation.

I am thankful to my Professor Dr. K. A. Radhakrishna Rao for having taught Digital Signal Processing during my undergraduate studies in an unconventional manner that instilled my interest in this subject.

I am grateful to Mina for her time and enthusiasm during our discussions on Latex. I also thank Ibrahim for his kind generosity and the fatherly affection he bestowed upon me.

I am thankful to my dear friend, Akshay, for being a great source of strength, and also for being there for me always. I am thankful to my sister and my brother-in-law for their support and encouragement. I also thank all my friends in Montreal for their companionship and support that made my life away from home easier.

Dedicated to my Father and my Mother

वक्रतुण्ड महाकाय सूर्य कोटि समप्रभ ।

निर्विघ्नं कुरु मे देव सर्व कार्येषु सर्वदा ॥

Contents

List of Figures

List of Tables

Nomenclature

List of Abbreviations

AD	average difference
AVC	advanced video coding
BMP	microsoft windows bitmap
bpp	bits per pixel
CCTV	closed circuit television
DCT	discrete cosine transform
DFT	discrete Fourier transform
DHT	discrete Hartley transform
DST	discrete sine transform
DTP	discrete Tchebichef polynomial
DTT	discrete Tchebichef transform
DWT	discrete wavelet transform
FIF	fractal image format

GIF	graphics interchange format
HD	high definition
HVS	human visual system
ICT	integer cosine transform
IDTT	inverse discrete Tchebichef transform
ITT	integer Tchebichef transform
ITU-T	international telecommunication union-radiocommunication sector
JPEG	joint photographic experts group
JPEG-LS	lossless JPEG
KLT	Karhunen Loeve transform
LMSE	Laplacian mean squared error
LOT	lapped orthogonal transforms
LWF	Lurawave format
LZW	Lempel-Ziv-Welch
MD	maximum difference
MOS	mean opinion score
MPEG	moving picture experts group
NAE	normalized absolute error
NK	normalized cross-correlation

PQE	picture quality evaluation
PQS	picture quality scale
PSNR	peak signal to noise ratio
RLE	run length encoding
RMSE	root mean square error
ROI	region of interest
SAM	spectral activity measure
SC	structural content
SFM	spatial frequency measure
SGI	image silicon graphics
TIFF	tagged image file format
VLC	variable length coding
XBM	X bitmap

List of Symbols

δ	Kronecker delta
Δ	forward difference operator
$\Gamma(x)$	gamma function
∇	backward difference operator
$\varrho(x)$	jump function

$\tau_k(n)$	orthonormal discrete Tchebichef polynomial, where n and k are time and frequency indices respectively
τ	kernel of the discrete Tchebichef transform
$\hat{\tau}$	kernel of the integer Tchebichef transform
ζ, σ, λ	coefficients of the the difference equation of hypergeometric type
$a^{(n)}$	falling factorial
$a_{(n)}$	Pochammer symbol or rising factorial
d_k^2	squared norm
$F(i, j)$	(i, j)-th DFT coefficient of the image
$h_k^{(\alpha,\beta)}(x, N)$	kth Hahn polynomial orthogonal over $[0, N-1]$
I	identity matrix
M, L	numbers of pixels in an image in the horizontal and vertical directions respectively
$P(i, j)$	probability of the occurrence of the pixel (i, j)
r	Pearson product-moment
$t_k(x)$	kth discrete Tchebichef polynomial
w_i	magnitude of jump function at jumps defined by i
X'	transpose of X
$x'_{i,j}$	distorted/ impaired image
$x(n)$	1-D input data sequence

$x(n_1, n_2)$	2-D input matrix
$x \overset{DTT}{\longleftrightarrow} Y$	Tchebichef transform pair
$x_{i,j}$	original image
$Y(k)$	1-D transform coefficient in Tchebichef domain
$Y(k_1, k_2)$	2-D transform coefficient in Tchebichef domain
${}_3F_2$	generalized hypergeometric function
$\langle , \rangle$	inner product
$\odot$	element-to-element multiplication
$[,]$	closed interval
$\binom{n}{k}$	binomial coefficient or n choose k
$\sum$	summation
$\{\}$	a set

Chapter 1

Introduction

1.1 Background and Motivation

Signal processing techniques have assisted in delivering high end technology solutions that have revolutionized consumer and industry products. Discrete transforms have received much attention owing to their applications in various classes of problems such as feature extraction, image/ video/ speech enhancement, pattern recognition, adaptive techniques, watermarking and image/ video/ speech compression. Consequently, these areas harbor enormous research potential. Active research has been carried out in all the above mentioned areas, and new algorithms are constantly being developed, and in turn, are being used in the industry. These algorithms are used in a variety of applications such as data archival, surveillance and security applications, process automation based on audio or graphics content, image transmission on the internet, space sciences, geophysical, medical and entertainment applications. Some specific examples may be mentioned. *High definition coding* or HD technology, as it is popularly termed, uses advanced video coding techniques, and is now taking over the multimedia landscape. Image compression is central to any space mission, for instance, the Galileo deep space mission used the integer cosine transform for end-to-end image compression [1]. Medical procedures inevitably deal with

images. Hence, some common clinical applications are de-noising, compression and enhancement of medical images. *Closed circuit television* (CCTV) is presently a very popular surveillance technique, which also requires video compression and video processing techniques for further analysis of data. Biometrics, which is based on the concept of pattern recognition, is being widely used for security procedures.

Most of the above mentioned applications and techniques deal with bulky data, be it video, image, speech or text, during real-time or offline periods. Hence, compression plays a crucial role in storage and transmission of data. Techniques such as predictive coding and transform coding are typically used in image compression. The latter uses discrete transforms for spatial to frequency domain, followed by coding methods such as *Huffman coding*. The last few years have witnessed the interplay of signal processing, semiconductor circuits, wireless networks and embedded systems to provide viable and cutting-edge technologies that are truly state-of-the-art. The challenge lies in delivering practically realizable and economic solutions, while retaining the quality.

The *discrete Tchebichef polynomials* (DTPs) belong to the family of orthogonal polynomials, and have been widely used for data approximation, data fitting and spectral methods. Four kinds of Tchebichef polynomials have often been mentioned in the literature in the context of their relationship with the well-known trigonometric transforms like discrete Hartley transform (DHT) and discrete sine transform (DST). The transform kernel of the widely popular discrete cosine transform (DCT) originates from the trigonometric representation of the Tchebichef polynomials of the first kind. *Discrete Tchebichef transform* (DTT) has a polynomial kernel and maps a finite sequence of data to the DTP space. DTP is a special case of the Hahn polynomial, and can be related to other families of polynomials such as Krawtchouk, Meixner and Charlier polynomials. Mukundan et al. were the first authors to propose the application of polynomial-based orthogonal moments in image analysis [2]. The advantage of using Tchebichef polynomials over polynomials that have been

used in image analysis such as Legendre, Zernike, pseudo-Zernike [3], is that the former are *directly orthogonal* in the discrete domain and hence, do not result in approximation or discretization errors during transformation to the image coordinate space. Since DTPs satisfy exact orthogonality property in the discrete domain of the image coordinate space, an image can be exactly reconstructed from the complete set of transformed coefficients in a lossless manner. DTT exhibits many useful properties required of a transform in signal processing applications such as good energy compaction, near-optimal decorrelation, inherent properties of symmetry and orthogonality. Like DCT [4], DTT is a close approximation of Karhunen Loeve transform (KLT), and is not signal dependant. Furthermore, since DTT can be derived recursively from a polynomial recurrence relation, it is possible to transmute the computations into non-float operations. Therefore, DTT is expected to show good results when employed in an image compression scheme for transform coding.

The current video coding standard recommended by ITU-T[1] and the *moving picture experts group* (MPEG), H.264/ MPEG-4 AVC, employs a 4x4 integer cosine transform (ICT) due to its low complexity. Given the polynomial nature of the DTT, it is interesting to note that the transform by itself, easily conforms to an integer representation without the need for either an approximation, or intermediate scaling (by 2), contrary to that in the case of the ICT.

DCT has been considered to be the de-facto standard in image compression for quite sometime, and is continuing to be used for video compression. Apart from DCT, discrete wavelet transform (DWT) is being widely used in the JPEG 2000 standard. DCT has been explored to a great extent for applications in image compression. Quality assessment of DCT-compressed images has been investigated [5]. Various fast DCT algorithms [6–8] have also been developed. Most of the work carried out using discrete orthogonal polynomials in the literature is based on the moment approach, and not on a discrete transform

[1]International telecommunication union-radiocommunication sector

approach. Corr et al. proposed a discrete transform based on the continuous Chebyshev (Tchebichef) polynomials [9]. Nakagaki and Mukundan proposed DTT with DTPs as the kernel [10]. Since then, no work has been published on the DTT. Heretofore, DTT has not been tested on a complete and practical image compression scheme such as the standard popularly known as the *joint photographic experts group* (JPEG). Also, the performance evaluation of DTT has been carried out only via the rudimentary means of using the number of coefficients used to reconstruct the data (which is suitable for moment-based approaches, and not transform-based approaches). There is no evidence of any work that has been carried out to reduce the complexity involved in the computation of the transform. All the above observations necessitate a deeper investigation of the DTT, which is currently in a nascent stage.

1.2 Scope of the Thesis

From the discussion above, it is noted that DTT has a good potential in transform coding applications. Also, as discussed in the previous section, DTT has not been explored completely. The key points which need to be addressed are threefold: firstly, the need for application of DTT in an image compression scheme that is comparable to the available schemes, secondly, a detailed analysis of compression performance of DTT and thirdly, the design of a fast DTT algorithm with reduced complexity. Hence, in this thesis we concentrate on all the afore-mentioned points. We perform lossy image compression using DTT, wherein gray-scale images of block sizes 4×4 and 8×8 are transformed, quantized, arranged in the zigzag sequence and Huffman-coded. The performance evaluation of DTT-compressed images are carried out using standard objective measures as well as comprehensive picture quality assessment measures. We undertake the application of DTT in an image/ video compression scenario. We have attempted to uncover some interesting functional aspects of DTT, and also to mould it into a form suitable for a fast realization of

the transform.

1.3 Organization of the Thesis

The rest of the thesis is organized as follows. Chapter 2 is concerned with discrete Tchebichef polynomials, which is the foundation of this thesis. An introduction to orthogonal polynomials is given and their basic properties are described, followed by a description of Hahn polynomials and their properties. On the basis of the Hahn polynomial, DTP, which is a special case of the Hahn polynomial, is defined, and various properties of the DTPs are then derived from the Hahn polynomials. In the final section of the chapter, the orthonormal DTPs are presented.

Chapter 3 begins with an introduction of discrete Tchebichef transform and its importance in transform and image coding. We present the formal definitions of 1-D and 2-D DTTs that are transform-notation-friendly versions of the DTT originally proposed in [10]. We then derive some fundamental properties of DTT that are necessary for their use in the image compression application studied in this thesis. In the second half of the chapter, an introduction to fast algorithms is given, where we describe the need for integer transforms. In this chapter, we also consider a framework to construct integer transforms of sizes 4×4 and 8×8. The possibility of using 4×4 integer Tchebichef transform (ITT) for transform coding in H.264 is discussed by comparing the transform kernels of ICT and ITT. Later in the chapter, the importance of multiplier-free designs is explained, new multiplier-free algorithms are proposed and their complexities compared with the existing algorithms. A 5-Stage pipelined design is also proposed to illustrate the hardware implementation of 8-point DTT using the proposed algorithm.

In Chapter 4, we describe the fundamental concepts concerning images, and give a brief account of image compression, wherein we describe the steps involved in a general compression process, methods of compression and the compression scheme used in this

thesis. Experimental results of image compression for 4×4 and 8×8 DTTs are given. The performance of 4×4 DTT is compared with that of 4×4 DCT and 4×4 ICT. In this chapter, we also give an introduction to the picture quality evaluation (PQE) of images. We compare the performances of 8×8 DTT and 8×8 DCT in a comprehensive manner by using PQE in combination with the spatial and spectral properties of images, and carry out a detailed analysis of the results. Finally, in this chapter, we compare the performances of 4×4 and 8×8 DTTs.

Chapter 5 gives a summary of the work carried out in this thesis. Scope for future work is discussed and possible directions in this regard are pointed out.

Chapter 2

Discrete Orthogonal Polynomials

Classical orthogonal polynomials have been studied under the theory of special functions, which are usually given as solutions of the Sturm-Louisville problem, which is a linear second-order differential equation with variable coefficients under certain boundary conditions. Their extensions from the continuous case to the discrete case lead to polynomials that are the difference analogs of Jacobi, Laguerre, and Hermite polynomials on uniform and nonuniform lattices. They have been encountered in various fields of study such as mathematics, physics and chemistry, in problems pertaining to difference equations, quantum mechanics, genetics, group representation theory, coding theory, combinatorics and numerical analysis, the main reason for this being that many problems in these fields can be modeled using equations of the hypergeometric type. The discrete analogs are given by solutions of difference equations of hypergeometric type. Among the most popular classical orthogonal polynomials of a discrete variable are the Hahn, Meixner, Krawtchouk and Charlier polynomials. The discrete Tchebichef polynomials, a special case of the Hahn polynomials, are a discrete analog of the Chebyshev polynomials, which are a special case of the Jacobi polynomials. The Hahn, Tchebichef, Meixner, Krawtchouk and Charlier polynomials are orthogonal on the uniform lattice i.e., $\{x = 0, 1, 2, ...\}$. The dual Hahn, q-Hahn, q-Meixner, q-Krawtchouk and Racah polynomials are examples of polynomials

orthogonal on the non-uniform lattice i.e., $\{x = x(s), s = 0, 1, 2, ...\}$. Most of the above mentioned polynomials can be related to one another in the form of limiting cases.

2.1 System of Orthogonal Functions

The system of orthogonal functions [11] can be described as

$$\langle p_1, p_2 \rangle = \sum_i w_i p_1(x_i) p_2(x_i) \tag{2.1}$$

where w_i is the magnitude of the jump function at $x = x_i$, and $p_k(x)$ is a real function of a discrete variable. Two functions are said to be orthogonal if their scalar product vanishes. The functions $\{p_k(x)\}$ form an orthonormal system if

$$\langle p_h, p_k \rangle = \begin{cases} 0 & \text{for } h \neq k, \\ 1 & \text{for } h = k \end{cases} \tag{2.2}$$

An orthogonal system can be normalized by letting

$$\hat{p}_h(x) = \langle p_h, p_h \rangle^{-\frac{1}{2}} p_h(x) \tag{2.3}$$

Classical orthogonal polynomials of a single variable are eigenfunctions of a second order differential operator. The difference equation of hypergeometric type that has orthogonal polynomials as solutions is given by [12]

$$\sigma(x)\Delta\nabla y(x) + \zeta(x)\Delta y(x) + \lambda y(x) = 0 \tag{2.4}$$

where $\sigma(x)$ and $\zeta(x)$ are polynomials of atmost second and first degree respectively, λ is an appropriate constant and Δ and ∇ are the forward and backward difference operators[1] , respectively, given by $\Delta f(x) = f(x+1) - f(x)$ and $\nabla f(x) = f(x) - f(x-1)$.

[1] See also Appendix A for more properties

The polynomial solutions [12] of (2.4), under the boundary conditions

$$\sigma(x)\varrho(x)x^l\Big|_{x=a,b} = 0 \quad (l = 0, 1, ...) \tag{2.5}$$

are orthogonal on $[a, b-1]$ with respect to weight function $\varrho(x)$ [12]

$$\sum_{x_i=a}^{b-1} \varrho(x_i)y_m(x_i)y_k(x_i) = \delta_{mk}d_k^2 \tag{2.6}$$

In (2.4), when $\sigma(x)$ is a polynomial of the second degree, a particular solution $y = y_k(x)$, obtained by the Rodrigue's formula, is given by [12]

$$y_k(x) = \frac{B_k}{\varrho(x)}\nabla^k[\varrho_k(x)] \tag{2.7}$$

2.2 Hahn Polynomials

The Hahn polynomials are a discrete analog of the Jacobi polynomials, and have been used in expressing functions such as Clebsch-Gordan [13] coefficients in the modeling of quantum chemistry problems [14] and also in coding theory [15]. The polynomial solutions of (2.4) are determined by (2.7) upto the normalizing factor B_k. Hence, by using the finite difference analogue of the Rodrigue's formula [12], different classes of orthogonal polynomials of a discrete variable can be generated. In (2.7), for $B_k = \frac{(-1)^k}{k!}$ and the weight function $\varrho(x)$ defined by

$$\varrho(x) = \frac{\Gamma(N+\alpha-x)\Gamma(x+\beta+1)}{\Gamma(x+1)\Gamma(N-x)}, \quad (\alpha > -1, \beta > -1), \tag{2.8}$$

where $\Gamma(x)$ is the Gamma function [2], we obtain the Hahn polynomials denoted by $h_k^{(\alpha,\beta)}(x, N)$. The Hahn polynomials are solutions of (2.4) corresponding to [12]

$$\sigma(x) = x(\gamma_1 - x), \ \sigma(x) + \zeta(x) = (x+\gamma_2)(\gamma_3 - x), \tag{2.9}$$

[2] See also Appendix A for some common properties

where

$\gamma_1 = N + \alpha,\ \gamma_2 = \beta + 1,\ \gamma_3 = N - 1,$ for $\alpha > -1, \beta > -1$ are constants

so that

$\zeta(x) = (\beta + 1)(N - 1) - x(\alpha + \beta + 2)$ and $\sigma(x) = x(N + \alpha - x)$.

In addition, λ is given by $k(\alpha + \beta + k + 1)$. (2.10)

2.2.1 Properties of Hahn Polynomials

Orthogonality

When $y(x) = h_k(x)$, $a = 0$ and $b = N$ in (2.6), we have for the Hahn polynomials [12],

$$d_k^2 = \frac{\Gamma(\alpha + k + 1)\Gamma(\beta + k + 1)(\alpha + \beta + k + 1)_{(N)}}{(\alpha + \beta + 2k + 1)k!(N - k - 1)!} \tag{2.11}$$

where $(a)_{(k)}$ is the Pochammer symbol[3] given by $(a)_{(n)} = a(a + 1)(a + 2).....(a + n - 1)$. The Hahn polynomials are orthogonal over $[0, N - 1]$.

Symmetry

The weight function $\varrho(x)$ has the following symmetry

$$\varrho(x,\ \alpha,\ \beta) = \varrho(N - 1 - x,\ \beta,\ \alpha) \tag{2.12}$$

As a consequence of the above symmetry, it is seen from [12] that the Hahn polynomials satisfy the symmetry property

$$h_k^{(\alpha,\beta)}(N - 1 - x, N) = (-1)^k h_k^{(\beta,\alpha)}(x, N) \tag{2.13}$$

[3]See also Appendix A for more properties

Some Special Values

The end-point values for Hahn polynomials at $x = 0,\ N-1$ are given by [12]

$$h_k^{(\alpha,\beta)}(0,N) = (-1)^k \frac{(N-1)!}{(k)!(N-k-1)!} \frac{\Gamma(k+\beta+1)}{\Gamma(\beta+1)} \tag{2.14}$$

By using (2.13) in (2.14), we have

$$h_k^{(\alpha,\beta)}(N-1,N) = \frac{(N-1)!}{(k)!(N-k-1)!} \frac{\Gamma(k+\alpha+1)}{\Gamma(\alpha+1)} \tag{2.15}$$

Recurrence Relation

The Hahn polynomials satisfy a three term recurrence relation given by [12]

$$x h_k^{(\alpha,\beta)}(x,N) = \alpha_n h_{k+1}^{(\alpha,\beta)}(x,N) + \beta_n h_k^{(\alpha,\beta)}(x,N) + \gamma_n h_{k-1}^{(\alpha,\beta)}(x,N) \tag{2.16}$$

with

$$\begin{aligned}
\alpha_k &= \frac{(k+1)(\alpha+\beta+k+1)}{(\alpha+\beta+2k+1)(\alpha+\beta+2k+2)}, \\
\beta_k &= \frac{(\alpha-\beta+2N-2)}{4} + \frac{(\beta^2-\alpha^2)(\alpha+\beta+2N)}{4(\alpha+\beta+2k)(\alpha+\beta+2k+2)}, \\
\gamma_k &= \frac{(\alpha+k)(\beta+k)(\alpha+\beta+N+k)(N-k)}{(\alpha+\beta+2k)(\alpha+\beta+2k+1)}
\end{aligned}$$

2.2.2 Hypergeometric Representation of Hahn Polynomials

The Hahn polynomials can be expressed in terms of the generalized hypergeometric function [12], ${}_3F_2(x)$

$$h_k^{(\alpha,\beta)}(x,N) = \frac{(-1)^k}{k!}(N-k)_{(k)}(\beta+1)_{(k)}\ {}_3F_2\left(\left.\begin{matrix}-k, \alpha+\beta+k+1, -x \\ \beta+1, 1-N\end{matrix}\right| 1\right) \tag{2.17}$$

where the hypergeometric function ${}_3F_2$ is given by

$$ {}_3F_2\left(\left.\begin{matrix}k_1, k_2, k_3 \\ m_1, m_2\end{matrix}\right| z\right) = \sum_{k=0}^{\infty} \frac{(k_1)_k (k_2)_k (k_3)_k}{(m_1)_k (m_2)_k} \frac{z^k}{k!} \tag{2.18} $$

Another form of the above is [16]

$$ h_k^{(\alpha,\beta)}(x, N) = \frac{(-1)^k}{k!} (N - x - k)_{(k)} (\beta + x + 1)_{(k)} \, {}_3F_2\left(\left.\begin{matrix}-k, \alpha + N - x, -x \\ N - x - k, -\beta - x - k\end{matrix}\right| 1\right) \tag{2.19} $$

2.3 Discrete Tchebichef Polynomials: A Special Case of Hahn Polynomials

Tchebichef polynomials, which also go by the names of Chebyshev, Tchebicheff and Tchebychev, are named after Pafnuty Tchebychev, who is known to have introduced the classical Tchebichef polynomials. The discrete Tchebichef polynomials are associated with distributions of Stieltjes type [17] that were studied by Tchebichef. When the prefix 'discrete' is dropped, it is understood that reference is being made to the continuous class of Tchebichef polynomials, which are a special case of the Jacobi polynomials. The Tchebichef polynomial of a discrete variable, denoted by $t_k(x)$, is a special case of the Hahn polynomial, with $\alpha = 0$ and $\beta = 0$. DTP is defined as [11]

$$ t_k(x) = k! \Delta^k \left[\binom{x}{k} \binom{x - N}{k} \right] \qquad \text{for} \quad k = 0, 1 \ldots\ldots N - 1 \tag{2.20} $$

By substituting for $\alpha = 0$ and $\beta = 0$ in (2.11), we have the squared norm for DTPs

$$ \begin{aligned} d_k^2 &= \frac{\Gamma(k+1)\Gamma(k+1)(k+1)_{(N)}}{(2k+1)k!(N-k-1)!} = \frac{k!(k+1)_{(N)}}{(2k+1)(N-k-1)!} && \text{(using (A.10))} \\ &= \frac{k!\binom{k+1+N-1}{N} N!}{(2k+1)(N-k-1)!} && \text{(using (A.14))} \\ \Rightarrow d_k^2 &= \frac{(k+N)!}{(2k+1)(N-k-1)!} && \text{(using (A.4))} \end{aligned} \tag{2.21} $$

By substituting for $\alpha = 0$ and $\beta = 0$ in (2.10), we obtain the following values for the DTP $\gamma_1 = N, \quad \gamma_2 = 1, \quad \gamma_3 = N - 1$, so that $\zeta(x) = N - 1 - 2x$ and $\sigma(x) = x(N - x)$. Also, λ_k is given by $k(k+1)$. The above values describe the parameters for Tchebichef polynomials in (2.4). In the next few sub-sections, some properties of DTP are derived.

2.3.1 Properties of DTPs

Orthogonality

By substituting $\alpha = 0$ and $\beta = 0$ in (2.8), we see that the weight function for the Tchebichef polynomials is unity

$$\varrho(x) = \frac{\Gamma(N - x)\Gamma(x + 1)}{\Gamma(x + 1)\Gamma(N - x)} = 1 \tag{2.22}$$

Hence, DTPs are orthogonal with respect to a unit weight function. The general orthogonality condition in (2.6), with $a = 0$ and $b = N$ is rewritten for DTPs as

$$\sum_{x=0}^{N-1} t_k(x)t_m(x) = \delta_{mk}d_k^2 \tag{2.23}$$

where d_k^2 is the squared norm of DTPs.

Symmetry

By substituting $\alpha = \beta = 0$ in (2.13), we obtain the symmetry property for the DTP

$$t_k(x) = (-1)^k t_k(N - 1 - x) \tag{2.24}$$

Difference Equation

The difference equation [11] for DTPs is given by

$$(x + 2)(x - N + 2)\Delta^2 t_k(x) + [2x - N + 3 - k(k + 1)]\Delta t_k(x) - k(k + 1)t_k(x) = 0 \tag{2.25}$$

$$\text{for } k = 1, 2 \ldots N - 1$$

Special Values

By substituting $\alpha = \beta = 0$ in (2.14), we have

$$\begin{aligned} t_k(0) &= (-1)^k \frac{(N-1)!}{(k)!(N-k-1)!} \frac{\Gamma(k+1)}{\Gamma(1)} \\ &= (-1)^k \frac{(N-1)!}{(N-k-1)!} && \text{(using (A.9) and (A.10))} \\ &= (-1)^k \binom{N-1}{k} k! && \text{(using (A.14))} \\ &= (-1)^k (N-1)^{(k)} && \text{(using (A.16))} \\ \Rightarrow t_k(0) &= (1-N)_{(k)} && \text{(using (A.15))} \end{aligned} \tag{2.26}$$

Similarly, by substituting $\alpha = \beta = 0$ in (2.15), we have

$$\begin{aligned} t_k(N-1) &= \frac{(N-1)!}{(N-k-1)!} \\ \Rightarrow t_k(N-1) &= (-1)^k (1-N)_{(k)} \end{aligned} \tag{2.27}$$

Central Values

The central values for the Tchebichef polynomial [11] are

$$t_{2k}\left(\frac{N-1}{2}\right) = (-1)^k (2k)! \binom{2k}{k} \binom{\frac{N-1}{2}+k}{k} \tag{2.28}$$

$$t_{2k+1}\left(\frac{N-1}{2}\right) = 0 \tag{2.29}$$

Recurrence Relations

(a) **Recurrence Relation in x**: The recurrence relation in x for DTPs is derived here. The difference equation in (2.25) can be re-written using (A.1) as

$$(x+2)(x-N+2)\left[t_k(x+2) - 2t_k(x+1) + t_k(x)\right] + \left[2x - N + 3 - k(k+1)\right]$$
$$\left[t_k(x+1) - t_k(x)\right] - k(k+1)t_k(x) = 0$$

In the above equation, we put x as $x-2$, and we re-group the terms to obtain

$$-2x(x-N)t_k(x-1)+x(x-N)t_k(x-2)+[2x-1-k-N(k+1)]t_k(x-1)$$
$$-[2x-1-N-k(k+1)]\,t_k(x-2)-k(k+1)t_k(x-2)+[x(x-N)]\,t_k(x)=0$$

On simplifying, we have the recurrence relation with respect to x

$$x(x-N)t_k(x)+[-2x^2+2kN+2x-N-k(k+1)-1]t_k(x-1)$$
$$+[x^2-kN-2x+1+N]t_k(x-2)=0 \tag{2.30}$$

(b) **Recurrence Relation in k**: By substituting for $\alpha=\beta=0$ in (2.16), we have

$$xt_k(x)=\frac{(k+1)(k+1)}{(2k+1)(2k+2)}t_{k+1}(x)+\frac{(2N-2)}{4}t_k(x)+\frac{k^2(N+k)(N-k)}{2k(2k+1)}t_{k-1}(x),$$

which can be rewritten as

$$\frac{(k+1)}{2(2k+1)}t_{k+1}(x)+\frac{(N-1)-2x}{2}t_k(x)+\frac{k(N^2-k^2)}{2(2k+1)}t_{k-1}(x)=0$$

which is simplified as

$$(k+1)t_{k+1}(x)+(N-1-2x)(2k+1)t_k(x)+k(N^2-k^2)t_{k-1}(x)=0$$

Hence, DTPs satisfy a three-term recurrence relation given by

$$(k+1)t_{k+1}(x)-(2k+1)t_1(x)t_k(x)+k(N^2-k^2)t_{k-1}(x)=0 \tag{2.31}$$

with $t_0(x)=1$ and $t_1(x)=2x-N+1$.

2.3.2 Hypergeometric Representation

The Tchebichef polynomials can be expressed in terms of the generalized hypergeometric function, ${}_3F_2(x)$. By substituting for $\alpha = \beta = 0$ in (2.14), we have

$$\begin{aligned} t_k(x) &= \frac{(-1)^k}{k!}(N-k)_{(k)}(1)_{(k)}{}_3F_2\left(\left.\begin{matrix}-k, k+1, -x \\ 1, 1-N\end{matrix}\right| 1\right) \\ &= (-1)^k(N-k)_{(k)}{}_3F_2\left(\left.\begin{matrix}-k, k+1, -x \\ 1, 1-N\end{matrix}\right| 1\right) \qquad \text{(using (A.13))} \end{aligned} \tag{2.32}$$

2.3.3 Direct Representation

By using (2.18) to expand the expression in (2.32) as a partial sum,

$$\begin{aligned} t_k(x) &= (-1)^k(N-k)_{(k)}\sum_{j=0}^{k}\frac{(-k)_{(j)}(k+1)_{(j)}(-x)_{(j)}}{(1)_{(j)}(1-N)_{(j)}j!} \\ &= (-1)^k(N-k)_{(k)}\sum_{j=0}^{k}\frac{(-1)^j(k)^{(j)}(k+1)_{(j)}(-1)^j(x)^{(j)}}{j!(-1)^j(N-1)^{(j)}j!} \qquad \text{(using (A.15))} \\ &= \sum_{j=0}^{k}\frac{(-1)^{(k-j)}\binom{k}{j}(k+1)_{(j)}\binom{x}{j}(N-k)_{(k)}}{\binom{N-1}{j}j!} \qquad \text{(using (A.16))} \\ &= \sum_{j=0}^{k}(-1)^{(k-j)}\binom{x}{j}\binom{k}{j}\binom{k+j}{j}k!\frac{\binom{N-1}{k}}{\binom{N-1}{j}} \qquad \text{(using (A.14))} \\ t_k(x) &= \sum_{j=0}^{k}(-1)^{(k-j)}\binom{x}{j}\frac{(k+j)!}{j!}\binom{N-1-j}{k-j} \qquad \text{(using (A.4))} \end{aligned} \tag{2.33}$$

or

$$t_k(x) = k!\sum_{j=0}^{k}(-1)^{(k-j)}\binom{x}{j}\binom{k+j}{k}\binom{N-1-i}{k-j}$$

2.3.4 Normalization of DTP

From (2.23), it is evident that $\{t_k(x)\}$ is orthogonal, but not orthonormal. In view of (2.2), (2.3) and (2.23), orthonormal DTP, denoted by $\tau_k(x)$, is given by

$$\tau_k(x) = \frac{t_k(x)}{d_k} \tag{2.34}$$

where d_k^2 is given by (2.21). Note that the classical DTP is denoted by lowercase $t_k(x)$ and the orthonormal DTP by $\tau_k(x)$. Since $\forall k, m : \langle \tau_k, \tau_m \rangle = \delta_{k,m}$, it can be seen from (2.2) that $\{\tau_k(x)\}$ form an orthonormal set. The orthonormal DTP can be evaluated by using $\tau_k(x)$ in (2.31). Henceforth in this thesis, the orthonormal DTP is used for all purposes.

Note: For the purpose of transform notation explained in the next chapter, we abuse the notation of $\tau_k(x)$ used in this chapter. From here on, in this thesis, we denote the orthonormal DTP as $\tau_k(n)$, since n is more commonly used over x to indicate discrete-time notation. In $\tau_k(n)$, n and k are the indices of the time and frequency components respectively.

The recurrence relation (2.31) may be rewritten for orthonormal DTPs in the form [10]

$$\tau_k(n) = (a_1 n + a_2)\tau_{k-1}(n) + a_3 \tau_{k-2}(n) \tag{2.35}$$

with

$$\tau_0(n) = \frac{1}{\sqrt{N}}, \quad \tau_1(n) = (2n + 1 - N)\sqrt{\frac{3}{N(N^2-1)}}$$

where

$$a_1 = \frac{2}{k}\sqrt{\frac{4k^2-1}{N^2-k^2}}, \quad a_2 = \frac{1-N}{k}\sqrt{\frac{4k^2-1}{N^2-k^2}}$$

and

$$a_3 = \frac{1-k}{k}\frac{2k+1}{2k-3}\sqrt{\frac{N^2-(k-1)^2}{N^2-k^2}}$$

Chapter 3

Discrete Tchebichef Transform, Integer Tchebichef Transform and Fast Algorithms

3.1 Introduction

Orthogonal transforms have been vastly explored. DCT, DFT, DWT, KLT, DHT and DST have been used in a variety of applications such as image compression, pattern recognition, video processing, feature extraction and speech processing, depending on how beneficial the *transform kernel* is to the said application. The fundamental rationale behind using transforms is as follows:

1. A signal presents itself in a very useful manner in the transform domain. Hence, information that may not be visible in the spatial domain is very often distinctly evident in the frequency domain.

2. Typical time domain implementations are not very efficient from the point of view of system complexity.

3. Orthogonal transforms decorrelate the original signal and also have a good energy compaction property, both of which aid in the compression of signals without losing vital information, since each coefficient can be treated independently. This is more clearly evident from Fig. 3.1, where (a) is the original image of *Lena*, (b) is the transformed image of *Lena* and (c) represents the distribution of the power of the Tchebichef transform coefficients in every block (divided into blocks of 16×16 for the image *Lena*, whose dimensions are 128×128). From these figures, it is seen that in the transform domain of the DTT, most of the energy tends to be concentrated in the low frequency regions (top left). This property is exploited for compression methods in quantization and coding stages.

3.1.1 Transform Coding

Transform coding is an important operation in image and video processing applications. Transform coding is used to convert spatial image pixel values to coefficient values pertaining to the transform domain, followed by quantization and entropy coding. Transform coding relies on the fact that pixels in an image are correlated with their neighboring pixels. In the DTT domain, an input image can be represented by fewer coefficients, since the image data along the rows and the columns are highly correlated. This means that the transform-coded image can be reconstructed without a drastic loss in information.

3.1.2 Orthogonal Polynomials in Image Coding

An image can be represented in the transform domain by a linear combination of the elements of the orthogonal basis. Polynomials have been used to code images on rectangular [3] and polar coordinates [18]. The application of orthogonal polynomials, especially Tchebichef polynomials, in image analysis can also be seen to follow naturally from the fact that these polynomials have been widely used in data fitting and approximation theory,

and are known to be good descriptors of real-world functions. Also, since these polynomials are directly orthogonal on the discrete domain, discretization errors do not occur. In some cases, image normalization can also be skipped.

3.2 Discrete Tchebichef Transform

Discrete Tchebichef transform is based on a polynomial kernel derived from discrete Tchebichef polynomials. By performing DTT on an image, we transform the pixel intensity values in the spatial domain to the frequency domain. From the standpoint of digital signal processing applications, the importance of DTT is evident. DTT and DCT share many common characteristics such as high energy compaction, near optimal decorrelation and computational tractability. Due to these properties, DTT is useful for transform operations in image and video processing applications like feature extraction, image compression and video coding.

3.2.1 Formulation of DTT

Before we formulate the DTT, for the sake of convenience of notation, we introduce $\tau(k,n)$ to replace $\tau_k(n)$ in (2.35). With this notation, we can define the matrix τ to be a matrix of k rows and n columns, a typical element of the matrix being $\tau(k,n)$. Now we proceed to formulate the DTT.

The DTP in (2.34) constitute an orthonormal basis in the interval $[0, N-1]$. In the DTT domain, an input data sequence $x(n)$ is given as

$$Y(k) = \sum_{n=0}^{N-1} \tau(k,n)x(n) \qquad \text{for } k,n = 0, 1 \cdots N-1 \tag{3.1}$$

In the above equation, the kernel $\tau(k,n)$ represents the orthogonal basis of DTP, where n and k are the indices of the time and frequency components respectively, corresponding to

the kth DTP evaluated at x in (2.34). The input data sequence is recovered by applying the *inverse DTT* (IDTT) defined as

$$x(n) = \sum_{k=0}^{N-1} \tau(k,n)Y(k) \qquad \text{for } k,n = 0,1\cdots N-1 \tag{3.2}$$

When we apply DTT to an N-point vector, $x(n)$, the input vector is decomposed into a linear combination of the basis functions of the transform kernel, $\tau(k,n)$. Here, n and k represent the time and frequency domain indices respectively. For $N = 8$, the plot of the Tchebichef basis functions are given in Fig. 3.2.

3.2.2 Properties of DTT

In addition to the properties inherited from the DTP explained in Section 2.3, DTT exhibits some interesting properties that are key factors in transform coding operations. Some properties also assist in scaling down the complexity involved in the computation of the transform. These properties are derived below.

Symmetry

As already explained in (2.24) of the previous chapter, DTP, which constitutes the kernel of the DTT exhibits the symmetry property

$$\tau(k,n) = (-1)^k \tau(k,N-1-n) \tag{3.3}$$

By using this property, the number of computations required for DTT can be reduced by one half.

Other useful relations for DTT including the orthogonality property given by (2.23) follow from the properties of DTP derived in the previous chapter.

Linearity

Theorem 3.1 (Linearity Property of DTT) *The DTT operation is linear.*

$$a_1 x_1(n) + a_2 x_2(n) \overset{DTT}{\longleftrightarrow} a_1 Y_1(k) + a_2 Y_2(k)$$

Proof: Let $x_1(n) \overset{DTT}{\longleftrightarrow} Y_1(k)$ and $x_2(n) \overset{DTT}{\longleftrightarrow} Y_2(k)$, and also, let $x(n) = a_1 x_1(n) + a_2 x_2(n)$ and $x(n) \overset{DTT}{\longleftrightarrow} Y(k)$.

From (3.1), we have

$$\begin{aligned} Y(k) &= \sum_{n=0}^{N-1} \tau(k,n) x(n) \\ &= \sum_{n=0}^{N-1} \tau(k,n) \{a_1 x_1(n) + a_2 x_2(n)\} \\ &= a_1 \sum_{n=0}^{N-1} \tau(k,n) x_1(n) + a_2 \sum_{n=0}^{N-1} \tau(k,n) x_2(n) \\ &= a_1 Y_1(k) + a_2 Y_2(k) \qquad \text{(using (3.2))} \end{aligned}$$

Energy Property

A variant of Plancherel's theorem, which corresponds to Parseval's theorem for Fourier series is given below.

Theorem 3.2 (Energy Property of DTT) *The summation of the squared modulus of a function is equal to the summation of the squared modulus of its spectrum i.e.,*

$$\sum_{n=0}^{N-1} |x(n)|^2 = \sum_{k=0}^{N-1} |Y(k)|^2$$

Proof: From (3.2),

$$\begin{aligned}\sum_{n=0}^{N-1}|x(n)|^2 &= \sum_{n=0}^{N-1}x(n)\left\{\sum_{k=0}^{N-1}\tau(k,n)Y(k)\right\}\\ &= \sum_{k=0}^{N-1}\sum_{n=0}^{N-1}\{x(n)\tau(k,n)\}\,Y(k)\\ &= \sum_{k=0}^{N-1}Y(k)Y(k) \qquad \text{(using (3.1))}\\ &= \sum_{k=0}^{N-1}|Y(k)|^2\end{aligned}$$

This implies that the information is preserved in the transform domain, in the sense that the transformed signal can be recovered completely.

Separability of DTT

DTT follows the separability property

$$\begin{aligned}Y(k_1,k_2) &= \sum_{n_1=0}^{N-1}\tau(k_1,n_1)\left\{\sum_{n_2=0}^{M-1}\tau(k_2,n_2)x(n_1,n_2)\right\}\\ &= \sum_{n_1=0}^{N-1}\tau(k_1,n_1)Y'(k_2,n_1) \qquad (3.4)\end{aligned}$$

where $Y'(k_2,n_1) = \sum_{n_2=0}^{M-1}\tau(k_2,n_2)x(n_1,n_2)$. This property is useful in applications such as image coding, where a 2-D transform needs to be performed. The 2-D DTT can be carried out just as we would perform a 1-D DTT, first along the rows and then along the columns.

Unitary Property

DTT operates as a unitary transform on matrices, i.e., τ is a unitary matrix. This follows from (2.34), since

$$\tau'\tau = \tau\tau' = \mathbf{I}, \qquad (3.5)$$

where **I** is the identity matrix.

3.2.3 2-D Discrete Tchebichef Transform

For a 2-D input sequence $x(n_1, n_2)$, the *2-D DTT of order* $N \times M$ is defined as

$$Y(k_1, k_2) = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{M-1} \tau(k_1, n_1)\tau(k_2, n_2)x(n_1, n_2) \tag{3.6}$$

$$\text{for } k_1, n_1 = 0, 1 \cdots N-1 \text{ and } k_2, n_2 = 0, 1 \cdots M-1$$

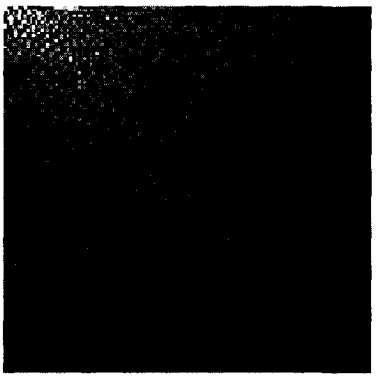
2-D transformation is achieved by either using the separability property of the DTT in (3.4) to perform a *row-column transform*, or by direct application of the 2-D transform kernel.

The inverse 2-D DTT applied to the transform coefficients restores the input data

$$x(n_1, n_2) = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{M-1} \tau(k_1, n_1)\tau(k_2, n_2)Y(k_1, k_2) \tag{3.7}$$

$$\text{for } k_1, n_1 = 0, 1 \cdots N-1 \text{ and } k_2, n_2 = 0, 1 \cdots M-1$$

Fig. 3.3 gives a 3-D visualization of the 2-D basis functions for $N = 4$. Fig. 3.4 gives the plot of the 2-D basis functions for $N = 8$, where gray represent zeroes, black represents negative and white represents positive amplitudes of the basis functions. In this figure, the top left coefficient is called the *DC component*, and the rest of the coefficients are called the *AC components*.



(a) **(b)**

69.0db	37.2db	29.2db	19.1db	13.7db	8.0db	5.9db	−3.6db
29.4db	23.9db	19.5db	13.6db	8.1db	5.4db	0.7db	−7.4db
12.9db	15.1db	14.2db	6.3db	6.5db	1.8db	−2.3db	−5.8db
6.6db	5.4db	6.3db	4.3db	2.8db	3.5db	−1.5db	−8.0db
−2.7db	0.9db	−0.4db	−0.3db	0.2db	−2.2db	−5.0db	−13.9db
−6.7db	−8.3db	−2.5db	−5.9db	−5.2db	−7.3db	−6.4db	−10.5db
−14.0db	−10.9db	−12.0db	−3.2db	−6.7db	−10.7db	−12.0db	−17.4db
−18.8db	−18.1db	−20.2db	−14.7db	−11.4db	−19.0db	−17.0db	−21.5db

(c)

Figure 3.1: Energy Distribution of Transformed Image Coefficients in the Tchebichef Domain (a) Original Image of *Lena* (b) Tchebichef Transformed Image (c) Power of the Tchebichef Transformed Coefficients

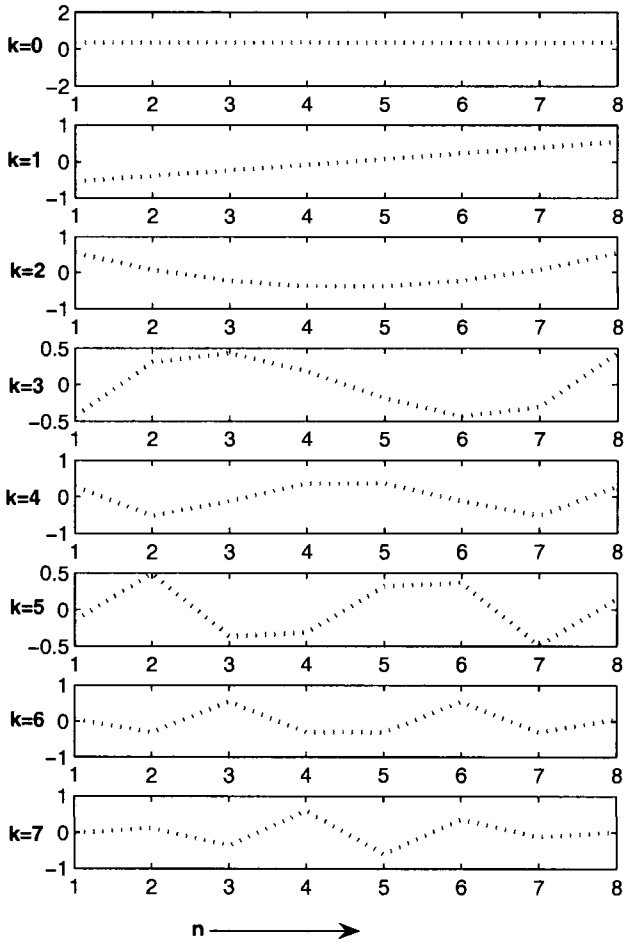


Figure 3.2: Plot of 1-D Tchebichef Basis Functions for $N = 8$

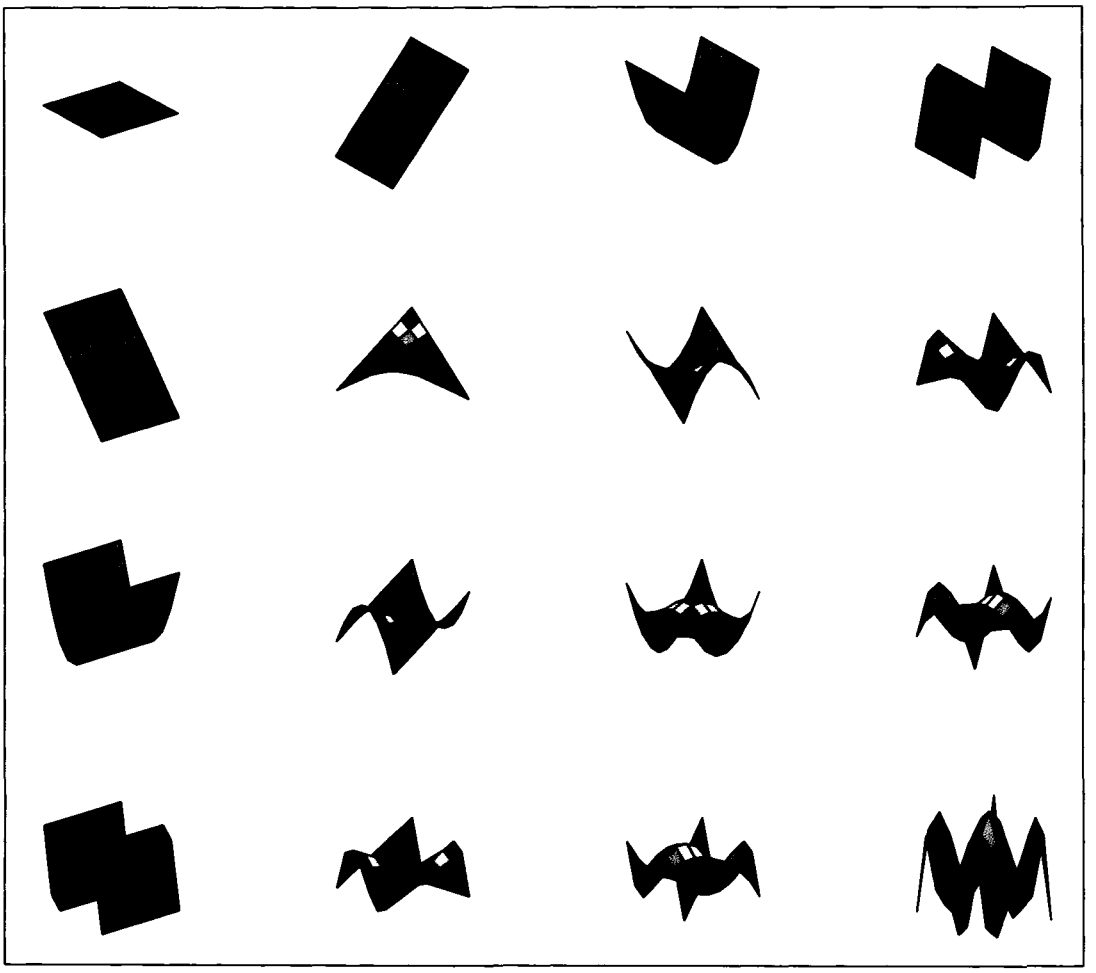


Figure 3.3: 2-D Tchebichef Basis Function Plot for N=4

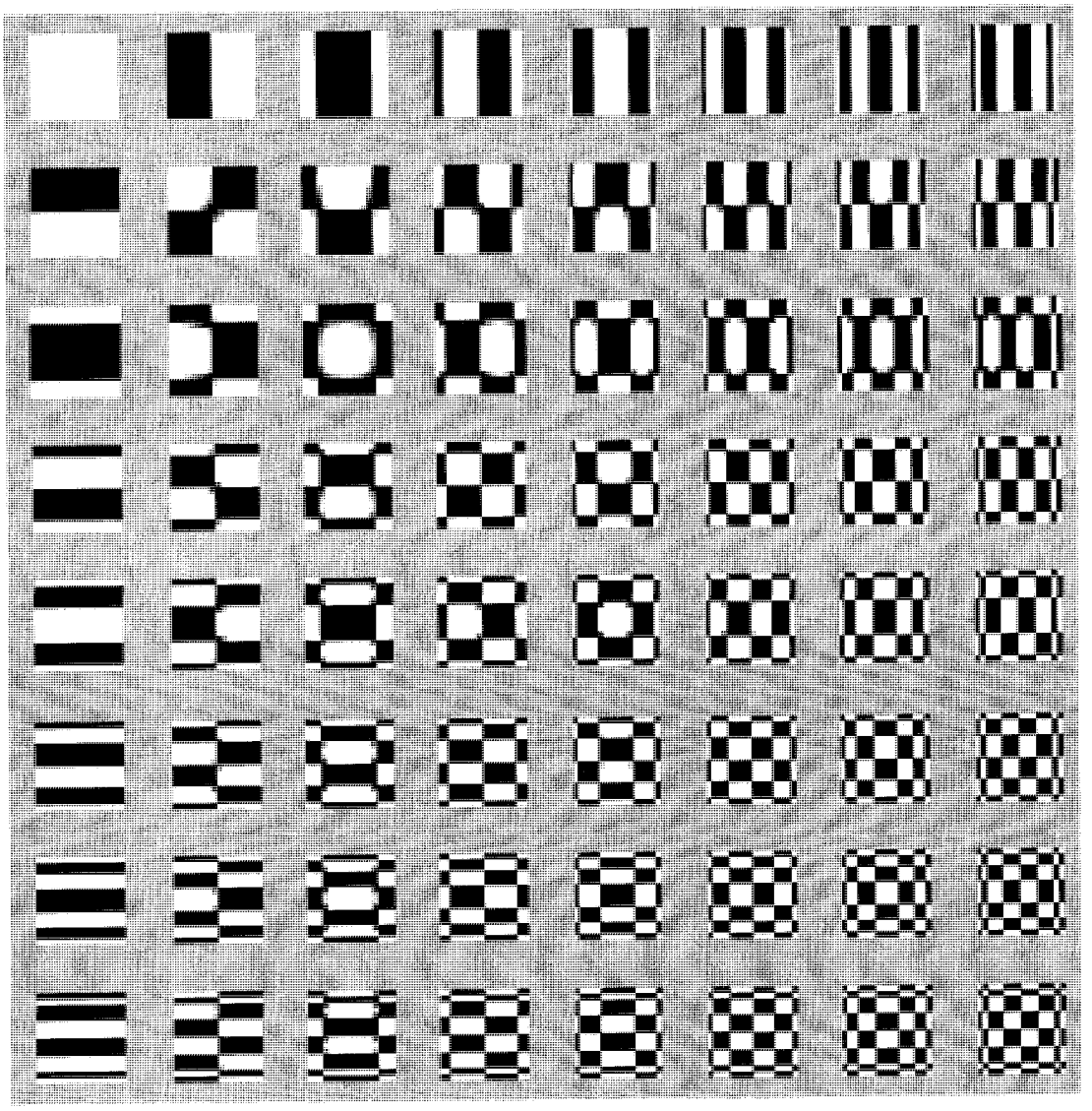


Figure 3.4: 2-D Tchebichef Basis Function Plot for N=8

3.3 Integer Tchebichef Transform for the Design of Fast Multiplier-free Algorithms

Transform coding is an important stage in many coding standards such as JPEG, H.264 and MPEG. These standards are used in image, audio and video codecs. Efficient designs with limited utilization of resources are well-suited for the implementation of transforms in hardware. Continuing cost improvements provide impetus to the design of fast transforms. Transform domain conversion accounts for over 60% of the total encoding time. From (3.1), it is seen that by using a brute-force method, the number of computations involved per 1-D DTT are N^2 multiplications and $N(N-1)$ additions.

The transform computation approach used in [10] to design a fast algorithm for 4×4 DTT was to consider all uniquely occurring DTT coefficients to form a new basis and express the remaining coefficients as a linear combination of these coefficients. This is the only fast algorithm available for DTT in the literature. In the Sub-section 3.3.4, fast algorithms for 8×8 and 4×4 DTT are proposed, which are computationally less-intensive.

Fixed-point operations are hardware-friendly, but rounding leads to loss of accuracy due to the lack of more significant digits. Floating-point operations preserve the full-precision value, but at the cost of increased resource consumption. Moreover, floating-point operations are not defined in all processors. Integer operations have many advantages over floating point operations such as increased computational speed, low complexity, low cost and low power consumption. Hence, in the following sub-section, a framework is proposed to construct an *integer Tchebichef transform*. Using this framework, 8×8 [19] and 4×4 [20] ITTs are proposed.

3.3.1 Integer Tchebichef Transform for 8x8 DTT

Due to the unitary property of DTT given in (3.5), the 2-D DTT of the input data, **X**, of size 8×8 given by (3.6), can also be expressed as

$$\mathbf{Y} = \tau \mathbf{X} \tau' \tag{3.8}$$

where τ, the 8×8 transform kernel for $N = 8$, is given by

$$\tau = \begin{pmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ -0.5401 & -0.3858 & -0.2315 & -0.0772 & 0.0772 & 0.2315 & 0.3858 & 0.5401 \\ 0.5401 & 0.0772 & -0.2315 & -0.3858 & -0.3858 & -0.2315 & 0.0772 & 0.5401 \\ -0.4308 & 0.3077 & 0.4308 & 0.1846 & -0.1846 & -0.4308 & -0.3077 & 0.4308 \\ 0.2820 & -0.5238 & -0.1209 & 0.3626 & 0.3626 & -0.1209 & -0.5238 & 0.2820 \\ -0.1498 & 0.4922 & -0.3638 & -0.3210 & 0.3210 & 0.3638 & -0.4922 & 0.1498 \\ 0.0615 & -0.3077 & 0.5539 & -0.3077 & -0.3077 & 0.5539 & -0.3077 & 0.0615 \\ -0.0171 & 0.1195 & -0.3585 & 0.5974 & -0.5974 & 0.3585 & -0.1195 & 0.0171 \end{pmatrix} \tag{3.9}$$

The transform equation in (3.9) is now factorized to separate the integer and float numbers. Let

$$q(k) = \frac{p(k)}{d_k}, \quad k = 0.....7 \tag{3.10}$$

where

$$p(k) = \begin{cases} k!\,k & \text{for } k = 1 \\ k!(k+1) & \text{for } k = 0,\ 2,\ 4,\ 6 \\ k!(N-k) & \text{for } k = 3,\ 5,\ 7 \end{cases}$$

and d_k^2 is the squared norm of DTPs given by (2.21).

Let **Q** be a diagonal matrix with $q(k)$ as the diagonal element of the kth row.

Therefore,

$$\mathbf{Q} = \begin{pmatrix} 0.3536 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0772 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0772 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0615 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0403 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0214 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0615 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0171 \end{pmatrix}$$

We define

$$\tau = \mathbf{Q}\hat{\tau} \tag{3.11}$$

Hence,

$$\hat{\tau} = \mathbf{Q}^{-1}\tau = \mathbf{Q_i}\tau \tag{3.12}$$

where, $\mathbf{Q_i}$ is the diagonal matrix which contains the reciprocal of the corresponding diagonal elements of $\mathbf{Q}$ as its diagonal elements and is given by

$$\mathbf{Q_i} = \begin{pmatrix} 1/q_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/q_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/q_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/q_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/q_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/q_5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/q_6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/q_7 \end{pmatrix}$$

or

$$\mathbf{Q_i} = \begin{pmatrix} 2.8284 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12.9615 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12.9615 & 0 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 16.2481 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 24.8193 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 46.7333 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 16.2481 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 58.5833 \end{pmatrix}$$

Therefore,

$$\hat{\tau} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -7 & -5 & -3 & -1 & 1 & 3 & 5 & 7 \\ 7 & 1 & -3 & -5 & -5 & -3 & 1 & 7 \\ -7 & 5 & 7 & 3 & -3 & -7 & -5 & 7 \\ 7 & -13 & -3 & 9 & 9 & -3 & -13 & 7 \\ -7 & 23 & -17 & -15 & 15 & 17 & -23 & 7 \\ 1 & -5 & 9 & -5 & -5 & 9 & -5 & 1 \\ 1 & -7 & 21 & -35 & 35 & -21 & 7 & -1 \end{pmatrix} \tag{3.13}$$

By substituting for τ from (3.11) in (3.8), we have

$$\begin{aligned} \mathbf{Y} &= (\mathbf{Q}\hat{\tau})\mathbf{X}(\mathbf{Q}\hat{\tau})' \\ &= \mathbf{Q}(\hat{\tau}\mathbf{X}\hat{\tau}')\mathbf{Q}' \qquad \text{(using matrix properties } (\mathbf{AB})' = \mathbf{B}'\mathbf{A}' \text{ and } \mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}) \\ &= (\hat{\tau}\mathbf{X}\hat{\tau}') \odot \hat{\mathbf{Q}} \qquad \text{(using Theorem B.1)} \end{aligned}$$

where $\odot$ represents *element-by-element multiplication.* Consequently, the transform equation of (3.8) may be rewritten as

$$\mathbf{Y} = (\hat{\tau}\mathbf{X}\hat{\tau}') \odot \hat{\mathbf{Q}} \tag{3.14}$$

where for $N = 8$, $\hat{\mathbf{Q}}$ is given by

$$\hat{\mathbf{Q}} = \begin{pmatrix} q_0^2 & q_0q_1 & q_0q_2 & q_0q_3 & q_0q_4 & q_0q_5 & q_0q_6 & q_0q_7 \\ q_0q_1 & q_1^2 & q_1q_2 & q_1q_3 & q_1q_4 & q_1q_5 & q_1q_6 & q_1q_7 \\ q_0q_2 & q_2q_1 & q_2^2 & q_2q_3 & q_2q_4 & q_2q_5 & q_2q_6 & q_2q_7 \\ q_0q_3 & q_3q_1 & q_3q_2 & q_3^2 & q_3q_4 & q_3q_5 & q_3q_6 & q_3q_7 \\ q_0q_4 & q_4q_1 & q_4q_2 & q_4q_3 & q_4^2 & q_4q_5 & q_4q_6 & q_4q_7 \\ q_0q_5 & q_5q_1 & q_5q_2 & q_5q_3 & q_5q_4 & q_5^2 & q_5q_6 & q_5q_7 \\ q_0q_6 & q_6q_1 & q_6q_2 & q_6q_3 & q_6q_4 & q_6q_5 & q_6^2 & q_6q_7 \\ q_0q_7 & q_7q_1 & q_7q_2 & q_7q_3 & q_7q_4 & q_7q_5 & q_7q_6 & q_7^2 \end{pmatrix} \tag{3.15}$$

or

$$\hat{\mathbf{Q}} = \begin{pmatrix} 0.1250 & 0.0273 & 0.0273 & 0.0218 & 0.0142 & 0.0076 & 0.0218 & 0.0060 \\ 0.0273 & 0.0060 & 0.0060 & 0.0047 & 0.0031 & 0.0017 & 0.0047 & 0.0013 \\ 0.0273 & 0.0060 & 0.0060 & 0.0047 & 0.0031 & 0.0017 & 0.0047 & 0.0013 \\ 0.0218 & 0.0047 & 0.0047 & 0.0038 & 0.0025 & 0.0013 & 0.0038 & 0.0011 \\ 0.0142 & 0.0031 & 0.0031 & 0.0025 & 0.0016 & 0.0009 & 0.0025 & 0.0007 \\ 0.0076 & 0.0017 & 0.0017 & 0.0013 & 0.0009 & 0.0005 & 0.0013 & 0.0004 \\ 0.0218 & 0.0047 & 0.0047 & 0.0038 & 0.0025 & 0.0013 & 0.0038 & 0.0011 \\ 0.0060 & 0.0013 & 0.0013 & 0.0011 & 0.0007 & 0.0004 & 0.0011 & 0.0003 \end{pmatrix}$$

(3.15) is seen to follow from (B.1) of Theorem B.1.

In (3.14), $\hat{\mathbf{Y}}$ is the 8×8 ITT given by

$$\hat{\mathbf{Y}} = \hat{\tau}\mathbf{X}\hat{\tau}' \tag{3.16}$$

The normalization is complete after element-by-element multiplication is carried out between the core transform, $(\hat{\tau}\mathbf{X}\hat{\tau}')$ and $\hat{\mathbf{Q}}$. This operation is incorporated into the quantization stage. The same framework applies for 4×4 ITT also as shown in the next sub-section.

3.3.2 Integer Tchebichef Transform for 4x4 DTT

Using the framework described in Sub-section 3.3.1, we now derive the 4×4 ITT. The 2-D DTT of the input data, $\mathbf{X}$, of size 4×4 given by (3.6), can also be expressed as

$$\mathbf{Y} = \tau\mathbf{X}\tau' \tag{3.17}$$

where τ, the 4×4 transform kernel for $N = 4$, obtained from (2.35), is given by

$$\tau = \begin{pmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ -0.6708 & -0.2236 & 0.2236 & 0.6708 \\ 0.5000 & -0.5000 & -0.5000 & 0.5000 \\ -0.2236 & 0.6708 & -0.6708 & 0.2236 \end{pmatrix} \tag{3.18}$$

The transform equation in (3.18) is now factorized to separate the integer and float numbers. Let

$$q(k) = \frac{p(k)}{d_k}, \quad k = 0.....3 \tag{3.19}$$

where

$$p(k) = \begin{cases} k!\,k & \text{for } k = 1 \\ k!(k+1) & \text{for } k = 0,\ 2 \\ k!(N-k) & \text{for } k = 3 \end{cases}$$

and d_k^2 is the squared norm of DTPs given by (2.21). Let $\mathbf{Q}$ be a diagonal matrix with $q(k)$ as the diagonal element of the kth row.

Therefore,

$$\mathbf{Q} = \begin{pmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.2236 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.2236 \end{pmatrix}$$

We define

$$\tau = \mathbf{Q}\hat{\tau} \tag{3.20}$$

Hence,

$$\hat{\tau} = \mathbf{Q}^{-1}\tau = \mathbf{Q}_{\mathbf{i}}\tau \tag{3.21}$$

where, $\mathbf{Q_i}$ is the diagonal matrix, which contains the reciprocal of the corresponding diagonal elements of $\mathbf{Q}$ as its diagonal elements, and is given by

$$\mathbf{Q}_{\mathbf{i}} = \begin{pmatrix} 1/q_0 & 0 & 0 & 0 \\ 0 & 1/q_1 & 0 & 0 \\ 0 & 0 & 1/q_2 & 0 \\ 0 & 0 & 0 & 1/q_3 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 4.4721 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 4.4721 \end{pmatrix}$$

Therefore,

$$\hat{\tau} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -3 & -1 & 1 & 3 \\ 1 & -1 & -1 & 1 \\ -1 & 3 & -3 & 1 \end{pmatrix} \tag{3.22}$$

By substituting for τ from (3.20) in (3.17), we have

$$\begin{aligned}
\mathbf{Y} &= (\mathbf{Q}\hat{\tau})\mathbf{X}(\mathbf{Q}\hat{\tau})' \\
&= \mathbf{Q}(\hat{\tau}\mathbf{X}\hat{\tau}')\mathbf{Q}' \qquad \text{(using matrix properties } (\mathbf{AB})' = \mathbf{B}'\mathbf{A}' \text{ and } \mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}) \\
&= (\hat{\tau}\mathbf{X}\hat{\tau}') \odot \hat{\mathbf{Q}} \qquad \text{(using Theorem B.1)}
\end{aligned}$$

Consequently, the transform equation of (3.17) may be rewritten as

$$\mathbf{Y} = (\hat{\tau}\mathbf{X}\hat{\tau}') \odot \hat{\mathbf{Q}} \tag{3.23}$$

where for $N = 4$, $\hat{\mathbf{Q}}$ is given by

$$\hat{\mathbf{Q}} = \begin{pmatrix} q_0^2 & q_0q_1 & q_0q_2 & q_0q_3 \\ q_0q_1 & q_1^2 & q_1q_2 & q_1q_3 \\ q_0q_2 & q_1q_2 & q_2^2 & q_2q_3 \\ q_0q_3 & q_1q_3 & q_2q_3 & q_3^2 \end{pmatrix} = \begin{pmatrix} 0.2500 & 0.1118 & 0.2500 & 0.1118 \\ 0.1118 & 0.0500 & 0.1118 & 0.0500 \\ 0.2500 & 0.1118 & 0.2500 & 0.1118 \\ 0.1118 & 0.0500 & 0.1118 & 0.0500 \end{pmatrix} \tag{3.24}$$

which follows from (B.1) of Theorem B.1.

In (3.23), $\hat{\mathbf{Y}}$ is the 4×4 ITT given by

$$\hat{\mathbf{Y}} = \hat{\tau}\mathbf{X}\hat{\tau}' \tag{3.25}$$

The normalization is complete after element-by-element multiplication is carried out between the core transform, $(\hat{\tau}\mathbf{X}\hat{\tau}')$ and $\hat{\mathbf{Q}}$. This operation is incorporated into the quantization stage. Alternatively, but on the same lines, we derive another possibility of deriving 4×4 ITT, so that ITT and ICT are easily comparable for the purpose of computational complexity.

Alternate Intermediate Framework

If we define $c = 1/2$ and $d = 1/\sqrt{5}$ in (3.18), τ is re-written as

$$\tau = \begin{pmatrix} c & c & c & c \\ -3cd & -cd & cd & 3cd \\ c & -c & -c & c \\ -cd & 3cd & -3cd & cd \end{pmatrix}$$

Hence, the 1-D transform $\mathbf{Y} = \tau\mathbf{X}$ is given as

$$\mathbf{Y} = (\hat{\tau}\mathbf{X}) \odot \mathbf{S} \tag{3.26}$$

where $\odot$ represents element-element multiplication, and

$$\mathbf{S} = \begin{pmatrix} c & c & c & c \\ cd & cd & cd & cd \\ c & c & c & c \\ cd & cd & cd & cd \end{pmatrix}, \quad \hat{\tau} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -3 & -1 & 1 & 3 \\ 1 & -1 & -1 & 1 \\ -1 & 3 & -3 & 1 \end{pmatrix} \tag{3.27}$$

$\mathbf{S}$ is the scaling matrix that can be separated from the core transform computation which involves the new matrix $\hat{\tau}$. Since $\hat{\tau}$ is orthogonal, but not orthonormal, normalization is complete when the scaling matrix $\mathbf{S}$ is merged into the quantization process.

The expression in (3.8) is factorized as

$$\mathbf{Y} = (\hat{\tau}\mathbf{X}\hat{\tau}') \odot \hat{\mathbf{S}} \tag{3.28}$$

where

$$\hat{\mathbf{S}} = \begin{pmatrix} c^2 & c^2d & c^2 & c^2d \\ c^2d & c^2d^2 & c^2d & c^2d^2 \\ c^2 & c^2d & c^2 & c^2d \\ c^2d & c^2d^2 & c^2d & c^2d^2 \end{pmatrix}$$

Multipliers are not needed for a transform such as (3.26) involving $\hat{\tau}$ given by (3.27), since multiplications by 3 can be implemented by shift and add operations. This formulation lays a framework to compute a fast integer DTT for applications with 4x4 transform block coding.

3.3.3 Transform Coding using DTT in Video Compression

Next, we illustrate the possible use of DTT in video compression by comparing the computational nature of DTT and ICT, which is currently being used in the video compression standard, H.264/ AVC. In the next chapter, we demonstrate via comparison of image compression performances of ITT, ICT (and DCT), that ITT performs better than ICT (and DCT). The H.264/ AVC standard is discussed below to explain the role of DTT in the transform coding stage.

Review of MPEG-4 Advanced Video Coding

Most of the performance gain in image and video processing applications is attributed to tailor-made quantization and coding schemes. DCT of block size 8x8 is generally used in image processing applications due to its superior performance. The H.264 standard uses a 4x4 ICT in order to facilitate exact reversibility for coding and decoding mainly due to its low computational complexity [21]. This has brought about a reduction in the ringing effects resulting from predictive coding, and has also made it possible to use 16-bit arithmetic in the transform calculation. The forward 2-D transform used in H.264 [22, 23] is $\mathbf{Y}_{2D} = (\mathbf{HXH}') \odot \mathbf{E}_{\mathbf{f}}$, where $(\mathbf{HXH}')$ is the core transform. The 1-D transform is given as $\mathbf{Y}_{1D} = (\mathbf{HX}) \odot \mathbf{E}$. Here, the transform matrix $\mathbf{H}$ is an integer approximation of the DCT

matrix ($\mathbf{H} = round(\alpha \mathbf{H_{DCT}})$) [24].

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}, \quad \mathbf{E} = \begin{pmatrix} a & b/2 & a & b/2 \\ b/2 & b/2 & b/2 & b/2 \\ a & b/2 & a & b/2 \\ b/2 & b/2 & b/2 & b/2 \end{pmatrix}$$

and

$$\mathbf{E_f} = \begin{pmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{pmatrix} \tag{3.29}$$

Values of a and b in E, E_f are $a = \frac{1}{2}, b = \sqrt{\frac{2}{5}}$ (b is modified from 0.6533 to 0.6325 in order to preserve the orthogonality of the transform). Effectively, $\mathbf{H} \odot \mathbf{E}$ represents the basis functions used by ICT given by

$$\mathbf{H} \odot \mathbf{E} = \begin{pmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ 0.6325 & 0.3162 & -0.3162 & -0.6325 \\ 0.5000 & -0.5000 & -0.5000 & 0.5000 \\ 0.3162 & -0.6325 & 0.6325 & -0.3162 \end{pmatrix} \tag{3.30}$$

whereas the actual kernel of DCT is

$$\begin{pmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ 0.6533 & 0.2706 & -0.2706 & -0.6533 \\ 0.5000 & -0.5000 & -0.5000 & 0.5000 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{pmatrix} \tag{3.31}$$

We note that the transition from DCT to ICT has inflicted some approximation errors. But, the computational ease of the core transform outweighs the consideration of these errors [22].

ITT as a Substitute for ICT in MPEG-4

In addition to all the benefits offered by ICT, DTT has added advantages and hence, can be used in video processing applications. It has already been shown in Sub-section 3.3.2 that a separable core transform, ITT, pertaining to the DTT, is arrived at with ease and without any approximation. By observing the following points, it is seen that ITT has an edge over ICT.

1. DTT is akin to DCT and their performances are very close in general. In particular, for a 4x4 block size, DTT outperforms DCT in most cases.

2. Approximation errors are not present in ITT inspite of having integralized the DTT, since modification of any kind is not required to make it multiplier-free or to obtain its integral representation.

3. DTT has a polynomial kernel. Hence, in a way, we are already dealing with integers by scaling the rows of DTT.

4. The dynamic range of ITT is comparable to that of ICT. The transform coefficients are bounded by g^2U, where U is the upper limit of the input [22]. In the case of ITT, g=8. Since $log_2 64 = 6$, for U=255, 6 more bits are required to compute the transform. Hence 16-bit arithmetic can be used in ITT.

5. **H** in (3.29) can also be obtained from the DTT basis in (3.18) just as in the case of DCT using $\mathbf{H} = round(\alpha\tau)$ for $\alpha = 2.5$, with a sign reversal for the even rows.

6. Since ITT does not involve approximations, the scaling matrix used in quantization does not have to be modified to satisfy the orthogonality condition, though this does not really affect the performance in any way.

3.3.4 Fast Algorithms for 8-point and 4-point ITTs

Multiplications increase complexity, power consumption and result in bulky designs, with increased computational time. Also, not all processors have in-built multiply units. Most modern-day processors have built-in instructions for addition and shift operations. Hence, later in this chapter, we propose multiplier-free algorithms with only shifts and additions for 4×4 and 8×8 ITTs.

8-point Discrete Tchebichef Transform

The DTT of an 8-point input sequence $x(n)$, given by (3.1), for $N = 8$, is

$$Y(k) = \sum_{n=0}^{7} \tau(k,n)x(n) \qquad \text{for } k,n = 0,1 \cdots 7 \tag{3.32}$$

where the kernel $\tau(k,n)$ is the orthogonal basis of DTP. The inverse DTT in (3.2) is given as

$$x(n) = \sum_{k=0}^{7} \tau(k,n)Y(k) \qquad \text{for } k,n = 0,1 \cdots 7 \tag{3.33}$$

2-D 8x8 Discrete Tchebichef Transform

The 2-D DTT in (3.6), for $N = 8$, is given as

$$Y(k_1,k_2) = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{M-1} \tau(k_1,n_1)\tau(k_2,n_2)x(n_1,n_2) \tag{3.34}$$

$$\text{for } k_1,n_1 = 0,1 \cdots 7 \text{ and } k_2,n_2 = 0,1 \cdots 7$$

The inverse 2-D DTT for $N = 8$ is given by

$$x(n_1,n_2) = \sum_{k_1=0}^{7} \sum_{k_2=0}^{7} \tau(k_1,n_1)\tau(k_2,n_2)Y(k_1,k_2) \tag{3.35}$$

$$\text{for } k_1,n_1 = 0,1 \cdots 7 \text{ and } k_2,n_2 = 0,1 \cdots 7$$

Fast 8-point ITT Algorithm

Here, we design [19] a multiplier-free algorithm, by using the ITT proposed in (3.16) in the previous section. As already explained in the introduction of this chapter, a multiplier-free design is computationally fast and efficient. The complexity involved in the core transform computation is optimized by devising a fast algorithm. The steps involved in the algorithm are given below.

Step 1:

$$\begin{aligned} u_0 &= x_0 + x_7 \\ u_1 &= x_1 + x_6 \\ u_2 &= x_2 + x_5 \\ u_3 &= x_3 + x_4 \end{aligned} \quad \text{and} \quad \begin{aligned} v_0 &= x_0 - x_7 \\ v_1 &= x_1 - x_6 \\ v_2 &= x_2 - x_5 \\ v_3 &= x_3 - x_4 \end{aligned} \tag{3.36}$$

Step 2:

$$\begin{aligned} k_0 &= u_0 + u_2 \\ k_1 &= u_1 + u_3 \end{aligned} \quad \text{and} \quad \begin{aligned} z_0 &= v_0 + v_3 \\ z_1 &= v_1 - v_2 \\ z_2 &= v_1 + v_2 \end{aligned} \tag{3.37}$$

Step-3:

$$\begin{aligned} m_0 &= k_0 + k_1 \\ m_1 &= k_0 - k_1 \\ m_2 &= bu_0 - au_2 \end{aligned} \quad \text{and} \quad \begin{aligned} w_0 &= -(z_1 + z_0) \\ w_1 &= z_1 - z_0 \end{aligned} \tag{3.38}$$

where $b = 6$ and $a = 4$

Step 4:

$$\begin{aligned} l_0 &= m_0 + m_2 & & l_5 = w_1 + az_1 \\ l_1 &= m_1 + m_2 & & l_6 = (av_3 + bz_2) \\ l_2 &= a(2k_1 + u_1) & \text{and} & \\ l_3 &= a(2u_2 - k_1) & & l_7 = 2a(v_1 - v_3) \\ l_4 &= w_0 - bv_0 & & l_8 = bv_3 - av_2 \end{aligned} \tag{3.39}$$

Step 5:

$$\boxed{\begin{aligned} y_0 &= m_0 & & y_4 = l_1 - l_2 \\ y_1 &= l_4 - az_2 & & y_5 = l_5 + l_7 \\ y_2 &= l_0 - au_3 & \text{and} & y_6 = m_1 + l_3 \\ y_3 &= l_4 + l_6 & & y_7 = l_5 + l_8 \end{aligned}} \tag{3.40}$$

The only multiplying coefficients that appear in the proposed algorithm are $a = 4$ and $b = 6$. The proposed fast algorithm thus eliminates multiplications altogether, since multiplication by a can be implemented by two shifts, and that with b can be implemented by adding the left-shifted value of the multiplicand to its twice left-shifted value i.e., $m \times a = \ll m$ and $m \times b = \ll m + \ll\ll m$, where $\ll$ represents one-bit left shift.

Table 3.1 gives a comparison of the number of computations of the proposed fast algorithm with those of pouplar fast DCT algorithms [6–8]. It is seen that the proposed algorithm requires only a reasonable number of additions to compute the DTT, inspite of the fact that all multiplications have been eliminated. Since a fast algorithm for DTT does not exist in the literature, we compare the computational complexity of our algorithm only with that of the DCT.

2-D ITT can be computed efficiently by using the proposed 1-D algorithm first along the rows and then along the columns in two steps by using the separability property of the DTT. Using the proposed fast algorithm, 8-point ITT can be implemented in hardware using a 5-stage pipelined hardware as proposed in Fig. 3.5. The functions of the individual cells of

Table 3.1: Arithmetic Complexities of 8x8 DCT and 8x8 ITT

Input Type	Operation	DCT [6]	DCT [7]	DCT [8]	ITT (proposed)
8-point 1-D array	add	N/A	28	29	38
	mult	N/A	11	5	0
	shift	N/A	N/A	N/A	24
(8×8)-point 2-D array	add	454	448	464	608
	mult	94	176	144	0
	shift	6	N/A	N/A	384

the architecture shown in Fig. 3.5 are given in Fig. 3.6. Fig. 3.7 gives the computational flowgraph for the proposed architecture for the computation of 2-D ITT of size 8×8.

4-point Discrete Tchebichef Transform

The DTT of a 4-point input sequence $x(n)$, given by (3.1), is

$$Y(k) = \sum_{n=0}^{3} \tau(k,n)x(n) \qquad \text{for } k,n = 0,1,2,3 \tag{3.41}$$

and the inverse 4-point DTT in (3.2) is given by

$$x(n) = \sum_{k=0}^{3} \tau(k,n)Y(k) \qquad \text{for } k,n = 0,1,2,3 \tag{3.42}$$

Orthonormal DTP can be generated recursively by using (2.35).

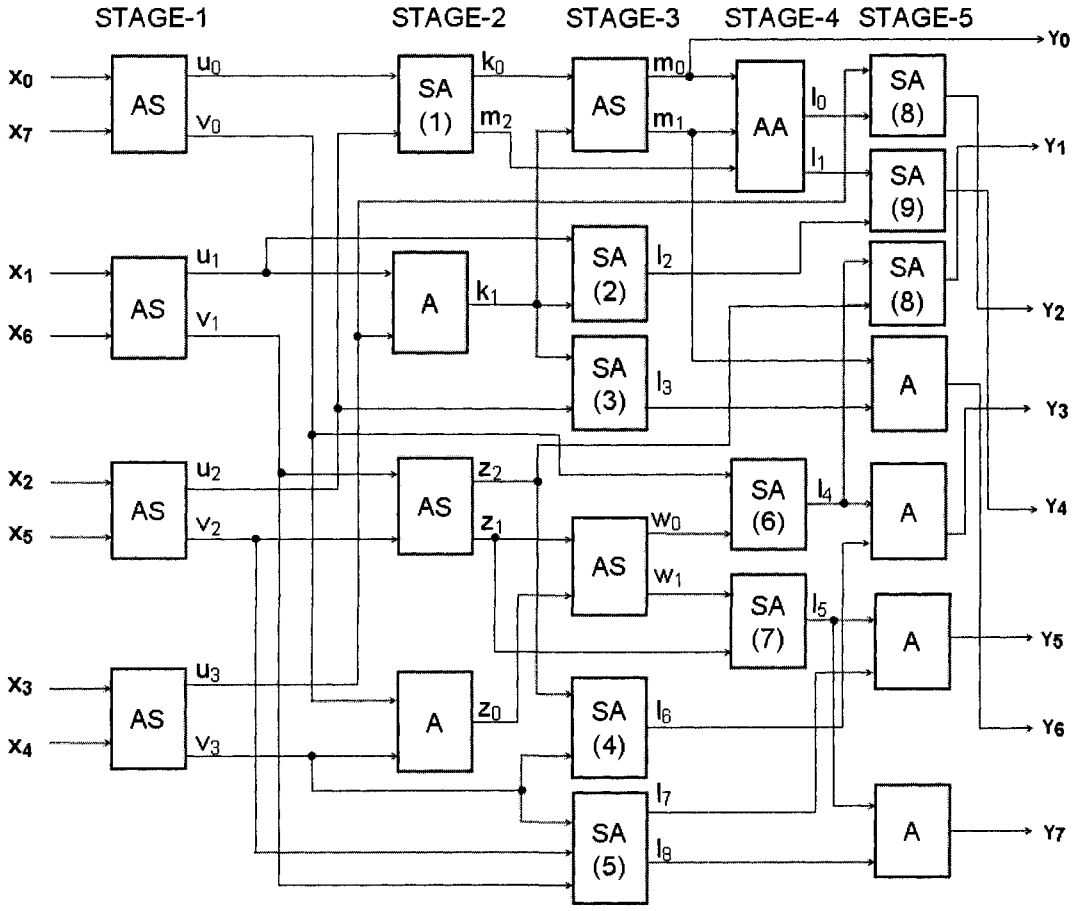


Figure 3.5: 5-Stage Pipelined Design for the Proposed Fast Algorithm for 8-point ITT

2-D 4x4 Discrete Tchebichef Transform

For a 2-D input sequence $x(n_1, n_2)$, the 2-D DTT of order 4×4 is defined as

$$Y(k_1, k_2) = \sum_{n_1=0}^{3} \sum_{n_2=0}^{3} \tau(k_1, n_1)\tau(k_2, n_2)x(n_1, n_2) \tag{3.43}$$

for $k_1, n_1 = 0, 1, 2, 3$ and $k_2, n_2 = 0, 1, 2, 3$

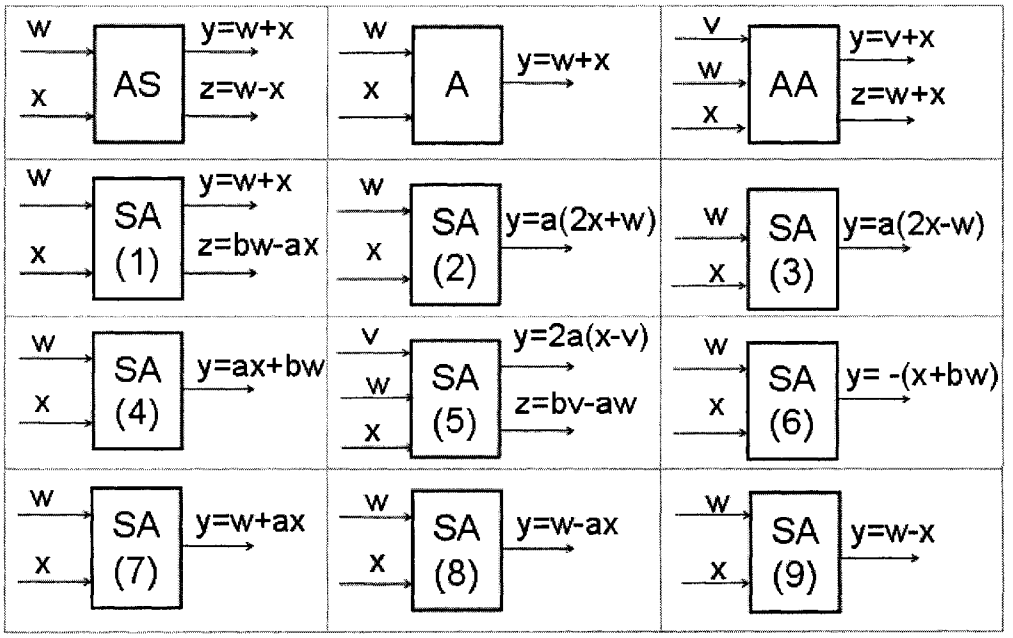


Figure 3.6: Function of Individual Cells of Proposed Pipelined Design in Fig. 3.5

The inverse 2-D 4×4 DTT is given by

$$x(n_1, n_2) = \sum_{k_1=0}^{3} \sum_{k_2=0}^{3} \tau(k_1, n_1)\tau(k_2, n_2) Y(k_1, k_2) \tag{3.44}$$

$$\text{for } k_1, n_1 = 0, 1, 2, 3 \text{ and } k_2, n_2 = 0, 1, 2, 3$$

Fast Algorithm for 4x4 ITT

A fast multiplier-free algorithm for 4×4 ITT is proposed here [20] to compute the core transform $\hat{\mathbf{Y}} = \hat{\tau}\mathbf{X}\hat{\tau}'$, otherwise known in this thesis as ITT. 1-D ITT is obtained from the

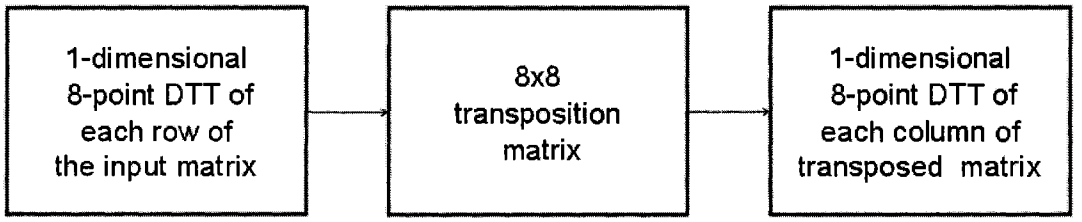


Figure 3.7: Computational Flowgraph of 2-D ITT of Size 8x8

matrix multiplication of the input data vector **X** with the coefficient matrix $\hat{\tau}$ in (3.26)

$$\begin{pmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -3 & -1 & 1 & 3 \\ 1 & -1 & -1 & 1 \\ -1 & 3 & -3 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{3.45}$$

Utilizing the symmetry behavior of DTT in (2.24), the input vector is grouped as

$$u_0 = x_3 + x_0,$$

$$v_0 = x_3 - x_0,$$

$$u_1 = x_2 + x_1,$$

$$v_1 = x_2 - x_1$$

The structure of the coefficient matrix $\hat{\tau}$ in (3.27) is utilized for its factorization into two sparse matrices to increase the computational efficiency. $\hat{\tau}$ is written as $\hat{\tau} = \hat{\tau_1} + 3\hat{\tau_2}$, where

$$\hat{\tau_1} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & 1 & 0 \\ 1 & -1 & -1 & 1 \\ -1 & 0 & 0 & 1 \end{pmatrix}, \quad \hat{\tau_2} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \end{pmatrix}$$

Output data points in (3.45) can be separated into even and odd parts, Ye and Yo.

$$\mathbf{Y} = \begin{pmatrix} Ye \\ Yo \end{pmatrix}$$

for

$$Ye = \begin{pmatrix} Y_0 \\ Y_2 \end{pmatrix}, \quad Yo = \begin{pmatrix} Y_1 \\ Y_3 \end{pmatrix}$$

or

$$Ye = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \end{pmatrix}$$

and

$$Yo = \left\{ 3 \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right\} \begin{pmatrix} v_0 \\ v_1 \end{pmatrix}$$

If we denote $[\mathbf{u}] = \begin{pmatrix} u_0 \\ u_1 \end{pmatrix}$ and $[\mathbf{v}] = \begin{pmatrix} v_0 \\ v_1 \end{pmatrix}$, Ye and Yo can be written compactly as

$$Ye = \mathbf{H_2}[\mathbf{u}] \text{ and } Yo = 3\mathbf{D_2}[\mathbf{v}] - \mathbf{E_2}[\mathbf{v}],$$

where $\mathbf{D_2}$ is a unit diagonal matrix with alternating sign, and $\mathbf{H_2}$ and $\mathbf{E_2}$ are the second order Hadamard and exchange matrices respectively. Multiplications by 3 can be substituted by shift and add operations as mentioned before. The transformed output data points in (3.45)

can also be written as

$$Y_0 = u_0 + u_1,$$

$$Y_1 = 3v_0 + v_1 = (v_0 + \ll v_0) + v_1,$$

$$Y_2 = u_0 - u_1,$$

$$Y_3 = v_0 - 3v_1 = v_0 - (v_1 + \ll v_1)$$

where $\ll$ represents one-bit left shift.

Fig. 3.8 shows the signal flow graph for this algorithm. Table 3.2 gives a comparison of complexity involved in transform computation. The complexities for the proposed algorithm and ICT are obtained by extending the proposed 1-D algorithms to 2-D using the separability property of the transforms.

The proposed method has eliminated multiplications altogether. The number of additions can be further reduced by designing algorithms specifically for the 2-D case. Since the implementation of shift operations is simpler than that of arithmetic operations, ITT is computationally more efficient than the other algorithms in Table 3.2. Owing to the separability property of DTT, 2-D transform can be evaluated using a row-column approach, wherein a 1-D transform is applied to the rows and then to the columns of the input.

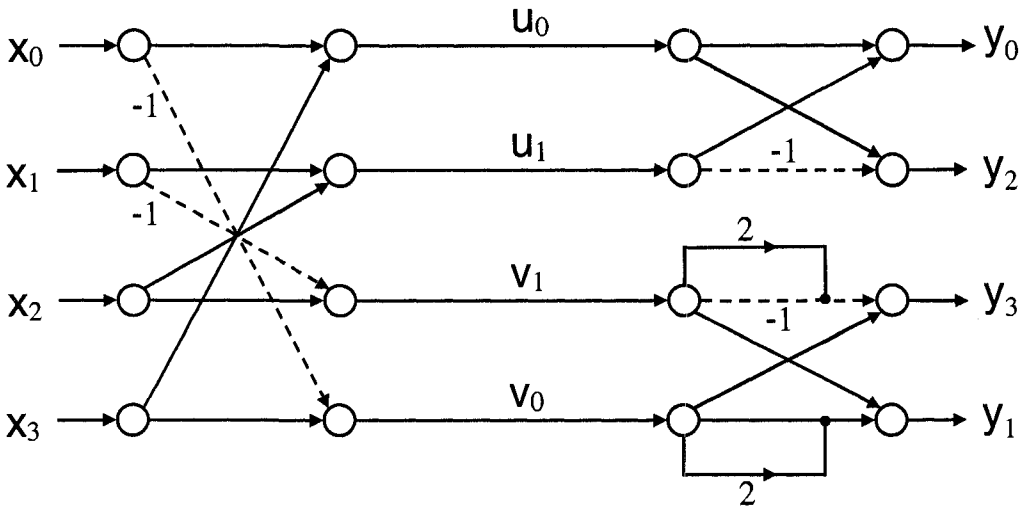


Figure 3.8: Signal Flow Diagram for the Proposed Fast Algorithm for 4-point ITT

Table 3.2: Complexity Comparison of Fast 4x4 2-D Transforms

	Conventional	DCT [25]	DTT [10]	ICT [21]	Proposed
Add	72	74	66	64	80
Mul	32	16	32	0	0
Shifts	-	-	-	16	16

Chapter 4

Image Compression

4.1 Introduction

Recent technological developments have spawned a generation of digital multimedia products applicable to the medical, biological, space, security, geo-physical and entertainment realms. The storage and transmission of graphics, video and audio consume a large amount of memory and transmission bandwidth. For example, a color image of dimension 100×100 represented by 8 bits per pixel requires $100 \times 100 \times 8 \times 3 = 240$ Kbits in memory. Compression of images enables us to reduce the storage and transmission requirements. Compression techniques rely on correlation and redundancies in images. Generally, compression also involves coding, which can be useful to encrypt data for security purposes. Image compression finds applications in many areas such as image sharing over the internet, teleconferencing for educational, professional and medical purposes, medical image archival and storage of personal multimedia data.

4.1.1 Digital Images

A digital image is a two-dimensional function, $f(x, y)$, where x and y are coordinates in the spatial domain, where the value at a coordinate represents the intensity of the image at that point in the spatial plane [26]. In general, we represent a digital image of dimension $M \times L$ as shown in Fig. 4.1. Pixels or picture elements are the building blocks of digital

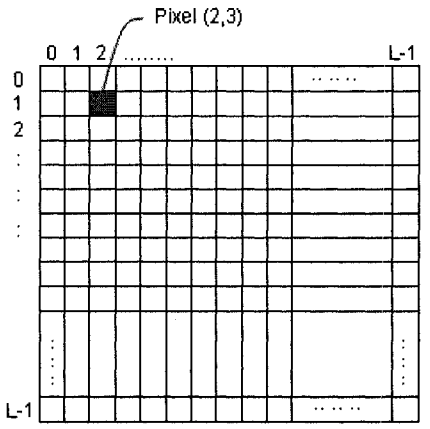


Figure 4.1: Representation of a Digital Image in Spatial Domain

images. It is easy to visualize an image as a 2-D matrix with each coordinate representing a pixel intensity value. Images are either color or gray (black and white). Pixel values in a black and white image can be any of the different shades of gray. A pixel represented by n bits can take any of the 2^n different values. If we assume 256 levels of gray, $2^8 = 256$, hence each pixel takes 8 bits in memory. RGB color spaces have three sub-channels red, green and blue, and hence need three times the space as that of gray images. If we again assume 256 levels for each sub-channel, each pixel takes $8 \times 3 = 24$ bits in memory. Among the many available file formats for images, the most common ones are graphics interchange format (GIF), tagged image file format (TIFF), image silicon graphics (SGI), PICT, microsoft windows bitmap (BMP), X bitmap (XBM), PGM and JPEG.

Some of the image formats such as JPEG and BMP convert the original image to a compressed format which maybe lossy, and some of them such as GIF, PGM and TIFF

allow the lossless storage of images.

Redundancies in Images

An image typically contains redundancies such as coding, psychovisual, spatial and statistical redundancies, that can be exploited in images for the purpose of compression [26]. *Coding redundancies* can be reduced by using code words of varying lengths, based on the probability of occurrence of certain pixel values. For instance, pixel values occurring with a higher probability can be represented by shorter bit lengths compared to those that occur less frequently. *Psychovisual redundancies* are based on the assumption that the components of an image, whose change is not detected by the human eye, are not considered to be significant by an image compression system. The *human visual system* (HVS) is more sensitive to edge information in an image. As shown in Fig. 4.2, the human eye is most sensitive to frequencies around 5 cycles per degree, and is not sensitive to frequencies above 100 cycles per degree [27]. It can also be seen from the graph that variations in *chrominance* components are less detected by the HVS compared to variations in *luminance* components. This type of irrelevance can be removed by performing *color quantization*, which is an irreversible and a lossy process in which the number of colors in the image are reduced. *Inter-pixel redundancies* rely on the fact that neighboring pixels have similar values, and hence these values are predictable. These *statistical redundancies* can be removed by performing variable length coding (VLC). Predictive coding techniques can be used to eliminate spatial correlations in addition to application of orthogonal transforms with good decorrelation capabilities such as DCT.

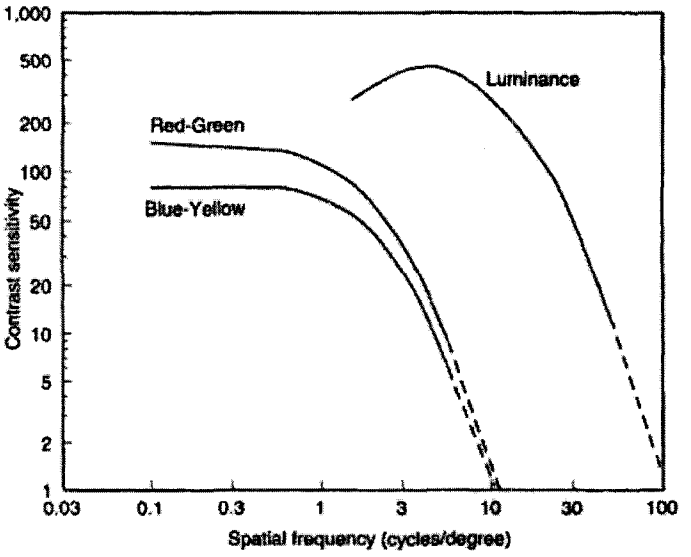


Figure 4.2: Sensitivity of the Eye to Luminance and Chrominance Changes

4.2 Existing Compression Techniques

Image compression can be either *lossy* or *lossless*. In lossless compression, all the information is preserved, and hence, the image can be reconstructed exactly. Lossless compression techniques depend largely on predictable characteristics of the images. Examples of lossless compression are run length encoding (RLE), lossless JPEG (JPEG-LS) and LZW compression. Common image formats for lossless compression techniques are TIFF using the LZW compression and BMP using the RLE compression [28]. Lossy compression techniques are used in a host of applications where some loss of information can be tolerated. Coding is carried out by examining the components of the image which are not influential in image perception. An optimal lossy compression technique is that which requires a low bandwidth utilization, while providing good visual quality along with a low system complexity. Image file formats of the lossy type are JPEG, LWF (Lurawave format)

and FIF (fractal image format). Popular lossy techniques which are also international compression standards are JPEG and JPEG 2000 (JPEG 2K). JPEG is an image compression standard [29] and operates in one lossless mode and three lossy modes which are sequential, progressive and hierarchical modes, respectively. In the baseline or the sequential JPEG scheme, the image is grouped into non-overlapping blocks of size 8x8. They are transformed using DCT and quantized followed by a zigzag ordering, which are then Huffman coded. Linear predictive coding is used in the lossless JPEG mode. Since image pixel values are highly correlated, they are predicted depending on neighboring pixels. Progressive JPEG compression is similar to the baseline JPEG scheme except that the quantized coefficients are coded in multiple scans until the desired quality is achieved. JPEG coding results in blocking artifacts in the compressed image, since correlation across the block boundaries is not eliminated. Lapped orthogonal transforms (LOT) have been proposed to deal with this problem.

JPEG 2000 performs a wavelet transform on the image, quantizes the transformed image and codes the quantized image using bit plane coding and arithmetic entropy coding. In comparison with JPEG, JPEG 2000 offers advantages such as higher compression ratios, better error resilience and region of interest (ROI) capability. The complexity of a JPEG 2000 system is much higher than that of JPEG, and hence, JPEG 2000 is used selectively. While lossless compression procedures can achieve lower *compression ratios* of about 3:1, lossy compression techniques can achieve relatively higher compression ratios ranging from 5:1 upto 200:1.

4.2.1 Steps involved in Image Compression

The following steps are carried out generally in compression systems

- *Color space conversion*: RGB formats, which are widely used in images exhibit a high visual correlation between the three color components. Hence, these images are

converted to the YCbCr space, where *luminance* is described by Y, whereas Cb and Cr represent the color information.

- *Chroma sub-sampling*: As explained in the previous section, the HVS is more sensitive to luminance components of an image compared to the *chrominance* components. Hence, chroma re-sampling is carried out as a part of the compression process.

- *Transform coding*: The image is converted from spatial to transform domain by applying a discrete transform such as DCT or DWT, followed by *quantization* and *entropy coding* techniques such as Huffman, arithmetic or progressive coding.

4.3 Image Compression Scheme

As explained above, image compression is a multi-stage process. For our purpose, we define the following scheme to carry out image compression using DTT. The concepts of the different stages of this scheme are mainly borrowed from the JPEG compression scheme [29,30]. The various steps in the *encoding* and *decoding* process involved in image compression using DTT are given below.

Encoding process:

- Images are divided into sub-blocks of size $N \times N$.

- The individual values of the image sub-block are level-shifted by 2^{p-1}, where p is the precision of the input image.

- The level-shifted $N \times N$ image block is transformed using DTT of block size $N \times N$.

- The transform coefficients are quantized and rounded to the nearest integer.

- The quantized coefficients are scanned in a zigzag order.

- The frequency-ordered coefficients are coded using differential and Huffman coding.

Decoding process:

- The encoded bit-stream is decoded.
- The decoded values are re-arranged from the zigzag order and de-quantized using the quantization tables.
- Inverse DTT is performed on the de-quantized values and the resulting values are level-shifted back to their original format.
- The image sub-blocks are re-combined to reconstruct the original image.

The codec for the image compression technique adopted in this thesis is shown in Fig. 4.3.

4.3.1 Encoding and Decoding Process

The steps mentioned above to obtain the compressed image are explained in detail below. Sample plots are also given for intermediate outputs in the coding and decoding paths to illustrate the working of the codec. Figs. 4.4 (a) and (b) show the mesh and histogram plots of an 8×8 image sub-block of *Einstein* shown in Fig. 4.5-(a).

Sub-block Extraction

The input image of size $M \times L$ is grouped into *sub-blocks* of size $N \times N$, and the subsequent operations are performed on each $N \times N$ sub-block. Additional samples are added if required to ensure integral number of blocks, and further additional samples are mirrored along the existing samples within a block, when the remaining existing samples (after grouping $N \times N$ sub-blocks) cannot completely form an $N \times N$ block. (This is consistent with the idea that neighboring pixels are highly correlated.) During decoding, the header information suggests the presence of additional samples that can be removed in order to

reconstruct the original image. This process is shown in Fig. 4.5, where 4.5(a) shows the original image of *Einstein* of size 256×256, 4.5(b) and 4.5(c) represent the first 8×8 image sub-block. After decoding, the reverse operation is carried out, i.e., the sub-blocks are re-grouped and the original image is reconstructed.

Level Shifting

In order to reduce internal precision requirements, the range of the input data is changed from 0 to $2^p - 1$ to the range -2^{p-1} to $2^{p-1} - 1$, where p is the precision of the input image, i.e., for $p = 8$, for an image of size 256×256 with unsigned integer values in the range 0 to 255, the level shifted values are signed integers in the range -128 to $+127$. Fig. 4.7(a) shows the mesh plot of the level-shifted image sub-block of *Einstein*. After decoding, the values are level-shifted back to the original unsigned representation.

Discrete Tchebichef Transform

The discrete Tchebichef transform is performed on the level-shifted image sub-block. This process decorrelates the components of the sub-block, and the energy compaction property of DTT also ensures that the image information contained in the sub-block is represented by fewer coefficients in the Tchebichef domain. This is evident in the transformed image sub-block coefficients in Fig. 4.6 of *Einstein* shown in Fig. 4.5. The transform stage does not perform any image compression by itself, but contributes to the process by structuring coefficients in a manner advantageous to the coding stage.

Fig. 4.7(b) shows the mesh plot of the transformed sub-block of *Einstein*.

Quantization

Quantization is an irreversible and a lossy process, which contributes to the compression of data. The transformed coefficients, $Y_{k_1 k_2}$, are quantized using uniform quantization with

different *quantization steps*, $Q_{k_1k_2}$, as

$$T_{k_1k_2} = round(\frac{Y_{k_1k_2}}{Q_{k_1k_2}}) \tag{4.1}$$

where k_1, k_2 are the frequency domain indexes. $Q_{k_1k_2}$ corresponds to the entry of the *quantization matrix* in the position (k_1, k_2). The quantization matrix is user-defined. The rounding function rounds the result to the nearest integer and ensures that we deal with integral values. The reverse operation is carried out to de-quantize the values in the decoding process. Figs. 4.8(a) and 4.8(b) show the mesh plots of the quantized and rounded sub-blocks of *Einstein* for lower and higher quantization values respectively.

Zigzag scanning

The transform stage results in the concentration of the significant image components in the lower spatial frequencies. The quantized DC coefficient undergoes differential encoding [31] and the quantized AC coefficients are arranged in a *zigzag scan* order, which sorts the quantized values in the increasing order of spatial frequencies. Since spatial frequencies increase horizontally as we move from left to right, and vertically as we move from top to bottom, the scanning order shown in Fig. 4.9 has been found to be useful [31].

Entropy Coding

Entropy coding techniques such as arithmetic coding and RLE help in further compression of the quantized coefficients by removing statistical redundancies. Huffman encoding maps the zigzag ordered quantized coefficients into shorter symbols depending on the probability of occurrence of the symbols [31]. The decoding process extracts the symbols based on the statistical information. User-defined Huffman code tables are required to be fed to the encoder and the decoder. Figs. 4.10(a) and 4.10(b) show the histogram plots of the decoded sub-block for lower quantization and higher quantization values respectively for *Einstein*.

Since a histogram plot shows the number of pixels at different intensity values of an image, it is an indicator of the entropy of the image. More uniformly distributed histogram plots with higher number of pixels suggest a higher entropy since, if all the source symbols are equally probable, then the source has a maximum entropy. An image with a higher entropy requires more number of bits to represent each pixel. Since higher quantization levels allow image encoding with fewer bits (higher compression), it is seen from Figs. 4.10(a) and 4.10(b) that Fig. 4.10(b) offers better compression. More evenly distributed histogram plots have higher image contrast and clustered histogram plots have lower image contrast. This observation directly translates to the principle of image compression, which takes advantage of the HVS characteristics, i.e., the HVS is not very sensitive to variation in contrast. Figs. 4.11 (a) and (b) show the mesh plots of the decoded sub-blocks of *Einstein* for lower and higher quantization values, respectively.

4.3.2 Sample Precision

The *precision* of the transformed coefficients is bounded by g^2U, where U is the upper limit of the input and g is the maximum value for the sum of the absolute values of any row [22]. For 8×8 DTT, from (3.9), g^2=8. Since $log_2 8 = 3$, 3 more bits are required for the computation of DTT. Hence, for U=255, that is for image values of 8-bit precision, the output images have 8-bit precision and the quantized DTT coefficients are restricted to 11 bits. (The quantization steps are restricted to 8-bit precision). For 4×4 DTT, from (3.18), g^2=4. Since $log_2 4 = 2$, 2 more bits are required for the computation of DTT. Hence, for U=255, that is, for image values of 8-bit precision, the output images have 8-bit precision and the quantized DTT coefficients are restricted to 10 bits.

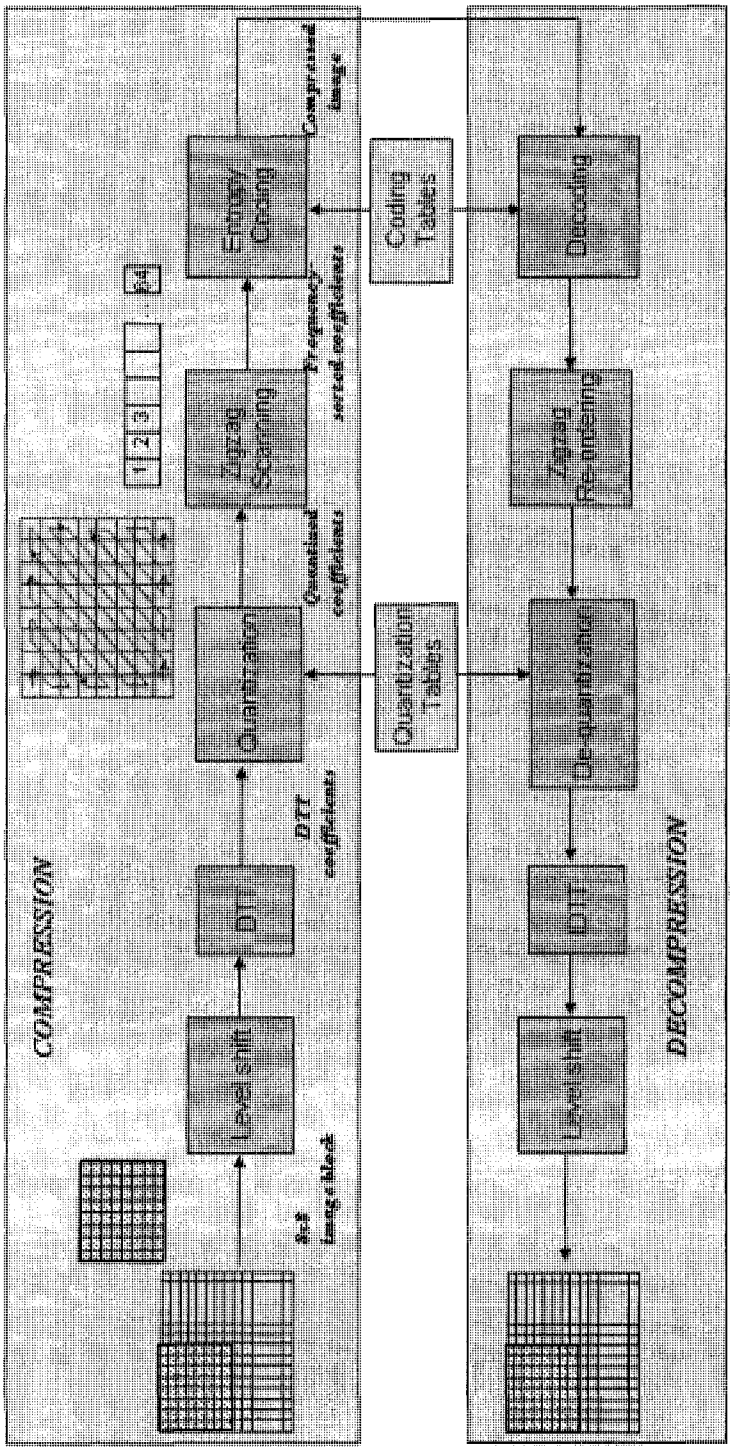
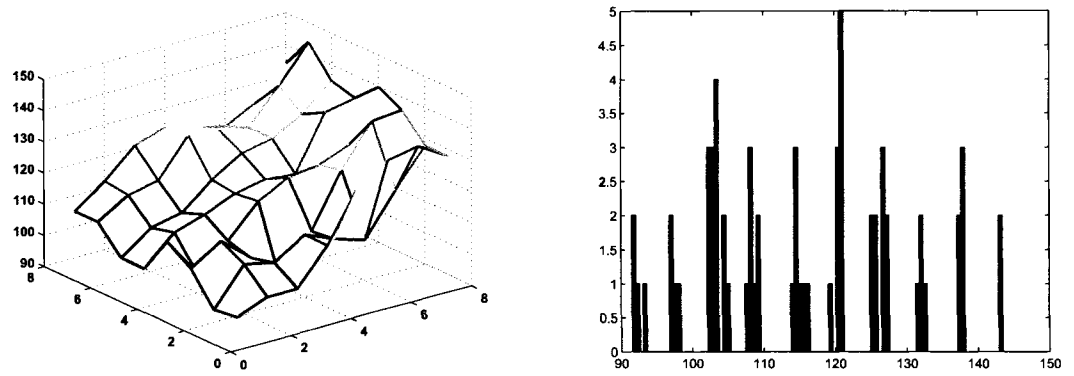


Figure 4.3: Image Compression Scheme for Discrete Tchebichef Transform



(a) (b)

Figure 4.4: (a) Mesh Plot and (b) Histogram plot Characteristics of the Input Sub-blocks of *Einstein*

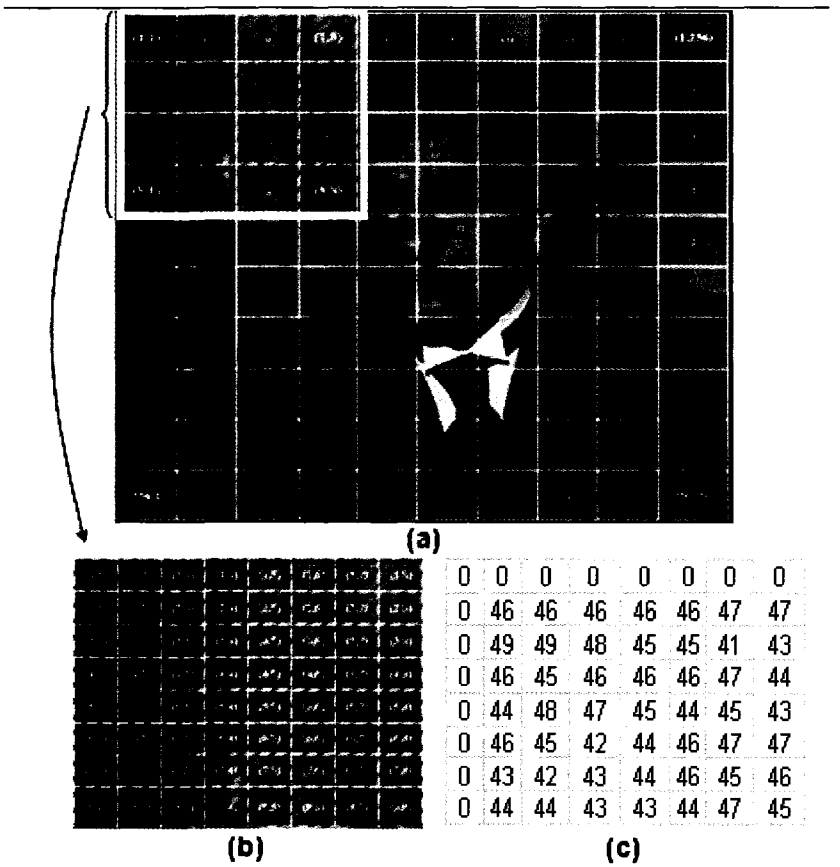
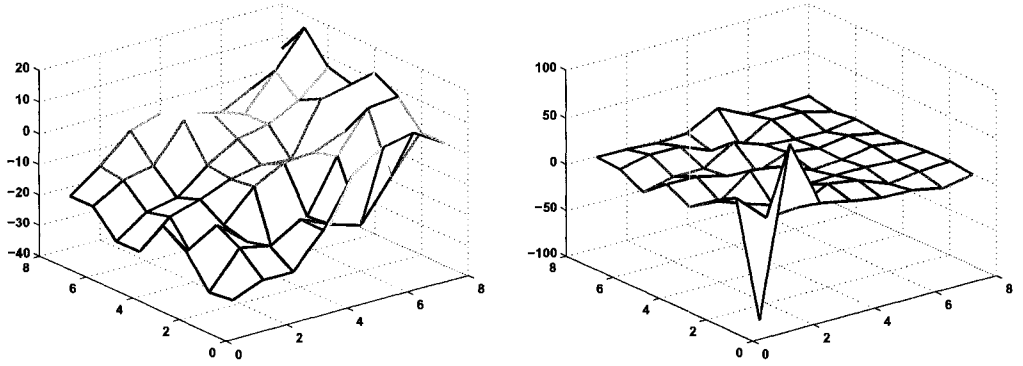


Figure 4.5: 8x8 Sub-block Extraction for *Einstein*

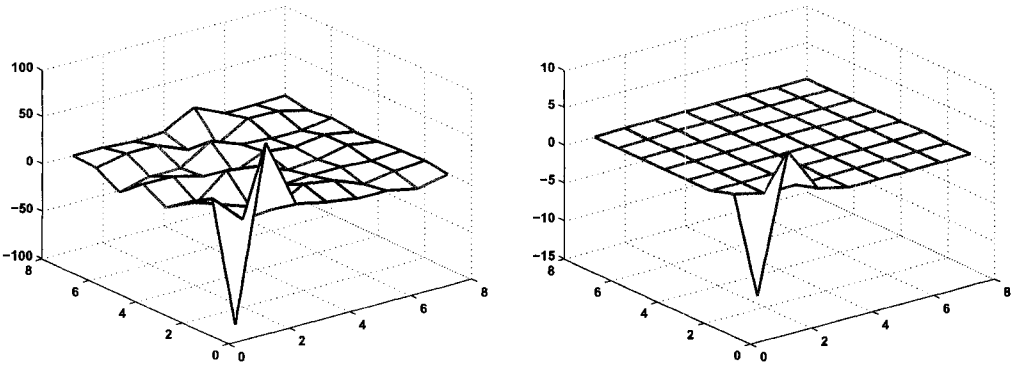
1.0863	-0.2137	-0.2002	-0.1836	-0.1618	-0.1192	-0.0837	-0.0424
-0.2004	0.0541	0.0264	0.0286	0.0318	0.0199	0.0157	0.0050
-0.2099	0.0289	0.0497	0.0383	0.0321	0.0219	0.0155	0.0079
-0.1843	0.0253	0.0410	0.0341	0.0268	0.0211	0.0165	0.0116
-0.1588	0.0313	0.0245	0.0271	0.0225	0.0275	0.0079	0.0113
-0.1269	0.0227	0.0231	0.0284	0.0198	0.0091	0.0092	-0.0006
-0.0839	0.0284	0.0164	0.0080	0.0058	0.0036	0.0081	0.0021
-0.0487	0.0240	-0.0041	0.0001	0.0107	0.0060	0.0100	-0.0013

Figure 4.6: Tchebichef Transformed Coefficients of 8x8 Sub-block for *Einstein*



(a) (b)

Figure 4.7: Mesh Plot Characteristics of (a) Level-shifted Inputs and (b) Transformed Coefficients of *Einstein*



(a) (b)

Figure 4.8: Mesh Plot Characteristics of the Quantized Sub-blocks of *Einstein* for (a) Lower and (b) Higher Quantization Values

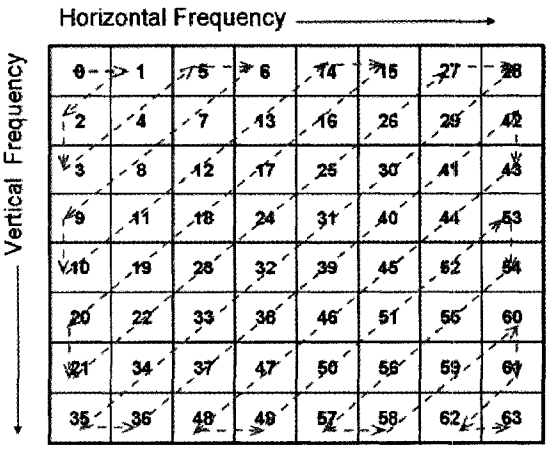
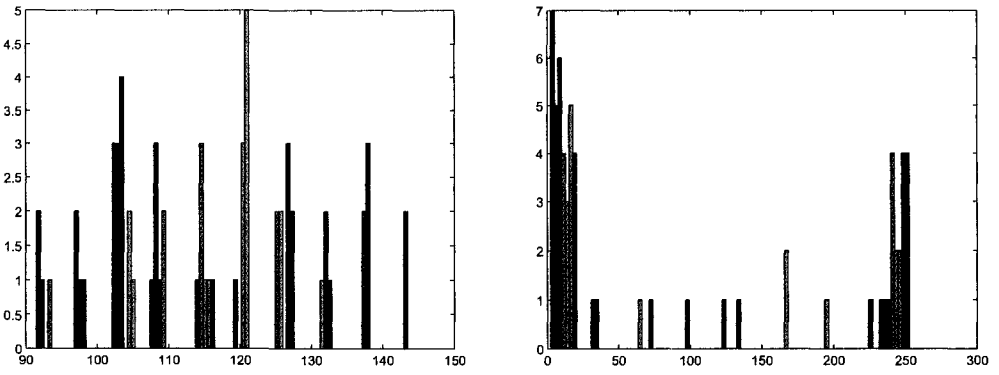
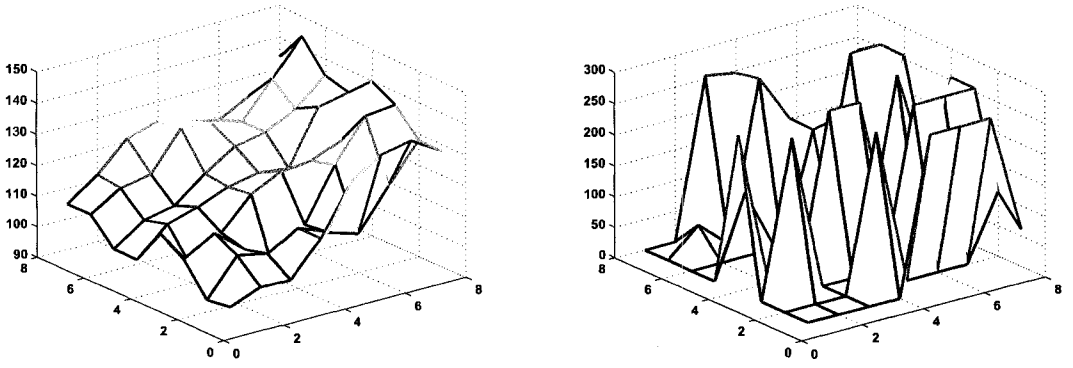


Figure 4.9: Example of Zigzag Scanning Sequence for 8x8 DTT



(a) (b)

Figure 4.10: Histogram Characteristics of the Decoded Sub-blocks of *Einstein* for (a) Lower and (b) Higher Quantization Values



(a) (b)

Figure 4.11: Characteristics of the Decoded Sub-blocks of *Einstein* for (a) Lower and (b) Higher Quantization Values

4.4 Experimental Results and Discussions

Using the scheme in Fig. 4.3 explained in the previous sub-section, we perform image compression on various gray scale images using 8×8 [19] and 4×4 [20] DTTs. The test images used in our experiments are shown in Figs. 4.12 (a) - (j). The images are (a) *Lena*, (b) *Einstein*, (c) *Airplane*, (d) *Mandrill*, (e) *Peppers*, (f) *Cameraman*, (g) *Grass*, (h) *Moon*, (i) *Leopard* and (j) *Texture Mosaic*.

All the images are in the TIFF format [32], except the images *Einstein* and *Leopard*, which are in the PGM format [33]. Images *Einstein*, *Leopard*, *Cameraman* and *Moon* are of dimension 256 x 256, and the rest of the images are of dimension 512 x 512. The types of images chosen for the experiment include natural, artificial, highly-textured and detailed, predictable, and non-predictable type of images. This is done to ensure that all types of images are considered for the performance evaluation of the transforms. The analysis of the experimental results has been carried out by using *image quality measures* meant for gray-scale compressed images [34]. *Objective* measures are more commonly employed in *image quality analysis* due to their simplicity and their popularity. However, it is necessary to study the performance based on the end-user. HVS models [35] have been developed and incorporated into coding and analysis from the *subjective* viewpoint of image compression. Hence, we evaluate the performance of DTT by using both subjective and objective measures.

4.4.1 Picture Quality Evaluation

The quality of perception of an image largely depends on the structural nature of the image. *Spectral activity measure (SAM)* and *spatial frequency measure (SFM)* [36] are quantities that identify the structural nature of an image. SAM gives a measure of image predictability in the spectral domain, and is defined as the ratio of the arithmetic mean to the geometric

mean of the discrete Fourier transform coefficients. It is given by

$$SAM = \frac{\frac{1}{M \cdot N} \sum_{i=0}^{M-1} \sum_{j=0}^{L-1} |F(i,j)|^2}{\left\{ \prod_{i=0}^{M-1} \prod_{j=0}^{L-1} |F(i,j)|^2 \right\}^{\frac{1}{M \cdot N}}}$$

and has a dynamic range of $[1, \infty)$. The SAM of an image is directly proportional to the predictability of the image. SFM indicates the overall activity level in a picture, defined by the *row frequency*, R and the *column frequency*, C. It is given by

$$SFM = \sqrt{R^2 + C^2}$$

where

$$R = \sqrt{\frac{1}{M \cdot N} \sum_{i=1}^{M} \sum_{j=2}^{L} (x_{i,j} - x_{i,j-1})^2},$$

and

$$C = \sqrt{\frac{1}{M \cdot N} \sum_{j=1}^{L} \sum_{i=2}^{M} (x_{i,j} - x_{i-1,j})^2}$$

An image with high frequency components has a large value of SFM, and an image with low predictability has a small value of SAM. For example, the *Mandrill* image has a very high SFM and a low SAM. It is not very well-correlated and contains a large amount of details. Therefore, images with a large SFM and a small SAM (closer to unity) are difficult to code, since they are highly active and are not very predictable.

By using SFM and SAM, it is possible to assess the performance of a discrete transform on different images. The test images in Figs. 4.12 (a)-(j) have been arranged below in the order of increasing SAM and decreasing SFM, i.e., the left-most quantities are difficult to

code and the right-most ones are relatively easier to code

$$\left.\begin{array}{c} \leftarrow\leftarrow\leftarrow\leftarrow\leftarrow\leftarrow\leftarrow\leftarrow \textit{Increasing level of coding difficulty} \leftarrow\leftarrow\leftarrow\leftarrow\leftarrow\leftarrow\leftarrow\leftarrow \\ \textbf{SAM} : \textit{Leopard} \ll \textit{Mandrill} \ll \textit{Grass} \ll \textit{Cameraman} \ll \textit{Einstein} \ll \textit{Moon} \\ \ll \textit{Peppers} \ll \textit{Lena} \ll \textit{Texture Mosaic} \ll \textit{Airplane} \\ \textbf{SFM} : \textit{Mandrill} \gg \textit{Leopard} \gg \textit{Cameraman} \gg \textit{Grass} \gg \textit{Einstein} \gg \textit{Peppers} \\ \gg \textit{Moon} \gg \textit{Airplane} \gg \textit{Lena} \gg \textit{Texture Mosaic} \end{array}\right\} \quad (4.2)$$

Picture quality evaluation (PQE) can be subjective or objective. While subjective measures are based on the psychovisual physics of human perception, objective measures deal with distortions that a human eye cannot capture. We use a new set of quality measures, in addition to the conventional metrics such as the peak signal to noise ratio (PSNR) and the root mean square error (RMSE). In [37], the authors have given a new set of quality measures such as *average difference (AD), structural content (SC), normalized cross-correlation (NK), maximum difference (MD), Laplacian mean squared error (LMSE), normalized absolute error (NAE), picture quality scale (PQS)* and *mean opinion score (MOS)*. Most of these image-quantifying parameters are recommended by the International Telecommunication Union-Radiocommunication Sector (ITU-R). The mathematical expressions for each of the above metrics are given in Table 4.1.

PSNR gives a quality measure for the image compression scheme used, and RMSE represents the average distortion in the recovered image. NK depicts the cross-correlation between the original and distorted images, normalized with respect to the undistorted image. LMSE is known to typify edge information, which impinges heavily on the image perception of the human eye. NAE represents the mean absolute error, normalized with respect to the undistorted image. PQS is a perception-based objective measure that incorporates the *human visual system* (HVS). Given the original ($x_{i,j}$) and distorted ($x'_{i,j}$) images, the local distortion maps and distortion factors are computed. Regression methods are then

Table 4.1: Quality Measures for PQE

Gain/ Error-based Measures	
$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[\hat{f}(i,j) - f(i,j)]^2$ $RMSE = \sqrt{(MSE)}$	$PSNR = 20log_{10}\left[\frac{2^n - 1}{RMSE}\right]$
Correlation-based Measures	
$NK = \frac{\sum_{i=1}^{M}\sum_{j=1}^{L} x_{i,j}x'_{i,j}}{\sum_{i=1}^{M}\sum_{j=1}^{L} x_{i,j}^2}$	$SC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{L} x_{i,j}^2}{\sum_{i=1}^{M}\sum_{j=1}^{L}(x'_{i,j})^2}$
HVS-based Measures	
$NAE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{L}\lvert x_{i,j} - x'_{i,j}\rvert}{\sum_{i=1}^{M}\sum_{j=1}^{L}\lvert x_{i,j}\rvert}$	$LMSE = \frac{\sum_{i=1}^{M-1}\sum_{j=1}^{L-1}(x_{i,j} - x'_{i,j})^2}{\sum_{i=1}^{M-1}\sum_{j=1}^{L-1} x_{i,j}^2}$
Rating-based Measures	
$PQS = b_0 + \sum_{j=1}^{J} b_j Z_j$	$MOS(k) = \frac{1}{n}\sum_{j=1}^{n} S(j,k)$
Difference-based Measures	
$AD = \frac{1}{M \cdot N}\sum_{i=1}^{M}\sum_{j=1}^{L} x_{i,j} - x'_{i,j}$	$MD = Max\lvert x_{j,k} - x'_{j,k}\rvert$

used to combine these factors into a single quantity which quantifies the image quality, taking local and global distortions into account. PQS is given by

$$PQS = b_0 + \sum_{j=1}^{J} b_j Z_j$$

where b_j are the partial regression coefficients and Z_j are the principal components. PQS values fall out of the valid range for very low quality images at low bit-rates. MOS [38] is

a perception-based subjective measure and is given by

$$MOS(k) = \frac{1}{n}\sum_{j=1}^{n} S(j,k)$$

where n denotes the number of observers and $S(j,k)$ is the score given by the j-th observer to the k-th image. The observers are asked to assign a score to each encoded image. Each score (from 1-5) is assigned according to the *impairment scale* shown in Table 4.2. For each encoded image, the scores are averaged to obtain the MOS for that image. A detailed account of PQS and MOS is given in [38].

Table 4.2: Impairment Scale for MOS

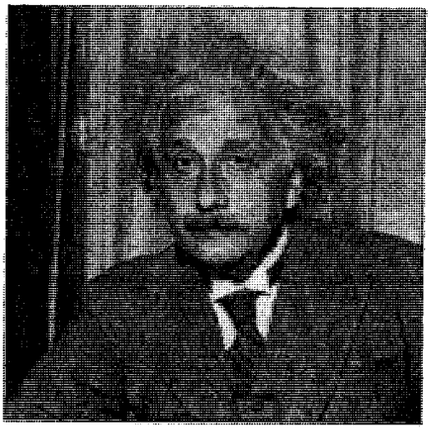
Scale	Impairment
5	imperceptible
4	perceptible, but not annoying
3	slightly annoying
2	annoying
1	very annoying

The Pearson product-moment (r) provides the correlation coefficient [36] defined as

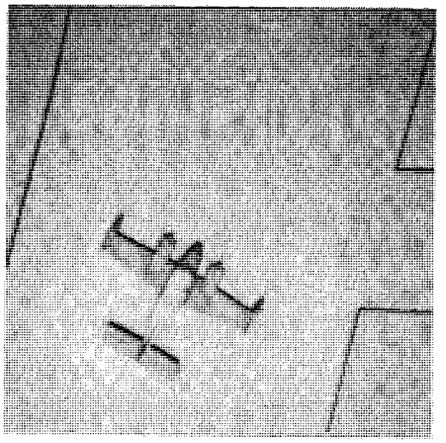
$$r = \frac{\sum_i (x_i - \overline{x})(x'_i - \overline{x'})}{\sqrt{\sum_i (x_i - \overline{x})^2 \sum_i (x'_i - \overline{x'})^2}} \tag{4.3}$$

where r gives the extent of correlation between the vectors x and x'. The closer 'r' is to ± 1, the better is the correlation.

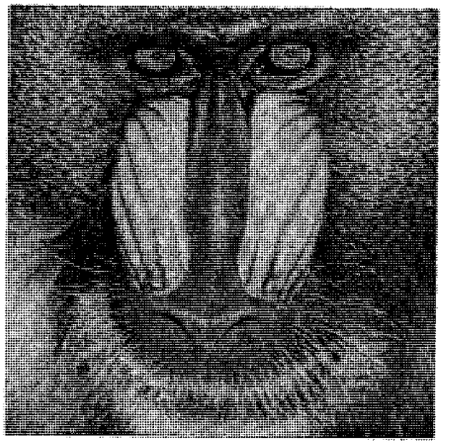
(a)



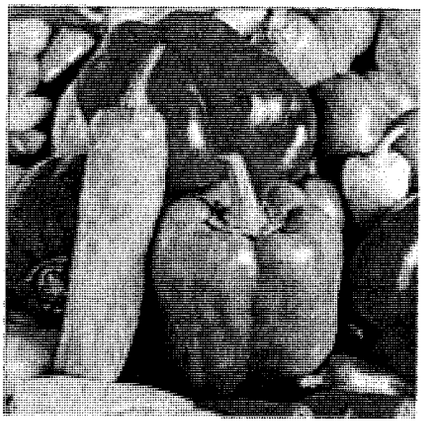
(b)



(c)



(d)



(e)

(f)

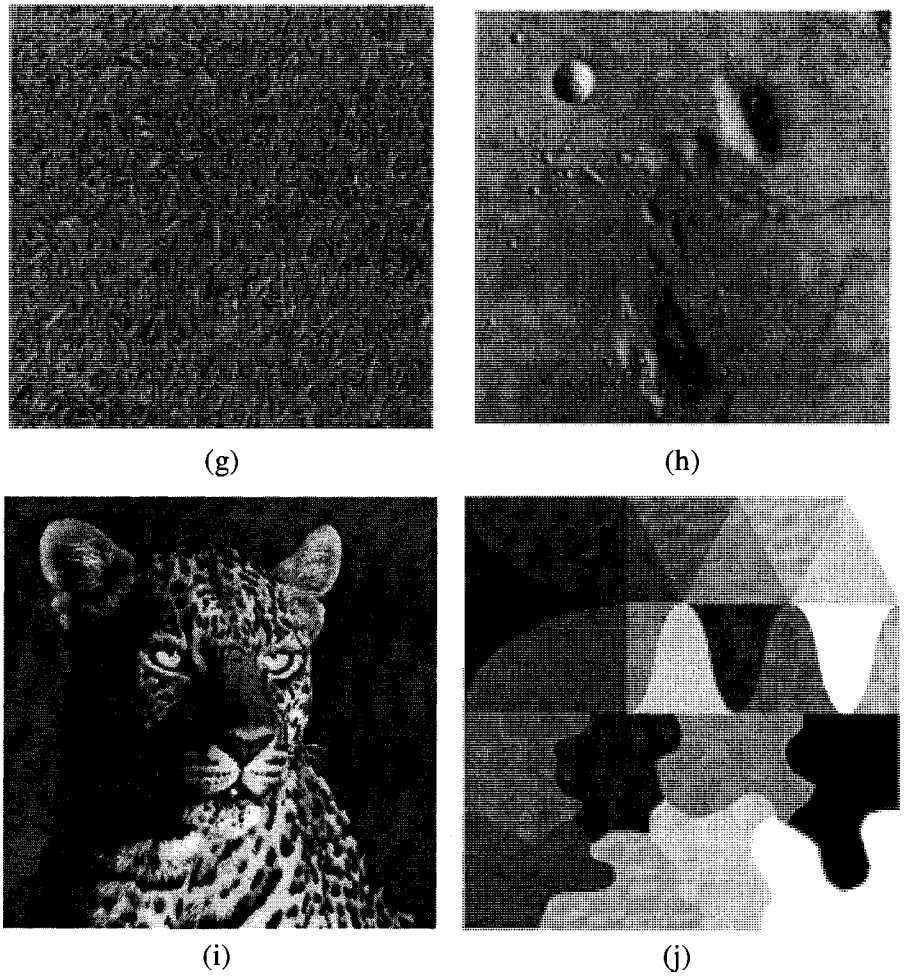


Figure 4.12: Original Test Images (a) *Lena*, (b) *Einstein*, (c) *Airplane*, (d) *Mandrill*, (e) *Peppers*, (f) *Cameraman*, (g) *Grass*, (h) *Moon*, (i) *Leopard* and (j) *Texture Mosaic*

4.4.2 Image Compression using 8x8 Discrete Tchebichef Transform

The performance of 8×8 DTT in image compression is analyzed in this sub-section. Since 8×8 DCT has a large presence in image compression algorithms, we compare the performance of 8×8 DTT with that of 8×8 DCT by using a set of quality measures specifically meant for gray scale image compression. We further analyze the results obtained with respect to the spatial and spectral properties of the images under consideration. We carry out compression of image blocks of size 8×8 using the image compression scheme shown in Fig. 4.3. The test images used are shown in Fig. 4.12. The performance evaluation is based on *subjective* and *objective* measures. Subjective tests using MOS, PQS, LMSE and NAE incorporate HVS characteristics in the performance analysis. In our experiment, 15 observers were asked to assign scores to images with different *impairments* according to the specifications in [39] for *Variant-1* using the *double-stimulus impairment scale* method. Tables 4.3 - 4.9 give the values of quality measures at different bit-rates for DCT and DTT for all the test images. Tabulated in the last column of all these tables, are the Pearson product-moment, r_{PQS}, which shows the extent of correlation of the various quality measures and PQS, that is,

$$r_{PQS} = corr(qualitymeasure, PQS)$$

where $qualitymeasure, PQS$ correspond to x and x' in (4.3).

In Table 4.10, MOS values at different bit-rates for some of the test images are given and columns $r_{MSE} - r_{PQS}$ contain the correlation measure (Pearson product-moment) of MOS with each quality measure, i.e.,

$$r_{qualitymeasure} = corr(qualitymeasure, MOS)$$

where $qualitymeasure$ and MOS correspond to x and x' in (4.3). Plots of PSNR against *bits per pixel* (bpp) obtained by performing 8x8 DTT compression are shown in Figs. 4.13

Table 4.3: Quality Measures at Different Bit-rates for 8x8 DTT-compressed Images of *Mandrill* *(SF:36.5146; SAM:100)*

	bpp	4.58	2.43	1.92	1.74	1.59	1.46	1.36	1.16	1.05	0.93	0.81	0.67	0.35	r_{PQS}
	MSE	3.06	32.01	54.71	65.97	76.78	87.19	96.98	120.19	134.72	152.88	175.93	207.31	321.56	-0.9992
	RMSE	1.61	3.03	3.84	4.24	4.63	5.02	5.40	6.35	7.01	7.92	9.10	10.98	21.04	-0.9132
	PSNR	43.27	33.08	30.75	29.94	29.28	28.73	28.26	27.33	26.84	26.29	25.68	24.96	23.06	0.8254
D	**AD**	0	-0.01	0	0	-0.01	0	0	0	0	0	-0.02	0	0	0.5390
T	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	-0.9797
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.9950
	MD	9.00	32.00	42.00	53.00	55.00	63.00	69.00	86.00	85.00	95.00	95.00	92.00	111.00	-0.9528
	LMSE	0.02	0.20	0.32	0.37	0.42	0.47	0.51	0.59	0.63	0.68	0.74	0.80	0.96	-0.9193
	NAE	0.01	0.03	0.04	0.05	0.05	0.05	0.06	0.06	0.07	0.07	0.08	0.08	0.10	-0.9808
	PQS	5.33	4.84	4.59	4.49	4.38	4.28	4.19	3.97	3.83	3.67	3.44	3.14	2.04	1.0000
	bpp	4.49	2.35	1.87	1.70	1.55	1.44	1.35	1.15	1.06	0.94	0.82	0.69	0.38	r_{PQS}
	MSE	3.74	35.79	58.34	69.05	79.16	88.82	97.79	118.89	131.78	147.77	167.43	193.53	295.79	-0.9980
	RMSE	1.64	3.13	3.94	4.34	4.73	5.11	5.46	6.38	6.96	7.82	8.93	10.65	19.51	-0.9218
	PSNR	42.41	32.59	30.47	29.74	29.15	28.65	28.23	27.38	26.93	26.43	25.89	25.26	23.42	0.8202
D	**AD**	0	0	0	0	0	-0.01	0.01	0	0	0	0	0	-0.03	0.4933
C	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	-0.8635
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.9936
	MD	10.00	35.00	45.00	54.00	60.00	65.00	74.00	84.00	100.00	92.00	99.00	97.00	104.00	-0.9164
	LMSE	0.03	0.23	0.35	0.40	0.44	0.48	0.51	0.58	0.62	0.67	0.71	0.76	0.91	-0.9183
	NAE	0.01	0.04	0.04	0.05	0.05	0.05	0.06	0.06	0.07	0.07	0.07	0.08	0.10	-0.9818
	PQS	5.37	4.92	4.71	4.61	4.51	4.42	4.34	4.14	4.02	3.86	3.65	3.37	2.26	1.0000

(a), (b) and (c) for the images *Mandrill*, *Lena* and *Einstein* respectively. Plots of RMSE against compression ratio as a result of 8x8 DTT compression are shown in Figs. 4.14 (a),(b) and (c) for the images *Mandrill*, *Lena* and *Einstein* respectively. Plots of LMSE against *bits per pixel* (bpp) obtained by performing 8x8 DTT compression are shown in Figs. 4.15 (a), (b) and (c) for the images *Mandrill*, *Lena* and *Einstein* respectively. Plots of NAE against *bits per pixel* (bpp) obtained by performing 8x8 DTT compression are shown in Figs. 4.16 (a), (b) and (c) for the images *Mandrill*, *Lena* and *Einstein* respectively.

Table 4.4: Quality Measures at Different Bit-rates for 8x8 DTT-compressed Images of *Grass* (*SF:25.2; SAM:117.94*)

	bpp	4.15	2.04	1.53	1.35	1.21	1.09	0.99	0.89	0.79	0.47	0.35	0.24	0.13	r_{PQS}
	MSE	2.78	26.36	44.74	54.27	63.54	73.17	82.63	93.77	106.62	175.28	217.95	277.14	366.98	-0.9976
	RMSE	1.63	3.29	4.41	4.98	5.56	6.19	6.81	7.56	8.50	14.44	19.39	28.40	51.23	-0.8873
	PSNR	43.69	33.92	31.62	30.79	30.10	29.49	28.96	28.41	27.85	25.69	24.75	23.70	22.48	0.8653
D	**AD**	-0.00	-0.00	-0.00	-0.01	-0.01	-0.00	-0.02	0.00	0.01	-0.01	-0.00	-0.02	-0.07	0.7558
T	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.01	-0.9121
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.98	0.97	0.9756
	MD	8.00	26.00	34.00	37.00	41.00	42.00	43.00	49.00	63.00	72.00	88.00	98.00	120.00	-0.9941
	LMSE	0.05	0.38	0.55	0.62	0.68	0.74	0.80	0.86	0.92	1.16	1.25	1.32	1.35	-0.7799
	NAE	0.01	0.04	0.06	0.06	0.07	0.07	0.08	0.08	0.09	0.11	0.12	0.14	0.16	-0.9898
	PQS	5.11	4.20	3.71	3.47	3.26	3.05	2.84	2.61	2.36	1.05	0.29	-0.72	-2.23	1.0000
	bpp	4.39	2.22	1.74	1.58	1.45	1.35	1.26	1.18	1.09	1.01	0.91	0.80	0.36	r_{PQS}
	MSE	3.83	28.80	41.36	46.78	51.56	56.02	60.31	64.71	70.19	76.29	84.69	95.81	189.27	-0.9932
	RMSE	1.53	3.04	3.88	4.27	4.65	5.01	5.33	5.70	6.17	6.68	7.41	8.42	18.49	-0.9245
	PSNR	42.30	33.54	31.97	31.43	31.01	30.65	30.33	30.02	29.67	29.31	28.85	28.32	25.36	0.8594
D	**AD**	-0.01	-0.01	-0.00	0.00	-0.01	0.00	-0.02	-0.00	0.01	-0.00	-0.03	0.03	-0.07	0.5890
C	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.9516
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.99	0.99	0.99	0.9903
	MD	9.00	31.00	33.00	39.00	39.00	43.00	46.00	46.00	52.00	46.00	51.00	54.00	79.00	-0.9831
	LMSE	0.07	0.44	0.56	0.60	0.63	0.66	0.69	0.71	0.74	0.77	0.80	0.85	1.08	-0.9007
	NAE	0.02	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.07	0.07	0.08	0.08	0.12	-0.9880
	PQS	5.31	4.77	4.50	4.37	4.24	4.12	4.01	3.89	3.74	3.57	3.35	3.05	0.88	1.0000

Figs. 4.17 (a), (b), (c) and (d) show the performance plots of PQS against bpp for the images *Mandrill*, *Lena*, *Einstein* and *Airplane* respectively. Due to the large number of plots generated from the variety of quality measures used, we selectively present the plots of quality measures that exhibit good correlation with subjective quality measures (some have been plotted in log scale for better perception).

In the sub-sections that follow, we first give a preliminary analysis for detailed account of experimental results for subjective and objective measures of quality for DTT and DCT.

Table 4.5: Quality Measures at Different Bit-rates for 8x8 DTT-compressed Images of *Cameraman* *(SF:28.86; SAM:184.22)*

	bpp	2.96	1.37	1.05	0.95	0.87	0.80	0.74	0.69	0.63	0.56	0.50	0.43	0.36	0.28	r_{PQS}
	MSE	1.91	13.55	23.48	28.51	33.12	37.89	42.50	47.84	53.96	62.21	72.21	86.04	106.46	136.12	-0.9443
	RMSE	2.37	5.12	6.67	7.38	8.02	8.78	9.49	10.22	11.18	12.41	13.92	16.20	19.62	25.03	-0.8997
	PSNR	45.33	36.81	34.42	33.58	32.93	32.35	31.85	31.33	30.81	30.19	29.54	28.78	27.86	26.79	0.9503
D	**AD**	0.00	-0.00	0.00	0.01	0.00	-0.02	-0.01	-0.02	-0.02	-0.05	-0.04	-0.08	-0.03	-0.09	0.2929
T	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.8357
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.9098
	MD	7.00	26.00	36.00	39.00	47.00	46.00	59.00	59.00	67.00	64.00	67.00	83.00	122.00	121.00	-0.9815
	LMSE	0.02	0.15	0.25	0.29	0.32	0.36	0.39	0.43	0.47	0.51	0.57	0.63	0.72	0.84	-0.9984
	NAE	0.01	0.02	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.06	-0.9949
	PQS	4.53	2.96	2.17	1.85	1.55	1.24	1.03	0.76	0.43	0.08	-0.37	-0.85	-1.49	-2.18	1.0000
	bpp	2.99	1.44	1.02	0.94	0.87	0.81	0.76	0.70	0.64	0.58	0.51	0.43	0.35	0.25	r_{PQS}
	MSE	2.50	17.12	31.17	35.52	39.67	43.52	47.61	52.96	58.62	66.23	76.24	89.96	110.25	147.67	-0.9585
	RMSE	2.34	4.88	6.85	7.44	8.05	8.61	9.23	10.05	10.94	12.15	13.85	16.33	20.27	27.96	-0.9336
	PSNR	44.15	35.80	33.19	32.63	32.15	31.74	31.35	30.89	30.45	29.92	29.31	28.59	27.71	26.44	0.9361
D	**AD**	-0.00	0.01	-0.01	-0.01	-0.03	-0.03	-0.02	-0.00	-0.00	-0.04	-0.10	-0.10	-0.05	-0.16	0.2353
C	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.5041
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	0.9170
	MD	8.00	31.00	44.00	51.00	54.00	52.00	55.00	73.00	77.00	80.00	86.00	102.00	104.00	111.00	-0.9895
	LMSE	0.03	0.21	0.34	0.37	0.40	0.43	0.46	0.50	0.53	0.57	0.62	0.67	0.75	0.85	-0.9889
	NAE	0.01	0.02	0.03	0.03	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.06	-0.9860
	PQS	4.50	2.96	2.00	1.72	1.48	1.27	1.05	0.82	0.51	0.23	-0.13	-0.73	-1.38	-2.37	1.0000

Later on, the performance of DTT is analyzed image-wise based on the structural properties of images explained in the Sub-section 4.4.1. We also present a HVS-based performance evaluation. For all the above, we refer to Figs. 4.13-4.17 for the performances of quality measure, and to Tables 4.3-4.9 for quality measures tabulated for various compression levels measured at the respective bpp. The last column of every table gives the correlation of the quality measure with PQS. We refer to Table 4.10 for MOS values and its correlation with quality measures.

Table 4.6: Quality Measures at Different Bit-rates for 8x8 DTT-compressed Images of *Lena* *(SF:14.0436; SAM:909.09)*

	bpp	2.70	1.09	0.82	0.74	0.61	0.57	0.49	0.44	0.39	0.34	0.29	0.23	0.17	0.14	r_{PQS}
	MSE	2.42	9.45	12.84	14.37	15.86	17.27	20.15	22.05	24.21	27.08	30.97	36.91	46.21	64.31	-0.9868
	RMSE	3.05	9.06	12.79	14.49	16.10	17.72	20.99	23.12	26.00	29.74	34.48	42.63	55.86	80.38	-0.9817
	PSNR	44.29	38.38	37.05	36.56	35.76	35.43	34.69	34.29	33.80	33.22	32.46	31.48	30.05	29.24	0.3685
D	**AD**	-0.00	-0.00	0.00	0.00	-0.01	0.01	0.00	0.00	0.00	0.01	-0.00	0.02	-0.02	-0.03	0.3685
T	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.0985
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.9721
	MD	8.00	33.00	32.00	38.00	38.00	42.00	41.00	48.00	57.00	53.00	59.00	58.00	78.00	79.00	-0.9709
	LMSE	0.18	0.52	0.63	0.67	0.71	0.74	0.80	0.84	0.88	0.92	0.98	1.05	1.17	1.33	-0.9640
	NAE	0.01	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04	-0.9959
	PQS	5.27	4.68	4.37	4.25	4.11	4.00	3.75	3.60	3.44	3.22	2.93	2.52	1.94	0.87	1.0000
	bpp	2.65	1.11	0.85	0.76	0.64	0.59	0.50	0.46	0.41	0.36	0.31	0.25	0.18	0.14	r_{PQS}
	MSE	2.71	9.11	12.04	13.40	15.92	17.07	19.97	21.76	24.29	27.71	32.87	41.55	59.16	73.48	-0.9739
	RMSE	2.81	6.73	8.75	9.74	11.66	12.52	14.76	16.13	18.07	20.63	24.35	30.23	42.14	51.62	-0.9826
	PSNR	43.81	38.54	37.33	36.86	36.46	36.11	35.50	35.13	34.75	34.28	33.70	32.96	31.95	30.41	0.9416
D	**AD**	-0.00	0.00	-0.00	-0.00	0.01	0.01	0.00	-0.01	0.01	-0.02	0.01	0.01	0.04	-0.05	0.7460
C	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.6730
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.9402
	MD	9.00	25.00	39.00	38.00	45.00	42.00	47.00	57.00	55.00	54.00	55.00	62.00	89.00	98.00	-0.9785
	LMSE	0.20	0.50	0.59	0.63	0.68	0.71	0.76	0.79	0.82	0.87	0.93	1.02	1.16	1.24	-0.9740
	NAE	0.01	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.03	0.03	0.03	0.04	0.04	0.05	-0.9961
	PQS	5.32	4.78	4.50	4.39	4.15	4.05	3.79	3.64	3.42	3.15	2.77	2.21	1.21	0.50	1.0000

Preliminary Analysis

Since a large number of tables for various quality measures have been given, for ease of view and understanding, we present a preliminary analysis based on a short table with only a few results in Table 4.11 for images *Einstein*, *Lena* and *Airplane*. In this table, by comparing the values of MSE, LMSE and MD for DTT and DCT, we see that the reconstruction accuracy of DTT is better than that of DCT. Also, by observing the values

Table 4.7: Quality Measures at Different Bit-rates for 8x8 DTT-compressed Images of *Moon* (*SF:15.61; SAM:448.49*)

	bpp	3.26	1.28	0.81	0.66	0.56	0.48	0.42	0.36	0.31	0.26	0.21	0.17	0.13	r_{PQS}
	MSE	2.75	22.04	34.13	39.33	43.86	48.29	52.17	56.52	61.03	66.26	72.30	79.68	88.96	-0.9624
	RMSE	2.06	5.25	8.28	10.13	11.90	13.86	15.98	18.49	21.72	26.07	31.36	39.14	49.83	-0.8542
	PSNR	43.73	34.70	32.80	32.18	31.71	31.29	30.96	30.61	30.28	29.92	29.54	29.12	28.64	0.9306
D	**AD**	0.00	-0.00	0.01	0.00	-0.00	-0.01	0.01	0.02	0.01	0.02	0.01	-0.08	0.04	-0.4555
T	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.9232
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.9484
	MD	8.00	23.00	33.00	35.00	38.00	48.00	48.00	49.00	50.00	56.00	67.00	75.00	82.00	-0.9572
	LMSE	0.10	0.60	0.79	0.84	0.88	0.93	0.96	0.99	1.02	1.05	1.07	1.10	1.12	-0.8874
	NAE	0.01	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.05	0.05	-0.9892
	PQS	4.07	1.14	-0.30	-0.90	-1.44	-1.90	-2.30	-2.72	-3.14	-3.69	-4.23	-4.83	-5.50	1.0000
	bpp	3.57	1.51	1.11	0.80	0.73	0.66	0.59	0.53	0.46	0.38	0.30	0.22	0.14	r_{PQS}
	MSE	3.76	21.65	28.73	36.25	38.28	40.32	42.82	45.60	49.38	54.06	60.63	70.17	86.95	-0.9580
	RMSE	1.88	4.46	6.05	8.43	9.24	10.13	11.33	12.61	14.68	17.61	22.42	30.56	48.22	-0.8841
	PSNR	42.38	34.78	33.55	32.54	32.30	32.08	31.81	31.54	31.20	30.80	30.30	29.67	28.74	0.9378
D	**AD**	0.00	-0.00	-0.01	-0.01	0.02	-0.00	-0.00	0.01	-0.07	-0.13	-0.01	0.09	-0.04	-0.5972
C	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.6573
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.9195
	MD	8.00	27.00	42.00	51.00	53.00	59.00	63.00	60.00	60.00	61.00	77.00	82.00	85.00	-0.9894
	LMSE	0.14	0.65	0.77	0.85	0.87	0.89	0.90	0.92	0.95	0.97	1.00	1.03	1.08	-0.8750
	NAE	0.01	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	-0.9878
	PQS	4.02	1.57	0.66	-0.29	-0.57	-0.81	-1.17	-1.49	-1.97	-2.49	-3.19	-3.98	-5.12	1.0000

of PSNR and NK for DTT and DCT, we see that DTT-reconstructed images have a higher gain and a better correlation with the original image as compared to those reconstructed using DCT. It should be noted that lower bit-rate results of DTT have been compared to corresponding higher bit-rate results of DCT. As a result of this comparison, the higher performance of DTT when compared with DCT, reflects the better coding efficiency and image reconstruction quality of DTT. Next, we give a detailed analysis of the performance of DTT for each quality measure.

Table 4.8: Quality Measures at Different Bit-rates for 8x8 DTT-compressed Images of *Einstein* (*SF:20.8394; SAM:344.8*)

	bpp	2.91	1.14	0.80	0.69	0.61	0.55	0.50	0.46	0.41	0.37	0.32	0.27	0.22	r_{PQS}
	MSE	2.35	15.52	24.13	28.08	31.69	35.37	38.64	42.48	47.27	53.02	60.34	70.61	86.08	-0.9336
	RMSE	2.37	6.01	8.64	9.92	11.22	12.54	13.72	14.99	16.70	18.80	21.56	25.63	31.40	-0.8708
	PSNR	44.42	36.22	34.31	33.65	33.12	32.64	32.26	31.85	31.38	30.89	30.32	29.64	28.78	0.9528
D	**AD**	-0.01	-0.02	-0.02	-0.03	-0.06	-0.03	-0.06	-0.08	-0.05	-0.05	-0.06	-0.03	-0.08	-0.4176
T	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.6557
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.8249
	MD	7.00	25.00	35.00	37.00	43.00	41.00	45.00	50.00	60.00	64.00	60.00	64.00	90.00	-0.9728
	LMSE	0.04	0.21	0.28	0.31	0.33	0.36	0.38	0.40	0.43	0.45	0.48	0.52	0.57	-0.9910
	NAE	0.01	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.05	0.06	0.06	-0.9853
	PQS	4.26	2.35	1.46	1.08	0.71	0.43	0.22	-0.11	-0.50	-0.95	-1.46	-2.12	-3.08	1.0000
	bpp	3.14	1.40	1.09	0.83	0.77	0.72	0.66	0.60	0.54	0.47	0.40	0.32	0.23	r_{PQS}
	MSE	3.06	14.97	20.14	27.41	29.78	32.16	34.87	37.76	41.82	47.58	55.78	69.48	94.20	-0.9503
	RMSE	2.19	4.93	6.31	8.25	8.89	9.58	10.50	11.50	12.84	14.69	17.42	21.67	30.52	-0.9414
	PSNR	43.27	36.38	35.09	33.75	33.39	33.06	32.71	32.36	31.92	31.36	30.67	29.71	28.39	0.9468
D	**AD**	-0.01	-0.02	-0.01	-0.02	-0.04	-0.04	-0.06	-0.03	-0.04	0.03	0.05	-0.03	-0.17	-0.7096
C	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.4111
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.8983
	MD	8.00	30.00	38.00	43.00	50.00	48.00	56.00	57.00	64.00	53.00	61.00	74.00	79.00	-0.9928
	LMSE	0.06	0.23	0.28	0.35	0.38	0.40	0.41	0.42	0.44	0.49	0.54	0.70	0.75	-0.9870
	NAE	0.01	0.03	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.05	0.05	0.06	0.07	-0.9901
	PQS	4.28	2.71	2.12	1.41	1.19	0.96	0.70	0.44	0.12	-0.40	-0.95	-1.83	-3.21	1.0000

Correlation-based Measures

NK and SC are measures of correlation of the compressed/ distorted image with respect to the original/ unimpaired image. From the 6th and 5th rows of Tables 4.3 - 4.9, we observe that DTT and DCT have almost the same values for NK and SC respectively across various images. From the last column r_{PQS} of Tables 4.3 - 4.9 and from columns r_{NK} and r_{SC} of Table 4.10, it is seen that NK and SC correlate very poorly with PQS and MOS respectively.

Table 4.9: Quality Measures at Different Bit-rates for 8x8 DTT-compressed Images of *Airplane* (*SF:15.1025; SAM:5000*)

	bpp	1.64	0.53	0.41	0.36	0.33	0.31	0.28	0.26	0.24	0.23	0.20	0.18	0.15	r_{PQS}
	MSE	1.52	5.97	8.70	10.18	11.61	13.06	14.56	16.02	17.90	20.60	23.72	28.80	36.79	-0.9392
	RMSE	3.94	12.12	15.90	17.71	19.40	21.13	22.84	24.47	26.54	28.45	31.87	36.22	42.18	-0.9309
	PSNR	46.31	40.37	38.73	38.05	37.48	36.97	36.50	36.08	35.60	34.99	34.38	33.54	32.47	0.9628
D	**AD**	0.00	0.05	0.03	0.04	0.03	-0.01	0.00	0.10	0.03	0.07	0.06	0.06	-0.00	0.0620
T	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.2897
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.7118
	MD	6.00	19.00	27.00	29.00	36.00	36.00	36.00	39.00	42.00	48.00	53.00	64.00	70.00	-0.9873
	LMSE	0.05	0.14	0.18	0.20	0.22	0.25	0.27	0.29	0.31	0.35	0.38	0.44	0.54	-0.9876
	NAE	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	0.02	-0.9915
	PQS	4.81	3.97	3.59	3.44	3.30	3.14	2.99	2.89	2.71	2.51	2.29	1.92	1.42	1.0000
	bpp	1.90	0.79	0.61	0.55	0.50	0.43	0.40	0.36	0.33	0.30	0.26	0.22	0.18	r_{PQS}
	MSE	1.65	4.90	6.68	7.66	8.68	10.72	11.83	12.99	14.44	16.72	19.66	25.09	32.83	-0.9424
	RMSE	3.39	8.15	10.60	11.76	12.93	15.10	16.24	17.89	19.42	21.63	24.66	29.29	35.72	-0.9737
	PSNR	45.96	41.23	39.89	39.29	38.75	37.83	37.40	36.99	36.54	35.90	35.20	34.14	32.97	0.9719
D	**AD**	0.01	0.01	0.01	0.00	0.02	-0.00	-0.00	0.06	-0.03	-0.03	0.03	0.09	-0.13	0.9331
C	**SC**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.9314
T	**NK**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	-0.9042
	MD	8.00	25.00	27.00	30.00	31.00	34.00	36.00	34.00	43.00	49.00	54.00	68.00	55.00	-0.9797
	LMSE	0.07	0.15	0.18	0.20	0.22	0.28	0.30	0.32	0.33	0.34	0.37	0.43	0.63	-0.9804
	NAE	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.02	0.02	-0.9880
	PQS	4.87	4.20	3.94	3.81	3.69	3.46	3.35	3.22	3.09	2.90	2.64	2.30	1.93	1.0000

Gain/ Error-based Measures

PSNR: From the 3rd row of Tables 4.3 - 4.9 and Fig. 4.13, we can see that the PSNR of DTT is consistently more than that of DCT. There is over one dB gain in the DTT domain as compared to the DCT domain for most images except for the *Mandrill* image, where their performances are almost similar. Also, from the corresponding values in the last column r_{PQS} of Tables 4.3 - 4.9, and from the column r_{PSNR} of Table 4.10, it is seen that for most of

the images, especially the *Grass* image, PSNR does not correlate well with PQS and MOS for both DCT and DTT.

RMSE: From the 2nd row of Tables 4.3 - 4.9 and Fig. 4.14, we notice that RMSE of DTT is less than that of DCT for images *Einstein*, *Peppers*, *Lena*, *Airplane* and *Moon*. For *Mandrill*, *Grass* and *Cameraman* images, by and large, RMSE of DTT is slightly less than that of DCT at higher compression ratios and is almost equal to that of DCT at lower compression ratios. From the corresponding r_{PQS} values in the last few columns of Tables 4.3 - 4.9, it is seen that for all the images, RMSE does not correlate well with PQS for both DCT and DTT (except for the *Peppers* and *Lena* images).

AD: From the 4th row of Tables 4.3 - 4.9, we can deduce that AD of DTT is lower than that of DCT for *Einstein*. It performs similar to DCT for *Mandrill*, *Peppers*, *Moon* and *Lena* images (at moderate and high bpp) and is higher than that of DCT for the *Airplane* image. From the corresponding values in the last column r_{PQS} of Tables 4.3 - 4.9 and from the column r_{AD} of Table 4.10, we see that AD correlates very poorly with PQS and MOS respectively.

MD: From the 7th row of Tables 4.3 - 4.9, we can see that MD of DTT is lower than that of DCT for *Einstein*, *Moon*, *Cameraman*, *Peppers* and *Lena* images at all bpp and for *Cameraman*, *Grass* and *Airplane* images at moderate and high bpp. DTT performs almost like DCT for *Mandrill* and *Grass* images. Also, from the corresponding values in the last column r_{PQS} of Tables 4.3 - 4.9 and from the column r_{MD} of Table 4.10, it is seen that MD does not correlate well with PQS (especially for the *Moon* image) and MOS respectively.

MSE: From the corresponding values in the last column r_{PQS} of Tables 4.3 - 4.9 and from the column r_{MSE} of Table 4.10, we can see that MSE correlates well with PQS but not with MOS respectively (especially for *Einstein*).

HVS-based Measures

LMSE: From the 8th row of Tables 4.3 - 4.9 and Fig. 4.15, we see that LMSE of DTT is lower than that of DCT for *Einstein*, *Peppers*, *Lena*, *Airplane* and *Grass* images at higher bpp and for *Moon* and *Cameraman* images at all bpp. LMSE of DTT is on par with that of DCT for *Mandrill* image. From the corresponding values in the last column r_{PQS} of Tables 4.3 - 4.9 and from the column r_{LMSE} of Table 4.10, we can see that LMSE does not correlate well with PQS and MOS respectively.

NAE: From the 9th row of Tables 4.3 - 4.9 and Fig. 4.16, we observe that NAE for DTT is lower than that of DCT for *Einstein*, *Moon*, *Cameraman*, *Peppers*, *Lena*, *Airplane* and *Grass* images. DTT performs almost similar to DCT for *Mandrill* image. From the corresponding values in the last column r_{PQS} of Tables 4.3 - 4.9 and from the column r_{NAE} of Table 4.10, we see that NAE correlates well with PQS and correlates marginally with MOS respectively.

PQS: From the 10th row of Tables 4.3 - 4.9 and Fig. 4.17, we see that PQS for DTT is higher than that of DCT for *Lena* and *Airplane* images. PQS of DTT is almost equal to DCT for *Einstein* and *Cameraman* images and is worse than that of DCT for *Mandrill*, *Peppers* (at high bpp) and *Grass* images. From the column r_{PQS} of Table 4.10, we see that PQS correlates quite well with MOS for all images except for *Einstein* and *Mandrill* images.

Rating-based Measures

MOS: From Table 4.10, we see that MOS ratings for DTT are higher than DCT for *Lena* image. MOS ratings for DTT are lower than DCT for *Einstein*, *Airplane* and *Mandrill* images at low bpp and for *Peppers* at moderate bpp, and overall for *Grass* and *Cameraman* images.

Thus far, by perusing the results for all the quality measures, it is evident that the

performance of DTT is almost similar to that of DCT for images with high SFM and greatly improves as the SFM of the image decreases.

We now further analyze the results we have obtained. For greater accuracy and consistency [37], we need to

(i) consider those quality measures which have the best correlation with the subjective measures PQS/ MOS and

(ii) obviate a quality measure for which the sign of the correlation coefficient changes (for the same quality measure across different techniques).

AD and SC exhibit the most poor correlation for all the test images. NK correlates poorly for *Einstein* and *Airplane* images. LMSE has a low value of correlation coefficient for *Grass* image. The correlation coefficient pertaining to MD changes sign for *Moon*, *Airplane* and *Lena* images. Hence, it is not suitable to use AD, SC, NK and MD quality measures for our purpose. We inspect the cases of quality measures for which DTT is outperformed by DCT based on (*i*) stated above. These are MD for *Mandrill* image, AD for *Lena* image, NK and AD for *Airplane* image, MD for *Cameraman* image, LMSE, NAE, AD, MD and SC for *Grass* image and SC for *Moon* image. In light of the stipulations mentioned above, it is apparent that we have to examine only the cases of NAE-*Grass* image and PSNR-*Grass* image in the range of low bpp as shown in Fig. 4.18. Hence, except for these cases, the overall performance of DTT is similar to or better than that of DCT. This shows that DTT has a competitive performance with respect to that of DCT.

Summary

The results reinstate some well-known observations

(a) RMSE increases with an increase in the number of higher frequency components in the image, which corresponds to increasing SFM.

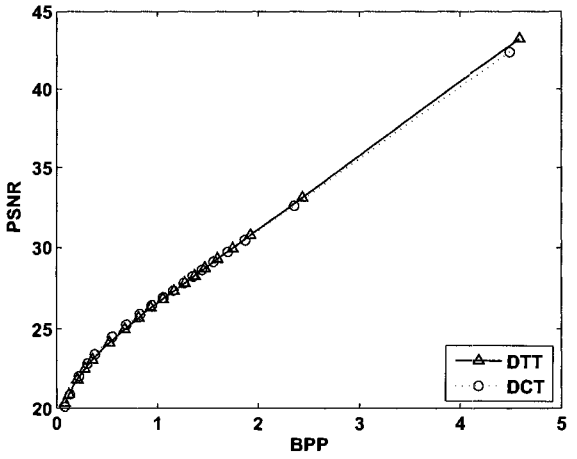
(b) PQS decreases with increasing SAM of images.

Another noteworthy observation is that the LMSE performance of DTT for most of the images is relatively much better than that of DCT for most of the images compared to other error measures. This is a desirable feature for subjective evaluation, since LMSE pertains to edge information, which the HVS is sensitive to.

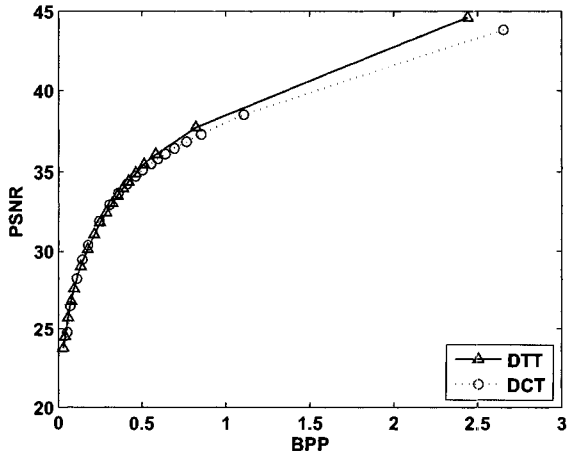
By and large, the perceptual-based performance measures (PQS, MOS, LMSE, NAE) of DCT and DTT are close to one another for the *Mandrill* image, and for the rest of the images, DTT performs better than DCT. However, from the point of view of objective measures, DTT scores over DCT in almost all the cases.

Table 4.10: MOS Performances at Different Bit-rates for 8x8 DTT-compressed Images of all Test Images

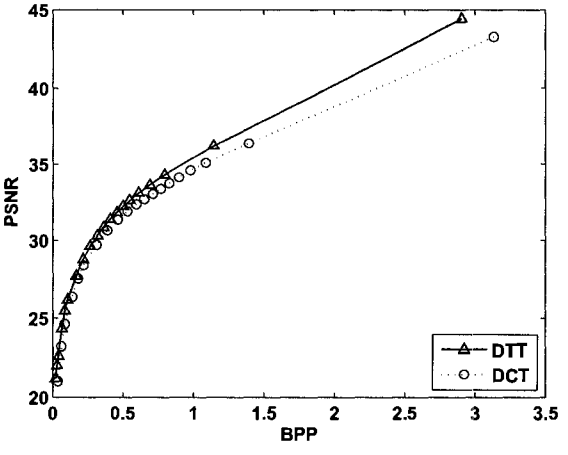
Image		MOS Values								r_{mse}	r_{psnr}	r_{ad}	r_{sc}	r_{nk}	r_{md}	r_{lmse}	r_{nae}	r_{pqs}
Lena	**bpp**	0.05	0.10	0.17	0.34	0.53	0.61	0.74	0.82									
	DTT	1.63	2.63	4.00	4.38	4.75	4.75	4.75	4.50	-0.98	0.93	0.58	0.22	0.95	-0.97	-0.93	-0.97	0.97
	bpp	0.05	0.11	0.18	0.36	0.55	0.64	0.76	0.85									
	DCT	1.13	2.00	2.88	4.50	4.50	4.50	4.38	4.50	-0.94	0.97	0.76	0.90	0.90	-0.96	-0.97	-0.97	0.98
Mandrill	**bpp**	0.11	0.20	0.35	0.52	0.67	0.81	0.93	1.05									
	DTT	2.00	2.75	3.75	3.63	4.50	4.25	4.25	4.38	-0.96	0.92	0.62	-0.97	0.97	-0.98	-0.88	-0.95	0.96
	bpp	0.13	0.22	0.38	0.55	0.69	0.82	0.94	1.06									
	DCT	2.25	3.50	4.00	4.13	3.63	4.25	4.38	4.75	-0.91	0.88	-0.82	-0.94	0.92	-0.86	-0.86	-0.90	0.90
Peppers	**bpp**	0.06	0.10	0.21	0.51	0.62	0.70	1.00	2.90									
	DTT	1.63	2.75	3.63	4.75	4.63	4.75	4.38	4.50	-0.98	0.81	-0.48	-0.89	0.96	-0.95	-0.88	-0.94	0.97
	bpp	0.06	0.12	0.19	0.52	0.61	0.72	0.89	2.98									
	DCT	1.50	2.50	3.63	4.75	4.75	4.75	4.75	4.38	-0.95	0.81	0.78	0.80	0.30	-0.94	-0.86	-0.94	0.97
Grass	**bpp**	0.13	0.24	0.35	0.47	0.68	0.79	0.89	0.99									
	DTT	1.88	2.13	3.25	3.38	3.38	3.88	4.13	4.13	-0.96	0.95	0.75	-0.90	0.95	-0.94	-0.92	-0.96	0.96
	bpp	0.11	0.28	0.36	0.54	0.67	0.80	0.91	1.18									
	DCT	1.88	3.25	3.50	3.75	3.88	4.13	4.00	4.50	-0.99	0.95	0.07	-0.93	0.99	-0.97	-0.93	-0.98	0.98
Einstein	**bpp**	0.09	0.11	0.17	0.32	0.41	0.61	0.80	1.14									
	DTT	1.75	1.50	1.63	2.38	2.88	3.63	4.00	4.50	-0.88	0.97	0.79	-0.81	0.90	-0.82	-0.92	-0.94	0.95
	bpp	0.09	0.18	0.23	0.32	0.40	0.60	0.83	1.09									
	DCT	1.63	1.75	2.13	2.50	3.00	4.00	3.75	4.50	-0.83	0.96	-0.12	-0.31	0.70	-0.87	-0.92	-0.90	0.93
Airplane	**bpp**	0.13	0.20	0.26	0.33	0.36	0.41	0.53	1.64									
	DTT	1.38	1.88	2.63	2.88	3.00	3.13	3.50	3.63	-0.94	0.91	-0.24	-0.15	0.83	-0.96	-0.96	-0.96	0.97
	bpp	0.12	0.18	0.26	0.33	0.36	0.46	0.55	1.90									
	DCT	1.75	2.38	2.75	2.88	3.50	3.38	3.38	3.50	-0.96	0.83	0.90	0.88	-0.83	-0.91	-0.92	-0.95	0.93
Cameraman	**bpp**	0.07	0.09	0.20	0.43	0.50	0.63	0.87	0.95									
	DTT	1.75	1.38	1.63	2.88	2.63	3.13	3.75	3.75	-0.86	0.96	0.35	-0.68	0.79	-0.91	-0.96	-0.91	0.95
	bpp	0.08	0.11	0.21	0.43	0.51	0.64	0.87	0.94									
	DCT	1.38	1.63	2.38	3.13	3.38	3.38	3.75	4.00	-0.98	0.99	0.30	-0.32	0.92	-0.99	-0.99	-0.98	1.00



(a)

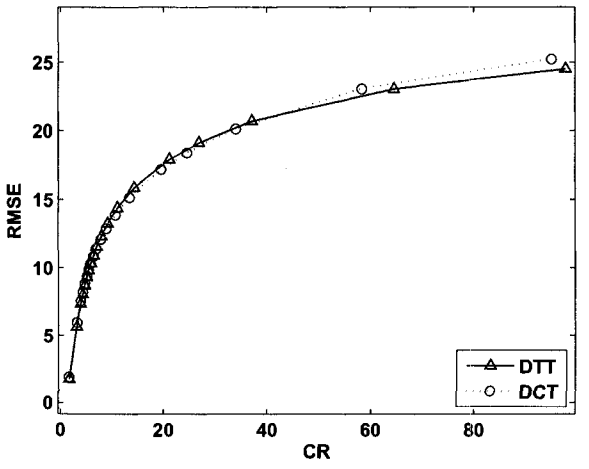


(b)

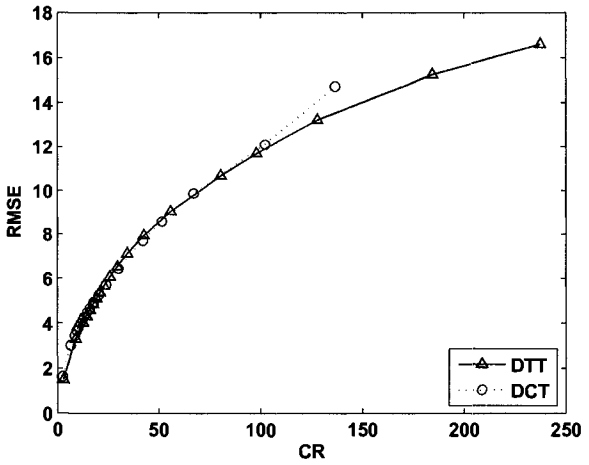


(c)

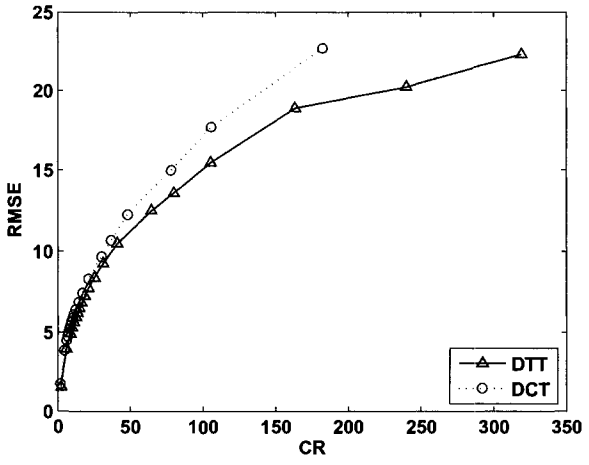
Figure 4.13: PSNR Performances of 8x8 DTT-Compressed Images (a) *Mandrill* (b) *Lena* and (c) *Einstein*



(a)

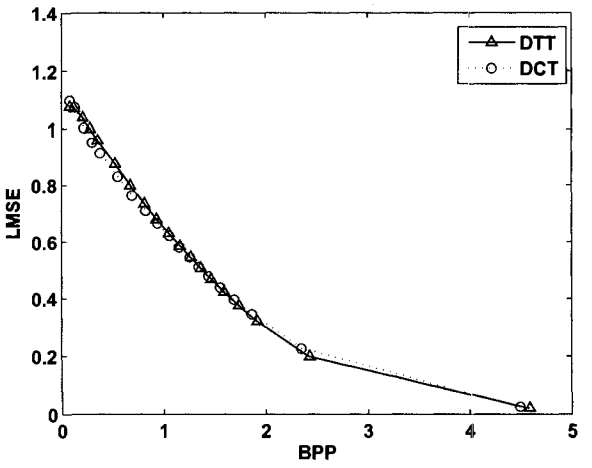


(b)

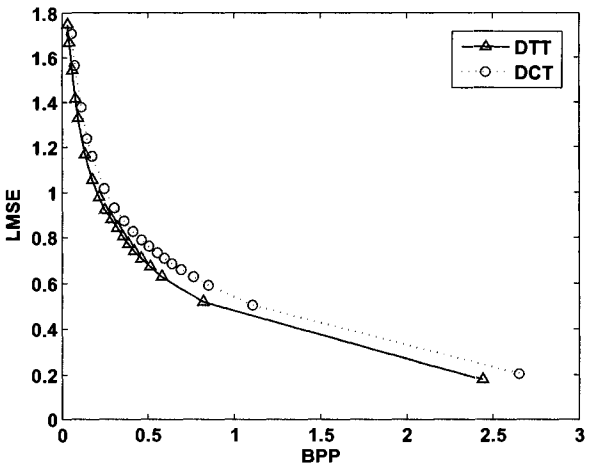


(c)

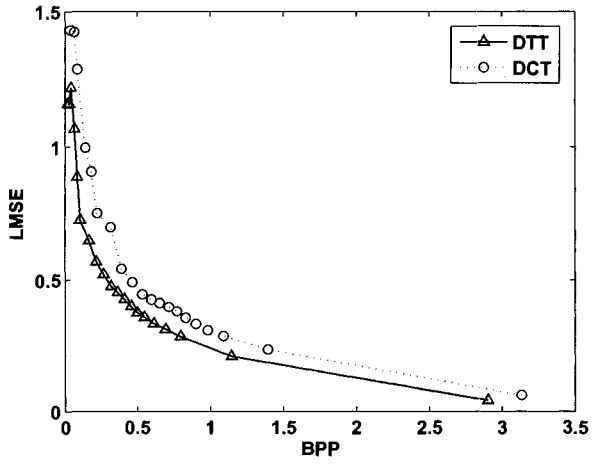
Figure 4.14: RMSE Performances of 8x8 DTT-Compressed Images (a) *Mandrill* (b) *Lena* and (c) *Einstein*



(a)

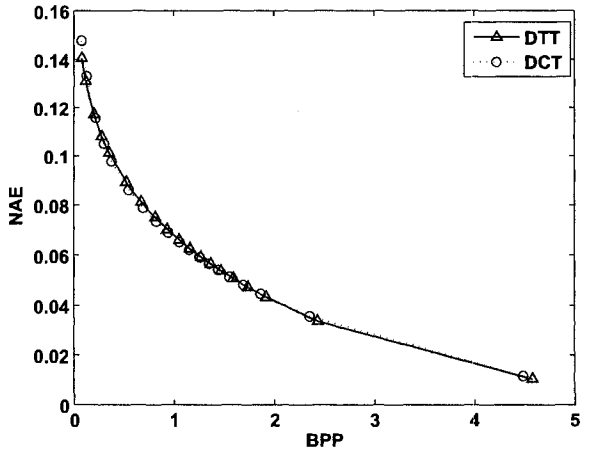


(b)

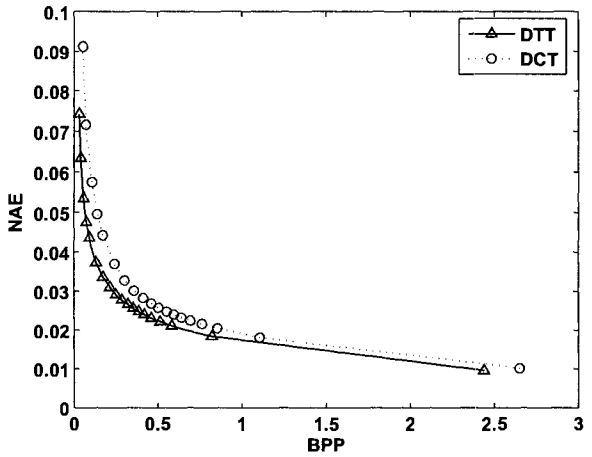


(c)

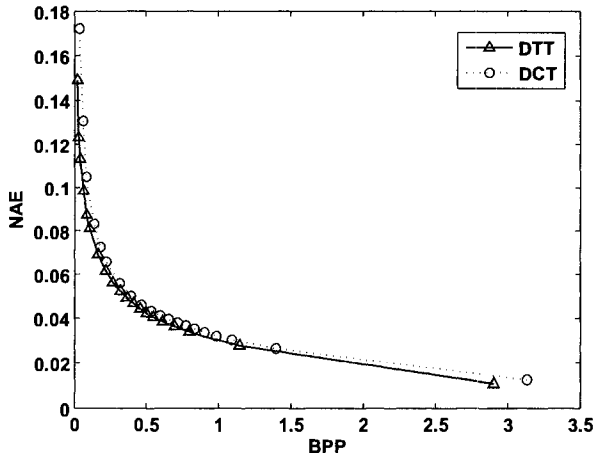
Figure 4.15: LMSE Performances of 8x8 DTT-Compressed Images (a) *Mandrill* (b) *Lena* and (c) *Einstein*



(a)

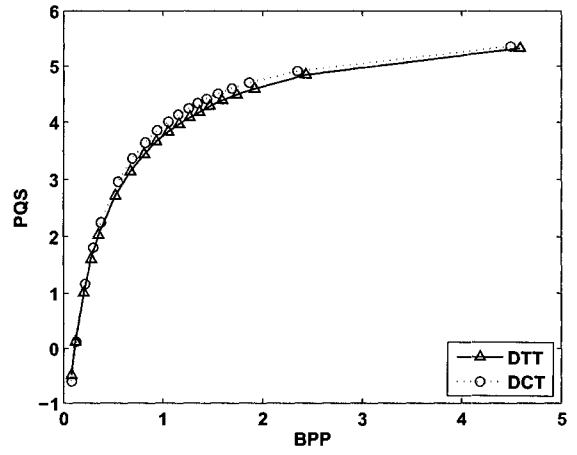


(b)

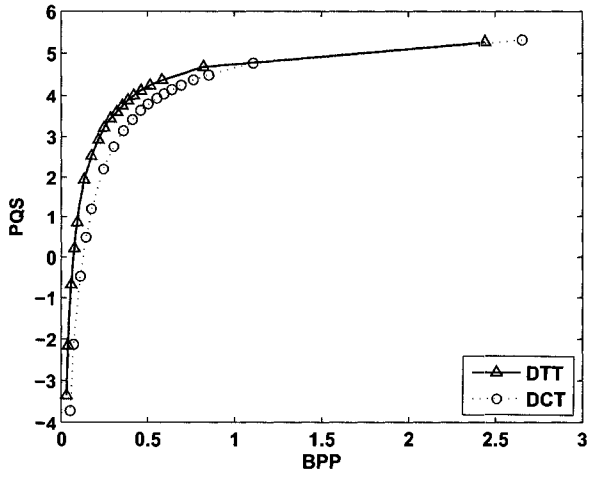


(c)

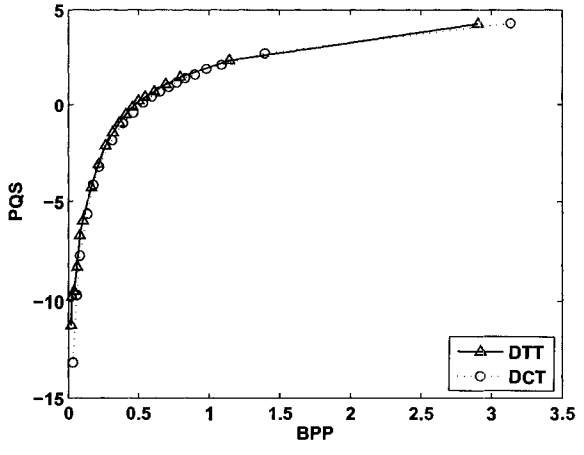
Figure 4.16: NAE Performances of 8x8 DTT-Compressed Images (a) *Mandrill* (b) *Lena* and (c) *Einstein*



(a)



(b)

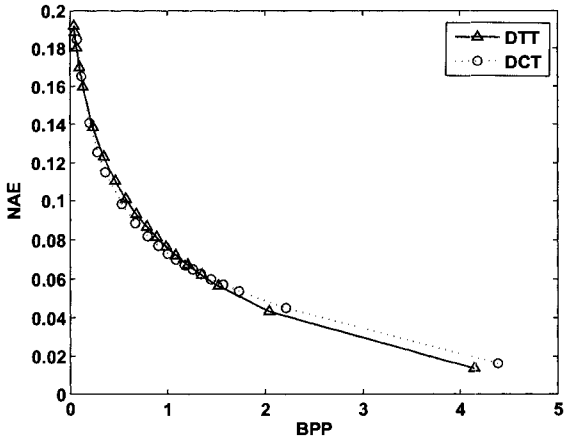


(c)

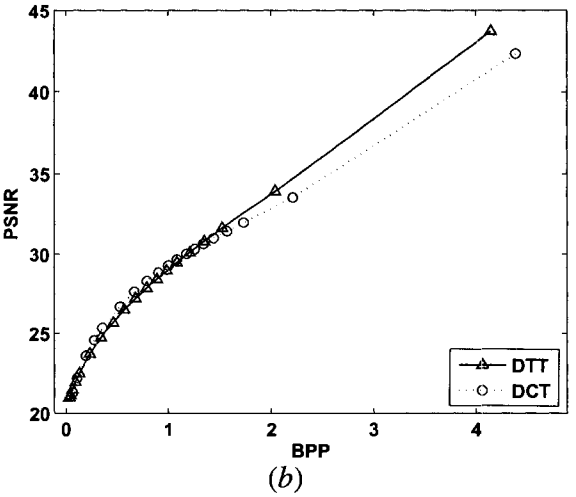
Figure 4.17: PQS Performances of 8x8 DTT-Compressed Images (a) *Mandrill* (b) *Lena* and (c) *Einstein*

Table 4.11: Short Table of Results for 8x8 DTT

		bpp	MSE	PSNR	NK	MD	LMSE
Einstein (bpp=6.8)	*DTT*	2.9	2.34	44.42	1.00	7	0.04
SF:20.83; SAM:344.8	*DCT*	3.1	3.06	43.27	1.00	8	0.05
Lena (bpp=7.4)	*DTT*	2.4	2.25	44.59	1.00	8	0.15
SF:14.04; SAM:909.09	*DCT*	2.6	2.72	43.78	0.99	9	0.20
Airplane (bpp=6.4)	*DTT*	1.6	1.52	46.30	1.00	6	0.05
SF:15.10; SAM:5000	*DCT*	1.9	1.64	45.96	0.99	8	0.06



(*a*)



(*b*)

Figure 4.18: Results of 8x8 DTT-Compressed Images of *Grass* for (a) NAE and (b) PSNR

4.4.3 Image Compression using 4x4 Discrete Tchebichef Transform

The advantages of using ITT over ICT in the transform coding stage of H.264/ AVC has been discussed in Sub-section 3.3.3. In this section, we compare the experimental results of 4x4 DTT, 4x4 DCT and 4x4 ICT. In this experiment, a gray image is divided into sub-blocks of size 4×4 as shown in Fig. 4.5. The test images used are those shown in Fig. 4.12. The image compression scheme shown in Fig. 4.3 is employed to obtain the transform coded image. Metrics for the picture quality evaluation of the compressed images are tabulated. The performance evaluation is carried out by carefully studying the correlation of the metrics to the HVS characteristics.

In the sub-sections that follow, we first give a preliminary analysis based on the conventional objective measures, PSNR and RMSE. Later, we delve into the performance analysis of DTT image-wise based on the structural properties of images explained in Sub-section 4.4.1. We give a detailed account of experimental results for various measures of quality for DTT, ICT and DCT. We refer to Figs. 4.19 - 4.25 for the performances of the quality measures, and we refer to Tables 4.12 - 4.19 for quality measures tabulated for various compression levels measured at the respective bpp. The last column of every table gives the extent of correlation of each of the quality measure and PQS, i.e.,

$$r_{PQS} = corr(qualitymeasure, PQS)$$

where $qualitymeasure, PQS$ correspond to x and x' in (4.3). The results of image quality analysis for 8x8 DTT compression are shown in Figs. 4.19 (a) - (f), Figs. 4.23 (a) - (f), Figs. 4.20 (a) - (f), Figs. 4.21 (a) - (f), Figs. 4.22 (a) - (f) and Figs. 4.24 (a) - (f) for the images *Mandrill*, *Cameraman*, *Peppers*, *Leopard*, *Lena* and *Airplane* respectively. In these figures, the rate distortion plots of PSNR, LMSE, RMSE, PQS, NAE and MD are presented. In Figs. 4.25 (a) and (b), PSNR results are shown for the images *Einstein* and *Texture Mosaic* respectively.In Figs. 4.25 (c) and (d), RMSE results are shown for the images *Einstein* and

Texture Mosaic respectively.

Preliminary Analysis

Since a large number of tables for various quality measures have been given, for ease of view, we present a short table with only a few PSNR and RMSE results in Table 4.20. Figs. 4.26 and 4.27 give the original and reconstructed images of *Peppers* and *Mandrill* respectively for these transforms. From these figures, it can be seen that DTT encodes a given image at a lower bit rate with a higher PSNR and a lower RMSE as compared to DCT and ICT. Overall, a comparative gain in PSNR of 0-4 dB has been observed. It is evident from Table 4.20, Figs. 4.27 (a) - (d) and Figs. 4.26 (a) - (d) that DTT outperforms DCT and ICT even at very low bit-rates. This is a significant factor that supports our proposal of using DTT for video compression, since encoding at low bit-rates is a prominent feature of the H.264 standard. Comparative gain in PSNR for DTT of 0-2.2 dB is noticed. From the RMSE results, it is seen that the *reconstruction accuracy* of DTT is seen to be better than that of DCT and ICT.

Picture Quality Evaluation of 4x4 DTT Compressed Images

Now we undertake a detailed image quality analysis for all the test images. Due to the large number of plots generated from the variety of quality measures used, we selectively present the plots of quality measures that exhibit good correlation with the subjective quality measure, PQS (some have been plotted in log scale for better perception). The values of PQS fall out of range at the lowest bpp for all images since the image quality is extremely low at these values of bpp. PQS falls out of range for more than half of the values for the images *Einstein* and *Texture Mosaic* as seen in the last rows of Tables 4.16 and 4.18. Hence, we will carry out only an objective image quality study for these two images. For almost all quality measures except PSNR and RMSE, the performance of ICT is significantly out of

range. Hence, we compare DTT with ICT only in valid cases.

For accurate analysis and consistency of the results [37], as explained in the 8x8 case, we

(i) consider those quality measures which have the best correlation with the subjective measures PQS and

(ii) obviate a quality measure for which the sign of the correlation coefficient changes (for the same quality measure across different techniques).

From the last column, r_{PQS} of Tables 4.12 - 4.19, it is seen that AD and SC have the poorest correlation for all the test images. Also, NK correlates poorly for all images except *Lena* and *Cameraman*. Moreover, the correlation coefficient pertaining to SC, AD and NK change signs for the images *Mandrill*, *Lena*, *Peppers* and *Airplane*. Hence, it is not suitable to use AD, SC and NK in our image quality analysis.

Mandrill:

The image *Mandrill* has a low SAM and a high SFM (in (4.2)), i.e., it is a fairly difficult image to code due to the large amount of details embedded in the image. From Figs. 4.19 (a) - (f) and Table 4.12, it can be seen that the DTT compressed images of *Mandrill* perform very well across various quality measures. It is seen that RMSE, LMSE, NAE, MD and AD for DTT are lower than those for DCT. There is a gain of 1dB at low bit-rates upto 2.5 dB at higher bit-rates for the PSNR of DTT as compared to that of DCT and ICT. LMSE, MD and RMSE of DTT are lesser than those of DCT. NAE is almost the same for DTT and DCT. PQS of DTT is higher at low bit-rates and is almost equal to that of DCT at higher bit-rates.

Peppers:

The image *Peppers* has a low SAM and a medium SFM (in (4.2)), i.e., it is a mixed image with details as well as a large number of correlated regions. The following observations are noted from Figs. 4.20 (a) - (f) and Table 4.17. PQS falls out of range at very low bpp.

At high bpp, the performance of DTT and DCT is almost same. LMSE, MD and NAE for DTT is lower than that of DCT, but is higher than that of ICT. RMSE for DTT is almost the same as that for DCT and ICT at low bpp, and lower than DCT and ICT at higher bpp. There is a gain of 1dB at low bit-rates upto 1.7 dB at higher bit-rates for the PSNR of DTT as compared to that of DCT and ICT.

Leopard:

The image *Leopard* is similar to the image *Mandrill*, in the sense that it has a low SAM and a high SFM (in (4.2)) leading to high frequency contents and low predictability. Analysis of results in Figs. 4.21 (a) - (f) and Table 4.13 lead to the following observations. The DTT compressed images of *Leopard* perform very well across various quality measures. PQS falls out of range at very low bpp. At high bpp, the performance of DTT and DCT is almost same. NAE for DTT is almost the same as that for DCT. LMSE and MD for DTT is lower than that of DCT, but is higher than that of ICT. RMSE for DTT is lower than that for DCT and ICT . There is a gain of 1dB at low bit-rates upto 1.7 dB at higher bit-rates for the PSNR of DTT as compared to that of DCT and ICT.

Lena:

The image *Lena* has a low SFM and a low SAM (in (4.2)). Hence, it has a medium level of coding difficulty. From Figs. 4.22 (a), (b) and Table 4.14, the following observations are made. PQS falls out of range at very low bpp. At high bpp, the performance of DTT and DCT is almost same. NAE, MD and LMSE for DTT is almost the same as that for DCT. RMSE for DTT is lower than that for DCT and ICT . There is a gain of upto 0.7 dB for the PSNR of DTT as compared to that of DCT and ICT.

Cameraman:

The image *Cameraman* has a low SAM and a high SFM (in (4.2)). From Figs. 4.23 (a) - (f) and Table 4.15, it can be seen that the DTT-compressed images of *Cameraman* perform very well across various quality measures. PQS falls out of range at very low bpp. At high

bpp, the performance of DTT and DCT is almost same. NAE for DTT is almost the same as that for DCT. RMSE (at low bit rate), LMSE and MD for DTT is lower than that for DCT. There is a gain of 0.5dB at low bit-rates upto 2dB at higher bit-rates for the PSNR of DTT as compared to that of DCT and ICT.

Airplane:

The image *Airplane* has a high SAM and a low SFM (in (4.2)), i.e., it has a high predictability and contains less amount of details. The following observations are noted from Figs. 4.24 (a) - (f) and Table 4.19. PQS falls out of range at very low bpp. At high bpp, the performance of DTT and DCT is almost same. MD and LMSE for DTT is almost the same as that for DCT. RMSE and NAE for DTT are lower than those for DCT and ICT . There is a gain of upto 1.3 dB for the PSNR of DTT as compared to that of DCT and ICT.

Texture Mosaic:

The image *Texture Mosaic* has a high SAM and a low SFM (in (4.2)), similar to the image *Airplane*. From Table 4.18, Figs. 4.25 (a) and (b), it can be seen that the PSNR for DTT is about 1.2 dB higher and RMSE of DTT is lower than that of DCT and ICT.

Einstein:

The image *Einstein* has a medium SAM and SFM (in (4.2)), which shows that it lies in the mid-range of coding difficulty. From Table 4.16, Figs. 4.25 (a) and (b), it can be seen that the PSNR for DTT is about 0.5-0.8 dB higher than that of DCT and ICT, and it can also be seen that the RMSE of DTT is lower than that of DCT and ICT, especially at higher compression ratios.

Summary

From the above we see that the performance gain of DTT over DCT is highest for the images *Mandrill*, *Leopard* and *Cameraman*. Considering the structural properties of images described by SFM and SAM, *Mandrill*, *Leopard* and *Cameraman* have high values of SFM

and low values of SAM (in (4.2)), i.e., these images have a high amount of details and a low predictability. Hence, it can be concluded that DTT performs better on images with a high SFM and a low SAM.

The trends observed in the results imply that RMSE increases with an increase in the number of higher frequency components in the image, which corresponds to increasing SFM. Reconstruction accuracy depends on SFM, while visual quality depends on SAM. These observations are consistent with those in the literature in the context of image compression.

From the observations made thus far, the following points can be noted

1. PQS of DTT and DCT are almost same across all images.

2. DTT has a significant performance improvement over DTT in error-based and gain-based measures. For HVS-based measures, the performance of DTT is almost the same as that of DCT.

3. DTT performs very well across all quality measures for images with high SFM lying in the higher coding difficulty region in (4.2).

4. RMSE of DTT is lower at higher compression ratios across all the images, and the PSNR performance of DTT is very good across all images. Performance gain of upto 2dB has been observed for DTT over DCT and ICT.

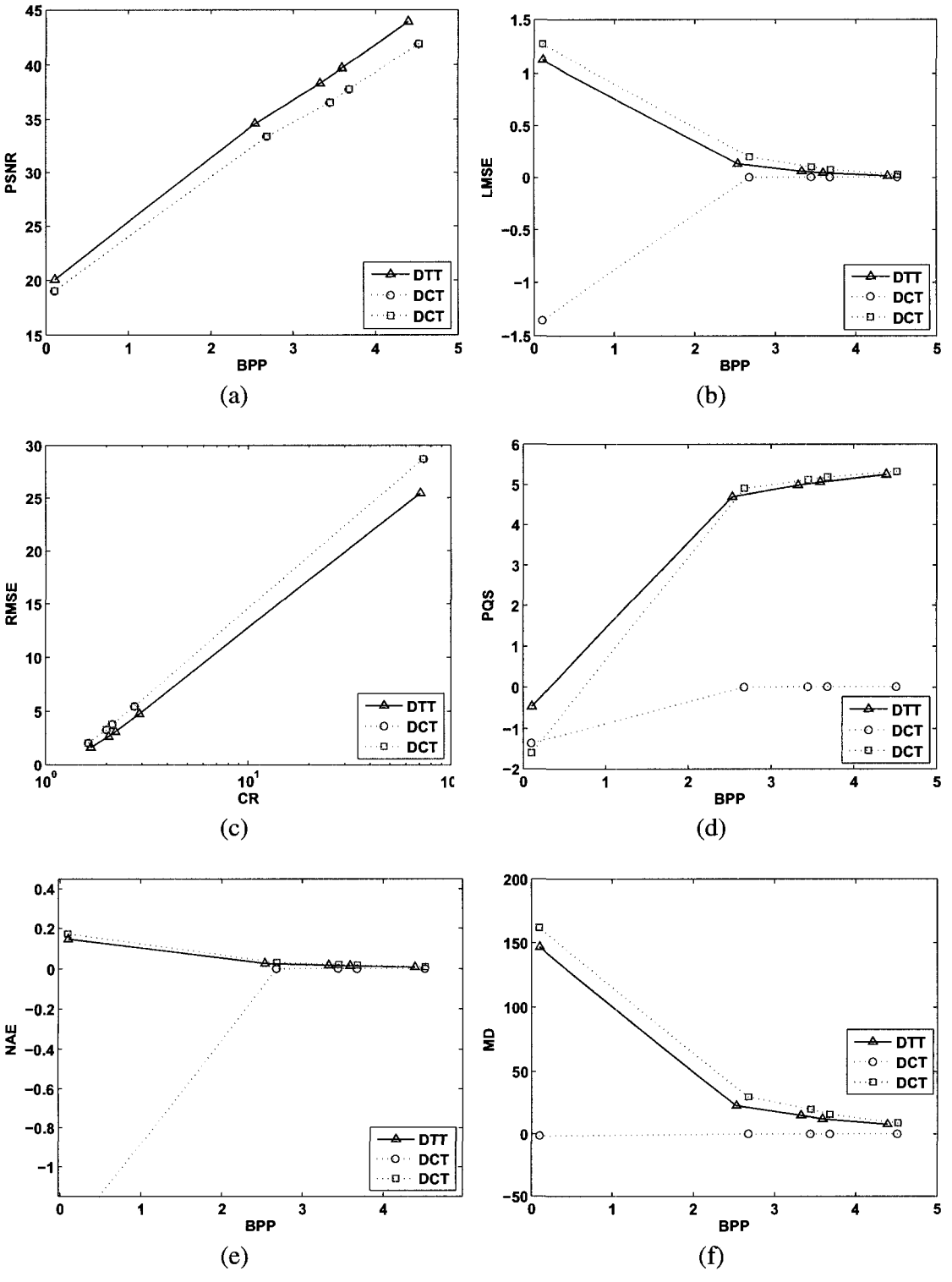


Figure 4.19: Results of 4x4 DTT-Compressed Images of *Mandrill* for Quality Measures (a) PSNR, (b) LMSE, (c) RMSE, (d) PQS, (e) NAE, and (f) MD

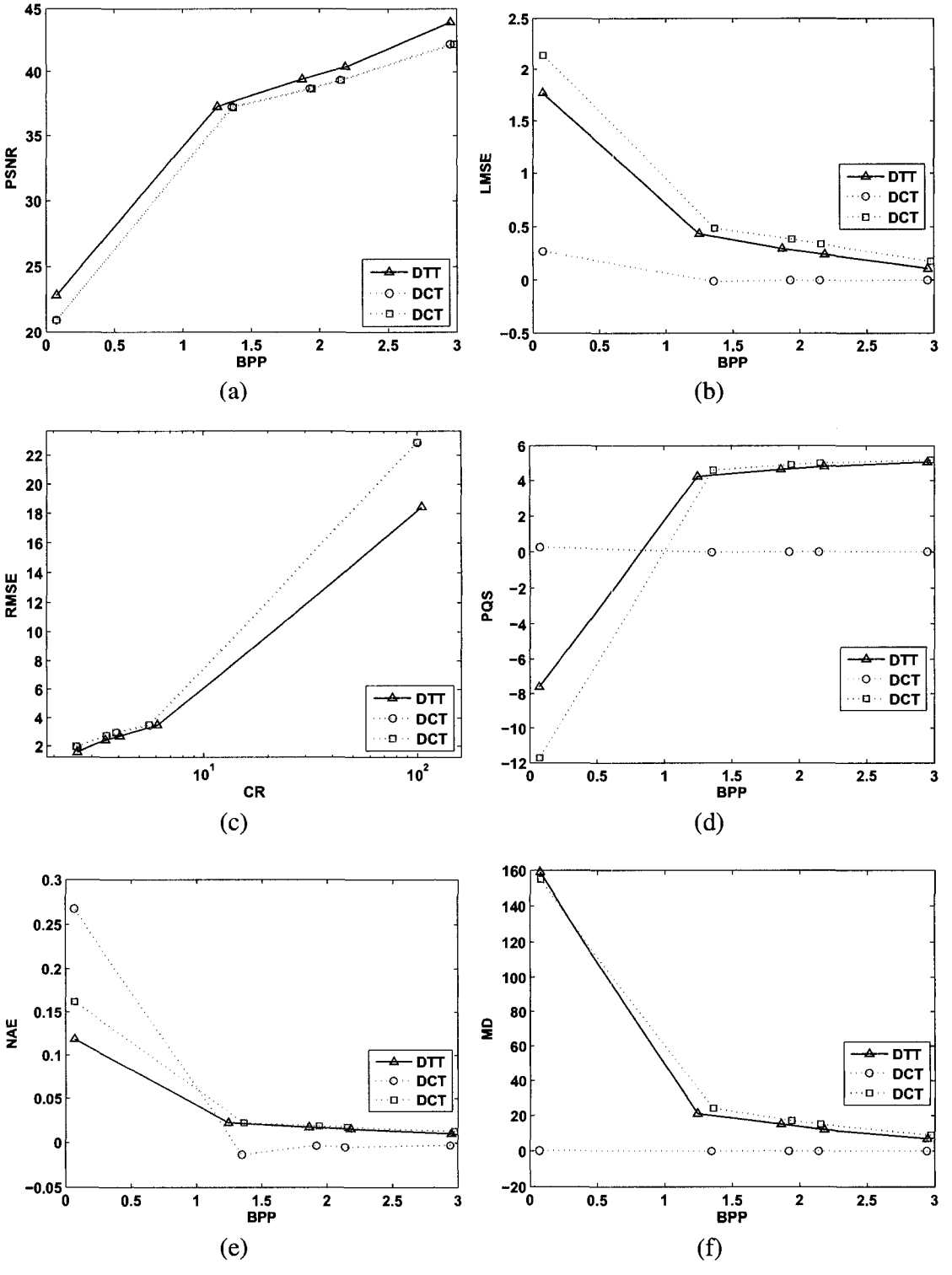


Figure 4.20: Results of 4x4 DTT-Compressed Images of *Peppers* for Quality Measures (a) PSNR, (b) LMSE, (c) RMSE, (d) PQS, (e) NAE and (f) MD

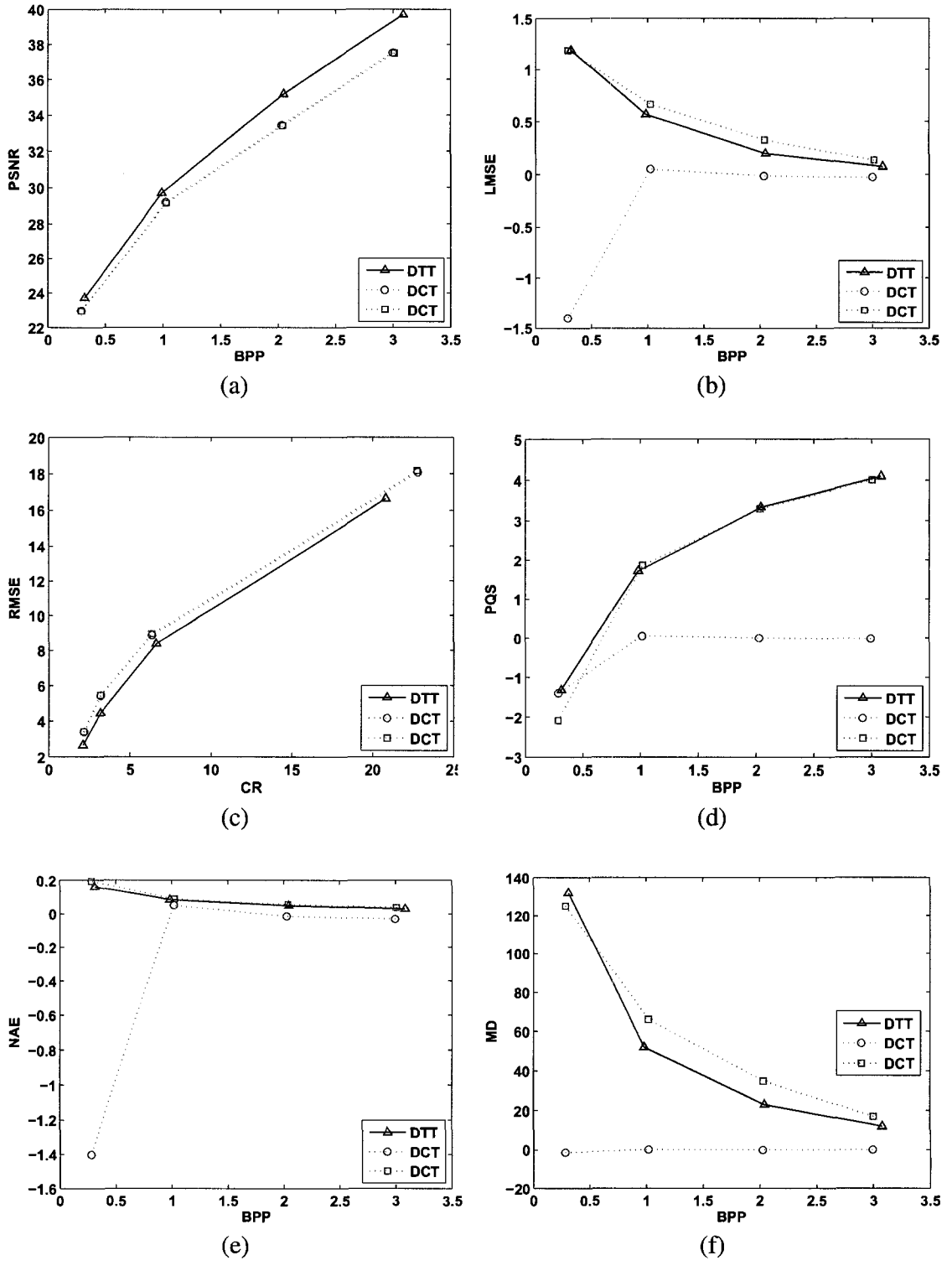


Figure 4.21: Results of 4x4 DTT-Compressed Images of *Leopard* for Quality Measures (a) PSNR, (b) LMSE, (c) RMSE, (d) PQS, (e) NAE and (f) MD

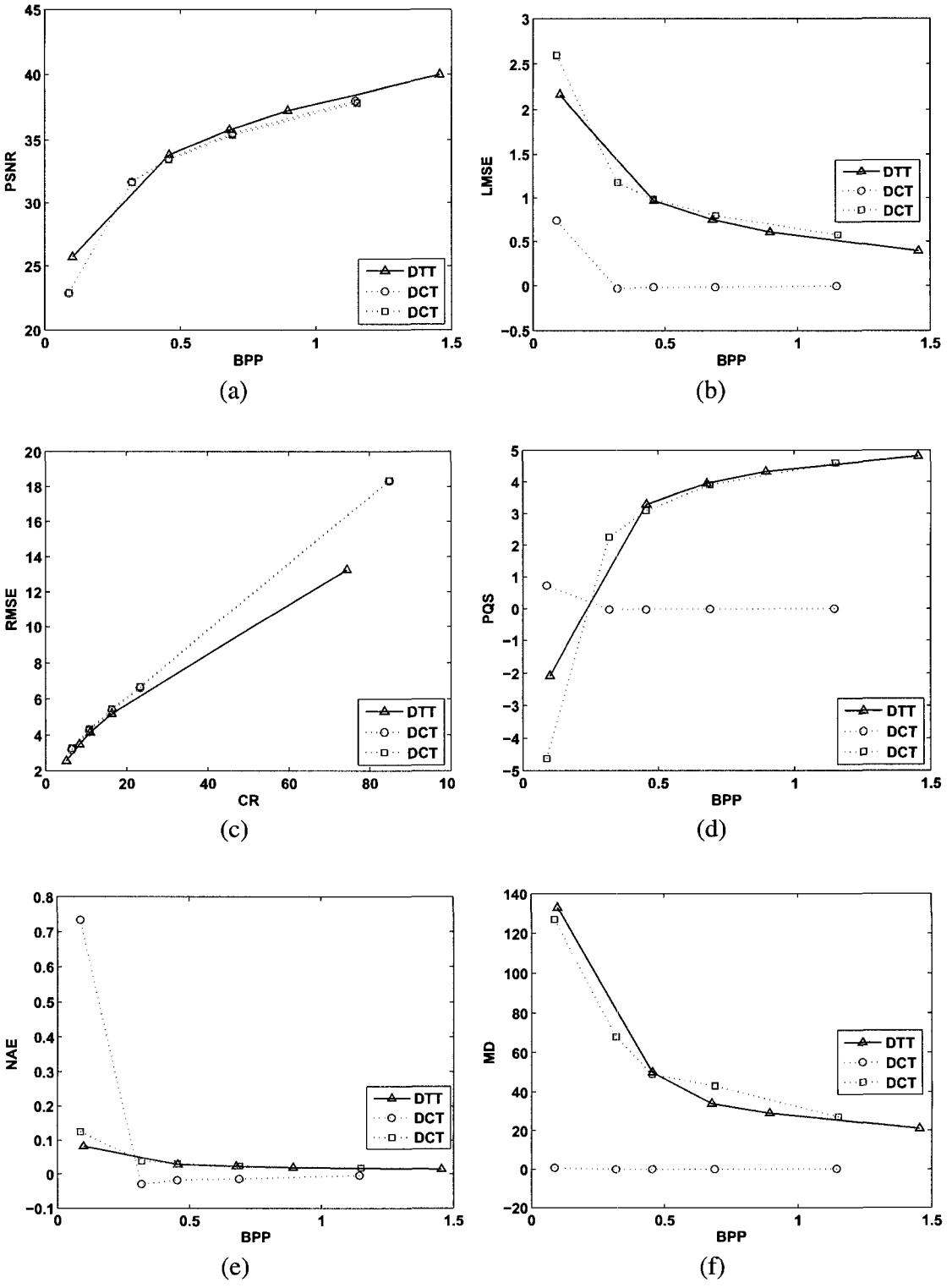


Figure 4.22: Results of 4x4 DTT-Compressed Images of *Lena* for Quality Measures (a) PSNR, (b) LMSE, (c) RMSE, (d) PQS, (e) NAE and (f) MD

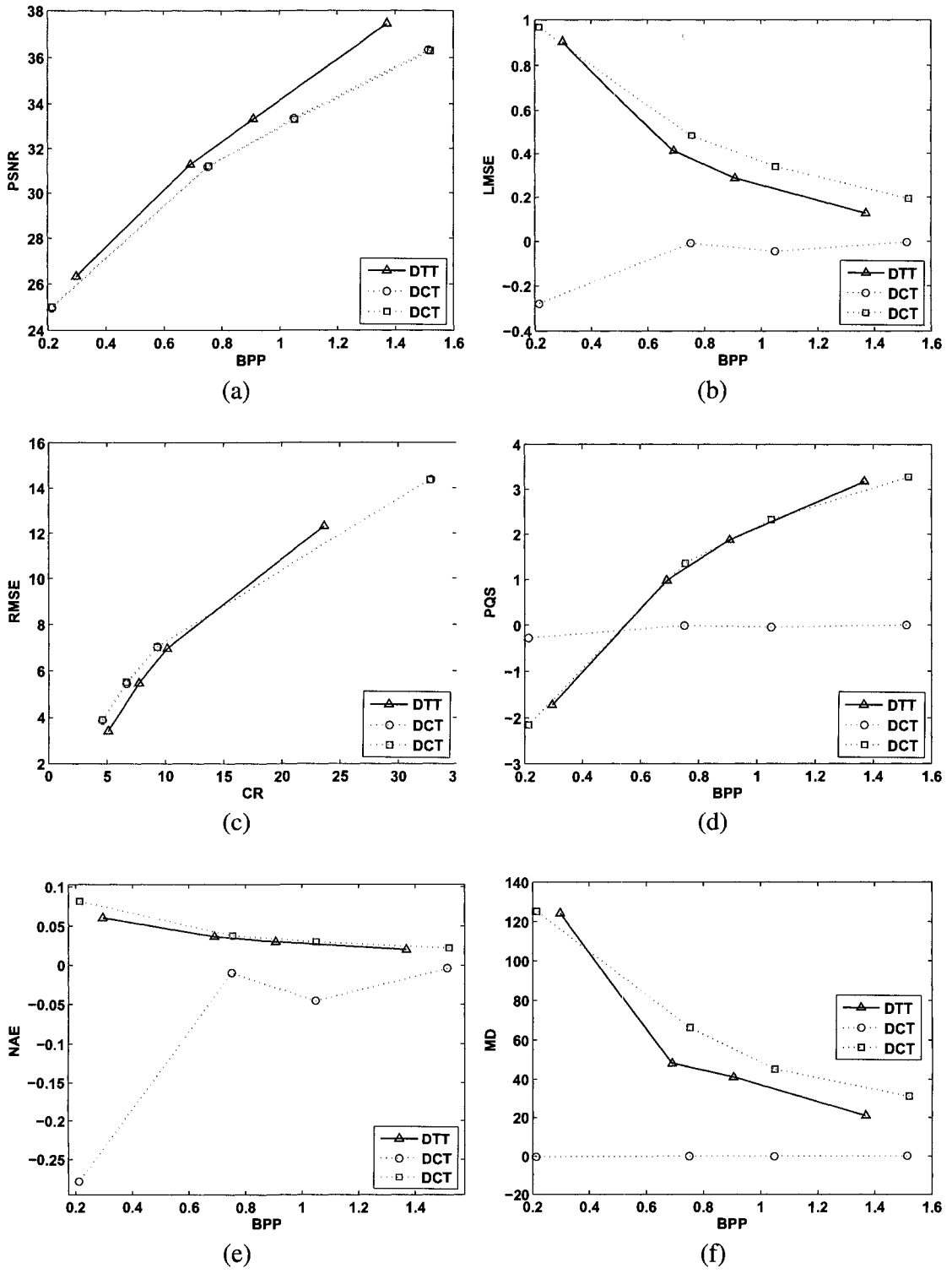


Figure 4.23: Results of 4x4 DTT-Compressed Images of *Cameraman* for Quality Measures (a) PSNR, (b) LMSE, (c) RMSE, (d) PQS, (e) NAE and (f) MD

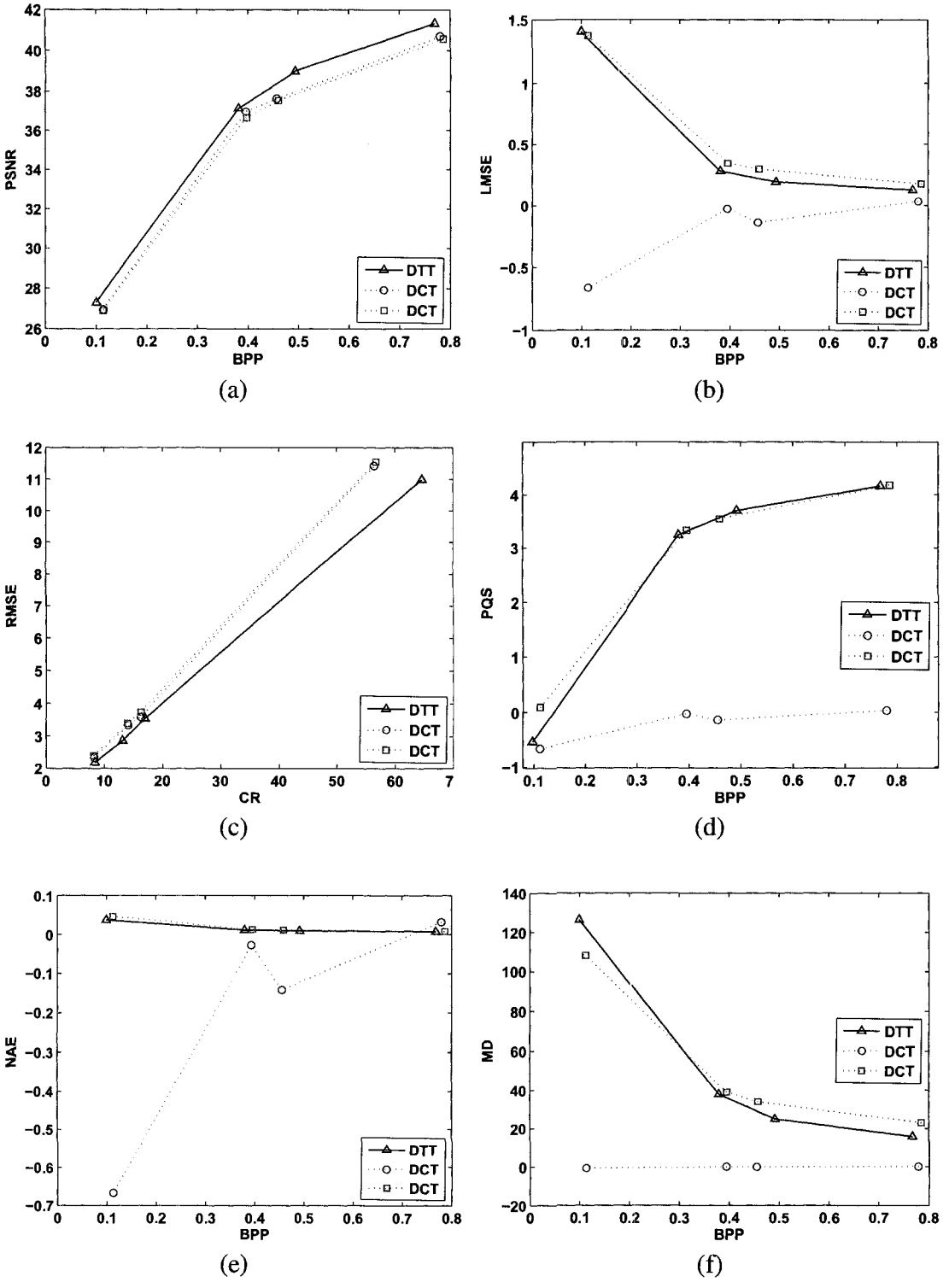


Figure 4.24: Results of 4x4 DTT-Compressed Images of *Airplane* for Quality Measures (a) PSNR, (b) LMSE, (c) RMSE, (d) PQS, (e) NAE and (f) MD

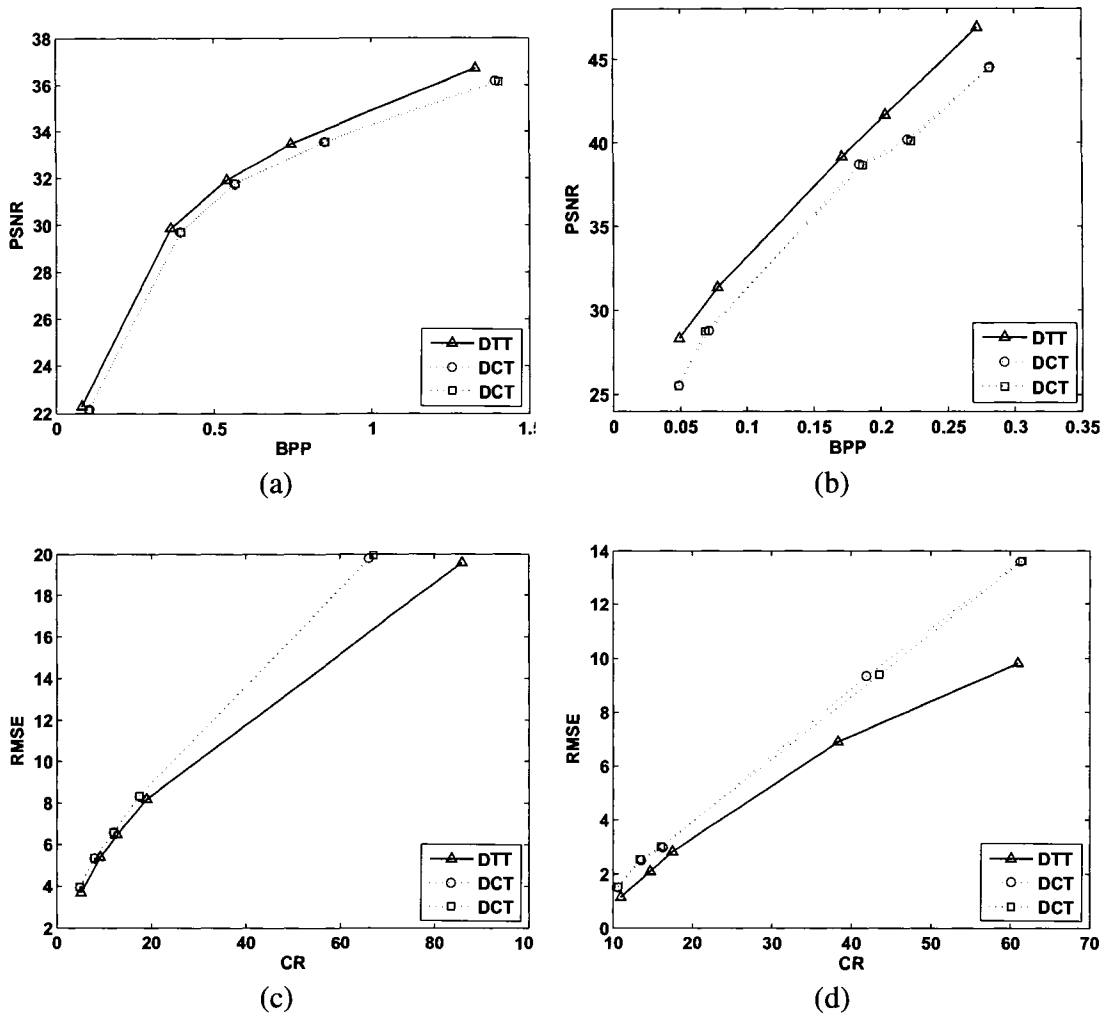


Figure 4.25: Results of 4x4 DTT-Compressed Images for (a) PSNR of *Einstein*, (b) PSNR of *Texture Mosaic*, (c) RMSE of *Einstein* and (d) RMSE of *Texture Mosaic*

Table 4.12: Quality Measures Results for 4x4 DTT-compressed Images at Different Bit-rates for *Mandrill (SF:36.5146; SAM:100)*

	bpp	4.39	3.59	3.33	2.53	0.10	r_{PQS}
	MSE	2.62	6.93	9.54	22.58	647.87	-0.9984
	RMSE	1.68	2.05	2.21	2.91	70.66	-0.9978
	PSNR	43.94	39.73	38.33	34.59	20.02	0.9575
D	**AD**	-0.00	-0.00	-0.00	0.00	-0.03	0.9760
T	**SC**	1.00	1.00	1.00	1.00	1.02	-0.9950
T	**NK**	1.00	1.00	1.00	1.00	0.97	0.9973
	MD	8.00	12.00	15.00	23.00	147.00	-0.9999
	LMSE	0.02	0.05	0.06	0.14	1.12	-0.9999
	NAE	0.01	0.02	0.02	0.03	0.15	-0.9993
	PQS	5.24	5.06	4.99	4.69	-0.46	1.0000
	bpp	4.51	3.68	3.44	2.68	0.10	r_{PQS}
	MSE	4.18	10.81	14.37	29.70	824.15	-0.9997
	RMSE	1.63	2.00	2.14	2.75	73.26	-0.9993
	PSNR	41.92	37.79	36.56	33.40	18.97	0.9533
D	**AD**	0.00	0.00	0.00	0.00	-1.36	0.9988
C	**SC**	1.00	1.00	1.00	1.00	0.98	0.9993
T	**NK**	1.00	1.00	1.00	1.00	0.99	0.9999
	MD	10.00	16.00	19.00	30.00	162.00	-0.9980
	LMSE	0.03	0.08	0.10	0.20	1.26	-0.9975
	NAE	0.01	0.02	0.02	0.03	0.17	-0.9984
	PQS	5.31	5.18	5.12	4.91	-1.60	1.0000
	bpp	4.52	3.68	3.45	2.68	0.10	r_{PQS}
	MSE	4.18	10.77	14.35	29.61	825.08	-0.9997
	RMSE	1.63	2.00	2.14	2.75	73.26	-0.9993
	PSNR	41.92	37.81	36.56	33.42	18.97	0.9535
I	**AD**	-0.00	-0.01	0.00	-0.01	-1.35	0.9989
C	**SC**	1.00	1.00	1.00	1.00	0.98	0.9993
T	**NK**	1.00	1.00	1.00	1.00	0.99	0.9999
	MD	9.00	16.00	20.00	30.00	162.00	-0.9976
	LMSE	0.03	0.08	0.10	0.20	1.27	-0.9976
	NAE	0.01	0.02	0.02	0.03	0.17	-0.9984
	PQS	5.31	5.18	5.12	4.90	-1.60	1.0000

Table 4.13: Quality Measures Results for 4x4 DTT-compressed Images at Different Bit-rates for *Leopard* (*SF:33.82; SAM:57.91*)

	bpp	3.08	2.04	0.98	0.31	r_{PQS}
	MSE	6.97	19.73	69.98	277.06	-0.9781
	RMSE	2.12	3.19	6.64	20.81	-0.9798
	PSNR	39.70	35.18	29.68	23.71	0.9756
D	**AD**	0.03	0.05	0.11	-0.02	0.4021
T	**SC**	1.00	1.00	1.00	1.01	-0.9562
T	**NK**	1.00	1.00	1.00	0.98	0.9740
	MD	12.00	23.00	52.00	132.00	-0.9940
	LMSE	0.07	0.20	0.57	1.19	-0.9995
	NAE	0.03	0.05	0.08	0.16	-0.99970
	PQS	4.09	3.33	1.72	-1.32	1.0000
	bpp	3.00	2.03	1.02	0.29	r_{PQS}
	MSE	11.46	29.45	78.74	329.37	-0.9910
	RMSE	2.18	3.22	6.39	22.78	-0.9901
	PSNR	37.54	33.44	29.17	22.95	0.9653
D	**AD**	-0.03	-0.01	0.05	-1.40	0.9300
C	**SC**	1.00	1.00	1.00	0.99	0.8329
T	**NK**	1.00	1.00	0.99	0.99	0.9852
	MD	17.00	38.00	72.00	127.00	-0.9863
	LMSE	0.13	0.32	0.65	1.18	-0.9871
	NAE	0.04	0.06	0.09	0.19	-0.9995
	PQS	3.98	3.28	1.88	-2.12	1.0000
	bpp	3.01	2.04	1.02	0.29	r_{PQS}
	MSE	11.57	29.70	79.94	331.83	-0.9904
	RMSE	2.18	3.22	6.39	22.78	-0.9892
	PSNR	37.50	33.40	29.10	22.92	0.9664
I	**AD**	0.02	0.02	0.10	-1.37	0.9268
C	**SC**	1.00	1.00	1.00	0.99	0.8173
T	**NK**	1.00	1.00	0.99	0.99	0.9846
	MD	17.00	35.00	66.00	125.00	-0.9941
	LMSE	0.13	0.33	0.66	1.19	-0.9872
	NAE	0.04	0.06	0.09	0.19	-0.9994
	PQS	4.00	3.29	1.86	-2.09	1.0000

Table 4.14: Quality Measures Results for 4x4 DTT-compressed Images at Different Bit-rates for *Lena (SF:14.0436; SAM:909.09)*

	bpp	1.46	0.89	0.68	0.46	0.10	r_{PQS}
	MSE	6.48	12.12	17.16	26.93	175.73	-0.9956
	RMSE	5.12	8.32	10.98	16.32	74.35	-0.9984
	PSNR	40.01	37.29	35.78	33.83	25.68	0.9724
D	**AD**	-0.00	-0.00	0.01	-0.00	-0.02	0.8795
T	**SC**	1.00	1.00	1.00	1.00	0.99	0.9731
T	**NK**	1.00	1.00	1.00	1.00	1.00	0.9859
	MD	21.00	29.00	34.00	50.00	133.00	-0.9990
	LMSE	0.39	0.60	0.75	0.97	2.17	-0.9942
	NAE	0.02	0.02	0.02	0.03	0.08	-0.9999
	PQS	4.81	4.33	3.95	3.29	-2.09	1.0000
	bpp	1.15	0.69	0.46	0.32	0.09	r_{PQS}
	MSE	10.21	18.22	29.01	44.08	336.04	-0.9890
	RMSE	6.50	10.80	16.35	23.31	84.76	-0.9991
	PSNR	38.04	35.52	33.51	31.69	22.87	0.9828
D	**AD**	-0.00	-0.01	-0.02	-0.03	0.73	-0.9651
C	**SC**	1.00	1.00	1.00	1.00	0.99	0.9800
T	**NK**	1.00	1.00	1.00	1.00	1.00	0.9940
	MD	26.00	39.00	54.00	66.00	127.00	-0.9874
	LMSE	0.56	0.78	0.97	1.16	2.60	-0.9988
	NAE	0.02	0.02	0.03	0.04	0.13	-0.9980
	PQS	4.60	3.90	3.10	2.24	-4.64	1.0000
	bpp	1.15	0.69	0.45	0.32	0.09	r_{PQS}
	MSE	10.58	18.80	29.68	44.66	336.84	-0.9890
	RMSE	6.50	10.80	16.35	23.31	84.76	-0.9990
	PSNR	37.89	35.39	33.41	31.63	22.86	0.9834
I	**AD**	-0.00	-0.01	-0.01	-0.02	0.73	-0.9678
C	**SC**	1.00	1.00	1.00	1.00	0.99	0.9779
T	**NK**	1.00	1.00	1.00	1.00	1.00	0.9920
	MD	27.00	43.00	49.00	68.00	127.00	-0.9874
	LMSE	0.57	0.79	0.99	1.18	2.60	-0.9988
	NAE	0.02	0.03	0.03	0.04	0.13	-0.9979
	PQS	4.60	3.90	3.09	2.24	-4.64	1.0000

Table 4.15: Quality Measures Results for 4x4 DTT-compressed Images at Different Bit-rates for *Cameraman* (*SF:28.86; SAM:184.22*)

	bpp	1.37	0.91	0.69	0.30	r_{PQS}
	MSE	11.73	30.04	48.43	151.38	-0.9782
	RMSE	5.12	7.73	10.15	23.64	-0.9803
	PSNR	37.44	33.35	31.28	26.33	0.9897
D	**AD**	0.00	-0.04	-0.04	-0.17	0.9760
T	**SC**	1.00	1.00	1.00	1.00	-0.7196
T	**NK**	1.00	1.00	1.00	1.00	0.9364
	MD	21.00	41.00	48.00	124.00	-0.9801
	LMSE	0.13	0.29	0.41	0.90	-0.9958
	NAE	0.02	0.03	0.04	0.06	-0.9990
	PQS	3.16	1.87	0.98	-1.72	1.0000
	bpp	1.51	1.05	0.75	0.21	r_{PQS}
	MSE	15.14	29.98	49.47	206.97	-0.9850
	RMSE	4.63	6.68	9.32	32.81	-0.9829
	PSNR	36.33	33.36	31.19	24.97	0.9920
D	**AD**	-0.00	-0.05	-0.01	-0.28	0.9415
C	**SC**	1.00	1.00	1.00	1.00	-0.8432
T	**NK**	1.00	1.00	1.00	0.99	0.9747
	MD	31.00	46.00	74.00	138.00	-0.9967
	LMSE	0.19	0.33	0.49	0.97	-0.9996
	NAE	0.02	0.03	0.04	0.08	-0.9964
	PQS	3.26	2.31	1.35	-2.22	1.0000
	bpp	1.52	1.05	0.75	0.21	r_{PQS}
	MSE	15.30	30.49	49.19	206.13	-0.9840
	RMSE	4.63	6.68	9.32	32.81	-0.9821
	PSNR	36.28	33.29	31.21	24.99	0.9925
I	**AD**	-0.02	-0.07	-0.02	-0.29	0.9333
C	**SC**	1.00	1.00	1.00	1.00	-0.8665
T	**NK**	1.00	1.00	1.00	0.99	0.9737
	MD	31.00	45.00	66.00	125.00	-0.9992
	LMSE	0.19	0.34	0.48	0.97	-0.9997
	NAE	0.02	0.03	0.04	0.08	-0.9960
	PQS	3.26	2.32	1.35	-2.15	1.0000

Table 4.16: Quality Measures Results for 4x4 DTT-compressed Images at Different Bit-rates for *Einstein* (*SF:20.8394; SAM:344.8*)

	bpp	1.33	0.74	0.54	0.36	0.08	r_{PQS}
	MSE	13.86	29.27	41.76	67.11	383.47	-0.9813
	RMSE	5.18	9.26	12.79	19.03	85.88	-0.9860
	PSNR	36.71	33.47	31.92	29.86	22.29	0.9855
D	**AD**	0.00	0.01	0.04	-0.01	0.31	-0.9403
T	**SC**	1.00	1.00	1.00	1.00	1.02	-0.9585
T	**NK**	1.00	1.00	1.00	1.00	0.98	0.9733
	MD	22.00	35.00	51.00	74.00	168.00	-0.9986
	LMSE	0.19	0.33	0.42	0.58	1.35	-0.9999
	NAE	0.03	0.04	0.04	0.05	0.13	-0.9966
	PQS	2.51	0.85	-0.09	-1.80	-10.24	1.0000
	bpp	1.39	0.85	0.57	0.39	0.10	r_{PQS}
	MSE	15.73	28.74	43.47	69.50	392.35	-0.9874
	RMSE	4.94	8.09	12.18	17.57	66.22	-0.9953
	PSNR	36.16	33.55	31.75	29.71	22.19	0.9831
D	**AD**	-0.03	-0.01	0.04	-0.11	0.06	-0.4849
C	**SC**	1.00	1.00	1.00	1.00	0.98	0.9571
T	**NK**	1.00	1.00	1.00	1.00	1.00	0.8604
	MD	32.00	44.00	73.00	72.00	139.00	-0.9777
	LMSE	0.25	0.39	0.50	0.83	1.47	-0.9817
	NAE	0.03	0.04	0.04	0.06	0.16	-0.9970
	PQS	2.70	1.34	0.17	-1.31	-10.68	1.0000
	bpp	1.40	0.85	0.57	0.39	0.10	r_{PQS}
	MSE	15.89	28.79	43.29	69.61	397.89	-0.9867
	RMSE	4.94	8.09	12.18	17.57	66.22	-0.9951
	PSNR	36.12	33.54	31.77	29.70	22.13	0.9843
I	**AD**	0.00	-0.00	0.06	-0.07	0.10	-0.6082
C	**SC**	1.00	1.00	1.00	1.00	0.98	0.9563
T	**NK**	1.00	1.00	1.00	1.00	1.00	0.9350
	MD	34.00	40.00	74.00	73.00	139.00	-0.9733
	LMSE	0.25	0.39	0.48	0.82	1.61	-0.9912
	NAE	0.03	0.04	0.04	0.06	0.16	-0.9967
	PQS	2.69	1.36	0.15	-1.36	-10.70	1.0000

Table 4.17: Quality Measures Results for 4x4 DTT-compressed Images at Different Bit-rates for *Peppers (SF:13.5756;SAM:100)*

	bpp	2.95	2.18	1.86	1.25	0.07	r_{PQS}
	MSE	2.62	5.96	7.47	12.23	341.22	-0.9995
	RMSE	2.57	3.48	4.07	6.10	104.65	-0.9997
	PSNR	43.96	40.38	39.40	37.26	22.80	0.9690
D	**AD**	0.00	-0.00	0.00	0.00	1.22	-0.9985
T	**SC**	1.00	1.00	1.00	1.00	1.02	-0.9989
T	**NK**	1.00	1.00	1.00	1.00	0.98	0.9992
	MD	7.00	12.00	15.00	21.00	159.00	-0.9997
	LMSE	0.11	0.24	0.29	0.43	1.77	-0.9929
	NAE	0.01	0.02	0.02	0.02	0.12	-0.9991
	PQS	5.04	4.78	4.63	4.23	-7.62	1.0000
	bpp	2.95	2.15	1.93	1.36	0.08	r_{PQS}
	MSE	3.90	7.56	8.81	12.18	524.18	-0.9999
	RMSE	2.57	3.54	3.94	5.60	100.32	-1.0000
	PSNR	42.22	39.34	38.68	37.27	20.94	0.9823
D	**AD**	-0.00	-0.00	-0.00	-0.01	0.27	-0.9981
C	**SC**	1.00	1.00	1.00	1.00	0.97	0.9995
T	**NK**	1.00	1.00	1.00	1.00	1.00	-0.9528
	MD	10.00	15.00	18.00	27.00	155.00	-0.9974
	LMSE	0.17	0.33	0.38	0.48	2.06	-0.9935
	NAE	0.01	0.02	0.02	0.02	0.16	-0.9996
	PQS	5.18	4.99	4.91	4.59	-11.69	1.0000
	bpp	2.98	2.16	1.94	1.36	0.08	r_{PQS}
	MSE	3.92	7.62	8.90	12.42	525.65	-0.9999
	RMSE	2.57	3.54	3.94	5.60	100.32	-1.0000
	PSNR	42.20	39.31	38.64	37.19	20.92	0.9818
I	**AD**	0.00	0.00	-0.00	-0.01	0.28	-0.9982
C	**SC**	1.00	1.00	1.00	1.00	0.97	0.9994
T	**NK**	1.00	1.00	1.00	1.00	1.00	-0.9364
	MD	9.00	15.00	17.00	24.00	155.00	-0.9983
	LMSE	0.17	0.33	0.38	0.48	2.14	-0.9938
	NAE	0.01	0.02	0.02	0.02	0.16	-0.9996
	PQS	5.18	4.99	4.90	4.59	-11.69	1.0000

Table 4.18: Quality Measures Results for 4x4 DTT-compressed Images at Different Bit-rates for *Texture Mosaic (SF:13.58; SAM:952.42)*

	bpp	0.27	0.20	0.17	0.08	0.05	r_{PQS}
	MSE	1.33	4.45	7.96	47.95	96.17	-0.9934
	RMSE	10.96	14.66	17.46	38.38	61.02	-0.9981
	PSNR	46.90	41.64	39.12	31.32	28.30	0.9492
D	**AD**	0.01	0.08	0.01	0.66	-0.10	-0.1262
T	**SC**	1.00	1.00	1.00	1.00	1.01	-0.5714
T	**NK**	1.00	1.00	1.00	1.00	0.99	0.7089
	MD	20.00	39.00	53.00	205.00	217.00	-0.9593
	LMSE	0.07	0.18	0.30	0.95	1.22	-0.9890
	NAE	0.00	0.00	0.01	0.03	0.04	-0.9987
	PQS	5.25	4.67	3.97	-0.31	-3.92	1.0000
	bpp	0.28	0.22	0.18	0.07	0.05	r_{PQS}
	MSE	2.31	6.28	8.80	87.70	184.12	-0.9852
	RMSE	10.59	13.54	16.25	41.96	61.35	-0.9987
	PSNR	44.49	40.15	38.69	28.70	25.48	0.9727
D	**AD**	0.07	0.11	0.08	0.87	-0.09	-0.1717
C	**SC**	1.00	1.00	1.01	1.05	0.99	-0.2675
T	**NK**	1.00	1.00	1.00	0.97	1.00	0.3877
	MD	25.00	42.00	75.00	146.00	203.00	-0.9826
	LMSE	0.12	0.26	0.40	0.91	1.12	-0.9806
	NAE	0.00	0.01	0.01	0.05	0.07	-0.9989
	PQS	5.22	4.41	4.32	-1.75	-5.43	1.0000
	bpp	0.28	0.22	0.19	0.07	0.05	r_{PQS}
	MSE	2.34	6.43	8.96	88.58	184.74	-0.9851
	RMSE	10.59	13.54	16.25	41.96	61.35	-0.9986
	PSNR	44.43	40.05	38.61	28.66	25.47	0.9729
I	**AD**	0.07	0.11	0.08	0.86	-0.09	-0.1688
C	**SC**	1.00	1.00	1.01	1.05	0.99	-0.2706
T	**NK**	1.00	1.00	1.00	0.97	1.00	0.3908
	MD	25.00	46.00	75.00	146.00	200.00	-0.9832
	LMSE	0.12	0.27	0.41	0.93	1.14	-0.9813
	NAE	0.00	0.01	0.01	0.05	0.07	-0.9991
	PQS	5.22	4.39	4.30	-1.76	-5.38	1.0000

Table 4.19: Quality Measures Results for 4x4 DTT-compressed Images at Different Bit-rates for *Airplane (SF:15.1025; SAM:5000)*

	bpp	0.77	0.49	0.38	0.10	r_{PQS}
	MSE	4.77	8.17	12.59	121.22	-0.9932
	RMSE	8.39	13.11	16.97	64.62	-0.9992
	PSNR	41.35	39.01	37.13	27.30	0.9939
D	**AD**	-0.04	-0.02	-0.21	0.61	-0.9250
T	**SC**	1.00	1.00	1.00	1.01	-0.9249
T	**NK**	1.00	1.00	1.00	1.00	0.9514
	MD	16.00	25.00	38.00	126.00	-0.9998
	LMSE	0.13	0.19	0.28	1.41	-0.9975
	NAE	0.01	0.01	0.01	0.04	-0.9986
	PQS	4.16	3.70	3.25	-0.55	1.0000
	bpp	0.78	0.46	0.40	0.11	r_{PQS}
	MSE	5.51	11.19	13.02	131.29	-0.9912
	RMSE	8.27	14.13	16.32	56.51	-0.9995
	PSNR	40.72	37.64	36.98	26.95	0.9958
D	**AD**	0.03	-0.14	-0.03	-0.67	0.9823
C	**SC**	1.00	1.00	1.00	0.99	0.9861
T	**NK**	1.00	1.00	1.00	1.00	-0.9780
	MD	24.00	31.00	38.00	109.00	-0.9980
	LMSE	0.17	0.29	0.30	1.24	-0.9973
	NAE	0.01	0.01	0.01	0.05	-0.9971
	PQS	4.15	3.55	3.37	0.11	1.0000
	bpp	0.79	0.46	0.40	0.11	r_{PQS}
	MSE	5.68	11.48	14.01	133.74	-0.9903
	RMSE	8.27	14.13	16.32	56.51	-0.9992
	PSNR	40.59	37.53	36.67	26.87	0.9958
I	**AD**	-0.02	-0.20	-0.09	-0.68	0.9789
C	**SC**	1.00	1.00	1.00	0.99	0.9849
T	**NK**	1.00	1.00	1.00	1.00	-0.9743
	MD	23.00	34.00	39.00	108.00	-0.9997
	LMSE	0.18	0.30	0.34	1.37	-0.9976
	NAE	0.01	0.01	0.01	0.05	-0.9967
	PQS	4.17	3.55	3.33	0.09	1.0000

Table 4.20: PSNR and RMSE for 4x4 Block Size Compression Experiment at Different Bit-rates

	Mandrill				*Peppers*			
bpp	4.39	3.68	2.68	0.1	3.01	1.86	1.25	0.1
PSNR								
DTT	43.94	40.21	35.27	20.03	43.98	39.40	37.26	23.42
DCT	41.42	37.79	33.40	18.98	42.24	38.44	35.27	21.65
ICT	41.42	37.81	33.42	18.98	42.20	38.40	35.19	21.64
RMSE								
DTT	1.52	2.53	4.45	25.45	1.60	2.73	3.50	17.93
DCT	2.05	3.29	5.45	28.68	1.97	3.06	5.24	21.89
ICT	2.04	3.28	5.44	28.69	1.98	3.07	5.26	21.93

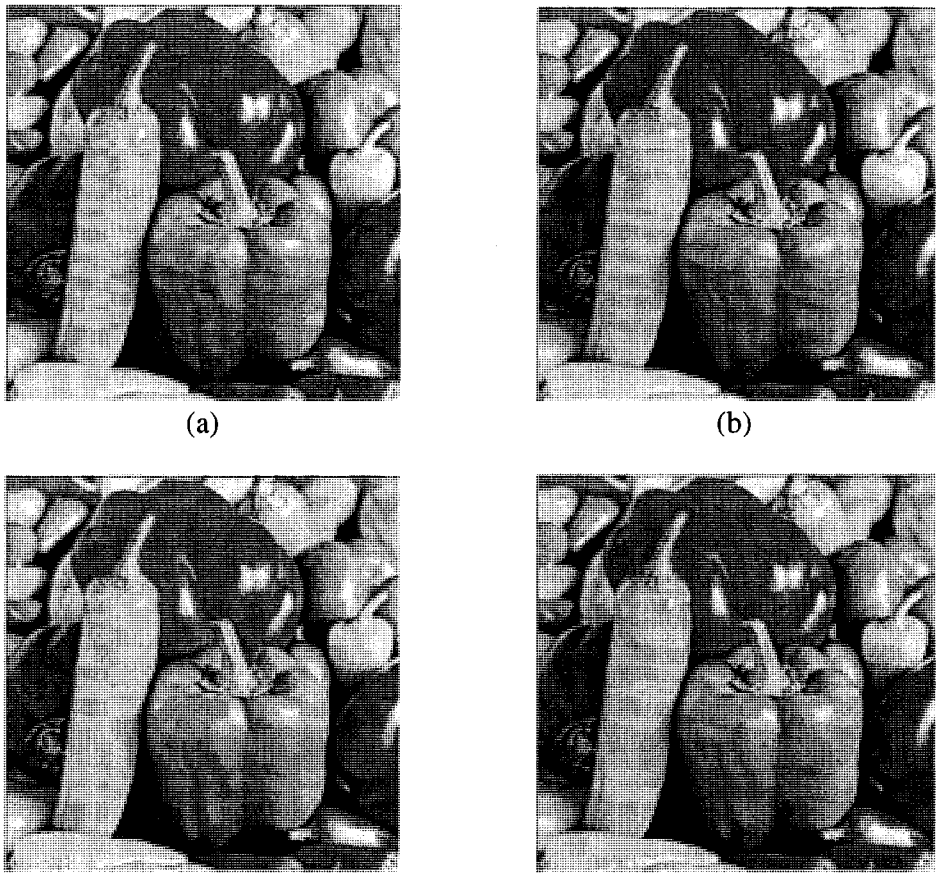


Figure 4.26: Original and Reconstructed Images of *Peppers* using 4x4 DTT (a) Uncompressed, bpp=7.59, (b) DCT-compressed, bpp=1.73, PSNR=38.24, RMSE = 3.12, (c) ICT - compressed bpp=1.74, PSNR=38.19, RMSE=3.14 and (d) DTT - compressed, bpp=1.65, PSNR=38.60, RMSE=2.99

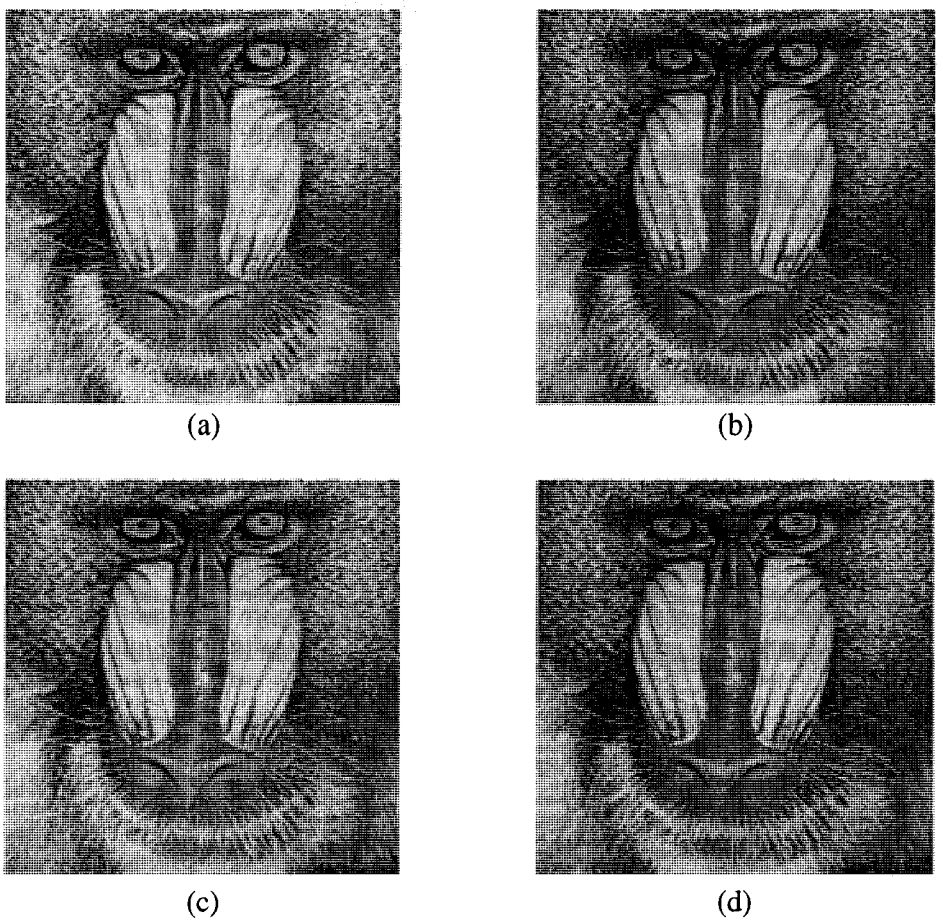


Figure 4.27: Original and Reconstructed Images of *Mandrill* using 4x4 DTT (a) Uncompressed, bpp=7.36, (b) DCT-compressed, bpp=1.58, PSNR=28.84, RMSE = 9.21, (c) ICT-compressed, bpp=1.58, PSNR=28.87, RMSE=9.18 and (d) DTT - compressed, bpp=1.48, PSNR=29.21, RMSE=8.82

4.4.4 DTT Block Sizes 4 and 8- A Comparison

In the literature, documentation for 8×8 DCT seems to be larger than that for 4×4 DCT. In the previous sub-sections, we have compared the performances of 8x8 and 4x4 DTTs with respect to 8×8 and 4x4 DCTs, respectively. We now examine the image compression performances of 8×8 DTT and 4×4 DTT. Figs. 4.28(a)-(d) give the reconstructed images of *Mandrill* for 4x4 and 8x8 DTTs at the indicated bits per pixel. The plots of PSNR vs

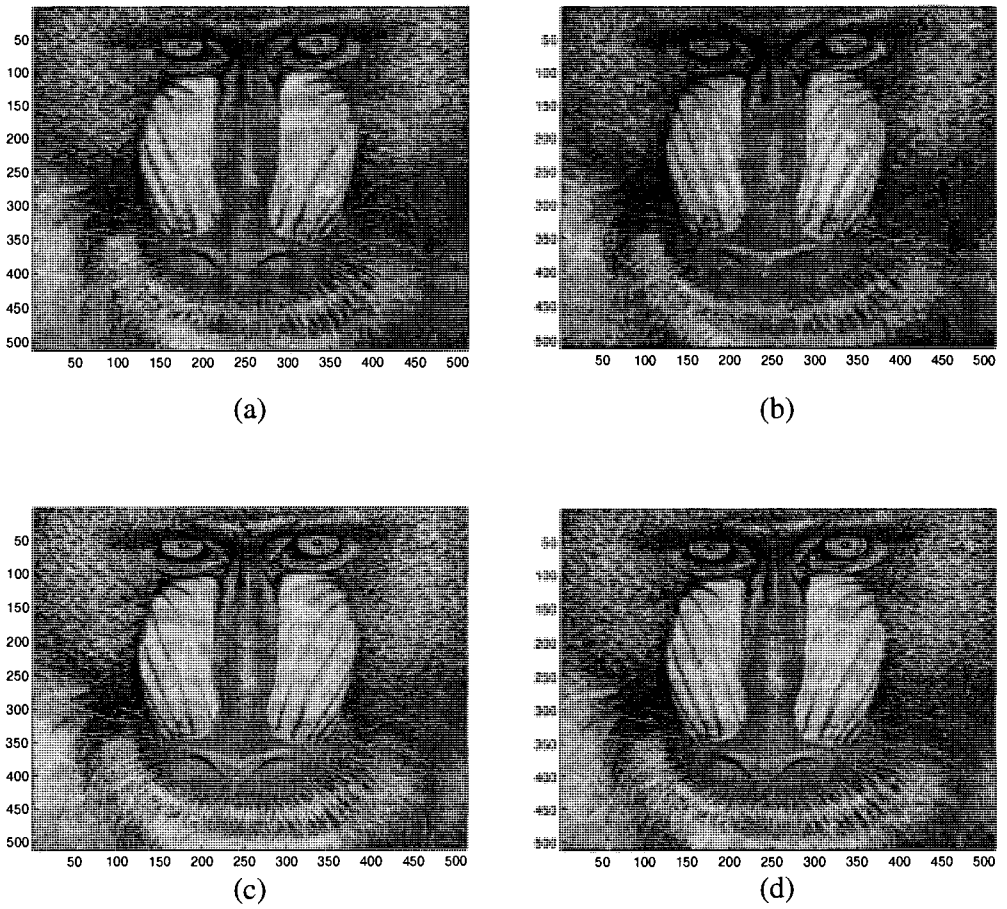


Figure 4.28: Reconstructed Images of *Mandrill* using (a) 8x8 DTT at 0.1 bpp, (b) 4x4 DTT at 0.1 bpp, (c) 8x8 DTT at 2.4 bpp and (d) 4x4 DTT at 2.5 bpp

bpp reconstructed by 4x4 and 8x8 DTTs are given in Figs. 4.29 for the images *Airplane*,

Mandrill, *Cameraman*, *Peppers*, *Einstein* and *Lena*. From these plots and from the results in the previous sub-sections for 4x4 DTT in Figs. 4.19-4.25 and for 8x8 DTT in Figs. 4.13-4.17 that 4×4 DTT has a higher gain in performance over 4×4 DCT, than 8×8 DTT has over 8×8 DCT. Higher resolution at low frequencies is a direct consequence of larger block size.

Figs. 4.28 (a) and (b) are compressed at 0.1 bits per pixel by 8x8 and 4x4 DTTs respectively. By comparing these two figures, we see that Fig. 4.28(b) is more degraded and has higher blocking artifacts than Fig. 4.28 (a). Figs. 4.28 (c) and (d) are compressed at 2.4 and 2.5 bits per pixel by 8x8 and 4x4 DTTs respectively. By comparing these two figures to the previous two figures, we see that 4×4 DTT induces more noticeable ringing artifacts and blocking artifacts when compared to 8×8 DTT. At very low bit-rates, 4x4 DTT has a lower performance than 8x8 DTT, and at higher bit-rates, for images with a high SFM and a low SAM (in (4.2)), the performance of 8x8 DTT is better than 4x4 DTT on a perceptual basis, i.e., although 4x4 DTT scores over 8x8 DTT in objective measures, the visual quality of 8x8 DTT is better than 4x4 DTT at higher bit-rates. For images with a lower SFM and a higher SAM (in (4.2)), the performance of 4x4 DTT is significantly lower than that of 8x8 DTT. From Figs. 4.19-4.25 and Figs. 4.13-4.17, it can be seen that there is a gain of almost 2dB in 8x8 DTT coded images as compared to 4x4 DTT coded images. A possible explanation for this behavior could be that since images with a high SFM and a low SAM are not very predictable, they are difficult to code. There is a higher correlation among the 4x4 transformed sub-blocks compared to the 8x8 transformed sub-blocks. This correlation is exploited during the differential coding of the DC coefficient (which is representative of the sub-block, since it has the maximum information in that sub-block), where predictive coding is used. This results in lower compression ratios for images with a high SFM and a low SAM, if better visual quality has to be preserved.

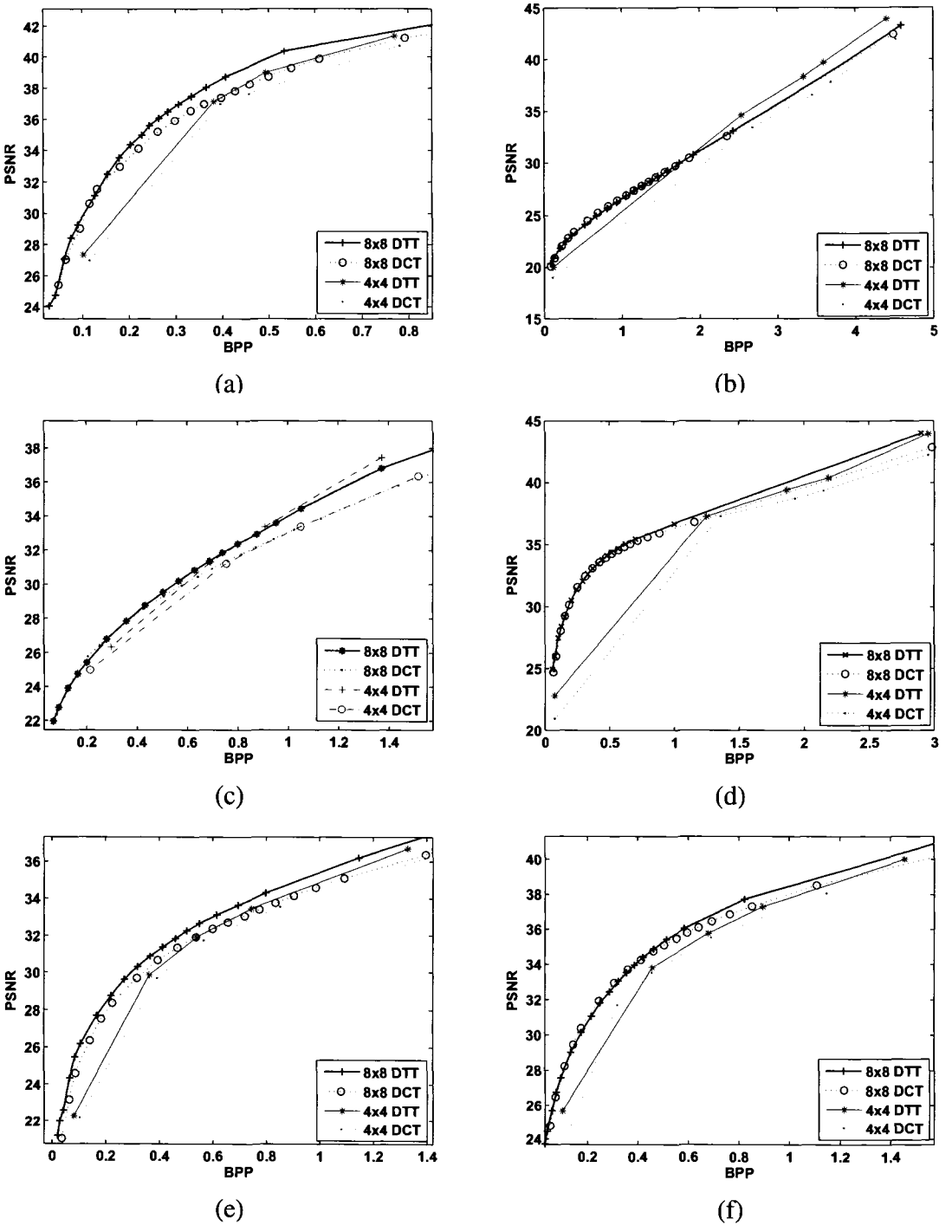


Figure 4.29: PSNR and RMSE performances of 4x4 and 8x8 DTTs and DCTs for Images (a) *Airplane*, (b) *Mandrill*, (c) *Cameraman*, (d) *Peppers*, (e) *Einstein* and (f) *Lena*

Chapter 5

Conclusions and Future Work

This thesis has been concerned with the study of the discrete Tchebichef transform, which is a relatively new transform compared to the more popular transforms such as the discrete cosine and wavelet transforms. The interesting properties of the DTT motivated us to conduct research on the DTT to evaluate its potential in applications based on discrete orthogonal transforms.

5.1 Summary

In the first half of the thesis, we have concentrated on the formulation, properties and fast algorithms for the DTT. The later half of the thesis is dedicated to the experimental study of image compression using DTT and evaluation of its performance for the same.

We have derived important properties such as orthogonality, symmetry and recurrence relations for the discrete Tchebichef polynomials. Based on these properties, we have presented the discrete Tchebichef transform with new notations that conform to the language of discrete transforms. Some important properties of DTT such as the energy property, unitary property and separability property, which lay the foundation for using DTT in transform coding applications, have also been derived. (Although the end results (of the

properties) have been mentioned elsewhere in the literature, the derivations and proofs for the Tchebichef case are not documented to the knowledge of the author of this thesis, and hence, have been presented here.)

Since the polynomial kernel of DTT allows for a multiplier-free approach with low-complexity and without any approximation errors, DTT is more appropriate for synthesis of an integer transform than DCT. The popularity of integer-based algorithms which offer highly simplified designs for implementation, and the significance of multiplier-free designs which bypass the restrictions of processors, find their application in this thesis as the integer Tchebichef transform and its multiplier-free design. We have proposed the concept of the integer Tchebichef transform, and designed a general framework to render 4×4 and 8×8 DTTs into their respective integer transforms. We have derived fast multiplier-free algorithms based on ITT for 4-point and 8-point DTTs, and have given a design for the hardware implementation of the 8-point algorithm. The proposed DTT algorithms have been found to be computationally more efficient than the existing algorithm for DTT. We have also compared the computational aspects of ITT and ICT to demonstrate the ease of use of the ITT over the ICT for video coding applications based on integer-based arithmetic, such as the MPEG-4 part 10 codec.

We have performed image compression using 4x4 and 8x8 DTTs by employing a practical scheme which is comparable to the existing standards. The image quality measures meant for the picture quality evaluation of gray scale-compressed images span a wider perspective of quality analysis compared to the conventional quality metrics. We have analyzed the performance of DTT-compressed images using these quality measures based on perceptual techniques in addition to the customarily-used objective techniques. Since the visual quality of coded images is essentially determined by the structural properties of an image, we have integrated these concepts in our analysis of the performance of DTT. We have compared the performances of compression using 4x4 DTT with those using DCT

and ICT. These experiments have generated interesting results. In general, a transform seizes to perform well for images with a high spectral activity and a low predictability. In our analysis, we have found that 4x4 DTT performs better than 4x4 DCT and 4x4 ICT for these kinds of images compared to images which are relatively easier to code. The *video quality expert group* has advocated the use of PSNR for video quality measurement methods [40] based on their comprehensive experiments. Our results have shown that 4x4 DTT exhibits a performance gain of 0.5dB - 2.5dB over that of 4x4 DCT and 4x4 ICT. Based on the encouraging results of image compression as well as the computational ease of DTT explained above, image compression using DTT has been proposed and the notion of video compression using DTT has been envisaged. The performances of 8x8 DTT and DCT have been compared. Our quality analysis shows that the performances of 8x8 DTT and DCT are almost similar to each other for perceptual quality measures, but a significant gain in performance is seen for DTT in the case of objective measures. However, we have found that unlike 4x4 DTT, 8x8 DTT has a relatively less gain in performance over DCT for images with a high spectral activity. But, we have also found that the visual quality of 8x8 DTT-compressed images is better than that of 4x4 DTT-compressed images.

5.2 Future Work

Compression algorithms drastically reduce the storage and bandwidth requirements. The implementation of high-performance algorithms, such as those obtained from discrete Tchebichef transform can help reduce the cost requirements. Since DTT is a novel transform with very interesting characteristics, the areas of its application are many. In this thesis, we have already laid a foundation for the application of DTT in video compression. Hence, the scope for the implementation of a complete video codec based on DTT is apparent. Based on methods such as the hexagonal fast Fourier transform, experiments on shifted Tchebichef

polynomials and non-linearly sampled Tchebichef polynomials may result in better compression performances. There seems to be good scope for application of DTP in filter banks. The scheme used for image compression in this thesis uses block coding, which is known to induce blocking artifacts due to block boundaries. It would certainly be interesting to see the results of DTT based on a non-block coding scheme, such as the lapped orthogonal transforms. Also, the analysis of DTT for noisy images is one of the possible research directions in future. It is possible to obtain a better gain in performance for DTT by using rate-distortion optimized quantization tables and also by studying a statistical model for the distribution of the DTT coefficients. While the strength of fast DCT designs lie in exploiting trigonometric properties, fast DTT designs can be developed by maneuvering around polynomial properties. Since polynomial theory is very well-developed, there is a possibility of developing more efficient DTT algorithms.

References

[1] K. M. Cheung, M. Belongie, and K. Tong. (1996, August) End-to-end system consideration of the Galileo image compression system. TDA Progress Report. [Online]. Available: http://ipnpr.jpl.nasa.gov/progress_report/42-126/126E.pdf

[2] R. Mukundan, S. H. Ong, and P. A. Lee, "Image analysis by Tchebichef moments," *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1357–1364, September 2001.

[3] P. Hew, "Geometric and Zernike moments," October 1996. [Online]. Available: citeseer.ist.psu.edu/hew96geometric.html

[4] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transfom," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.

[5] T. H. Oh and R. Besar, "JPEG2000 and JPEG: Image quality measures of compressed medical images," *Proceedings of the 4th National Conference on Telecommunication Technology, 2003 (NCTT 2003)*, pp. 31–35, January 2003.

[6] Feig, Ephraim, Winograd, and Shmuel, "Fast algorithms for the discrete cosine transform," *IEEE Trans. Signal Process.*, vol. 40, no. 9, pp. 2174–2193, 1992.

[7] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1989. (ICASSP-89)*, pp. 988–991 vol.2, May 1989.

[8] Y. Arai, T. Agui, and M. Nakajima, "A fast DCT-SQ scheme for images," *Trans. IEICE*, vol. E-71, no. 11, pp. 1095–1097, 1988.

[9] P. Corr, D. Stewart, P. Hanna, J. Ming, and F. Smith, "Discrete Chebyshev transform. a natural modification of the DCT," *Proceedings of the 15th International Conference on Pattern Recognition, 2000.*, vol. 3, pp. 1142–1145 vol.3, 2000.

[10] K. Nakagaki and R. Mukundan, "A fast 4x4 forward discrete Tchebichef transform algorithm," *IEEE Signal Process. Lett.*, vol. 14, pp. 684–687, October 2007.

[11] A. Erdelyi, W.Magnus, F. Oberhettinger, and G. F. Tricomi, *Higher Transcendental Functions. Vol. II.* Bateman Manuscript Project. New York-Toronto-London: McGraw-Hill Book Co., Inc. XVII, 396 p. , 1953.

[12] A. F. Nikiforov, S. K. Suslov, and V. B. Uvarov, *Classical Orthogonal Polynomials of a Discrete Variable*. Springer, Berlin, 1991.

[13] H. Rosengren, "Multivariable orthogonal polynomials and coupling coefficients for discrete series representations," *SIAM J. Math. Anal.*, vol. 30, no. 2, pp. 233–272, 1999.

[14] D. D. Fazio, S. Cavalli, and V. Aquilanti, "Orthogonal polynomials of a discrete variable as expansion basis sets in quantum mechanics: Hyperquantization algorithm," *International Journal of Quantum Chemistry*, vol. 93, pp. 91–111, March 2003.

[15] P. Delsarte and V. I. Levenshtein, "Association schemes and coding theory," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2477–2504, October 1998.

[16] W. Koepf, "Identities for families of orthogonal polynomials and special functions," *Integral Transforms and Special Functions*, vol. 5, pp. 69–102, May 1997.

[17] G. Szego, *Orthogonal polynomials. 4th ed.* American Mathematical Society Colloquium Publications. Vol. XXIII., 1975.

[18] C. Chong, P. Raveendran, and R. Mukundan, "Translation invariants of Zernike moments," *Pattern Recognition*, vol. 36, pp. 1765–1773(9), August 2003.

[19] S. Ishwar, P. K. Meher, and M. N. S. Swamy, "Quality assessment of discrete Tchebichef transform compressed images, integer Tchebichef transform and a multiplier-free algorithm," under review.

[20] ——, "Discrete Tchebichef transform-A fast 4x4 algorithm and its application in image/video compression," in *IEEE International Symposium on Circuits and Systems*, 2008, pp. 260–263.

[21] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," *IEEE Trans. Circuits Syst.*, vol. 13, pp. 598–603, July 2003.

[22] A. Hallapuro, M. Karczewicz, and H. S. Malvar, *Low Complexity Transform and Quantization Part 1: Basic Implementation*, JVT-B038, February 2002.

[23] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression.* John Wiley and Sons, 2003.

[24] H. S. Malvar, *Low-Complexity Length-4 Transform and Quantization with 16-Bit Arithmetic*, VCEG-N44, September 2001.

[25] N. I. Cho and S. U. Lee, "A fast 4x4 DCT algorithm for the recursive 2-D DCT," *IEEE Trans. Signal Process.*, vol. 40, pp. 2166–2173, September 1992.

[26] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2003.

[27] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard.* Norwell, MA, USA: Kluwer Academic Publishers, 1992.

[28] M. T. Uwe, "New perspectives on image compression." [Online]. Available: citeseer.ist.psu.edu/693069.html

[29] *Digital Compression and Coding of Continuous tone Still Images*, International Organization for Standards / International Electrotechnical Commission (ISO/EC) International Standard 10918-1, 1993.

[30] L. P.Y. [Online]. Available: http://www.eng.tau.ac.il/~yaro/

[31] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. xviii–xxxiv, February 1992.

[32] (1977-2007) USC-SIPI image database. [Online]. Available: http://sipi.usc.edu/database/

[33] (2002-2007) CVG-UGR image database. [Online]. Available: http://decsai.ugr.es/cvg/dbimagenes/g256.php

[34] A. Eskicioglu, "Quality measurement for monochrome compressed images in the past 25 years," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2000 (ICASSP '00)*, 2000, pp. 1907–1910.

[35] N. Nill, "A visual model weighted cosine transform for image compression and quality assessment," *IEEE Trans. Commun.*, vol. 33, no. 6, pp. 551–557, June 1985.

[36] M. Mrak, S. Grgic, and M. Grgic, "Picture quality measures in image compression systems," *International IEEE Region 8 Conference on Computer as a tool EUROCON 2003.*, vol. 1, pp. 233–236, September 2003.

[37] A. M. Eskicioglu and P. S. Fisher, "Image quality measures and their performance," *IEEE Trans. Commun.*, vol. 43, no. 12, pp. 2959–2965, December 1995.

[38] M. Miyahara, K. Kotani, and V. R. Algazi, "Objective picture quality scale (PQS) for image coding," *IEEE Trans. Commun.*, vol. 46, no. 9, pp. 1215–1226, September 1998.

[39] *Methods for the Subjective Assessment of the Quality of Television Pictures*, ITU-R Rec. BT. 500-7, August 1998.

[40] A. M. Rohaly, P. J. Corriveau, J. M. Libert, A. A. Webster, V. Baroncini, J. Beerends, J.-L. Blin, L. Contin, T. Hamada, D. Harrison, A. P. Hekstra, J. Lubin, Y. Nishida, R. Nishihara, J. C. Pearson, A. F. Pessoa, N. Pickford, A. Schertz, M. Visca, A. B. Watson, and S. Winkler, "Video quality experts group: current results and future directions," in *VCIP*, 2000, pp. 742–753.

Appendices

Appendix A

Miscellaneous Identities and Formulae

Difference Operator

Δ, the forward difference operator and ∇, the backward difference operator, are given by

$$\Delta f(x) = f(x+1) - f(x) \tag{A.1}$$

$$\nabla f(x) = f(x) - f(x-1) \tag{A.2}$$

and are related by

$$\nabla f(x) = \Delta f(x) - \Delta \nabla f(x) \tag{A.3}$$

Binomial Coefficients

$$\binom{n}{k} = \frac{n!}{(n-k)!k!}, \quad \binom{n}{0} = 1 \tag{A.4}$$

$$\binom{n}{k} = \binom{n}{n-k} = (-1)^k \binom{k-n-1}{k} \tag{A.5}$$

$$\binom{n}{k+1} = \binom{n}{k} \frac{n-k}{k+1} \tag{A.6}$$

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1} \tag{A.7}$$

Gamma Function

The gamma function, also known as the generalized factorial is denoted by $\Gamma(x)$. The gamma function can be evaluated recursively by using

$$\Gamma(x+1) = x\Gamma(x) \tag{A.8}$$

$$\Gamma(1) = 1 \tag{A.9}$$

When x is a positive integer

$$\Gamma(x+1) = x! \tag{A.10}$$

Rising and Falling Factorial

$(a)_{(n)}$ is the Pochammer symbol, also known as the rising factorial and is given by

$$(a)_{(n)} = a(a+1)(a+2).....(a+n-1) \tag{A.11}$$

$$= \frac{(a+n-1)!}{(a-1)!} \qquad \text{for } n \geq 1 \tag{A.12}$$

$$(a)_{(0)} = 1,\ (1)_{(n)} = (n)_{(n)} = n! \tag{A.13}$$

where n, m are positive integers. Some more useful relations for the rising factorial are

$$(a)_{(n)} = \frac{\Gamma(a+n)}{\Gamma(a)}$$

$$\frac{(a)_{(n)}}{n!} = \binom{x+n-1}{n} \tag{A.14}$$

$$(-a)_{(n)} = (-1)^n (a)^{(n)}, \tag{A.15}$$

where $(a)^{(n)}$ is the falling factorial given by

$$\frac{(a)^{(n)}}{n!} = \binom{a}{n} \tag{A.16}$$

Appendix B

Some Properties of a Diagonal Matrix

The well-known properties of diagonal matrices used in this thesis are given below.

Let **A**, **B** and **C** be matrices of orders such that the given operations are defined.

Lemma B.1 (Product of Diagonal Matrices) *Let* **D** *and* **E** *be diagonal matrices of the same order* $M \times M$. *The product of* **D** *and* **E** *is a third diagonal matrix,* **F** *of order* $M \times M$, *with diagonal entries equal to the products of the corresponding entries of* **D** *and* **E**, *i.e., if the diagonal elements of* **F**, **D** *and* **E** *be* f_{kk}, d_{kk} *and* e_{kk} *respectively for the kth row, then*

$$\mathbf{F} = \mathbf{DE} \quad implies \quad f_{kk} = d_{kk}e_{kk}, \qquad where \quad k = 1 \cdots M.$$

The proof is straightforward and is omitted.

Lemma B.2 (Pre-multiplication by a Diagonal Matrix) *Pre-multiplication of a matrix* **A**, *of order* $M \times N$ *by a diagonal matrix,* **D** *of order* $M \times M$, *results in multiplication of the entries in each row of* **A** *by the row-corresponding diagonal element in* **D** *i.e., if* d_{kk} *be the diagonal element in the kth row of* **D**, *and* a_{kj} *and* b_{kj} *be the elements in the kth row and the jth column of* **A** *and* **B** *respectively, then*

$$\mathbf{B} = \mathbf{DA} \quad implies \quad b_{kj} = d_{kk}a_{kj}, \qquad where \quad k = 1 \cdots M \ and \ \ j = 1 \cdots N.$$

The proof is straightforward and is omitted.

Lemma B.3 (Post-multiplication by a Diagonal Matrix) *Post-multiplication of a matrix* **A**, *of order* $N \times M$ *by a diagonal matrix* **D**, *of order* $M \times M$, *multiplies the entries in each column of* **A**, *by the column-corresponding diagonal element in* **D**, *i.e., if* d_{kk} *be the diagonal element in the kth row of* **D**, *and* a_{kj} *and* b_{kj} *be the elements in the kth row and the jth column of* **A** *and* **B** *respectively, then*

$$\mathbf{B} = \mathbf{AD} \text{ implies } b_{kj} = d_{jj}a_{kj}, \qquad \text{where } k = 1 \cdots N \text{ and } j = 1 \cdots M.$$

The proof is straightforward and is omitted.

Theorem B.1 (Pre-post-multiplication by a Diagonal Matrix) *If* **D** *is a diagonal matrix, and* **A** *is a matrix of the same order as that of* **D**, *then* $\mathbf{DAD}' = \mathbf{A} \odot \mathbf{Q}$, *where* q_{kj} *is the element in the kth row and the jth column of* **Q** *is given by* $q_{kj} = d_{kk}d_{jj}$ *for* $k, j = 1 \cdots N$.

Proof:

Let $\mathbf{B} = \mathbf{DAD}'$. Consider $\mathbf{C} = \mathbf{DA}$.

Using Lemma B.2, $c_{kj} = d_{kk}a_{kj}$.

Now $\mathbf{B} = \mathbf{CD}'$. Using Lemma B.3, $b_{kj} = d_{jj}c_{kj}$

or $b_{kj} = d_{jj}d_{kk}a_{kj}$

$\Rightarrow b_{kj} = q_{kj}a_{kj}$, where $q_{kj} = d_{jj}d_{kk}$. (B.1)

or $\mathbf{B} = \mathbf{A} \odot \mathbf{Q}$, where $\odot$ represents element-by-element multiplication. (B.2)

Appendix C

Definitions

Hue

The hue of a color identifies what is commonly called color.

Saturation

The saturation of a color represents the intensity of the color.

Brightness

Brightness is an attribute of visual perception, in which a source appears to emit a given amount of light.

Luminance

The luminance of a color is a measure of its perceived brightness.

Chrominance

The chrominance of a color is the difference of the color with respect to a specified reference color.

DC coefficient

The DC coefficient is the component in the image, which has a zero frequency with respect to the rows and columns of the image.

Transform coefficient

Transform coefficient refers to the transformed value as represented by the basis of the transform. With respect to the Tchebichef transform, the Tchebichef-transformed coefficient is the amplitude of the input when Tchebichef-transformed, as seen in the Tchebichef domain.

Tchebichef transform pair

The definitions of the DTT and the inverse DTT combined are called the Tchebichef *transform pair* i.e.,

$$Y(k) = \sum_{n=0}^{N-1} \tau(k,n)x(n) \qquad \text{for } k,n = 0,1 \cdots N-1 \tag{C.1}$$

and

$$x(n) = \sum_{k=0}^{N-1} \tau(k,n)Y(k) \qquad \text{for } k,n = 0,1 \cdots N-1 \tag{C.2}$$

are called the Tchebichef transform pair denoted by $x(n) \overset{DTT}{\longleftrightarrow} Y(k)$.

Entropy

Entropy quantifies the amount of information contained in a set of data and is given by

$$Entropy = -\sum_{i=0}^{M-1}\sum_{j=0}^{L-1} P(i,j)logP(i,j) \tag{C.3}$$

where M and L are the number of rows and columns of pixels, and $P(i, j)$ is the probability of the occurrence of the pixel (i, j).

Compression ratio

Compression ratio is the ratio of the size of the original image to that of the compressed image. A trade-off between the compression ratio and the quality of the image is the key to decide the level of compression.

Precision

The precision of a value is determined by the number of bits allocated to a particular sample.

Histogram plot

A histogram plot is a graph that shows the number of pixels in an image at each different intensity value found in that image. The number of intensity values depends on the image precision.