Development of 2-D MRT based Image Compression Techniques

A THESIS

Submitted by

Anish Kumar M S

for the award of the degree of

DOCTOR OF PHILOSOPHY

Under the guidance of

Prof. R. GOPIKAKUMARI

DEPARTMENT OF ELECTRONICS
FACULTY OF TECHNOLOGY
COCHIN UNIVERSITY OF SCIENCE & TECHNOLOGY
KOCHI – 682 022, INDIA

August 2013

Development of 2-D MRT based Image Compression Techniques

PhD Thesis under the Faculty of Technology

Author
Anish Kumar M S
Research Scholar
Department of Electronics
Cochin University of Science and Technology
E-mail: anishdoecusat@gmail.com

Supervising Guide
Prof. R Gopikakumari
Division of Electronics Engineering
School of Engineering
Cochin University of Science and Technology
E-mail: gopika@cusat.ac.in

Department of Electronics
Cochin University of Science and Technology
Kochi – 682 022

DIVISION OF ELECTRONICS ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
KOCHI – 682 022, KERALA, INDIA

13th August 2013

CERTIFICATE

This is to certify that the thesis entitled “**Development of 2-D MRT based Image Compression Techniques”** is a bona fide record of the research work carried out by **Anish Kumar M S**. under my supervision and guidance in the **Department of Electronics, Cochin University of Science and Technology** and that no part thereof has been presented for the award of any other degree.

Prof. R. Gopikakumari
(*Supervising Guide)*

DECLARATION

I hereby declare that the work presented in the thesis entitled "**Development of 2-D MRT based Image Compression Techniques"** is based on original research work carried out by me under the supervision and guidance of Prof. (Dr.) R. Gopikakumari in the **Department of Electronics, Cochin University of Science and Technology** and that no part thereof has been presented for the award of any other degree.

Kochi
13th August 2013

Anish Kumar M S.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervising guide Dr. R. Gopikakumari, Professor & former Head, Division of Electronics Engineering, School of Engineering, Cochin University of Science and Technology, for her invaluable guidance, advice, timely care and parental concern extended towards me during my entire research period and providing me with an excellent atmosphere for doing research. Her flawless way of thinking helped me to explore my research abilities. She gave freedom to pursue research on my line of thought. It was something which helped me to achieve progress by self-analysis of the work. I was able to successfully finish the work and deliver this thesis only because of her able guidance and immense patience.

I am much grateful to Dr. C. K. Aanandan, Professor & Head, Department of Electronics, Cochin University of Science and Technology, for his constant support, concern and extending the facilities of the department.

I would like to express my sincere thanks to Dr. K. Vasudevan, Professor & former Head, Department of Electronics, Cochin University of Science and Technology, for his support, timely advice and concern throughout the entire period of my research.

I am extremely thankful to Dr. Tessamma Thomas, Professor, Department of Electronics, Cochin University of Science and Technology, for the support and care rendered during my research career. Her timely advices were a boon.

I am much indebted to all my senior research fellows, especially Dr. Rajesh Cheriyan Roy, Dr. Bhadran V. and Dr. K. Meenakshy, who lead me to experience the research in the field and practical issues beyond textbooks. Their support and advices were a great help for me.

I take this opportunity to express my sincere thanks to Dr. Deepa Sankar, Division of Electronics engineering, School of Engineering, Cochin University of Science and Technology, for her support. Her timely advices gave me confidence.

Sincere thanks to my co-workers Mrs. Jaya V. L. and Mrs. Preetha Basu, the fruitful discussions with them enriched my knowledge.

Thanks are due to my friends Mr. Ullas G. Kalappura and Mr. Lindo A. O., for their extreme help and immense support for my research work. Only because of them, I was able to submit my thesis.

Sincere thanks to my friends Dr. Shynu S. V., Dr. Praveen Kumar A. V., Dr. Anupam R. Chandran, Dr. Robin Augustine and Mr. Paulbert Thomas for their open hearted talks and discussions which made a homely environment in the department.

I thankfully bear in my mind the sincere co-operation I received from all non-teaching staff of the Department of Electronics & Division of Electronics Engineering. Special regards to Mr. Rajeev, Mr. Siraj, and Mr. Ibrahimkutty for their amicable relation and innumerable helps.

I sincerely acknowledge the Kerala State Council for Science Technology and Environment (KSCSTE) for the financial support they provided in the form of fellowships.

My beloved Sanathana and Sarovar Hostels and inmates, for creating the perfect atmosphere of a campus hostel.

I am grateful to Appa, Amma, Anu and Ambily for their deep love, care and patience to move on with my research work. They were always with me as a constant source of energy. Without their blessings, I couldn’t achieve my aim. I am grateful to my wife, Mrs Resmy for giving me the motivation, courage and confidence during my research. Thanks to my daughter, Nitha for innocent smile, that made me relaxed.

I finally thank you, the reader of this thesis for sparing your time to go through my findings in search of knowledge.

Above all, I thank Almighty for his blessings showered on me for the successful completion of the thesis work.

Anish Kumar M S

ABSTRACT

The thesis explores the area of still image compression. The image compression techniques can be broadly classified into lossless and lossy compression. The most common lossy compression techniques are based on Transform coding, Vector Quantization and Fractals. Transform coding is the simplest of the above and generally employs reversible transforms like, DCT, DWT, etc. Mapped Real Transform (MRT) is an evolving integer transform, based on real additions alone. The present research work aims at developing new image compression techniques based on MRT.

Most of the transform coding techniques employ fixed block size image segmentation, usually 8×8. Hence, a fixed block size transform coding is implemented using MRT and the merits and demerits are analyzed for both 8×8 and 4×4 blocks. The N^2 unique MRT coefficients, for each block, are computed using templates. Considering the merits and demerits of fixed block size transform coding techniques, a hybrid form of these techniques is implemented to improve the performance of compression. The performance of the hybrid coder is found to be better compared to the fixed block size coders. Thus, if the block size is made adaptive, the performance can be further improved. In adaptive block size coding, the block size may vary from the size of the image to 2×2. Hence, the computation of MRT using templates is impractical due to memory requirements. So, an adaptive transform coder based on Unique MRT (UMRT), a compact form of MRT, is implemented to get better performance in terms of PSNR and HVS.

The suitability of MRT in vector quantization of images is then experimented. The UMRT based Classified Vector Quantization (CVQ) is implemented subsequently. The edges in the images are identified and classified by employing a UMRT based criteria.

Based on the above experiments, a new technique named “MRT based Adaptive Transform Coder with Classified Vector Quantization (MATC-CVQ)”is developed. Its performance is evaluated and compared against existing techniques. A comparison with standard JPEG & the well-known Shapiro’s Embedded Zero-tree Wavelet (EZW) is done and found that the proposed technique gives better performance for majority of images.

TABLE OF CONTENTS

Abbreviations

List of Figures

List of Tables

Chapter 1: Introduction to Still Image Compression........1

Chapter 2: Review of the Past work in Image Compression........39

Chapter 3: MRT based Fixed Block size Transform Coding........63

Chapter 4: Variable Block size Transform coders using MRT........101

Chapter 5: MRT based Adaptive Transform Coder with Classified Vector Quantization (MATC-CVQ)........127

Chapter 6: Summary, Conclusion and Future work........163

References........169

List of Publications........191

Index........193

About the Author

ABBREVIATIONS

bpp	Bits per pixel
CCD	Charge Coupled Device
CVQ	Classified Vector Quantization
DCT	Discrete Cosine Transform
DFrFT	Discrete Fractional Fourier Transform
DFT	Discrete Fourier Transform
DST	Discrete Sine Transform
DWHT	Discrete Walsh-Hadamaard Transform
DWT	Discrete Wavelet Transform
EZW	Embedded Zero-tree Wavelet
FrFT	Fractional Fourier Transform
GIF	Graphics Interchange Format
GLA	Generalized Lloyd Algorithm
IMRT	Inverse Mapped Real Transform
JPEG	Joint Photographic Experts Group
KLT	Karhunen-Loeve Transform
LBG	Linde-Buzo-Gray

LOT	Lapped Orthogonal Transforms
MATC	MRT based Adaptive Transform Coding
MATC-CVQ	MRT based Adaptive Transform Coding with Classified Vector Quantization
MRT	Mapped Real Transform
MSE	Mean Squared Error
PCA	Principal Component Analysis
PSNR	Peak Signal to Noise Ratio
QMF	Quadature Mirror Filters
QT	Quad- Tree
RGB	Red Green Blue
SA-DCT	Shape-Adaptive DCT
SNR	Signal to Noise Ratio
SPIHT	Set Partitioning in Hierarchical Trees
SVD	Singular Value Decomposition
TIFF	Tagged Image File Format
UMRT	Unique Mapped Real Transform
VQ	Vector Quantization

LIST OF FIGURES

Figure 1.1: An alternate view of image data 4

Figure 1.2: A transform based image compression system 9

Figure 1.3: The Lena image and its Quad-Tree based segmented form 10

Figure 1.4: Diagram of the 5 DCT basis functions or primitives 29

Figure 1.5: Wavelet filtering 31

Figure 1.6: Third stage of wavelet filtering 31

Figure 1.7: Power distribution produced by a wavelet transform 32

Figure 1.8: Demonstration of simple vector quantization 33

Figure 1.9: Fractal transform (without rotation considered) 34

Figure 3.1: A Transform Coding System, (a) Encoder and (b) Decoder 64

Figure 3.2: Indices k1 k2 p of unique coefficients in 8×8 MRT matrix 68

Figure 3.3: Template for the computation of MRT coefficient $Y_{0,1}^{(0)}$ 70

Figure 3.4: Block diagram of 8×8 MRT based transform coder 70

Figure 3.5: Illustration of coding an 8×8 sub-image 73

Figure 3.6: Decompression block diagram 74

Figure 3.7: Lena (256×256) (a) original and (b) – (d) reconstructed after compressed using 8×8 MRT based transform coder with thresholds 100, 150 & 200 75

Figure 3.8: Threshold versus bpp for Lena (256×256) 75

Figure 3.9: Threshold versus PSNR for Lena (256×256) 76

Figure 3.10: bpp versus PSNR for image Lena (256×256) 76

Figure 3.11: Templates for computing unique coefficients of 8×8 MRT....78

Figure 3.12: Lena (512×512) (a) original and (b) – (h) reconstructed after compressed using 8×8 MRT based transform coder with threshold varied from 100 to 400 in a step of 50....82

Figure 3.13: Baboon (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 1.32, PSNR 25.87)....83

Figure 3.14: Barbara (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.75, PSNR 28.58)....83

Figure 3.15: Boat (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.63, PSNR 28.89)....84

Figure 3.16: Bridge and stream (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 1.01, PSNR 26.29)....84

Figure 3.17: Couple (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.63 , PSNR 30.11)....85

Figure 3.18: Elaine (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.40 , PSNR 29.40)....85

Figure 3.19: Goldhill (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.50, PSNR 29.09)....86

Figure 3.20: Peppers (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.53, PSNR 30.17)....86
Figure 3.21: Cameraman (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.49, PSNR 31.13)....87
Figure 3.22: (a) – (d) Original Color images and (e) – (h) their reconstructed versions after compressed using 8×8 MRT based transform coder 90
Figure 3.23: Templates for the computation of the 16 unique coefficients of 4×4 MRT.... 91
Figure 3.24: The 4 ×4 MRT based transform coding scheme 92
Figure 3.25: Lena (512×512) (a) original and (b) – (i) reconstructed after compressed using 4×4 MRT based transform coder with bpp 0.91, 0.58, 0.44, 0.38, 0.34, 0.32, 0.31 and 0.30 respectively....95
Figure 3.26: (a) – (d) Original Color images and (e) – (h) their reconstructed versions after compressed using 4×4 MRT based transform coder....98
Figure 3.27: Performance comparison of 4×4 and 8×8 MRT based coders for Lena image....100
Figure 4.1: Flow chart of Hybrid Coder using 4×4 and 8×8 MRT....105
Figure 4.2: (a) Original Lena (512×512) image (b) – (d) reconstructed versions after compressed at 0.30 bpp using hybrid coder, 8×8 MRT based transform coder and 4×4 MRT based transform coder respectively....107
Figure 4.3: Performance comparison of MRT based fixed block size and hybrid techniques for Lena image 108
Figure 4.4: (a) – (d) Original Color images and (e) – (h) their reconstructed versions after compressed using hybrid transform coder 112

Figure 4.5: Examples of templates for the computation UMRT Coefficients 114

Figure 4.6: Flowchart for the Quad-Tree segmentation of images using UMRT.............................115

Figure 4.7: K level Quad-Tree 117

Figure 4.8: Quad-Tree Partitioned Lena image for different values of t_s
(a) $t_s = 100$ (b) $t_s = 150$ (c) $t_s = 200$ and (d) $t_s = 300$..118

Figure 4.9: Flow chart of MATC technique... 120

Figure 4.10: Lena (a) Original and (b) reconstructed after compressed
using MATC coder (bpp =0.31, PSNR =30.27)...122

Figure 4.11: Baboon (a) Original and (b) reconstructed after compressed
using MATC coder (bpp =0.85, PSNR = 24.84)...122

Figure 4.12: Cameraman (a) Original and (b) reconstructed after compressed using
MATC coder (bpp =0.30, PSNR = 31.03)................................... 123

Figure 4.13: (a) – (d) Original Color images and (e) – (h) their reconstructed
versions after compressed using MATC coder.. 126

Figure 5.1: Standard Vector Quantization scheme..128

Figure 5.2: Codewords in 2-D space.. 129

Figure 5.3: 8 isometric transformations of Lena image..131

Figure 5.4: Images of Brodatz album used for Codebook Generation 135

Figure 5.5: Generated Codebook 136

Figure 5.6: Encoding process of MRT based VQ 137

Figure 5.7: Decoding Process of MRT based VQ 138

Figure 5.8: (a) – (e) Original images & (f) – (j) their Vector quantized versions 141

Figure 5.9: Block diagram of Classified Vector Quantization scheme.. 144

Figure 5.10: Types of 2×2 image blocks and their respective UMRT coefficients......................... 146

Figure 5.11: (a) – (d) Original images and (e) – (h) their detected edges...................................... 148

Figure 5.12: (a) Original Lena image and (b) Edge blocks of Lena image 149
Figure 5.13: Codebook of Class 1..........150
Figure 5.14: Codebook of Class 2..........151
Figure 5.15: Codebook of Class 3..........151
Figure 5.16: Codebook of Class 4..........152
Figure 5.17: Flow chart of MATC-CVQ 154
Figure 5.18: Lena (a) Original and (b) reconstructed after compressed using MATC-CVQ coder (bpp =0.33, PSNR =32.89)..........156
Figure 5.19: Baboon (a) Original and (b) reconstructed after compressed using MATC-CVQ coder (bpp =1.22, PSNR = 27.18)..........156
Figure 5.20: Cameraman (a) Original and (b) reconstructed after compressed using MATC-CVQ coder (bpp =0.430, PSNR = 33.25)..........157
Figure 5.21: Comparison of DCT, DWT and MRT based techniques 160
Figure 5.22: (a) – (d) Original Color images and (e) – (h) their reconstructed versions after compressed using MATC - CVQ coder.......... 162

LIST OF TABLES

Table 1.1: Merits and demerits of various compression techniques....34
Table 3.1: Bpp and PSNR obtained for 8×8 MRT based transform coder for Lena (256×256) image for various threshold values 74
Table 3.2: Performance of 8×8 MRT based transform coder for various images for a fixed threshold value 77
Table 3.3: Time taken by the direct MRT and template based MRT methods to compress various images....79
Table 3.4: A comparison of 8×8 MRT and DCT based transform coders....80
Table 3.5: Performance of 8×8 MRT based transform coder for Lena (512×512) image for various threshold values....81
Table 3.6: Performance of 8×8 MRT based transform coder for various 512×512 images for a fixed threshold....82
Table 3.7: Performance of 8×8 MRT based transform coder for color images 88
Table 3.8: Performance of 4×4 MRT based transform coder for Lena image for various threshold values 94
Table 3.9: Performance of 4×4 MRT based transform coder for various images for a fixed threshold....94
Table 3.10: Performance of 4×4 MRT based transform coder for color images....96
Table 3.11: Comparison between 8×8 DCT and 4×4 MRT based methods 99
Table 3.12: Comparison between 8×8 MRT and 4×4 MRT based methods 99

Table 4.1: Performance of Hybrid Coder using 4×4 and 8×8 MRT......106

Table 4.2: Comparison of the hybrid coder with fixed block size transform coders 108

Table 4.3: Performance of hybrid coder in color images......109

Table 4.4: No. of blocks of each size in Lena image for various values of t_s......116

Table 4.5: Performance of MATC 121

Table 4.6: Comparison of MATC with fixed block size transform coders 124

Table 4.7: Comparison of MATC with DCT and DWT based coders......124

Table 4.8: Performance of MATC in color images......124

Table 5.1: Performance of MRT based VQ 139

Table 5.2: Performance of MATC-CVQ 155

Table 5.3: Comparison of MRT based compression techniques for 0.3 bpp......158

Table 5.4: Comparison of MRT based compression techniques for 0.5 bpp......158

Table 5.5: Comparison of MRT based compression techniques for 0.75 bpp......159

Table 5.6: Comparison between DCT, DWT and MRT based techniques 159

Table 5.7: Performance of MATC-CVQ in color images 160

1

Introduction to Still Image Compression

Contents

1.1 Image....2
1.2 Redundancy in Image....6
1.3 Image Compression....8
1.4. Classification of Image Compression Techniques....14
  1.4.1. Lossless vs Lossy compression....14
  1.4.2. Lossless Compression Techniques....15
  1.4.3 Lossy Compression Techniques....18
1.5 Transform coding....20
  1.5.1 Transforms used in Transform Coding....21
1.6 Image Compression Algorithms....28
  1.6.1 JPEG: DCT-Based Image Coding Standard....28
  1.6.2 Image Compression Using Wavelet Transform....30
  1.6.3 Vector Quantization....32
  1.6.4 Fractal Compression....33
1.7 Comparison of Compression techniques....34
1.8 Motivation....35
1.9 Organization of Thesis....36

1.1 Image

The Oxford dictionary defines the word image as the optical appearance of something produced in a mirror or through a lens. Image may be formed by other types of radiant energy and devices. However, optical images are most common and most important. Amount of light energy received at a point of a scene by an observer or by an image sensor varies with direction and distance of that point. This energy is recorded at corresponding points on a plane to form an image. Hence, the brightness and color recorded in an image may be represented as a function of several variables. The simplest kind of intensity image that can be thought of is a black and white image.

Digital Image

A digital image is a numeric representation of a two-dimensional image. Digital images have a finite set of digital values, represented by picture elements or pixels. They contain a fixed number of rows and columns of pixels. Pixels are the smallest individual element in an image, holding quantized values that represent the brightness of a given color at any specific point. Typically, the pixels are stored in computer memory as a two-dimensional array of small integers. These values are often stored in memory or transmitted.

Capturing Digital Image

In an ideal system, a simple pin hole camera would be used to produce an image since any object in the viewed environment is always perfectly in focus.

Unfortunately the intensity of the light available in a real system makes it impossible to use a pin hole camera. Real systems are forced to use a lens to focus the light from objects onto the image plane of the camera. When using a lens, the depth of field where the image sharply focuses is limited. The depth of field can be improved by using a stronger and smaller lens but this tends towards the limited image intensity. Sampling of the image is the most important feature, from the point of view of image compression. There are other effects caused by using lenses in cameras such as spherical and chromatic aberration but these do not affect the performance of image compressors.

Charge Coupled Device (CCD) cells are generally used to convert the image produced in the camera to an electrical signal. The final stage of the sampling is to convert the analogue charge in each CCD cell to a digital value. This is done by using a 'flash' analogue to digital converter (ADC) and the intensity of the signal is usually split into 2^8 levels (sometimes 2^{12} for high precision work).

Although all these stages are necessary to form the final digital image, they are nearly always ignored, and the digital images produced are assumed to be lossless. But the effects of the camera and digitizer should always be appreciated so that there are no major problems caused, when applying the algorithms to real systems. The result of sampling and quantization is a matrix of real numbers. An M×N digital image can be expressed as

$$\begin{bmatrix} x(0,0) & \cdots & x(0,N-1) \\ \vdots & \ddots & \vdots \\ x(M-1,0) & \cdots & x(M-1,N-1) \end{bmatrix}$$

Each element of this matrix array represents a pixel or pel [58], as shown in Figure 1.1.

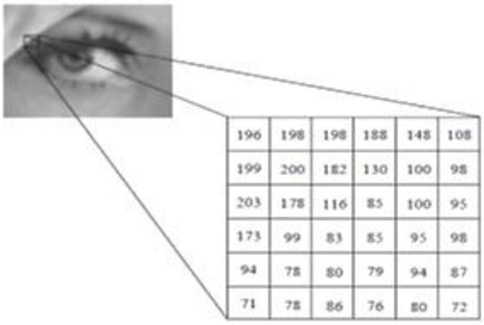


Figure 1.1: An alternate view of image data

The digital images can be classified into three categories, Black & White, Gray Scale and Color images, depending on the number of colors that each pixel in the image can represent.

Black & White (Binary) image

Black & White image uses single bit only to represent each pixel. Since a bit can only exist in two states, on or off, every pixel in a binary image must be one of two colors, usually black or white. This inability to represent intermediate shades of gray is what limits their usefulness in dealing with photographic images.

Gray Scale Image

A gray scale image is made up of pixels, each of which holds a single number corresponding to the gray level of the image at a particular location. These gray levels span the full range from black to white in a series of very fine steps, normally 256 different grays. Since the eye can barely distinguish about 200 different gray levels, this is enough to give the illusion of a step less tonal scale.

Color Image

A color image is made up of pixels, each of which holds three numbers corresponding to the red, green, and blue levels of the image at a particular location. Red, green, and blue, sometimes referred to as RGB, are the primary colors for mixing light. Any color can be created by mixing the correct amounts of red, green, and blue light. Assuming 256 levels for each primary color, any color pixel can be stored in three bytes (24 bits) of memory. This corresponds to roughly 16.7 million different possible colors. Note that for images of the same size, a gray scale version uses three times less memory than a color version.

A binary image is represented by an M×N logical matrix where pixel values are 1 (true) or 0 (false). Hence, M×N bits are to be stored. A gray scale image of M pixels height and N width is represented as a matrix of size M×N. Element values denote the pixel gray scale intensities in the range 0 to 255, with black and white represented by 0 and 255 respectively. Hence, the memory requirement to store the gray scale image is M×N×8 bits. A true color (RGB) image is represented as a three-dimensional M×N×3 matrix. Each pixel has red, green, blue components along the third dimension with values in the range 0 to 255. So, the storage requirement of the color image is M×N×24. As the resolution increases, the storage requirement also increases.

Image data requires considerable storage capacity and transmission bandwidth.

Despite rapid progress in mass-storage density, processor speeds, and digital communication system performance, the demand for data storage capacity and data transmission bandwidth continues to outstrip the capabilities of available technologies. The recent growth of data intensive multimedia-based web applications have not only sustained the need for more efficient ways to encode signals and images but have made compression of such signals central to storage and communication technology. The analysis of redundancies present in the images is to be explored to attain compression.

1.2 Redundancy in Image

A common characteristic of most images is that the neighboring pixels are correlated and therefore contain redundant information. The foremost task is to find less correlated representation of the image. Two fundamental components of compression are redundancy and irrelevancy reduction. Redundancy reduction aims at removing duplication from the signal source (image/video). Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver, usually the Human Visual System (HVS). In general, three types of redundancy can be identified: Coding Redundancy, Spatial Redundancy & Temporal Redundancy.

Coding Redundancy

A code is a system of symbols (letters, numbers, bits, etc.) used to represent a body of information or set of events. Each piece of information or event is assigned a sequence of code symbols, called a code word. The number of symbols in each code word is its length. The 8-bit codes used to represent the intensities in most of the images contain more bits than needed. The length of overall symbols can be reduced by using smaller number of bits to represent more frequent symbols while larger number of bits for less frequent symbols. Thereby coding redundancy can be reduced.

Spatial Redundancy

Spatial Redundancy is directly related to the inter-pixel correlations within an image. The pixels of most images are correlated spatially and hence, information is unnecessarily replicated in the representations of the correlated pixels. The value of any pixel can be reasonably predicted from the value of its neighbors. The information carried by individual pixel is relatively small. Much of the visual contribution from a pixel to an image is redundant. It can be predicted on the basis of the value of its neighbors [58].

Temporal Redundancy

Video data may be represented as a series of still image frames. The sequence of frames contains spatial and temporal redundancy. Video compression algorithms attempt to eliminate it or code in a smaller size. Similarities between images can be encoded by storing only the differences between frames, or by using perceptual features of human vision system.

For example, small differences in color are more difficult to perceive than changes in brightness.

Irrelevant Information

Most images contain information that is ignored by the human visual system and extraneous to the intended use of the image. It is irrelevant in the sense that it is not used. Image compression research aims at reducing the image data by eliminating all the irrelevant information, thereby reducing the number of bits needed to represent the images.

1.3 Image Compression

The term image compression refers to the process of reducing the amount of data required to represent a given image to store or transmit it in an efficient form. Ideally, an image compression technique removes redundant and/or irrelevant information, and efficiently encodes what remains. Most image compression techniques use a reversible transform to de-correlate the image data by exploiting the redundancies present in the image data. The transform can be applied either directly to the whole image or can be applied to smaller segments of the image. The techniques that use transforms to de-correlate image data prior to compression are usually referred to as transform coding.

In transform coding, the original image is usually segmented and transformed into another domain, where the transform coefficients are highly de-correlated. This de-correlation concentrates the important image information into a more compact form. The compressor then removes the redundancy in the transform coefficients and stores it into a compressed file or data stream. The decompression is the reverse process to generate the recovered image. The recovered image may have lost some information, due to the compression, and may have an error or distortion compared to the original image. A typical image compression system is shown in Figure 1.2. Some compression methods do not have the transform stage but in those cases the transform can be considered to have no effect.

The transform, used in the encoder, is rarely applied to the whole image. It usually deals with small regions or image blocks independently. This has the advantage of exploiting local similarities within the image but also leads to a ‘blocking artifact’ effect. The blocks do not have to be a fixed size or shape, but they are usually non-overlapping. Applying an overlapping blocked system duplicates data already contained in other image blocks, and hence can be wasteful.

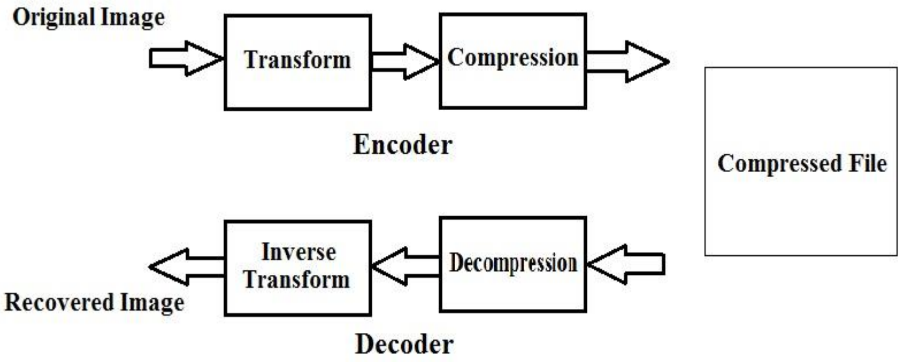


Figure 1.2: A transform based image compression system.

Image segmentation

Image segmentation is the process of dividing the image into different sized and shaped regions. If the transform is applied to fixed-size blocks in the image, it would not be very effective. So, the regions are shaped to maximize the effectiveness of the transform used. In general this means that small regions are used to approximate areas of high detail, e.g. the hair in Figure 1.3, while larger regions are used for flatter (less detailed) areas, e.g. the plain background in Figure 1.3.

Since the functions which describe a surface are usually defined for a square block, the simplest form of hierarchical image partition is a quad-tree structure. A Quad-Tree structure works by splitting square blocks into 4 equally sized sub-blocks, the blocks that are split and the depth of split is governed by the method used, not the quad-tree structure. This produces an effective method to segment the image, as shown in Figure 1.3, while a small amount of data is required to describe the quad-tree (approximately 1 - 5% of the total compression).

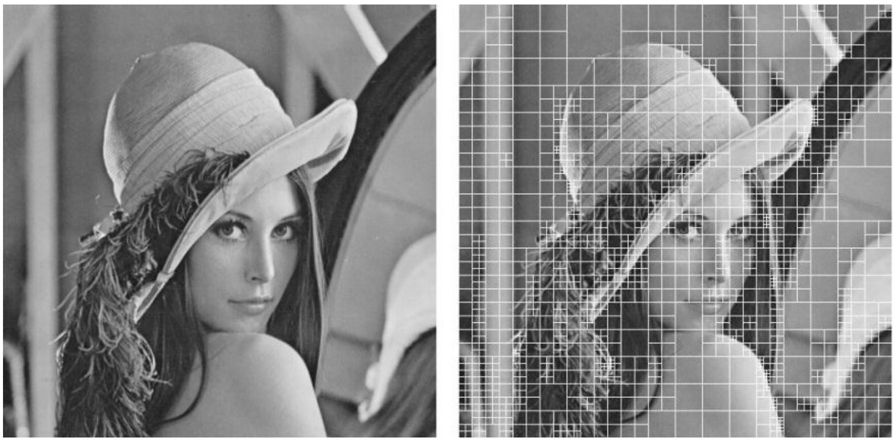


Figure 1.3: The Lena image and its Quad-Tree based segmented form.

The partition of the image can be more complex [89], such as using N sided shapes to describe the image, but as the complexity of the partition increases, so does the overhead to store the segmented structure. The more complex the partition is, the better the approximation fits to the image, but as a result there are less bits available to approximate the image. This is a common compromise reached in image compression and leads to a large variety of image segmentation based coders.

In general, segmentation coders cannot reach the image quality produced by other image compressors at similar compressions; however the research in these areas has not been fully explored.

Transform

The transform is the defining part of an image compression system. The image transform should de-correlate the image, so that the image data is in a more compact form in the new transform domain.

Transforms usually come in pairs of forward and inverse transforms. If both the forward and inverse transforms are applied without compression, then the transform is either perfectly reconstructing (lossless), or the image information is quantized and lost after the transform stage (lossy). A lossless transform does not further complicate an image compressor since it makes no decisions about which parts of the image data are useful. However a lossy transform can often produce more compression or allow the transform algorithm to run faster, both of which may be beneficial.

The transform can either be orthogonal, orthonormal or non-orthogonal. It is common to use orthogonal/orthonormal transforms in image compression; because they are efficient and the transform coefficients are highly de-correlated.

The 'Discrete Cosine Transform (DCT)' and the 'Wavelet Transform' are examples of orthonormal.

Compression

Once the image has been transformed, it is necessary to compress the de-correlated data. Compression is achieved by a combination of two methods: Quantization and entropy coding.

Quantization is the process of reducing the accuracy of the transformed coefficients, sometimes completely truncating the coefficients. This is often used before entropy coding to improve the compression. Quantization makes the coder inherently lossy. Through lossless compression or entropy coding, the decompressed data is converted into an efficient data set that takes up the minimum size possible. This is achieved using variable length coders such as the Huffman [5] or Arithmetic [101] coders.

The compression of images is usually measured in two ways: Compression ratio (The size of the original image is compared to the size of the compressed image) and Bits Per Pixel -bpp (the number of bits necessary to describe one pixel of the image, generally an average over the whole image).

For gray scale images, the relation between compression ratio and bpp is

$$bpp = \frac{Total\ bits\ needed\ to\ represent\ the\ compressed\ image}{Total\ number\ of\ pixels\ in\ the\ image} \qquad (1.1)$$

$$Compression\ ratio = \frac{8}{bpp} \tag{1.2}$$

Generally bpp is used to measure compression. The ultimate aim of any image compressor is to produce the maximum compression with minimum distortion. Although this is a relatively simple statement, it is a difficult task.

Distortion Measure

The distortion or error caused in the recovered image, by the image compression process, can be measured in several ways. The distortion measures commonly used may be classified into two broad groups, subjective and objective.

This is an area called psycho-visual image analysis, and is an area of research with immense scope. Unfortunately little progress has been made into an automated method for calculating a psycho-visual distortion measure. The subjective error measure is performed as follows. Original image and the reconstructed image are shown to a large group of examiners. Each examiner assigns grade to the reconstructed image with respect to the original image. These grades may be drawn from a subjective scale divided as, say, excellent, good, reasonable, poor and unacceptable. Finally, based on grades assigned by all the examiners, an overall grade is assigned to the reconstructed image. This grade gives an idea of the subjective error.

The standard objective distortion measures are Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR) and Signal to Noise Ratio (SNR). None of the main methods for measuring image objective distortion takes into account how good the recovered image looks to the human visual system. These distortions are calculated as in Equations (1.3) to (1.5) respectively.

$$MSE = \frac{1}{N}\sum_{i,j}(x(i,j) - x'(i,j))^2 \qquad (1.3)$$

$$PSNR = -10log_{10}\left[\frac{MSE}{Max|x(i,j)|^2}\right] \qquad (1.4)$$

$$SNR = -10log_{10}\left[\frac{\sum_{i,j}(x(i,j)-x\prime(i,j))^2}{\sum_{i,j}x(i,j)^2}\right] \qquad (1.5)$$

where N is the number of pixels in the image, $x(i, j)$ is the pixel intensity of original image at i, j and $x'(i, j)$ is the pixel intensity of compressed image at i, j .

1.4. Classification of Image Compression Techniques

The image compression techniques can be broadly classified into two: lossless and lossy compression techniques. Lossless and lossy compressions are terms that describe whether or not all original data can be recovered when the image is decompressed.

1.4.1. Lossless vs. Lossy compression

Every single bit of data that was originally in the image remains after the image is decompressed using lossless techniques. All the information is completely restored. This is generally the technique for medical image compression, where losing image data could pose problems.

On the other hand, lossy compression reduces the data in an image by permanently eliminating certain information, especially irrelevant information. When the image is decompressed, only a part of the original information is restored, although the user may not notice it. Lossy compression is generally employed for applications where a certain amount of information loss will not affect further use. The JPEG image file, commonly used for photographs and other complex still images on the Web, is an image that has lossy compression. The lossy compression techniques generally permit a trade-off between compression ratio and image quality.

The reconstructed image is numerically identical to the original image in lossless compression schemes. However lossless compression can only achieve a modest amount of compression. An image reconstructed following lossy compression contains degradation compared to the original due to the removal of irrelevant information. However, lossy schemes are capable of achieving much higher compression. But no visible loss is perceived (visually lossless) under normal viewing conditions.

1.4.2. Lossless Compression Techniques

Lossless methods yield lower compression ratios but preserve every pixel in the original image. Here, the main factor behind the reduction in size of the image is the removal of coding redundancy. Coding redundancy removal is based on the idea that in an image some colors are frequently used and others occasionally. Presence of repeated colors causes a certain amount of redundancy and can be eliminated by assigning short codes to the frequently used colors and longer codes to the infrequent ones. These methods are very common & simple and used in many of today's graphics file formats, including GIF, PCX, and BMP. Most of the graphical images contain redundancies in the form of adjacent pixels of identical values.

Run Length Coding

Run length coding is a very simple method for compression of sequential data. It takes advantage of the fact that, in many data streams, consecutive single tokens are often identical. Run length encoding checks the stream for this fact and inserts a special token each time a chain of more than two equal input tokens are found. This special input advises the decoder to insert the following token 'n' times into its output stream.

Huffman Coding

The most popular technique for removing coding redundancy is Huffman coding. When coding the symbols of an information source individually, Huffman coding yields the smallest possible number of code symbols per source symbol. The first step in this approach is to create a series of source reductions by ordering the probabilities of the symbols under consideration and combining the lowest probability symbols into a single symbol that replaces them in the next source reduction. This process is repeated until a reduced source with two symbols is reached. The second step in Huffman's procedure is to code each reduced source, starting with the smallest source and working back to the original source. The minimal length binary code for a two-symbol source is the symbols 0 & 1 [58].

Arithmetic Coding

One-to-one correspondence between source symbols and code words does not exist in Arithmetic coding. Instead, an entire sequence of source symbols (or message) is assigned a single arithmetic code word. The code word itself defines an interval of real numbers between 0 and 1. As the number of symbols in the message increases, the interval used to represent it becomes smaller and the number of information units (say, bits) required to represent the interval becomes larger. Each symbol of the message reduces the size of the interval in accordance with its probability of occurrence. It achieves the bound established by the noiseless coding because the technique does not require translating each source symbol into an integral number of code symbols [58].

Area Coding

Area coding is an enhanced form of the run length coding, reflecting the two dimensional character of images. This is a significant advancement over other lossless methods. The image can be considered as a two dimensional object. For coding an image, it does not make too much sense to interpret it as a sequential stream, as it is in fact an array of sequences. Therefore, as the two dimensions are independent and of same importance, it is obvious that a two dimensional coding scheme will be advantageous. The algorithms for area coding try to find rectangular regions with the same characteristics. These regions are coded in a descriptive form as an element with two points and a certain structure. The whole input image has to be described in this form to allow lossless decoding.

The possible performance of this coding method is limited mostly by the very high complexity of the task of finding largest areas with the same characteristics. Practical implementations use recursive algorithms for reducing the whole area to equal sized sub-rectangles until a rectangle does fulfill the criteria defined as having the same characteristic for every pixel. This type of coding can be highly effective but it bears the problem of a nonlinear method, which cannot be implemented in hardware. Therefore, the performance in terms of compression time is not competitive, although the compression ratio is competitive [58].

1.4.3 Lossy Compression Techniques

Lossy methods deliver higher compression ratios, but sacrifice the ability to reproduce the original, decompressed pixel for pixel. JPEG is the best known lossy compression standard and widely used to compress still images. It is considerably more complicated than run length coding, but it produces correspondingly higher compression ratios – even for images containing little or no redundancy. Except where every piece of information of a scan is critical – for example, scientific data – a scan must only provide enough information to meet the needs of the reproduction process and the viewer. The idea behind JPEG compression is to segregate the information in an image by level of importance, and then discard the less important information to reduce the overall quantity of data that must be stored. It does so by transforming a matrix of pixel values into a matrix of amplitude values corresponding to precise frequencies in the image. The eye doesn’t perceive all the subtle color shifts in a typical bitmapped image, so some of the detail can be discarded without affecting the overall information content.

Typical lossy image coder

A typical lossy image compression system consists of three closely connected components namely (a) Source Encoder (b) Quantizer, and (c) Entropy Encoder. Compression is accomplished by applying a linear transform to de-correlate the image data, quantize the resulting transform coefficients, and entropy code the quantized values.

Source Encoder

Over the years, a variety of linear transforms have been developed which include Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and many more, each with its own advantages and disadvantages. One of the reversible transforms can be used as the source encoder based on its merits & demerits and the requirements of the application.

Quantizer

A quantizer simply reduces the number of bits required to store the transformed coefficients by reducing the precision. Quantization is a many-to-one mapping. Hence, it is a lossy & irreversible process and is the main source of compression in an almost all encoders. Either Scalar Quantization (SQ) or Vector Quantization (VQ) can be performed (applied) on transformed coefficients. Both uniform and non-uniform quantizers can be applied depending on the application and problem at hand.

Entropy Encoder

An entropy encoder further compresses the quantized values in a lossless manner to give better overall compression. It uses a model to accurately determine the probabilities for each quantized value and produces an appropriate code based on these probabilities so that the resultant output code stream will be smaller than the input stream. The most commonly used entropy encoders are the Huffman encoder and the Arithmetic encoder. But, simple Run-Length Encoding (RLE) has proven very effective for applications that require fast execution.

1.5 Transform Coding

A general transform coding scheme involves subdividing an n×n image into smaller N×N blocks and performing a unitary transform on each sub-image. A unitary transform is a reversible linear transform whose kernel describes a set of complete, orthonormal discrete basis functions. The goal of the transform is to de-correlate the original signal, and there by the signal energy being redistributed among a small set of transform coefficients. In this way, many coefficients may be discarded after quantization and prior to encoding.

Transform coding is implemented in four stages:

- Image subdivision
- Sub-image transformation
- Coefficient quantization
- Entropy encoding.

Logical modeling for a transform coding scheme is done in two steps: a segmentation step and a transformation step. The segmentation step involves subdivision of the image in bi-dimensional vectors, possibly of different sizes. The transformation step involves application of the chosen transform (e.g. KLT, DCT, Hadamard). Quantization can be performed in several ways. Most classical approaches use zonal coding or threshold coding. Zonal coding consists of scalar quantization of the coefficients belonging to a predefined area (with a fixed bit allocation). Threshold coding involves the choice of the coefficients in each block characterized by an absolute value exceeding a predefined threshold. Another possibility, that leads to higher compression factors, is to apply a vector quantization scheme to the transformed coefficients.

The entropy coding produces the output bit stream. In most cases a classical Huffman coding or Arithmetic coding can be used successfully.

The JPEG and MPEG standards are examples of standards based on transform coding.

1.5.1 Transforms used in Transform coding

A linear transformation matrix [W], that maps the data array X to produce a diagonal covariance matrix for the transformed variable Y, where $X=[x_1,x_2,x_3,\ldots.x_N]^T$ is a vector having N pixel or data points. Then,

$$Y = [W]^T X \tag{1.6}$$

Each column vector w_i of [W] is a basis of new vector space. So alternatively each element y_i of Y is calculated as

$$y_i=w_i^T X \quad (1.7)$$

The inverse transform is calculated as

$$X=[W]Y \quad (1.8)$$

Karhunen-Loeve transform

The Karhunen-Loeve transform [15] was originally introduced as a series expansion for continuous random processes by Karhunen and Loeve. Hotelling first studied what was called a method of principal components, which is the discrete equivalent of the KL series expansion for random sequences. Consequently, the KL transform is also called the Hotelling transform or the method of principal components. For a real M×N image, the basis vectors of the KL transform are given by the orthonormalized eigenvectors of its autocorrelation matrix.

The Karhunen-Loeve transformation has the general form

$$T(u,v) = \sum_{j=0}^{N-1}\sum_{k=0}^{N-1} f(j,k)A(j,k;u,v) \quad (1.9)$$

for which the kernel *A(j,k;u,v)* satisfies the equation

$$\varphi(u,v)A(j,k;u,v) = \sum_{j'=0}^{N-1}\sum_{k'=0}^{N-1} K_f(j,k;j',k')A(j',k';u,v) \quad (1.10)$$

where $K_f(j,k;j',k')$ denotes the covariance function of the image array and $\varphi(u,v)$ is a constant for fixed (u,v). The set of functions defined by the kernel are the Eigen functions of the covariance function and $\varphi(u,v)$ represents the eigenvalues of the covariance function. It is usually not possible to express the kernel in explicit form.

Discrete Cosine transform

The discrete cosine transform (DCT) [13] expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations. In particular, the DCT is a Fourier-related transform similar to the Discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry.

The forward and inverse Cosine transform is defined as

$$T(u,v) = \frac{2}{N} C(u)C(v) \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j,k) cos\left\{\frac{\pi}{N}\left[u\left(j+\frac{1}{2}\right)\right]\right\} cos\left\{\frac{\pi}{N}\left[v\left(k+\frac{1}{2}\right)\right]\right\} \quad (1.11)$$

$$f(j,k) = \frac{2}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} C(u)C(v)T(u,v) cos\left\{\frac{\pi}{N}\left[u\left(j+\frac{1}{2}\right)\right]\right\} cos\left\{\frac{\pi}{N}\left[v\left(k+\frac{1}{2}\right)\right]\right\} \quad (1.12)$$

where $C(0) = (2)^{-1/2}$ and $C(w) = 1$ *for* $w = 1,2,\ldots,N-1$

Discrete Sine transform

The discrete sine transform (DST) [68] is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using a purely real matrix. It is equivalent to the imaginary parts of a DFT of roughly twice the length, operating on real data with odd symmetry.

The two dimensional Sine transform pair is defined as

$$T(u,v) = \frac{2}{N+1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} f(j,k) sin\left\{\frac{(j+1)(u+1)\pi}{N+1}\right\} sin\left\{\frac{(k+1)(v+1)\pi}{N+1}\right\} \quad (1.13)$$

$$f(j,k) = \frac{2}{N+1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} T(u,v) sin\left\{\frac{(j+1)(u+1)\pi}{N+1}\right\} sin\left\{\frac{(k+1)(v+1)\pi}{N+1}\right\} \quad (1.14)$$

Discrete Walsh-Hadamard transform

The Walsh–Hadamard transform [58] (also known as the Hadamard transform, Walsh transform, or Walsh–Fourier transform) is an example of a generalized class of Fourier transforms. It performs an orthogonal, symmetric, involutional, linear operation on 2^m real numbers (or complex numbers, although the Hadamard matrices themselves are purely real). The Hadamard transform can be regarded as being built out of size-2 discrete Fourier transforms (DFTs), and is in fact equivalent to a multidimensional DFT of size 2×2×…×2×2. It decomposes an arbitrary input vector into a superposition of Walsh functions.

The forward and inverse transform kernels, *g(x,y,u,v) and h(x,y,u,v),* of Walsh-Hadamard transform is as follows.

$$g(x,y,u,v) = h(x,y,u,v) = \frac{1}{N}(-1)^{\sum_{i=0}^{m-1}[b_i(x)p_i(u)+b_i(y)p_i(v)]} \qquad (1.15)$$

where $N=2^m$. The summation in the exponent of this expression is performed in modulo 2 arithmetic and $b_k(z)$ is the kth bit (from right to left) in the binary representation of *z.* The $P_i(u)$ are computed using

$$\begin{aligned} p_0(u) &= b_{m-1}(u) \\ p_1(u) &= b_{m-1}(u) + b_{m-2}(u) \\ p_2(u) &= b_{m-2}(u) + b_{m-3}(u) \qquad (1.16) \\ &\vdots \\ p_{m-1}(u) &= b_1(u) + b_0(u) \end{aligned}$$

where the sums are performed in modulo 2 arithmetic.

Discrete Wavelet transform

Wavelets are functions defined over a finite interval and having an average value of zero. The basic idea of the wavelet transform [92] is to represent any arbitrary function (t) as a superposition of a set of such wavelets or basis functions. The discrete wavelet transform of function *f(x,y)* of size M×N can be expressed as

$$W_\varphi(j_0,m,n) = \frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\varphi_{j_0,m,n}(x,y) \qquad (1.17)$$

$$W^i{}_\gamma(j,m,n) = \frac{1}{\sqrt{MN}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\gamma^i_{j,m,n}(x,y) \qquad i=\{H,V,D\} \qquad (1.18)$$

where j_0 is an arbitrary starting scale and $W_\varphi(j_0, m, n)$coefficients define an approximation of $f(x, y)$ at scale j_0. The ${W^i}_\gamma(j, m, n)$ coefficients add horizontal, vertical and diagonal details for scales for $j \geq j_0$.

$\varphi_{j,m,n}(x, y)$ and $\gamma^i_{j,m,n}(x, y)$ are the scaled and translated basis functions as follows.

$$\varphi_{j,m,n}(x, y) = 2^{j/2}\varphi\left(2^j x - m, 2^j y - n\right) \tag{1.19}$$

and

$$\gamma^i_{j,m,n}(x, y) = 2^{j/2}\gamma^i\left(2^j x - m, 2^j y - n\right), \quad i = \{H, V, D\} \tag{1.20}$$

Given W_φ and ${W^i}_\gamma$, $f(x, y)$ is obtained by the inverse discrete wavelet transform

$$f\,(x, y) = \frac{1}{\sqrt{MN}}\sum_m \sum_n W_\varphi(j_0, m, n)\varphi_{j_0,m,n}(x, y)$$

$$+ \frac{1}{\sqrt{MN}}\sum_{i=H,V,D}\sum_{j=j_0}^{\infty}\sum_m \sum_n W^i_\gamma(j, m, n)\gamma^i_{j,m,n}(x, y) \tag{1.21}$$

Mapped Real Transform

A transform named Mapped Real Transform, originally M-dimensional Real Transform (MRT), was proposed by R. C. Roy et al. [135], represents 2-D signals in terms of real additions alone rather than using complex multiplications. This transform maps the data matrix into M matrices using real additions alone.

Let the N×N data matrix be *X* and the corresponding MRT be $Y^{(p)}_{k1,k2}$ **,** then $Y^{(p)}_{k1,k2}$ can be expressed as

$$Y_{k1,k2}^{(p)} = \sum_{\forall(n1,n2)|z=p} x(n1,n2) - \sum_{\forall(n1,n2)|z=p+M} x(n1,n2) \tag{1.22}$$

Where

$$0 \le n1, n2, k1, k2 \le N-1$$

$$z = \big((n1k1 + n2k2)\big)_N \tag{1.23}$$

$$M = \frac{N}{2} \tag{1.24}$$

The matrices $Y_{k1,k2}^{(p)}$ for $p = 0, 1 \ldots M-1$ are the MRT matrices.

The above transformation can be expressed in matrix form as given below

$$[\mathrm{X}] \xrightarrow{\mathbf{A}} [\mathrm{Y}] \quad \text{where}$$

$$[\mathrm{Y}] = [[\mathrm{Y}^{(0)}], [\mathrm{Y}^{(1)}], \ldots, [\mathrm{Y}^{(M-1)}]]$$

$$\mathbf{A} = [[\mathrm{A}^{(0)}], [\mathrm{A}^{(1)}], \ldots, [\mathrm{A}^{(M-1)}]]$$

$$[\mathrm{Y}^{(p)}] = [\mathrm{A}^{(p)}] \otimes [\mathrm{X}] \quad \text{where } \otimes \text{ represents Kronecker Product}$$

$$= \begin{bmatrix} [\mathbf{A}_{0,0}^{(\mathbf{P})}][\mathbf{X}] & [\mathbf{A}_{0,1}^{(\mathbf{P})}][\mathbf{X}] & \cdots & [\mathbf{A}_{0,N-1}^{(\mathbf{P})}][\mathbf{X}] \\ [\mathbf{A}_{1,0}^{(\mathbf{P})}][\mathbf{X}] & [\mathbf{A}_{1,1}^{(\mathbf{P})}][\mathbf{X}] & \cdots & [\mathbf{A}_{1,N-1}^{(\mathbf{P})}][\mathbf{X}] \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ [\mathbf{A}_{N-1,0}^{(\mathbf{P})}][\mathbf{X}] & [\mathbf{A}_{N-1,1}^{(\mathbf{P})}][\mathbf{X}] & \cdots & [\mathbf{A}_{N-1,N-1}^{(\mathbf{P})}][\mathbf{X}] \end{bmatrix}$$

where $\left[A_{k1,k2}^{(p)}\right][X]$ is the sum of the products of corresponding elements of the two matrices and each $\left[A_{k1,k2}^{(p)}\right]$ is an N×N matrix with elements

$$A_{k1,k2,n1,n2}^{(p)} = \begin{cases} 1 & if\ ((n1k1 + n2k2))_N = p \\ -1 & if\ ((n1k1 + n2k2))_N = p + M \\ 0 & Otherwise \end{cases} \qquad (1.25)$$

where $0 \le p \le \frac{N}{2} - 1$

- **Inverse MRT (IMRT)**

The inverse MRT relation is as follows

$$x(n1, n2) = \frac{1}{N^2} \sum_{p=0}^{M-1} \left[\sum_{\forall(k1,k2)|z=p} Y_{k1,k2}^{(p)} - \sum_{\forall(k1,k2)|z=p+M} Y_{k1,k2}^{(p)} \right] \qquad (1.26)$$

$$0 \le n1, n2 \le N - 1$$

where $M = \dfrac{N}{2}$ (1.27)

1.6 Image Compression Algorithms

Most common image compression algorithms are explained below.

1.6.1 JPEG : DCT-Based Image Coding Standard

The Discrete Cosine Transform (DCT) [13] is one of the best transforms used in image compression. It is effectively the real part of the Fourier transform, offset and sampled at twice the rate so that it is calculated over the center of the pixels. The general form of 2-D DCT is shown in Equations (1.11) and (1.12):

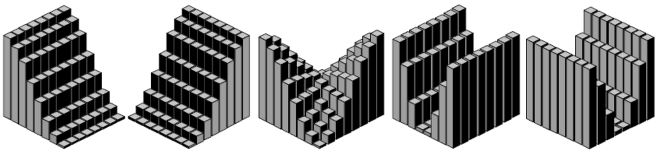


Figure 1.4: Diagram of the 5 DCT basis functions or primitives.

The DCT basis functions, shown in Figure 1.4, are very similar to the basis functions found by Principal Component Analysis (PCA) [90], [67].

JPEG is designed for compressing full-color or gray scale images of natural, real-world scenes. To exploit this method, an image is first partitioned into non overlapped 8×8 blocks. A discrete Cosine transform (DCT) [52] & [48] is applied to each block to convert the gray levels of pixels in the spatial domain into coefficients in the frequency domain. The coefficients are normalized by different scales according to the quantization table provided by the JPEG standard conducted by some psycho visual evidence. The quantized coefficients are rearranged in a zigzag scan order to be further compressed by an efficient lossless coding strategy such as Run length coding, Arithmetic coding, or Huffman coding. The decoding is simply the inverse process of encoding. So, the JPEG compression takes about the same time for both encoding and decoding. The encoding/ decoding algorithms provided by an independent JPEG group [52] are available for testing real world images. The information loss occurs only in the process of coefficient quantization. The JPEG standard defines a standard 8×8 quantization table [52], for all images which may not be appropriate. To achieve better decoding quality for various images with the same compression, by using the DCT approach, an adaptive quantization table may be used instead of using the standard quantization table.

1.6.2 Image Compression Using Wavelet Transform

Over the past several years, the wavelet transform has gained widespread acceptance in signal processing in general and in image compression research in particular. In many applications wavelet-based schemes outperform other coding schemes like the one based on DCT. Since there is no need to block the input image and its basis functions have variable length, wavelet coding schemes at higher compressions avoid blocking artifacts. Wavelet-based coding [112] is more robust under transmission and decoding errors, and also facilitates progressive transmission of images. In addition, they are better matched to the HVS characteristics.

Wavelet filters basically have two possible constraints:

- Regularity: The sum of a regular wavelet's filter coefficients must be $\sqrt{2}$, as shown in Equation (1.28).

$$\sum_i h_i = \sqrt{2} \tag{1.28}$$

where h_i is the i^{th} filter coefficient,

- Orthonormality (or perfect reconstruction): The inner product of 2 wavelet filters is 1 or 0, depending on the filter as shown below
-

$$\sum_i h_i . h_{i+2k} = \delta_{k0} \tag{1.29}$$

$$\text{where } \delta_{k0} = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases} \tag{1.30}$$

In image compression, the wavelet filters are first applied to the image in the horizontal direction and then in the vertical direction to produce images as in Figure 1.5.

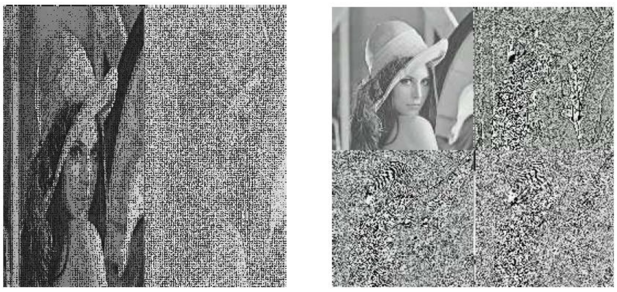


Figure 1.5: Wavelet filtering.

The four bands formed are referred as low-low (LL), low-high(LH), high-low (HL) and High-High (HH). The LL band still has image-like information and so it is possible to apply the set of wavelet filters, in the same way as applied to the original image. This process of dividing the image into sub-bands can be continued as far as desired, but for image compression, it is usually only continued to 4 or 5 levels. A typical final image is shown in Figure 1.6.

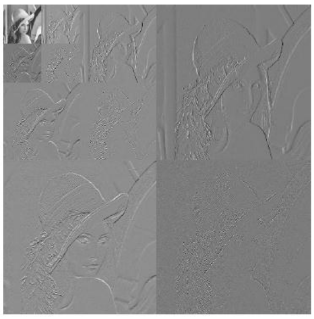


Figure 1.6: Third stage of wavelet filtering.

The actual information content of the sub-bands can be visualized by showing their average signal power, as shown in Figure 1.7. It shows that the wavelet transform is ideal for image compression, giving a high de-correlation, while also being orthogonal.

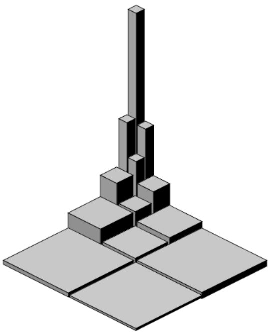
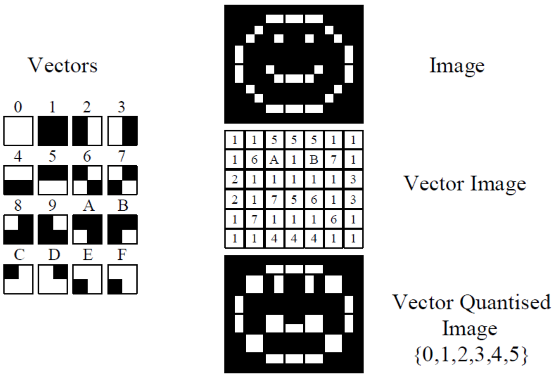


Figure 1.7: Power distribution produced by a wavelet transform.

1.6.3 Vector Quantization

A vector quantizer is composed of two operations. The first is the encoder, and the second is the decoder. The encoder takes an input vector and outputs the index of the codeword that offers the lowest distortion. The lowest distortion is found by evaluating the Euclidean distance between the input vector and each codeword in the codebook. Once the closest codeword is found, the index of that codeword is sent through a channel When the decoder receives the index of the codeword, it replaces the index with the associated codeword.

The fundamental idea of VQ [50] for image compression is to establish a codebook consisting of code vectors such that each code vector can represent a group of image blocks of size m×m, (m = 4 is always used). An image or a set of images is first partitioned into m×m non overlapping blocks which are represented as *m 2-tuple* vectors, called training vectors. The goal of codebook design is to establish a few representative vectors. The encoding procedure is to look for a closest code vector in the codebook for each 4×4 block of an image to be encoded. An example of vector quantization for a binary image is shown in Figure 1.8.



1	1	5	5	5	1	1
1	6	A	1	B	7	1
2	1	1	1	1	1	3
2	1	7	5	6	1	3
1	7	1	1	1	6	1
1	1	4	4	4	1	1

Figure 1.8: Demonstration of simple vector quantization.

In Figure 1.8 it was possible to manually quantize the vectors because of the simplistic situation, but with real images the number of vectors is huge. It is clearly not easy to find suitable vectors for a specific image.

1.6.4 Fractal Compression

The fractal compression works by exploiting self-similarity within the image. There are usually certain features within the image that are repeated at different resolutions. The fractal transform copies these features from a higher resolution onto features at a lower resolution, enhancing the image. Before this can be achieved, there needs to be some level of image approximation and this is produced by using a simple function to describe image blocks (as with the segmentation methods). It can be seen from Figure 1.9 that a larger parent block, with relative coarse features, can be shrunk onto the smaller child block improving the initial approximation.

$$f(i,j) = \sum_{k=0}^{N-1} c(k,l) \,.\, h^{-1}(i,j,k,l) + c_f \,.\, g(i,j) \qquad (1.31)$$

where c_f is the fractal coefficient and $g(i,j)$ is the parent shrunk block.

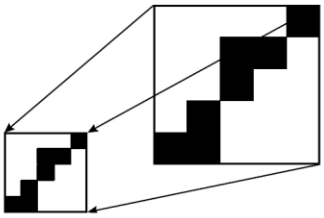


Figure 1.9: Fractal transform (without rotation considered).

There has been a lot of research in the area of fractals, since the method was proposed by Barnsley and Jacquin [37] and there has been substantial improvement by Monro [79], [80] & [88], Jacquin [75] and Oien [49].

1.7 Comparison of Compression techniques

There are certain advantages and disadvantages for various image compression techniques as shown in the Table 1.1.

Table 1.1: Merits and demerits of various compression techniques

Technique	Advantages	Disadvantages
Wavelet	High Compression Ratio, State-of-the-Art	Coefficient Quantization, Bit Allocation.
JPEG	Previous Standard	Coefficient Quantization, Bit Allocation, Blocking artifact.
VQ	Simpler Decoder	Time Consuming, Codebook generation is tedious.
Fractal	Good mathematical encoding frame	Slow Encoding.

1.8 Motivation

The benefits of image compression include less required storage space and faster transmission of images. Today the image compression is being put to work in industries such as fax transmission, satellite remote sensing, and high definition television, to name but a few.

In certain industries, the archiving of large number of images is required. A good example is the health industry, where the constant scanning and/or storage of medical images and documents take place. Image compression offers many benefits here, as information can be stored without placing large loads on system servers. Depending on the type of compression applied, images can be compressed to save storage space, or to send to multiple physicians for examination. And conveniently, these images can be decompressed when they are ready to be viewed, retaining the original high quality and details that medical imagery demands.

In the security industry, image compression can greatly increase the efficiency of recording, processing and storage. For example, in a video networking or closed-circuit television application, several images at different frame rates may be required. Time is also a consideration, as different areas may need to be recorded for various lengths of time.

Regardless of industry, image compression has virtually endless benefits wherever improved storage, viewing and transmission of images are required. Even though many image compression techniques are available today, it is sure that the industries are in the search of better compression techniques.

In literature, several transforms have been employed for the compression of images, of which Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) are popular. However, the DCT based compression scheme poses problems of slow decay rate of DCT coefficients and introduction of blocking artifacts and the DWT is not experimented with block based transform coding.

MRT [135] is an alternate representation of a signal which is derived from Discrete Fourier Transform (DFT). The computation of MRT coefficients involves only real additions, which is a good feature that can be exploited to reduce the computational overheads in real time applications involving image compression. Also, since MRT coefficients are formed from simple additions and subtractions of input data, each unique MRT coefficient signifies a particular manner in which input data are combined. Thus, the Unique MRT coefficients of image data are strength-indicators of different patterns in the image data. The MRT thus becomes a suitable tool for image analysis to determine the existence of well-defined patterns in images.

Hence, this work aimed at finding the suitability of the Mapped Real Transform (MRT) in image compression applications.

1.9 Organization of Thesis

The thesis is organized in six chapters.

Chapter 1 introduces various topics relevant to the research work on image compression. Basic concepts about digital images, capturing digital images and terminologies in digital image compression techniques, like image partition, transform, distortion measure etc. are discussed briefly. The basic concept behind the image compression techniques and their classification are also mentioned. Various image compression algorithms based on DCT, Wavelet, Vector Quantization and Fractals are illustrated and their merits and demerits are noted. An introduction to Mapped Real Transform (MRT) is also presented. The chapter ends by discussing the motivation and organization of the thesis.

A review of the past work in the field of digital image compression is presented in chapter 2. The past work in transform coding, Vector quantization and Fractal image compression are reviewed. Various transforms used in image compression, such as KLT, DCT, DST, DWHT, DWT, FrDT and DrDT are reviewed subsequently. A review on the developments in MRT is also presented. Chapter concludes by discussing the merits and demerits of various transforms used in image compression.

Chapter 3 focuses on the application of MRT in fixed block size transform coding. 8×8 and 4×4 MRT based transform coding methods are presented. A comparison between the two and also with DCT based method is incorporated. They are applied to both gray scale & color images and the performance is analyzed.

Chapter 4 concentrates on the development of MRT based variable block size transform coding techniques. The need for variable block size transform coders is discussed. A hybrid form of 8×8 & 4×4 MRT based transform coding technique is presented and its performance is evaluated and compared against fixed block size transform coders. A compact form of MRT, called Unique MRT (UMRT) is explained subsequently. An adaptive block size transform coding system is implemented using UMRT based Quad-Tree partitioning of images and its performance is studied. The performance of the adaptive block size transform coding system and the hybrid form of 8×8 & 4×4 MRT based transform coding are compared and the results are analyzed. A comparison with the DWT based approach is also presented.

The development of a new compression technique "MRT based Adaptive Transform Coder with Classified Vector Quantization (MATC-CVQ)" is described in chapter 5. Initially, the Vector Quantization in MRT domain, using isometric transformations and Scaling, is implemented and its results are analyzed for merits and demerits. UMRT based criterion for identification and classification of edges present in image blocks are then discussed. A classified vector quantization scheme is implemented using these features. Finally the new technique, "MRT based Adaptive Transform Coder with Classified Vector Quantization (MATC-CVQ)", is implemented by integrating all these techniques with adaptive block size transform coding and its performance is compared against existing techniques.

Summary and conclusions of the research work are narrated in chapter 6. Scope for future work in this area is also examined in the chapter.

2

Review of the Past work in Image Compression

Contents

2.1 Brief History……40
2.2 Transforms and Transform Coding……41
2.3 Fractal Compression……59
2.4 Vector Quantization……60
2.5 Conclusion……61

2.1 Brief History

Morse code, invented in 1838 for use in telegraphy, is an early example of data compression based on using shorter codewords for letters such as "e" and "t" that are more common in English. Modern work on data compression began in late 1940s with the development of information theory. In 1949 Claude Shannon and Robert Fano devised a systematic way to assign codewords based on probabilities of blocks. An optimal method for doing this was then found by David Huffman in 1951. Early implementations were typically done in hardware, with specific choices of codewords being made as compromises between compression and error correction. In the mid-1970s, the idea emerged of dynamically updating codewords for Huffman encoding, based on the actual data encountered. In late 1970s, with online storage of text files becoming common, software compression programs began to be developed, almost all based on adaptive Huffman coding. Abraham Lempel and Jacob Ziv suggested the basic idea of pointer-based encoding in 1977. Following the work by Terry Welch in the mid-1980s, the so-called LZW algorithm rapidly became the method of choice for most general-purpose compression systems. It was used in programs such as PKZIP, as well as in hardware devices such as modems. Digital images became more common in the late 1980s and standards for compressing them emerged. In the early 1990s, lossy compression methods also began to be widely used. Current image compression standards include: FAX CCITT 3 (run-length encoding, with codewords determined by Huffman coding from a definite distribution of run lengths), GIF (LZW), JPEG (lossy discrete cosine transform, then Huffman or arithmetic coding), BMP (run-length encoding, etc.), JPEG 2000 (lossy discrete wavelet transform). Typical compression ratios currently achieved for text are around 3:1, for line diagrams and text images around 3:1, and for photographic images around 2:1 lossless, and 20:1 lossy.

2.2 Transforms and Transform Coding

It is very true that images, often used in a variety of computer and other scientific & engineering applications, are difficult to store and transmit due to their sizes. One possible solution to overcome this problem is to use an efficient digital image compression technique. An image is usually viewed as a matrix and the processing are performed on that matrix. Almost all the digital image compression systems use various mathematical transforms for compression. The compression performance is closely related to the performance of these transforms, in terms of energy compaction and spatial frequency isolation, by exploiting inter-pixel redundancies present in the image data.

Most images have some degree of correlation between neighbouring pixels. Correlation is closely related to inter-pixel redundancy. It requires a reversible transform to remove the inter-pixel redundancy by de-correlating the image in a more compact manner [31] & [90]. Thus any image having the correlated pixels can be compressed using transform coding methods where the transform coefficients are highly de-correlated. An image transform can achieve a compression if the numbers of non-zero transform coefficients are smaller on average than the original pixels or data points. By the quantisation of the transform coefficients, lossy compression can be achieved [67]. An image transform aiming for compression should follow two properties: (a) inter-pixel redundancy reduction; and (b) isolation of spatial frequency.

In digital images the spatial frequencies are important as the low-frequency components correspond to important image features and the high-frequency ones to image details. High frequencies are a less important part of the images and can be quantised more heavily than low frequency coefficients to achieve low-bit rates. Also, the image transforms should be fast and simple, giving a choice for linear transformations [2], [8], [47], [67], [91] & [120].

The optimum transform coder which minimizes the mean square distortion of the reproduced data for a given bit rate is the KLT [2]. Other transforms investigated for image or picture compression include DCT, piecewise Fourier Transform, slant transform, linear transform with block quantization and Hadamard transform [6], [9], [10], [12] & [14]. Though the energy compaction efficiency of the KLT is very suitable for compression, it is not used in real applications due to its computational complexity [9], [17] & [19]

Transform coding has been the standard in image compression (e.g., JPEG [52], [72]), where the DCT is used because of its nice de-correlation and energy compaction properties [48].

In [2] H. Hotelling, developed a method of principal components for removing the correlation from discrete random variable.

Karhunen and Loeve, in [3], developed a continuous version of Hotelling’s transform in 1960’s.

In [15], N. Ahmed and K. R. Rao explains the Karhunen-Loeve transform (KLT) as a linearly reversible, orthogonal transformation which accomplishes the removal of redundancy by de-correlating the data block elements and is defined by Eigen values of covariance matrix.

N. S. Jayant and P. Noll explained in [27] that the KLT minimizes the geometric mean of the variance of transform coefficients thus providing largest coding gain.

From [3] & [30], it is understood that KLT is also known as Hotelling Transform or PCA (principal component analysis). The covariance matrix of an arbitrary data block is real and symmetric, so the real Eigen values and corresponding Eigen vectors can be found easily.

K. R. Rao and P. C. Yip stated, in [130], that the orthonormal Eigen vectors are found by using Gram-Schmidt orthonormal process.

From [121] & [134], it is understood that the basis vectors of KLT are calculated from the original image pixels and are therefore data-dependent. In practical applications these vectors should also be included in the compressed bit streams, making this transform less ideal for practical applications of image compression.

N. Ahmed et al., in [13], explained Discrete Cosine transform (DCT) and found that it is very important for data compression.

Through [48], K. R. Rao and P. Yip described the algorithms, advantages and applications of DCT.

K. R. Rao and P. Yip stated that DCT is a discrete time version of Fourier-cosine series and can be computed with fast-Fourier-transform-like algorithms. They found that DCT is one of the best transforms to be used in image compression. Unlike DFT, DCT is real valued and provides a better approximation of signals with fewer transform coefficients

From [13] and [48], it is understood that the DCT has as good energy compaction as KLT. Also, the advantage of DCT over KLT is that the former uses a fixed basis which is independent of data or signal. Also, DCT is a block-based transform so performance and complexity is compromised with the block size.

In [52], G. K. Wallace explained JPEG standard and the JPEG algorithm. It is designed for compressing full-color or gray scale images of natural, real-world scenes. To exploit this method, an image is first partitioned into non overlapped 8×8 blocks.

The encoding/ decoding algorithms provided in [52] are available for testing real world images. G. K. Wallace gave a standard 8×8 quantization table also in it.

In [72], W. B. Pennebaker and J. L. Mitchell explained the basic components of the JPEG standard, the DCT transform, scalar quantization, zig-zag scan, and Huffman coding.

The DCT basis functions are very similar to the basis functions found by Principal Component Analysis (PCA). In [67] & [90] PCA is explained.

The DCT has been used as the standard for still image compression in the form of JPEG [44] for about 20 years.

In [54], A. Hung and T. Meng proposed an improvement in the JPEG algorithm by designing Optimal quantizer step sizes for transform coders.

S. Wu and A. Gersho in [74] reported Rate-constrained picture adaptive quantization for JPEG baseline coders.

The baseline JPEG coding results are far from the best that JPEG offers. In [54] and [74] an optimal quantization matrix (Q- matrix) design is explained for obtaining much better performance with JPEG.

In [87], K. Ramchandran and M. Vetterli suggested optimal fast thresholding for obtaining much better performance with JPEG.

M. Crouse and K. Ramchandran proposed an improvement by joint optimization of thresholding and quantizer in [110].

Since the input image needs to be “blocked” in JPEG, correlation across the block boundaries is not eliminated. This causes noticeable and annoying “blocking artifacts” particularly at low bit rates.

In [69], H. S. Malavar, proposed Lapped Orthogonal Transforms (LOT) based attempts to solve the problem of “blocking artifacts” by using smoothly overlapping blocks. Although blocking effects are reduced in LOT compressed images, increased computational complexity of such algorithms do not justify wide replacement of DCT by LOT.

[68] states that the Discrete Sine transform (DST) is a complementary transform of DCT. DCT is an approximation of KLT for large correlation coefficients whereas DST performs close to optimum KLT in terms of energy compaction for small correlation coefficients. DST is used as low-rate image & audio coding and in compression applications.

The Discrete Wlash-Hadamard transform (DWHT) is the simplest transform implemented for any application and is a rearrangement of discrete Hadamard transform matrix.

From [134], it is clear that the amount of energy compaction efficiency of DWHT is poorer than that of DCT or KLT, so it does not have a potential to use for data compression.

H. Samet, in [26], explained the Quad-Tree and Related Data Structures.

In [35], J. Vaisey and A. Gersho, reported a variable block-size image coding concept.

P. Strobach, in [41], reported image coding based on Quad-Tree structured recursive least-squares approximation.

In [64], J. Vaisey and A. Gersho, also put forward image compression with variable block size segmentation..

E. Shusterman and M. Feder developed improved Quad-Tree decomposition algorithms for image compression in [82].

G. J. Sullivan and R. L. Baker, in [85], suggested efficient quadtree coding of images and video.

In [102], S. H. Yang and S. S. Yang introduced a new classified vector quantization for image coding with quadtree segmentation.

A. A. Kassim et al., in [152], reported hierarchical segmentation based image coding using hybrid quad-binary trees.

Recent years have witnessed explosive growth in research activities involving wavelet image coding.

In [16], A. Croiser et al. stated that Wavelet transforms are based on sub-sampling high and low pass filters (Quadature Mirror Filters (QMF)). These filters are matched in such a way that they split the data into high and low pass bands without losing any information.

S. G. Mallat, in [42], reported a theory of multiresolution signal decomposition and the wavelet representation and stated that the wavelet transform does not require blocking of signal or data points before transformation, resulting in removal of blocking artifacts even at very low bit rates. Also, wavelet-based sub-band coding is robust under decoding error and has a good compatibility with human visual system.

M. Antonini et al. explained Image coding using wavelet transform in [59].

In [92], Y. T. Chan gives the basics of Wavelets and states that all the linear orthogonal transformations, i.e. KLT, DST and DCT, are blocked transformations which remove the correlation among the pixels or data points inside the block. These transforms do not take care of correlation across the block boundaries. He reported that Wavelets are functions defined over a finite interval and having an average value of zero. The basic idea of the wavelet transform is to represent any arbitrary function (t) as a superposition of a set of such wavelets or basis functions. These basis functions or baby wavelets are obtained from a single prototype wavelet called the mother wavelet, by dilations or contractions (scaling) and translations (shifts).

Over the past several years, the wavelet transform has gained widespread acceptance in signal processing in general and in image compression research in particular. It is more robust under transmission & decoding errors, and also facilitates progressive transmission of images. In many applications wavelet-based schemes outperform other coding schemes like the one based on DCT. Since there is no need to block the input image and its basis functions have variable length, wavelet coding schemes even at higher compression ratios avoid blocking artifacts. In addition, they are better matched to the HVS characteristics.

Wavelet filters have been designed for a wide range of applications and many different sets of filters have been proposed for different applications.

In [39] & [60], I. Daubechies and in [61], A. Cohen suggested that in the case of image compression, a common choice is constituted by the family of Daubechies orthonormal filters or the bi-orthogonal filters, which have the advantage of linear phase.

DeVore et al., in [51] & [63], suggested effective methods to compress the wavelet transform coefficients. Originally the wavelet sub-bands were linearly quantized in some way to produce compression and this led to Vector Quantization being used to quantize the sub-bands [99].

In [70], J. Forment and S. Mallat reported second generation compact image coding with Wavelets. They introduced a compact image coding algorithm that separates the edge from the texture information. It allowed adapting the coding precision to the properties of the human visual perception.

From [94], [104] & [138], it is understood that there are several ways to decompose a signal into various sub-bands using the wavelet transform, such as octave, adaptive and packet decompositions. The octave decomposition is the most used decomposition technique, which non-uniformly splits the bands, rendering the lower frequency part narrower and narrower while leaving out any further decomposition of higher frequency coefficients.

In [77], J. M. Shapiro reported an embedded image coding using zero-trees of wavelet coefficients (EZW). The reported EZW algorithm was based on four key concepts: (1) a discrete wavelet transform or hierarchical sub band decomposition, (2) prediction of the absence of significant information across scales by exploiting the self-similarity inherent in images, (3) entropy-coded successive-approximation quantization, and (4) universal lossless data compression which is achieved via adaptive arithmetic coding.

A. Said and W. A. Pearlman proposed, in [105], a new fast and efficient image coding based on set partitioning in hierarchical tree (SPIHT), which provides even better performance than EZW. The image coding results were either comparable to or surpass previous results obtained through much more sophisticated and computationally complex methods. In addition, the coding and decoding procedures were extremely fast.

M. J. Tsai et.al. reported a stack-run image coding, in [106]. This works by raster scanning within sub-bands, and therefore involves much lower addressing complexity than other algorithms such as zero-tree coding that require the creation and maintenance of lists of dependencies across different decomposition levels. Despite its simplicity, the proposed algorithm was competitive with the best enhancements of zero-tree coding. In addition, it performs comparably with adaptive sub-band splitting approaches that involve much higher implementation complexity

In [109], Z. Xiong et al. suggested a DCT-based Embedded Image Coding technique. If the SPIHT quantizer is used to quantize the DCT coefficients, a DCT-based embedded image coder will get generated. It observes that an 8×8 DCT image representation can be thought of as a 64 –sub-band decomposition, and that each 8×8 DCT block can be treated as a depth-three tree of coefficients. Here, an 8×8 DCT block is treated as a depth-three tree of coefficients.

M. Crouse and K. Ramchandran stated, in [110], that for the standard 512×512 Lena image at a moderate bit rate, the optimal JPEG coder gives even better peak signal-to- noise ratio (PSNR) than the original EZW coder proposed by Shapiro in [77].

R. de Queiroz et al., in [111], suggested a Wavelet-Based JPEG-Like Image coding technique. When the wavelet transform is coupled with the baseline JPEG quantizer, the resulting coder becomes the one described in [111], where only the DCT in baseline JPEG is replaced by a three-level wavelet transform. The wavelet coefficients are rearranged into wavelet blocks and scanned into vectors before scalar quantization and Huffman coding. The Author reported a gain of about 1 dB for Lena with the wavelet-based JPEG-like coder over the baseline JPEG. Here, the three-level wavelet transform coefficients are rearranged into blocks and scanned into vectors before scalar quantization and Huffman coding.

R. Buccigrossi and E. P. Simoncelli, proposed, in [112], an embedded predictive wavelet image coder (EPWIC), which is a more powerful coder that uses a within-subband and inter-subband conditional statistical model for natural images. Specifically, it estimates the amplitude of wavelet coefficients based on the amplitudes of neighboring coefficients within the subband, and the amplitudes of coefficients in nearby spatial locations in other subbands.

Z. Xiong et al., in [113], introduced space-frequency quantization for wavelet image coding. It concentrated on how spatial quantization modes and standard scalar quantization can be applied in a jointly optimal fashion in an image coder.

In [115], M. Boliek et al. proposed next generation image compression and manipulations using Compression with reversible embedded wavelets (CREW).

R. C. Colderbank et al., in [122], stated that invertible wavelet transforms that map integers to integers have important applications in lossless coding. They presented two approaches to build integer to integer wavelet transforms. The first approach was to adapt the precoder, which is used in information transmission. They combined it with expansion factors for the high and low pass band in subband filtering. The second approach is built upon the idea of factoring wavelet transforms into so called lifting steps. That allowed the construction of an integer version of every wavelet transform. Finally, they used these approaches in a lossless image coder

In [124], Z. Xiong et al. has done a comparative study of DCT- and wavelet-based coding for still images. Based on comparison of performances, it can be illustrated that the main factors in image coding are the quantizer and entropy coder rather than the difference between the wavelet transform and the DCT.

A new image compression algorithm is proposed, in [125], by D. Taubman, based on independent embedded block coding with optimized truncation of the embedded bit-streams (EBCOT). The algorithm exhibits state-of-the-art compression performance while producing a bit-stream with a rich set of features, including resolution and SNR scalability together with a “random access” property. The algorithm had modest complexity and was suitable for applications involving remote browsing of large compressed images.

G. H. Golub and C. Reinsels stated, in [7], that singular-value decomposition-based transformation has an optimal energy-compaction property, making it the most appropriate for compression in spite of computation complexity.

C. S. McGoldrick, et al., in [95], found that in the case of singular-value decomposition (SVD), the singular values are image-dependent and must therefore be coded with the associated singular vectors as side information. McGoldrick et al. calculated singular values as well as singular vectors and the latter were coded by variable-rate vector quantizer. The optimal energy compaction property was exploited and utilized by McGoldrick et al.

JPEG image coder based on DCT was superior to SVD-based method. Yang and Lu in [96] also used SVD in conjunction with vector quantization giving a superior method by reducing the computational complexity to that of DCT based method. However, with the invention of fast DCT algorithm by W. Chen et al. in [20], the technique proposed in [96] was not a preferred technique.

Waldemar and Ramstad in [116] & [117] proposed hybrid KLT-SVD image compression using transform adaptation technique exploiting the local variation of images. This hybrid method was better than KLT based methods in terms of energy compaction but could not be sustained due to a large number of vectors to be coded.

S. O. Aase et al., in [123], gave a critique on SVD-based image compression and pointed out the major drawback of using lossless SVD transform for image compression. According to them, the singular vectors along with the singular values are stored for lossless reconstruction, which requires more space for image.

In 2000, J. Chen [127] used rank approximation method for SVD-based lossy image compression. In rank approximation for SVD-based image compression an image of size N×N was transformed by SVD to obtain matrices $U_{N\times N}$, $S_{N\times N}$ and $V_{N\times N}$, where S is a diagonal N×N matrix whose number of non-zero diagonal elements determines the rank "k" of the original matrix where $k \leq N$. In this method a smaller rank is used to approximate the original image. The total storage space required to restore the original approximated image is $2Nk + k$, where $k \leq N$. So, by rank approximation method there is a restriction on reconstructed image quality for compressed image.

B. Arnold and A. McInnes in [128] reported block-based adaptive rank approximation method similar to most of the popular image compression methods, to exploit the uneven complexity and correlation of image.

The work reported by B. Arnold and A. McInnes was based on singular-value distribution of different sub-blocks in which higher ranks were used for complex sub-bands. Also, for the same storage space, smaller block sizes of sub-blocks produced better results. Arnold and McInnes further reduced rank of the blocks by rank-one update, in which the respective mean was subtracted from all the elements of the blocks and then SVD and adaptive rank approximation was used.

Dapena and Ahalt, in [132], and Wongsawat et al., in [137], reported hybrid DCT-SVD and modified hybrid DCT-SVD image coding algorithms in 2002 and 2004 respectively. Both methods were based on an adaptive selection of block transforms to be used on the basis of complexity and correlation of different blocks. For high correlation, SVD was used while for the rest, DCT was used.

In 2003, a hybrid DWT-SVD-based image coding, which is also a block-based method, was reported by H. Ochoa and K. R. Rao [133] & [144], who used a criterion of threshold standard deviation for all blocks of Y component to determine whether DWT or SVD has to be used for any particular block. If standard deviation is high, rank-one update is used for that block, otherwise DWT method is used.

In [140] and [141], H. Ochoa and K. R. Rao further extended this method for color image compression also.

In 2007, Ranade et al., in [148], proposed a modified SVD image compression based on SSVD (shuffled SVD). In this work the block-based shuffling operator was used to get sub-blocks. The performance of SSVD was shown to be better than SVD in terms of space for the same quality but involved more complex operations. Also, the performance was not even near to DCT-based coding systems.

As per [1], N. Wiener introduced a concept of fractional transforms in 1929.

Subsequently, in [22], V. Namias introduced the fractional Fourier transform (FrFT) in 1980.

L. B. Almeida, in [84], explored the time-frequency localization property and provided a possible application of FrFT in image compression.

In the case of fractional transform, one extra free parameter is also there besides time and frequency.

In [126], Gerek and Erden proposed a discrete fractional cosine transform in 2000 by taking the advantage of the relation between DCT and DFT, which was similar to the method of finding DFrFT by Ozaktas et al. in [107].

K. Singh, through [139], explored the possible application of DFrCT and DFrFT in image compression. The compression performance of fractional transforms depends on the value of free parameter. However, any direct relation between free parameter and compression performance has not been reported. Hence, it is impractical to optimize the free parameter, which results in a recursive and a very slow process for image compression.

All the transforms, as discussed above, are 2-D transforms implemented by using 1-D separable architectures and are not suitable to preserve the image features with arbitrary orientation that is neither vertical nor horizontal [86]. In these cases, they result in large-magnitude high-frequency coefficients. At low bit rates, the quantization noise from these coefficients is clearly visible, in particular causing annoying Gibbs artifacts at image edges with arbitrary directions.

Some work on wavelet and sub-band transform to incorporate directional information into transforms has been reported.

The lifting structure developed by W. Sweldens in [86] provides a good way to incorporate directional information into the wavelet transform.

[86], [136] & [146] describes various techniques to incorporate directional information in to transforms based on lifting.

Zeng and Fu [143] are the first authors to propose how to incorporate directional information into DCT. Their directional DCT is motivated by SA-DCT (shape-adaptive DCT).

Hao et al. in [147] proposed a lifting-based directional DCT-like transform for image coding and used it for image compression. The main problem with directional transforms is the selection of optimum direction.

In [119], R. Gopikakumari modified the computation of N×N DFT in terms of 2×2 DFT so as to minimize complex multiplications and presented a visual representation. A parallel distributed architecture, based on the neural network concept, was developed involving real additions only in the first three layers and complex multiplication in the fourth layer.

In the DFT computation, the data will be real values whereas DFT coefficients will be complex. In [119], 2-D DFT representation was modified in terms of real additions, which originally requires N/2 complex multiplication in the computation of each of the N^2 DFT coefficients.

Rajesh Cherian Roy et al. in [135] & [149] developed a new transform called, M-

Dimensional Real Transform (MRT) based on the modified DFT computations reported in [119] by eliminating the complex multiplications in it.

However, as per [149], the complex multiplications can be avoided if the signal is represented in terms of the signal components which would otherwise be multiplied with the exponential term in the DFT representation developed in [119].

Hence in [135] & [149], a new transform, M-Dimensional Real Transform (MRT) was developed to represent 2-D signals in terms of real additions alone rather than using complex multiplications.

Bhadran V, in [150], reported a visual representation of MRT coefficients in terms of 2×2 data and modified the hardware implementation reported in [119] to make it more efficient. Also, developed an algorithm to compute the unique MRT coefficients and place in UMRT matrix.

K. Meenakshy, in [153], reported an application of MRT in the development and implementation of a CAD system to predict the fragmentation of renal stones based on texture analysis of CT Images. Also, M-dimensional real transform is renamed as Mapped Real Transform.

Preetha Basu et al., in [154], modified the UMRT algorithm reported in [150] for N, a power of 2.

2.3 Fractal Compression

There has been a lot of research in the area of fractals, since the method was proposed by Barnsley and Jacquin in [37] and there has been substantial improvement by Oien [49], Jacquin [75] and Monro in [57], [79], [80] & [88].

As per [66], Fractal image coding was introduced in late 1980s and early 1990s.

In [76], M.F. Barnsley and L.P. Hurd reported that initially it is used for encoding/ decoding images in Encarta/Encyclopedia.

As per [65] & [76], the most attractive property of fractal encoding is the resolution-independent decoding. One can enlarge an image by decoding an encoded image of smaller size so that the compression ratio may increase exponentially.

In [83], Y. Fisher reported that fractal coding is based on the Collage theorem and the fixed point theorem for a local iterated function system consisting of a set of contraction affine transformations.

Y. Fisher suggested two serious problems also that can occur in fractal encoding are the computational demands and the existence problem of best range-domain matches.

The fractal encoding can be used with image segmentation methods to improve the underlying approximating function, and this has the advantage of being able to use the fractal at different resolutions where necessary. Although fractal methods have had a reasonable commercial success in the form of Iterated System's products, there is no comparison of how effective they are compared to plain image segmentation.

2.4 Vector Quantization

The most important work in VQ Compression is to design a versatile codebook. In [21], Y. Linde, et al. explained a codebook generation algorithm.

N. M. Nasrabadi and R.A. King gave a good review of VQ, in [40].

In [50], A. Gersho, and R.M. Gray explained the fundamental idea of VQ for image compression. Its basic idea is to establish a codebook consisting of code vectors such that each code vector can represent a group of images.

In [118], Y. W. Chen's comparison indicates that a codebook developed based on LBG [21] algorithm generally has higher PSNR values over some other schemes despite its slow off-line training.

It is clearly not easy to find suitable vectors for a specific image, but this problem can be solved by Lloyd [24] and Max [4] quantization. Lloyd-Max quantization allows an optimal set of vectors to be found for any set of test images. This gives a library of blocks that are easy to fit to the desired image. The method for finding the best library and how to apply this library has been developed in several works [40], [46], [50] & [55].

In [99], P.C. Cosman proved that the Vector quantization can be applied directly to an image, but more favourable results can be obtained by applying it to a transformed image and wavelets have been shown to be effective. Although vector quantization could be confused as part of a wavelet transform, it stands alone as an image compression method.

2.5 Conclusion

On the basis of the above review, it can be concluded that any image transform applied for image compression should have minimum entropy, maximum coding gain, minimum quantization error, minimum truncation error, and moderate block size. Although the KLT shows highest energy compaction, it is a very complex transform and usually takes unfeasible time delay during the transformation. DCT shows as good performance as KLT though the advantage of DCT over KLT is that the former employs fixed basis which is independent of data or signal. Also, DCT is a block-based transform so performance and complexity is compromised with the block size.

Disadvantage of DCT is its blocking effect for low bit rate applications. DST is also a block-based transform and can be used only for the image or data which have very small correlation. DWHT is very simple to implement but has a very poor performance in terms of energy compaction efficiency. The compaction efficiency of DWT is not very good compared to that of DCT but it can provide a satisfactory performance for the entire range of bit rates.

The blocking effect as shown in DCT is removed in the case of DWT as it is a global transform and not the block-based transform. The compression performance of fractional transforms depends on the value of free parameter and it is impractical to optimize the free parameter due to a recursive and very slow process, which is not favorable for compression. Directional discrete transforms are used at low bit rates when the quantization noise from the transform coefficients is clearly visible, in particular causing Gibbs artifacts at the image edges with arbitrary directions. The optimization of direction makes it unsuitable for compression. SVD transform has an optimum energy compaction property but needs the requirement of more storage space for lossless compression and has a high level of complexity if it is used globally.

MRT [135] seems to be computationally simple. From the literature, it is evident that the unique coefficients of MRT are very powerful and contains all of the information in the input data. With its strong properties, MRT can become a good transform for image analysis and compression. Researchers had not put much effort in developing image compression techniques based on MRT. This work explores the effectiveness of MRT based image compression techniques.

3

MRT based Fixed Block size Transform Coding

Contents

3.1 Transform Coding............64
  3.1.1 Transform Selection............65
  3.1.2 Sub-image size selection............66
  3.1.3 Bit Allocation............67
3.2 Transform coding using 8×8 MRT............68
  3.2.1 Unique MRT Coefficients............68
    3.2.1.1 Significance of Unique MRT coefficients............69
  3.2.2 Encoding............70
  3.2.3 Decoding............72
  3.2.4 Results and Analysis............74
    3.2.4.1 Comparison between DCT and MRT based image compression techniques............79
    3.2.4.2 Application in images with higher resolutions............81
  3.2.5 Application in Color images............87
3.3 Transform coder based on 4×4 MRT............88
  3.3.1 Unique Coefficients in 4×4 MRT matrices............91
  3.3.2 Encoding............92
  3.3.3 Decoding............93
  3.3.4 Results and Analysis............93
  3.3.5 Application in Color images............95

3.1 Transform Coding

Transform coding techniques use a reversible, linear mathematical transform to map the pixel values onto a set of coefficients, which are then quantized and encoded. The key factor behind the success of transform-based coding schemes is due to the fact that many of the resulting coefficients, for most natural images, have small magnitudes and can be quantized (or discarded altogether) without causing significant distortion in the decoded image. Different mathematical transforms, such as Discrete Fourier (DFT), Walsh-Hadamard (WHT), and Karhunen-Loeve (KLT), have been considered for the task. For compression purposes, the higher the capability of compressing information in fewer coefficients, the better the transform; for that reason, the Discrete Cosine Transform (DCT) had become the most widely used in transform coding techniques.

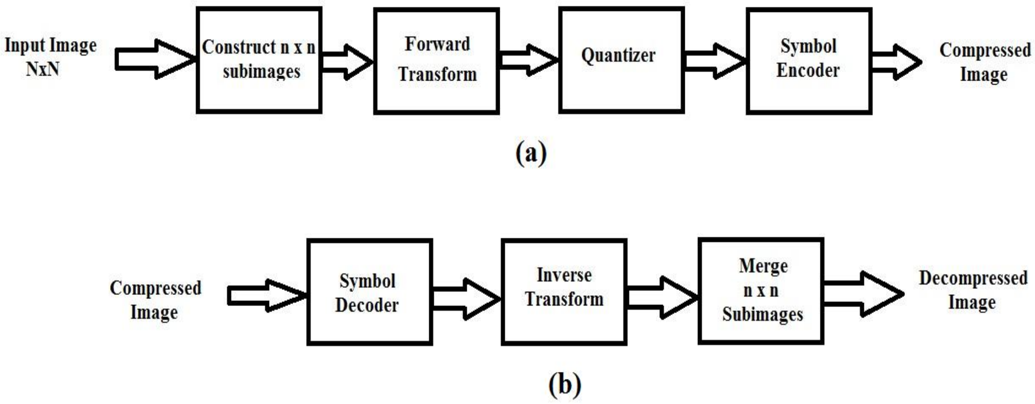


Figure 3.1: A Transform Coding System, (a) Encoder and (b) Decoder

A typical transform coding system is shown in Figure 3.1. Transform coding algorithms usually start by partitioning the original image into sub-images or blocks of small size, usually 8×8. The transform coefficients are computed for each block, effectively converting the original 8×8 array of pixel values into an array of coefficients within which certain coefficients usually contain most of the information needed to quantize and encode (and eventually perform the reverse process at the decoder's side) the image with little perceptual distortion. The resulting coefficients are then quantized and the output of the quantizer is used by a (combination of) symbol encoding technique(s) to produce an output bit stream representing the encoded image. The reverse process takes place at the decoder's side, with the obvious difference that the 'de-quantization' stage will only generate an approximated version of the original coefficient values; in other words, whatever loss was introduced by the quantization process in the encoder stage is not reversible.

3.1.1 Transform Selection

Transform coding systems based on a variety of discrete transforms have been studied extensively in section 1.5. The choice of a particular transform in a given application depends on the amount of reconstruction error that can be tolerated and the computational resources available. Compression is achieved during the quantization of the transformed coefficients and not during the transformation step.

Consider an image $x(n1, n2)$ of size N×N, whose forward discrete transform, $Y(k1, k2)$ can be expressed in terms of the general relation

$$Y(k1, k2) = \sum_{n1=0}^{N-1} \sum_{n2=0}^{N-1} x(n1, n2) g(n1, n2, k1, k2) \qquad (3.1)$$

for $0 \leq k1, k2 \leq N-1$.

Given $Y(k1, k2)$, $x(n1, n2)$ can similarly be obtained using the general inverse transform relation

$$x(n1, n2) = \sum_{k1=0}^{N-1} \sum_{k2=0}^{N-1} Y(k1, k2) h(n1, n2, k1, k2) \qquad (3.2)$$

for $0 \leq n1, n2 \leq N-1$.

In these equations, *g(n1,n2,k1,k2)* and *h(n1,n2,k1,k2)* are called forward and inverse transformation kernels or basis functions or basis images. *Y(k1,k2),* for *0≤k1,k2≤N-1* are called transform coefficients. The forward and inverse transform kernels determine the type of transform that is computed and the overall computational complexity & reconstruction error of the transform coding system in which they are employed.

3.1.2 Sub-image size selection

The sub-image size is a significant factor affecting transform coding error and computational complexity. In most applications, images are subdivided so that the correlation (redundancy) between adjacent sub-images is reduced to some acceptable level. Usually the sub-image size, ‘N’, is an integer power of 2. The latter condition simplifies the computation of the sub-image transforms. In general, both the level of compression and computational complexity increase as the sub-image size increases. The most popular sub-image sizes are 4×4, 8×8, and 16×16.

3.1.3 Bit Allocation

The reconstruction error associated with the transform coding is a function of the number and relative importance of the transform coefficients that are discarded as well as the precision that is used to represent the retained coefficients. In most of the transform coding systems, the retained coefficients are selected on the basis of maximum variance, zonal coding, or on the basis of maximum magnitude, called threshold coding.

Zonal coding is based on the information theory concept of viewing information as uncertainty. Therefore the transform coefficients of maximum variance carry the most image information and should be retained in the coding process. Zonal coding is usually implemented by using a fixed mask for all sub-images. Threshold coding, however, is inherently adaptive in the sense that the location of the transform coefficients retained for each sub-image vary from one sub-image to another. In fact, threshold coding is the adaptive transform coding approach most often used in practice because of its computational simplicity.

There are three basic ways to threshold a transformed sub-image.

- A single global threshold applied to all the sub-images
- A different threshold used for each sub-image; or
- The threshold is varied as a function of the location of each coefficient within the sub-image.

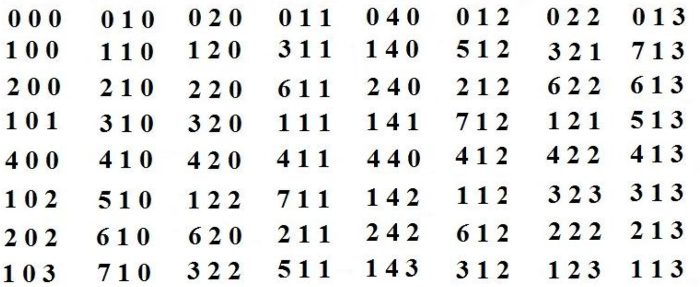
Bit allocation is the overall process of truncating, quantizing and coding the coefficients of a transformed sub-image.

3.2 Transform coding using 8×8 MRT

3.2.1 Unique MRT Coefficients

As explained in [135], the raw MRT of a signal has a considerable amount of redundancy. For a two-dimensional signal of size N×N, the total number of MRT coefficients is $N^3/2$, from N/2 MRT matrices of size N×N. However, of these $N^3/2$ coefficients, only N^2 coefficients are unique. Hence, in order to save memory space and to make it simpler, it is necessary to eliminate the redundancy in the MRT matrices and retain only the unique coefficients so that a compact MRT representation is obtained.

MRT coefficients are unique for certain values of k1 and k2: for k2 = 0 and all powers of 2 up to and including N/2 and all or certain values for k1 from among 0,1,2,3,…,N- 1 depending on the value of p. Thus it is sufficient to compute the MRT coefficients for these specific values of k1, k2 and p. The values of k1, k2 and p which yield unique MRT coefficients for an image block of size 8×8 are shown in Figure 3.2.



0 0 0	0 1 0	0 2 0	0 1 1	0 4 0	0 1 2	0 2 2	0 1 3
1 0 0	1 1 0	1 2 0	3 1 1	1 4 0	5 1 2	3 2 1	7 1 3
2 0 0	2 1 0	2 2 0	6 1 1	2 4 0	2 1 2	6 2 2	6 1 3
1 0 1	3 1 0	3 2 0	1 1 1	1 4 1	7 1 2	1 2 1	5 1 3
4 0 0	4 1 0	4 2 0	4 1 1	4 4 0	4 1 2	4 2 2	4 1 3
1 0 2	5 1 0	1 2 2	7 1 1	1 4 2	1 1 2	3 2 3	3 1 3
2 0 2	6 1 0	6 2 0	2 1 1	2 4 2	6 1 2	2 2 2	2 1 3
1 0 3	7 1 0	3 2 2	5 1 1	1 4 3	3 1 2	1 2 3	1 1 3

Figure 3.2: Indices k1 k2 p of unique coefficients in 8×8 MRT matrix.

In Figure 3.2, ’000’ means that the MRT coefficient at that position is formed from values of k1 = 0, k2 = 0, p = 0, and similarly for the other entries. [149].

3.2.1.1 Significance of Unique MRT coefficients

Since MRT coefficients are formed from simple addition and subtraction of image data, each unique MRT coefficient signifies a particular manner of combination of image data and thus a unique pattern in the image domain. Thus, the N^2 Unique MRT coefficients of an N×N image block are in fact strength-indicators of N^2 different patterns in the image block. The MRT thus becomes a suitable tool for image analysis to determine the existence of N^2 well defined patterns in an image. Figure 3.3 shows the pattern associated with the MRT coefficient for $\boldsymbol{Y}_{\boldsymbol{0},\boldsymbol{1}}^{(\boldsymbol{0})}$. The ’+’ sign indicates that the image data at that position is an addend in the formation of this MRT coefficient, and ’-’ denotes a subtrahend. ’0’ indicates positions of image data that do not contribute to the MRT coefficient. In image blocks with strong vertical structure, this MRT coefficient will have a significant value.

The N^2 patterns can be considered to be the basis images of the MRT. An important consideration here is that the number of image data elements which contribute to MRT coefficient formation is not equal for all MRT coefficients. This number can range from N^2 downward to binary fractions of N^2. This factor may be used to perform the normalization of the MRT coefficient obtained from simple additions and subtractions of the image data. If this division is not performed in the forward transform, then it needs to be done while performing the inverse MRT [149].

$$\begin{matrix} + & 0 & 0 & 0 & - & 0 & 0 & 0 \\ + & 0 & 0 & 0 & - & 0 & 0 & 0 \\ + & 0 & 0 & 0 & - & 0 & 0 & 0 \\ + & 0 & 0 & 0 & - & 0 & 0 & 0 \\ + & 0 & 0 & 0 & - & 0 & 0 & 0 \\ + & 0 & 0 & 0 & - & 0 & 0 & 0 \\ + & 0 & 0 & 0 & - & 0 & 0 & 0 \\ + & 0 & 0 & 0 & - & 0 & 0 & 0 \end{matrix}$$

Figure 3.3: Template for the computation of MRT coefficient $^{(0)}$

3.2.2 Encoding

The overall encoding process is explained in Figure 3.4. The major steps in the process are explained below

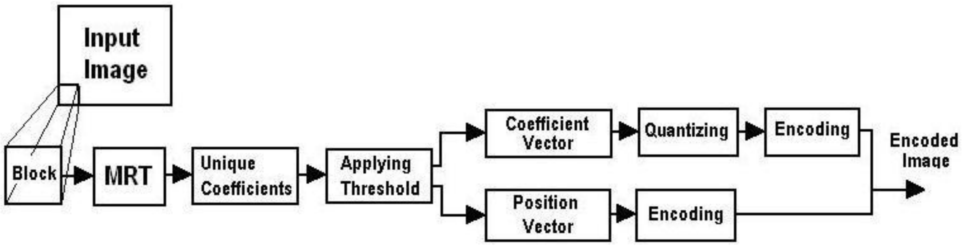


Figure 3.4: Block diagram of 8×8 MRT based transform coder.

- **Partition the image into sub-images**

As explained in section 3.1.2, size of the sub-image is a significant factor. Since the most popular sub-image size is 8×8, the proposed method also adopts 8×8 as the size of the sub-image. So, the image to be compressed is first partitioned into 8×8 non-overlapping sub-images.

- **Application of MRT to each sub-image**

After the image to be compressed is partitioned into 8×8 non-overlapping blocks, 8×8 MRT is applied to each block using Equation (1.22).

For calculating 8×8 MRT, N should be replaced with 8 in the Equations (1.22), (1.23) and (1.24). Application of 8×8 MRT to a sub-image generates 4 × 8×8 MRT coefficients.

- **Identification of the unique MRT coefficients**

The MRT maps an 8×8 sub-image into 256 coefficients, of which only 64 are unique [135]. All the remaining coefficients are just repetitions or sign changed versions of the unique coefficients. The 64 unique MRT coefficients are selected, ignoring the remaining, as per indices given in Figure 3.2. The exact reproduction of the sub-image is possible using these unique MRT coefficients.

- **Implementation of Threshold Coding**

The MRT based Image compression utilizes the 64 unique MRT coefficients to represent each 8×8 sub-image, for encoding. Certain coefficients even among these unique coefficients are irrelevant (section 1.2) for reproducing the original sub-image with some acceptable error. This irrelevancy can easily be removed either by zonal coding or by threshold coding. The proposed method utilizes threshold coding, because it is inherently adaptive in the sense that the location of the transform coefficients retained from each sub-image varies from one sub-image to another. The underlying concept is that, for any sub-image, the transform coefficients of largest magnitude make the most significant contribution to the quality of the reconstructed sub-image.

The proposed algorithm uses single global threshold, as it is the computationally simplest method and the level of compression varies from image to image depending on the number of coefficients that exceed the global threshold. This gives flexibility to the proposed algorithm as the compression levels can be varied by varying the threshold appropriately. Value of the threshold can be chosen according to the required compression level and visual quality of reconstructed image.

- **Grouping the retained coefficients and their respective positions**

The MRT coefficients that overcome the threshold in each block and their respective positions in the MRT matrices are stored in two linear arrays. After the above process is applied to all the sub-images, the coefficient array contains all the retained MRT coefficients corresponding to all the sub-images and position array contains the positions of all the retained MRT coefficients. Subsequently, these two linear arrays are subjected to entropy coding.

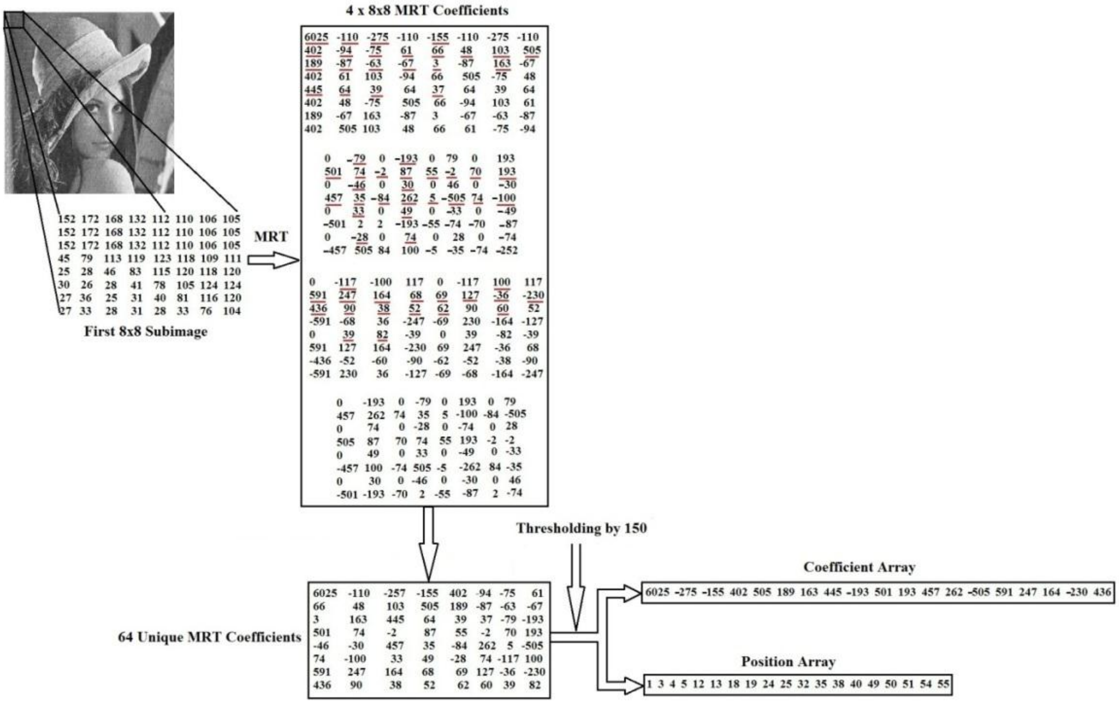
- **Symbol Encoding**

Arithmetic coding and Huffman coding are the two major symbol encoding techniques currently in use. Either of the coding techniques can be used with the proposed algorithm, as both are producing almost the same results.

Figure 3.5 shows the compression process in detail.

3.2.3 Decoding

The Decoding process is shown in Figure 3.6. The encoded bit stream received at the receiver is decoded and the coefficient array & the position array are separated. The coefficient array contains all the retained MRT coefficients and the position array contains the positions of all the retained MRT coefficients in the MRT matrices. The MRT matrices are formed for each sub-image by placing the unique MRT coefficients from the coefficient array with reference to the positions in the position array. The sub-image is reconstructed by applying the inverse MRT (IMRT) to the MRT matrices. Finally the output sub-image is placed in the corresponding position in the image. This process is repeated for all the sub-images.



First 8x8 Subimage

152	172	168	132	112	110	106	105
152	172	168	132	112	110	106	105
152	172	168	132	112	110	106	105
45	79	113	119	123	118	109	111
25	28	46	83	115	120	118	120
30	26	28	41	78	105	124	124
27	36	25	31	40	81	116	120
27	33	28	31	28	33	76	104

4 x 8x8 MRT Coefficients

6025	-110	-275	-110	-155	-110	-275	-110
402	-94	-75	61	66	48	103	505
189	-87	-63	-67	3	-87	163	-67
402	61	103	-94	66	505	-75	48
445	64	39	64	37	64	39	64
402	48	-75	505	66	-94	103	61
189	-67	163	-87	3	-67	-63	-87
402	505	103	48	66	61	-75	-94

0	-79	0	-193	0	79	0	193
501	74	-2	87	55	-2	70	193
0	-46	0	30	0	46	0	-30
457	35	-84	262	5	-505	74	-100
0	33	0	49	0	-33	0	-49
-501	2	2	-193	-55	-74	-70	-87
0	-28	0	74	0	28	0	-74
-457	505	84	100	-5	-35	-74	-252

0	-117	-100	117	0	-117	100	117
591	247	164	68	69	127	-36	-230
436	90	38	52	62	90	60	52
-591	-68	36	-247	-69	230	-164	-127
0	39	82	-39	0	39	-82	-39
591	127	164	-230	69	247	-36	68
-436	-52	-60	-90	-62	-52	-38	-90
-591	230	36	-127	-69	-68	-164	-247

0	-193	0	-79	0	193	0	79
457	262	74	35	5	-100	-84	-505
0	74	0	-28	0	-74	0	28
505	87	70	74	55	193	-2	-2
0	49	0	33	0	-49	0	-33
-457	100	-74	505	-5	-262	84	-35
0	30	0	-46	0	-30	0	46
-501	-193	-70	2	-55	-87	2	-74

64 Unique MRT Coefficients

6025	-110	-257	-155	402	-94	-75	61
66	48	103	505	189	-87	-63	-67
3	163	445	64	39	37	-79	-193
501	74	-2	87	55	-2	70	193
-46	-30	457	35	-84	262	5	-505
74	-100	33	49	-28	74	-117	100
591	247	164	68	69	127	-36	-230
436	90	38	52	62	60	39	82

Coefficient Array

6025 -275 -155 402 505 189 163 445 -193 501 193 457 262 -505 591 247 164 -230 436

Position Array

1 3 4 5 12 13 18 19 24 25 32 35 38 40 49 50 51 54 55

Figure 3.5: Illustration of coding an 8×8 sub-image

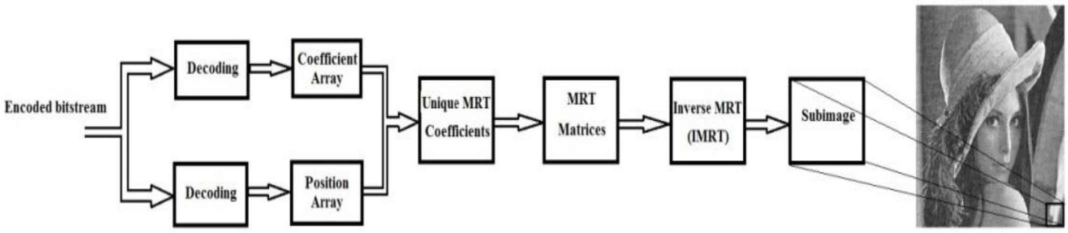


Figure 3.6: Decompression block diagram

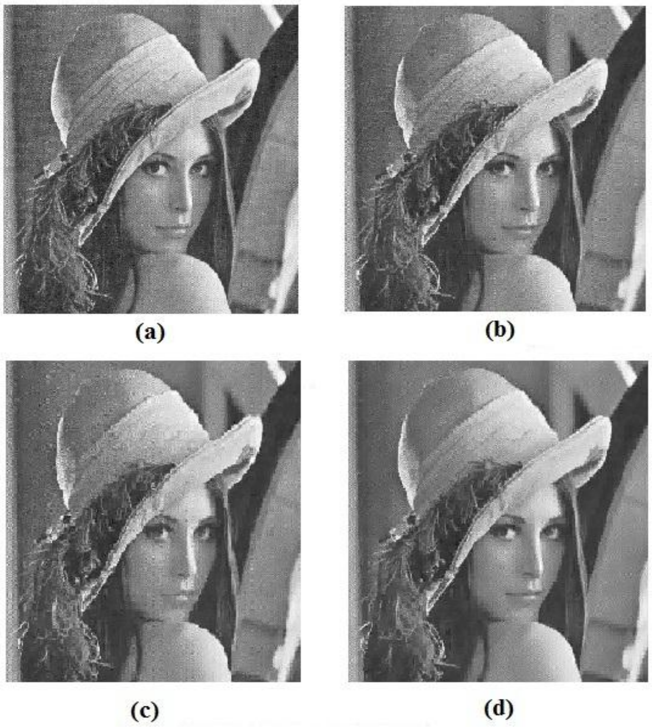
3.2.4 Results and Analysis

The transform coding technique developed using 8×8 MRT is simulated on MATLAB platform over intel(R) core TM(2) CPU T5600 1.83GHz processor, 3GB RAM based system. The compression technique developed above is simulated using *Lena* **(**256×256) image for different values of threshold, from 100 to 400 in steps of 50, and the results are tabulated in Table 3.1.

Table 3.1: Bpp and PSNR obtained for 8×8 MRT based transform coder for Lena (256×256) image for various threshold values

Threshold	% of compression	bpp	PSNR(dB)
100	79.5393	1.6369	30.2151
150	86.6009	1.0719	27.8763
200	89.9725	0.8022	26.2880
250	92.3851	0.6092	25.1379
300	93.6525	0.5078	24.3103
350	94.7740	0.4181	23.5874
400	95.4840	0.3613	22.9757

For each threshold value, bpp, percentage of compression and PSNR are computed using Equations (1.1), (1.2) & (1.4) respectively. Figure 3.7 shows the original and reconstructed Lena images for different values of threshold.



(a) (b)

(c) (d)

Figure 3.7: Lena (256×256) (a) original and (b) – (d) reconstructed after compressed using 8×8 MRT based transform coder with thresholds 100, 150 & 200

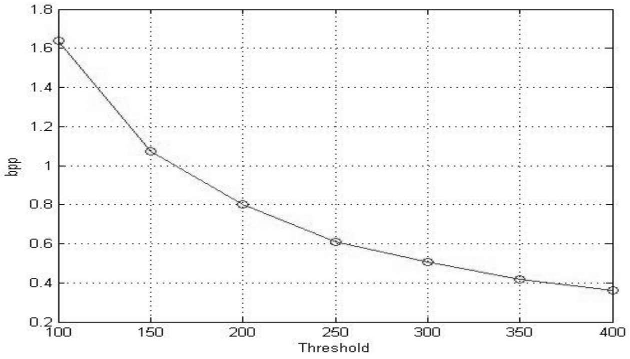


Figure 3.8: Threshold versus bpp for Lena (256×256)

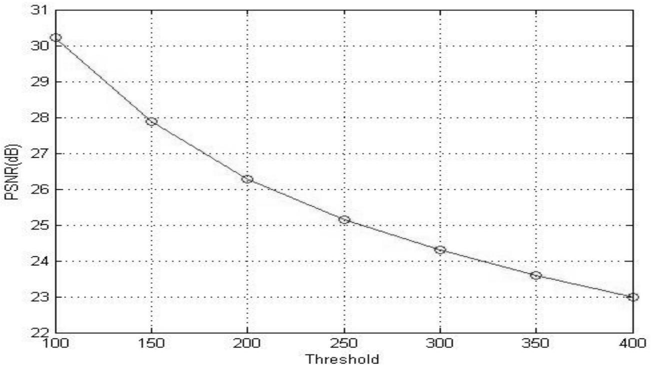


Figure 3.9: Threshold versus PSNR for Lena (256×256)

The compression and PSNR achieved through the proposed coding technique for different values of threshold are graphically represented in Figure 3.8 and 3.9 respectively. As the threshold increases, both compression and PSNR decrease exponentially. Hence, the required PSNR and bpp can be achieved through proper selection of the value for threshold. The relation between bpp and PSNR is shown in Figure 3.10.

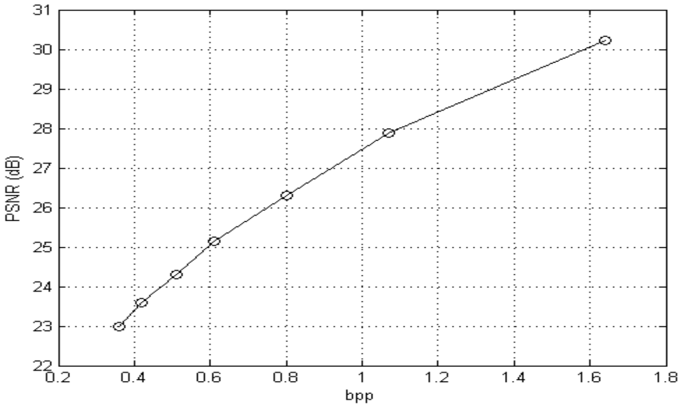


Figure 3.10: bpp versus PSNR for image Lena (256×256)

The algorithm is applied to other standard gray scale images for a fixed threshold value of 150 and the results obtained are shown in Table 3.2. The table shows that other standard gray scale images are also being compressed with similar results as that of *Lena* image. From the analysis it is clear that all types of general gray scale images can be compressed using this method. The selection of threshold is an important factor for obtaining an optimum compression with required quality of reconstructed image.

Table 3.2: Performance of 8×8 MRT based transform coder for various images for a fixed threshold value

Image	Threshold	bpp	PSNR	Time(s)
Lena	150	1.07	27.88	3.0181
Cameraman	150	0.91	28.91	2.9815
Ic	150	0.78	30.14	2.9847
Eight	150	0.52	30.90	3.3771
Rice	150	0.90	29.16	2.9943
Bacteria	150	0.73	30.23	1.4585
Tyre	150	0.90	28.90	2.1646

The last column in Table 3.2 shows the time taken by the proposed algorithm for compressing various images. The time is bit high because of the fact that the algorithm computes all 256 MRT coefficients corresponding to each 8×8 block in the input image, even though only 64 are unique. This is a computational overhead.

Templates of unique MRT coefficients, shown in Figure.3.11, are used to directly compute the unique MRT coefficients from the image data itself, so as to reduce the computational overhead. It is sufficient to add/subtract/don’t care the pixel values of the sub-image corresponding to 1/-1/0 in the template, to compute the 64 unique MRT coefficients of an 8×8 sub-image. The template based MRT computation also eliminates the requirement of computing the parameter ‘z’ in Equation (1.22).

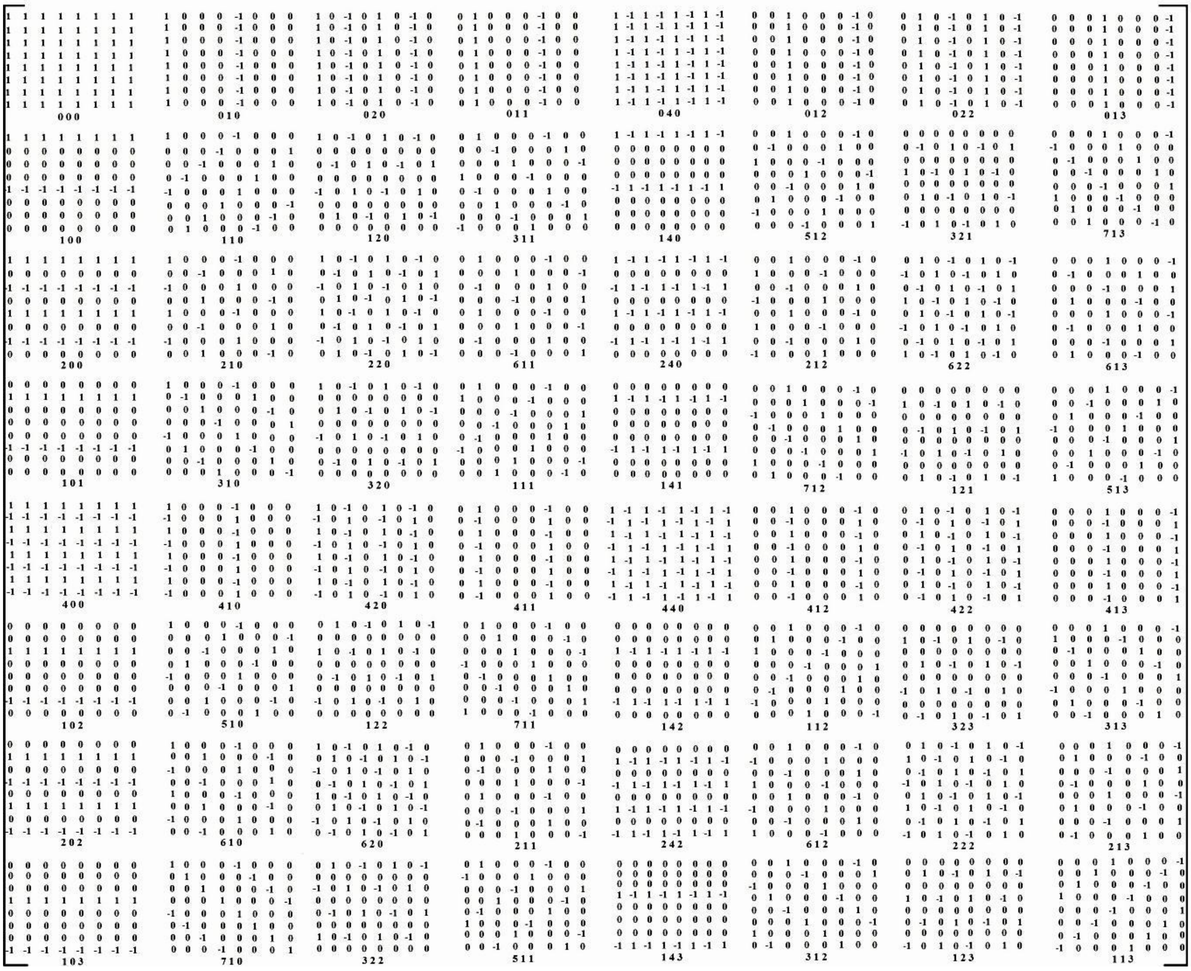


Figure 3.11: Templates for computing unique coefficients of 8×8 MRT

Table 3.3 shows the time taken by the proposed technique when the MRT computation is done through the direct method and template method. The template method is much faster than the direct method.

Table 3.3: Time taken by the direct MRT and template based MRT methods to compress various images

Image	Time(s) (Direct)	Time(s) (Template)
Lena	3.0181	0.3922
Cameraman	2.9815	0.3901
Ic	2.9847	0.3987
Eight	3.3771	0.4461
Rice	2.9943	0.3961
Bacteria	1.4585	0.1896
Tyre	2.1646	0.2846

3.2.4.1 Comparison between DCT and MRT based image compression techniques

Presently DCT is the most common tool for block based transform coding techniques. Hence, a comparison is made between the proposed technique and DCT based transform coding. The results of the DCT based transform coding is obtained by developing an algorithm as in [58], as follows. Initially, the image to be compressed is partitioned into 8×8 non overlapping sub-images. Then for each sub-image, 128 is subtracted from all its elements and 8×8 DCT is applied to the resulting sub-images. The DCT coefficients are quantized using a specially designed fixed matrix called Z [58]. Then the first 4×4 [*Virtual Block Size (VBS) =4*] coefficients of each DCT matrix is stored in a linear array. The linear array is entropy coded after the above process is applied to all the sub-images.

The comparison between DCT and MRT based image compression techniques, presented in Table 3.4 shows that the MRT based image compression technique gives better PSNR for nearly equal or less bpp over baseline DCT based approach for some of the images. The values given in brackets, along with the name of images in column 1 of Table 3.4, is the threshold used for MRT based technique to get a compression equivalent to that of DCT based method. MRT based approach has the advantage over DCT based approach in compromising between quality and percentage of compression by varying threshold. The MRT based image compression technique, proposed here takes almost the same computation time compared to DCT based approach.

Table 3.4: A comparison of 8×8 MRT and DCT based transform coders

Image (Threshold)	**bpp**		**% of Compression**		**Compression Time (s)**		**PSNR (dB)**	
	DCT	**MRT**	**DCT**	**MRT**	**DCT**	**MRT**	**DCT**	**MRT**
Lena (195)	0.84	0.82	89.46	89.74	0.38	0.39	27.81	26.40
Cameraman (200)	0.73	0.70	90.87	91.22	0.37	0.39	26.61	27.34
Ic (180)	0.71	0.70	91.09	91.27	0.38	0.39	27.05	29.41
Tire (170)	0.83	0.82	89.64	89.78	0.28	0.28	31.77	28.33
Eight (150)	0.53	0.52	93.39	93.46	0.43	0.44	30.87	30.90
Enamel (265)	0.78	0.77	90.25	90.39	0.59	0.59	27.31	25.00
Cell (125)	0.46	0.45	94.29	94.42	0.16	0.17	37.74	33.41
Bacteria (170)	0.69	0.68	91.39	91.50	0.17	0.19	31.28	29.72
Rice (170)	0.82	0.82	89.79	89.76	0.37	0.39	37.57	28.50
Nodules (140)	0.65	0.65	91.82	91.90	0.72	0.77	36.67	30.95

3.2.4.2 Application in images with higher resolution

The above mentioned results are obtained when the proposed method is applied to images with resolution 256×256 or less. The proposed technique can be applied to images with higher resolutions (*ie*, images with resolutions 512×512 or more). The performance of the proposed technique when applied to the *Lena (512×512)* image for different values of threshold is shown in Table 3.5. Figure 3.12 shows the original *Lena (512×512)* image and a set of reconstructed images of *Lena* for values of threshold from 100 to 400 with step of 50. At higher compression ratios, threshold 350 or more, blocking artifacts are visible.

Table 3.5: Performance of 8×8 MRT based transform coder for Lena (512×512) image for various threshold values

Threshold	% of compression	bpp	PSNR(dB)
100	90.3112	0.7751	32.2258
150	93.5428	0.5166	30.4469
200	94.9712	0.4023	29.1633
250	96.0501	0.3160	28.1436
300	96.7055	0.2636	27.2951
350	97.2016	0.2239	26.6725
400	97.5261	0.1979	26.1467

The output of the proposed technique for various 512×512 images for a fixed threshold of 150 is shown in Table 3.6. Figures 3.13 to 3.21 show the original and reconstructed version of various gray scale images for a fixed threshold of 150.

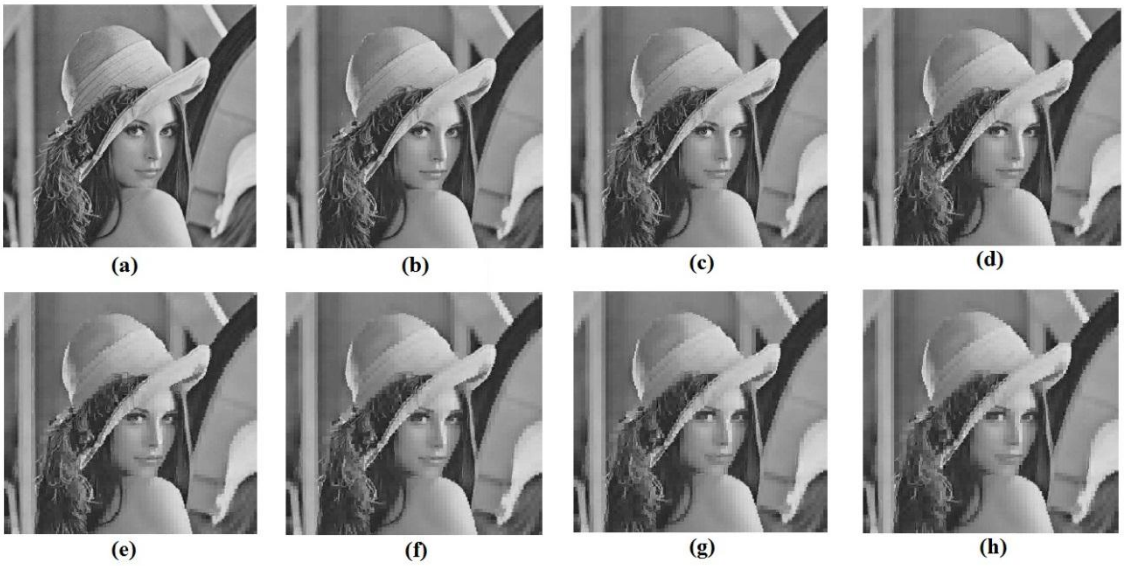


Figure 3.12: Lena (512×512) (a) original and (b) – (h) reconstructed after compressed using 8×8 MRT based transform coder with threshold varied from 100 to 400 in a step of 50

Table 3.6: Performance of 8×8 MRT based transform coder for various 512×512 images for a fixed threshold

Image(512x512)	Threshold	bpp	PSNR	Time(s)
Lena	150	0.52	30.45	1.5688
Baboon	150	1.32	25.87	1.6343
Barbara	150	0.75	28.58	1.5921
Goldhill	150	0.50	29.09	1.5679
peppers	150	0.53	30.17	1.5502
Couple	150	0.63	30.11	1.5821
Elaine	150	0.40	29.40	1.5713
Boat	150	0.63	28.89	1.5600
Cameraman	150	0.49	31.13	1.5593
Bridge and Stream	150	1.01	26.29	1.5672

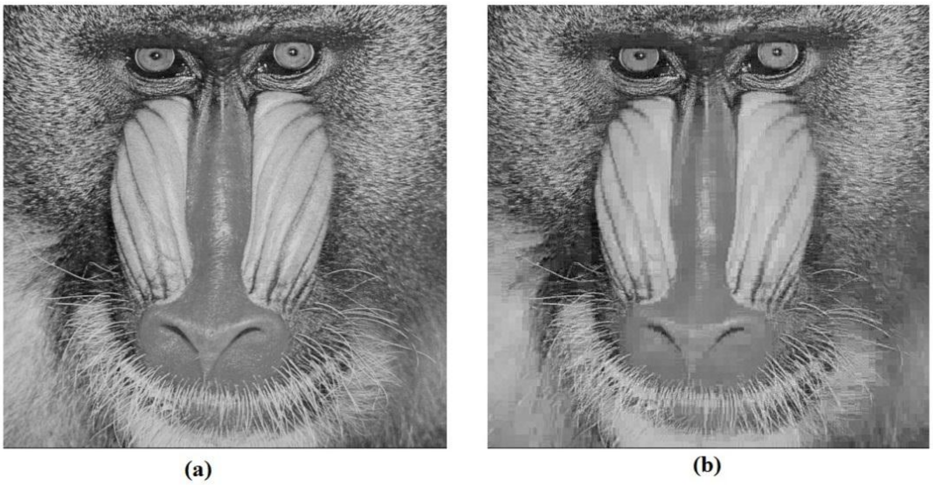


Figure 3.13: Baboon (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 1.32, PSNR 25.87)

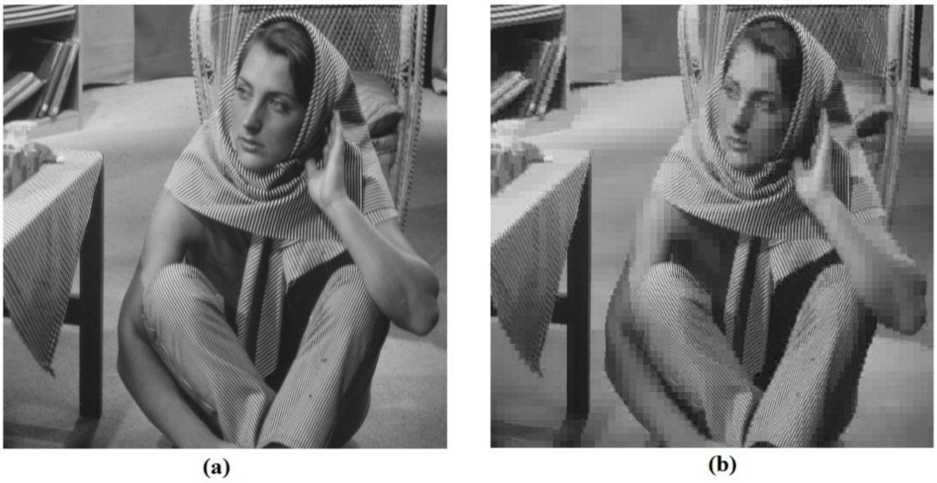


Figure 3.14: Barbara (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.75, PSNR 28.58)

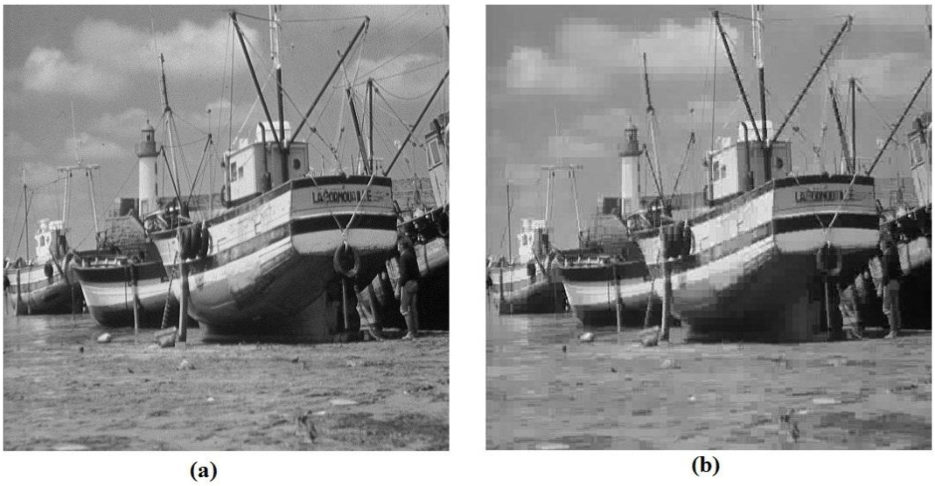


Figure 3.15: Boat (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.63, PSNR 28.89)

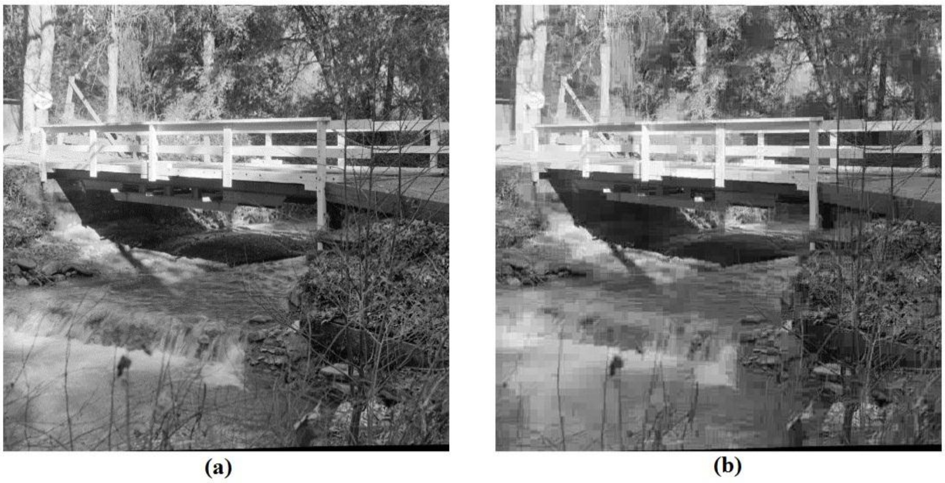
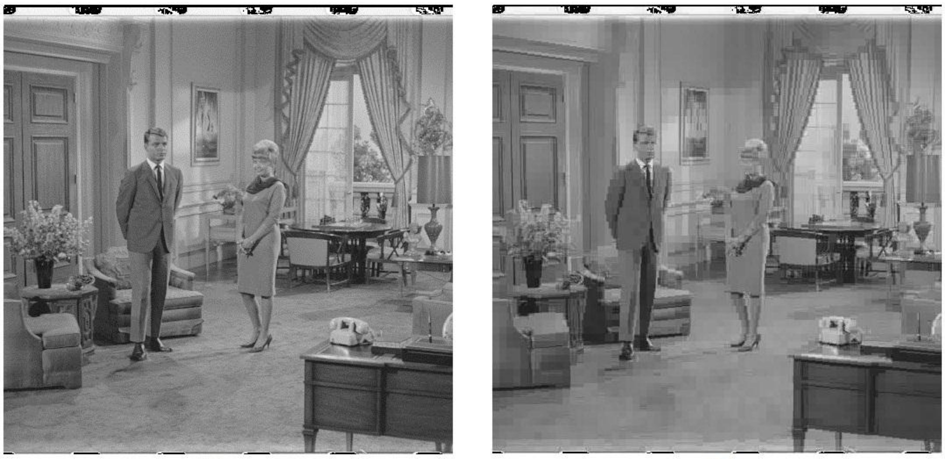
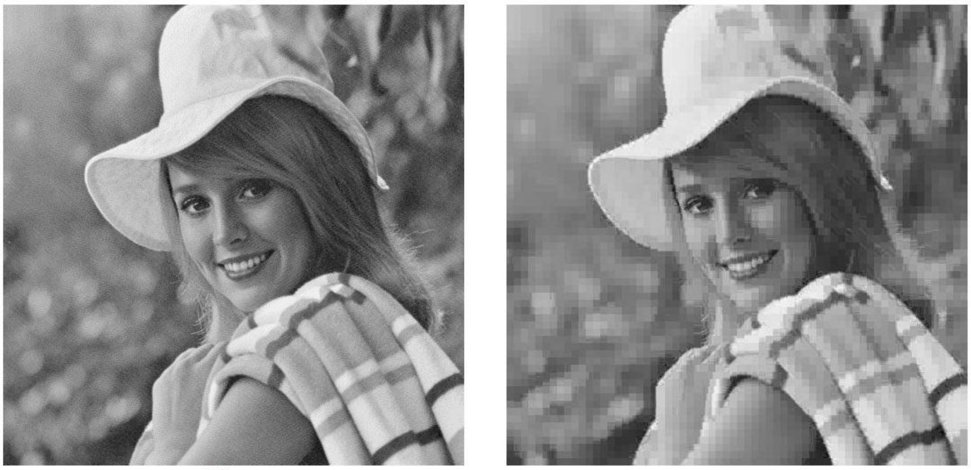


Figure 3.16: Bridge and stream (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 1.01, PSNR 26.29)



(a) (b)

Figure 3.17: Couple (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.63 , PSNR 30.11)



(a) (b)

Figure 3.18: Elaine (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.40 , PSNR 29.40)

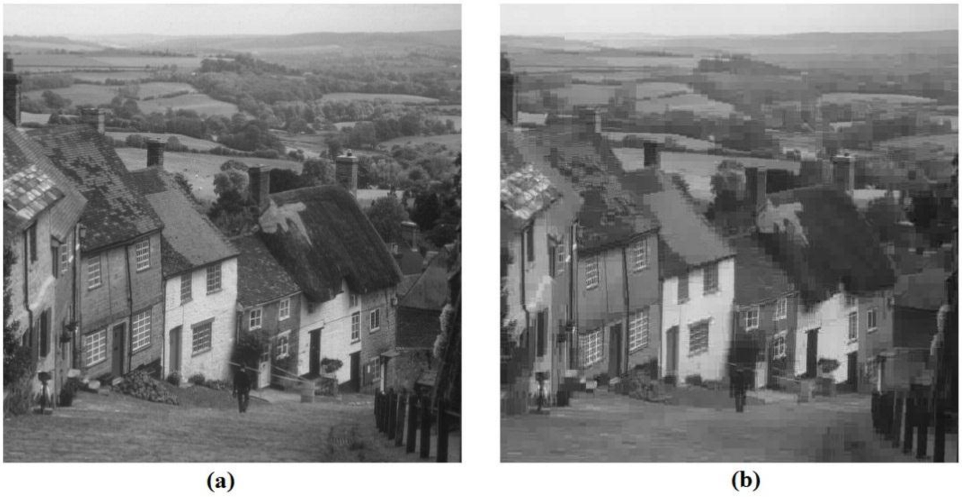

(a)
(b)

Figure 3.19: Goldhill (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.50, PSNR 29.09)

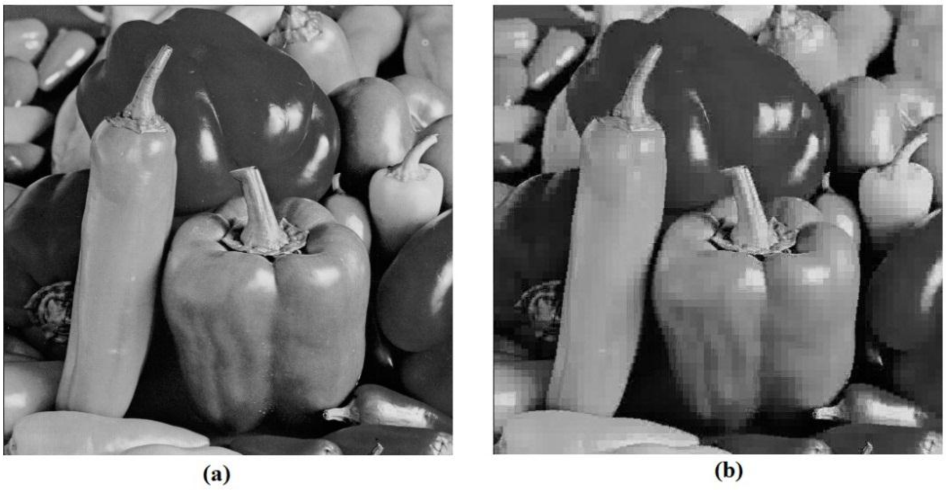

(a)
(b)

Figure 3.20: Peppers (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.53, PSNR 30.17)

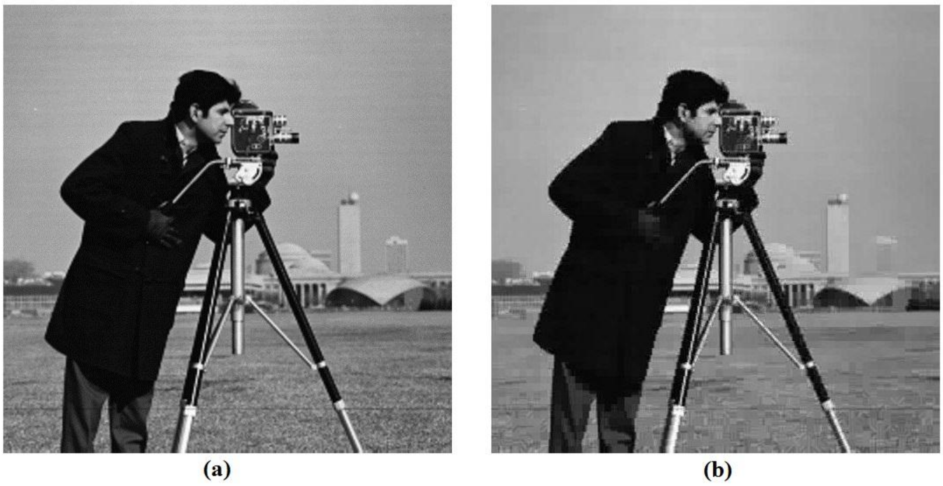


Figure 3.21: Cameraman (512×512) (a) original and (b) reconstructed after compressed using 8×8 MRT based transform coder (threshold 150, bpp 0.49, PSNR 31.13)

3.2.5 Application in Color images

Since the number of bits required to represent color is typically three to four times greater than that employed in the representation of gray levels, data compression plays an important role in the storage and transmission of color images. RGB model is preferred in this work because of its simplicity. Images represented in the RGB color model consist of three component images, one for each color. As a result, each color plane can be compressed separately just like a single gray level image.

The Table 3.7 shows the results of compression for certain color images using the proposed algorithm with the value of threshold as 150. Figure 3.22 shows the original and decompressed versions of various color images like, *Lena, Baboon, Crown* and *Peppers*.

Table 3.7: Performance of 8×8 MRT based transform coder for color images

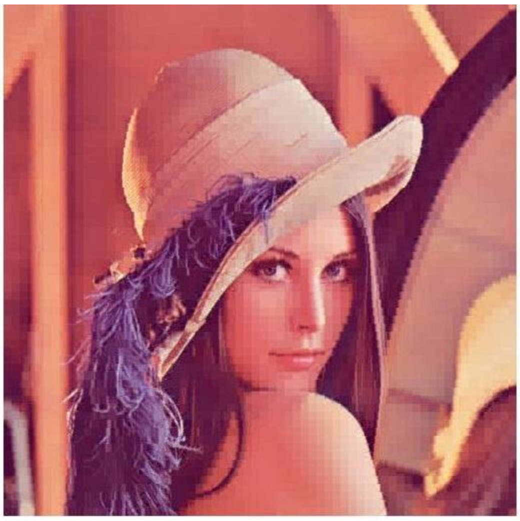
Image	bpp	PSNR
Lena	1.43	30.62
Crown	1.00	33.70
Baboon	3.87	25.93
Peppers	1.52	30.24

3.3 Transform coder based on 4×4 MRT

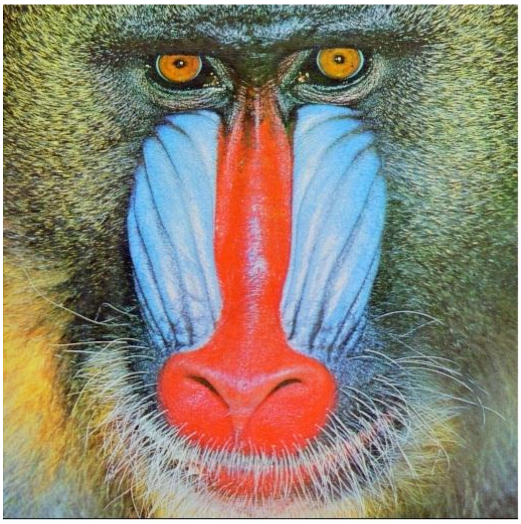
The 8×8 MRT based coding technique, discussed in section 3.2, introduces blocking artifacts in the reconstructed image for higher compression ratios (*i.e.*, for lower bpp). This will be a major problem when bpp is below 0.4. Also, the Table 3.4 shows that the performance of the 8×8 MRT based coder is lower than that of DCT based approach for certain images. These problems can be solved by choosing a smaller size for the sub-images. So the sub-image size is chosen as 4×4.

The technique used in 4×4 MRT based transform coder is the same as that used in 8×8 MRT based transform coder.

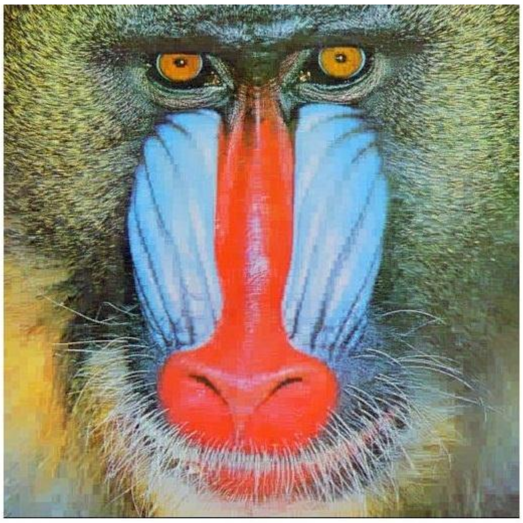
(a)



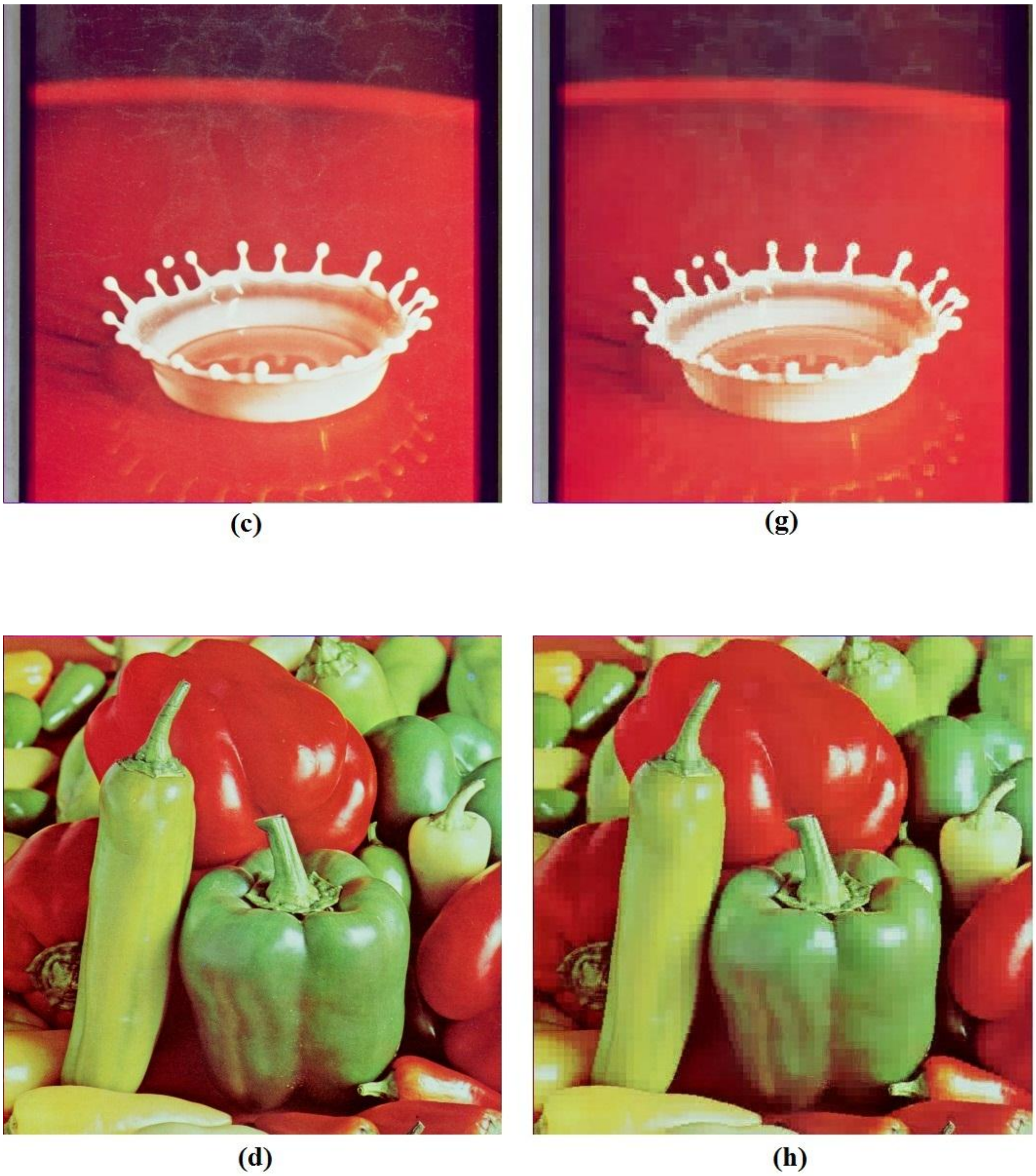
(e)



(b)



(f)

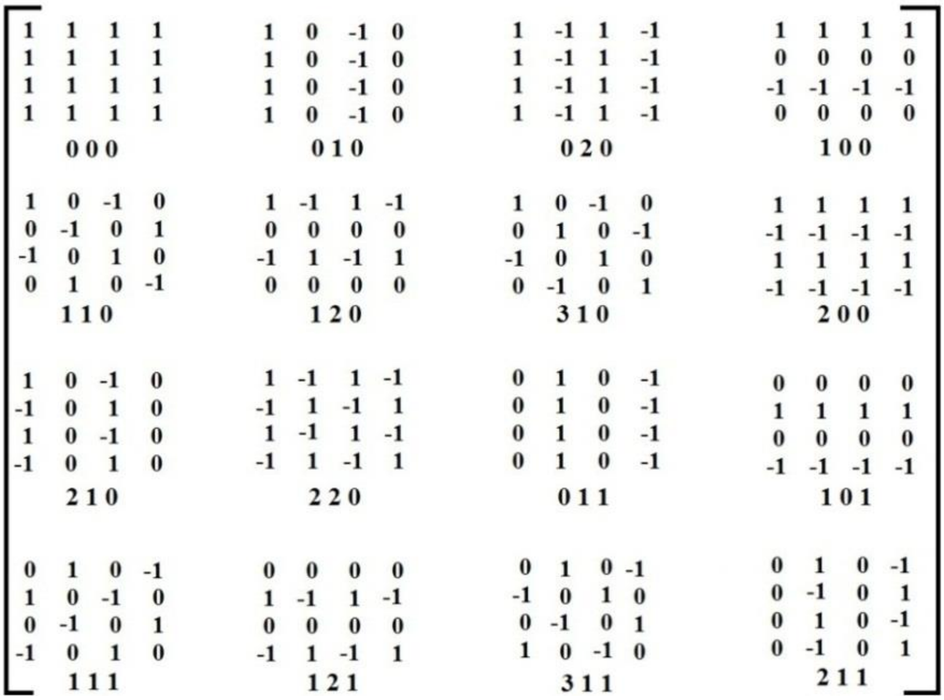

(c) (g)

(d) (h)

Figure 3.22: (a) – (d) Original Color images and (e) – (h) their reconstructed versions after compressed using 8×8 MRT based transform coder

3.3.1 Unique Coefficients in 4×4 MRT matrices

It is quite evident from the MRT matrices that a majority of its coefficients are redundant. The MRT of a 4×4 image data contains 32 coefficients of which only 16 are unique ($Y_{0,0}^{(0)}$, $Y_{0,1}^{(0)}$, $Y_{0,2}^{(0)}$, $Y_{1,0}^{(0)}$, $Y_{1,1}^{(0)}$, $Y_{1,2}^{(0)}$, $Y_{3,1}^{(0)}$, $Y_{2,0}^{(0)}$, $Y_{2,1}^{(0)}$, $Y_{2,2}^{(0)}$, $Y_{0,1}^{(1)}$, $Y_{1,0}^{(1)}$, $Y_{1,1}^{(1)}$, $Y_{1,2}^{(1)}$, $Y_{3,1}^{(1)}$ and $Y_{2,1}^{(1)}$). The templates for computing these unique coefficients of 4×4 MRT are shown in Figure 3.23.



$$\left[\begin{array}{cccc}
\begin{matrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{matrix} &
\begin{matrix}1&0&-1&0\\1&0&-1&0\\1&0&-1&0\\1&0&-1&0\end{matrix} &
\begin{matrix}1&-1&1&-1\\1&-1&1&-1\\1&-1&1&-1\\1&-1&1&-1\end{matrix} &
\begin{matrix}1&1&1&1\\0&0&0&0\\-1&-1&-1&-1\\0&0&0&0\end{matrix}\\
000 & 010 & 020 & 100\\
\begin{matrix}1&0&-1&0\\0&-1&0&1\\-1&0&1&0\\0&1&0&-1\end{matrix} &
\begin{matrix}1&-1&1&-1\\0&0&0&0\\-1&1&-1&1\\0&0&0&0\end{matrix} &
\begin{matrix}1&0&-1&0\\0&1&0&-1\\-1&0&1&0\\0&-1&0&1\end{matrix} &
\begin{matrix}1&1&1&1\\-1&-1&-1&-1\\1&1&1&1\\-1&-1&-1&-1\end{matrix}\\
110 & 120 & 310 & 200\\
\begin{matrix}1&0&-1&0\\-1&0&1&0\\1&0&-1&0\\-1&0&1&0\end{matrix} &
\begin{matrix}1&-1&1&-1\\-1&1&-1&1\\1&-1&1&-1\\-1&1&-1&1\end{matrix} &
\begin{matrix}0&1&0&-1\\0&1&0&-1\\0&1&0&-1\\0&1&0&-1\end{matrix} &
\begin{matrix}0&0&0&0\\1&1&1&1\\0&0&0&0\\-1&-1&-1&-1\end{matrix}\\
210 & 220 & 011 & 101\\
\begin{matrix}0&1&0&-1\\1&0&-1&0\\0&-1&0&1\\-1&0&1&0\end{matrix} &
\begin{matrix}0&0&0&0\\1&-1&1&-1\\0&0&0&0\\-1&1&-1&1\end{matrix} &
\begin{matrix}0&1&0&-1\\-1&0&1&0\\0&-1&0&1\\1&0&-1&0\end{matrix} &
\begin{matrix}0&1&0&-1\\0&-1&0&1\\0&1&0&-1\\0&-1&0&1\end{matrix}\\
111 & 121 & 311 & 211
\end{array}\right]$$

Figure 3.23: Templates for the computation of the 16 unique coefficients of 4×4 MRT

It is sufficient to add/subtract/don't care the pixel values of the sub-image corresponding to 1/-1/0 in the template, to compute the 16 unique MRT coefficients of a 4×4 sub-image.

3.3.2 Encoding

The image is partitioned into 4×4 sub-images. The 16 unique MRT coefficients corresponding to each sub-image are computed, using the templates discussed in section 3.3.1, and placed in a 4×4 matrix. Irrelevancy reduction is performed on these unique MRT coefficients by the application of a fixed threshold. The threshold should be selected according to the required compression level and visual quality of the reconstructed image. Typically, the value of the threshold can be varied from 50 to 400 as per the requirement of the application in which this compression scheme is used. Two linear arrays are formed, for the image, from the value and position of the unique MRT coefficients that overcome the threshold. The resulting two arrays are then encoded separately using simple Arithmetic encoding. The proposed coder is shown in Figure 3.24.

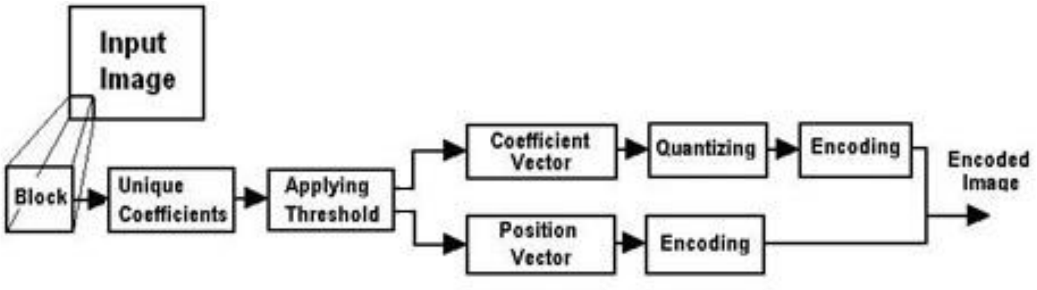


Figure 3.24: The 4×4 MRT based transform coding scheme.

3.3.3 Decoding

In the decoding process, both the linear arrays (coefficient and position) are decoded using Arithmetic decoding. The MRT matrices are formed, for each sub-image, by placing the unique coefficients from the coefficient array with reference to the positions in the position array and deriving the redundant coefficients using the relations, $Y_{0,3}^{(0)} = Y_{0,1}^{(0)}$, $Y_{2,3}^{(0)} = Y_{2,1}^{(0)}$, $Y_{3,0}^{(0)} = Y_{1,0}^{(0)}$, $Y_{1,3}^{(0)} = Y_{3,1}^{(0)}$, $Y_{3,2}^{(0)} = Y_{1,2}^{(0)}$, $Y_{3,3}^{(0)} = Y_{1,1}^{(0)}$, $Y_{0,3}^{(1)} = -Y_{0,1}^{(1)}$, $Y_{2,3}^{(1)} = -Y_{2,1}^{(1)}$, $Y_{3,0}^{(1)} = -Y_{1,0}^{(1)}$, $Y_{1,3}^{(1)} = -Y_{3,1}^{(1)}$, $Y_{3,2}^{(1)} = -Y_{1,2}^{(1)}$, $Y_{3,3}^{(1)} = -Y_{1,1}^{(1)}$, $Y_{0,0}^{(1)} = 0$, $Y_{0,2}^{(1)} = 0$, $Y_{2,0}^{(1)} = 0$ and $Y_{2,2}^{(1)} = 0$. The output sub-images are derived through inverse MRT.

3.3.4 Results and Analysis

The transform coder discussed above is simulated and the results for the *'Lena' (512×512)* image under different values of threshold are shown in Table 3.8. The selection of threshold is very crucial for getting a pre-determined compression ratio. Figure 3.25(a) shows the original '*Lena*' image and Figures 3.25(b) to (i) show the reconstructed images corresponding to the threshold values 50, 100, 150, …, 400 respectively.

The transform coder is simulated for various images of size 512×512 for a fixed threshold of 150 and the quality of the corresponding reconstructed image is represented in terms of bpp & PSNR as given in Table 3.9.

Table 3.8: Performance of 4×4 MRT based transform coder for Lena image for various threshold values

Threshold	bpp	PSNR
50	0.91	33.25
100	0.58	31.18
150	0.44	29.61
200	0.38	28.63
250	0.34	28.00
300	0.32	27.62
350	0.31	27.35
400	0.30	27.15

Table 3.9: Performance of 4×4 MRT based transform coder for various images for a fixed threshold

Image(512x512)	Threshold	bpp	PSNR	Time(s)
Lena	150	0.44	29.61	1.4418
Baboon	150	0.76	24.28	1.4832
Barbara	150	0.61	27.03	1.4547
Goldhill	150	0.43	28.60	1.4401
peppers	150	0.46	29.98	1.4459
Couple	150	0.51	29.16	1.4635
Elaine	150	0.39	29.63	1.4592
Cameraman	150	0.45	30.76	1.4395
Boat	150	0.50	28.29	1.4402
Bridge and Stream	150	0.63	25.21	1.4434

As it is observed from the reconstructed images of *Lena*, the blocking artifacts present in the 8×8 MRT based technique are reduced to a large extent.

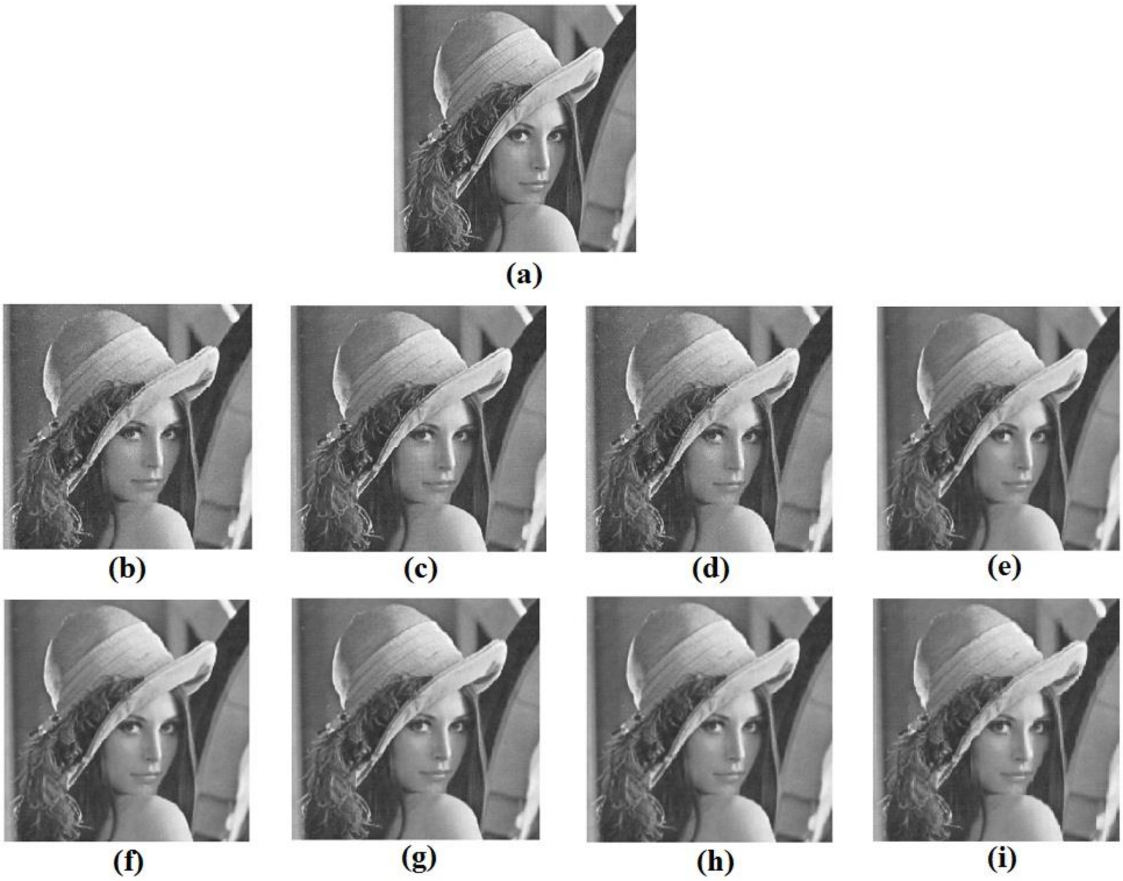


Figure 3.25: Lena (512×512) (a) original and (b) – (i) reconstructed after compressed using 4×4 MRT based transform coder with bpp 0.91, 0.58, 0.44, 0.38, 0.34, 0.32, 0.31 and 0.30 respectively

3.3.5 Application in Color images

The change in block size of the transform coder from 8×8 to 4×4 is applied on color images to analyze the effect. The simulation results of the proposed algorithm on various color images are tabulated in Table 3.10. Figure 3.26 shows the original and decompressed versions of various color images, using the proposed algorithm, for a fixed threshold of 150.

Table 3.10: Performance of 4×4 MRT based transform coder for color images

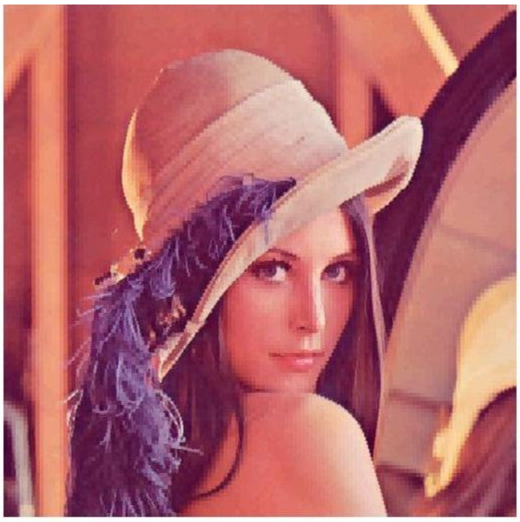
Image	bpp	PSNR
Lena	1.41	30.87
Crown	1.06	33.82
Baboon	3.92	25.96
Peppers	1.47	30.31

The analysis of the simulation results shows that this algorithm can be used to compress all types of gray scale and color images. The value of threshold determines the compromise between bpp and PSNR.

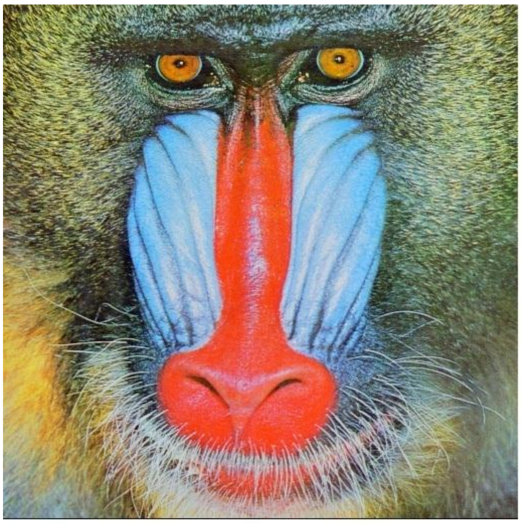
A comparison between the proposed coder and the DCT based technique for a given bpp is shown in Table 3.11. On analysis of the table, it is clear that the proposed 4×4 MRT based technique performs equal to or better than the DCT based technique for 6 out of 13 images.

A comparison between proposed coding scheme using 4×4 MRT and 8×8 MRT is shown in Table 3.12. From the table it is quite evident that the proposed 4×4 MRT gives better performances for almost all images.

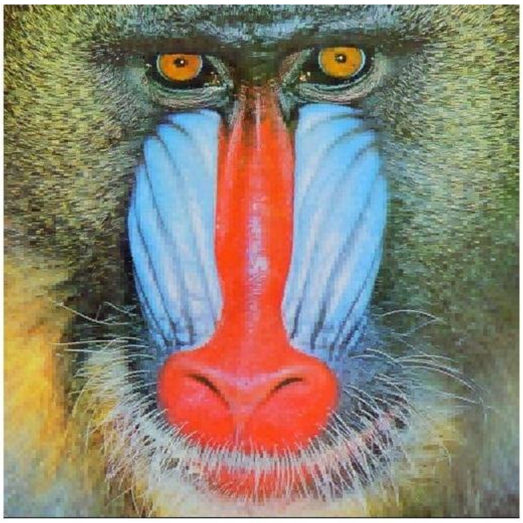
(a)



(e)



(b)



(f)

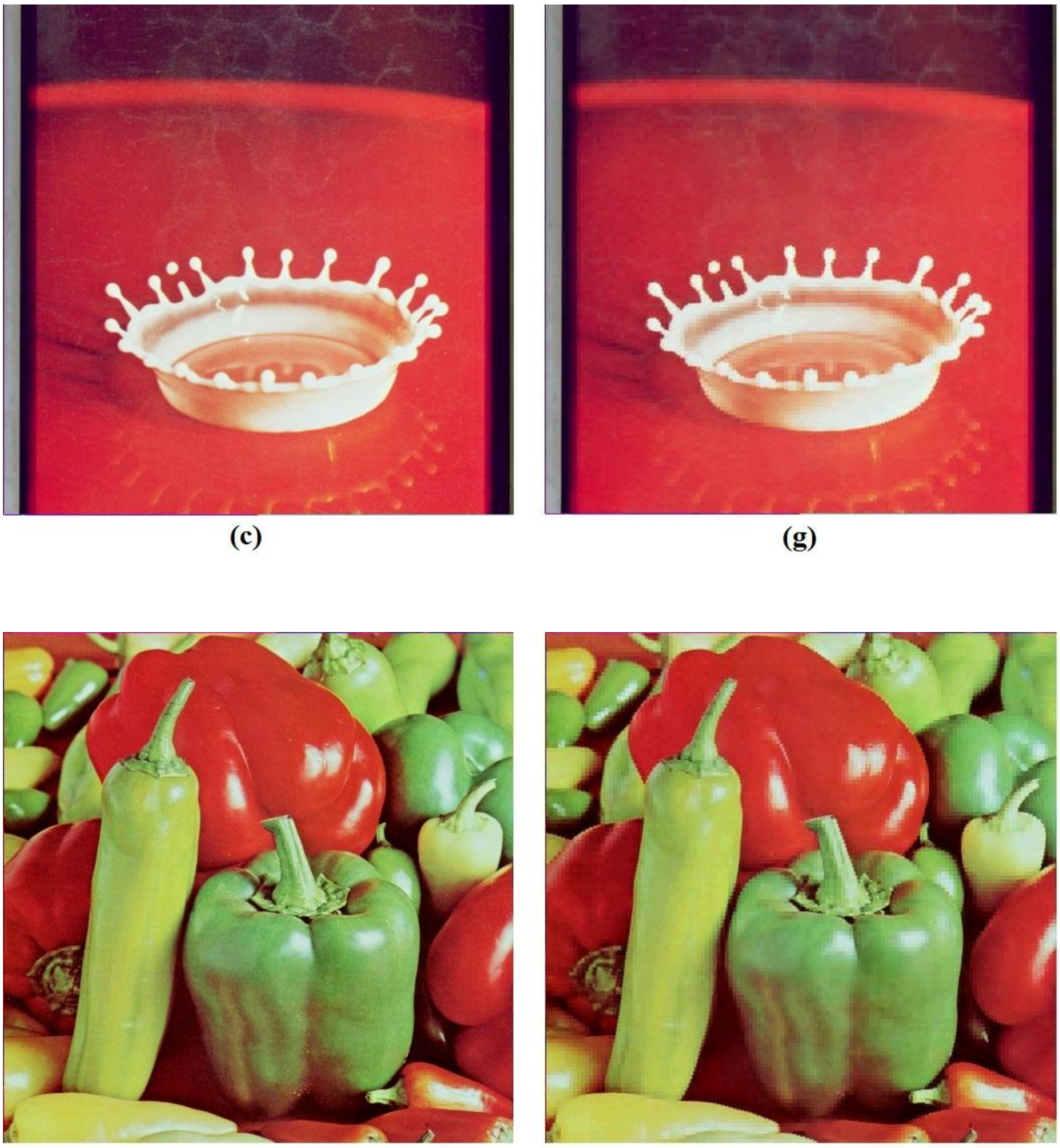


Figure 3.26: (a) – (d) Original Color images and (e) – (h) their reconstructed versions after compressed using 4×4 MRT based transform coder

Table 3.11: Comparison between 8×8 DCT and 4×4 MRT based methods

Image (Threshold)	bpp	PSNR (DCT)	PSNR (MRT)
Barbara (120)	0.74	25.58	28.00
Baboon (130)	0.93	23.99	25.00
Couple (90)	0.70	27.89	31.31
Enamel (135)	0.78	27.31	29.56
Boat (85)	0.76	30.10	30.48
Bridge and stream (105)	0.95	26.79	26.79
Goldhill (70)	0.75	31.42	31.22
Lena (83)	0.65	34.00	31.95
Peppers (75)	0.67	32.28	32.23
Cameraman (80)	0.63	35.75	33.21
Elaine (60)	0.66	32.76	31.45
Alumgrns (75)	0.50	40.99	36.44
Satelite_image (105)	0.96	27.76	27.52

Table 3.12: Comparison between 8×8 MRT and 4×4 MRT based methods

Image	PSNR(dB)					
	0.3 bpp		0.5 bpp		0.75 bpp	
	8×8	4×4	8×8	4×4	8×8	4×4
Lena	27.68	27.05	30.18	30.27	32.03	32.72
Baboon	21.42	21.13	22.53	22.71	23.60	24.25
Peppers	27.43	26.52	29.86	30.41	31.55	32.71
Goldhill	27.25	26.68	29.04	29.16	30.34	31.22
Couple	26.47	25.87	28.82	29.16	30.86	31.70
Cameraman	28.55	26.79	31.20	31.47	33.35	34.25
Boat	25.94	25.19	27.88	28.29	29.48	30.39
Elaine	28.46	28.37	29.93	30.60	31.15	31.57

Figure 3.27 shows the variation of PSNR with respect to bpp for the two block sizes corresponding to *Lena* image. A careful observation of the Table 3.12 reveals that, for all the images, at higher bit rates, 4×4 MRT based technique gives better performances and at lower bit rates 8×8 MRT based technique gives better performances. This is because, at lower bit rates, the threshold will be high. As a result, only the DC coefficient needs to be retained for each sub-image. In the case of 8×8 MRT based technique only one DC coefficient is present in an 8×8 sub-image whereas in 4×4 MRT based approach four DC coefficients will be involved for a sub-image of size 8×8 and are to be retained. Hence, the 8×8 MRT based technique performs better than 4×4 MRT based approach at lower bpp. Also for images having lesser gray scale variations, the 8×8 MRT based technique performs better than 4×4 MRT based technique.

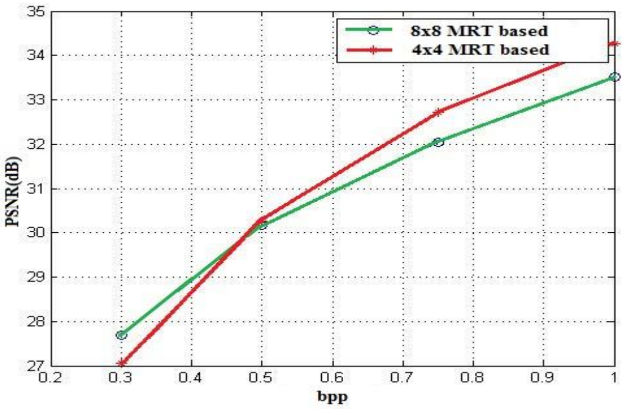


Figure 3.27: Performance comparison of 4×4 and 8×8 MRT based coders for Lena image

A hybrid transform coder, utilizing a combination of 4×4 and 8×8 sub-images, can improve the performance of MRT based transform coding technique by incorporating the good features of both techniques discussed above. The next chapter deals with the implementation of such a coder.

4

Variable block size Transform Coders using MRT

Contents

4.1 Need for variable block size transform coders......102
4.2 Hybrid coder using 4×4 and 8×8 MRT......103
    4.2.1 The Concept......103
    4.2.2 Implementation......104
    4.2.3 Simulation Results......106
    4.2.4 Performance comparison......106
    4.2.5 Application in Color images......109
4.3 MRT based Adaptive block size Transform Coder (MATC)......109
    4.3.1 Unique Mapped real Transform (UMRT)......110
    4.3.2 UMRT based criterion to segment images......113
    4.3.3 Quad-tree segmentation of images......114
    4.3.4 MRT based Adaptive block size Transform Coding technique (MATC)......117
    4.3.5 Simulation Results......121
    4.3.6 Performance comparison......123
    4.3.7 Application in Color images......124

4.1 Need for Variable block size Transform Coders

Majority of natural images can be segmented into regions of high and low detail. Since high detail regions are intrinsically less compressible than regions of low detail, it is very attractive to implement a variable block size image coding scheme that varies the size of the sub-image according to the local detail. Traditional coding schemes, such as transform coding and Vector Quantization, break an image into sub-images prior to the actual encoding. Hence, if the block-size is too small, it is difficult to take advantage of the fact that large areas of the image may consist of low detail. And if the block-size chosen is very high, the large non-uniform areas of the image may not be segmented into smaller uniform areas. The block-size in transform coding is usually large enough to meet the above mentioned barriers. However, when the block size is kept constant, as is the case with most of the transform coding schemes, it is impossible to do an adequate job of isolating the high detail regions. The solution to this problem is to segment an image into blocks of varying sizes. Large blocks are used in large regions of low detail, and small blocks when the detail gets high.

Transform-based image coding algorithms have been the object of intense research during the last three decades. Eventually they have been selected as the main mechanism of data compression in the definition of digital image and video coding standards. A transform-based image coding method involves subdividing the original image into smaller N×N blocks of pixels and applying a unitary transform, such as DCT, on each of these blocks. In general, once the value of N has been selected for a particular algorithm, it remains fixed. In JPEG, for instance, the value of N is 8, and thus the input image is divided exclusively into blocks of 8×8 pixels.

The fact that image statistics may be inhomogeneous and vary from area to area is not taken into consideration in fixed block size image coding scheme. Some areas of an image may have only smooth changes and contain no high contrast edge. In such areas, higher compression can be obtained by using a larger block size. However, in areas where high activities and contrast edges are present, a transform of a smaller block size should be used to obtain better visual quality. Therefore, in order to yield a better tradeoff between the bit rate and the quality of decoded image, a transform coding system should vary the block size so as to truly adapt to the internal statistics of the image in different areas.

4.2 Hybrid Coder using 4×4 and 8×8 MRT

Transform coding based on 8×8 MRT alone and 4×4 MRT alone are discussed in chapter 3. The results reveal that at higher bit rates, 4×4 MRT based technique gives better performances and at lower bit rates 8×8 MRT based technique gives better performances. For exploiting the advantages of both the coders discussed in chapter 3, the use of a technique which combines 4×4 and 8×8 MRT is proposed.

4.2.1 The Concept

The homogeneous portions of the image should be segmented into 8×8 sub-images and the inhomogeneous portions should be partitioned into 4×4 sub-images. Hence, there should be a criterion for deciding whether a portion of the image is homogeneous or not. Several measures can be taken, like, standard deviation, mean difference etc., as this criterion. The difference between the maximum pixel value and the minimum pixel value within the area under consideration is a simple and effective criterion for the decision making.

4.2.2 Implementation

The compression process starts by tiling (segmenting) the image into non-overlapping 8×8 sub-images. The difference between the maximum pixel value and minimum pixel value within each sub-image is computed. If the difference is less than the segmentation threshold t_s, the current sub-image is considered to be of uniform nature and it should be transform coded without any further segmentation as in section 3.2.2. If the difference is greater than or equal to the threshold t_s, the given 8×8 sub-image is considered to be of non-uniform nature and it should further be segmented into four 4×4 sub-images. These four 4×4 sub-images are individually transform coded as in section 3.3.2. The above process is repeated for all the 8×8 sub-images in the image and the resulting coefficient and position vectors are coded in binary using Arithmetic encoding.

The unique coefficients of the MRT are directly computed from the image data itself by using templates shown in Figure 3.11 and 3.23. The proposed coder is shown as a flowchart in Figure.4.1.

4.2.3 Simulation Results

A computer simulation of the flowchart shown in Figure 4.1 is carried out using MATLAB to study the performances and efficiency of the proposed hybrid transform coding system. The simulation is carried out for gray scale images and color images. The input images have resolution of 8 bits per pixel and the image size is 512×512. From various trial & error experiments based on section 4.2.2 proved that the segmentation threshold t_s can be chosen as 25. The compression threshold t_c is selected as 150 (as in sections 3.2 & 3.3) for both block sizes 8×8 & 4×4. The quantized coefficients are entropy coded using Arithmetic coding. Table 4.1 shows the simulation results for various gray scale images.

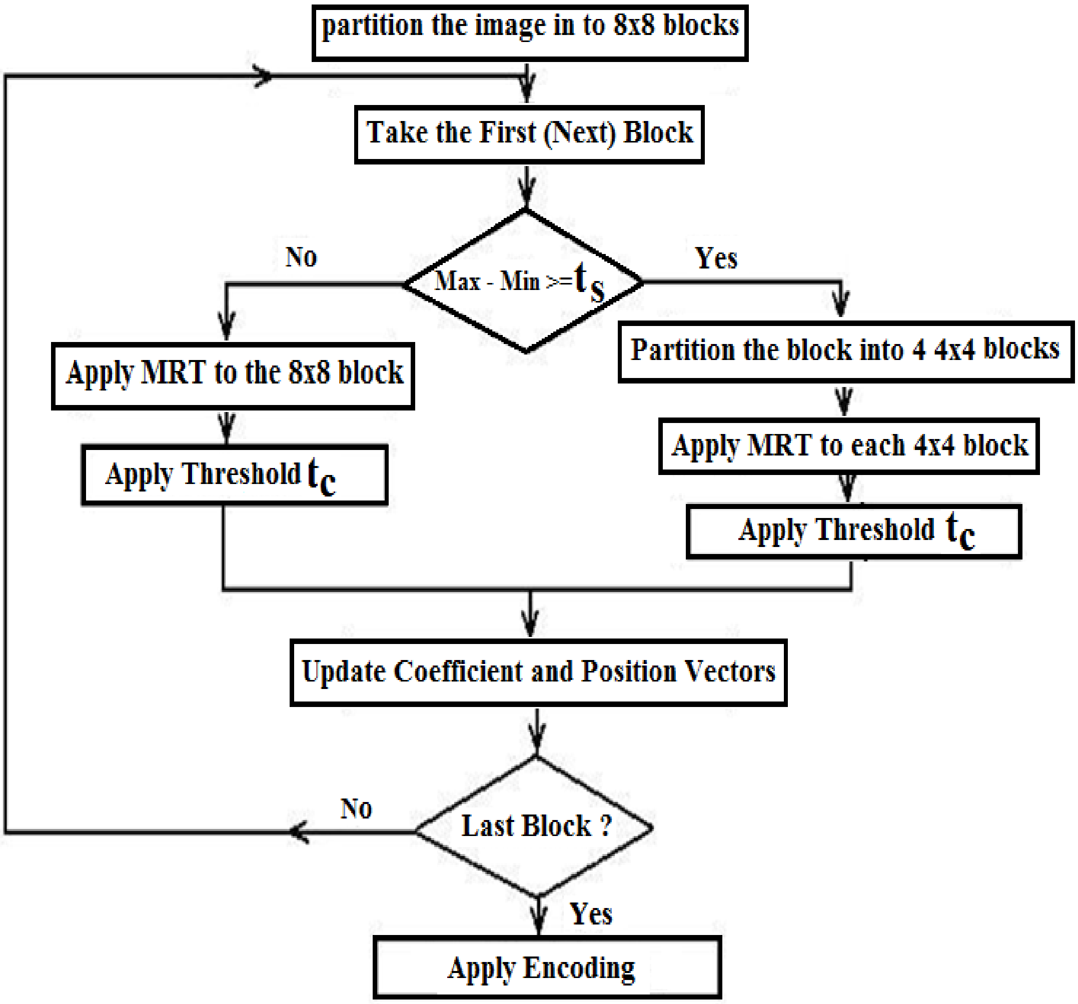

partition the image in to 8x8 blocks
Take the First (Next) Block
No
Max - Min >= t_S
Yes
Apply MRT to the 8x8 block
Partition the block into 4 4x4 blocks
Apply MRT to each 4x4 block
Apply Threshold t_C
Apply Threshold t_C
Update Coefficient and Position Vectors
No
Last Block ?
Yes
Apply Encoding

Figure 4.1: Flow chart of Hybrid Coder using 4×4 and 8×8 MRT

Table 4.1: Performance of Hybrid Coder using 4×4 and 8×8 MRT

Image	Threshold (t_c)	bpp	PSNR (dB)
Lena	150	0.3923	29.5766
Baboon	150	0.7584	24.2795
Barbara	150	0.5681	26.9981
Goldhill	150	0.4162	28.5866
Peppers	150	0.4179	29.9555
Couple	150	0.4928	29.1391
Elaine	150	0.3737	29.6106
Enamel	150	0.6489	28.8857
Cameraman	150	0.3868	30.8105
Boat	150	0.4812	28.2918
Bridge and Stream	150	0.6264	25.2091

4.2.4 Performance Comparison

Performance of the proposed hybrid coder is compared with that of the fixed block size transform coders, discussed in chapter 3, and is shown in Table 4.2. It can be seen from Table 4.2 that the proposed MRT based hybrid block size transform coding system using 8×8 and 4×4 MRT produces better performance at lower and higher bit rates for almost all images. Figure 4.2 shows the Original Lena image and its compressed (at bpp of 0.3) versions using the proposed variable block sized coder and fixed block sized coders. The proposed coder gives a PSNR of 27.98dB while the fixed block size coders using 8×8 MRT and 4×4 MRT give PSNR of 27.68dB and 27.05dB respectively. The performance comparison is plotted in Figure 4.3.

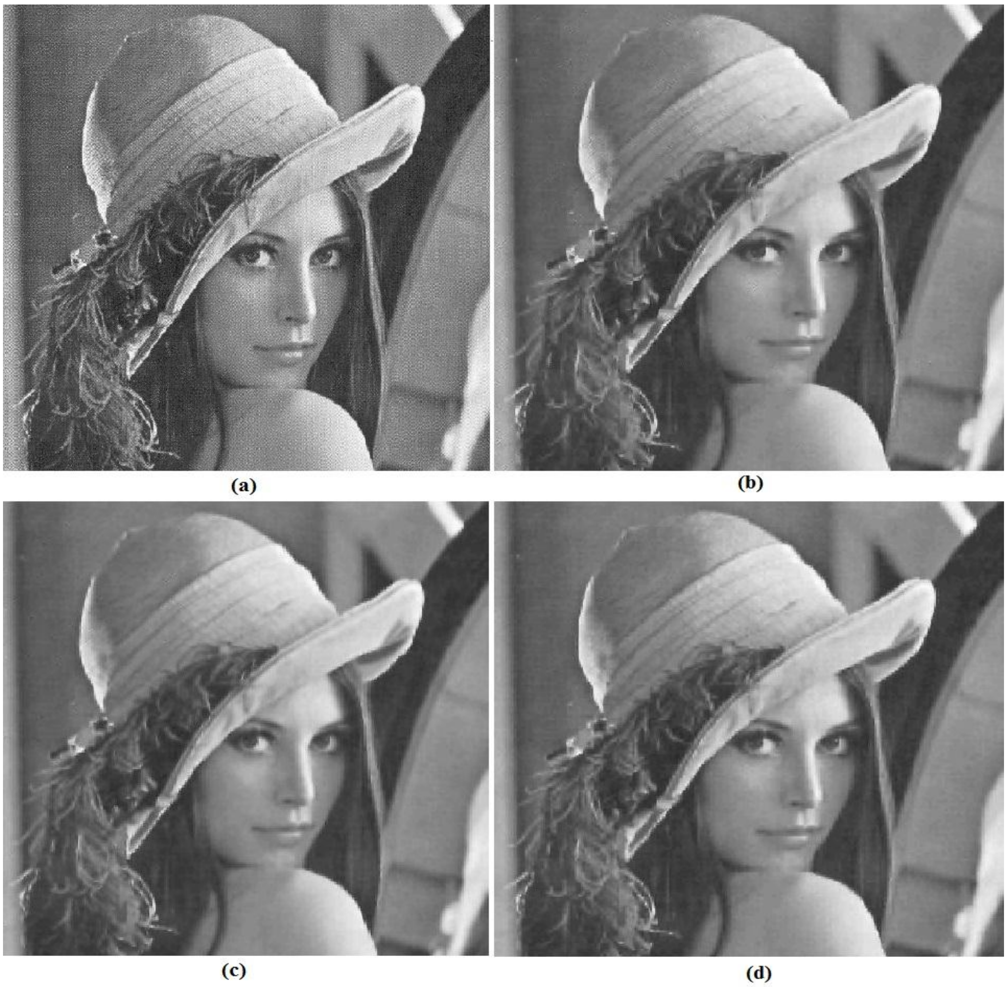

(a)
(b)
(c)
(d)

Figure 4.2: (a) Original Lena (512×512) image (b) – (d) reconstructed versions after compressed at 0.30 bpp using hybrid coder, 8×8 MRT based transform coder and 4×4 MRT based transform coder respectively.

Table 4.2: Comparison of the hybrid coder with fixed block size transform coders

Image	PSNR(dB)								
	0.3 bpp			0.5 bpp			0.75 bpp		
	8×8	4×4	Hybrid	8×8	4×4	Hybrid	8×8	4×4	Hybrid
Lena	27.68	27.05	27.98	30.18	30.27	30.83	32.03	32.72	32.99
Baboon	21.42	21.13	21.13	22.53	22.71	22.71	23.60	24.25	24.25
Peppers	27.43	26.52	27.48	29.86	30.41	30.95	31.55	32.71	32.86
Goldhill	27.25	26.68	26.98	29.04	29.16	29.28	30.34	31.13	31.20
Couple	26.47	25.87	25.94	28.82	29.16	29.21	30.86	31.70	31.80
Cameraman	28.55	26.79	28.63	31.20	31.47	32.56	33.35	34.25	34.69
Boat	25.94	25.19	25.49	27.88	28.29	28.45	29.48	30.39	30.52
Elaine	28.46	28.37	28.50	29.93	30.60	30.65	31.15	31.57	31.57

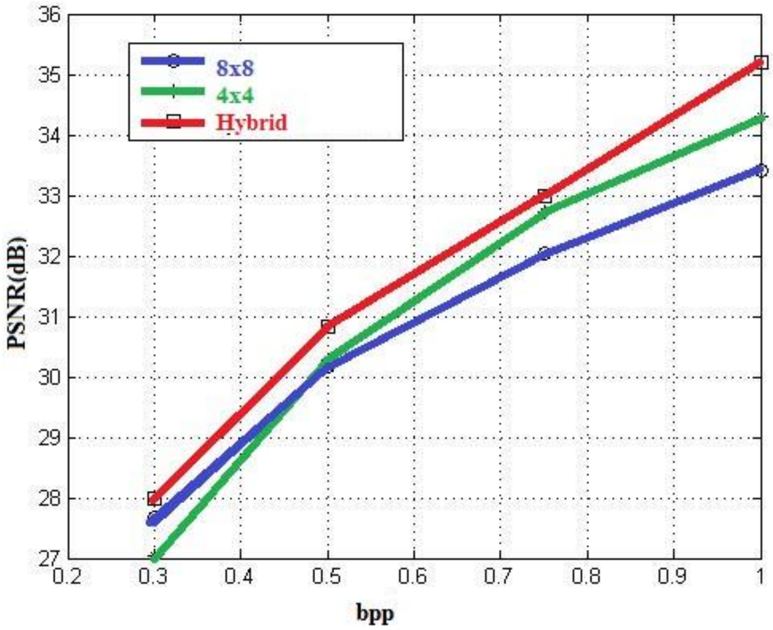


Figure 4.3: Performance comparison of MRT based fixed block size and hybrid techniques for Lena image

4.2.5 Application in Color Images

The proposed hybrid algorithm is applied on various color images and the results are tabulated in Table 4.3. Figure 4.4 shows the original and decompressed versions of various color images using the proposed algorithm for a compression threshold ($\boldsymbol{t_c}$) of 150 for hybrid coder and for 8×8 & 4×4 coders, threshold is varied to obtain the same bpp as in hybrid coder.

Table 4.3: Performance of hybrid coder in color images

Image	**bpp**	**PSNR**		
		Hybrid	**8×8**	**4×4**
Lena	1.34	30.83	30.52	30.83
Crown	1.12	33.84	33.75	33.85
Baboon	3.94	25.98	25.96	25.98
Peppers	1.49	30.33	30.22	30.33

4.3 MRT based Adaptive block size Transform Coder (MATC)

Recently, variable block-size (VBS) image coding techniques have received increasing attention due to their superior adaptability to local image characteristics. Conventional block-based image compression approaches, such as JPEG & Vector Quantization (VQ), simply partition image into fixed-size blocks and seek for spatial-domain and/or frequency-domain redundancy in the blocks. Despite the optimality achieved within each block, fixed block size techniques lack the ability to reduce the bit-rate, as VBS does, to encode low-detailed regions on the images using large blocks.

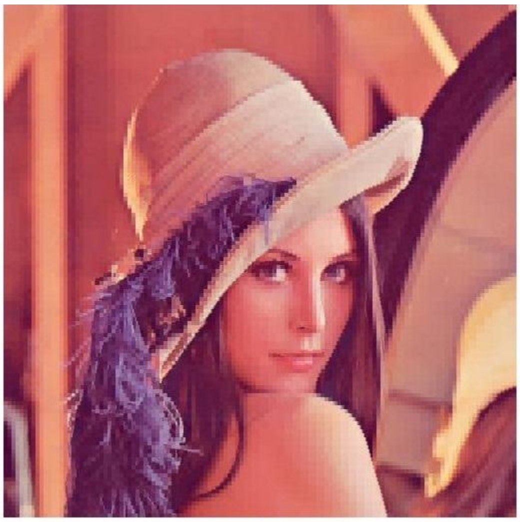
Segmenting images into blocks of variable sizes in the VBS techniques is usually accomplished by using the well-known Quad-Tree (QT) decomposition. Researchers have applied it with several block-based coding schemes. QT decomposition is favourable for its tree structure representation, which allows efficient bit storage for encoding the tree structure, and straightforward binary decision on split nodes.

The unique MRT coefficients are computed using templates, for encoding the sub-images, in hybrid coding technique discussed in section 4.2. The fixed block size transform coding techniques and the hybrid technique discussed earlier needs the template representation for block sizes 4×4, 8×8 or both. Whereas in variable block size coding technique, block sizes vary in powers of 2 from size of the image to the smallest block size 2×2. There will be N^2 templates of size N×N for any block size N×N, which demands large memory to store and takes more retrieval time. So, the existing template method is not suitable for adaptive block size transform coding techniques. Hence, an alternative method, Unique Mapped Real Transform (UMRT) [154], is adopted for computation of unique MRT coefficients.

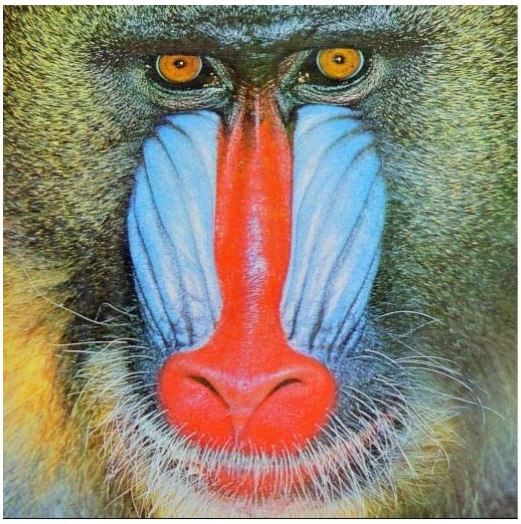
4.3.1 Unique Mapped Real Transform (UMRT)

Both MRT and MRT based Image coding are discussed in previous chapters. The MRT representation shows a specific pattern of redundant elements in the MRT matrices. A compact unique MRT (UMRT) representation, a single matrix of size N×N, is derived by eliminating the computation of the redundant elements present in the MRT representation [154]. Different algorithms are proposed for computing and placing the N^2 UMRT coefficients in an N×N matrix [150] & [154].

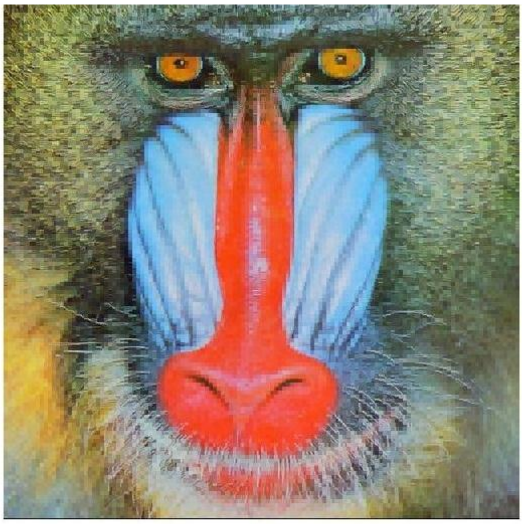
(a)



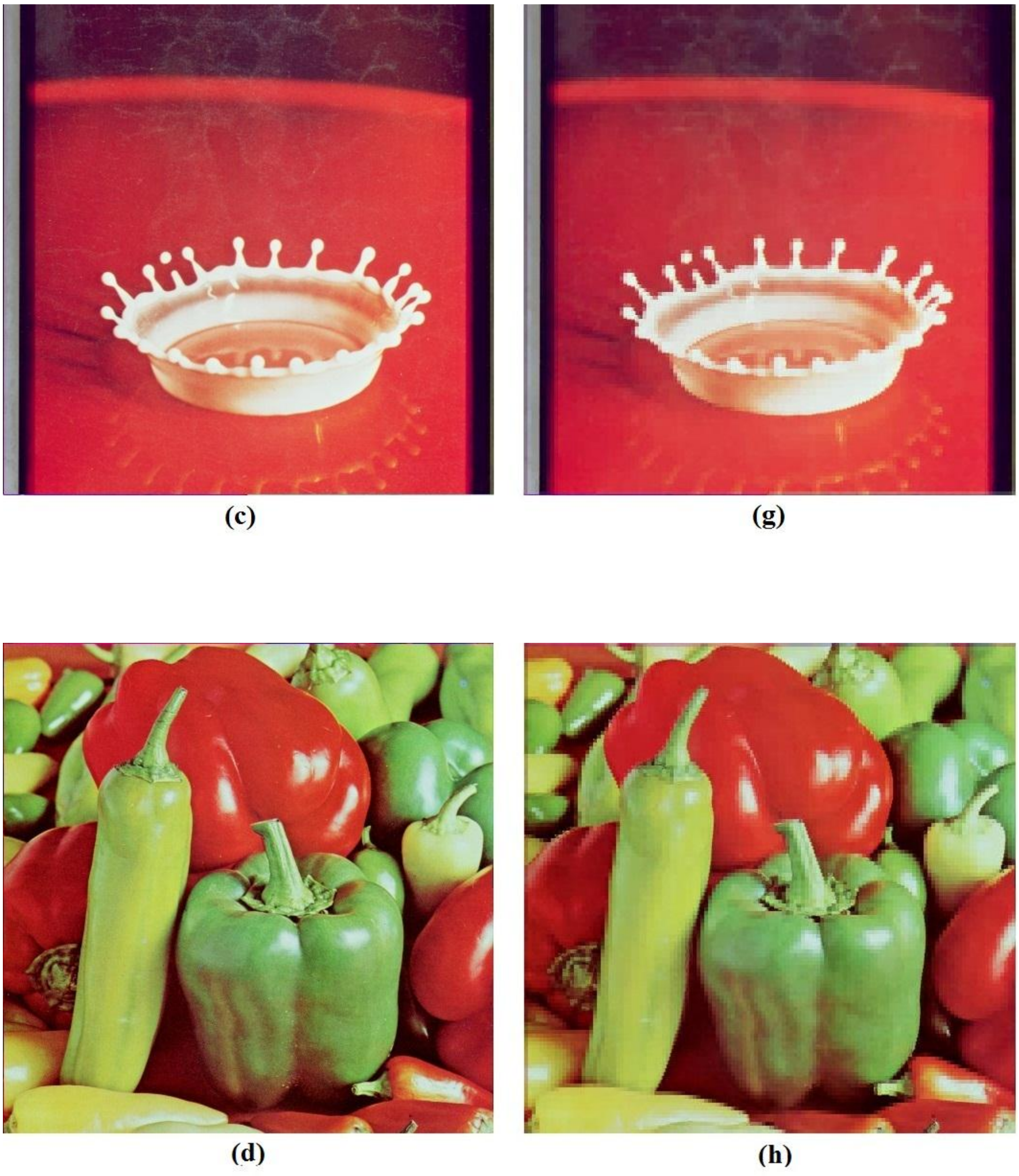
(e)



(b)



(f)



(c) (g)

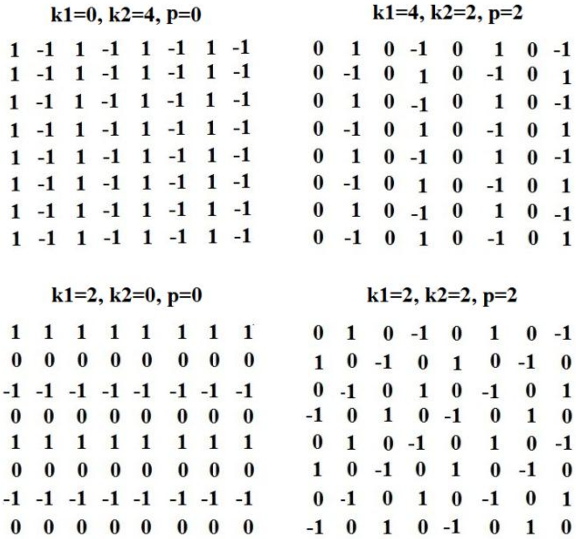
(d) (h)

Figure 4.4: (a) – (d) Original Color images and (e) – (h) their reconstructed versions after compressed using hybrid transform coder

4.3.2 UMRT Based Criterion to segment images

As discussed in the previous section, the UMRT is a compact form of MRT. Since UMRT coefficients are computed from simple addition and subtraction of selected image data, each UMRT coefficient signifies a particular manner of combination of image data under consideration. Consequently, each UMRT coefficient signifies a unique pattern in the image domain. Thus, the N^2 UMRT coefficients of an N×N image block are, in fact, strength-indicators of N^2 different patterns in the image block. These N^2 patterns can be considered to be the basis vectors of the UMRT. The UMRT thus becomes a suitable tool for image analysis to determine the existence of N^2 well defined patterns in an image. Figure 4.5 shows the patterns associated with the computation certain UMRT coefficients for an 8×8 image data. The symbol '1' indicates that the image data at that position is an addend in the formation of that particular UMRT coefficient, and '-1' denotes a subtrahend. '0' indicates positions of image data that do not contribute to that UMRT coefficient. From the patterns of UMRT coefficients, as shown in Figure 4.5, it can be observed that the 64 UMRT coefficients signify 64 different patterns in the image block. Thus, by examining the strength of these UMRT coefficients, the homogeneity of the image block can be identified.

Thus the magnitude of the UMRT coefficients, other than the DC coefficient, can be used as measure of the homogeneity of the image block under consideration. If the magnitude of any of the UMRT coefficients, other than the DC coefficient, of a particular N×N image block is higher than a threshold value t_s, then the image block contains edge or inhomogeneous area and is to be partitioned into four N/2×N/2 blocks. This process is shown in Figure 4.6. Experiments proved that 100 is a good value for the segmentation threshold t_s.



k1=0, k2=4, p=0

$$\begin{matrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{matrix}$$

k1=4, k2=2, p=2

$$\begin{matrix} 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 \end{matrix}$$

k1=2, k2=0, p=0

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

k1=2, k2=2, p=2

$$\begin{matrix} 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 \end{matrix}$$

Figure 4.5: Examples of templates for the computation UMRT Coefficients

4.3.3 Quad-Tree segmentation of images

A Quad-Tree is a data structure that can be used to efficiently address regions of different size in an image. This technique can thus be applied to problems where the segmentation of images has to be done in regions that are of homogeneous in some property. Other methods of image segmentation such as Region Growing do better jobs of isolating regions of constant pixel statistics; however, in these techniques the shape of the regions is determined solely by the image being examined and this implies that an excessive number of bits are needed to describe the shape and location of each region. This large amount of overhead is unacceptable in applications where it is necessary to code an image efficiently and there are not enough bits left over to encode detailed regions with good fidelity. With structures such as Quad-Trees, a small overhead rate is achieved by decomposing the image into sub-images whose size, shape and location are predetermined.

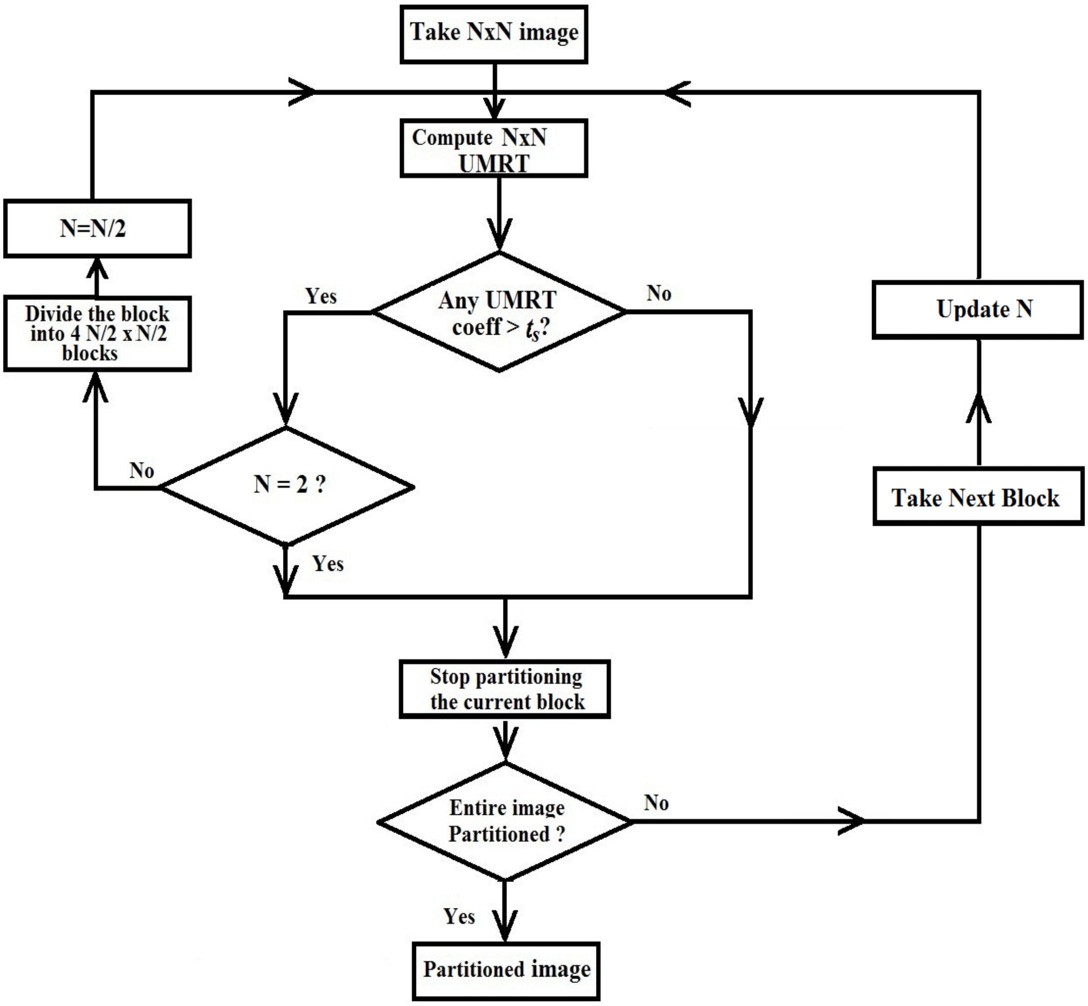


Figure 4.6: Flowchart for the Quad-Tree segmentation of images using UMRT

The Quad-Tree is a tree structure in which each node, unless it is a leaf, generates four children. Each node corresponds to a sub-image of the image that is determined, both in size and location, by its position in the tree. Each child represents a quarter area of its parent. When using this structure to segment an image, start at some initial node level, and travel through the tree. At each node a test is performed to see if the sub-image represented by the node is homogeneous in the property of interest. If the test is positive then the node becomes a leaf, otherwise the children of the node are generated and examined in turn.

A Quad-Tree is said to be a K-level Quad-Tree if the lowest level allowed is K-1. An example of a K-level Quad-Tree structure is shown in Figure.4.7. It can be represented by assigning 0 to non-leaf nodes and 1 to leaf nodes. Each node in a Quad-Tree represents a block. If the block can be split into four quadrants, then the node will generate four children nodes. Otherwise the node becomes a leaf. Quad-Tree partitioned *Lena* images using the proposed UMRT based criterion for selected values of threshold t_s are shown in Figure 4.8 and number of blocks in each category for various values of t_s is shown in Table 4.4.

Table 4.4: No. of blocks of each size in Lena image for various values of t_s

Segmentation Threshold (t_s)	**No. of Blocks**					
	2×2	**4×4**	**8×8**	**16×16**	**32×32**	**64×64**
100	10384	5392	1555	136	0	0
150	5712	4628	1634	221	4	0
200	3040	4256	1514	280	13	0
300	988	3177	1472	338	26	0
400	304	2112	1437	388	35	0

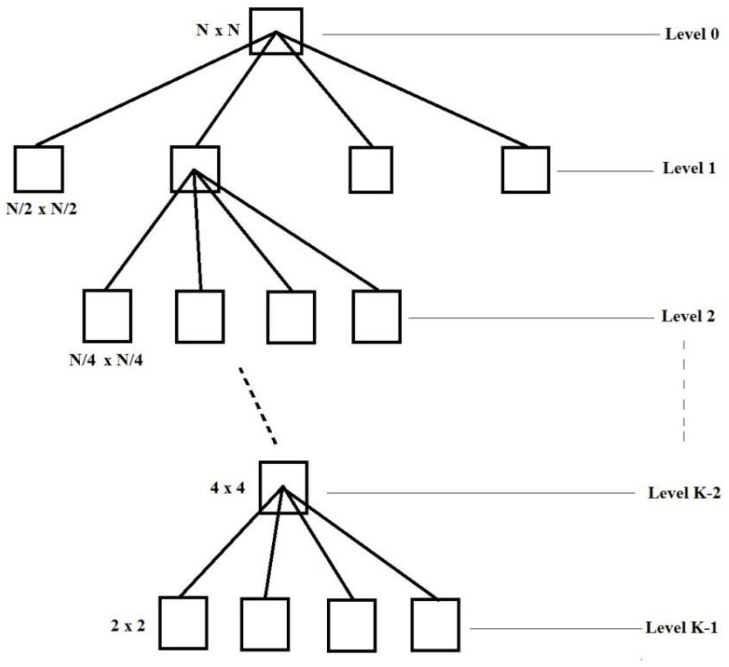
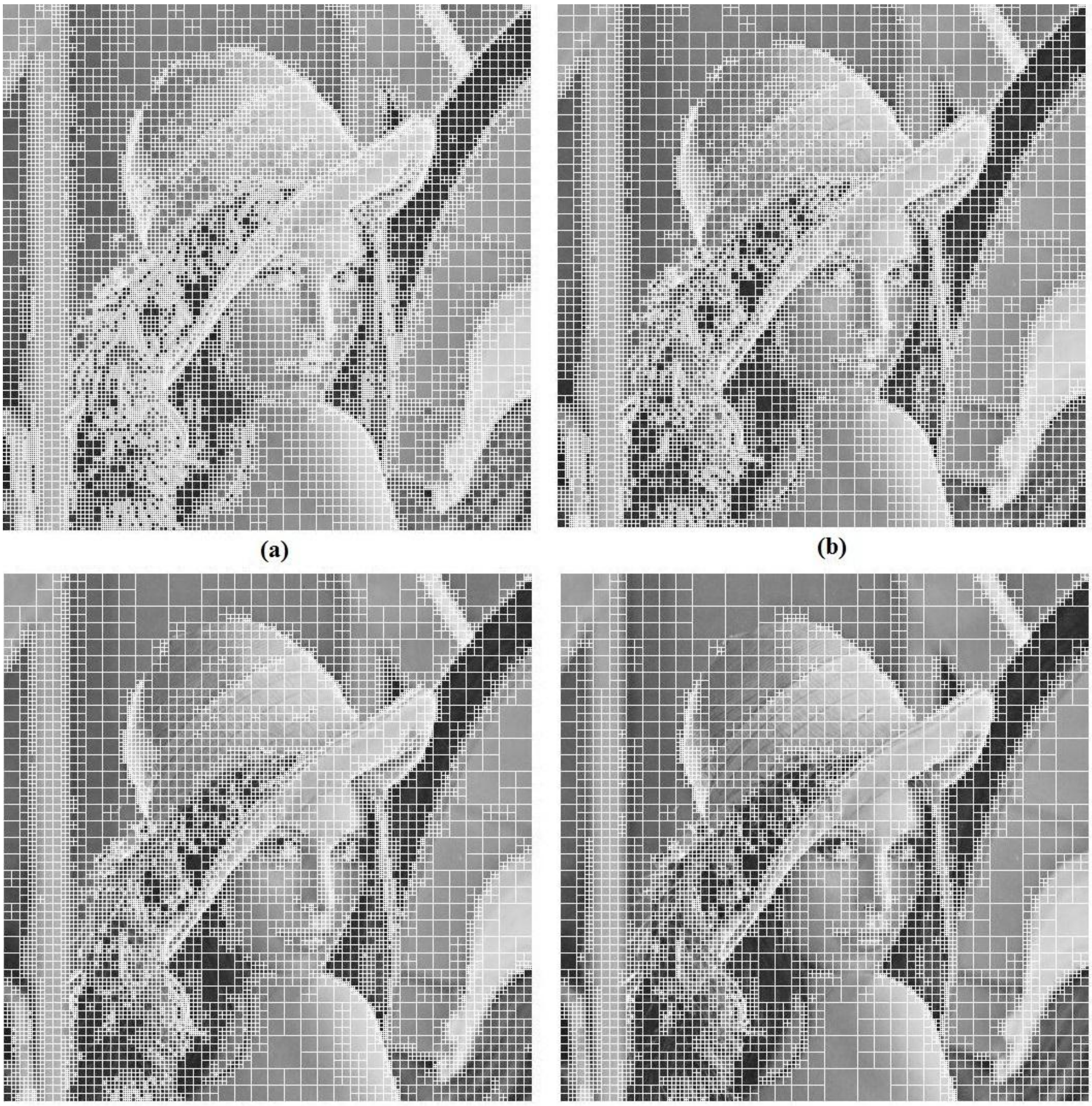


Figure 4.7: K level Quad-Tree

4.3.4 MRT based Adaptive block size Transform Coding Technique (MATC)

Transform coding has been proven to be a promising technique to achieve image data compression. As real images often have inhomogeneous statistics over different areas, adaptation has to be integrated into a transform coding system for the greatest benefit. Usually, there are several parameters in a transform coding system which can be made adaptive to an image. Quantization step size, bit allocation, transform kernel, and the block size used to partition an image can be made adaptive. Here, a variable block size adaptive transform coding system based on Quad-Tree partitioning is proposed.



(a) (b)

(c) (d)

Figure 4.8: Quad-Tree Partitioned Lena image for different values of t_s (a) $t_s = 100$ (b) $t_s = 150$ (c) $t_s = 200$ and (d) $t_s = 300$.

The proposed MATC coding scheme is shown in Figure 4.9. In the proposed coder, initially, the block size is taken as the size of the image, N×N, itself and the minimum block size is taken as 2×2. All homogenous areas in the image are partitioned into larger blocks and inhomogeneous areas are partitioned into smaller blocks by applying the UMRT based criterion described in section 4.3.2. This process continues until the image is partitioned into homogeneous blocks or the minimum block size is reached. In other words, the maximum and minimum size of partitioned blocks may vary between N×N and 2×2, in powers of 2. Ideally, if the criterion can make appropriate decision at boundary where there is a change in the image statistics, the divided blocks should have only uniform change within it.

Each partitioned block is then transformed by UMRT of corresponding size. The UMRT coefficients, other than the DC coefficients, are subject to the threshold t_c to eliminate irrelevant information. The coefficients overcoming the threshold and their corresponding positions in the UMRT matrices are stored in two separate linear arrays. Then entropy coding is applied to these two linear arrays to obtain the compressed data.

The Decoding process of MATC is very simple. The received bit stream is entropy decoded using Arithmetic decoding and the coefficient array and position array are separated. The size and position of each sub-image is obtained from the Quad-Tree data structure. The coefficients corresponds to each sub-image are placed in a matrix, from the coefficient array according to the positions in the position array. The output sub-image is produced by taking the Inverse UMRT (IUMRT) of the coefficient matrix.

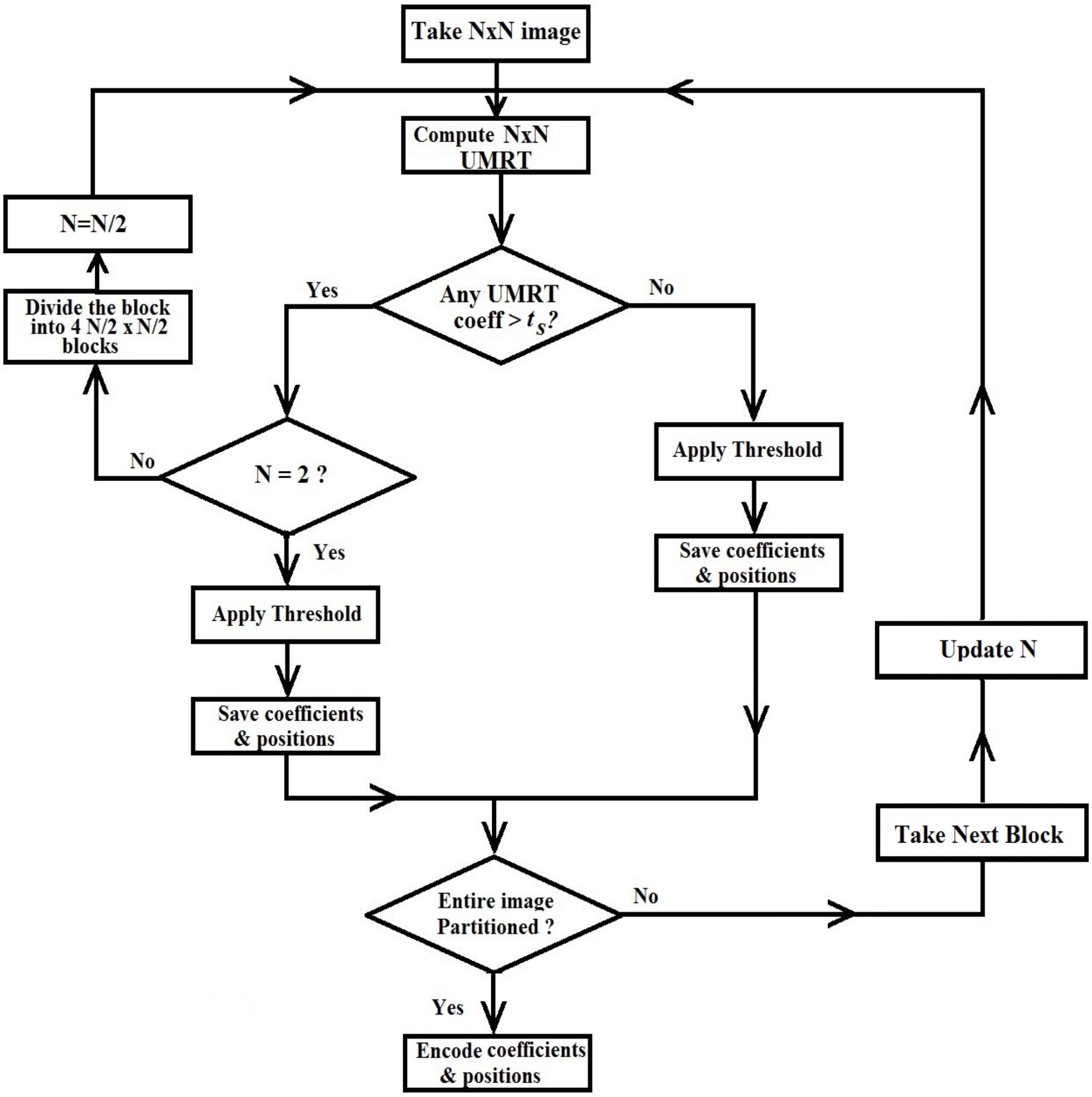

Take NxN image
Compute NxN UMRT
Any UMRT coeff > t_s?
Yes
No
N=N/2
Divide the block into 4 N/2 x N/2 blocks
No
N = 2 ?
Yes
Apply Threshold
Save coefficients & positions
Apply Threshold
Save coefficients & positions
Update N
Take Next Block
Entire image Partitioned ?
No
Yes
Encode coefficients & positions

Figure 4.9: Flow chart of MATC technique

To reconstruct the original image, the decoding algorithm has to know the block size used to encode different part of an image. For that, the hierarchical data structure, Quad-Tree, is used. In the variable block size transform coding system described, the number of levels in the Quad-Tree depends on the size of the image N×N, and the homogeneity of the image.

4.3.5 Simulation Results

A computer simulation is carried out to find the performance of the proposed coder. The input image has resolution of 8 bits per pixel and the image size is 512×512. The largest & smallest block sizes allowed are the size of the image itself and 2×2 respectively. The segmentation threshold t_s is chosen as 100 as in section 4.3.2. The quantized coefficients are entropy coded using Arithmetic encoding. Table 4.5 shows the simulation results for various gray scale images.

Table 4.5: Performance of MATC

Image (512×512)	bpp	PSNR(dB)
Lena	0.31	30.27
Baboon	0.85	24.84
Barbara	0.61	27.08
Goldhill	0.42	28.96
Peppers	0.32	30.49
Couple	0.51	29.25
Elaine	0.31	29.93
Cameraman	0.30	31.03
Boat	0.49	28.75
Bridge and Stream	0.75	25.88

Figure 4.10 to 4.12 show original and reconstructed versions of *Lena, Baboon* and *Cameraman* images.

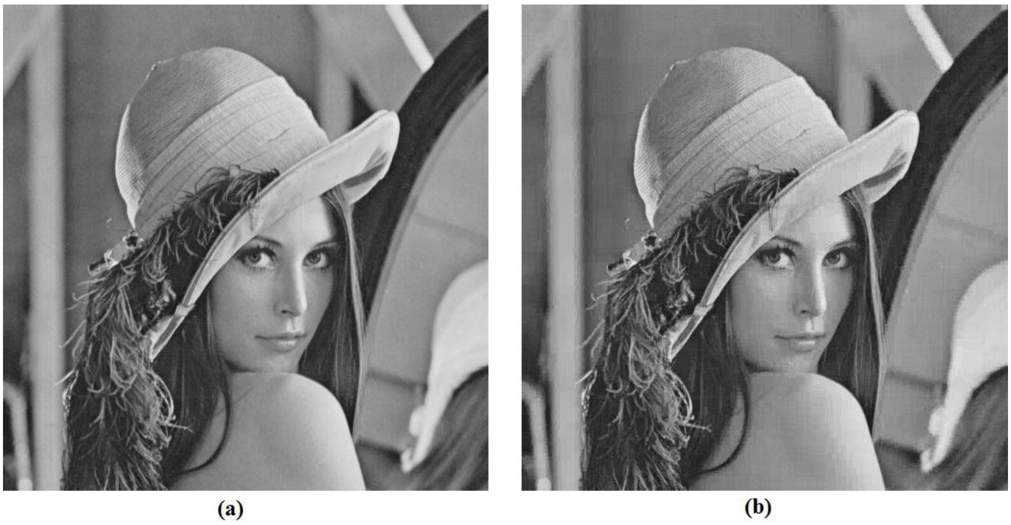


Figure 4.10: Lena (a) Original and (b) reconstructed after compressed using MATC coder (bpp =0.31, PSNR =30.27)

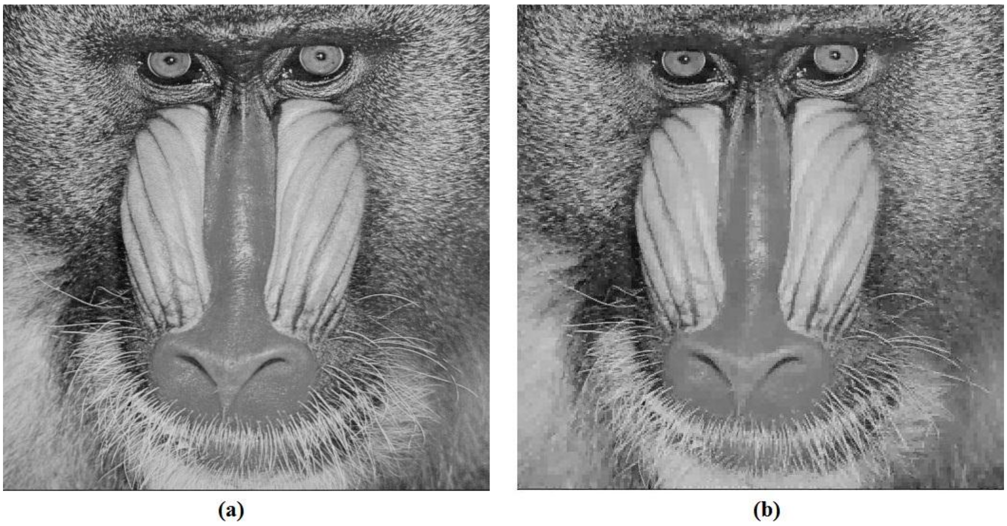
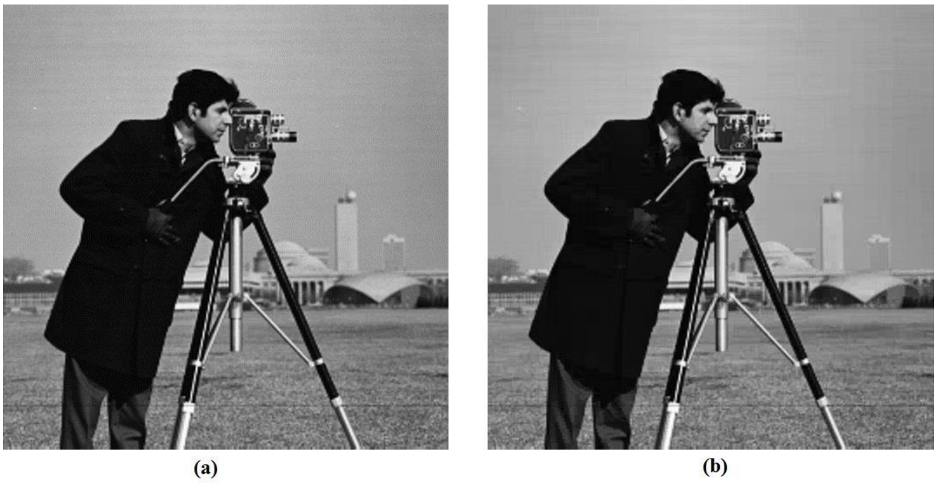


Figure 4.11: Baboon (a) Original and (b) reconstructed after compressed using MATC coder (bpp =0.85, PSNR = 24.84)


(a)

(b)

Figure 4.12: Cameraman (a) Original and (b) reconstructed after compressed using MATC coder (bpp =0.30, PSNR = 31.03).

4.3.6 Performance comparison

Table 4.6 shows a comparison between the performances of the proposed adaptive block size coder using UMRT and fixed block size coders using 8×8 MRT and 4×4 MRT. It is evident that the proposed method gives better performances at all bit rates. At lower bit rate (0.3 bpp), the PSNR values of the reconstructed images using the proposed coder are almost 2dB more than that of the fixed block size methods.

The Table.4.7 shows a comparison of the proposed coder with the DCT and DWT (Discrete Wavelet Transform) based transform coding schemes [131]. The table shows that the proposed coder also produces almost the same PSNR for a compression ratio of 10:1. Here the advantage of the proposed coder is that the UMRT can be computed using real additions only, compared to additions and multiplications required for DCT and DWT.

Table 4.6: Comparison of MATC with fixed block size transform coders

Image	PSNR(dB)								
	0.3 bpp			0.5 bpp			0.75 bpp		
	8×8	4×4	MATC	8×8	4×4	MATC	8×8	4×4	MATC
Lena	27.68	27.05	30.27	30.18	30.27	31.01	32.03	32.72	33.08
Peppers	27.43	26.52	30.49	29.86	30.41	31.88	31.55	32.71	33.40
Goldhill	27.25	26.68	27.42	29.04	29.16	29.73	30.34	31.13	31.20
Couple	26.47	25.87	25.94	28.82	29.16	29.25	30.86	31.70	31.90
Cameraman	28.55	26.79	31.03	31.20	31.47	32.45	33.35	34.25	34.85
Boat	25.94	25.19	25.79	27.88	28.29	28.76	29.48	30.39	30.55
Elaine	28.46	28.37	29.93	29.93	30.60	30.71	31.15	31.57	31.60

Table 4.7: Comparison of MATC with DCT and DWT (Embeddd Zerotree Wavelet) based coders

Image	Compression Ratio	PSNR(dB)		
		DCT (Baseline)	DWT (EZW)	MRT (MATC)
Lena	10:1	32.90	32.51	33.24
Peppers	10:1	34.30	34.43	33.82
Baboon	10:1	25.30	24.91	24.84

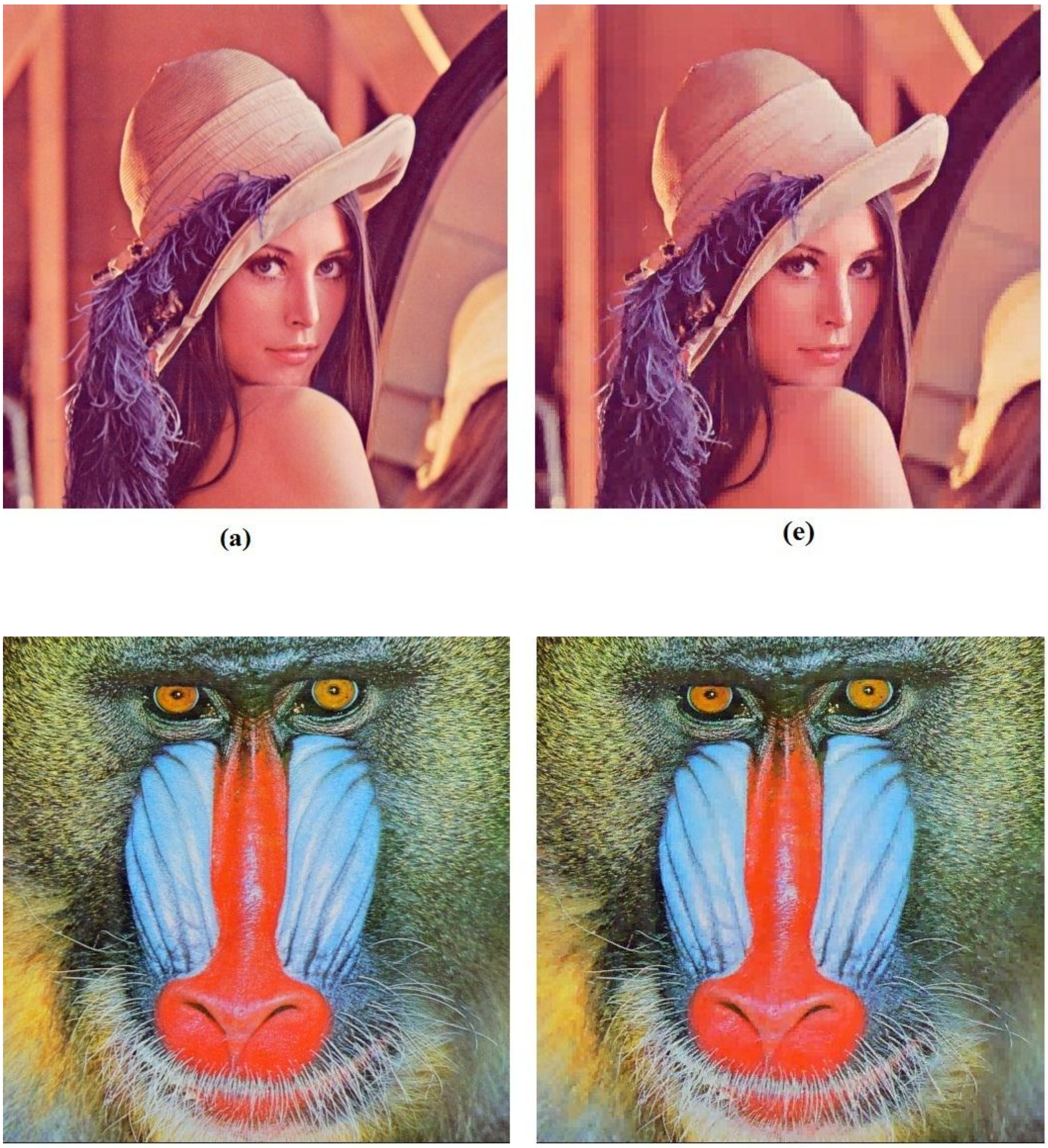
4.3.7 Application in Color Images

The proposed algorithm is applied on various color images (RGB model) and the results are shown in Table 4.8.

Table 4.8: Performance of MATC in color images

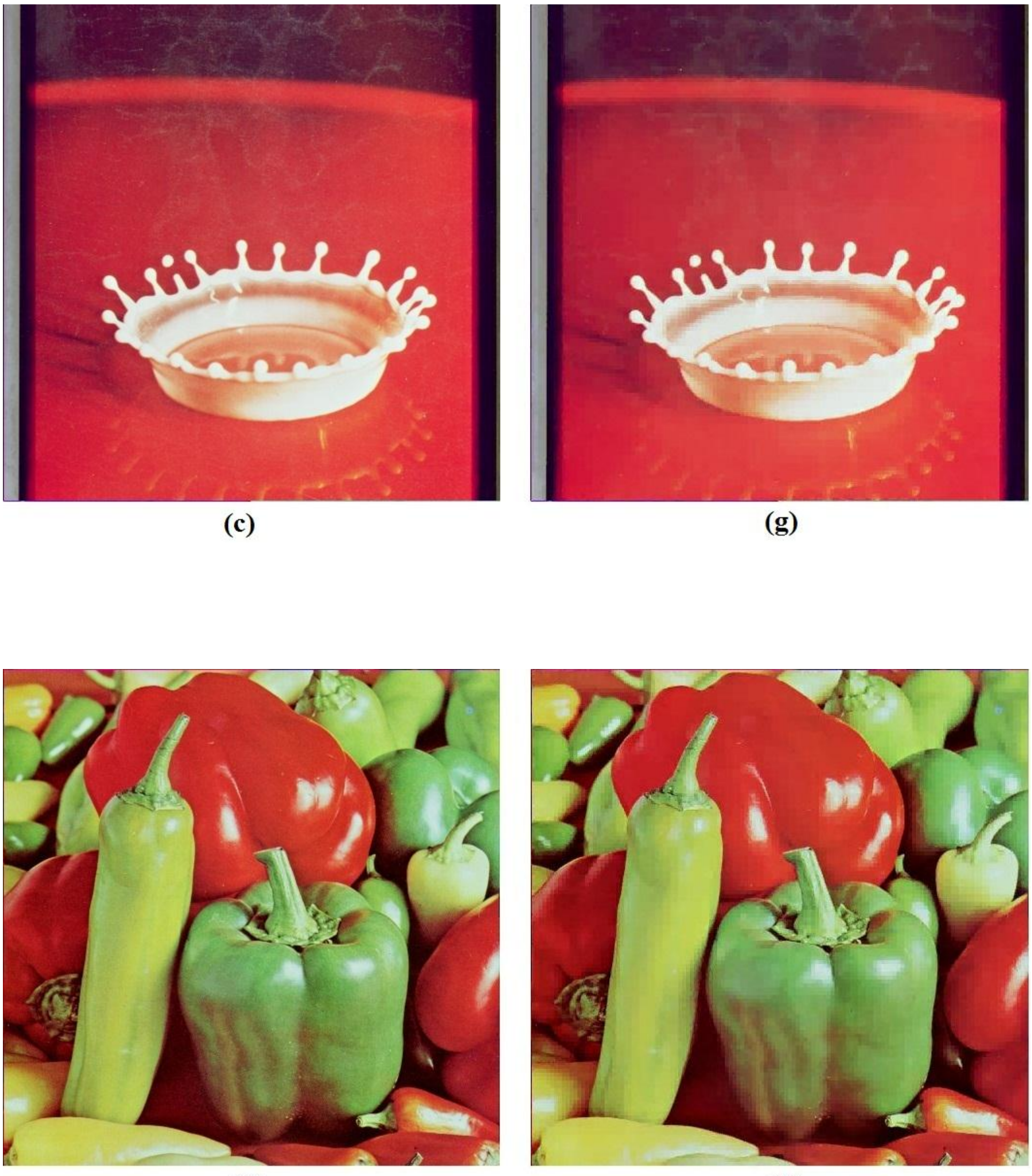
Image	bpp	PSNR
Lena	1.45	32.94
Crown	0.89	33.97
Baboon	3.80	26.03
Peppers	1.49	31.64

Figure 4.13 shows the original and decompressed versions of various color images using the proposed MATC algorithm.



(a) (e)

(b) (f)



(c) (g)

(d) (h)

Figure 4.13: (a) – (d) Original Color images and (e) – (h) their reconstructed versions after compressed using MATC coder

5

MRT based Adaptive Transform Coder with Classified Vector Quantization (MATC-CVQ)

Contents

5.1 Introduction....128
5.2 Vector Quantization in MRT Domain Using Isometric Transformations and Scaling....130
    5.2.1 Isometric transformations....130
    5.2.2 Isometric transformations in MRT domain....132
    5.2.3 Codebook generation in MRT domain....134
    5.2.4 Encoding and Decoding process....136
    5.2.5 Results of Compression....138
5.3 MRT based Adaptive Transform coder with Classified Vector quantization (MATC-CVQ)....141
    5.3.1 Classified Vector Quantization (CVQ) of images....143
        5.3.1.1 UMRT based Edge identification....144
        5.3.1.2 UMRT based Classification of Edge Blocks....149
        5.3.1.3 CVQ Codebook generation....150
    5.3.2 Encoding & Decoding Process of MATC-CVQ....152
    5.3.3 Simulation results....153
    5.3.4 Performance comparison....157
    5.3.5 Application in Color images....160

5.1 Introduction

Vector Quantization (VQ) is an efficient and simple approach for data compression. Since it is simple and easy to implement, VQ has been widely used in different applications, such as pattern recognition, image compression, speech recognition, face detection and so on. For the purpose of image compression, the operations of VQ include dividing an image into, mostly non-overlapping, vectors (or blocks) and each vector is mapped to the codewords of a codebook to find its reproduction vector. The basic VQ system is shown in Figure 5.1.

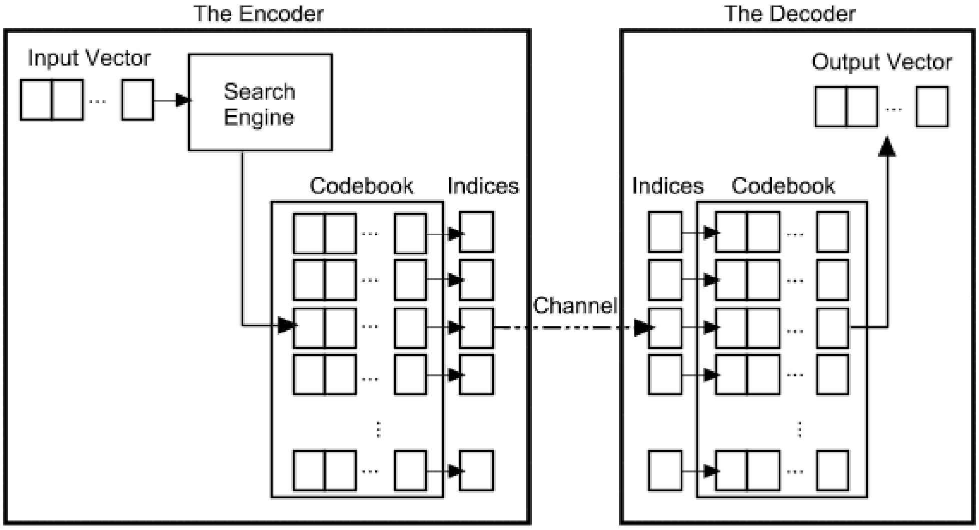


Figure 5.1: Standard Vector Quantization scheme

VQ provides many attractive features in applications where high compression ratios are desired. In these applications, the performance objective is good subjective visual quality rather than an accurate match between the original and coded images. One unique feature of VQ is that high compression ratios are possible with relatively small block sizes, unlike other compression techniques such as transform coding. Use of smaller block sizes in block coding has been known to lead to better subjective quality. A second feature of VQ is that the decoder is very simple to implement, making VQ attractive for single-encoder, multiple- decoder applications such as videotext and archiving.

There are three major procedures in VQ, namely codebook generation, encoding procedure and decoding procedure. In the codebook generation process, various images are divided into several k-dimension training vectors. The representative codebook is generated from these training vectors by the clustering techniques shown in Figure 5.2.

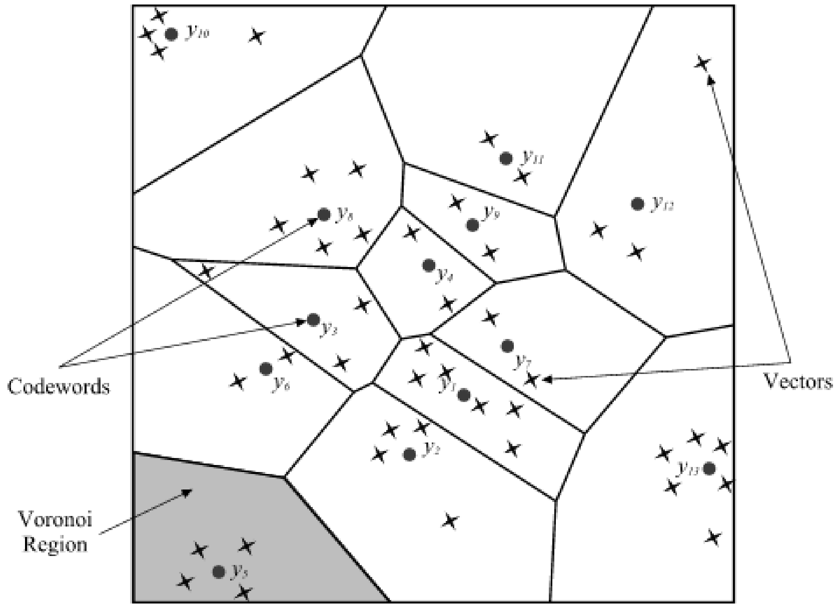


Figure 5.2: Codewords in 2-D space.

In the encoding procedure, an original image is divided into several k-dimension vectors and each vector is encoded by the index of codeword by a table look-up method. The encoded results are called an index table. During the decoding procedure, the receiver uses the same codebook to translate the index back to its corresponding codeword for reconstructing the image. In order to limit the scope of investigation, only still gray scale images are used. However, VQ can easily be extended to compression of colour images and image sequences

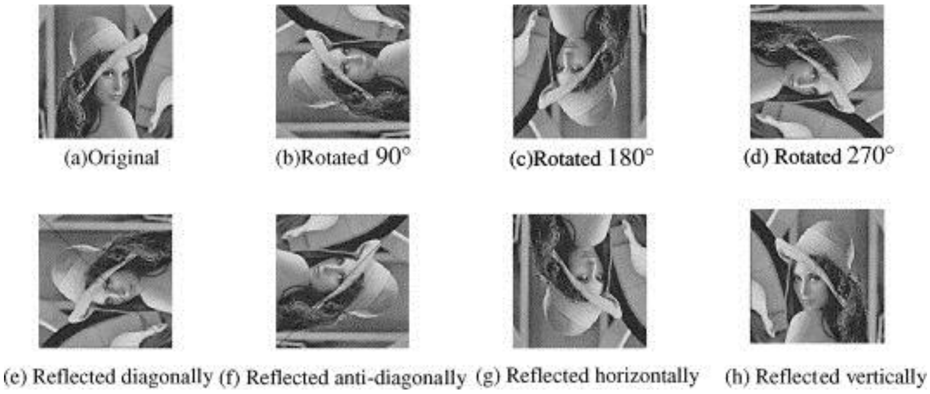
One of the key points of VQ is to generate a good codebook such that the distortion between the original image and the reconstructed image is the minimum. Moreover, since the codebook generation procedure is a time consuming process, how to reduce the computation time is another important issue for the VQ codebook generation. The most commonly used method in VQ is the Generalized Lloyd Algorithm (GLA) which is also called Linde-Buzo-Gary (LBG) algorithm.

5.2 Vector Quantization in MRT Domain Using Isometric Transformations and Scaling

5.2.1 Isometric transformations

A transformation is a process which changes the position, and possibly the size and orientation, of a shape. There are four types of transformations: reflection, rotation, translation and enlargement. Translation or Slide moves a shape by sliding it up, down, sideways or diagonally, without turning it or making it bigger / smaller. Reflection or Flip about a line produces a mirror image in which corresponding points on the original shape and the mirror image are always at the same distance from the mirror line. Rotation turns a shape through a clockwise or anti-clockwise angle about a fixed point known as the Centre of Rotation. All lines in the shape rotate through the same angle. Rotation, just like reflection, changes the orientation and position of the shape, but everything else stays the same. Enlargement or Dilation is a transformation that changes the size of the object.

Some of these transformations on an image give a new image, whose dimensions are different from that of the original image. Others produce an image whose dimensions are the same as those of the object. In other words the object and the image are invariant. Transformations which leave the dimensions of the object and its image unchanged are called isometric transformations. Examples of isometrics are reflection, rotation and translation. Transformations which do alter the dimension of the object when they act on them are called non-isometric transformation. Example is the enlargement. Figure 5.3 shows the original and seven isometric transformations of *Lena* image.



(a)Original (b)Rotated 90° (c)Rotated 180° (d) Rotated 270°

(e) Reflected diagonally (f) Reflected anti-diagonally (g) Reflected horizontally (h) Reflected vertically

Figure 5.3: 8 isometric transformations of Lena image

5.2.2 Isometric transformations in MRT domain

The above mentioned isometric transformations are usually implemented in spatial domain. Since the proposed Vector Quantization is implemented in MRT/UMRT domain, the isometric transformations also need to be implemented in the same domain in order to avoid unnecessary computational overheads.

If $Y_{k1,k2}^{(P)}$, $0 \le k1,\ k2 \le N-1$ and $0 \le p \le N/2 -1$, is the MRT coefficients of the input data matrix $x_{n1,n2}$, $0 \le n1,\ n2 \le N-1$, Then the following relations can be used to obtain the isometric transformations of $x_{n1,n2}$ in the MRT domain

Reflection about Main Diagonal line

$$Ynew_{k1,k2}^{(P)} = {Y_{k1,k2}^{(P)}}^{T} \tag{5.1}$$

90^0 Rotation

$$Ynew_{k1,k2}^{(P)} = S.Y_{((N-k2))_N,\ k1}^{((k2+P))_{N/2}} \tag{5.2}$$

where
$$S = \begin{cases} -1, & \frac{N}{2} \le (P+k2) \le N-1 \\ 1, & otherwise \end{cases} \tag{5.3}$$

Reflection about Mid Horizontal line

$$Ynew_{k1,k2}^{(P)} = S.Y_{((N-k1))_N,\ k2}^{((k1+P))_{N/2}} \tag{5.4}$$

where $$S = \begin{cases} -1, & \frac{N}{2} \le (P + k1) \le N - 1 \\ 1, & otherwise \end{cases} \quad (5.5)$$

Reflection about Mid Vertical line

$$Ynew_{k1,k2}^{(P)} = S.Y_{k1\,,\;((N-k2))_N}^{((k2+P))_{N/2}} \quad (5.6)$$

where $$S = \begin{cases} -1, & \frac{N}{2} \le (P + k2) \le N - 1 \\ 1, & otherwise \end{cases} \quad (5.7)$$

180^0 Rotation

The MRT matrices of the 180^0 rotated version of $x_{n1,n2}$ can be obtained by applying the relation for 90^0 rotation twice.

270^0 Rotation

The MRT matrices of the 270^0 rotated version of $x_{n1,n2}$ can be obtained by applying the relations for reflection about mid vertical line followed by reflection about main diagonal line.

Reflection about Off Diagonal line

The MRT matrices of the off diagonal reflected version of $x_{n1,n2}$ can be obtained by applying the relations for 180^0 rotation followed by reflection about main diagonal line.

5.2.3 Codebook generation in MRT domain

The objective of this codebook generation is to generate a universal codebook. The codebook generation process in the proposed VQ technique is different from the traditional codebook generation techniques like LBG algorithm. Here, instead of saving the codewords, the UMRT coefficients of the codewords are saved. Initially, a wide range of images are selected as training set. Here**,** the images from brodatz album, shown in Figure 5.4, are selected as training set. The first codeword in the codebook is the UMRT coefficients of a uniform 4×4 image block. Each image in the training set is taken in a sequential manner and segmented into 4×4 blocks. UMRT of each 4×4 block is computed and compared against all the codewords already in the codebook. The comparison is done using Euclidean distance measure in terms of UMRT coefficients. While comparing, each block in the training image is compared against all the eight isometric transformations of the already available codewords using the relations discussed in the section 5.2.2. If any of the blocks in the training set is identical to any of the codeword or any of its isometric transformations, then that block will not be added to the codebook. Also while comparing, the DC coefficient in the UMRT of the training block and codeword will not be considered because of the fact that, if all the corresponding UMRT coefficients, except the DC coefficient, of two blocks are identical, then that two blocks will be brightness shifted versions of each other. So, that training block is not necessary for the codebook. In such a manner all the images in the training set are compared and a universal codebook is generated as shown in Figure 5.5.

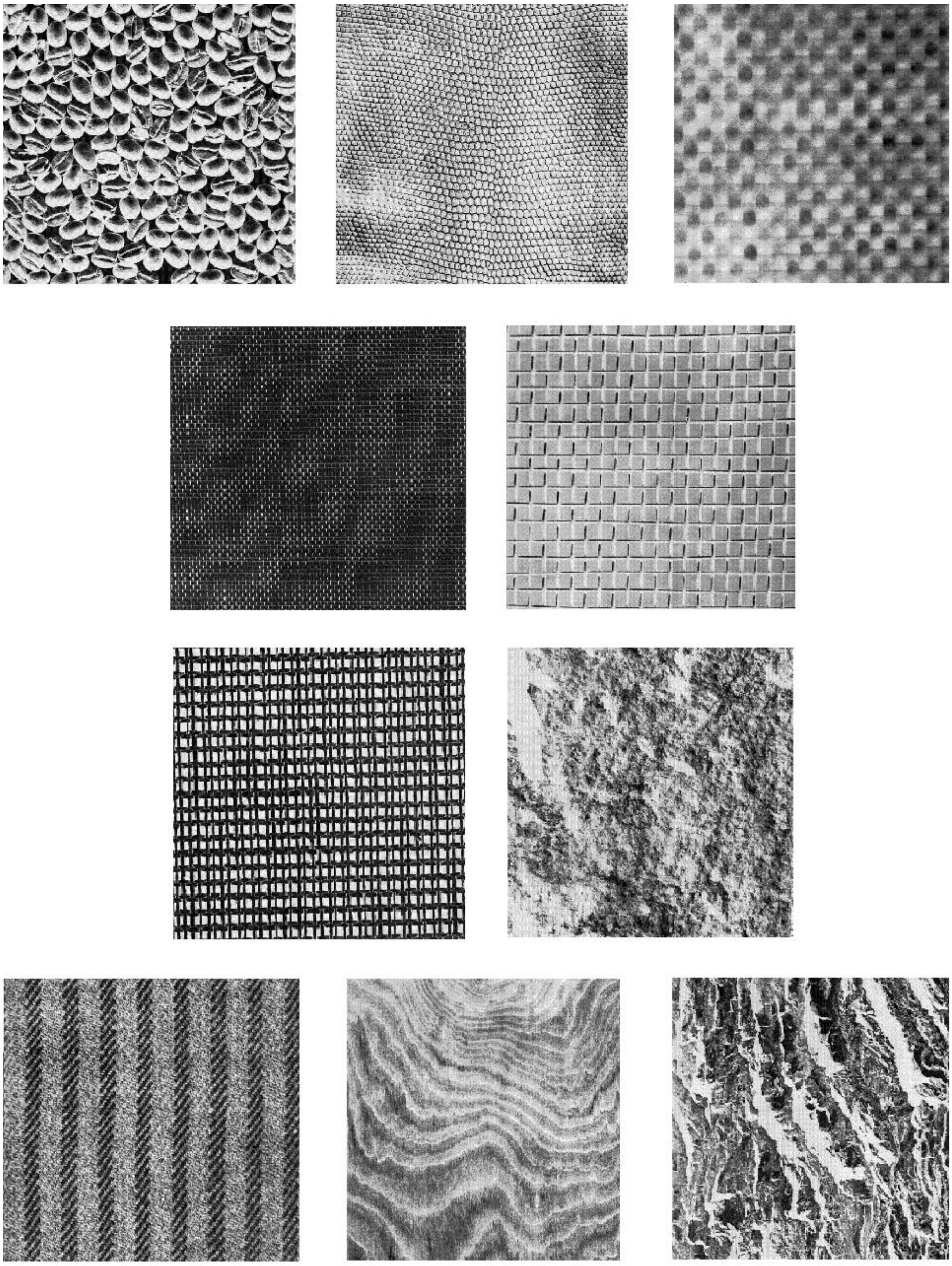


Figure 5.4: Images of Brodatz album used for Codebook Generation

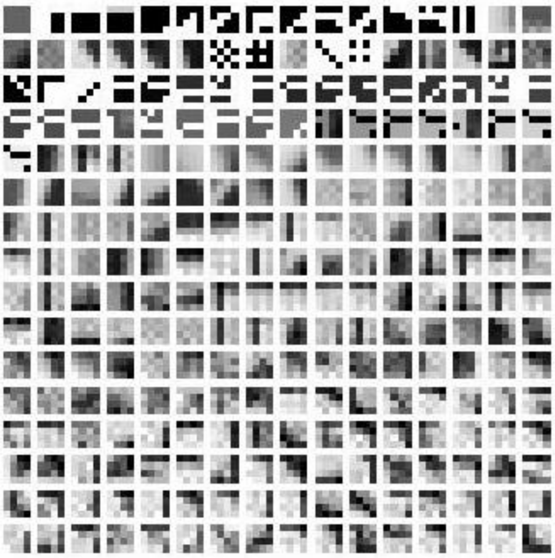


Figure 5.5: Generated Codebook

5.2.4 Encoding and decoding process

A vector quantization scheme is composed of two operations, encoding and decoding. The encoder takes an input vector and outputs the index of the codeword that offers the least distortion. The least distortion is found by evaluating the Euclidean distance between the input vector and each codeword in the codebook. Once the closest codeword is found, the index of that codeword is sent through a channel (the channel could be computer storage, communications channel, and so on). When the decoder receives the index of the codeword, it replaces the index with the associated codeword. Figure 5.1 shows the block diagram of the operation of the encoder and decoder.

The encoding procedure of the proposed scheme is shown in Figure 5.6. Before the encoding starts, the codebook is expanded by including the seven isometric transformations of all the codewords in the codebook. The image to be compressed is then partitioned in to 4×4 blocks and the expanded codebook is searched for best match for each 4×4 block in the image. For each image block, the index of the most suitable codeword, the isometric transformation used to obtain such a best match and the scaling factor for brightness adjustment are identified and stored. After the above process is applied to all the blocks in the image, the stored data are entropy coded and the output bit stream is either stored in memory or sent to the receiver.

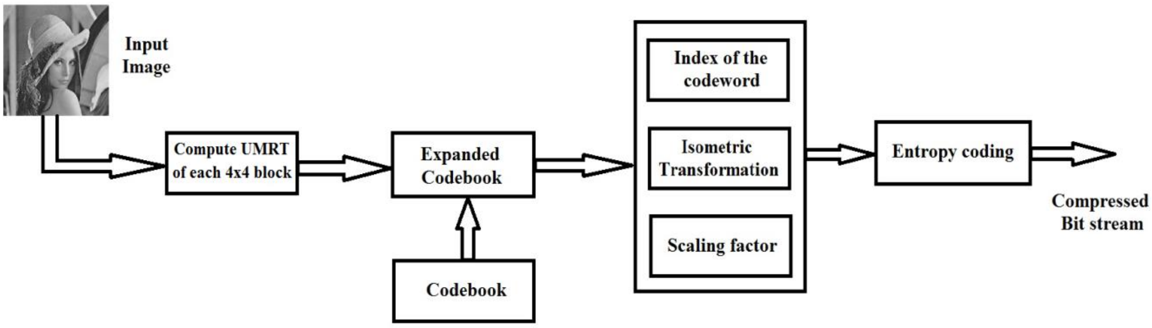


Figure 5.6: Encoding process of MRT based VQ

At the receiver, the reverse process occurs, as shown in Figure 5.7. The same codebook, which is used in the transmitter, is available at the receiver. The compressed bit stream is decoded using the entropy decoder and for each image block, the index in the codebook, isometric transform used and the scaling factor are identified and the blocks are reconstructed using the decoded information.

5.2.5 Results of Compression

The codebook is generated by selecting the images from Brodatz album as the training set. The objective of the proposed coder is to develop a universal codebook, such that any natural images can be compressed using the developed codebook.

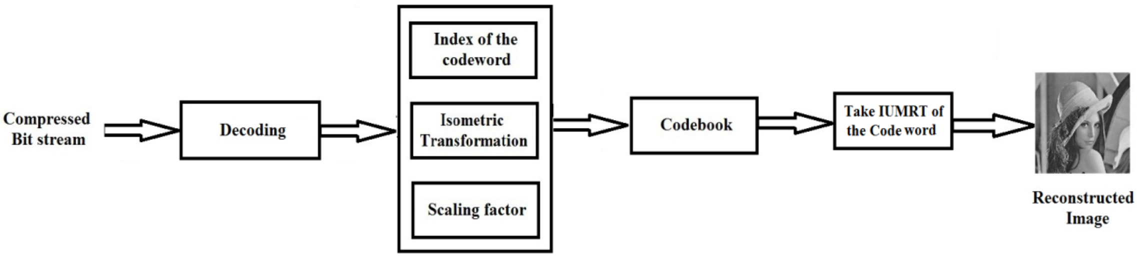


Figure 5.7: Decoding Process of MRT based VQ

The proposed VQ scheme is simulated and standard natural images are compressed using the generated universal codebook. Table 5.1 shows the results of compressing various images like *Lena, Cameraman, Baboon, Barbara, Goldhill etc.* using the generated codebook shown in Figure 5.5. Figure 5.8 shows the original and vector quantized version of these images with a codebook of size 256. Moderate compression and reconstructed image quality are obtained through the proposed scheme. The quality of the reconstructed images can be improved by expanding the codebook by increasing the training set.

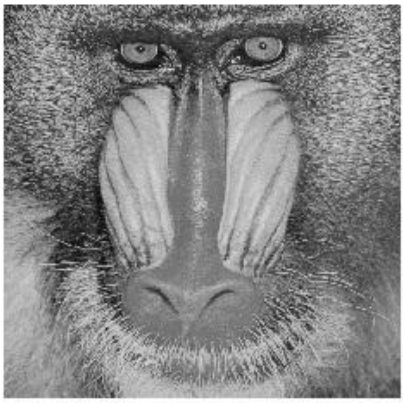
The main disadvantage of this method is the time taken for the codebook generation. The codebook shown in Figure 5.5 is generated from ten images of the brodatz album, shown in Figure 5.4. It took almost two hours for the codebook generation algorithm to settle down to the above mentioned codebook. Also, as the number of codewords increases, the time for encoding process increases drastically. Time taken by the proposed VQ scheme for compressing various standard gray scale images is shown in Table 5.1. The time is very high and as a result, such algorithms cannot be used in real time applications. This is the major drawback of vector quantization.

Table 5.1: Performance of MRT based VQ

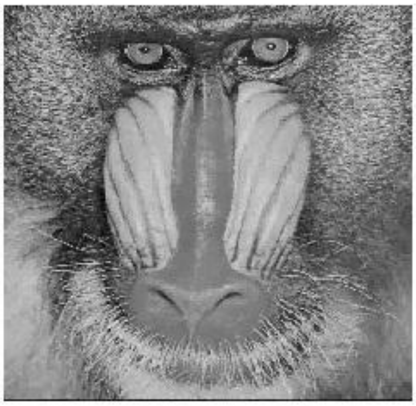
Image	bpp	PSNR (dB)	Compression Time (s)
Lena	0.3246	27.75	221
Baboon	0.3999	22.21	223
Barbara	0.3754	24.54	220
Goldhill	0.3138	26.89	201
Peppers	0.3251	27.84	205
Couple	0.3328	26.14	211
Elaine	0.3412	28.03	216
Cameraman	0.3477	28.32	208
Boat	0.3392	26.69	204
Bridge and Stream	0.3264	22.20	212

(a)

(f)



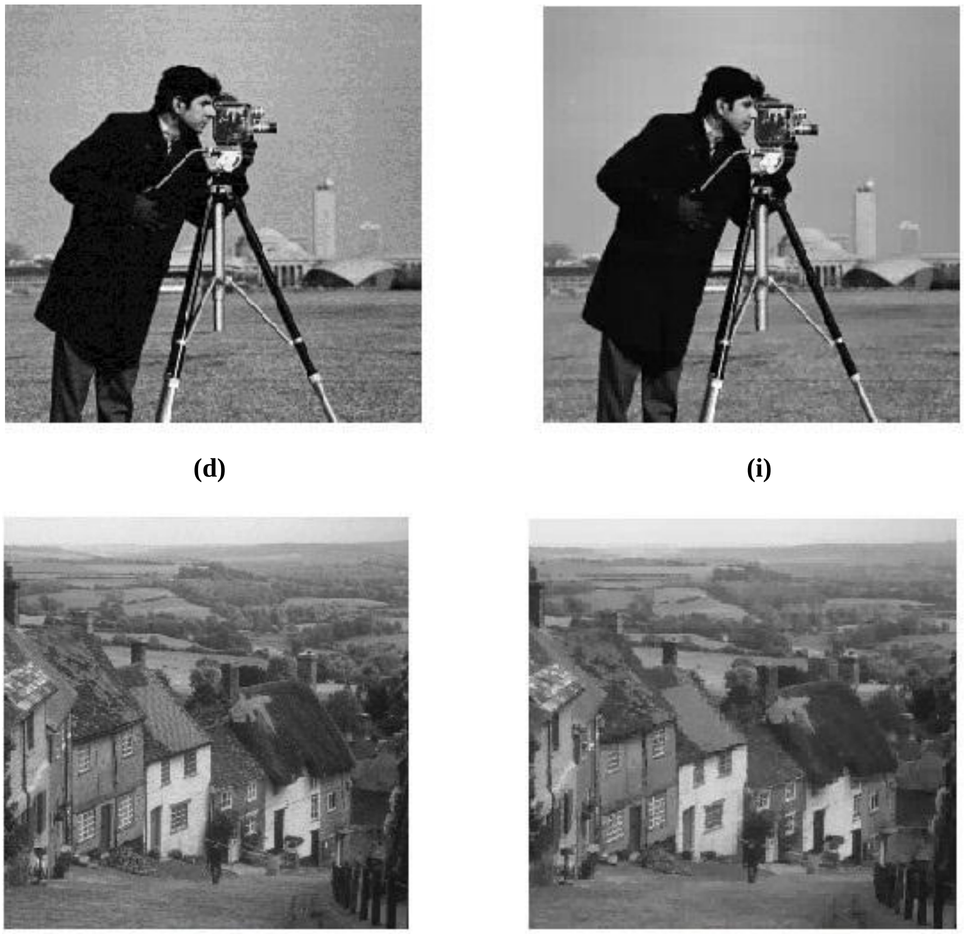
(b)



(g)

(c)

(h)



(d) (i)

(e) (j)

Figure 5.8: (a) – (e) Original images & (f) – (j) their Vector quantized versions

5.3 MRT based Adaptive Transform coder with Classified Vector Quantization (MATC-CVQ)

Vector quantization is a promising technique in many areas of digital signal processing, such as image coding, pattern recognition, speech analysis etc. One major problem for real-time application of vector quantization is its high computational complexity during encoding process. The computational complexity is O(kNM) for an exhaustive search VQ, with codebook size N, vector dimension k and M input vectors. This is too large for real-time image coding. Many constrained VQ techniques have been designed using sub-optimal codebooks, sacrificing accuracy, to reduce complexity. Another major problem in vector quantization is the edge degradation. Edges constitute significant portion of the information content in an image, and their degradation is visually very annoying. The Classified Vector Quantization (CVQ) [34], can be used to reduce both edge degradation and coding complexity.

Also, in most natural images, majority portions are of uniform nature and a minority area is of discontinuities (edge area). Vector quantization of such images is a waste of time. These uniform areas in images can easily be coded using transform coding techniques. Vector quantization is better for those edge areas. So, a combination of these two techniques will serve better for most of the natural images.

Hence, a new MRT based Adaptive block-size Transform Coder with Classified Vector Quantization (MATC-CVQ) scheme is developed for image compression. Quad-Tree segmentation is employed to generate blocks of variable size according to their uniformity. Uniform areas in images are coded using transform coding, while edge areas are coded by an edge-classified VQ to avoid edge degradation. In this new algorithm, initially the image to be compressed is subject to an edge identification procedure using UMRT and then the identified edge blocks are vector quantized using a UMRT based Classified Vector Quantization (UCVQ) algorithm. Both the segmentation and edge classification are determined by the strength of the UMRT coefficients.

5.3.1 Classified Vector Quantization (CVQ) of images

Initial studies of image coding with VQ have revealed several key problems. The major problems are edge degradation and high computation time. When conventional distortion measures such as the mean squared error (MSE) are employed, the edges are severely degraded. Edges constitute significant portion of the information content of an image, and their degradation is visually very annoying. The complexity of VQ grows exponentially with the codebook size and the code vector size. The edge degradation and coding complexity can be reduced through CVQ.

The crucial feature of edge perception is that it must appear essentially as sharp, continuous, and well defined in the coded image as in the original. In short, edge integrity must be preserved. The CVQ enables to preserve perceptual features while employing distortion measures such as MSE. The general structure of a CVQ coder is shown in Figure 5.9. CVQ is best understood in terms of a composite source model for images, where the image source is viewed as a bank of vector sub-sources. At each instance, a switch selects one of the sub-codebooks whose output becomes the output of the coder. The sub-codebook selection is based on a specified measure. Each sub-codebook contains blocks of a single perceptual class, viz. blocks with an edge at a particular orientation and location. In order to preserve the perceptual feature associated with each class, blocks belonging to one class are to be coded with code vectors belonging to the corresponding sub-codebook. A classifier is employed to identify the most suited sub-codebook. A distortion measure is then used to pick the best code vector from the sub-codebook for that class. The task of the distortion measure is simply that of matching the intensity levels as desired for that particular class, and also perhaps to perform some “fine tuning” in the edge-matching process necessitated by the restriction on the number of edge classes.

The main hurdles involved in the design of the CVQ technique are the identification of edges present in an image and the classification of these edges. Here, a UMRT based criterion is proposed for both edge identification and edge classification.

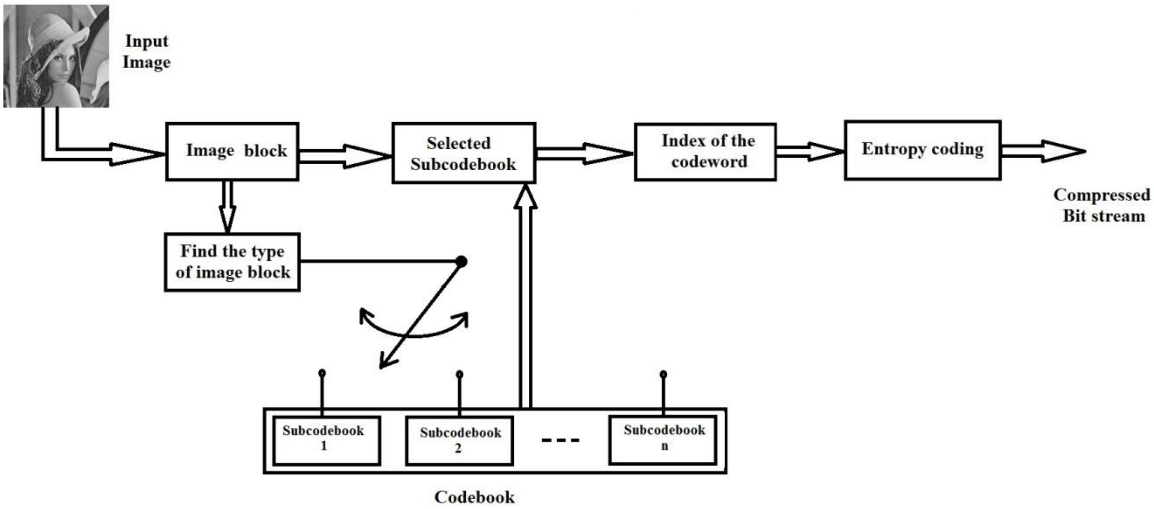


Figure 5.9: Block diagram of Classified Vector Quantization scheme

5.3.1.1 UMRT based Edge identification

Edge is the most important feature of an image. It contains much of the information in an image. Therefore, edge identification has a significant influence on the performance of image compression techniques. Edges are created by occlusions, shadows, highlights, roofs, etc. They have very different local intensity profiles. Edges within an image correspond to intensity discontinuities that result from different surface reflectance of objects, various illumination conditions, or varying distance and orientations of objects from a viewer. Therefore, the analysis based on edge can provide theoretic gist for image restoration, enhancement and reconstruction. Significant intensity changes in an image normally occur at different spatial resolutions or scales. Here, a new UMRT based edge detection technique is proposed.

Since UMRT coefficients are computed from addition and subtraction of selected image data, each UMRT coefficient signifies a particular manner of combination of image data under consideration. Consequently, each UMRT coefficient signifies the presence of a unique pattern in the image domain. Thus, the N^2 UMRT coefficients of an N×N image block are, in fact, strength-indicators of N^2 different patterns in the image block. The UMRT thus becomes a suitable tool for image analysis to determine the existence of N^2 well defined patterns in an image. Thus, by examining the strength of these UMRT coefficients, the presence and location of the edges in the image blocks can be identified. The edges present in images can be plotted using the following algorithm

Step 1: Partition the image using UMRT based Quad-Tree segmentation criteria

Step 2: Replace all the higher sized (higher than 2×2) sub-images and uniform 2×2 sub-images with black sub-images of same size.

Step 3: Replace all non-uniform 2×2 sub-images with the appropriate sub-images shown in Figure 5.10 by examining the magnitude & sign of their UMRT coefficients

The Figure 5.11 shows the identified edge locations of various standard gray scale images using UMRT based edge identification.

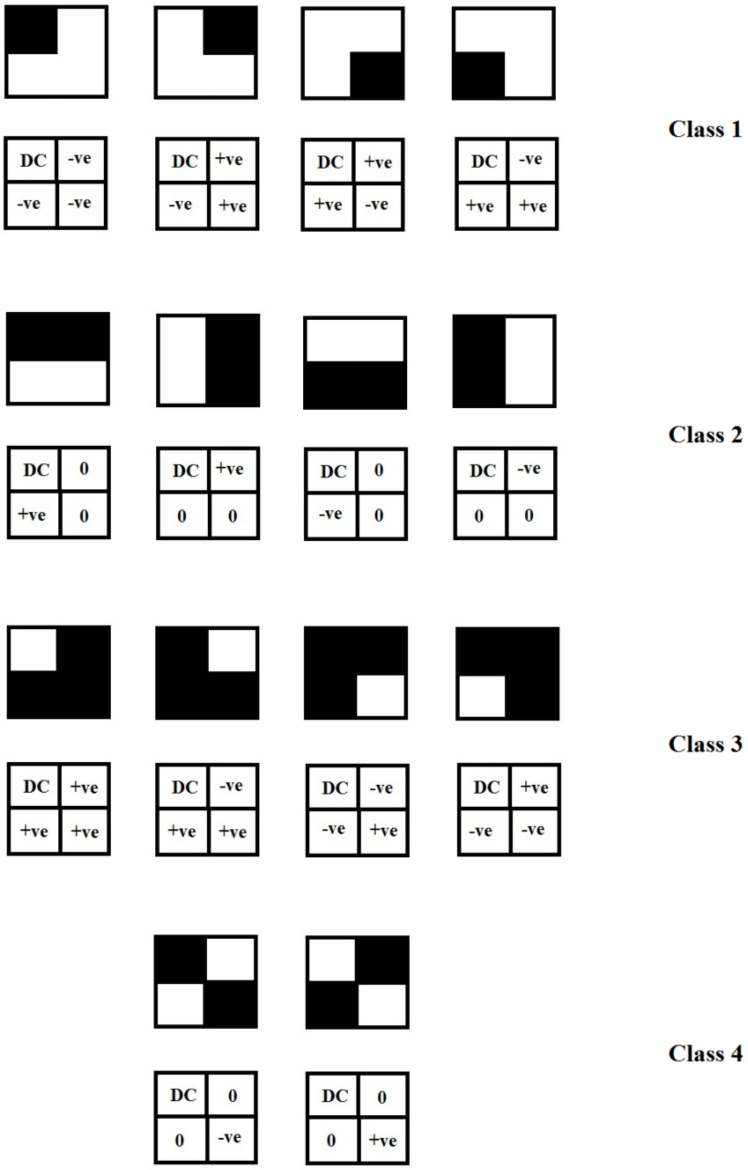
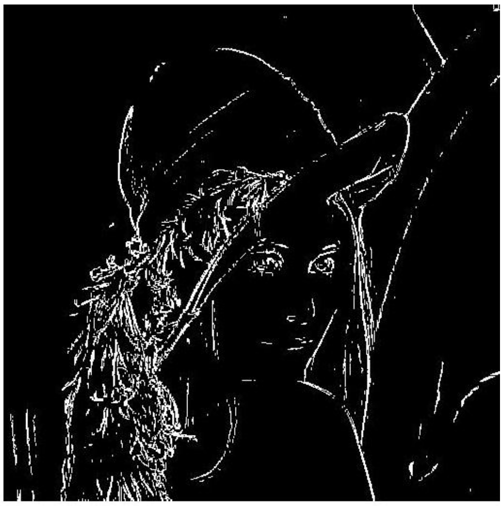
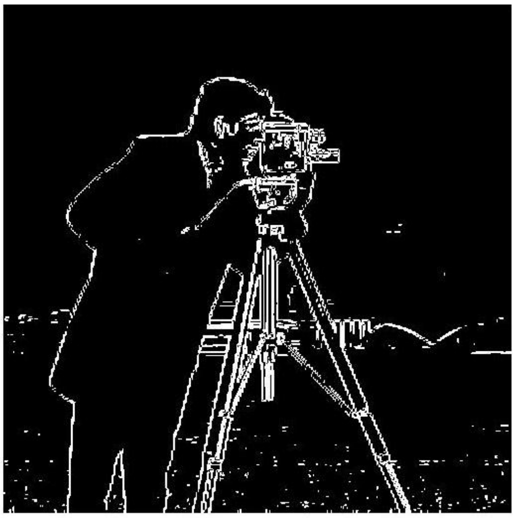


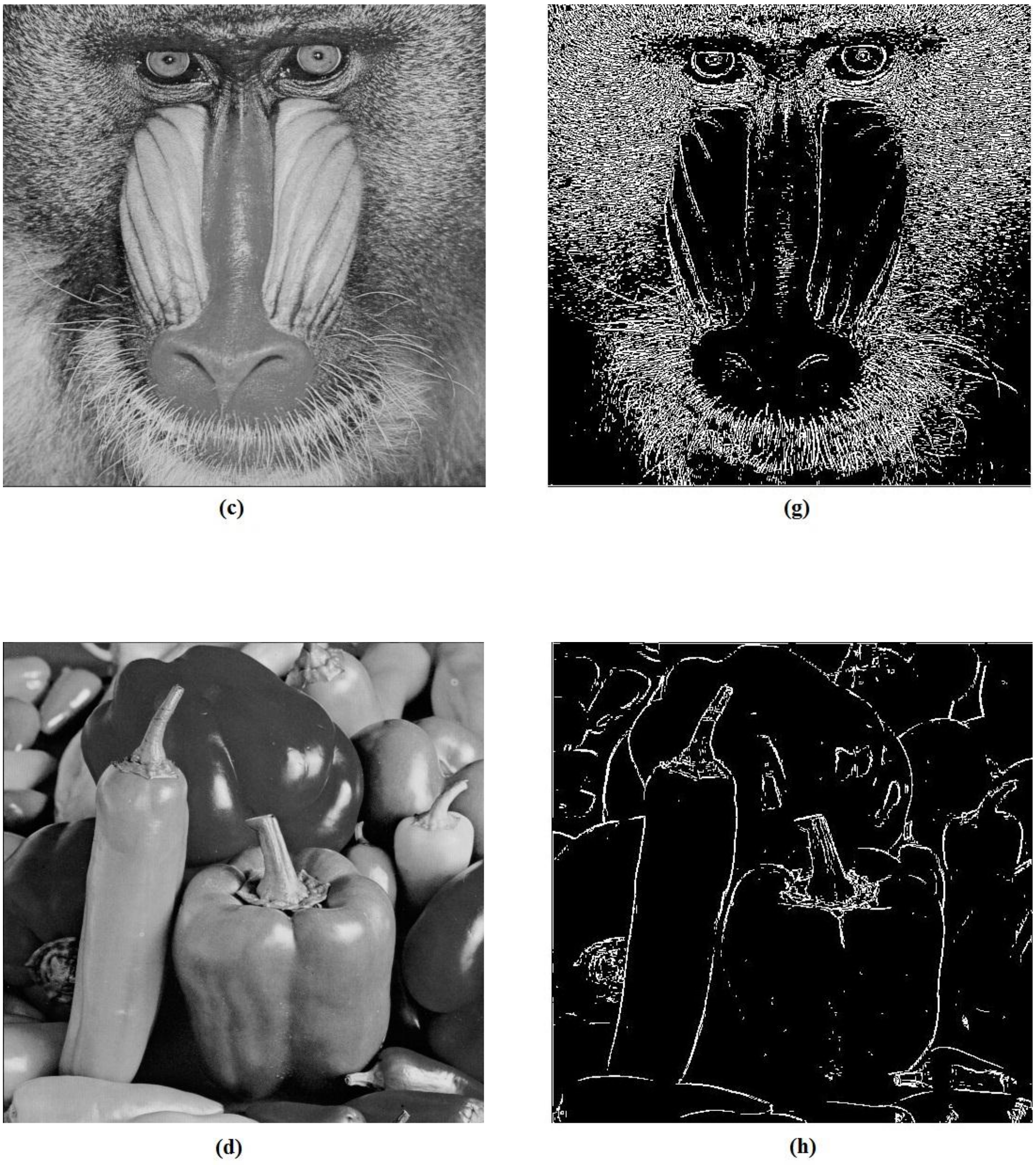
Figure 5.10: Types of 2×2 image blocks and their respective UMRT coefficients

(a)


(e)

(b)


(f)



(c) (g)

(d) (h)

Figure 5.11: (a) – (d) Original images and (e) – (h) their detected edges

The edge blocks identified by the UMRT based edge identification criterion, from the *Lena* image, is shown in Figure 5.12. These edge blocks can be coded using classified vector quantization and other uniform blocks can be coded using adaptive block size transform coder to get better reconstructed images, avoiding edge degradation.

5.3.1.2 UMRT based Classification of Edge Blocks

Different types of edges that can be present in a 2×2 image block are shown in Figure 5.10. There are fourteen types of edges that are possible to occur in 2×2 image blocks. The UMRT is an efficient tool to classify image blocks into these categories. The sign of certain UMRT coefficients along with the position of zeros determine the class of the image block. The 14 classes of 2×2 edge blocks and the respective UMRT coefficients are shown in Figure 5.10. By identifying the sign and magnitude of the UMRT coefficients, the edges can be classified.

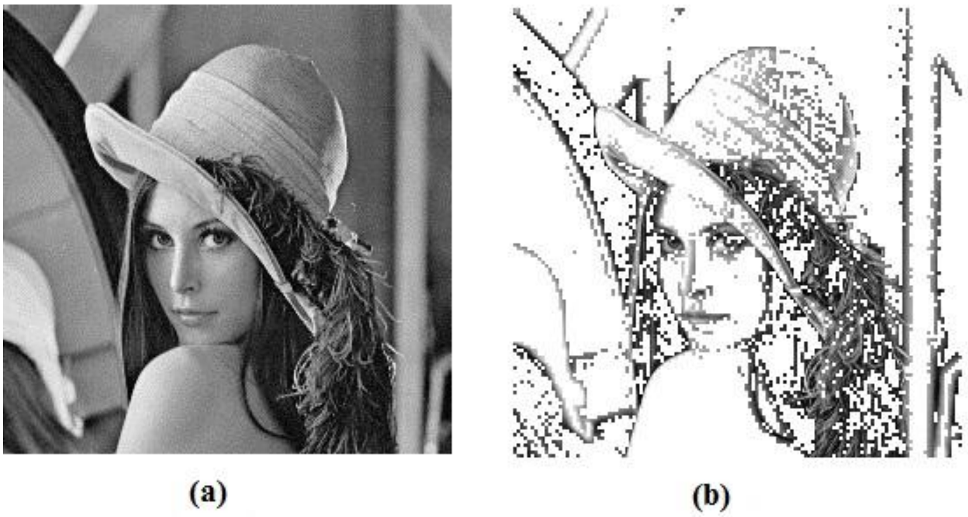

(a) (b)

Figure 5.12: (a) Original Lena image and (b) Edge blocks of Lena image

5.3.1.3 CVQ Codebook generation

Even though there are fourteen different types of edges that can appear in a 2×2 image block, as in Figure 5.10, only four types of codebooks needs to be generated. The remaining 10 types of codebooks can be obtained from the rotated version of the above mentioned four types. The four basic classes of the codebooks generated are shown in Figures 5.13 to 5.16.

The total number of code vectors $N = \sum_{i=1}^{M} N_i$, where M is the total number of types and N_i is the size of the corresponding codebook. The index of the closest code vector is transmitted to the decoder. The decoder simply retrieves the corresponding code vector from its codebooks, hence, generating the output

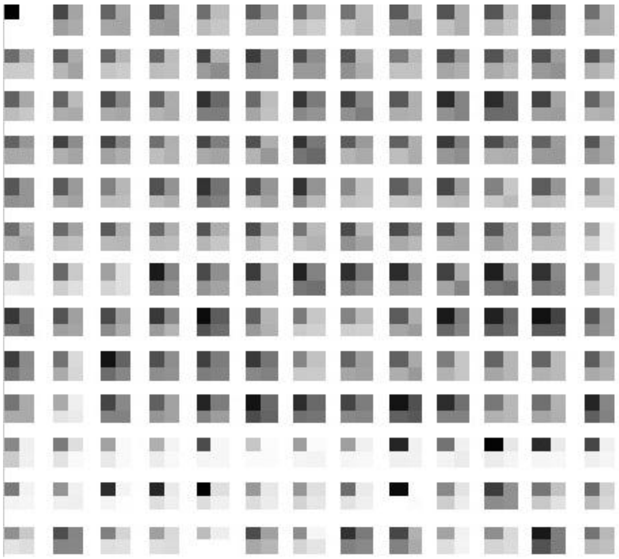


Figure 5.13: Codebook of Class 1

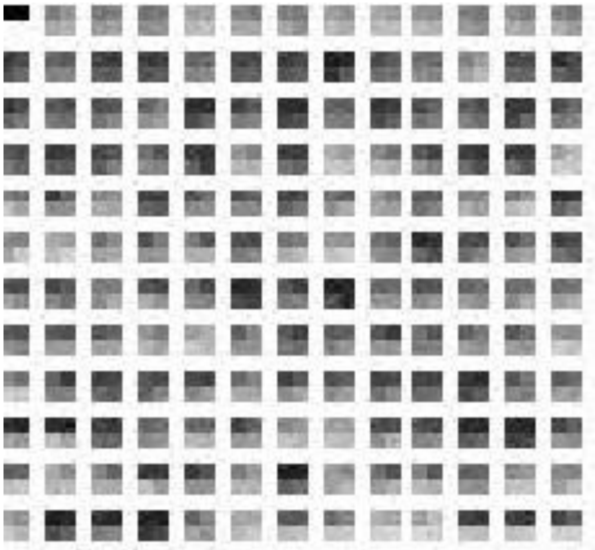


Figure 5.14: Codebook of Class 2

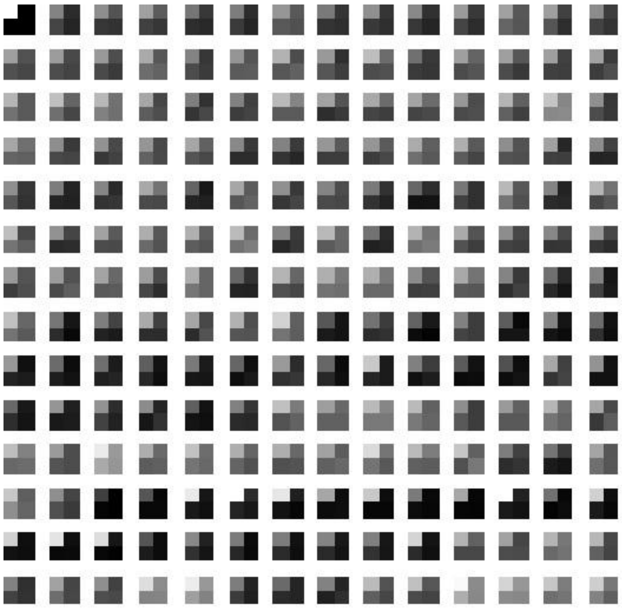


Figure 5.15: Codebook of Class 3

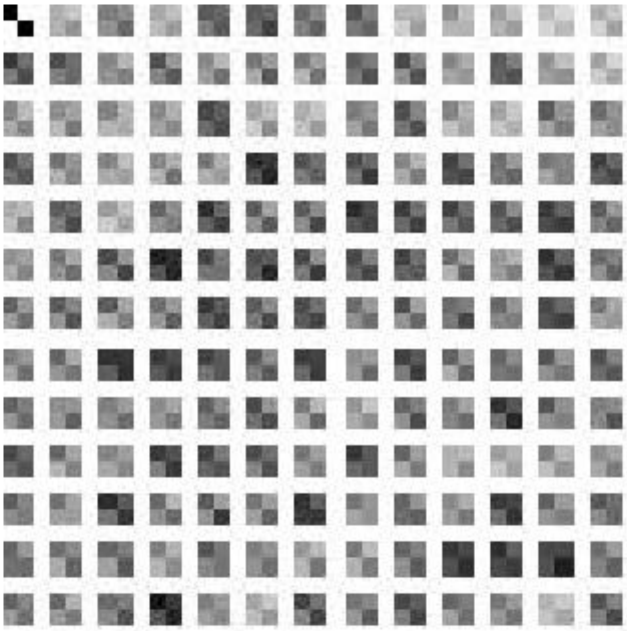


Figure 5.16: Codebook of Class 4

5.3.2 Encoding & Decoding Process of MATC-CVQ

The proposed coding scheme is shown in Figure 5.17. Here, initially, the block size is taken as the size of the image, N×N, and the minimum block size is taken as 2×2. All homogenous regions in the image are segmented into larger blocks and non-homogenous regions are segmented into smaller blocks by applying the UMRT based Quad-Tree partitioning scheme described in section 4.3.2. This process continues until the image is partitioned into homogeneous blocks or the minimum block size is reached. In other words, the maximum and minimum size of partitioned blocks can vary between N×N and 2×2 respectively.

Ideally, if the criterion can make appropriate decision at boundary where there is a change in the image statistics, the partitioned higher sized (larger than 2×2) blocks will be homogeneous in nature. These higher sized blocks are transform coded as explained in the section 4.3.4, because only a small amount of data are required to represent them. The 2×2 image blocks, obtained after the Quad-Tree decomposition process, are of different types. They may be either uniform blocks or edge blocks. The homogeneous 2×2 blocks are transform coded, just like other higher sized blocks. The non-homogeneous 2×2 blocks should fall in any one of the 14 categories shown in Figure 5.10. These 14 categories of edges, broadly grouped as 4 different classes, can be identified using the sign and magnitude of their UMRT coefficients. Each type has its own pre-generated codebooks and are used to vector quantize the inhomogeneous 2×2 blocks. A special symbol is added to the Quad-Tree data structure to identify the vector quantized 2×2 blocks.

The Decoding process of MATC-CVQ is as follows. The received bit stream is entropy decoded using Arithmetic decoding and the coefficient array & position array are separated. The size and position of each sub-image is obtained from the Quad-Tree data structure. All the higher sized (higher than 2×2) sub-images are formed as explained in section 4.3.4. The special symbol added to the Quad-Tree data structure during the coding process is used to distinguish transform coded and vector quantized 2×2 blocks and these 2×2 blocks are decoded according to the way in which they are encoded.

5.3.3 Simulation Results

A computer simulation is carried out to analyze the performance of the proposed MATC-CVQ technique. The input image has resolution of 8 bits per pixel and the image size is 512×512.

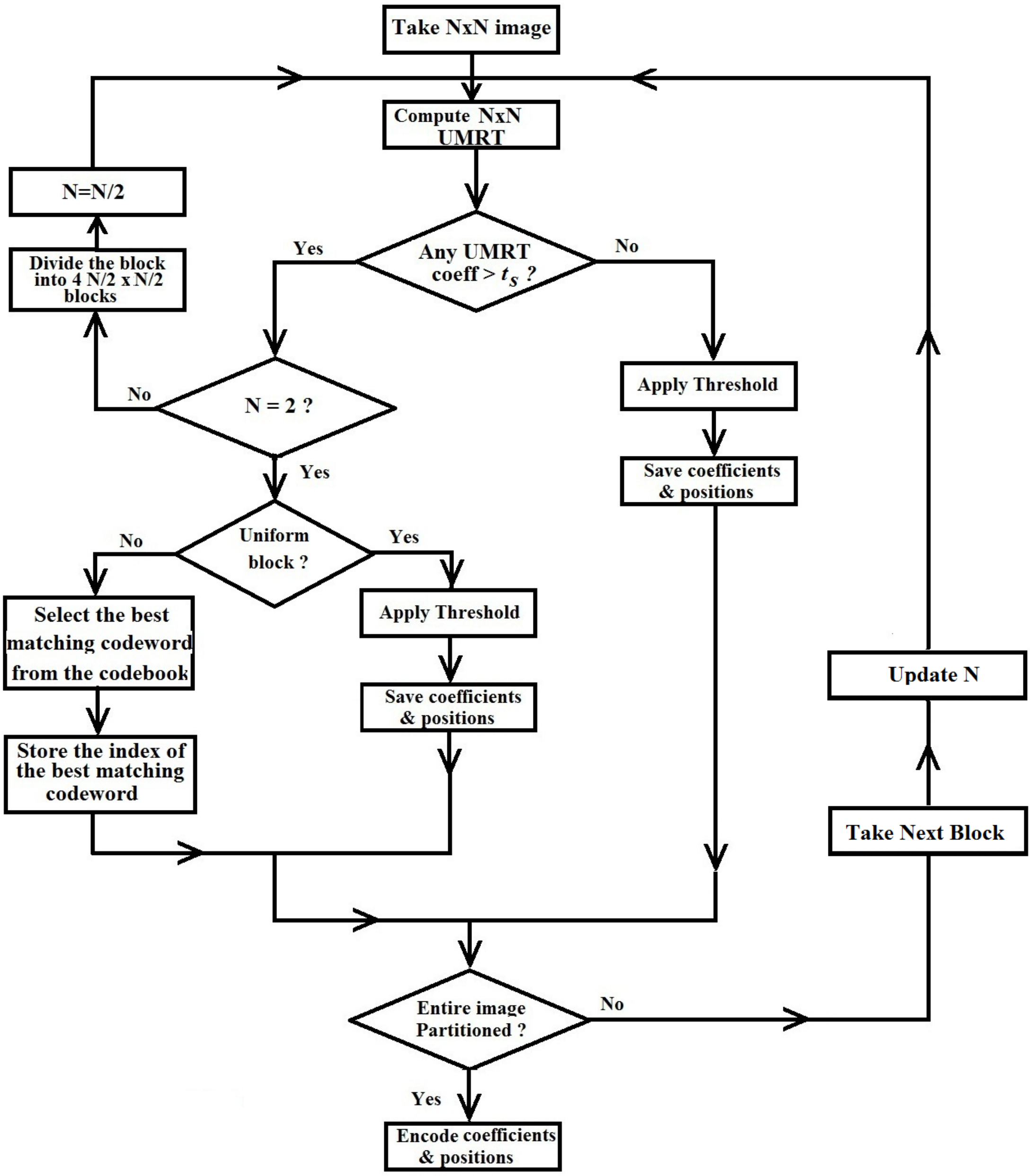

Take NxN image
Compute NxN UMRT
Any UMRT coeff > t_s ?
Yes
No
N=N/2
Divide the block into 4 N/2 x N/2 blocks
N = 2 ?
No
Yes
Apply Threshold
Save coefficients & positions
Uniform block ?
No
Yes
Select the best matching codeword from the codebook
Apply Threshold
Save coefficients & positions
Store the index of the best matching codeword
Update N
Take Next Block
Entire image Partitioned ?
No
Yes
Encode coefficients & positions

Figure 5.17: Flow chart of MATC-CVQ

The largest block size allowed is the size of the image itself (512×512) and the smallest is 2×2. The segmentation threshold t_s is experimentally selected as 100. The quantized coefficients are Arithmetic coded. Table 5.2 shows the simulation results for various gray scale images. It can be seen that the proposed system produces lesser bits per pixel with better reconstructed image quality for most of the images. Figure 5.18 – 5.20 show original and reconstructed versions of *Lena, Baboon* and *Cameraman* images.

By analyzing the reconstructed images, it is quite evident that they possess high subjective fidelity.

Table 5.2: Performance of MATC-CVQ

Image	bpp	PSNR(dB)
Lena	0.33	32.89
Baboon	1.22	27.18
Barbara	0.71	30.13
Goldhill	0.47	30.34
Peppers	0.57	31.91
Couple	0.50	30.59
Elaine	0.35	30.51
Cameraman	0.40	33.25
Boat	0.51	29.71
Bridge and Stream	0.77	26.37

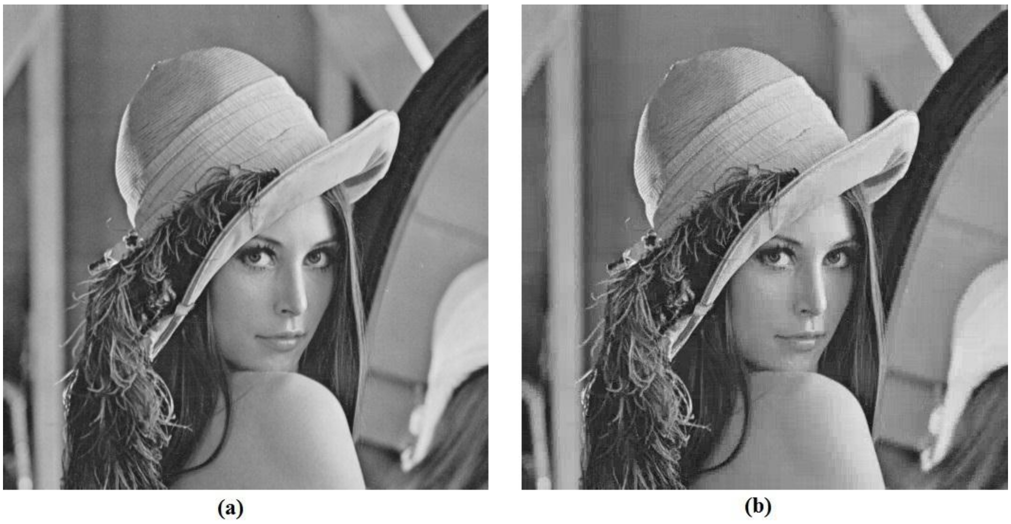

(a) (b)

Figure 5.18: Lena (a) Original and (b) reconstructed after compressed using MATC-CVQ coder (bpp =0.33, PSNR =32.89)

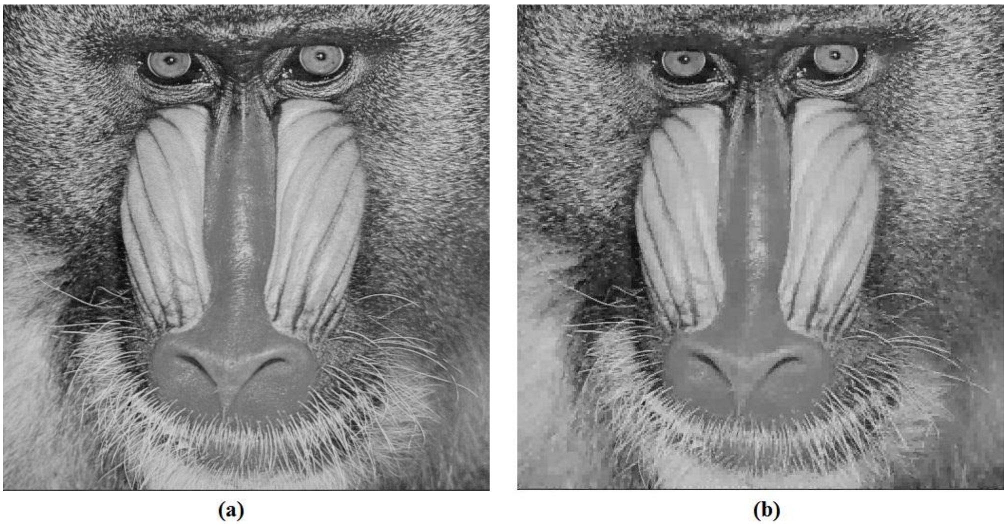

(a) (b)

Figure 5.19: Baboon (a) Original and (b) reconstructed after compressed using MATC-CVQ coder (bpp =1.22, PSNR = 27.18)

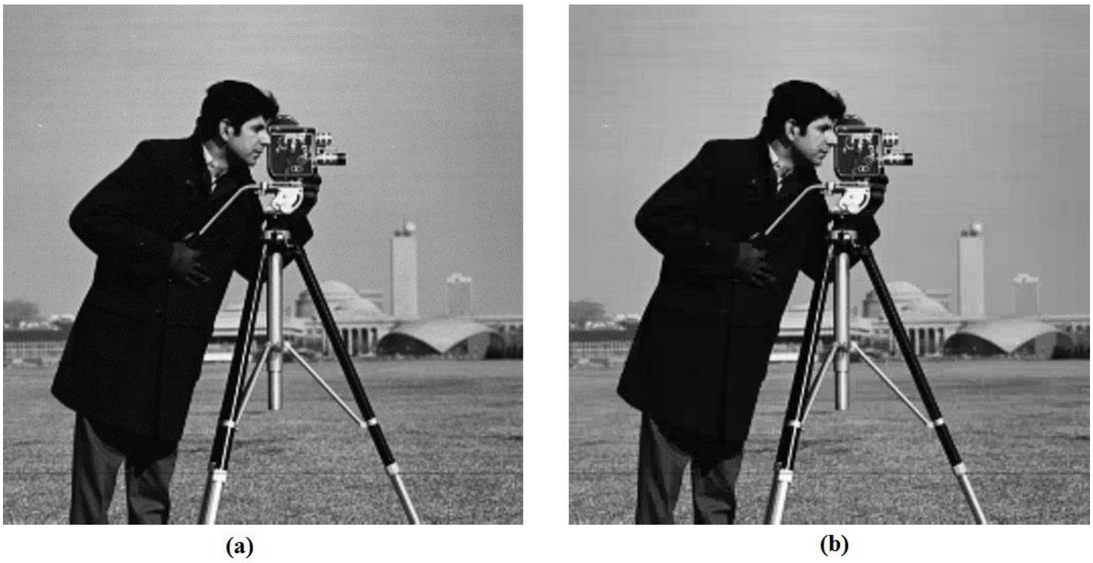


Figure 5.20: Cameraman (a) Original and (b) reconstructed after compressed using MATC-CVQ coder (bpp =0.430, PSNR = 33.25).

5.3.4 Performance comparison

A comparison between the performances of the proposed MATC-CVQ coder and the coders explained in the previous chapters is shown in Table 5.3, Table 5.4 and Table 5.5 for bit rates 0.3, 0.5 and 0.75 respectively. It is evident that the proposed method gives better performance at all bit rates. At lower bit rate (0.3 bpp), the PSNR values of the reconstructed images using the proposed coder are 2dB more than that of the MATC coder presented in section 4.3.4. Also SSIM (Structural Similarity) metric is measured to analyze the subjective visual quality of the reconstructed images.

Table 5.3: Comparison of MRT based compression techniques for 0.3 bpp

Image	8×8		4×4		Hybrid of 8×8 & 4×4		MATC		MATC -CVQ	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	27.68	0.92	27.05	0.914	27.98	0.922	30.27	0.940	32.84	0.96
Peppers	27.43	0.915	26.52	0.910	27.48	0.915	30.49	0.937	31.08	0.940
Goldhill	27.25	0.913	26.68	0.911	26.98	0.912	27.42	0.915	29.21	0.932
Couple	26.47	0.904	25.87	0.900	25.94	0.901	25.94	0.910	28.32	0.917
Cameraman	28.55	0.92	26.79	0.893	28.63	0.920	31.03	0.937	32.60	0.946
Boat	25.94	0.898	25.19	0.896	25.49	0.897	25.79	0.90	27.11	0.923
Elaine	28.46	0.910	28.37	0.91	28.50	0.91	29.93	0.918	30.42	0.921

Table 5.4: Comparison of MRT based compression techniques for 0.5 bpp

Image	8×8		4×4		Hybrid of 8×8 & 4×4		MATC		MATC -CVQ	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	30.18	0.933	30.27	0.936	30.83	0.94	31.01	0.948	33.60	0.967
Peppers	29.86	0.928	30.41	0.933	30.95	0.938	31.88	0.943	32.41	0.951
Goldhill	29.04	0.921	29.16	0.922	29.28	0.924	29.73	0.929	30.54	0.942
Couple	28.82	0.912	29.16	0.914	29.21	0.915	29.25	0.918	30.59	0.929
Cameraman	31.20	0.929	31.47	0.932	32.56	0.941	32.45	0.943	33.72	0.961
Boat	27.88	0.919	28.29	0.923	28.45	0.925	28.76	0.928	29.70	0.937
Elaine	29.93	0.921	30.60	0.924	30.65	0.924	30.71	0.925	31.48	0.935

Table 5.5: Comparison of MRT based compression techniques for 0.75 bpp

Image	8×8		4×4		Hybrid of 8×8 & 4×4		MATC		MATC -CVQ	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	32.03	0.945	32.72	0.951	32.99	0.953	33.08	0.958	34.42	0.971
Peppers	31.55	0.936	32.71	0.952	32.86	0.953	33.40	0.960	34.17	0.967
Goldhill	30.34	0.933	31.13	0.941	31.20	0.942	31.20	0.942	32.56	0.953
Couple	30.86	0.926	31.70	0.935	31.80	0.936	31.90	0.939	32.63	0.946
Cameraman	33.35	0.947	34.25	0.961	34.69	0.962	34.85	0.965	35.97	0.975
Boat	29.48	0.931	30.39	0.937	30.52	0.937	30.55	0.940	31.27	0.948
Elaine	31.15	0.929	31.57	0.932	31.57	0.932	31.60	0.935	32.48	0.947

A comparison of the proposed MATC-CVQ coder with the DCT and DWT based transform coding schemes [131] is shown in Table.5.6 & Figure 5.21. The table shows that the proposed coder produces better PSNR for a compression ratio of 10:1. Here the advantage of the proposed coder is that the UMRT can be computed using real additions only, compared to additions and multiplications required for DCT and DWT.

Table 5. 6: Comparison between DCT, DWT and MRT based techniques

Image	Compression Ratio	PSNR(dB)		
		DCT (Baseline)	DWT (Baseline EZW)	MRT (MATC-CVQ)
Lena	10:1	32.90	32.51	34.44
Peppers	10:1	34.30	34.43	34.42
Baboon	10:1	25.30	24.91	24.92
Zebra	10:1	23.71	23.58	23.30

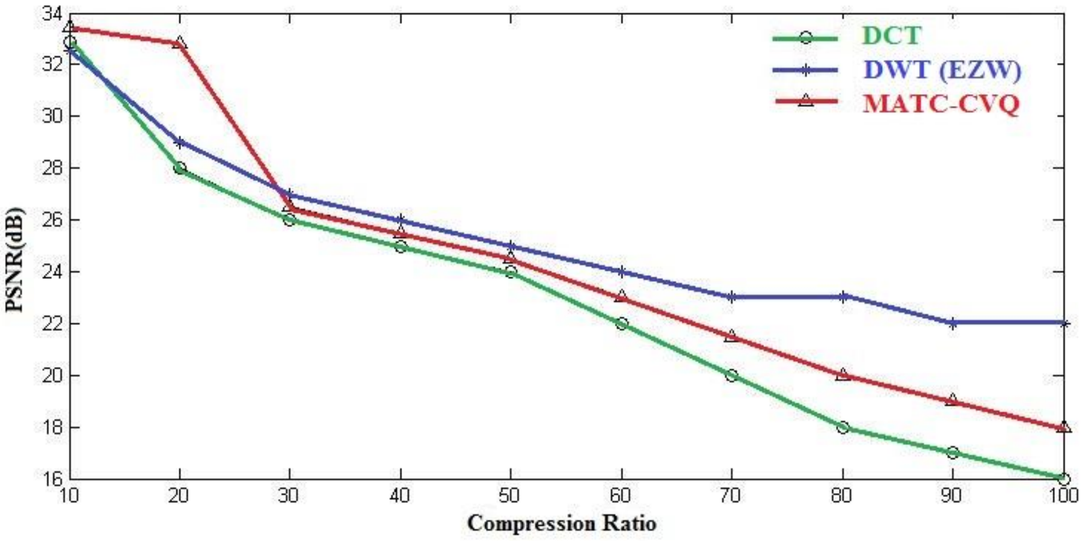


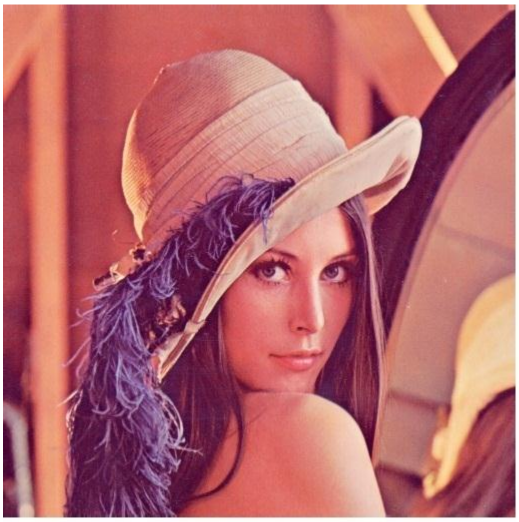
Figure 5.21: Comparison of DCT, DWT and MRT based techniques

5.3.5 Application in Color Images

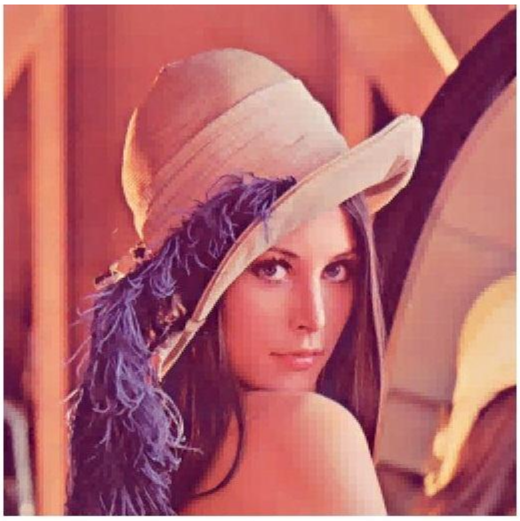
The results of applying the proposed MATC-CVQ algorithm on various color images (RGB model) are shown in Table 5.7. Figure 5.22 shows the original and decompressed versions of various color images using the MATC-CVQ algorithm

Table 5.7: Performance of MATC-CVQ in color images

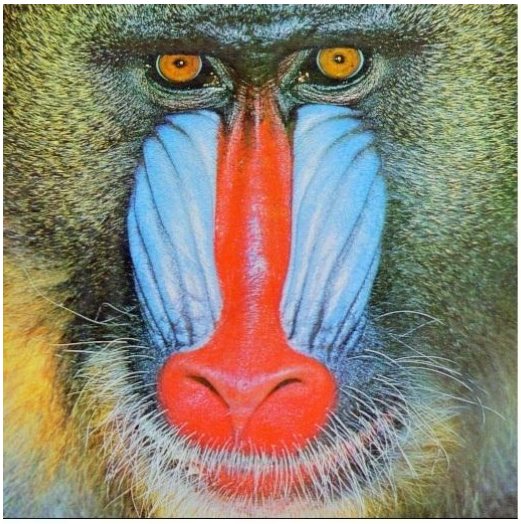
Image	bpp	PSNR
Lena	1.44	31.43
Crown	0.88	33.93
Baboon	3.96	26.01
Peppers	1.48	31.14



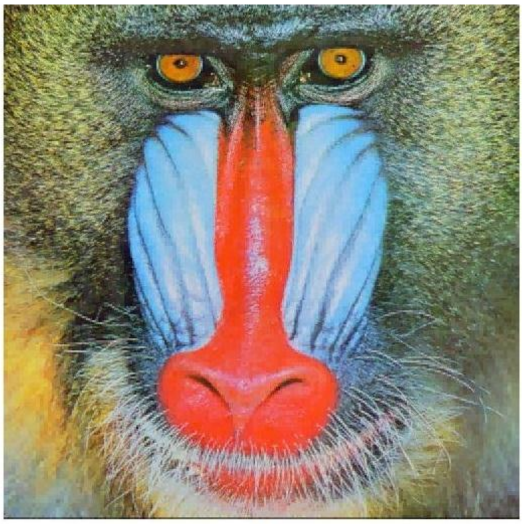
(a)



(e)



(b)



(f)

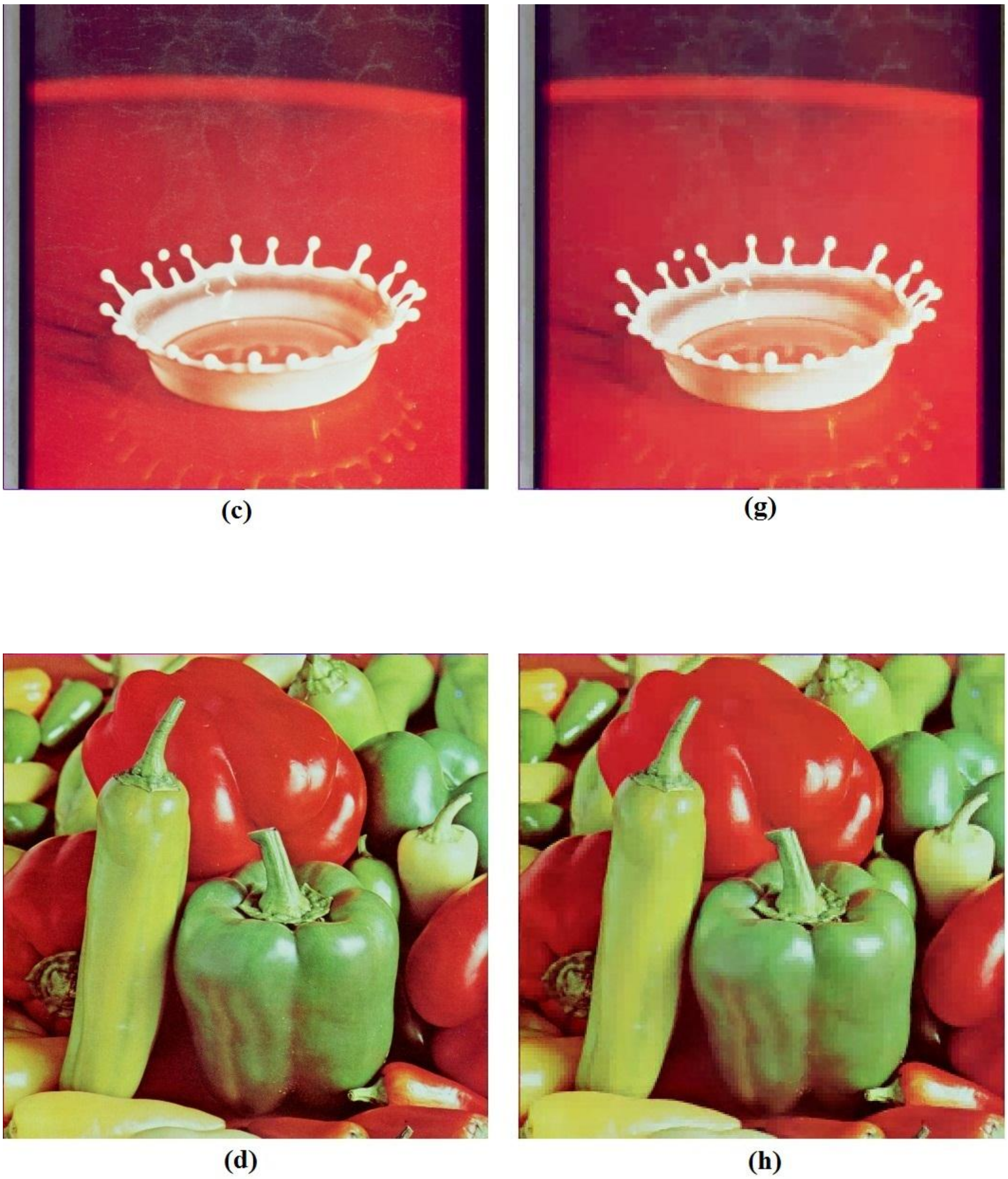


Figure 5.22: (a) – (d) Original Color images and (e) – (h) their reconstructed versions after compressed using MATC - CVQ coder

6

Summary, Conclusion and Future Work

Contents

6.1 Summary of the work....164
  6.1.1 MRT based Fixed Block size Transform Coding....164
  6.1.2 Variable Block size Transform coders using MRT....165
  6.1.3 MRT based Adaptive Transform coder with Classified Vector Quantization (MATC-CVQ)....165
6.2 Conclusion....166
6.3 Suggestions for future work....168

6.1 Summary of the work

This chapter presents the summary of the investigations carried out and comments on the results obtained therein. The essence of the research work is incorporated in chapters 3 to 5. The application of MRT in fixed block size transform coding is explained in chapter 3. Both 8×8 MRT based transform coder and 4×4 MRT based transform coder are developed and the simulation results are analyzed. The implementation of variable block size transform coders using MRT is discussed in chapter 4. Chapter 5 describes the implementation of Vector Quantization in MRT domain and the design of a new "MRT based Adaptive Transform Coder with Classified Vector Quantization (MATC-CVQ)".

6.1.1 MRT based Fixed Block size Transform Coding

MRT based fixed block size transform coders are implemented using both 8×8 MRT and 4×4 MRT. Their results are analyzed and found that the 8×8 MRT based technique gives better performances at lower bits per pixel and 4×4 MRT based technique gives better performances at higher bits per pixels. The template based computation of unique MRT coefficients is introduced to reduce the time taken for the coding process. The time difference between the normal MRT computation and template based MRT computation are analyzed and found that the template based MRT computation works much faster than the normal MRT computation.

6.1.2 Variable Block size Transform coders using MRT

A hybrid transform coder based on 8×8 MRT and 4×4 MRT is designed to combine the good features of both, and its results are analyzed. This hybrid approach outperforms the fixed block size methods both at lower and higher bits per pixel. UMRT is a new algorithm for the computation of the unique MRT coefficients. The significance of the UMRT coefficients is analyzed and its use in the segmentation of images is studied. A criterion, based on UMRT coefficients, is introduced to perform the Quad-Tree segmentation of images. MRT based Adaptive Transform Coder (MATC), employing Quad–Tree segmentation of images using UMRT coefficients, is designed and simulated. Its performance is evaluated and compared against fixed block size transform coders and existing techniques.

6.1.3 MRT based Adaptive Transform coder with Classified Vector Quantization (MATC-CVQ)

A Classified Vector Quantization scheme based on UMRT is designed. The identification of edge blocks in the image and classification of those edge blocks are done using UMRT coefficients. The strength of the UMRT coefficients is used to distinguish between edge blocks and smooth blocks. These edge blocks are then classified into different categories, according to the orientation of edges, by identifying the position of zeros and sign of certain UMRT coefficients. A codebook is generated for each category of edges using UMRT based classification methods.

Applying the CVQ for all the blocks in the images is a waste of time. So, the MRT based Adaptive Transform coder with Classified Vector Quantization (MATC-CVQ) is introduced. The CVQ is needed for only the edge blocks. The remaining smooth blocks in the image are transform coded using MRT based Adaptive Transform Coder (MATC). The results of the MRT based Adaptive Transform coder with Classified Vector Quantization (MATC-CVQ) is obtained and analyzed. The results are compared against all the methods already discussed in this work and existing techniques like DCT and DWT based techniques.

6.2 Conclusion

A variety of image compression techniques based on MRT are implemented. These methods include:

1) 8×8 MRT based Transform Coder
2) 4×4 MRT based Transform Coder
3) Hybrid Transform Coder based on 8×8 MRT and 4×4 MRT
4) MRT based Adaptive block size Transform Coder (MATC)
5) Vector Quantization based on MRT
6) MRT based Adaptive block size Transform Coder with Classified Vector Quantization (MATC-CVQ)

The 8×8 MRT based Transform Coder has the merit of reasonable PSNR at lower bits per pixel. But it lacked in terms of subjective visual quality as the blocking artifacts were visible at lower bits per pixel. The 4×4 MRT based Transform Coder is introduced to reduce the blocking artifacts at lower bits per pixels. It reduced the blocking artifacts at the expense of increased bpp.

A hybrid coder based on 8×8 MRT based coder and 4×4 MRT based coder is designed and its performance analyzed to combine the good features of both. The hybrid coder outperformed the fixed block sized coders both at lower and higher bits per pixel. The study extended to the design of an MRT based Adaptive block size Transform Coder (MATC). UMRT based Quad-Tree segmentation is incorporated in the design of the MATC. This method utilizes block sizes from the size of the image itself to 2×2. It produced better results in terms of bpp, PSNR and subjective visual quality. In this method, the problem of blocking artifacts is reduced to a large extent. A comparison of this technique with the existing techniques like DCT and DWT based techniques is performed. The comparison proved that the MATC is as good as many existing techniques.

A Vector Quantization Scheme based on MRT and isometric transformations is designed subsequently. For this, a universal codebook is generated. This scheme produced lower bpp at the expense of increased coding time. The time taken for codebook generation is also high. MRT based Adaptive block size Transform Coder with Classified Vector Quantization (MATC-CVQ) is introduced to overcome such a computational overhead. This technique combines the good features of MRT based Adaptive block size Transform Coder (MATC) and Classified Vector Quantization (CVQ). It is implemented and the performance is compared against all the methods already discussed in this work and existing techniques. The performance of MATC-CVQ technique is better, compared to existing techniques, for certain images.

6.3 Suggestions for future work

- The thesis presents the development of image compression techniques based on MRT. Recently, in literature, a sequency based MRT computation algorithm (SMRT) is introduced. The pattern in which each coefficient is placed in the SMRT matrix is a promising feature. The effect of this SMRT in compressing images can be studied as an extension of this thesis work.
- A quantization matrix, just like the one used in the JPEG compression, can be developed by analyzing the significance of each SMRT coefficient for the reconstruction of the original data.
- A compression technique that doesn’t involve the segmentation of images, like wavelet based, can also be developed as an extension of this work.
- In MATC-CVQ, intelligence and faster codebook search techniques can be incorporated to find the best suited codeword by using Fuzzy logic, Genetic Algorithm or Neural network.

REFERENCES

[1] N. Weiner, “Hermitian polynomials and Fourier analysis”, J. Math. Phys., 8, 70-73., 1929

[2] H. Hotelling, “Analysis of complex of statistical variables into principal components”, J. Educ. Psychol., 24, 417-447, 1933.

[3] K. Karhunen (translated by I. Selin), “On linear methods in probability theory”, Doc. T-131, RAND Corporation, Santa Monica, CA, 1960.

[4] J. Max “Qunatizing for minimum Distortion”, IEEE Trans. Inform. Theory, March, pp7-12, 1960.

[5] D. A. Huffman “A method for Comstruction of Minimum Redundancy Codes”, Proc. IRE, Vol 40, pp1016-1021, 1962.

[6] W. K. Pratt, J. Kane and H. C. Andrews, “Hadamard transform image coding”, IEEE Proc. Comm., 57, 58-68, 1969.

[7] G. H. Golub and C. Reinsels, “Singular value decomposition and least square solutions”, Numer. Math., 14, 403-420, 1970.

[8] T. Berger, Rate Distortion Theory, Prentice-Hall, Englewood Cliffs, NJ, 1971.

[9] G. B. Anderson and T. S. Huang, “Picture bandwidth compression by piecewise Fourier transformation”, IEEE Trans. Comm. Technol., CT-19, 133-140, 1971.

[10] A. Habibi and P. A. Wintz, “Image coding by linear transformation and block quantization”, IEEE Trans. Comm. Tech., CT-19, 50-62, 1971.

[11] P. A. Wintz, “Transform picture coding”, Proc. IEEE, 60, 809-820, 1972.

[12] W. Chen and W. K. Pratt, “Color image coding with slant transform”, Proceedings of Symposium on Applications of Walsh Functions, Washington, DC, USA, pp. 155-161, 1973.

[13] N. Ahmed, T. Natarajan and K. R. Rao, “Discrete cosine transform”, IEEE Trans. Comm.,C-23, 90-93, 1974.

[14] W. K. Pratt, L. R. Welch and W. Chen, “Slant transform for image coding”, IEEE Trans. Comm., C-22, 1075-1093, 1974.

[15] N. Ahmed and K. R. Rao, Orthogonal Transforms for Digital Signal Processing, Springer Verlag, New York, 1975.

[16] A. Croiser, D. Esteban, and C. Galand, “Perfect Channel Splitting by use of Interpolation/Decimation Tree Decomposition Techniques”, International Conference on Information Science Systems, August, pp443-446, 1976.

[17] H. C. Andrews and C. L. Paterson, “Singular value decomposition (SVD) image coding”, IEEE Trans. Comm., 24, 425-432, 1976.

[18] F. J. Mac Williams and N. J. A. Sloane, The Theory of Error Correcting Codes, Elsevier, Amsterdam, 1977.

[19] W. H. Chen and C. H. Smith, “Adaptive coding of monochrome and color images”, IEEE Trans. Comm., C-25, 1285-1292, 1977.

[20] W. Chen, C. H. Smith and S. Fralick, “A fast computational algorithm for the discrete cosine transform”, IEEE Trans. Comm., C-25, 1004-1009, 1977.

[21] Y. Linde, A. Buzo, R. M Gray, “An algorithm for vector quantizer design”, IEEE Trans. on Communications, vol.36, 84-95, 1980.

[22] V.Namias, “The fractional order Fourier transform and its application to quantum mechanics”, J. Inst. Math. Appl., 25, 241-265, 1980.

[23] M. S. Fu and J. K. Mui, “A Survey On Image Segmentation”, Pattern Recognition, Vol.8, pp.3-16, 1981.

[24] S. P. Lloyd “Least squares quantisation in PCM”, Unpublished Bell Labs. Techn. Note: portions presented in the March 1982 special issue on quantisation of the IEEE Trans. Inform. Theory.

[25] H. Musmann, P.Pirsch, and H. Grallert, “Advances in picture coding”, proc, IEEE, Apr., pp. 523-542, 1982.

[26] H. Samet, “The Quad-Tree and Related Data Structures”, ACM Computing Surveys, Vo1.16, No.2, pp.188-260, June 1984.

[27] N. S. Jayant and P. Noll, Digital Coding of Waveforms, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[28] R. M. Haralick and L. G. Shapiro, “Survey, Image Segmentation Techniques”, Computer vision, graphics, and image processing, Vo1.29, pp.100-132, 1985.

[29] M. Kunt, A. Ikonomopoulos, and M. Kocher, “Second-Generation Image-Coding”, Proceedings of the IEEE, Vo1.73, pp.549-574, April 1985.

[30] E. Oja and J. Karhunen, “On stochastic approximation of eigen vectors and eigen values of the expectation of a random matrix”, J. Math. Anal. Appl., 106, 61-68, 1985.

[31] R. J. Clark, “Transform Coding of Images”, Academic Press, San Diego, CA, 1985.

[32] I. Jolliffe, Principal Component Analysis, Springer Verlag, New York, 1986.

[33] P. M. Farrelle and A. K. Jain, “Recursive block coding: A new approach to transform coding”, IEEE Trans. Comm., C-34, 161-179, 1986.

[34] B. Ramamurthi, and A. Gersho, “Classified Vector Quantization of Images”, IEEE Transactions on communications, vol. com -34, no. 11, pp. 1105-1115, November 1986.

[35] J. Vaisey and A. Gersho, “Variable Block-Size Image Coding”, International Conference on Acoustics, Speech, and Signal Processing, pp.1051-1054, 1987.

[36] M. Kunt, M. Benard, and R. Leonardi, “Recent Results in High-Compression Image Coding”, IEEE Transactions on Circuits and Systems, Vol.CAS-34, pp. 1306- 1336, November 1987.

[37] M. F. Barnsley and A. Jacquin “Application of Recurrent Iterated Function Systems to Images”, Proc. SPIE, Vol.1001, pp122-131, 1988.

[38] P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. C. Chen, “A Survey of thresholding Techniques”, Computer vision, graphics, and image processing, Vol.41, pp.233-360, 1988.

[39] I. Daubechies “Othonormal Bases of Compactly Supported Wavelets”, Commum. Pure Appl. Math, Vol. 44, pp909-996, November 1988.

[40] N.M. Nasrabadi, R.A. King, “Image coding using vector quantization: a review”, IEEE Trans. On Communications,vol. 36, 957-971, 1988.

[41] P. Strobach, “Image Coding Based on Quadtree Structured Recursive Least-Squares Approximation”, Proc. IEEE ICASSP-89, pp1961-1964, 1989.

[42] S. G. Mallat, “A theory of multiresolution signal decomposition: The wavelet representation”, IEEE Trans. Pattern Anal. Mach. Intell., 11, 674-693, 1989.

[43] K. Rose, A. Heinman, and I. Dinstein, DCT/DST alternate transform image coding, IEEE Trans. Commun., vol. 38, pp. 94-101, Jan. 1990.

[44] G. K. Wallace,”Overview of the JPEG (ISO/CCITT) Still Image Compression Standard”, Image Processing Algorithms and Techniques, Proc. SPIE, Vol 1244, pp220-233, February 1990.

[45] E. Feig, “A fast scaled DCT algorithm,” in Proc. SPIE Image Processing Algorithms and Techniques, vol. 1244, pp. 2–13, Feb. 1990.

[46] H. Abut, Ed. “Vector Quantization”, IEEE Reprint Collection, Piscataway, NJ, IEEE, May 1990.

[47] R. M. Gray, Source Coding Theory, Kluwer Academic Publishers, Norwell, MA, 1990.

[48] K. R. Rao and P. Yip, Discrete Cosine Transform: Algorithms, Advantages and Applications, Academic Press, New York, 1990.

[49] G. E. Oien, S. Lepsy and T. A. Ramstad “An Inner product Space Approach to Image Coding by Contractive Transformations”, Proc. ICASSP-91, pp2773-2776, 1991.

[50] A. Gersho and R. M. Gray Vector Quantization and Signal Compression, Kluwer Academic Publishers, 1991.

[51] R. A. DeVore, B. Jawerth, and B. Lucier “Data Compression Using Wavelets : Error, Smoothness and Quantization”, Proc. IEEE Data Compression Conf., pp186-195, April 1991.

[52] G. K. Wallace, “The JPEG Still Picture Compression Standard”, Comm. ACM, vol. 34, no. 4, pp. 30- 44, April 1991.

[53] D. S. Kim and S. U. Lee, “Image vector quantizer based on a classification in the DCT domain,” IEEE Trans. Commun., vol. 39, pp. 549-556, Apr. 1991.

[54] A. Hung and T. Meng, “Optimal quantizer step sizes for transform coders,” in Proc. ICASSP’91, pp. 2621–2624, Apr. 1991.

[55] E. A. Riskin and R. M. Gray “A Greedy Tree Growing Algorithm for the Design of Variable Rate Vector Quantizers”, IEEE Trans. Signal Processing, Vol. 39, pp2500-2507, November 1991.

[56] Wai-Kuen Cham and Percy Pui-Chiu Yip, Integer Snusoidal transforms for image processing, Int. J. Electronics, volume 70, no. 6, pp.1015-1030, 1991.

[57] D. M. Monro and F. Dudbridge “Fractal Approximation of Image Blocks”, Proc. ICASSP-92, ppIII:485-488, 1992.

[58] R. C. Gonzales and R. E. Woods Digital Image Processing, MA: Addison-Wesley, 1992.

[59] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, “Image coding using wavelet transform,” IEEE Trans. Image Processing, vol. 1, pp. 205–221, Apr. 1992.

[60] I. Daubechies “Ten Lectures on Wavelets”, Philadelphia : Soc, Ind. Appl. Math. 1992.

[61] A. Cohen, I. Daubechies, and J. C. Feauveau “Biorthogonal Bases of Compactly Supported Wavelets”, Commun. Pure Appl. Math, Vol.45, pp485-560, 1992.

[62] M. Vetterli, and C. Herley “Wavelets and Filter Banks : Theory and Design”, IEEE Trans. Signal Processing, Vol.40 No. 9, pp2207-2232, September 1992.

[63] R. A. DeVore, B. Jawerth, and B. Lucier “Image Compression Through Wavelet Transform Coding”, IEEE Trans. Inform. Theory, Vol. 38 No. 2, pp719-746, 1992.

[64] J. Vaisey and A. Gersho, “Image Compression with Variable Block Size Segmentation’’, IEEE Transactions on Signal Processing, Vo1.40, pp.2040-2060, 1992.

[65] Y. Fisher, “Fractal Image Compression”, SIGGRAPH Course Notes, 1992.

[66] A.E. Jacquin, “Image coding based on a fractal theory of iterated contractive image transformations”. IEEE Trans. on Image Processing, vol. 1, pp.18-30, 1992.

[67] R. D. Dony, S. Haykins C. Coblentz and C. Nahmias, “Compression of digital chest radiographs using a mixture of principle components neural network: An evaluation performance”, Radio Graphics, 16, 1481-1488, 1992.

[68] M. Bosi and G. Davidson, “High quality low bit-rate audio transform coding for transmission and multimedia applications”, J. Audio Eng. Soc., 32, 43-50, 1992.

[69] H. S. Malavar, Signal Processing with Lapped Transforms, Artech House, Norwood, MA, 1992.

[70] J. Forment and S. Mallat, “Second Generation Compact Image Coding with Wavelets in Wavelets: A Tutorial in Theory and Applications”, Vol. 2, Academic Press, New York, 1992.

[71] R. R. Coifman and M. V. Wickerhauser, “Entropy based algorithm for best basis selection”, IEEE Trans. Inf. Theory, 38, 713-718, 1992.

[72] W. B. Pennebaker and J. L. Mitchell, JPEG Still Image Data Compression Standard. New York: Van Nostrand Reinhold, 1992.

[73] W. Y. Chan, S. Gupta, and A. Gersho, “Enhanced multistage vector quantization by joint codebook design,” IEEE Trans. Commun., vol. 40, pp. 1693-1697, Nov. 1992.

[74] S. Wu and A. Gersho, “Rate-constrained picture adaptive quantization for JPEG baseline coders,” in Proc. ICASSP’93, vol. 5, pp. 389–392, Apr. 1993.

[75] A. E. Jacquin “Fractal Coding : a Review”, Proc. IEEE, Vol. 81 No. 10, pp1451-1465. , October 1993

[76] M.F. Barnsley, L.P. Hurd, “Fractal Image Compression”, AK Peters, Ltd. Wellesley, Massachusetts, 1993.

[77] J. M. Shapiro, “Embedded image coding using zero-trees of wavelet coefficients”, IEEE Trans. Signal Process., 41, 3445-3462, 1993.

[78] M. H. Lee and G. Crebbin, “Classified vector quantisation with variable block-size DCT models,” IEE Proc.-Vis. Image Process., vol. 141, pp. 39-48, Feb. 1994.

[79] D. M. Monro, and S. J. Woolley, “Fractal Image Compression Without Searching”, Proc. ICASSP-94, pp: 557-560 , 1994.

[80] S. J. Woolley, and D. M. Monro “Rate Distortion Performance of Fractal Transforms for Image Compression”, Proc. Fractals, Vol.2 No.3, pp395-398, 1994.

[81] M R. Pal and S. K. Pal, “A Survey on Image Segmentation Techniques”, Pattern Recognition, Vo1.26, pp.1277-1294, 1994.

[82] E. Shusterman and M. Feder, “Image Compression via Improved Quadtree Decomposition Algorithms”, IEEE Transactions on Image Processing, Vol.3, pp.207-215, 1994.

[83] Y. Fisher, “Fractal Image Compression: Theory and Applications”, Springer-Verlag, 1994.

[84] L. B. Almeida, “The Fractional Fourier transform and time-frequency representation”, IEEE Trans. Signal Process., 42, 3084-3091, 1994.

[85] G. J. Sullivan and R. L. Baker, “Efficient quadtree coding of images and video,” IEEE Transactions on Image Processing, vol. 3, no. 3, pp. 327–331, May 1994.

[86] W. Sweldens, “The lifting scheme: A custom-design construction of biorthogonal wavelets”, Technical Report #7 (1994), Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina.

[87] K. Ramchandran and M. Vetterli, “Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility,” IEEE Trans. Image Processing, vol. 3, pp. 700–704, Sept. 1994.

[88] S. J. Woolley and D. M. Monro “Optimal Parameters for Hybrid Fractal Image Coding”, Proc. ICASSP-95, pp2571-2574, 1995.

[89] X. Wu and Y. Fang "A Segmentation Based Predictive Multiresolution Image Coder", IEEE Trans. Image Processing, Vol.4 No.1, pp34-47, January 1995.

[90] R. D. Dony and S. Haykins, "Optimally adaptive transform coding", IEEE Trans. Image Process., 4, 1358-1370,1995.

[91] R. J. Clark, "Digital Compression of Still Images and Video", Academic Press, San Diego, CA, 1995.

[92] Y. T. Chan, "Wavelet Basics", Kluwer Academic Publishers, Norwell, MA, 1995.

[93] S. A. Mohamed and M. M. Fahmy, "Image compression using VQ-BTC," IEEE Trans. Commun., vol. 43, pp. 2177-2182, July 1995.

[94] M. Vetterli and J. Kovacevic, "Wavelet and Sub-band Coding", Prentice-Hall, Englewood Cliffs, NJ, 1995.

[95] C. S. McGoldrick, W. J. Doweling and A. Bury, "Image coding using the singular value decomposition and vector quantization", Proceedings of International Conference on Image Processing and Its Applications, Edinburgh, Scotland, pp.296-300, 1995.

[96] J. F. Yang and C. L. Lu, "Combined techniques of singular value decomposition and vector quantization", IEEE Trans. Image Process., 4, 1141-1146, 1995.

[97] "MPEG-2 video," ITU-T Recommendation H.262-ISO/IEC 13818-2, Jan. 1995.

[98] "Video coding for low bitrate communication," ITU-T Recommendation H.263, Dec. 1995.

[99] P. C. Cosman, R. M. Gray and M. Vetterli "Vector Quantization of Image Subbands : A Survey", IEEE Trans. Image Processing, Vol. 5 No. 2, pp202-225, February 1996.

[100] R. L. Queiroz, T. Q. Nguyen and K.R. Rao "The GENLOT: Generalized Linear-Phase Lapped Orthogonal Transform", IEEE Trans Signal Processing, Vol 44no.3, pp497-507, March 1996.

[101] M. Nelson and J. Gailly "The Data Compression Book", 2nd Edition, MRT Books, pp113-152, 1996.

[102] S. H. Yang and S. S. Yang "New classified vector quantization with quadtree segmentation for image coding." Proceedings of ICSP pp1051-1054, 1996.

[103] K. I. Diamantaras and S. Y. Kung, "Principal Component Neural Networks: Theory and Applications", John Wiley and Sons, New York, 1996.

[104] G. Strang and T. Nguyer, "Wavelet and Filter Banks", Welesley-Cambridge Press, Welesley, MA, 1996.

[105] A. Said and W. A. Pearlman, “A new fast and efficient image coding based on set partitioning in hierarchical tree”, IEEE Trans. Circuits Syst. Video Technol., 6, 243-250, 1996.

[106] M. J. Tsai, J. D. Villasenor and F. Chen, “Stack-run image coding”, IEEE Trans. Circuits Syst. Video Technol., 6, 519-521, 1996.

[107] H. M. Ozaktas, O. Arikan, M. A. Kutay and G. Bozdagt. “Digital computation of the fractional Fourier transform”, IEEE Trans. Signal Process., 44, 2141-2150, 1996.

[108] C. F. Barnes, S. A. Rizvi, and N. M. Nasrabadi, “Advances in residual vector quantization: a review,” IEEE Trans. Image Processing, vol. 5, pp. 226-262, Feb. 1996.

[109] Z. Xiong, O. Guleryuz, and M. T. Orchard, “A DCT-based embedded image coder,” IEEE Signal Processing Lett., vol. 3, pp. 289–290, Nov. 1996.

[110] M. Crouse and K. Ramchandran, “Joint thresholding and quantizer selection for transform image coding: Entropy-constrained analysis and applications to JPEG,” IEEE Trans. Image Processing, vol. 6, pp. 285–297, Feb. 1997.

[111] R. de Queiroz, C. Choi, Y. Huh, and K. Rao, “Wavelet transforms in a JPEG-like image coder,” IEEE Trans. Circuits Syst. Video Technol., vol. 7, pp. 419–424, Apr. 1997.

[112] R. Buccigrossi and E. P. Simoncelli, “EPWIC: Embedded Predictive Wavelet Image Coder”, Technical Report #414, GRASP Laboratory, University of Pennsylvania (May 1997).

[113] Z. Xiong, K. Ramachandran and M. T. Orchard, “Space frequency quantization for wavelet image coding”, IEEE Trans. Image Process., 6, 677-693, 1997.

[114] H. S. Chu, “A very fast fractal compression algorithm”, M.S. Thesis, National Tsing Hua University, June, 1997.

[115] M. Boliek, M. J. Gormish, E. L. Schwartz and A. Keith, “Next generation image compression and manipulations using CREW”, Proceedings of International Conference on Image Processing, Santa Barbara, CA, USA, pp. 567-570, 1997.

[116] P. Waldemar, “Image compression using singular value decomposition”, PhD. Thesis, Norwegian University of Science and Technology, Norway, 1997.

[117] P. Waldemar and P. A. Ramstad, “Hybrid KLT-SVD image compression”, Proceedings of International Conference on Acoustics,

Speech and Signal Processing, Vol.4, Munich, Germany, pp. 2713-2716, 1997.

[118] Y.W. Chen, “Vector Quantization by principal component analysis”, M.S. Thesis, National Tsing Hua University, June, 1998.

[119] R. Gopikakumari, “Investigations on the development of an ANN model & Visual manipulation approach for 2-D DFT computation in Image Processing” Ph.D. Dissertation, Cochin University of Science and Technology, Kochi, August 1998.

[120] T. Chen, Y. Hua and W. Y. Wan, “Global convergence of Oja’s subspace algorithm for principle component extraction”, IEEE Trans. Neural Networks, 9, 58-67, 1998.

[121] V. Solo and X. Kong, “Performance analysis of adaptive eigen analysis algorithms”, IEEE Trans.Signal Process., 46, 636-645, 1998.

[122] R. C. Colderbank, I. Daubechies, W. Sweldens and B. L. Yeo, “Wavelet transforms that map integer to integer”, Appl. Comput. Harmonic Anal., 5, 332-369, 1998.

[123] S. O. Aase, J. H. Husoy and P. Waldemar, “A critique of SVD-based image coding systems,” Proceedings of the 1999 IEEE International Symposium on Circuits and Systems on VLSI, Vol. 4, Orlando, Florida, USA, pp.13-16, 1999.

[124] Z. Xiong, K. Ramchandran, M. T. Orchard, and Y.-Q. Zhang "A Comparative Study of DCT- and Wavelet-Based Image Coding" IEEE Trans. Circuits Syst. Video Technol., vol. 9, No.5, pp. 285–296, Aug. 1999.

[125] D. Taubman, "High performance scalable image compression with EBCOT", IEEE Trans. Image Process., 9, 1158-1170, 2000.

[126] O. N. Gerek and M. F. Erden, "Discrete fractional cosine transform", Proceedings of IEEE Conference on Signal Processing, Communications, Circuits, and Systems, Istanbul, Turkey, pp. 322-329, 2000.

[127] J. Chen, "Image compression with SVD", Lecture Notes (2000), University of California Davis, USA.

[128] B. Arnold and A. McInnes, "An investigation into using singular value decomposition as a method of image compression", Technical Report (2000), College of Redwood, University of Canterbury, New Zealand.

[129] C. Christopoulos, A. Skodras and T. Ebrahimi. "The JPEG2000 Still Image Coding System, An Overview". IEEE Trans. on Consumer Electronics, 46(4): 1103-1127, 2000.

[130] K. R. Rao and P. C. Yip, "The Transform and Data Compression Handbook", CRC Press, New York, 2001.

[131] Sonja Grgic, Mislav Grgic, Member, IEEE, and Branka Zovko-Cihlar, Member, IEEE, “Performance Analysis of Image Compression Using Wavelets”, IEEE Transactions on Industrial Electronics, vol. 48, no. 3, pp. 682-695, june 2001.

[132] A. Dapena and S. Ahalt, “A hybrid DCT-SVD image-coding algorithm”, IEEE Trans. Circuits Syst. Video Technol., 12, 114-121, 2002.

[133] H. Ochoa and K. R. Rao, “A hybrid DWT-SVD image coding system”, Proceedings of IEEE Symposium on Micro-Nano Mechatronics and Human Science, Vol. 2, Cairo, Egypt, pp.532-535, 2003.

[134] D. Solomon, “Data Compression: The Complete Reference”, Springer Verlag, New York, 2004.

[135] Rajesh Cherian Roy, and R. Gopikakumari “A new transform for 2-D signal representation (MRT) and some of its properties”, 2004 IEEE International Conference on Signal Processing & Communications (SPCOM), pp. 363- 367, Bangalore, India, Dec. 2004.

[136] W. Ding, F. Wu and S. Li, “Lifting-based wavelet transform with directionally spatial prediction”, Proceedings of Picture Coding Symposium, San Francisco, USA, pp. 483-488, 2004.

[137] Y. Wongsawat, H. Ochoa, K. R. Rao and S. Oraintara, “A modified hybrid DCT-SVD image coding system for color image”, Proceedings of

International Symposium on Communication and Information Technology, Sapporo, Japan, pp. 766-769, 2004.

[138] S. P. Nanavati and P. K. Panigrahi, "Wavelets: Applications to image compression-I", J.Resonance, 10, 52-59, 2005.

[139] K. Singh, "Performance of discrete fractional Fourier transform classes in signal processing applications", PhD. Thesis, 2005, Thapar Institute of Engineering and Technology, India.

[140] H. Ochoa and K. R. Rao, "A hybrid DWT-SVD image coding system (HDWTSVD) for colour images", Int. J. System. Cybernet. Informat., 1, 64-69, 2005.

[141] H. Ochoa and K. R. Rao, "A new modified hybrid DWT-SVD coding system for color images", WSEAS Trans. Circuits Syst., 4, 1246-1253, 2005.

[142] Soo-Chang Pei and Jian-Jiun Ding, "The integer transforms analogous to discrete trigonometric transforms", IEEE Transactions on Signal Processing, volume 48, no. 12, pp. 3345-3364, December 2006.

[143] B. Zeng and J. Fu, "Directional discrete cosine transforms for image coding", Proceedings of IEEE Conference on Multimedia and Expo, Toronto, Canada, pp. 721-724, 2006.

[144] H. Ochoa and K. R. Rao, “A new modified version of the HDWTSVD coding system for monochromatic images”, WSEAS Trans. Syst., 5, 1190-1195, 2006.

[145] D. Xu and M. N. Do, “On the number of rectangular tilings,” IEEE Transactions on Image Processing, vol. 15, no. 10, pp. 3225–3230, Oct. 2006.

[146] W. Ding, F. Wu, X. Wu, S. Li and H. Li, “Adaptive directional lifting based wavelet transform for image coding”, IEEE Trans. Image Process., 16, 416-427, 2007.

[147] X. Hao, X. Jizheng and W. Feng, “Lifting based directional DCT-like transform for image coding”, IEEE Trans. Circuits Syst. Video Technol., 17, 1325-1335, 2007.

[148] A. Ranade, S. S. Mahabalarao and S. Kale, “A variation on SVD based image compression”, J. Image Vision Comput., 25, 771-777, 2007.

[149] R. C. Roy, “Development of a new transform: MRT,” Ph.D. Dissertation, Cochin University of Science and Technology, Kochi, 2009.

[150] V. Bhadran, “Development and implementation of visual approach and parallel distributed Architecture for 2D DFT and UMRT computation” Ph. D Dissertation, CUSAT, 2009.

[151] M. Sindhu and R. Rajkamal “Images and Its Compression Techniques – A Review” International Journal of Recent Trends in Engineering, pp 71-75, Vol 2, No. 4, November 2009.

[152] A. A. Kassim, W. S. Lee, and D. Zonoobi, “Hierarchical segmentation based image coding using hybrid quad-binary trees,” IEEE Transactions on Image Processing, vol. 18, no. 6, pp. 1284–1291, Jun. 2009.

[153] K. Meenakshy, “Development and Implementation of a CAD system to predict the fragmentation of renal stones based on texture analysis of CT Images” Ph.D. Dissertation, Cochin University of Science and Technology, Kochi, May 2010.

[154] Preetha Basu, Bhadran. V, R. Gopikakumari “A New Algorithm to Compute Forward and Inverse 2-D UMRT for N - a power of 2”., 2012 Int. Conf. on Power, Signals, Controls and Computation (EPSCICON), Kerala, India pp. 1- 5, Jan. 2012.

PUBLICATIONS

INTERNATIONAL JOURNAL

- **M S Anish Kumar**, Rajesh Cherian Roy and R Gopikakumari "A New Image Compression and Decompression Technique based on 8x8 MRT", *ICGST International Journal on Graphics, Vision and Image Processing (GVIP)*, pp 51-53 Vol. 6 issue 1, July 2006

- **M. S. Anish Kumar**, Rajesh Cherian Roy, R. Gopikakumari," A New Transform coder for Gray Scale Images Using 4x4 MRT," *AEU, International Journal of Electronics and Communications*, vol. 62, no. 8, pp. 627-630, September 2008.

- **M. S. Anish Kumar**, Rajesh Cherian Roy, R. Gopikakumari," A Hybrid Coder for GRAY Scale Images Using 4x4 And 8x8 MRT,", *AEU, International Journal of Electronics and Communications*, Communicated.

- **M. S. Anish Kumar**, Preetha Basu, R. Gopikakumari," UMRT based Criteria for Quad-tree partitioning of Images,", *Image and Vision Computing*, Communicated.

- **M. S. Anish Kumar**, Preetha Basu, R. Gopikakumari," UMRT based Adaptive Block Size Transform Coder for Images Using Quad-tree partitioning”, *Journal of Visual Communication and Image Representation*, Communicated.

CONFERENCES

- Rajesh Cherian Roy, **M. S. Anish Kumar** and R. Gopikakumari “MRT: An alternate frequency domain representation” Proc. of IETE zonal conference on New Trends in Communication, Feb 2006 Kochi.

- Rajesh Cherian Roy, **M. S. Anish Kumar**, R. Gopikakumari “An invertible transform for image representation and its application to image compression,” *9th ISSPA 2007*, 12-15 February 2007, Sharjah, UAE.

INDEX

A

Arithmetic Coding · 17

B

Bit Allocation · 42, 71, 75
blocking artifact · 9
bpp · 12, 13, 82, 83, 84, 85, 88, 89, 90, 91, 92, 93, 94, 95, 96, 101, 102, 103, 104, 107, 108, 114, 115, 116, 117, 129, 130, 131, 132, 147, 163, 164, 165, 166, 167, 168

C

Classified Vector Quantization · 46
Codebook · 42, 135, 142, 143, 144, 158, 159, 160
compression · 1, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 23, 27, 29, 30, 31, 32, 33, 34, 35, 36, 38, 39, 41, 42, 43, 44, 45, 46, 48, 49, 50, 51, 52, 54, 55, 56, 57, 59, 60, 61, 62, 63, 64, 65, 67, 68, 69, 70, 71, 72, 74, 79, 80, 82, 83, 84, 85, 87,88, 89, 95, 96, 100, 101, 110, 111, 112, 117, 125, 131, 136, 138, 146, 150, 152, 166, 167

D

Discrete Cosine Transform · 11, 19, 29, 43, 72
Discrete Sine transform · 24, 54
Discrete Wavelet transform · 25
Distortion Measure · 13

E

Edge identification · 135, 152
entropy coding · 12, 21, 80, 127

F

Fractal Compression · 1, 39, 47, 66

H

homogenous · 127, 160
Huffman Coding · 16
Human Visual System · 6
Hybrid Coder · 111, 113, 114

I

image compressors. · 3
IMRT · 29, 81
Isometric Transformation · 135, 138

L

Lossless Compression · 1, 15
Lossy Compression · 1, 18

M

MATC · 46, 109, 117, 125, 127, 128, 129, 130, 131, 132, 133, 134, 135, 149, 150, 160, 161, 162, 163, 164, 165, 166, 167, 168, 170
MATC-CVQ · 46, 168
mixing · 5
MRT · 27, 28, 29, 44, 45, 46, 65, 66, 70, 71, 76, 77, 78, 79, 80, 81, 82, 83, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 98, 99, 100, 101, 102, 103, 104, 106, 107, 108, 109, 111, 112, 113, 114, 115, 116, 117, 118, 121, 125, 131, 132, 135, 138, 140, 141, 142, 145, 146, 147, 149, 150, 166, 167, 168

P

Peak Signal to Noise Ratio · 13
PSNR · 13, 36, 59, 68, 82, 83, 84, 85, 88, 89, 90, 91, 92, 93, 94, 95, 96, 101, 102, 104, 107, 108, 114, 115, 116, 117, 129, 130, 131, 132, 147, 163, 164, 165, 166, 167, 168

Q

Quad-Tree · 10, 45, 54, 55, 118, 122, 123, 124, 125, 126, 127, 129, 150, 153, 160, 161
Quantization · 1, 12, 19, 21, 35, 42, 45, 46, 47, 57, 68, 110, 117, 125, 135, 136, 138, 140, 149, 150, 151, 152

R

Redundancy · 1, 6, 7
RGB · 5, 95, 132, 168
Run Length Coding · 16

T

Templates · 86, 99
threshold · 21, 63, 75, 79, 80, 82, 83, 84, 85, 88, 89, 90, 91, 92, 93, 94, 95, 100, 101, 102, 104, 108, 112, 121, 124, 127, 129, 163
Transform Coding · 1, 20, 47, 49, 71, 72, 109, 125

U

UMRT · 45, 46, 66, 109, 118, 121, 122, 123, 124, 127, 131, 135, 140, 142, 150, 152, 153, 154, 157, 160, 161, 167

V

Vector Quantization · 19, 46, 150

About the Author

Anish Kumar M S
Research Scholar
Department of Electronics,
Tel: +91-484-2576418, Mob: +91-9447500698
Email: anishdoecusat@gmail.com
anishdoe@cusat.ac.in

Education:

- M.Sc Electronics Science
  Department of Electronics
  Cochin University of Science and Technology (2003)
  Score: 8.9/10 (CGPA) (First class with Distinction)

- B.Sc Computer Science
  School of Technology and applied sciences, Chuttippara, Pathanamthitta
  M G University (2001)
  Score: 73% (First class)

Achievements

- Qualified UGC NET in December 2012
- Awarded Junior Research Fellowship by Kerala State Council for Science Technology & Environment (KSCSTE) in the year 2005.
- Secured fifth rank for MSc Electronics from Department of Electronics, CUSAT(2003)

Research Experience:

Worked as a Research scholar, in the field of image processing and compression, in Department of Electronics, Cochin University of Science And Technology, from August 2005 to July 2013.

Computer Proficiency:

1. Programming Languages: C, C++, MATLAB, Visual Basic, VB.NET, HTML, Assembly Level Programming
2. Softwares & Packages: SigmaPlot, OrCAD, Proteus, MPLAB IDE, HiTech C Compiler, Adobe Photoshop, Corel Draw, Origin, MS Office,

In depth knowledge in Computer hardware, software and Operating systems