NEW APPROACHES TO IMAGE COMPRESSION

THESIS

SUBMITTED TO

THE UNIVERSITY OF CALICUT

IN FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF

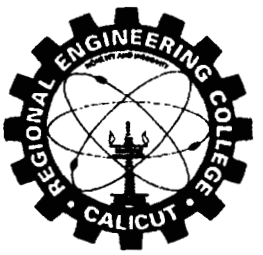
DOCTOR OF PHILOSOPHY

IN

ELECTRICAL ENGINEERING

BY

H.R. MAHADEVASWAMY



DEPARTMENT OF ELECTRONICS ENGINEERING

Regional Engineering College

CALICUT - 673 601, KERALA

DECEMBER 2000

CERTIFICATE

This is to certify that the thesis entitled " **New Approaches to Image Compression**" being submitted by **H.R.Mahadevaswamy** to the University of Calicut, for the award of the degree of **Doctor of Philosophy**, is a record of the bonafide research work carried out by him under our supervision and guidance. The results contained in this thesis have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Dr.P.Janardhanan
Professor
Department of Electronics Engineering
Regional Engineering College
Calicut – 673 601

Dr.Y.Venkataramani
Professor
Department of Electronics Engineering
Regional Engineering College
Calicut – 673 601

Date : 20.12.2000

Dedicated

To

My father-in-law

Late Sri B.V.Natarajappa

ABSTRACT

Still image compression is an important issue in internet, mobile communications, digital library, digital photography, multimedia, teleconferencing applications, etc. It would be obvious that the present and future applications in multimedia and other areas would focus on the problem of optimising storage space and transmission bandwidth. In many of these applications, big savings in terms of bit rate can be achieved by a tolerable loss of quality.

Research advances in fractal and wavelet theory have created a surge of interest and image compression is an important application that has benefited significantly from the fractal and wavelet theory.

Fractal Image Compression is a fairly nascent field of research. This method is based on the theory of Iterated Function Systems (IFS) and its performance relies on the presence of self-similarity between the regions of an image. Since most images possess high degree of self-similarity, fractal compression contributes an excellent tool for compressing them. However, fractal compression is an asymmetrical process, taking much longer time to compress an image than to decompress it. During encoding an image, for every range block the best matching domain block is searched among all domain blocks by performing a set of transformations on the blocks. This searching step is computationally expensive and takes long time. Two methods are proposed to reduce this searching time.

In the first method, while comparing the range with domain blocks, only partial distance is calculated. If that distance is more than the previously computed distance, then the domain block can be rejected before finishing the computation of the complete metric distance. This approach leads to reduction in computational time compared to other

methods. Hence encoding time is reduced with little degradation in quality. For still images, experiments show that at comparable quality levels this algorithm is faster than Fisher's quadtree based fractal image compression.

In the second method, classification technique based on local fractal dimension is proposed. To reduce the searching time in encoding process, only the 'best matching' domain blocks are selected in advance by classification technique based on local fractal dimension and then only the transformations on these blocks are computed. Domains and ranges are classified into a fixed number of classes according to local fractal dimension feature. For still images, experiments show that, this method speeds up with little degradation in quality.

The past and present research activities in image compression made use of block oriented principles. This is because of two reasons: first, the inherent architectural simplicity of block-oriented image coding makes very appealing for hardware realisation. Second, segmenting an image into blocks of equal size requires no overhead information for transmission/storing of the corresponding image partition. But, this fixed partition strongly neglect the image content. As a consequence, it leads to severe blocking artefacts at higher compression ratios. Instead, we can use region or object-oriented image coding which attempts to use a more natural model for image representation. This method assumes that certain connected regions in an image will be homogeneous with respect to some property and that set of regions can completely describe the image. The region-based partition of the image increases the compression ratio over the traditional block-based partition. Local fractal dimension based segmentation for partitioning an image is developed. Then transformations for each region is found. Chain coding is used to code

the boundaries between the regions. Better results are obtained compared to existing methods.

In many of the image storage/transmission applications, the data to be encoded is corrupted by noise. Images scanned from photographic films are corrupted by data dependent film-grain noise while scanning. The effects of Gaussian and film-grain noise on Fractal Image Compression algorithms are studied. Experimental results shows that noise degrades the performance of the algorithms.

Lossless image compression schemes are useful in applications such as medical imaging and satellite imagery, where loss of data is not tolerable. Integer to Integer wavelet transforms are built based upon the idea of factoring wavelet transforms into lifting steps. . Due to the non-adaptivity of the classical wavelet coding techniques the results heavily depend on the images considered. In the conventional methods of lossless image compression using integer wavelets transforms, one uses a set of well chosen filter coefficients to perform decomposition from fine to coarse scales. Commonly these coefficients do not change from one scale level to the next one. Since all the transformations at each level are performed independently, it is possible to use different filter-coefficients (which correspond to different scaling and wavelet functions) at every scale. But, the question arises, how to choose the best suited filters for different scales and directions? Adaptive integer to integer wavelet transform method is developed for lossless image coding. Filter library is used and then best filter is selected for each scale of transformations so the adaptivity can be achieved. This adaptive method is performing better for all images compared to existing methods.

Acknowledgements

As usual when a piece of work of this size comes to an end there are much more people to Thank than is practically possible. To those I have forgotten to mention, **Thank You First Of All.**

I wish to express my most sincere gratitude to my Research Guides Dr.P.Janardhanan, Professor, ECED and Dr.Y.Venkataramani, Professor,ECED for all the support, guidance, advice and constructive suggestions during my research work.

I wish to thank Dr.E.Gopinathan, HOD, ECED, for reviewing my work and help rendered during my course of research work.

I wish to thank Doctoral Committee members Dr.V.K.Govindan, HOD,CSED, Dr.K.P.Mohandas, Professor, EED, Dr.P.C.Subramanian, Assistant Professor,ECED, Dr.M.J.Jacob, Assistant Professor, Maths Dept., for reviewing my work and help rendered during my course of research work.

I owe special thanks to Mr.R.Suresh, Lecturer, ECED for his help at all stages of my work.

I would like to thank Dr.P.K.Bharathan, Dean(Academic) for providing me an opportunity to take up my research work here and help rendered during my course of work.

I would like to thank Dr.M.N.Neelakantan, Dean(Adm), Dr.N.Ganesan, Professor, CED , Dr.T.L.Jose, HOD, EED for their help during my research work.

It is my privilege to say thanks to Dr.M.P.Chandrasekharan, Principal for providing me all the facilities and treating me as one of the colleague during my research work.

I thank Dr.Elizebath Elias, Dr.Lillykutty Jacob, Mrs.P.S.Sathidevi and all staff members of ECED, and non-teaching staff of ECED for their co-operation and help during my research work.

I wish to thank Dr.V.S.Prasad , HOD, CED, Dr.B.N.Nagaraj, Professor, CED, Dr.D.R.Balasubramaniam, Professor, DASH, Mr.G.Veerashetty, Lecturer, MED and their families for kind help and co-operation during my course of work.

I wish to thank Mr.Rohit, Mr.SubbaReddy, Mr.Rajesh Mohoan and Mr.Mrudul Muralidaran students of REC who helped me at critical stages of my work.

I thank SIT, Tumkur in particular my friends in EED for their invaluable help and encouragement during my research work.

I am very much indebted to my parents, in-laws, brother-in-law's Mr.Manjunath, Mr.Pradeep for their help, encouragement throughout my research work.

I have to thank whole heartedly my wife Mrs..Anitha for her understanding and support throughout my research work.

Answer to my lovely daughter SUSHMA's everyday question " When you will stop going to college in night" is 'Thank you very much for your understanding, patience and sacrificing your personal priorities to cope up with my busy schedule.

H.R.Mahadevaswamy

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

CT	Computer Tomography	8
MRI	Magnetic Resonance Imaging	8
CCD	Charge Coupled Device	8
Hz	Hertz	9
HVS	Human Visual System	10
DB	Decibels	15
RMS	Root Mean Square	15
PSNR	Peak Signal To Noise Ratio	15
VQ	Vector Quantization	19
FFT	Fast Fourier Transform	21
DCT	Discrete Cosine Transform	21
JPEG	Joint Photographic Expert Group	22
IFS	Iterated Function System	22
PIFS	Partitioned Iterated Function System	22
HV	Horizantal Vertical	33
SNR	Signal to Noise Ratio	33
FIC	Fractal Image Compression	57
FD	Fractal Dimension	70
Lfd	Local Fractal Dimension	74
DCC	Derivative Chain Code	95
RBFIC	Region-Based Fractal Image Compression	97
DWT	Discrete Wavelet Transform	115
S+P	S-Transform + Prediction	122

PPP	Precision Property Preservation	124
ICFB	Integer Coefficient Filter Bank	125
LL	Low pass	120
LH	Lowpass Vertical	120
HL	Highpass horizantal	120
HH	Highpass diagonal	120

INTRODUCTION

H.R. Mahadevaswamy “New approaches to image compression ” Thesis.
Department of Electronics Engineering, Regional Engineering College ,
University of Calicut, 2000

Chapter 1

INTRODUCTION

1.1 Background

The use of digital images has increased at a rapid pace over the past decade. Photographs, printed text, and other hard copy media are now routinely converted into digital form, and the direct acquisition of digital images is becoming more common as sensors and associated electronics of very high quality are becoming available now. Many recent imaging modalities in medicine, such as magnetic resonance imaging (MRI) and computer tomography (CT), also generate images directly in digital form. Computer generated (Synthetic) images are becoming an additional source of digital data, particularly for special effects in advertising and entertainment. The reason for this interest in digital images is clear. Representing images in digital form allows visual information to be easily manipulated in useful and novel ways.

The amount of data associated with visual information is so large that its storage would require enormous storage capacity. Although the capacities of several storage media are substantial, their access speeds are usually inversely proportional to their capacity. Typical television images generate data rates exceeding 10 million bytes per second. There are other image sources that generate data at even higher rates. Storage and/or transmission of such data require large capacity and/or bandwidth, which could be expensive. Since a considerable degree of redundancy is present in the images and in many cases an image perceptually equivalent but not identical to the original image is acceptable,

a decrease in the size of the digital representation by more than an order of magnitude is possible by employing suitable image compression techniques.

Image compression reduces redundancy in image data in order to store or transmit only a minimal number of samples from which a good approximation of the original image can be reconstructed in accordance with human visual perception. Many compression methods have been developed. Among them transform coding, multiresolution coding, vector quantization, predictive methods, and other more recent schemes such as fractal image coding and wavelet image coding are important .

The proposed work focuses on fractal image compression, lossless image compression using wavelets and image dependent fractal image compression methods.

1.2 MOTIVATION FOR THE THESIS

Research advances in fractal and wavelet theory have created a surge of interest in applications like image compression. The investigation and design of computationally efficient and effective software algorithms for lossy and lossless image compression forms the primary objective of this thesis.

In fractal image compression, parts of an image is described with reference to other parts of the same image and, by doing so, the redundancy of piecewise self-similarity is exploited. Details on fractal compression are found in books of Barnsley and Hurd [31], Fisher [43] and Lu [39]. There are a number of problems to be solved in fractal image compression to make the process viable and more efficient.. They are:

i) Complexity reduction: How can the process of finding a good fractal code be speeded up?

ii) Coding efficiency : How can the parameters of a fractal code be coded in the most efficient way? How to select the parameters in a rate distortion optimal fashion?

iii) Partitioning Methods : Does the flexibility in partitioning scheme lead to improved results in the rate-distortion sense?

iv) Class of transforms : Is there any gain in using wider class of transforms than just block based operators?

v) Decoding complexity : How can the decoding process be speeded up?

In this thesis efforts are made towards solving the problems ii) & iii). Here speed up of the Fractal Image Compression is addressed and adaptive partitioning method is developed. Study of these methods for noisy image is also considered.

A lot of work has been done in the area of wavelet based lossy image compression. However, very little work has been done in lossless image compression using wavelets. The main reason for this is the lack of adequate floating-point precision of the various wavelet filters. This deficiency causes a finite precision error after the reconstruction of the signal from the transformed signal. Very few filters can be "Integerized" (rounding is one way of converting the floating points to integers) to make the transform, a reversible one. The integer to integer wavelet transforms based upon the idea of factoring wavelet transforms into lifting step are used for lossless image compression. To determine these transforms only integer addition and bit-shift operations are needed. But most of these integer wavelet transforms does not perform well on all the images. In this thesis work, adaptive integer wavelet coding for lossless image compression is developed to obtain better results compared to existing methods.

1.3 Outline of the thesis:

The outline of the rest of the chapters in this thesis is as follows:

In chapter 2, overview of image compression, various methods of compressing images, scope of image compression and applications are given. Also introduction to fractal image compression, wavelet image compression and region based coding is given.

Chapter 3 consider various aspects of fast fractal image compression. Here, various techniques for partitioning the image and searching strategies are explained. Also, decoding method is explained. Fast methods for fractal image compression are developed. Partial distance based fractal image compression has been developed. The algorithm, implementation and results are presented. Another method based on , classification of blocks using fractal dimension in fractal image compression, is explained. Details of the algorithm, implementation and results are given.

Chapter 4 gives Region-Based Fractal Image Compression based on fractal dimension . Here image dependent partition (segmentation) is explained. Then comparison is made with image independent partitioning in fractal image compression. In Region-Based Fractal Image Compression fractal dimension based merging of blocks is explained. The algorithm, implementation and results are given in this chapter. Also coding the partition information using derivative chain code is explained.

Chapter 5 is a study of Fractal Image Compression for noisy images. The noisy images corrupted by data dependent film-grain noise compressed using various Fractal

Image Compression is explained. The comparison is made with the original and noisy images.

In Chapter 6, Lossless Image Compression using Wavelets is explained. Various integer wavelet transforms for lossless image compression are briefly explained. Adaptive integer wavelet lossless image compression is given in this chapter. The method, Algorithm, Implementation and Results of this methods are explained.

Chapter 7 contains conclusions and future work in this area.

Appendix A contains set of test images used in the work.

Appendix B contains Contraction Mapping Theorem and Collage Theorem.

Appendix C describes the computation of RMS error for digital image.

1.4 CONTRIBUTIONS OF THE THESIS

The work reported is aimed at developing computationally efficient and effective algorithms for lossy and lossless still image compression using fractal and wavelet techniques.

The work is particularly targeted towards fractal image compression with an idea to minimise the computational requirements to achieve good reproduction image quality. In this direction the following methods are developed.

1. Partial distance based fractal image compression which reduces the computational complexity.
2. Local fractal dimension based classification in fractal image compression which

speeds up the encoding step in fractal image compression.

3. Highly image adapting partitioning scheme for fractal image compression is developed which is better than existing methods.

4. In addition, the above methods are tested for various noisy images.

5. The remaining part of the thesis is devoted to the development of lossless image compression methods using wavelets. In this work, existing methods are critically reviewed and the adaptive integer wavelet coding method for lossless image compression is developed. This gives better results compared to existing methods.

IMAGE COMPRESSION

H.R. Mahadevaswamy “New approaches to image compression ” Thesis.
Department of Electronics Engineering, Regional Engineering College ,
University of Calicut, 2000

Chapter 2

IMAGE COMPRESSION

2.1 Introduction

The ability to see is one of the truly remarkable characteristics of living beings. It enables them to perceive and assimilate in a very short time an incredible amount of knowledge about the world around them [1]. Pictorial information provides much more detail than words; a fact which can be precisely summarised by the saying that "a picture is worth a thousand words".

The downside to representing images in digital form is that a large number of bits are required to represent even a single digital image. A single page scanned at 300 dots per inch and quantized to two levels generates more than 8,000,000 bits (1,000,000 bytes) of data [2] . Thus, for a one-page high quality picture typically takes megabytes of memory to store. For instance, more than 25 gigabytes of data are required to represent the Encyclopaedia Britannica, which contains 25,000 pages, in digital form..

Due to the above fact it becomes necessary to find efficient representations for digital images in order to : (1) reduce the memory required for storage (2) improve the data access rate from storage devices, and (3) reduce the band width and/or the time required for transfer across communication channels. The branch of digital image processing that deals with this problem is called image compression (or sometimes as picture coding) [3]. Image compression is concerned with minimising the number of bits required to represent an image.

Perhaps the simplest and most dramatic form of image data compression is the sampling of band limited images, where an infinite number of pixels per unit area are reduced to one sample without loss of information (assuming an ideal low-pass filter is available). Consequently, the number of samples per unit area is infinitely reduced.

Applications of image data compression are primarily in transmission and storage of information. Image transmission applications are in broadcast television, remote sensing via satellite, military communications via aircraft, radar and sonar, teleconferencing, computer communications, facsimile transmission, and the like. Image storage is required for educational and business documents, medical images that arise in computer tomography (CT), magnetic resonance imaging (MRI) and digital radiology , motion pictures, satellite images, weather maps, geological surveys and many other applications [4].

2.2 Need for Image Compression: Examples

The need for image compression can best be illustrated by considering few storage and/or transmission examples based on current or proposed technologies.

1. Storage: An electronic still camera using a single-chip CCD sensor (with a color filter array) requires about 484 x 768 pixels x 8 bits per pixels, resulting in about 3 x 10^6 bits/image. A 1-megabyte solid-state memory card used in the camera could store three images in uncompressed form. Obviously, this is neither convenient nor cost-effective. If an 8:1 compression ratio (defined as the ratio of the number of bits used for the original image to the number of bits for the compressed image) can be achieved, the memory card can then hold 24 images.

2. Transmission: A low resolution facsimile (bilevel) image (example: an 8 1/2 inch by 11 inch document scanned at 200 pixels/inch horizontally and 100 pixels inch vertically) transmitted using a 4800 bits/sec. modem over ordinary telephone lines requires approximately 6.5 minutes for transmission in uncompressed form. With compression, current fax compression standards allow the same document to be sent usually in less than 30 seconds. For a multiple document, the saving in total transmission time is significant.

3. Storage/Transmission: A full motion color video signal (consumer TV quality) requires (512 x 480 pixels/frame x 24 bits/pixel x 30 frame/s), i.e., about 180 x 10^6 bits for 1 second of video. A CD-ROM with a 650-megabyte storage capacity can only hold about 30 seconds of full motion video. Furthermore, the CD-ROM data transfer rate is about 150 Kilobytes/sec. This means that it would take 1.2 hours to play back the 30 sec. of uncompressed video. Appropriate compression techniques would not only allow a longer video sequence to be stored, but it would also allow more video data (in compressed form) to be transferred per second from CD-ROM. If the compression ratio is high enough (around 150:1), it could potentially be played back in real time [3].

The mere process of converting an analog video signal into a digital signal results in increased bandwidth requirements for transmission. For example, a 4-MHz television signal sampled at Nyquist rate with 8 bits per sample would require a bandwidth of 32 MHz when transmitted using a digital modulation scheme, such as phase shift keying, which requires 1 HZ per 2 bits. Thus, although digitized information has advantages over its analog form in terms of processing flexibility, random access in storage, higher signal – to- noise ratio for transmission with the possibility of errorless communication, and so on, one has to pay the price in terms of this eight fold increase in bandwidth. Data compression

techniques seek to minimise this cost and sometimes try to reduce the bandwidth of the digital signal below its analog bandwidth requirements.

2.3 Achieving Compression: Redundancy and Irrelevancy

If each pixel value represents a unique and perceptually important piece of information, it would be difficult indeed to compress an image. Fortunately (at least from the standpoint of compression), the data comprising a digital image or sequence of images are often redundant and/or irrelevant. Redundancy relates to the statistical properties of images, while irrelevancy relates to the observer viewing an image. Redundancy can be classified into three types (1) Spatial (due to the correlation between neighbouring pixels in an image) (2) Spectral (due to the correlation between color planes or spectral bands) and (3) Temporal (due to the correlation between neighbouring frames in a sequence of images). Similarly, irrelevancy can be classified as spatial, spectral, and/or temporal in nature, but the key issues in this case are the limitations and variations of the human visual system (HVS) when presented with certain stimuli under various viewing conditions. Ideally, an image compression technique removes redundant and irrelevant information and then efficiently encodes what remains. Practically, it is often necessary to throw away some both non-redundant and relevant information to achieve the necessary degree of compression [3].

2.4 Image Compression Techniques

There are two main classes of image compression:

1. Lossless Technique

2. Lossy Technique

2.4.1 Lossless Technique

The original image may be perfectly reconstructed and as the name suggests, exact reconstruction of coded image data. This is also known as bit preserving or reversible compression.

Lossless image compression (reversible) is used where image must be reconstructed exactly to its pre-compressed form. High-fidelity images generated from studio quality video, medical images, seismic data, satellite images and images scanned from manuscripts for preservation purposes typically demand lossless encoding. Unfortunately, only modest compression ratios (an average of 2:1 for a single-band image) are possible with lossless compression.

The classical lossless compression schemes are based on tree-based codes that represent a large class of variable-length encoding schemes such as Huffman codes [4], universal codes of Elias [6], run length coding [4], lossless predictive coding [2] etc. The tree based compression methods have been used for large scientific and text files as well as image data and usually yield compression ratios in the range from 2 to3. The redundancy in digital image representation can be classified into two categories: local and global. Local redundancy corresponds to the coherence, smoothness or correlation in the image data. This is due to the fact that the gray level values within a neighbourhood vary gradually rather than abruptly. Global redundancy could be attributed to the repetition of patterns within an image.

Image compression algorithms eliminate or reduce local redundancy by representing the neighbourhood in a compact form. For example, in run length coding [5] scheme, runs of pixels with the same gray level are replaced by the gray level and its run length. In

compression schemes based on differential pulse code modulation [7], [8] and predictive coding [2] methods, the next input is predicted based on the digitally coded past.

Global redundancy can be eliminated or reduced by encoding the repeated patterns by using suitable coding methods. Lempel-Ziv scheme [9], [10] searches for repeated linear patterns within a fixed size window and replaces the repeated patterns by suitable pointers to the previous occurrences. Arithmetic coding [11] and Huffman [4] schemes exploit the skewness in the distribution of patterns for optimal encoding of the given data.

Lossless image compression using wavelets

There are some algorithms available for lossless image compression [5], [7], [8], [2]. However, these produce a single resolution after transformation and these retard the progressive transmission or recovery. The wavelet transform is one of the best tools for multiresolution analysis and is also ideal for progressive transmission. Therefore integer reversible wavelets are created for lossless image compression [12], [13], [14], [15] and [16]. Further details of this method are discussed in Chapter 5.

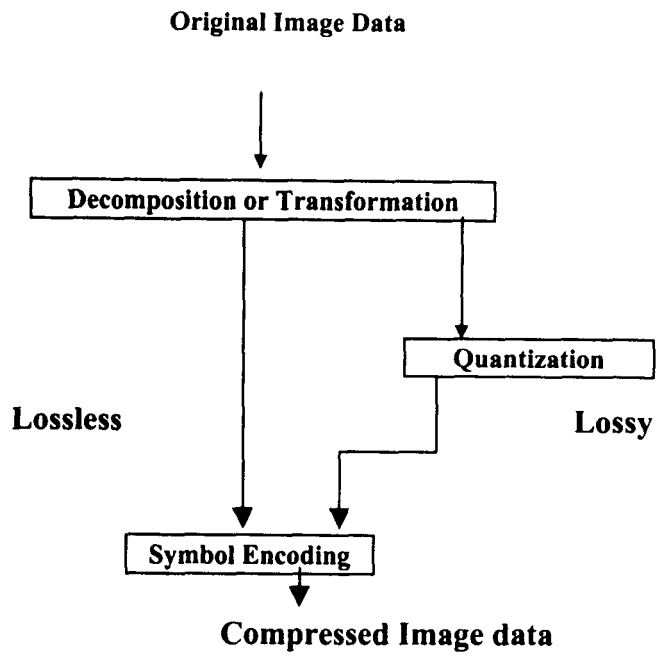
2.4.2 Lossy Technique

In lossy compression methods are applied which extract the essential information from the original, discarding the remainder. The extracted information is coded compactly, and when it is to be viewed the decompressor transforms it into a appropriate reproduction of the original. In lossy compression (also known as irreversible compression), the reconstructed image contains degradations relative to the original image. However, under certain conditions, these degradations may not be visually apparent (sometimes called visually lossless compression). Lossy image compression or coding is based on the concept

of compromising the accuracy of the reconstructed image in exchange for increased compression. If the resulting distortion (which may or may not be visually apparent) can be tolerated, higher compression ratios can be obtained with lossy techniques.

2.5 Compression System Components

Fig. 2.1 shows the general compression scheme which consists of three basic components : (1) image decomposition or transformation (2) Quantization, and (3) Symbol encoding. The image decomposition or transformation is usually a reversible operation and is performed to eliminate the redundant information, or more generally, to provide a representation that is more amenable to efficient coding. Both lossless and lossy techniques are used in this stage. Next stage, quantization, is a many-to-one mapping found only in lossy techniques, and invariably leads to introduction of errors at this stage.



Figure(2.1) General Compression framework

The type and degree of quantization has a large impact on the bit rate and the reconstructed picture quality of a lossy scheme. In essence, quantization can be viewed as a control knob that trades off image quality for bit rate. The final stage, symbol encoding, is a means for mapping the symbols (values) resulting from the decomposition and/or quantization stages into strings of 0's and 1's, which can then be transmitted or stored. It should be noted that the three components shown in Fig. 2.1 often mutually interact, and their joint optimisation is a complicated task [3].

2.6 Measurement of Image Reproduction Quality

With lossless compression the reproduction is identified to the original, and hence, quality is not an issue. In the case of lossy compression, however, the reproduction is only an approximation to the original image. Measurement of quality is thus a central issue with lossy compression, and in particular in video and image compression. There are basically two approaches to estimate quality. The first is to rely on panels of human beings to compare reproductions with the original. This is a direct method, but subject to a variety of difficulties:

- Human judgement may be significantly affected by the presence of system introduced errors or artefacts.

- Human judgements vary from one person to another, and from one time to another.

The second approach is to compute a numeric comparisons between reproduction and original, and use this as an objective measurement of quality. In carrying out the evaluation as part of the research in this thesis, numerical evaluation is adopted. The Peak Signal to Noise Ratio (PSNR) is used to measure the quality of the images.

It is defined as

$$PSNR = 20 \log_{10}(b / rms),$$

Where b is the largest possible value of the signal (b is 255 for 8-bit digital image), and rms is the root mean square error between two images. The PSNR is given in decibels (dB) which measures the ratio of peak signal and the RMS difference between two images [43].

2.7 Main Parameters in Compression system

Figure 2.2 shows the main parameters relating the compression ratio and the processing power requirement in a three dimensional co-ordinate system. Quality is highly dependent on a number of factors, and two of the most important are compression ratio and available computational resources. It is not surprising that quality is inversely related to the compression ratio since a larger compression ratio implies representing a picture with a smaller amount of data. This inevitably implies a greater degree of approximation, and hence a loss in quality. With a larger computational resource it is usually possible to

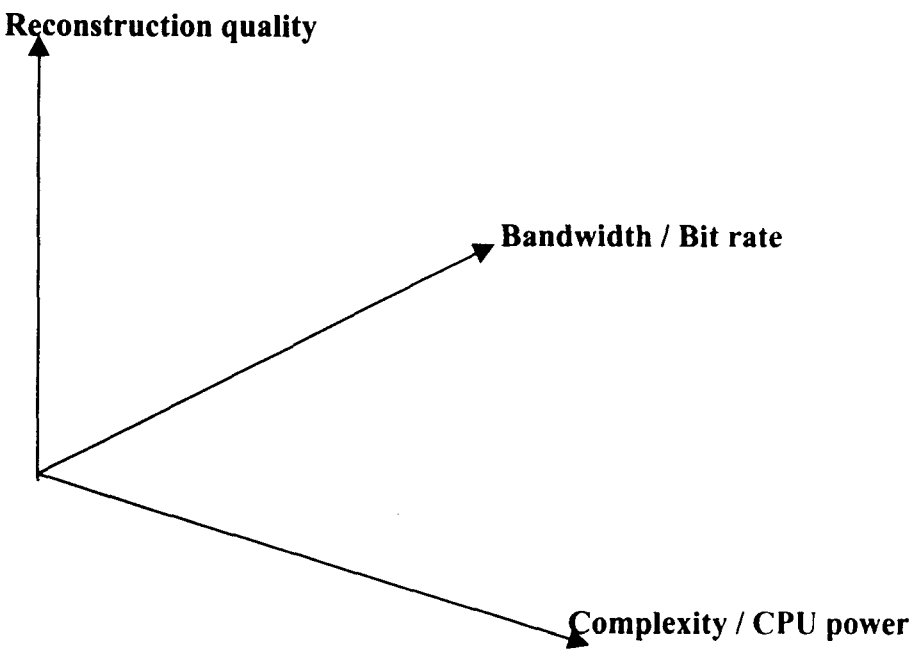


Figure (2.2) Design parameter space for compression system

Produce a higher quality reproduction (at the same compression ratio) [89]. This is because of two factors :

1. With more computational resources there is less need to approximate the data processing.

2. Lossy compression is inherently an optimisation process attempting to select the best representation from a number of alternatives.

2.8 Spatial Frequency Image Compression Technique

In spatial frequency image compression technique, transforms are used to represent an image.

2.8.1 Transforms used in Image Compression

A transform represents a patterned area of an image by a set of waves. A basis function describes the basic wave shape. After transformation, the representation is a set of coefficients representing the amplitudes and phases of the waves which combine to represent the image. Transforms are popular in image compression because not all transform coefficients need to be stored/transmitted to achieve good image reconstruction, and coded coefficients do not always need the maximum accuracy.

The aim of the transformation to the spatial frequency domain is to calculate a set of coefficients from the input pixels of varying amplitude, each of which contributes to all the pixels in its area. If the smaller coefficients are eliminated, or represented with reduced accuracy this will have little effect on the accuracy of the pixel representation when inverse transformation is carried out.

In transform coding, a reversible, linear transform (such as the Fourier transform) is used to map the image into a set of transform coefficients, which are then quantized and coded. For most natural images, a significant number of the coefficients have small magnitudes and can be coarsely quantized (or discarded entirely) with little image distortion. Any of the transformations of [2, Chapter 3] can be used to transform the image data.

Fig. 2.3 shows a typical transform coding system. An N x N input image is first subdivided into subimages of size n x n, which are then transformed to generate (N^2/n^2) subimage of size n x n. The goal of the transformation process is to decorrelate the pixels of each subimage, or to pack as much information as possible into the smallest number of transform coefficients. The quantization stage then selectively eliminates or more coarsely quantizes the coefficients that carry the least information. These coefficients have the smallest impact on reconstructed subimage quality. The encoding process terminates by symbol coding the quantized coefficients.

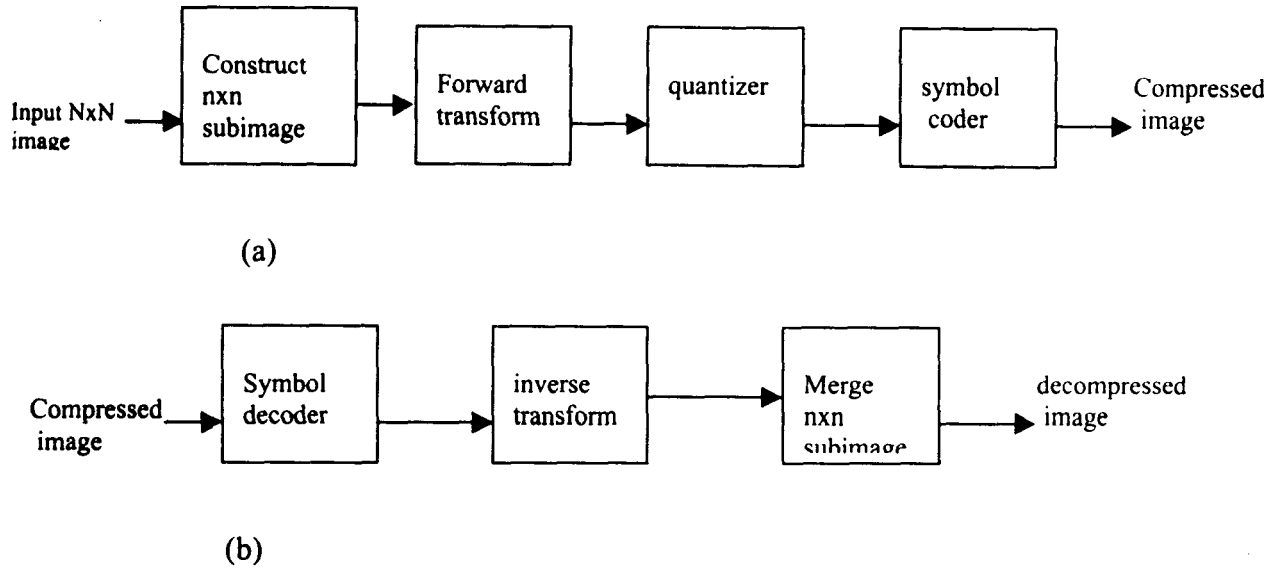


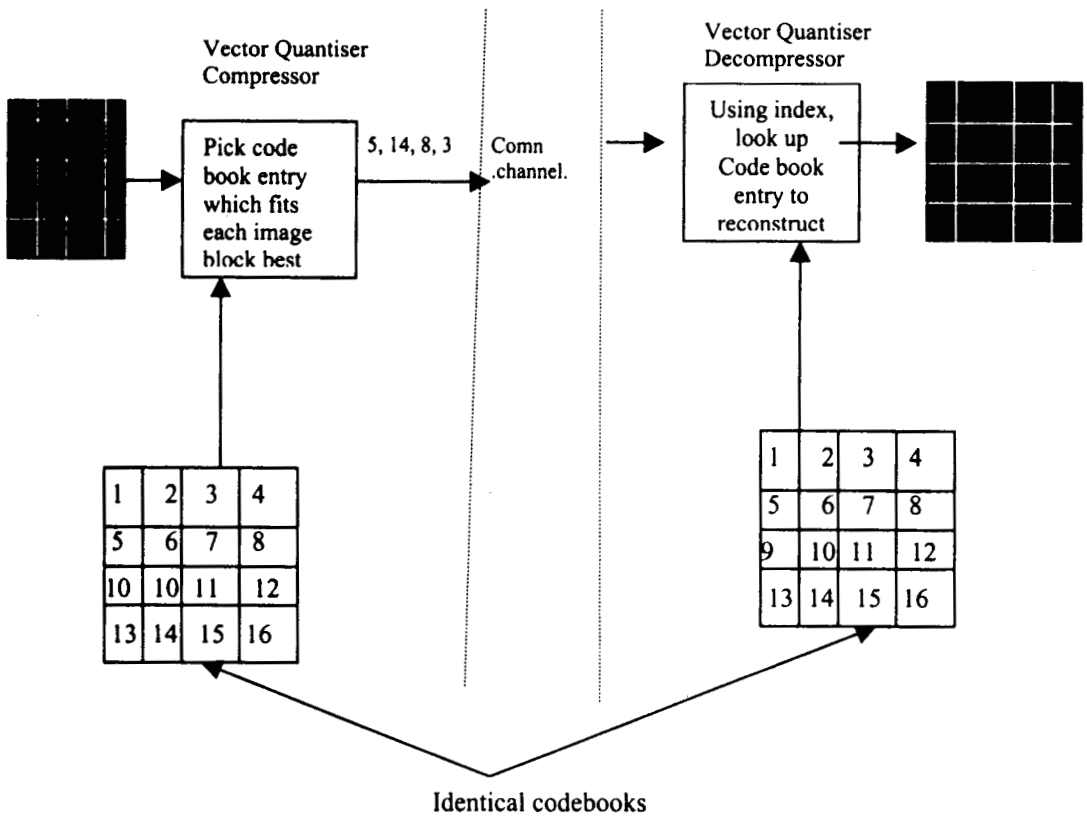
Figure (2.3) A Transform Coding System (a) encoder (b) decoder

Karhunen - Loeve (KLT), discrete Fourier (DFT), discrete cosine (DCT), Walsh-Hadmard (WHT) and various other transforms can be used in transform coding of images [2]. 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.

2.9 Real Space Image Compression Technique

In real space approach, parts of the picture are approximated by a collage or tessilation of tiles each of which is one of a generic set of patterns.

2.9.1 Vector Quantization (VQ)Techniques



Figure(2.4) Vector Quantization (VQ) Image Compression System

Vector quantizer can be used to encode image data in a lossy manner. The principle of a Vector Quantization (VQ) image compression system is illustrated in Figure 2.4. The image is first segmented as a number of individual blocks, and these are matched by the compressor against its codebook of typical block patterns. The compressor then treats the index number of the selected best-matching patch as the representation for the image block. The mean square error (MSE) distortion measure is used to compare two blocks. When the compressed image is to be reconstructed, the decompressor selects the appropriate patch of pixels from its identical codebook as the representation for the corresponding part of the image. At a given quality higher compression ratio can be obtained with VQ methods. A major problem for full search Vector Quantization algorithms is the expensive computational time required to select the optimal reproduction code word. A wide variety of vector quantization techniques for image compression have been reported in the literature [2], [24], [25], [26], [27] and [28].

2.9.2 Fractal Image Compression:

A recent approach to image compression (and one that has generated much interest) involves using fractals to represent the structure of images. Fractals can be loosely defined as shapes that are irregular (compared to lines, circles, etc., of standard Euclidean geometry) and that have the interesting properly of self-similarity. A self-similar object is one that looks roughly the same regardless of the scale at which it is viewed. Fractals have been found to be good models to describe natural objects, and their usefulness in image compression lies in the fact that images can be described very concisely by a fractal code. A fractal code merely consists of one or more simple transformations of a particular type that when applied iteratively to an arbitrary initial image will converge to the desired

image. However, the trick in fractal compression is in finding the necessary transformations for a given image.

Much of the initial knowledge of fractal based compression is due to Barnsely and Sloan [29-31]. Their claims of extremely high compression ratios (while supplying some-what limited technical descriptions of the exact process) , have left many with doubts regarding the validity of fractal-based compression. However, Jacquin has described a compression method that uses fractals in a relatively straightforward way [32] , [33]. More information on fractal image compression are given in Chapter 3.

2.9.3 Region Based Fractal Image Compression:

To date, fractal coding has progressed only as far as other non-intelligent compression schemes, which make no assumptions about the data they are handling. Region-based image coding schemes, the so called second generation techniques, have gained much favour in recent years. For still picture coding, they can increase the compression ratio, while maintaining adequate image quality [34].

2.10 Subband And Wavelet Coding:

In subband coding, an image is first filtered to create a number of subimages representing various spatial frequency bands of the original full-band signal. Since each of these subband images has a reduced band width compared to the original image, they may be down sampled. This process of filtering and down sampling (called the analysis stage) does not produce any compression in itself since the total number of samples remains the same. However, the subband images can be coded more efficiently than the original full-band image, typically using different bit rates or even different coding techniques for each

subband, thus taking advantage of the properties of each subband. Reconstruction is achieved by upsampling the decoded subbands, applying appropriate interpolation filters and adding the results together [3], [19].

Pictures are non-stationary in frequency and spatial content and hence data representing picture content may be anywhere in the actual picture. They also spread across the spatial frequency domain with no fixed distribution. Significant visual information in pictures are sharp transitions such as edges, but features or changes in the background of a picture may also be important. Wavelets are suited for extracting this kind of information spread across space because of its localisation properties. If wavelets are used in a multiresolution analysis, the representation of the picture at different resolutions enables determination of spatial frequency as well as spatial localisation in the wavelet domain. Image coding methods that use wavelet transforms can successfully provide high rates of compression while maintaining good image quality and have generated much interest in the scientific community as competitors to either classical or modern image coding methods. Wavelet image coding is closely related to many long-known image coding techniques. Basically the fast wavelet transform is nothing but a subband coding algorithm using specially designed filters. Further more wavelet coding can be viewed as some sort of multiresolution and pyramidal coding and can be classified as a sort of transform coding using wavelet transform instead of FFT or DCT. A wide variety of wavelet based image coding schemes have been reported in the literature [18], [19], [20], [21], [22], [23], [12], [13], [14], [15], [16], [17].

FAST FRACTAL IMAGE COMPRESSION

H.R. Mahadevaswamy “New approaches to image compression ” Thesis.
Department of Electronics Engineering, Regional Engineering College ,
University of Calicut, 2000

Chapter 3

FAST FRACTAL IMAGE COMPRESSION

3.1 Introduction

Standard methods of image compression are of several types. The most popular method currently used relies on eliminating high-frequency components of the signal by storing only the low-frequency Fourier coefficients. This method uses a discrete cosine transform (DCT) [35], and is the basis of the JPEG standard, which comes in many incompatible forms. Another method, called vector quantization [24], uses a building block approach, breaking up images into a small number of canonical pieces and storing only a reference. While JPEG is the industry standard for image compression technology, there is ongoing research in alternative methods. Currently there are at least two exciting new developments: wavelet based methods and fractal image compression. Fractal image compression is a fairly nascent field of research. Barnsley and Sloan [31], [30] were the first to propose compression of specially generated images using iterated function systems (IFS). The first practical and automated implementation of fractal image compression was done by Jacquin [33], who used block-based partitioned iteration function system (PIFS) to compress images. Fractal image compression, based on concepts of iterated function system (IFS) can be seen as a kind of vector quantization and multiresolution coding. The method consists of partitioning blocks (vectors) and approximates each vector by a transformed code book block derived from the image itself [33]. Each transform, described by a linear term and a translation term, maps a block onto another block with a different resolution and composes the coded information. A number of papers on fractal

image compression have been published by researchers,since the original idea of Barnsley and Sloan in 1988 [30] was implemented in a fully automated algorithm by Jacquin in 1989 [33].

The goal of this chapter is to explain fractal image compression and to review the work done by various researchers in the area of fractal image compression.

3.2 Basics of Fractal Image Compression

Deterministic fractals have the intrinsic property of having extremely high visual complexity which requires less number of bits for representing a given information. They can be described and generated by simple recursive deterministic algorithms [36]. They are mathematical objects with a high degree of redundancy in the sense that they are recursively made of transformed copies of either themselves or parts of themselves.

3.2.1 Fractals – a brief introduction

The term fractal was first used by Benoit B. Manderlbrot to designate objects that are self-similar at different scales [36]. Such objects have details at every scale. Unfortunately, a good definition of the term fractal is elusive. It is best to regard a fractal as a set that has properties such as those listed below, rather than to look for a precise definition.

If we consider a set F to be fractal, it will have some of the following properties:

1. F has detail at every scale
2. F is (exactly, approximately, or statistically) self-similar.
3. The 'fractal dimension' of F is greater than its topological dimension.

4. There is a simple algorithmic description of F.

Self-similarity and Self-affinity:

The property of objects whereby magnified subsets appear similar or identical to the whole and to each other is known as self-similarity. It is a characteristic of fractals and sets them apart from Euclidean shapes which generally are smoother. Thus fractal shapes are self-similar and independent of scale or scaling and possess no characteristic size. Euclidean shapes may be described by a simple algebraic formula where as fractals are generally constructed using a recursive algorithm suited for computers.

The non-uniform scaling, where shapes are (statistically) invariant under transformations that scale different co-ordinates by different amounts, is known as self-affinity.

3.2.2 Mathematical Principle Behind Fractal Image Compression:

Fractal image compression or coding exploits the piecewise self-similarity of the image. The basic idea is to construct a contractive operator ω in the metric space X of digital images, for which the image to be encoded is the unique fixed point x_o.

For this, the class of affine operators is preferred because of their particularity to be 'nearly linear', which makes them easy to analyse. The fixed point x_o of such an operator :X $\omega \rightarrow$X verifies

$$\omega x_o = Ax_o + b \qquad (3.1)$$

$= x_o$, where A is a linear operator and b belongs to x.

Real-world images are usually not self-similar (except for contrived examples) and it is impossible to find an operator that maps a whole image x onto another image x_o such as x equals x_o.

However, real-world images present piecewise self-similarity, i.e, parts of the image resemble other parts. Starting from this observation, one can define an operator as the sum of piecewise mappings on X. A solution is to decompose the image into N non-overlapping range blocks R and into M different domain blocks D and define the operator ω as

$$\omega_x = \omega\left(\bigcup_{i=1}^{N} R_i\right) = \bigcup_{i=1}^{N} \omega_i(D_i) \tag{3.2}$$

The notation $D_i = x|D_i$ denotes that the image x is restricted to the domain part D_i and ω_i denotes the transformation mapping the domain block Di onto the range block R_i. The blocks R_i and D_i can have different sizes and shapes. The number M of blocks D_i is less than, greater than, or equal to N and therefore the blcoks D_i can be overlapping and picked only from parts of the image.

An additional constraint that must be imposed on the operator ω is that it is eventually contractive. ω is contractive if there exists a constant s<1 such that $d[\omega(x),\omega(y)] \leq s.d(x,y)$, $\forall$x,y, $\in$ X, where d is a given distance measure and s is called contractivity of ω. ω is said to be eventually, contractive (at the k^{th} iterate) if there exists a constant k such that k^{th} iterate of ω (noted ω^{ok}) is contractive.

If this is the case, the operator ω has a unique fixed point x_0 obtained with

$$X_o = \lim_{k \to \infty} \omega_x^{ok}$$

for an arbitrary x. The collage theorem [29] proves that if the distance between an image x and its transformation by the eventually contractive operator ω is small, then the distance between the image x and the fixed point x_o of ω will also be small. It provides a boundary given by:

$$d(x, x_o) \leq \frac{1}{1 - S_k} \cdot \frac{1 - S_1^k}{1 - S_1} \cdot d(x, \omega_x) \tag{3.3}$$

where S_1 is the Lipschitz constant of ω, k is the iterate number under which the operator ω becomes contractive, and S_k is the contractivity of ω^{ok}. These results were initially published in [37], [38].

3.3 Overview of Fractal Image Compression

Fractal image compression is a new technique and has already received a great deal of attention[43]. The most significant advantages of fractal image compression are:

(i) High reconstruction quality at low coding rates

(ii) Fast decoding

(iii) Resolution independence i.e.; an encoded image may be decoded at a higher resolution than the original.

Fractal image compression allows fast decoding, but encoding is very slow. The merits and drawbacks of fractal image compression in comparison to JPEG and other methods are outlined in [43], [30], [39].

Fractal image compression is based on the observation that real-world images in general are rich in affine redundancy. That is, under suitable affine transformations, larger blocks(domain blocks) of the image look like smaller blocks (range blocks) in the same image. The encoding process consists of finding an affine transformation for every range block with domain block, which fits best in the sense of the used image metric. These affine maps give a compact representation of the original image and are used to regenerate that image, usually with some amount of loss.

Fractal image compression is based on the concepts and mathematical results of iterated function systems (IFS). Several researchers have taken up the challenge to design an automated algorithm to solve the inverse (i.e. the encoding) problem using the basic IFS method and its generalizations. In a way, the work by Jacquin [33] and his subsequent papers broke the ice for fractal image compression providing a starting point for further research and extensions in many possible directions. A literature survey of fractal image compression reveals that, the following main subjects are addressed so far :

1. The partitioning of the image into ranges

2. Encoding: Choice of the domain pool, including several fixed basis blocks and even several image domain blocks for the code of a range, choice of the transformations defining the operator.

3. Classification methods for the complexity reduction of the encoding step: based on image values and intensity variance, clustering of domains, fast algorithms from computational geometry to solve nearest neighbor problems.

4. The decoding: standard iteration versus fast hierarchical or direct numerical.

These aspects were studied in order to obtain the best compromise in the key issues of every image compression scheme, namely:

1. Image fidelity

2. Compression ratio

3. Time complexity of the encoder/decoder.

3.4 Implementation of Fractal Image Compression :

The three main issues involved in the design of a fractal block coding system are : (1) the partitioning of an image (2) the choice of a measure of distortion and (3) the specifications of a finite class of contractive image transformations defined consistently with a partition and scheme for the quantization of their parameters.

3.4.1 Block based fractal image compression :

The encoding algorithm of Jacquin [33] is based on the observation that a real-world image is 'affine redundant'. It exploits the partial self-transformability of the image and is given below:

The image is partitioned into non-overlapping square blocks of size r x r called range blocks. The same image also houses domain blocks which are twice the size of the range blocks and overlap such that a new domain block starts at each pixel.

To find the PIFS for the image, for each range block, all the domain blocks are considered in sequence. Each of these domains is reduced spatially and in the gray scale domain to the size of the range block. Then the mean-squared error between the blocks is computed. The domain block with minimum error is chosen and its transformation to the range block is stored. For comparison, the domain blocks are also oriented along eight symmetries. All these transformations together make up the required PIFS.

3.4.2 Tools

The affine operator ω_i is defined by

$$\omega_i \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & s_i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \\ 0_i \end{pmatrix}$$

$$= A \begin{pmatrix} x \\ y \\ z \end{pmatrix} + b \qquad (3.4)$$

where A is the linear operator of ω_i and b is a translation vector.

A spatial transformation defined by the coefficients a_i, b_i, c_i, d_i, e_i, and f_i maps the coordinates (x,y) of the block D_i pixels to the pixels in the block R_i and a gray scale

transformation (in the z direction) defines how the pixel gray values are transformed. This is written as

$$v_i(z) = s_i.z + o_i \tag{3.5}$$

Fractal image coding requires the definition of a similarity measure between the image block R_i and the transformed image block D_i. For this, the simple L_2-metric, also called root mean square (RMS) distortion measure, is preferred because it is easily computed and fits well with visual perception for relatively small block sizes. It is defined as the square root of the sum over the range block R_i of the squared differences of B pixel values r_i and d_i, i.e.

$$d(R_i, D_i) = \left[\sum_{i=1,B} (d_i - r_i)^2 \right]^{\frac{1}{2}} \tag{3.6}$$

3.5 Fractal Image Compression – Encoding :

Encoding of an image in Fractal image compression consists of :

(i) Partitioning the given image into ranges and domains, (ii) search for an appropriate domain for each range (iii) Find an affine transformation that adjusts the intensity values in the domain to those in the range. Fractal image compression is capable of yielding comparative rate-distortion curves; however it suffers from long encoding times. Therefore, many efforts have been undertaken to speed up the encoding process. Most of the techniques attempt to accelerate the searching either using classification of domains and ranges or by using fewer domains [88]. In this section, several techniques proposed by

various researchers for partitioning, searching and finding an affine map are described briefly.

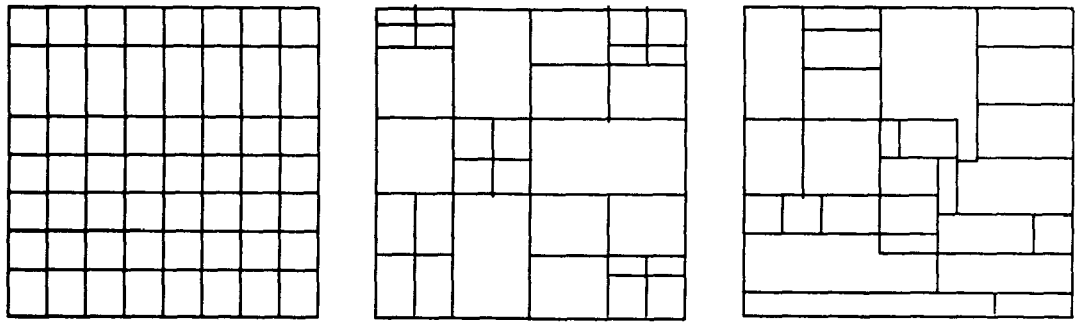
3.5.1 Partitioning of images :

In fractal image compression the image to be coded is partitioned into blocks called ranges. Each range is approximated by another part of the image called domain. Finding a partitioning that minimizes the approximation error while not exceeding a given bit-rate is a difficult problem in fractal image compression. Methods of image partitioning can be classified as image —independent tiling and image-dependent segmentation.

Image-independent tiling is any partitioning scheme that does not take the structure of the image into account, but instead uses simple geometric shapes to tile the image. This method has the advantage of simplicity i.e. the shapes of the domain regions can be simple, and the data can be structured simply, so that the decompressor can easily decode the domain tiles. The disadvantage is that this method cannot take into account regional difficulties and the natural connection between areas, exactly the advantage of image-dependent segmentation. Image-dependent segmentation will be the way of the future [39] and is explained in Chapter 4. A wide variety of image-independent partitions are investigated. However the majority of coding systems are based on a square or rectangular partition.

The simplest possible range partition consists of the fixed square blocks depicted in Fig.3.1(a) [31]. The most prominent example of a fractal compression system based on this partition was developed by Monro et al. [40], [41], 42]. This type of block partition is successful in transform coding individual image blocks, since an adaptive quantization mechanism is able to compensate for the varying 'activity' levels of different blocks,

allocating few bits to blocks with little details and many to 'active' blocks. Fractal coding based on the usual block transforms, in contrast, is not capable of such adaptation representing a significant disadvantage of this type of block partition for fractal coding.



a)Fixed block b)Quad tree c)Horizontal-Vertical

Fig. 3.1 Rectangular range partition schemes

In quadtree partition Fig.3.1 (b), a square in the image is broken up into four equal-sized sub-squares when it is not covered well enough by some domain. This process repeats recursively starting from the whole image and continuing until the squares are small enough to be covered within some specified RMS tolerance. Small squares can be covered better than large ones because contiguous pixels in an image tend to be highly correlated [43].

Quadtree partitions are described in considerable detail by Fisher [43, Ch.3] and Lu and Yew [44]. Jacquin used a variant of the quadtree partition in which the block splitting was restricted to two levels [45] [33]. If an error threshold is exceeded, the larger block prior to splitting into four subblocks will not be discarded automatically. Instead it is retained if additional transforms up to two subblocks were sufficient to reduce the error below the threshold.

Reusens [46] implemented a quadtree scheme in which the range blocks were overlapped in order to reduce blocking artifacts. A significant reduction in blocking artifacts was observed, but without corresponding improvement in SNR.

A weakness of quad-tree based partitioning is that it makes no attempt to select the domain pool in a content-dependent way. The collection must be chosen to be very large so that a good fit to the given range can be found. A way to remedy this, while increasing the flexibility of the range partition, is to use an HV-partition. In HV-partition (Fig.3.1(c)), a rectangular image is recursively partitioned either horizontally or vertically to form two new rectangles. The partitioning repeats recursively until a covering tolerance is satisfied, as in the quadtree scheme [43]. This partitioning scheme improves the compression ratio, while preserving a good image reconstruction. Rectangles permit a good cover of the horizontal and vertical edges in the image, but in case of other oriented shapes, we require many rectangles.

In another approach using HV-partitioning, the domains are selected from the partition itself, as opposed to a scan of the image [47]. In this method, when a rectangle contains either horizontal or vertical edges, the split should occur along such an edge and when a rectangle contains other edges, the split should occur in such a way that the edge runs 'diagonally' through one of the generated rectangles, with the other rectangle having no edge (See Fig. 3.2).

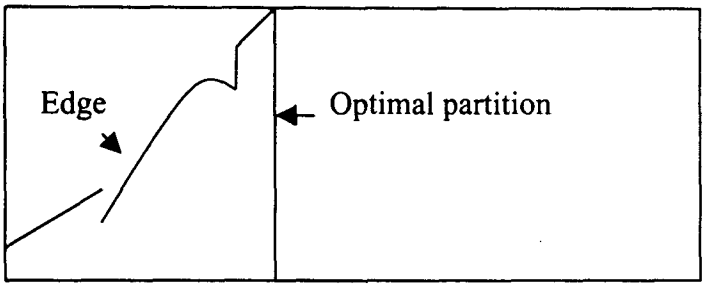


Fig. 3.2 A rectangle is split vertically so that an edge runs diagonally through one of the sub-rectangles.

This partition will tend to be self-similar at different scale lengths and facilitates efficient matching of domains and ranges. Also, the number of domains is very small compared with a domain pool generated by scanning the image, so encoding time is low [43].

In the triangular partitioning scheme, a rectangular image is divided diagonally into two triangles. Each of these is recursively subdivided into four triangles by segmenting the triangle along lines that join three partitioning points along the three sides of the triangle. This scheme has several potential advantages over the H-V partitioning scheme. It is flexible, so that triangles in the scheme can be chosen to share self-similar properties. The artifacts arising from the covering do not run horizontally and vertically, and this is less distracting. Also, the triangles can have any orientation; so we break away from the rigid 90 degree rotations of the quad tree and HV partitioning schemes [43], [48].

In Delaunay triangular partition, partition is computed on a set of points placed on the image support in a gray level-dependent way with the help of a split and merge approach. The main advantage of such an algorithm is to reduce the block effects in the

reconstructed image, at least on the diagonal image edges. Delaunay triangulation which allows control of the compression ratio as well as the quality of the attractor image [49].

Evolutionary computing is proposed by D.Saupe to find good partitioning [50]. Here the image to be encoded is subdivided into atomic blocks of same size. In each cycle of evolution an offspring generation of new configuration is produced as follows: in the parent population, maintain a list of all neighboring range pairs. From this list one range pair is chosen at random. These two ranges are united yielding a new partition with two old ranges removed and with their union as a new image. In order to obtain a matching domain block for the new enlarged range it need not search through the full respective domain pool, but, rather consider only those domains inherited from the parent ranges. Each evolution has one less range, thus bit rate decreases and evolution is halted when a given tolerance threshold for error is achieved. Compared to quadtree method, the evolutionary algorithm give better performances at high compression ratios. It gives larger gain up to several dB of PSNR.

In adaptive partitioning method, ranges are unions of edge-connected small square image blocks, called atomic blocks [51]. Then, neighboring ranges are iteratively merged to yield sequence of partitionings with a decreasing number of ranges. During this process, not only partitioning is maintained, also an associated fractal code is maintained. After each merger, a new fractal code is derived by a local modification of the previous one. This modification avoids an exhaustive search for an optimal domain for the newly merged range. This algorithm is the best in terms of image fidelity, but cuts down the encoding time to a fraction of the running time of fractal coders that achieve comparable image quality.

The coding scheme using range blocks of variable shape is also used to reduce the bit rate keeping quality of image [52]. In this scheme, the range blocks are determined by splitting and merging method. Also, there are two kinds of range blocks(shade blocks and non-shade blocks) according to the variance of brightness level inside the blocks. In encoding range blocks, a shade block is represented by the average of brightness level, and a non-shade block is represented by the parameters of contractive transforms of the corresponding domain block. The shape of the range block is encoded using chain coding. This method is better than the one using range blocks of fixed shape. Further it reduces the bit-rate keeping quality.

Comparison :

In designing an adaptive partition there is always a trade-off between the lower distortion expected by adapting the partition to the image content, and the additional bits required to specify the partition details.

Reusens [53] compared rate distortion results obtained for the polygonal partition with those for the HV and quadtree partitions, concluding that the simplest partition (quadtree) provided the best results. This conclusion is not in agreement with the comparison [43,Ch-6] of the quadtree and HV-partitions by Fisher, in which HV-partition was found to be superior. An irregular partition [34], [52] was found to offer superior performance to a fixed square block partition, but was not compared with other partitions. Saupe and Ruhl found that a similar irregular partition outperformed a quadtree partition [50].

A significant disadvantage of the non-rectangular partition is the additional computation involved in the block transformations. The performance of the non-rectangular partition based coders does not appear to justify the additional complexity.

3.5.2 Search for domain blocks :

The time consuming part of the encoding step is the search for an appropriate domain for each range. The image transformation has the form

$$\tau = \sum_{0 \le i < N} \tau_i$$

with τ =TioSi (3.7)

Given a range block R_i (of size B x B), the construction of a transformation τ_i which maps on to this cell is broken into two distinct steps corresponding to the transformations S_i and T_i, respectively. The first part is the construction of the spatial contraction S_i. This amounts to selecting an image domain block of size D x D, which will be contracted to a block S_i of size B x B. The specification of the domain block D_i (location and size) is equivalent to the description of the spatial contraction S_i (translation vector and contraction factor). The second part of the construction of τ_i consist of selecting the proper approach to processing of the contracted domain block S_i, i.e. of finding the block transformation T_i which minimizes the distortion between $T_i o S_i$ ($\mu \rceil D_i$) and $T_i o S_i$ ($\mu \rceil D_i$) and $\mu \rceil R_i$, where μ is the given digital image.

Thus, a pool of domain blocks D is made up of all image blocks which can be extracted from the original image, and which are larger than the range block R_i. The

encoding of the range block $\mu_i R_i$ consists in finding a 'best pair' $(D_i, T_i) \varepsilon DxT$, which is such that

$$d(\mu_i R_i, S_i \mu_i D_i) \text{ is minimum}$$

Pools of Domain Blocks:

The maximal domain pool corresponding to a range block of size B x B will be all image blocks of size D x D (D>B) located anywhere in the image . It is typically very large, but it can be trimmed and organized in order to make the search for an optimal domain block tractable. An initial domain pool D can be obtained by sliding a window of size D x D (D = 2B is used) across the original image. The window is first located with its bottom left corner at (0,0). It then moves from one position to the next by steps of either δ_h pixels horizontally to the right or δ_v pixels vertically upwards, in such a way that it remains entirely inside the image support, at all times. The steps can typically be chosen equal to B or B/2 [54].

In a fixed square block or quadtree partitions, domain blocks may be placed at intervals as small as one pixel, but this results in large domain pool and takes more time for searching. The larger domain increments are usually selected, typically equal to domain block width [43, Ch.3] or half the domain block width [55] . Improved convergence is obtained with either of these increments [56].

It was observed that, the range block is spatially close to the matching domain block and hence the domain pool for each range block may be restricted to a region about the range block [57].

Search Techniques :

The design of efficient domain search techniques to reduce long encoding time has been one of the most active areas of research in fractal image compression, resulting in a number of solutions. Most of the proposed techniques attempt to accelerate the searching and are based on some kind of feature vector assigned to ranges and domains. The features can be discrete (leading to classification and clustering) or continuous (yielding functional or nearest neighbor methods). When applied, these methods provide greater speed which is traded in for some loss in image fidelity and compression ratio.

The domain pool consists of all square subimages of a particular size and only a fraction of this large pool is actually used in the fractal code. This subset be can characterized in two related ways: (1) It contains domains with relatively large intensity variation, (2) The collection of used domains are localized in those image regions with a high degree of structure. A priori discarding of those domains with least variance can accelerate the encoding process and these are unlikely to be chosen for the fractal code.

In this way, the most useless domains from the pool are removed achieving lean and more productive domain pool. This happens at the expense of compression ratio, which is reduced. The localization of the domains can be exploited for an improved encoding in effect raising the compression ratio back up without any penalty [58].

The domain-range comparison step of encoding is computationally very intensive. Classification scheme is used to minimize the number of domains compared with a range. Before encoding, all the domains in the domain library are classified and this avoids reclassification of domains. During encoding, a potential range is classified, and only domains with the same (or near) classification are compared with the range. This

significantly reduces the number of domain-range comparisons. Many classification schemes are possible. Some of them are discussed below.

Jacquin used a scheme that classified a sub-image into flat, edge, and texture regions [33]. Fisher used a method of classification into 72 classes based on relative brightness or averages of the quadrants of each block [43, Ch.3].

Reduction in computational complexity can be obtained by using a geometric construction, allowing clustering, as well as providing upper and lower bounds for the best match between domain and range blocks. This allows blocks to be excluded from the computationally costly matching process [59].

By arranging the domain blocks in a tree structure and utilizing this tree structure procedure to direct the search, coding process in fractal image compression is speeded up without noticeable loss of image quality [60].

A classification scheme based on a notion of distance between blocks and cluster centres designed with Kohonen's self-organizing maps, reduces the encoding time [61].

It is common practice in fractal image compression to include all eight isometric versions of a codebook block in the codebook. It is reasoned that such enlarged domain pools yield better rate-distortion curves. It has been shown that, the plain codebook without using isometries provides same or better results [62].

To speed up the encoding process, the pyramidal search algorithm is used. The pyramidal search is first carried out on an initial coarse level of the pyramid. This search increases the encoding speed significantly, because not only the number of the domain blocks to be searched is reduced, but also the data within each domain block are only a

fraction of those in the finest level. Then, only a few numbers of the fractal codes from the promising domain blocks in the coarse level are refined through the pyramid to the finest level with little effort. The main advantage of this algorithm is the greatly reduced computational complexity [63].

By using enveloped domain search in fractal image compression, encoding time is reduced and compression ratio gets increased. The enveloped domain search algorithm uses correlation and only those domain blocks which envelop the given range blocks are chosen for the search. The reduction in encoding time is significant [64].

Encoding time in fractal image compression can be reduced by using multi-dimensional nearest neighbor search [74]. The traditional search takes linear time whereas the nearest neighbor search can be organized to require only logarithmic time. The speed up can be adjusted so that it comes with only minor degradation in the image quality and compression ratio and with improvements in both fidelity and compression.

3.6 Image Reconstruction From A Fractal Code :

A major disadvantage of fractal image compression is the long encoding time. However, the decoding is both simple and fast. Thus it makes a good candidate for storage and retrieval applications where the encoding is performed once using special hardware, while the decoding is to be repeated several times in software, by the user. Hence, it is worth asking if one can further simplify the basic decoding algorithm for these environments. The decoding algorithm, in the most general case, is similar to the one proposed by Jacquin [33] and is based on the Contractive Mapping Theorem [29], [43]. The reconstruction of the fixed point x_T by the decoder proceeds by iteratively applying T to any arbitrary initial image $x^{(o)}$. The majority of existing fractal coding schemes restrict

T to be an affine transform $T_x = Ax + b$, where A is a linear transform and b is an offset vector. Banach's fixed point theorem [29] ensures the convergence of the sequence of iterates $\{T^{(k)}(x^{(o)})\}_k$ to the fixed point $x_T = (I-A)^{-1}b$, where I is the identity matrix of order N. Here a vector $T^{(k)}(x^{(o)})$ which is denoted by $x^{(k)}$ is defined by the recurrence relation.

$$x^{(k+1)} = T(x^{(k)} = A\, x^{(k)}+b \tag{3.8}$$

The fixed point x_T is also known as the attractor of T. The encoding consists of finding a contractive affine mapping T such that the attractor x_T and original image are as close as possible.

In conventional fractal image compression, the decoding amounts to iterating an affine map on an arbitrary initial image until convergence occurs. It has been shown that, the decoding in fractal image compression can be improved by combining a pixel update scheme with an ordering technique based on the frequency of the ranges (or the pixels) in the fractal code. Besides the computational gains inherent in the non-iterative decoding of zero frequency ranges (or pixels), this method allows better approximations at each iteration step and a faster convergence [65].

R. Hamzaoui has proposed a new decoding technique based on an iterative method with pixel updating. Let $x^{(k)}$ be the decoded image at iteration step k. In conventional decoding, it is necessary to keep all the components of the image $x^{(k)}$ until we achieve the computation of $x^{(k+1)}$. In the method proposed by Hamzaoui, latest estimates of the components $x_u^{(k+1)}$ are used, as in the Gauss-Seidal method. Thus simultaneous storage of $x^{(k)}$ and $x^{(k+1)}$ is not needed and thus only half the storage requirement of the conventional decoding method is needed. Convergence of this method was proved in the most general case of variable size and shape segmentations where searching, unconstrained domains and

pixel shuffling were allowed. This decoding scheme converges faster than the conventional one [66], [67].

The review of literature indicates that several investigations have been made to achieve better results in image compression. Also, substantial research has been done in image compression by the method of fractal and wavelet coding. The objective of the proposed research work is to develop new methods for compressing digital images. While doing so, it is intended to achieve better compression ratio and image quality with reduced encoding time. The subsequent sections presents the work done so for and proposed methods.

3.7 Exhaustive Search Method:

The first practical and automated implementation of fractal image compression was done by Jacquin[33], who used block based partitioned iterated function systems(PIFS) to compress images. The simplest implementation of fractal image compression is the Exhaustive search method.

Consider an n x n digital gray level image μ. The image μ is partitioned into non-overlapping square blocks of size r × r called range blocks. The first block is aligned with the bottom left corner of the image. If the blocks do not reach up to the right or the top edge, then the margin can be covered by rectangular blocks. Next, the domain blocks of size 2r × 2r in the same image μ is found. These domain blocks overlap such that a new domain block starts at each pixel. However, some authors suggest that the blocks be aligned on a grid size of r or r/z . The more domain blocks, the better the decoded image quality, but the encoding time is more.

The compression process is as follows: For every range block Ri, search the domain pool to find a block D and a transformation T such that T(D) is the best match for Ri. The closeness is measured by a distance metric. The distance of blocks $A=(a_i)$ to $B=(b_i)$, $0 \le i \le m$ is the root mean squared (RMS) difference.

$$d_{rms}(A,B) = \sqrt{\frac{1}{m}\sum (a_i - b_i)^2} \tag{3.9}$$

The transformation T is composed from a contracting map followed by a geometric map followed by a massic map. The contracting map shrinks the domain block to half the size, replacing 2×2 pixel areas by their average. The geometric map is one of the eight flips(or symmetries) of the square. The massic map changes the contrast and brightness by a scale factor s and an offset o. The domain block d_{ij} is mapped to $(sd_{ij}+o)$. The map T is contracting if $s<1$. The image is thus encoded as a set of transformations. The mappings are found for each range block and their union defines a mapping on the image as a whole. The decoding process consists in applying transformations to an arbitrary initial image and repeating the process. If the transformations are contractive, the original image is obtained after some iterations.

The number of tiled range blocks is $\frac{n \times n}{r \times r} = \frac{n^2}{r^2}$, while the number of domain blocks is $(n-2r+1)^2$. The complexity of computation of best match between a range block and a domain block is $O(r^2)$. Considering r to be constant, the computational complexity of an exhaustive search is $O(n^4)$.

The computational requirements for an exhaustive search is prohibitive (in the range of 8 hours on a Pentium PC, 166 MHz. for a 256 x 256 image) .

This algorithm is implemented and tested on a Pentium PC, 166 MHz for various images. Results for lenna image of size 256x256 is shown in Table 3.1. This study indicates the need for better method of encoding in fractal image compression.

Compression Ratio	Exhaustive search		
	Enc.Time (hrs:min)	RMS Error	PSNR (dB)
4.5	3:57	7.212	30.97
8.6	2:26	7.999	30.07
10.4	2:13	8.692	29.438
21.1	1:24	12.609	26.117
42.7	1:00	16.941	23.552

Table 3.1 : Encoding Time and Quality of decoded lenna image of size 256x 256 for Exhaustive Search Method

3.8 Fractal Image Compression Using Quadtrees :

In this section, a quadtree-based fractal encoding scheme is given [43].

The encoding step follows the pseudo-code in Table 3.1. In this case, the collection of ranges comes from a quadtree partition of the image, and the collection of domains consists of subsquares of the image that are twice the range size

The Ranges:

A quadtree partition is a representation of an image as a tree in which each node, corresponding to a square portion of the image, contains four subnodes, corresponding to the four quadrants of the square. The root of the tree is the initial image. See Fig. (3.3).

A. Encoding:

- Choose a tolerance level Ec.
- Set $R_1 = I^2$ and mark it uncovered.
- While there are uncovered ranges R_i do {
  - Out of the possible domains **D**, find the domain D^i and the corresponding transformation w_i that best cover R_i (i.e., that minimises the distance in some metric)
- If $d_{rms} < e_c$ or size $(R_i) \leq r_{min}$ then
  - Mark R_i as covered, and write out the transformation w_i;
- else
  - Partition R_i into smaller ranges that are marked as uncovered, and remove R_i from the list of uncovered ranges

Table 3.2 Pseudo-code for fractal encoding

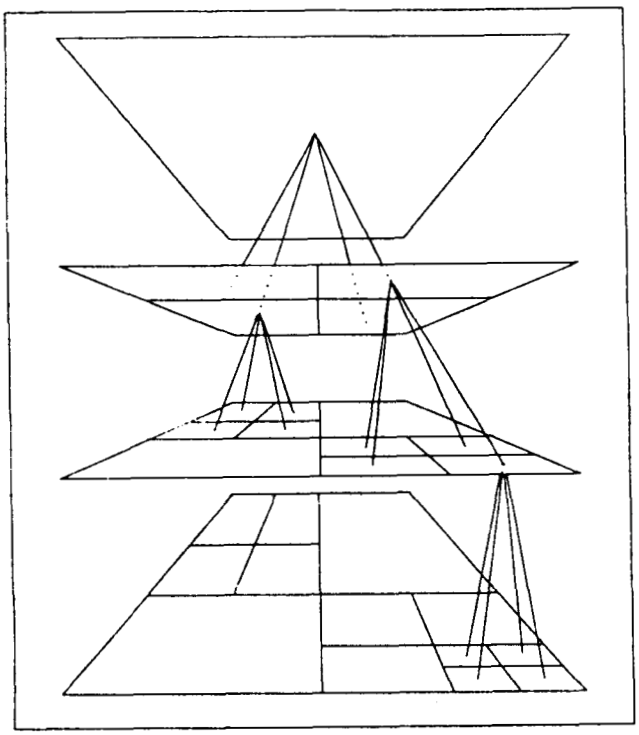


Fig. 3.3 Quadtree structure

The ranges are selected as follows: after some initial number of quadtree partitions are made (corresponding to a minimum tree depth), the square at the nodes are compared with domains from the domain library (or domain pool) **D** which are twice the range size. The pixels in the domain are averaged in groups of four so that the domain is reduced to the size of the range, and the affine transformations of the pixel values (that is scaling and offset) is found that minimises the RMS difference (See Appendix C) between the transformed domain pixel values and the range pixel values. All the potential domains are compared with a range. If the resulting optimal rms value is above a pre-selected threshold and if the depth of the quadtree is less than a pre-selected maximum depth, then the range square is subdivided into four quadrants (which means adding four subnodes to the corresponding range), and the process is repeated. If the rms value is below the threshold, the optimal domain and the affine transformation on the pixel values are stored. This constitutes one map w_i and collection of all such maps $W = Uw_i$ constitutes the encoding.

The above method must be modified in order to fit the actual algorithm. First, the affine transformation of the pixel values consists of a multiplication by a scaling factor s_i and an addition of an offset value o_i (these can be thought of as a contrast and brightness adjustment). These values must be quantized for efficient storage, and these quantized values are also used for computing the RMS difference. Second, if the magnitude of the optimal scaling factor $|\ S_i\ |$ is greater than 1 for any of the w_i, there is a chance that the resulting map W will not be eventually contractive.. Thus, $|s_i|$ larger than some maximal value s_{max} are truncated.

The domains

There are three major types of domain libraries $\mathbf{D_1}$, $\mathbf{D_2}$, $\mathbf{D_3}$. The domains are selected as subsquares of the image whose upper-left corners are positioned on a lattice determined by a parameter *l*. The lattice spacing is selected as follows:

$\mathbf{D_1}$ has a lattice with a fixed spacing equal to *l* and there will be equal number of domains of each size.

$\mathbf{D_2}$ has a lattice whose spacing is the domain size divided by *l*. Thus, there will be more small domains than large.

$\mathbf{D_3}$ has a lattice as above, but with the opposite spacing-size relationship. The largest domains have a lattice corresponding to the smallest domain size divided by *l*, and vice- versa.

Each domain can be mapped onto a range in eight different ways - four rotations and a flip with four more rotations. Thus, the domain pool can be thought of as containing the domains in eight possible orientations.

The Classification:

The domain-range comparison step of the encoding is computationally, expensive. A classification scheme is used in order to minimise the number of domains compared with a range. Before the encoding, all the domains in the domain library are classified and this avoids reclassification of domains. During the encoding, a potential range is classified, and only domains with the same (or near) classification are compared with the range. This significantly reduces comparisons.

There are many classification schemes available. Jacquin[33] used a scheme that classified a sub-image into flat, edge, and texture regions.

In the present algorithm, the following scheme is used.

A square sub-image is divided into upper left, upper right, lower left, and lower right quadrants, numbered sequentially. On each quadrant, compute values proportional to the average and the variance. If the pixel valued in quadrant I are r_1^i --------- r_n^i

for $i = 1,2,3,4$, compute $A_i = \sum_{j=1} r_j^i$ and $V_i = \sum_{j=1}^{n} (r_j^i)^2 - A_i^2$ (3.10)

It is always possible to orient the sub-image so that the A_i are ordered in one of the following three ways:

Major class 1 : $A_1 \geq A_2 \geq A_3 \geq A_4$

Major class 2 : $A_1 \geq A_2 \geq A_4 \geq A_3$

Major class 3 : $A_1 \geq A_4 \geq A_2 \geq A_3$

These correspond to the brightness levels shown in Fig 3.4

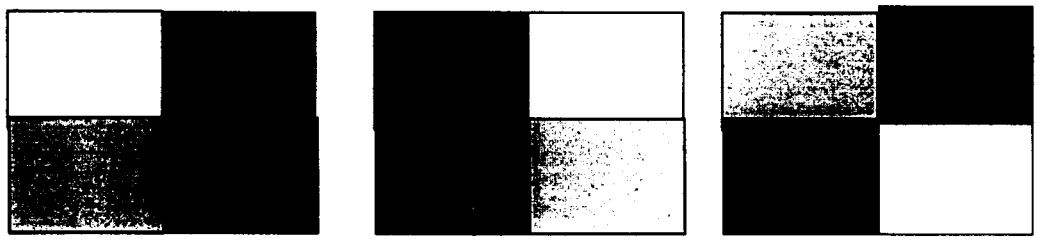


Fig 3.4 Classification of Image based on brightness

Once the rotation of the square has been fixed, each of the three major classes has 24 subclasses consisting of 24 orderings of the V_i . Thus, there are 72 classes in all if the scaling value s_i is negative, the orderings in the classes above are rearranged. Therefore, each domain is classified in two orientations, one orientation for positive s_i and the other for negative s_i .

The Encoding Parameters:

The parameters for this algorithm are:

i) The RMS. tolerance threshold, e_c

ii) The maximum depth of the quadtree partition

iii) The minimum depth of the quadtree partition.

iv) The type of domain pool and the lattice spacing *l*.

v) The maximum allowable scaling factor s_{max}

vi) The number of classes compared with a range.

An encoding of an image consists of the following data:

i) The final quadtree partition of the image

ii) The scaling and offset values s_i and o_i for each range.

iii) For each range, a domain that is mapped to it.

iv) The symmetry operation (orientation) used to map the domain pixels onto the range pixels.

B. Decoding:

Decoding an image consists of iterating ω from any initial image. The quadtree partition is used to determine all the ranges in the image. For each range R_i , the domain D_i that maps to it is shrunk by two in each dimension by averaging non-overlapping groups of 2×2 pixels. The shrunken domain pixel values are then multiplied by s_i , added to o_i, and placed in the location in the range determined by the orientation information. This constitutes one decoding iteration. The decoding step is iterated until the fixed point is approximated, i,e. until further iteration doesn't change the image or until the change is below some small threshold value.

Decoding at a larger size:

The transformations that encode an image are resolution independent. The underlying model uses, functions of infinite resolution, and the fixed point reached here is such a function. Therefore, during decoding we can select any resolution.

Post processing:

Since the ranges are encoded independently, there is no guarantee that the pixel values will be smooth at the block boundaries. The eye is sensitive to such discontinuities, even when they are small. It is possible to minimise these artifices by postprocessing the image. In this implementation, only pixel values at the boundary of the range blocks is modified using a weighted average of their values.

Efficient storage:

To increase the compression ratio, the following bit allocation schemes are used.

Quad tree: One bit is used at each quadtree level to denote a further recursion or ensuing transformation information. At the maximum depth, however, no such bit is used, since the decoder knows that no further division is possible.

Scaling and offset: Five bits are used to store the scaling and seven for the offset.

Domains: The domains are indexed and referenced by this index. However, when the scaling value is zero, the domain is irrelevant, and so no domain or orientation information is stored in this case.

Orientation: Three bits are used to store the orientation of the domain-range mapping.

The above algorithm is implemented and tested for various gray level images. The results obtained are better compared with those for exhaustive search method. The results are shown in Table (3.1) for exhaustive search method. The confidence gained by the above completed works has motivated the researcher to suggest new methods. It is proposed to improve the above methods by incorporating new classification scheme and searching techniques.

3.9 Fast Fractal Image Compression

The encoding step in Fractal Image Compression is slow because most of the time is spent in searching and discarding of unrelated domain blocks. In the proposed work, it is intended to find means to speed up the encoding step in fractal image compression, while maintaining and / or increasing the image quality and compression ratio.

3.10 Quadtree based fractal image compression using partial distance search algorithm:

A significant fraction of the computational cost of the domain search lies in the actual calculation of distances between domain and range blocks. The time required for the search may be reduced by improving the efficiency of these calculations. One of the method for decreasing search time is partial distance method [82].

In fractal image compression, while finding the best domain block for each range block, instead of calculating complete distance, only partial distance is found. If this partial distance is more than the previous value of distance that domain is rejected before finding the complete distance. This reduces lot of computations and hence searching time can be reduced.

3.10.1 Algorithm :

Let us assume that a sampled image is partitioned into non-overlapping square blocks of size BxB called range blocks. The domain pool is a collection of square blocks, which are typically larger than the ranges (normally twice the range size) and also taken from the image, called domain blocks. The domain pool may be enlarged by including blocks obtained after applying the eight isometries of the square to the domain blocks. By pixel averaging, the size of these blocks is reduced to size of a range block. The resulting blocks are called code book blocks.

In the encoding process for a range block R a search through the code book blocks $D_1 \ldots . D_{ND}$ is required. For range R and code book block D, we let

$(s,o) = \arg\min_{s,o \in R} \| R - (sD+o\mathbf{1}) \|^2$ where **1** is the constant block with unit intensity at every pixel. The scaling co-efficient s is clamped to [-1,1] to ensure convergence in the decoding and then both s and o are uniformly quantized. The error for range R is E(D,R) = $\| R-(sD+o\mathbf{1}) \|^2$. The code book block $D_{i,j}$ with minimal rms error $E(D_{I,j},R)$ yields the fractal code for each range R consisting of the index I,j and offset parameters s and o. For a range block R and code book block D the optimal co-efficients are

$$s = \frac{n\langle D,R\rangle - \langle D,1\rangle \langle R,1\rangle}{n\langle D,D\rangle - \langle D,1\rangle^2}, \quad o = 1/n\,(\langle R,1\rangle - s\langle D,1\rangle) \tag{3.11}$$

For any (s,o) the error E(D,R) can be regarded as a function of $\langle D,R\rangle$, $\langle D,1\rangle$, $\langle R,R\rangle$ and $\langle R,1\rangle$,

$$E(D,R) = s(s\langle D,D\rangle + 2(o\langle D,1\rangle - \langle D,R\rangle)) + o(on - 2\langle R,1\rangle) + \langle R,R\rangle \tag{3.12}$$

The calculation of the inner products $\langle D,R\rangle$ dominates the computational cost in the encoding, e.g. direct computation of $\langle D,R\rangle$ with R containing 16.16 values requires 512 floating point operations. Using the partial distance method the rms error and inner product is calculated as follows:

For a given range block R, assume rmsold as the actual minimum distance (rmse) to the nearest domain block. In testing a new code book domain block D_I, if the partial distance (rmse) exceeds the previous minimum distance, rmsold, the codebook block D is eliminated. Further computation of distance is not needed because E(D,R) is larger than the smallest distortion yet found in the search. This technique reduces the computational complexity as compared to full search method.

Step 1 : rmsold = ∞

Step 2 : for i=1.... ND

Step 3 : rms=0

Step 4 : for j=1,..... 4

Step 5 : calculate rms for block Bx B/4(j)

Step 6 : if rms > rmsold go to step 10

Step 7 : next j

Step 8 : rmsold = rms

Step 9 : best_domain = i

Step 10 : next I

3.10.2 Implementation and Results :

The quad tree based fractal image compression [43, Ch.3] as given in Section 3.8 and partial distance method based fractal image compression algorithms are coded in C and run on Pentium PC, 166 MHz under linux O.S. for various gray scale images.

The algorithm for fractal image compression with partial distance method is summarised in the following :

A : Input : Image of size NxN, range size BxB

B : Global pre-processing : Compute he down filtered image by pixel averaging.

Compute $\langle D, D \rangle$, $\langle D,1 \rangle$ for all code book domain blockls of size BxB/4, BxB/2, Bx3B/4 and BxB.

C : Searching : For each range R do steps C1 to C7.

1. Compute $\langle R, R \rangle$ and $\langle R, 1 \rangle$ of size BxB/4, BxB/2, Bx3B/4 and BxB

   rdsum_t =0.

2. For m = 1,.. 4

3. rdsum = rdsum_t + $\langle D, R \rangle$

4. rdsum_t = rdsum

5. For each code book block D compute the (quantized) co-efficients, s,o and the collage error E(D,R) using the results of step B, C1, C2

6. If E(D,R) is greater than $E(D,R)_{old}$ go to next domain block

7. Go to step 3

8. Extract the minimal collage error and output the fractal code for each range

To compare the partial distance method based fractal image compression with quad tree based fractal image compression, encoding time is used at different compression ratio and PSNR. We use 8-bit gray scale standard test images and non-standard images. The fractal encoding is run with and without partial distance method and resulting encoding time, PSNR at different compression ratio for lenna image of size 256x256, 512x512, collie image of size 256x256 and sanfranscisco image of size 256x256 pixels are given in Table 3.3 and 3.4, 3.5 and 3.6. The results for non-standard image administrative building of CREC is given in Appendix D. The results show speed up with little degradation in quality. Figure (3.5), (3.6), and (3.7) shows plot of encoding time and PSNR versus compression ratio for lenna image of size 256x256 and 512x512 and collie image of size

256x256. We have used 8 isometries with both negative and positive scaling as in [43, Ch.3]. Figures (3.7) and (3.8) shows original and decoded images after compression.

Compr. ratio	Fisher's Quad Tree Method			Partial Distance based FIC			Speed up Factor
	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	
4	680	7.187	31	500	7.613	30.5	1.36
8	455	8.064	30	300	8.542	29.5	1.52
12	375	9.474	28.6	255	9.92	28.2	1.47
20	250	12.78	26	190	13.078	25.8	1.32
30	210	14.674	24.8	150	15.365	24.4	1.4
40	180	16.559	23.75	120	17.24	23.4	1.5
50	175	17.54	23.25	105	18.473	22.8	1.66

Table (3.3): Encoding time and PSNR for various Compr.ratio for lena image, size 256x256 for Partial distance based Fractal Image Compression

Compr. ratio	Fisher's Quad Tree Method			Partial Distance based FIC			Speed up Factor
	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	
10	2200	4.041	36	1800	4.136	35.8	1.22
20	1400	6.047	32.5	1050	6.332	32.1	1.33
50	1200	10.152	28	800	9.92	28.2	1.5
80	1000	12.065	26.5	600	12.634	26.1	1.66
100	980	12.78	26	580	13.078	25.8	1.68
200	780	15.723	24.2	440	16.464	23.8	1.77
250	760	16.464	23.8	400	17.24	23.4	1.9

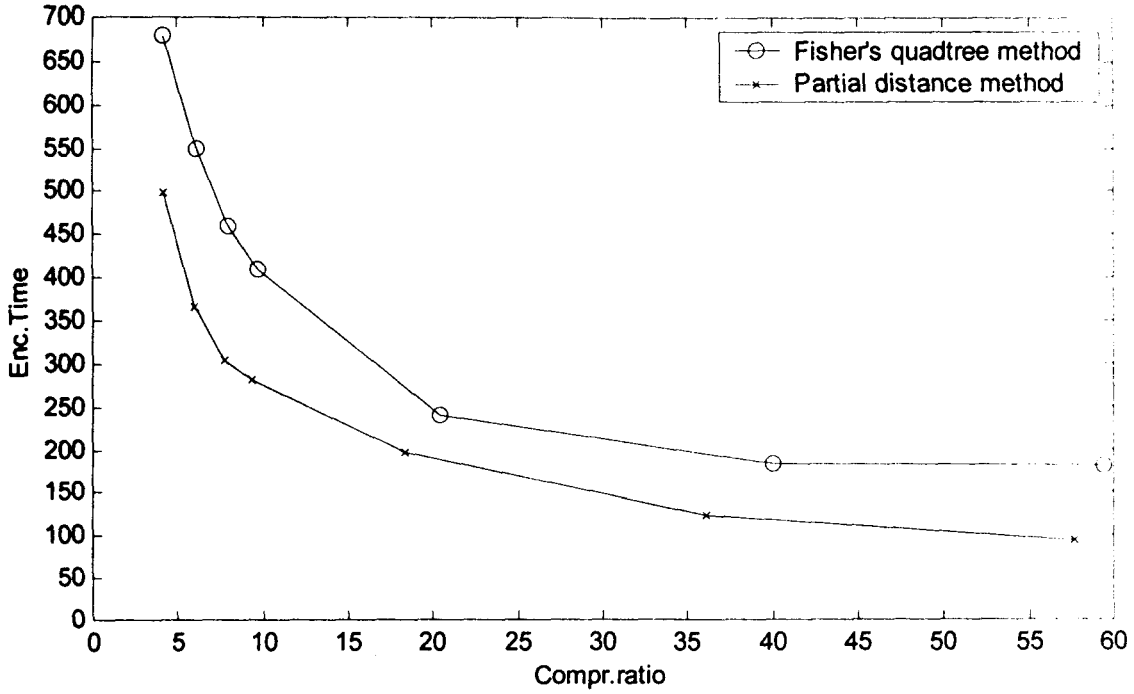
Table (3.4): Encoding time and PSNR for various Compr.ratio for lena image, size 512x512 for Partial distance based Fractal Image Compression

Compr. ratio	Fisher's Quad Tree Method			Partial Distance based FIC			Speed up Factor
	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	
4	800	3.323	37.7	500	3.602	37	1.6
8	550	4.535	35	320	4.64	34.8	1.72
12	400	5.709	33	300	5.909	32.7	1.33
20	330	6,863	31.4	248	7.187	31	1.33
30	290	8.064	30	200	8.347	29.7	1.45
40	250	9.258	28.8	160	9.695	28.4	1.56
50	215	10.63	27.6	140	10.878	27.4	1.54

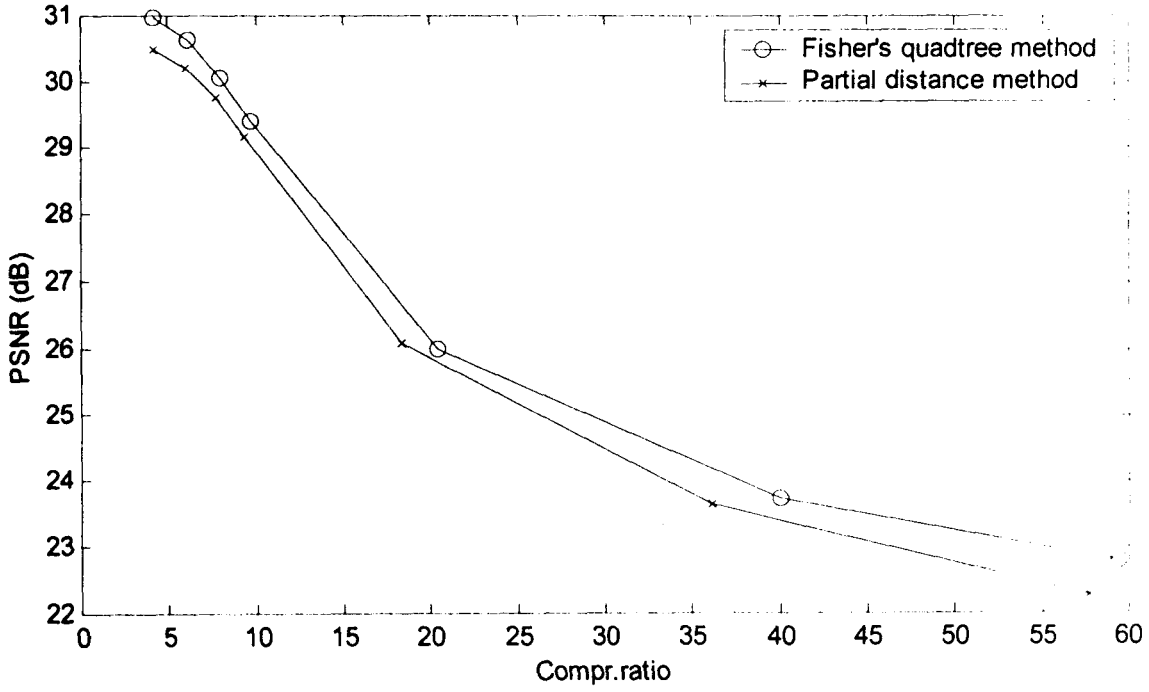
Table (3.5): Encoding time and PSNR for various Compr.ratio for collie image, size 256x256 for Partial distance based Fractal Image Compression

Compr. ratio	Fisher's Quad Tree Method			Partial Distance based FIC			Speed up Factor
	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	
6	650	7.354	30.8	470	7.70	30.4	1.38
8	525	8.064	30	400	8.542	29.5	1.31
12	425	10.152	28	300	11.131	27.2	1.42
20	340	14.103	25.2	225	14.844	24.7	1.51
30	270	17.043	23.5	160	17.846	23.1	1.69
40	235	19.344	22.4	140	19.794	22.2	1.68
50	210	20.498	21.9	120	21.209	21.6	1.75

Table (3.6): Encoding time and PSNR for various Compr.ratio for sanfransisco image, size 256x256 for Partial distance based Fractal Image Compression



(a) Plot of Compression Ratio Vs. Encoding Time



(b) Plot of Compression Ratio Vs. PSNR

Figure (3.5) Performance graph for lenna image of size 256x256

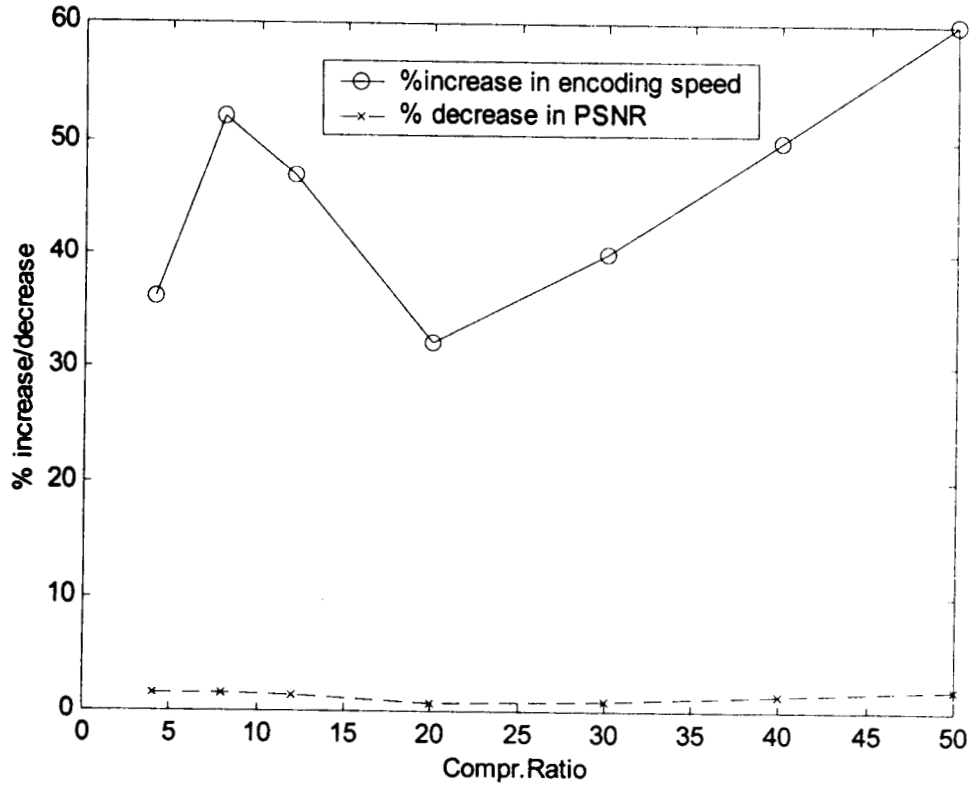
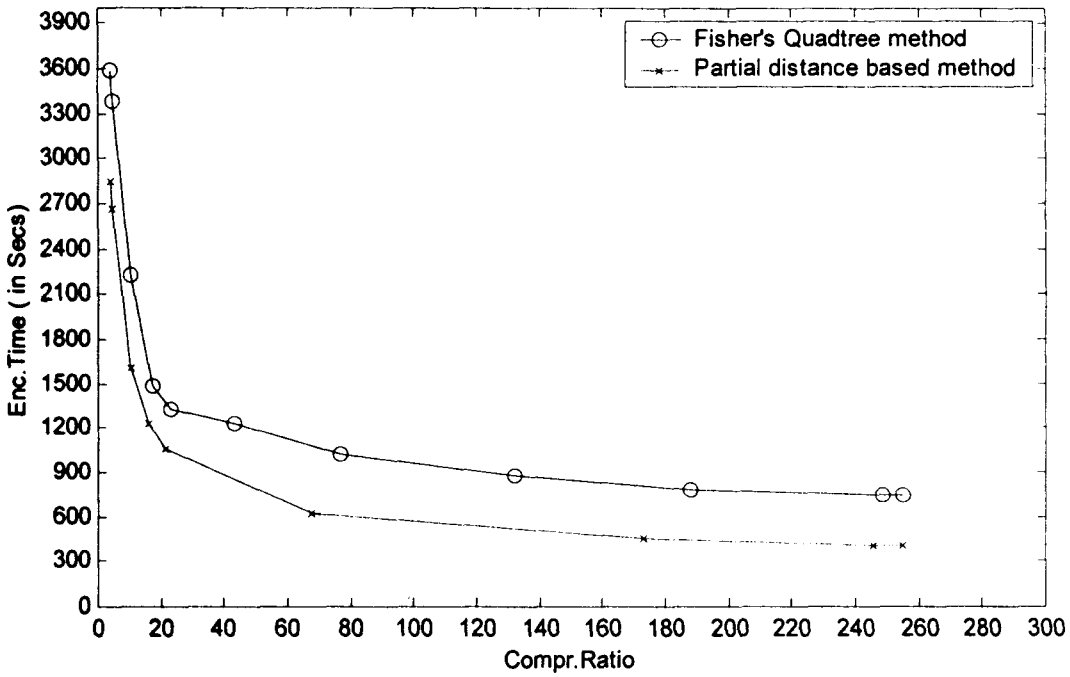
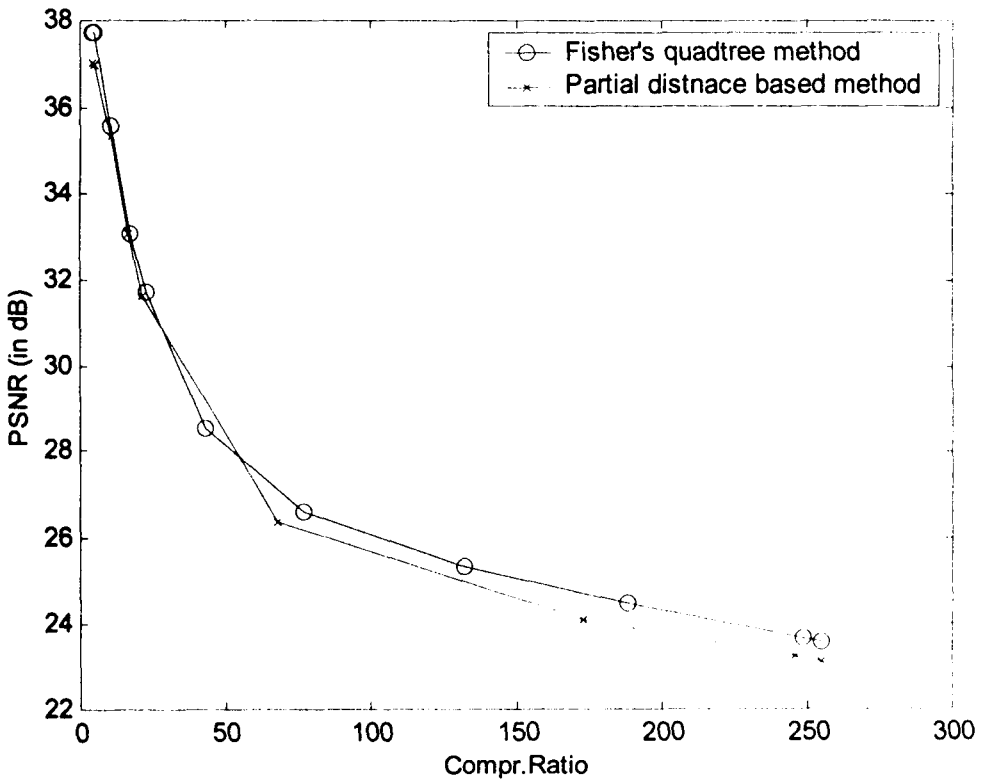


Figure (3.6) Plot of Compression Ratio Vs. % increase in encoding speed and % decrease in PSNR for lenna image of size 256x256 (Partial distance based FIC)



(a) Plot of Compression Ratio Vs. Encoding Time



(a) Plot of Compression ratio Vs. PSNR

Figure (3.7) Performance graph for lenna image of size 512x512

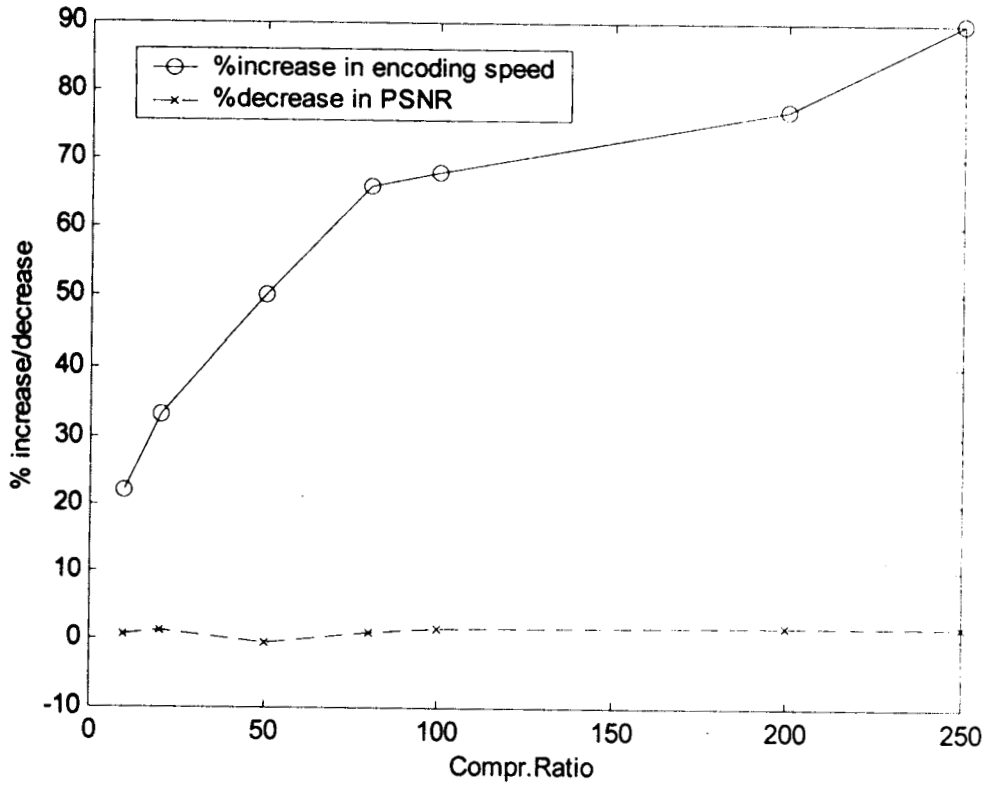
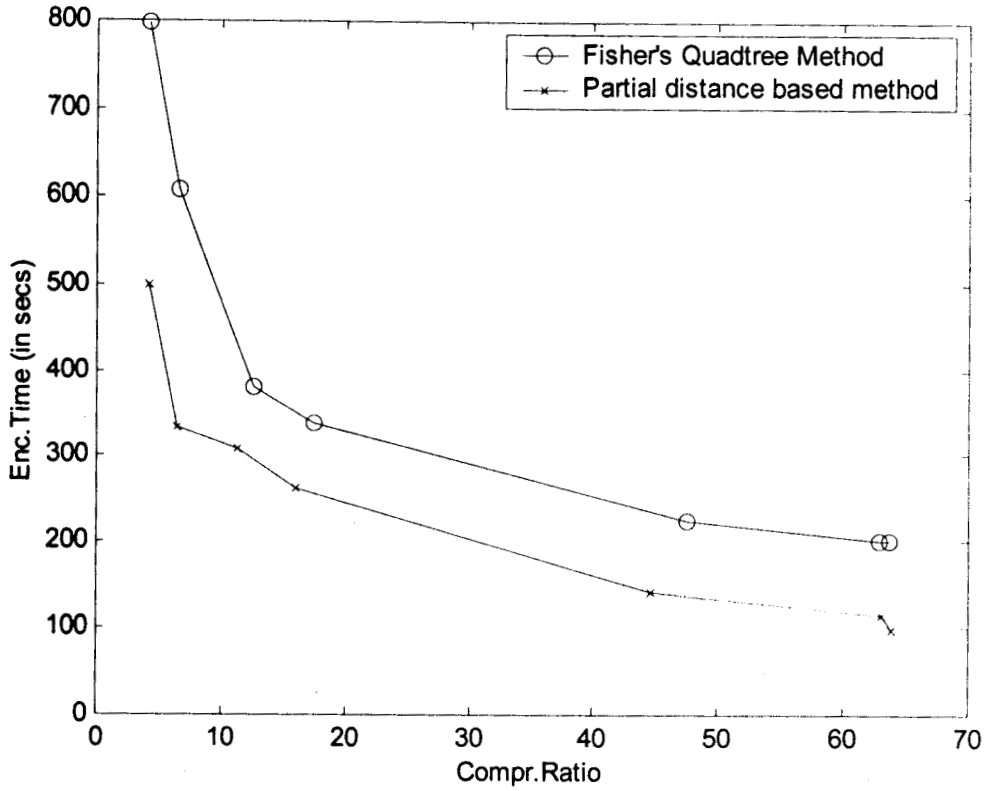
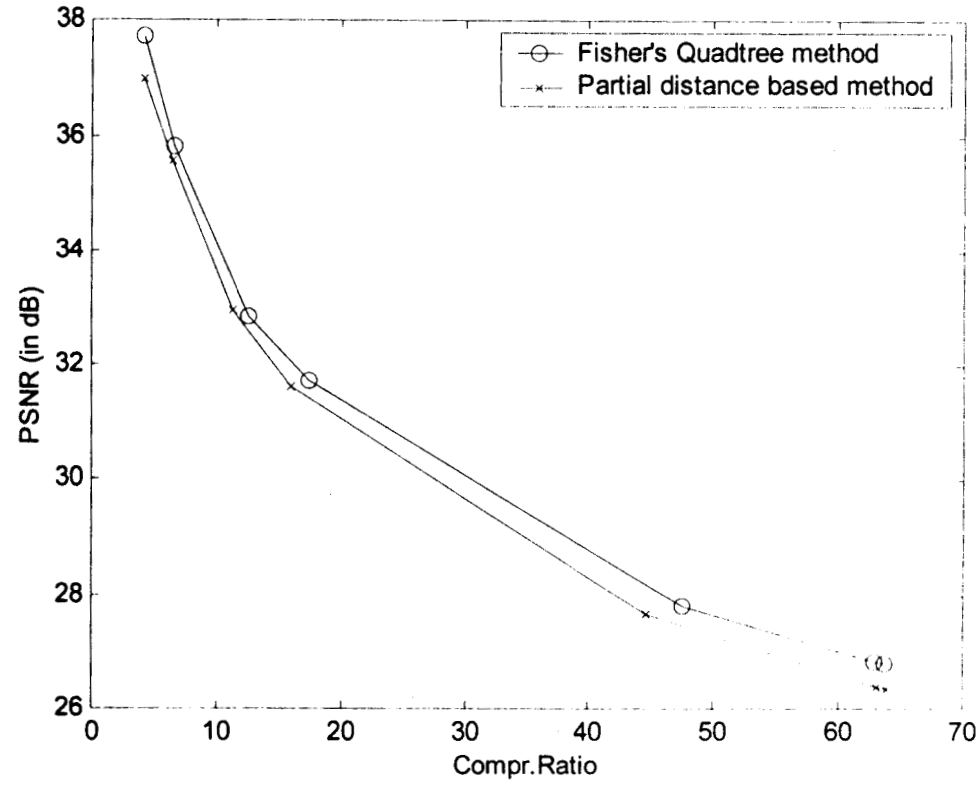


Figure (3. 8) Plot of Compression Ratio Vs. % increase in encoding speed and % decrease in PSNR for lenna image of size 512x512 (Partial distance based FIC)



(a) Plot of Compression ratio Vs. Encoding Time



(b) Plot of Compression Ratio Vs. PSNR

Figure (3.9) Performance graph for collie image of size 256x256

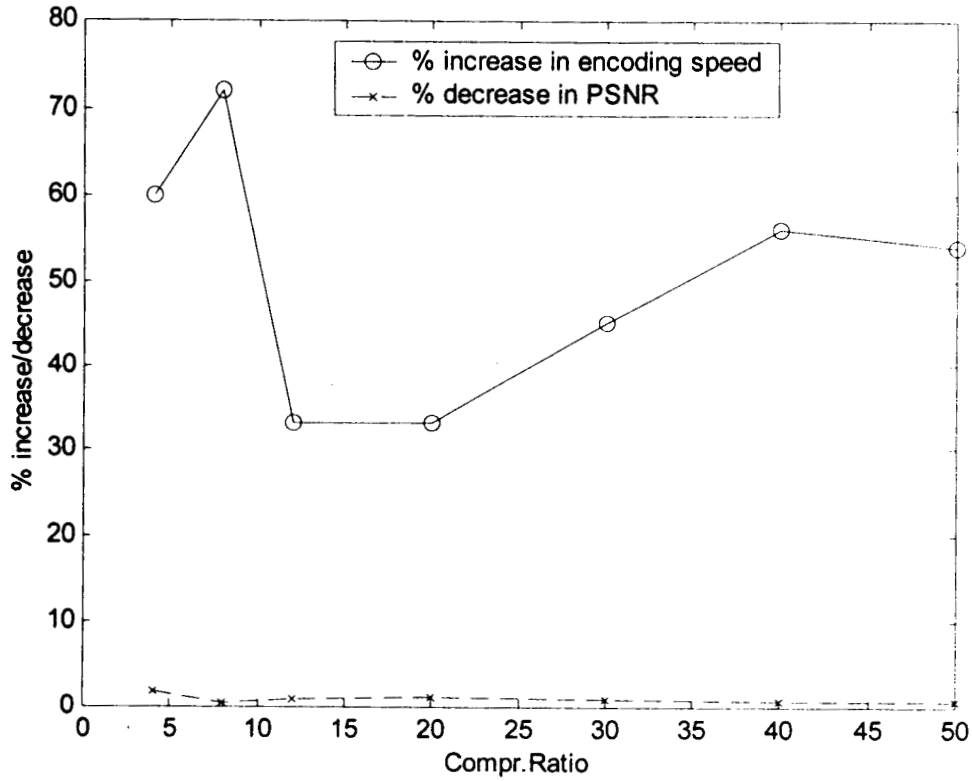


Figure (3.10) Plot of Compression Ratio Vs. % increase in encoding speed and % decrease in PSNR for collie image of size 256x256 (Partial distance based FIC)

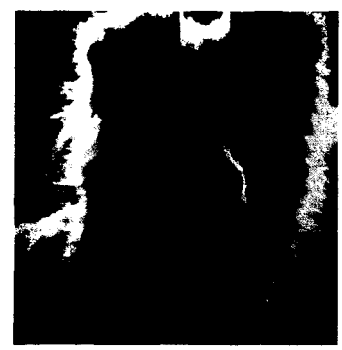
(a)Original lenna image of size 256x256

b) Fisher'squadtree method compr.ratio=7.98 PSNR = 30.06

(c) Partial distance based FIC compr.ratio=7.98, PSNR=29.8dB

Figure(3.11) lenna image of size 256x256

(a)Original collie image



(b) quadtree method, compr.ratio=7.98 PSNR =32.945dB

(c) partial distancebased FIC compr.ratio=7.98, PSNR=32.816dB

Figure(3.12) collie image of size 256x256

3.11 Local Fractal Dimension Based Domain Classification In Fractal Image Compression

In this section, classification of square blocks based on local fractal dimension is proposed. This selects only the 'best matching' domain blocks in advance and computes the transformations only on these blocks. This reduces the encoding time.

3.11.1 Fractal Dimension:

Fractal dimensions are non-integer values and are greater than the topological dimension. They are attempts to quantify a subjective feeling we have about how densely the fractal occupies the metric space in which it lies. Fractal dimensions provide an objective means for comparing fractals. They are important because then can be defined in connection with real-world data and they can be measured approximately by means of experiments [29]. Fractals provide a proper mathematical framework to study the irregular, complex shapes found in nature.

Consider a bounded set A in Euclidean n-space. The set A is said to be self-similar when A is the union of N distinct (non-overlapping) copies of itself, each of which has been scaled down by a ratio r in all coordinates. The fractal or similarity dimension of A is given by the relation

$$1 = N\, r^{D} \quad \text{or} \quad D = \frac{\log N}{\log(1/r)} \tag{3.13}$$

Natural fractal surfaces do not, in general, possess this deterministic self-similarity. Instead, they exhibit statistical self-similarity. That is, they are composed of N distinct subsets each of which is scaled down by a ratio r from the original and is identical in all statistical respects to the scaled original. The fractal dimension for these surfaces is also

given by eqn.(3.13) [36]. Real images and surfaces can not be true mathematical fractals, because the later are defined to exist at all scales. Therefore, a surface is fractal if the fractal dimension is stable over a wide range of scales. One of the important properties of fractal object is their surface area. For pictures, the area of the gray level surface is measured at different scales. The change in measured area with changing scale is used to measure fractal dimension.

3.11.2 Proposed Fast Fractal Image Compression Scheme

The optimal encoding of an image and hence, the optimal quality of the decoded image, requires the selection of best domain block from domain pool. This process takes large time and hence encoding time is very large. The design of efficient domain search techniques has been one of the most active areas of research in fractal image compression, resulting in a wide variety of solutions. Some of the techniques are domain pool reduction [58], classification [33] [43], multiresolution search [63], efficient distance computation [90], nearest neighbor search [74] etc.

In order to reduce the encoding time, it is important to be selective regarding the set of domain blocks in which the best match to a specific range block is to be found. Hence, for a particular range block, search for the best domain block among those domain blocks which possess the same characteristics as the range block. This is called as classification of the image blocks. Both range and domain blocks must be submitted to the same classification algorithm in order to be able to compare blocks of different sizes. Classification based search techniques rely on features that are approximately invariant to the transforms applied. Domain and range blocks may be classified into a fixed number of classes according to features and a matching domain for each range being sought within the same class.

Many classification schemes are possible. Jacquin used a scheme that classified a sub-image into flat, edge and texture regions [33]. In the Fisher algorithm [43, Ch.3] subimage is classified based on average and variance of four quadrants in each block as given in Section 3.9.

In the proposed scheme, local fractal dimension feature is used to classify the sub-image blocks.

The study of changes in picture properties resulting from changes in scale has been accelerated by Mandelbrot's work on Fractals [36]. A theoretical fractal object is self-similar under all magnifications, and property changes, while undergoing scale changes, are limited. Doubling resolution, for example, will always yield an identical change regardless of the initial scale. For pictures, the area of the gray level surface has been measured at different scales. Using this fractal dimension is calculated. Among the several image features, the fractal dimension (FD) of the image is a powerful measure, since it has been shown that the FD has a strong correlation with human judgement of surface roughness. The local fractal dimension is used to classify the domains and ranges into fixed number of classes. The fractal dimension is used as a local feature that possesses the region information.

3.11.3 Estimation of local fractal dimension :

'Blanket' method is used in the proposed algorithm to estimate the local fractal dimension [86], [91]. In the Blanket method, an image is observed as a relief surface in which the heights are proportional to the gray levels of the image. All points at distance ε from both sides of the image surface constitute a blanket of thickness 2ε. The covering blanket is defined by its upper surface u an d its lower surface b . The surface areas of the blanket for different $\mathcal{E}$ values can be repeatedly calculated. At first, the gray level value g(i,j) is given at a pixel (i, j) and we set

$u_0(i,j) = b_0(i,j) = g(i,j)$. Then, the surfaces of the blanket are defined as follows:

$$u_\varepsilon(i,j) = \max\{u_{\varepsilon-1}(i,j)+1, \max_{d(i,j,m,n)\le 1} u_{\varepsilon-1}(m,n)\} \quad (3.14)$$

$$b_\varepsilon(i,j) = \min\{b_{\varepsilon-1}(i,j)+1, \min_{d(i,j,m,n)\le 1} b_{\varepsilon-1}(m,n)\} \quad (3.15)$$

where d(i,j,m,n) is a distance between the pixel (i,j) and the pixel (m,n). The volume of the blanket is computed from equation (3.14) and (3.15) as

$$V_\varepsilon = \sum_{i,j}(u_\varepsilon(i,j) - b_\varepsilon(i,j)) \quad (3.16)$$

By using the volume in equation (3.16), the surface area can be measured as

$$A(\varepsilon) = (V_\varepsilon - V_{\varepsilon-1})/2 \quad (3.17)$$

On the other hand, the area of a fractal surface behaves like

$$A(\varepsilon) = F\varepsilon^{2-D} \quad (3.18)$$

Where F is a constant and D is the fractal dimension of the image. Therefore, the FD can be obtained from the slope of the straight line if $A(\varepsilon)$ versus ε is plotted on a log-log scale. After calculating the FD from local area for each pixel, we can get the FD for each square block.

FD is calculated for all the domain and range blocks and classified into fixed number of classes. The classification of range and domain blocks serves the purpose of reducing the number of domains in the domain pool which need to be considered the'match' for a given range.

3.11.4 Implementation and Results :

Implementation of our method is based on Fisher's quadtree algorithm [43, Ch.3]. The image region is sub-divided into squares using a quadtree data structure. The leaf nodes of the tree correspond to the ranges while the domain pool for a given range consists of image sub squares of twice the size of range block. For each node, a matching domain is

searched in the corresponding domain pool and that node becomes a leaf node if the search yielded a matching satisfying a given tolerance criterion. If the tolerance is not met, the square image region corresponding to the node is broken up into four sub-image squares which become the child nodes of the given node. In Fisher's algorithm, classification of sub-image blocks is based on average pixel intensity, A_i and variance V_i of four quadrants in sub-image.

In proposed method, the classification of sub-image blocks is based on fractal dimension. First, the fractal dimension is computed for each pixel in an image using Blanket method given above. The algorithm for fractal dimension based classification in fractal image compression is as follows:

A: Input: Image of size NxN, range size BxB

B; Global Pre-processing: Compute down filtered image by pixel averaging.

Compute $\sum_{i=1}^{n} d_i^2$, $(\sum_{i=1}^{n} d_i)$ for all code book blocks, where d is the domain pixel.

C: Classify the domain blocks based on fractal dimension using E.

D: Searching : For each range R do steps D1 to D5.

1. Compute $\sum_{i=1}^{n} r_i^2$ and $\sum_{i=1}^{n} r_i$ where n=BxB.
2. Classify the range using step E.
3. For each code book block D, compute $\sum_{i=1}^{n} r_i d_i$ and the coefficients s and o using equation (3.11)

4. Quantize the coefficient s and o and compute the error E(R,D) using equation (3.12).
5. Extract the minimal collage error and output the fractal code for each range.

E: Compute the fractal dimension for each block of size BxB

Based on fractal dimension FD classify domain/range as follows:

1. j=1
2. i= 0 to 4
3. if (i<FD<(FD+0.01) put domain/range into class j
4. i=i+0.01

   j= j+1

   If(i>4) go to end
5. go to step 3
6. End.

The quadtree based Fisher's algorithm [43, Ch.3] and fractal dimension based algorithms are programmed in 'C' language and tested for standard images.

To compare the fractal dimension based classification in fractal image compression with Fisher's quadtree based method, encoding time is considered at different compression ratio. 8-bit gray scale standard images and non-standard imagesare used for testing the algorithm. Table 3.7 and 3.8 shows the results obtained at different compression ratio for the lenna image of size 256x256 and 512x512 and Table 3.9 shows for collie image of size 256x256. Results for non-standard image admistrative building of CREC is given in Appendix D. It is seen that, there is speed up in encoding compared to Fisher's quadtree method with plain classification. Thus, by using fractal dimension for classification significant computation time can be saved while providing image fidelity and compression with only minor degradation. Figure (3.13),and (3.14) shows original and decoded lenna and collie images of size 256x256 at different compression ratio. Figure (3.10), (3.11) and (3.12) shows the performance graphs for lenna and collie images.

Compr. ratio	Fisher's Quad Tree Method			Lfd based FIC			Speed up Factor
	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	
4	680	7.354	30.8	244	9.921	28.2	2.787
8	460	8.064	30	180	10.878	27.4	2.556
12	374	9.584	28.5	160	12.780	26	2.336
20	240	12.78	26	140	15.543	24.3	1.715
30	216	14.674	24.8	120	19.568	22.3	1.8
40	180	16.655	23.7	116	19.568	22.3	1.552

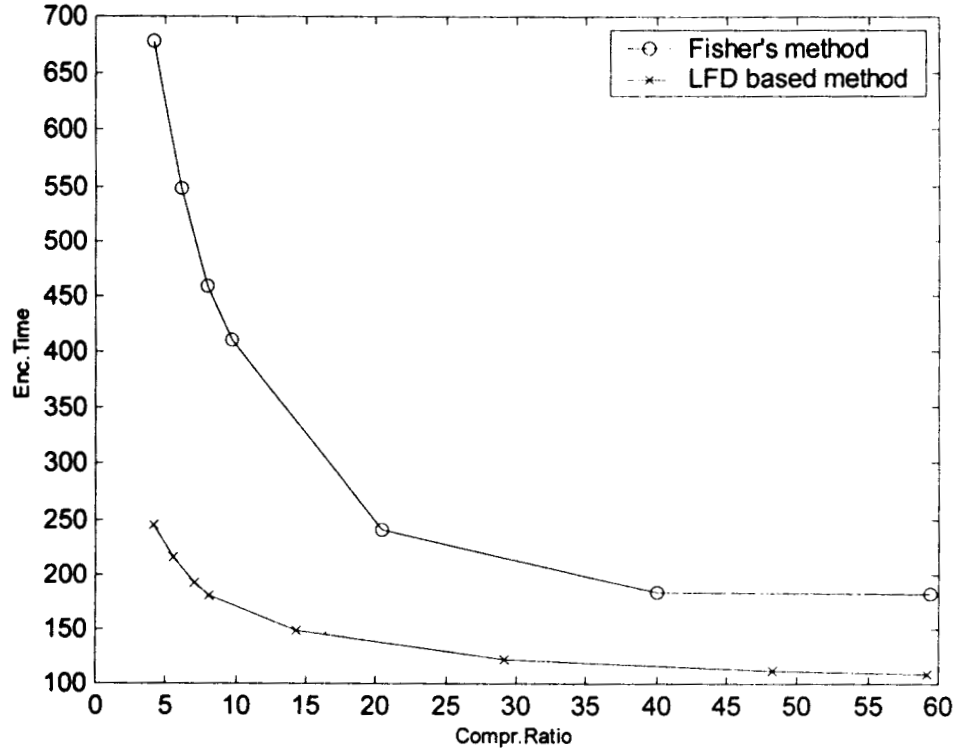
Table (3.7): Encoding time and PSNR for various Compr.ratio for lena image, size 256x256 for Local fractal dimension based Fractal Image Compression

Compr.ratio	Fisher's Quad Tree Method			Lfd based FIC			Speed up Factor
	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	
5.573	657.56	7.354	30.8	263.68	9.798	28.308	2.494
6.25	605.98	7.483	30.649	249.06	9.899	28.219	2.433
6.923	557.4	7.746	30.349	235.49	10.051	28.087	2.367
7.3	527.7	8.064	30	229.3	10.199	27.96	2.301
11.249	368.17	12.692	26.06	193.22	13.189	25.726	1.905
24.834	237.96	19.131	22.496	156.45	18.627	22.728	1.521
52.767	192.34	21.540	21.466	136.54	23	20.896	1.409
63.938	180.87	21.829	21.35	133	24.555	20.328	1.359

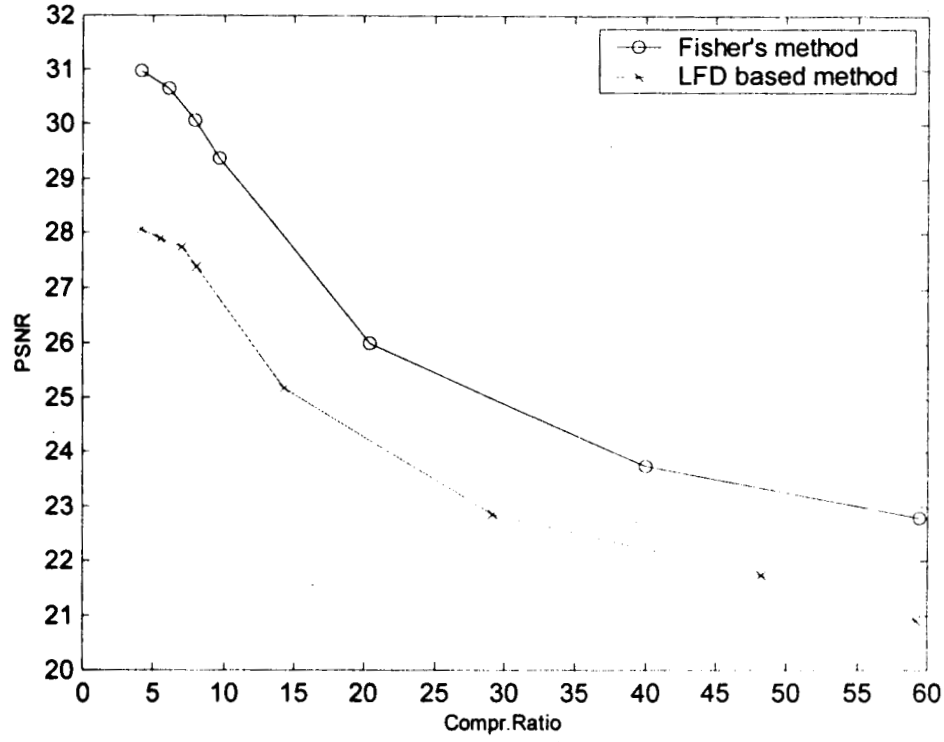
Table (3.8): Encoding time and PSNR for various Compr.ratio for lena image, size 512x512 for Local fractal dimension based Fractal Image Compression

Compr.ratio	Fisher's Quad Tree Method			Lfd based FIC			Speed up Factor
	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	
4	800	3.285	37.8	200	4.535	35	4
8	610	4.184	35.7	170	5.088	34	3.59
12	480	5.088	34	140	6.405	32	3.343
20	330	6.943	31.3	120	8.542	29.5	2.75
30	290	8.064	30	105	10.753	27.5	2.762
40	255	9.258	28.8	100	12.065	26.5	2.55

Table (3.9): Encoding time and PSNR for various Compr.ratio for collie image of size 256x256 for Local fractal dimension based Fractal Image Compression



(a) Plot of Compression Ratio Vs. Encoding Time



(b) Plot of Compression Ratio Vs. PSNR

Figure(3.13) Performance graph for lenna image of size 256x256

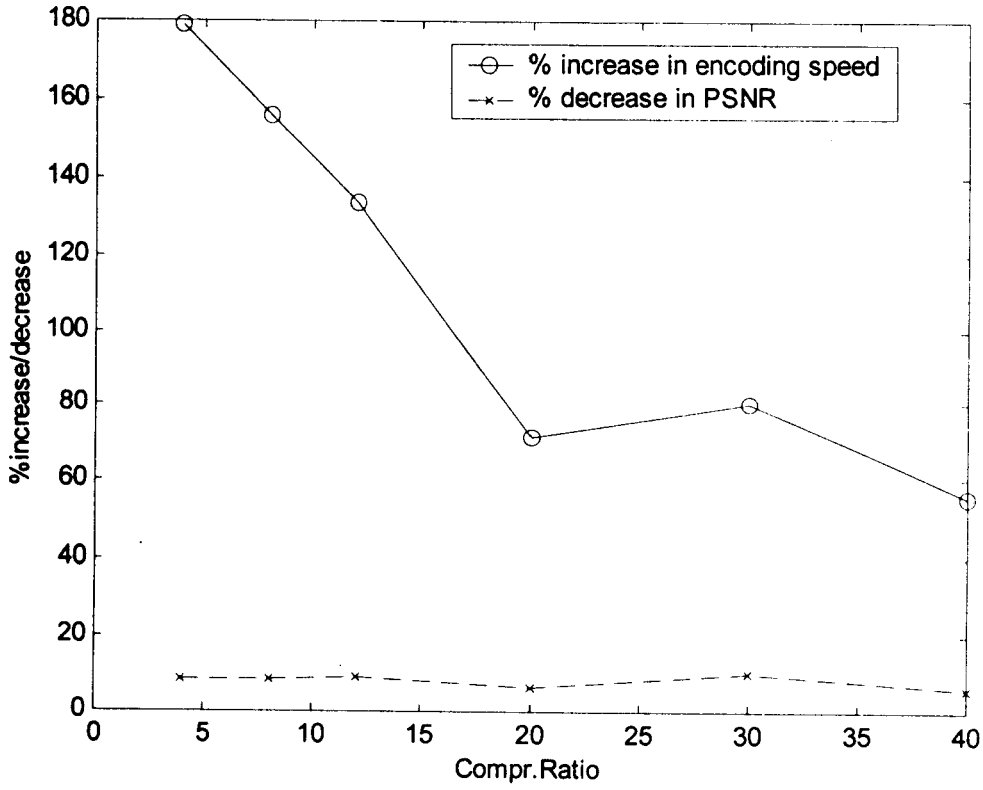
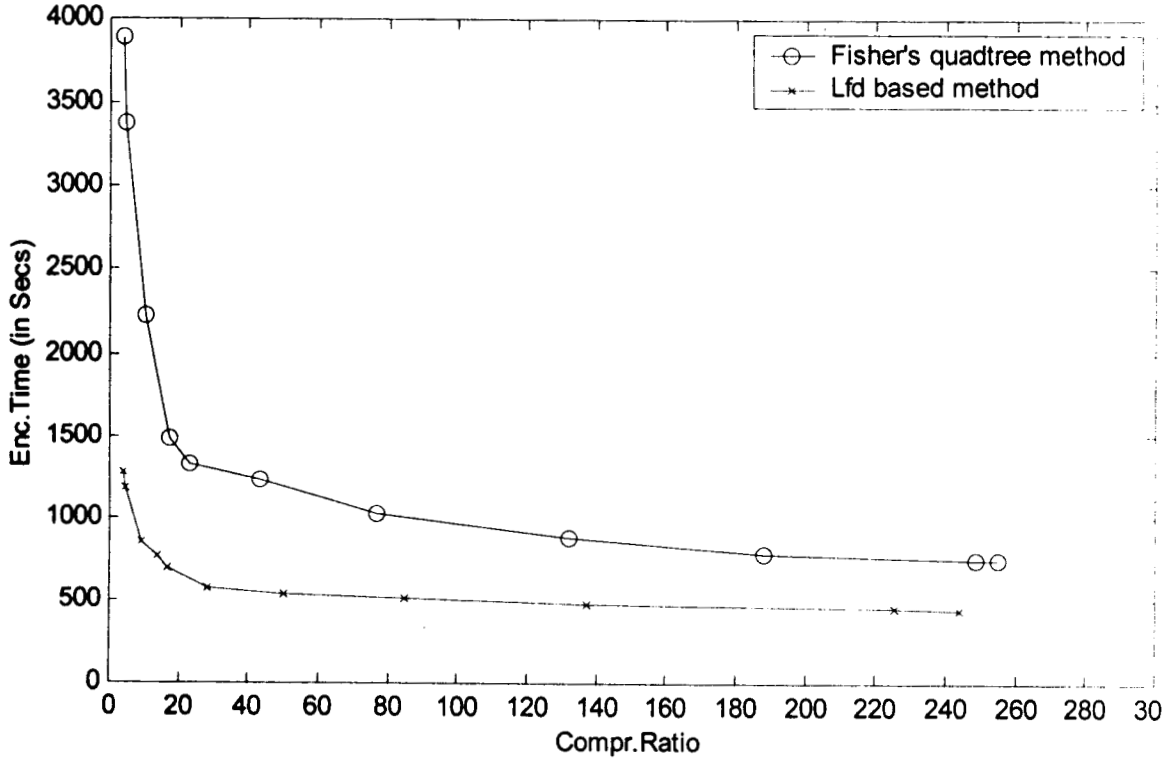
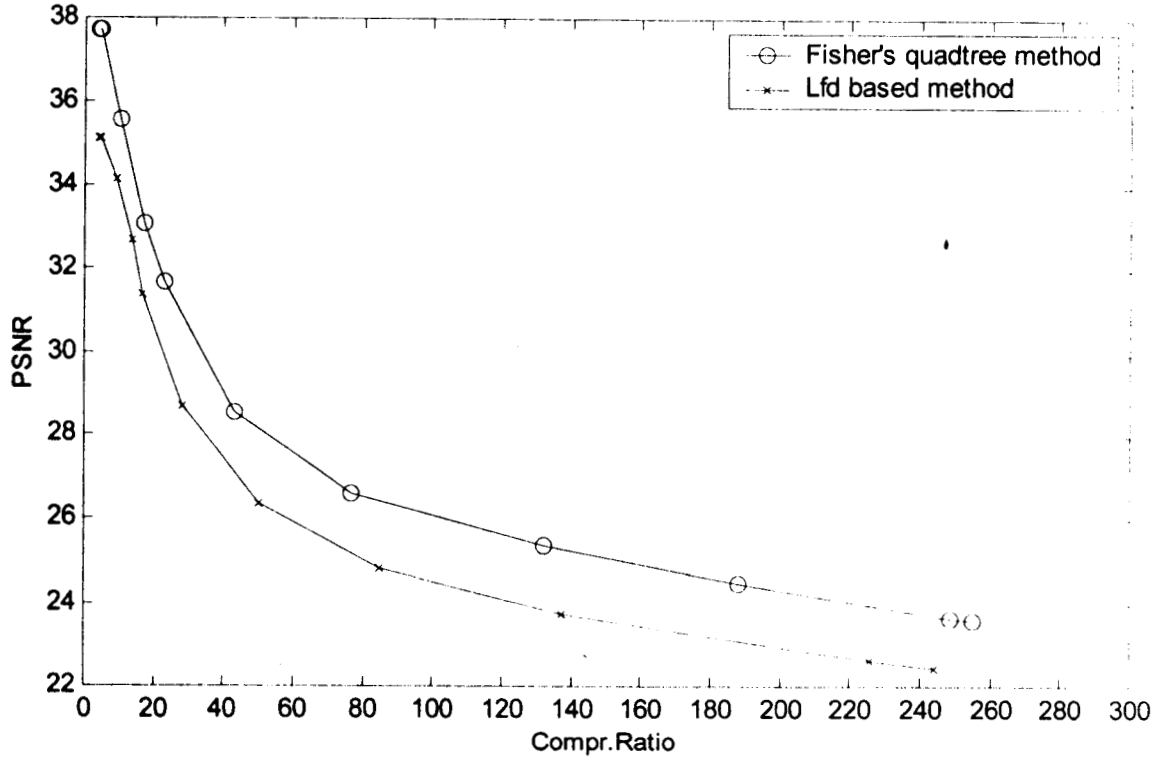


Figure (3.14) Plot of Compression Ratio Vs. % increase in encoding speed and % decrease in PSNR for lenna image of size 256x256 (Lfd based FIC)



(a) Plot of Compression Ratio Vs. Encoding Time



(b) Plot of Compression Ratio Vs. PSNR

Figure (3.15) Performance graph for lenna image of size 512x512

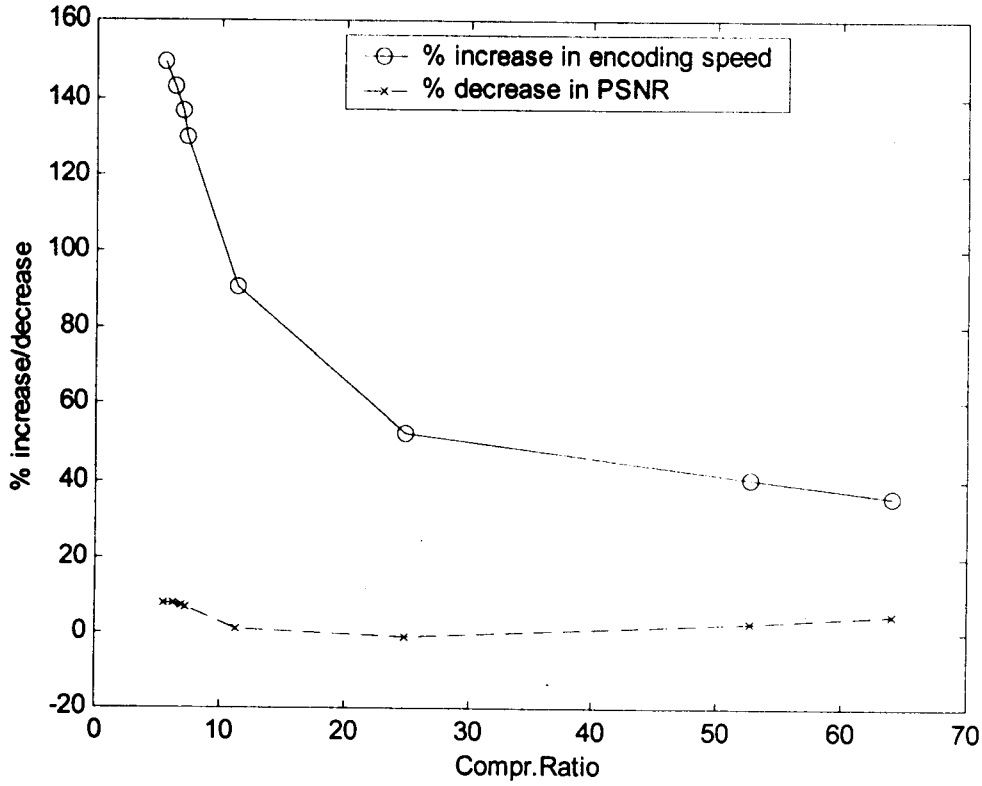
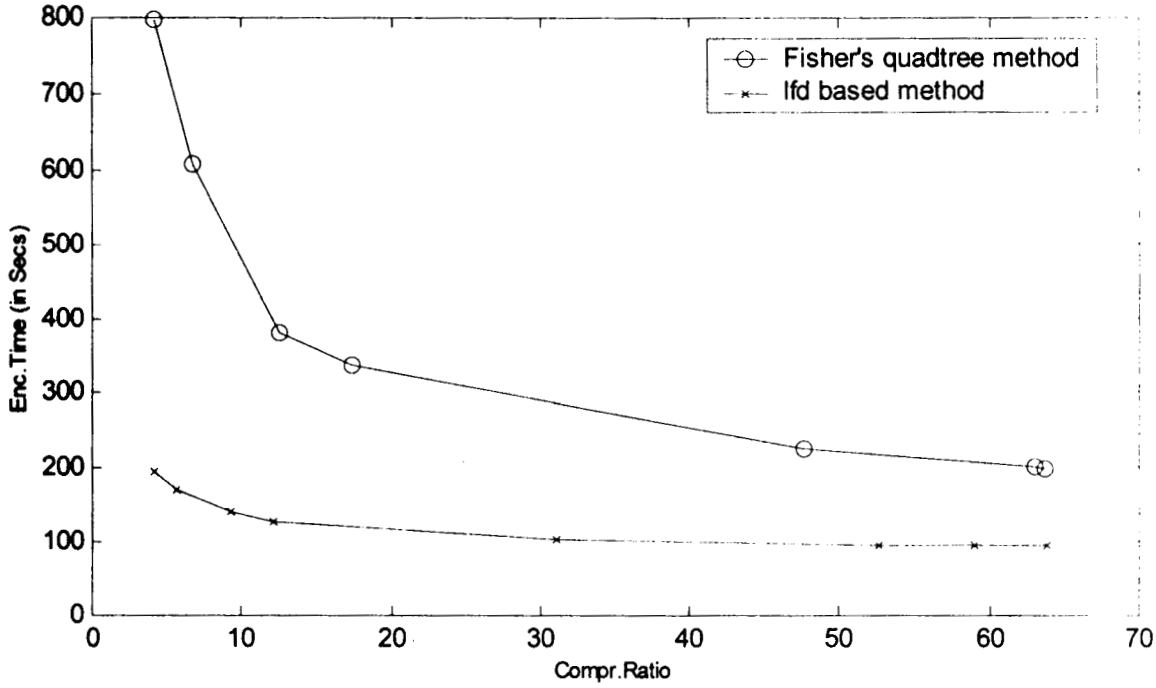
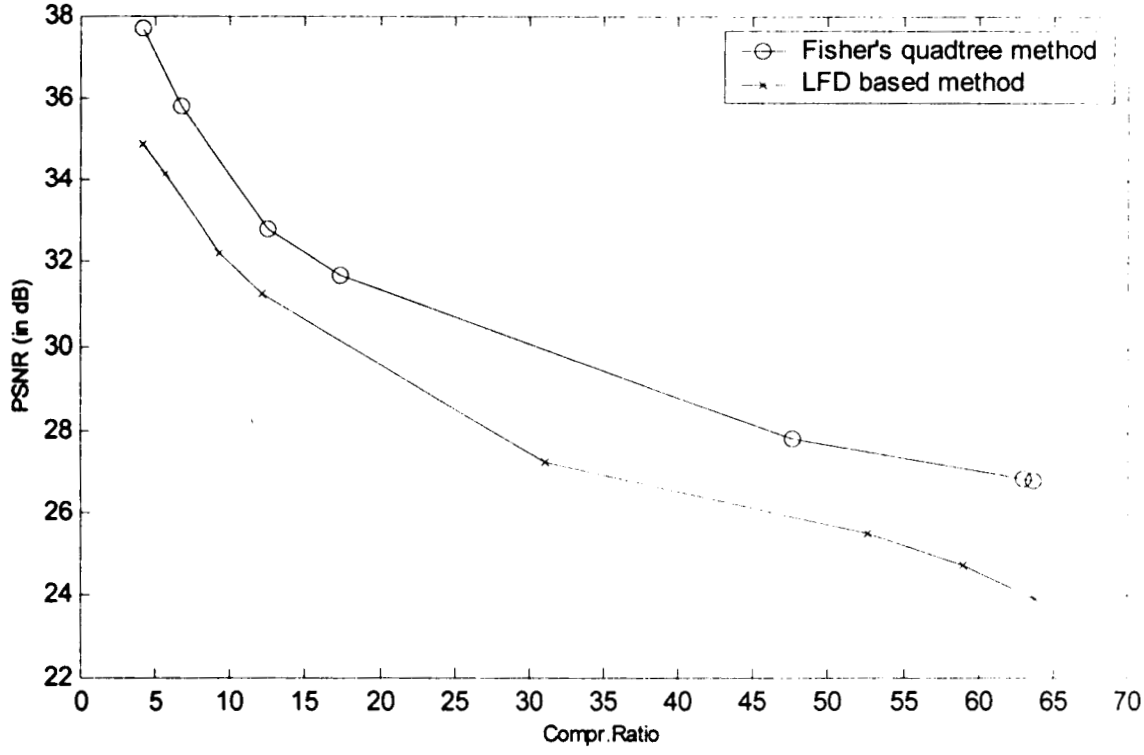


Figure (3.16) Plot of Compression Ratio Vs. % increase in encoding speed and % decrease in PSNR for lenna image of size 512x512 (Lfd based FIC)



(a) Plot of Compression Ratio Vs. Encoding Time



(b) Compression Ratio Vs. PSNR

Figure (3.17) Performance graph for collie image of size 256x256

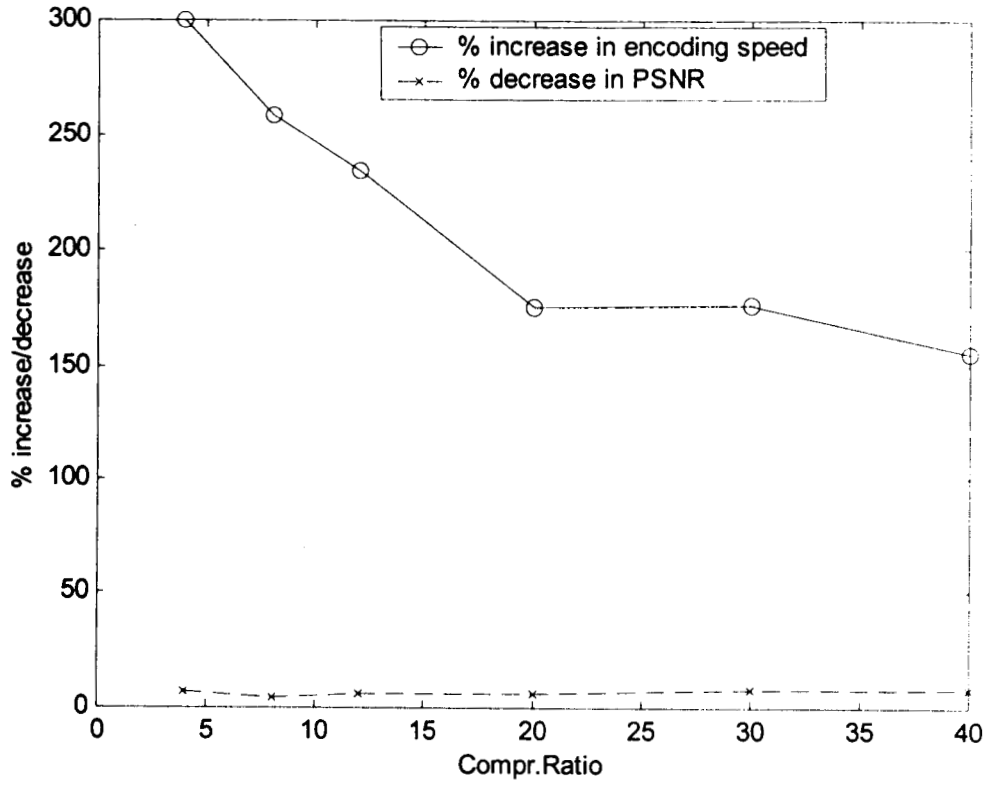


Figure (3.18) Plot of Compression Ratio Vs. % increase in encoding speed and % decrease in PSNR for collie image of size 256x256 (Lfd based FIC)

(a)Original lenna image 256x256
256x256

(b) Fisher's Quadtree method, Compr.Ratio=8

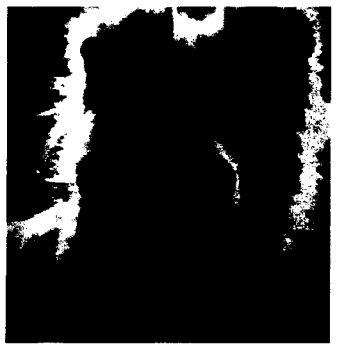
PSNR = 32.816 dB

(c) LFD based method Compr.Ratio =8,

PSNR = 32.002

Figure (3.19)Original and decoded images of lenna of size 256x256

(a)Original collie image



b) Fisher's quadtree method
Compr.ratio=10.2
PSNR= 30.069dB

(c) LFD based method
Compr.Ratio = 10.2

PSNR=27.412dB

Figure(3.20) Original and decoded collie Image o`f size 256x256

REGION-BASED FRACTAL IMAGE COMPRESSION

H.R. Mahadevaswamy “New approaches to image compression ” Thesis.
Department of Electronics Engineering, Regional Engineering College ,
University of Calicut, 2000

Chapter 4

REGION-BASED FRACTAL IMAGE COMPRESSION

4.1 Introduction :

An image independent partition such as the uniform square image partition gives quite a good image representation. The practical approaches fall somewhere between image independent partition and the image dependent segmentation.

As with dividing land into countries and states, the goal here, in image partition is the same: that is managing, controlling, and handling. It involves an art of balancing between idealism and realism, between theoretical clarity and practical naturality. When a straight line was drawn between USA and Canada, it wasn't known how to avoid splitting some one's house between two separate countries. When the border between China and Russia was established along the Amur river, it sparked a conflict lasting a century. Similarly, real-world imaging allows us to segregate a picture either into regular geometric pieces, with no concern for the image content, or into some irregular pieces that are defined ambiguously by some image content segmentation algorithm.

What is the best way to split a picture for image compression ? There are many factors that need to be considered. Mathematical simplicity, engineering manageability, and performance quality are the main elements in choosing a scheme. As a common principle, all methods are intended to cut the complicated and detailed regions into smaller pieces and to group the simple predictable ones together.

Recently, there has been an increasing interest in object based coding, rather than pixel based coding, for image data reduction [68], [69].

4.2 Region-Based Image Coding :

A major characteristic of past and present research activities in image coding is their dedication to block oriented techniques having resulted in a large variety of concepts for both still picture and image sequence coding. This adherence to a fixed block partition has two cardinal reasons. First, the inherent architectural simplicity of block-based image coding makes this concept very appealing for hardware realization. Second, segmenting an image into blocks of equal size requires no overhead information for transmission of the corresponding image partition. Rigid block partitions, however, strongly neglect the image content. As a consequence, increasing data compression inevitably leads to severe blocking artifacts and a rapidly deteriorating image quality. This underlines the flaws in the transform coding methodology: firstly, correlation between blocks is ignored, and secondly, there are many areas in images that cannot easily be described as a realization of a stationary process, such as sharp edges between distinct objects. Novel coding concepts thus try to abandon fixed block partitions in favor of a content related picture representation. The image is segmented into quasistationary regions with uniform texture according to a suitable homogeneity criterion. The internal gray-level and the image partition of resulting segments are coded and transmitted/stored.

Region-based image coding attempts to use a more natural model for image representation [69]. Algorithms for coding images, using a region-based model, assume that certain connected regions in an image will be homogeneous with respect to some property, such as color, and that a set of these regions can completely describe the image. A segmentation algorithm is used to identify the homogenous regions, and classify every

pixel in the image as belonging to one of these regions. A border coding is then used to code the boundaries between these regions. Once the borders have been coded, the contents, or values, of the homogeneous region can also be coded, thereby forming a complete, region based description of the image. As the number of regions is usually considerably less than the number of pixels in the original image, the data required to code the image is also much less than the data used in the raw format. Fig. 4.1 shows the process for coding a single image.

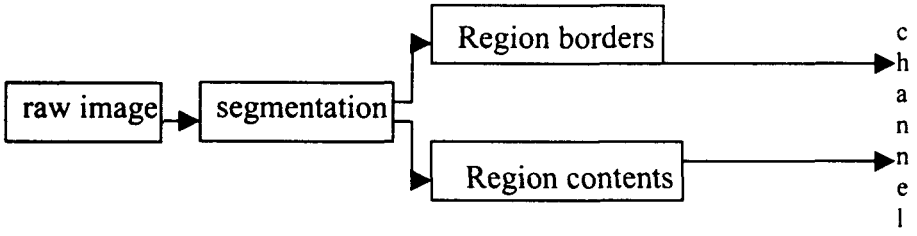


Fig. 4.1 Block diagram of region-based image coding.

Segmentation has been used in object based coding. It can be performed in many ways: contour extraction, split and merge, region growing, etc. All these methods do not, of course, give the same results, but they all attempt to extract regions whose boundaries match the borders of the objects in the scene. There are many segmentation algorithms based on reducing a region adjacency graph [69], that attempt to minimize the mean square error between the original image and the resulting reconstruction.

Many segmentation algorithms are based on a merging process, where a pair of regions from an initial set are chosen to be merged, forming a new set with one region less.

Merging then continues until the desired number of regions is produced. The standard merge order is determined purely by the minimum distortion induced over all possible merges. By using the knowledge of the border coding process in the segmentation procedure to affect the order in which regions are merged, better rate-distortion can be obtained [70]. Region-based image coding based on this segmentation method provides better rate-distortion values.

The split and merge method can be applied to image sequences, to reduce the information contained in an image sequence, while retaining the information necessary for the interpretation of the sequence by a human observer [71]. To find the regions, a 3-D split and merge algorithm is used. This 3-D split and merge algorithm consists of starting from an initial region space and then merging these regions according to the properties they share. A region is represented by the co-efficients of its approximating polynomial and by the description of its 3-D border.

The coding scheme based on shape independent basis functions is proposed by A. Kaup [72]. To decouple texture description and contour coding, the texture inside each region is approximated using shape independent basis functions that are defined on a rectangle circumscribing the image segment. The resulting texture representation is independent of the exact contour information and has a high energy compactness, making it well suited for low bit rate image coding. The computational load is low, especially for the decoding process.

4.3 Region -Based Fractal Image Compression :

Progress in fractal coding is only as much as in the non-intelligent compression schemes. Non-intelligent compression schemes make no assumptions about the data they

are handling. All such schemes are limited in the amount of compression they can achieve. To make an order-of-magnitude leap in terms of image compression, it is widely believed that object-oriented schemes would have to be used. In an object-oriented coding, data would be sent representing the types of objects in the image, instead of sending data representing gray-levels of pixels in the image. To encode a picture of a red mug for example, a parameter for a mug would be required, and a parameter for the color red; these would be transmitted to the decoder, which would draw a red mug. The problem with this scheme is that the decoder does not know what type of red mug it is, where in the scene it is, and from what angle it is being viewed, etc. However all this extra data can be parameterized if necessary. The point is that only the minimum number of parameters that are important for the application are transmitted from encoder to decoder. Fractals fit into this vision in the following way. A decoder drawing a red mug would have to use computer graphics techniques. The surface of the mug would be segmented into regions that are homogeneous and the boundaries of these regions would be drawn first. The regions would then be rendered with different textures in this case, shades of red depending on an arbitrary set of lighting conditions [73].

The first practical fractal image compression system to appear in the literature was by Jacquin, who used transformations defined on image blocks to encode image data [32], [33]. Most of the fractal image compression systems to appear in literature are based on Jacquin's block-based compression system [73].

The region-based partition of the image increases the compression ratio over that obtainable by traditional block-based partitioning [34]. The block-based system demonstrates the potential of using fractal techniques for image compression. The

algorithm proposed by Thomas [34] extends the block-based scheme by allowing a different class of transformations. The image transformations used are not fixed size square block-based transformation but region-based transformations. These regions can be large and irregularly shaped segments of the image.

The digital image μ of size r x r is partitioned into non overlapping regions of variable size and shape. These regions form the range of the transformations. Each range region is composed of a group of square range blocks of size 8x8 pixels. The actual construction of a partition is image dependent. The image partition is denoted as $\{R_i\}_{i \in c}$, where c is the set of possible range regions indices obtained from uniform partitioning and each R_i is a transformation range region. For each range region, there is a corresponding domain region denoted D_i. The transformation from D_i to R_i is denoted τ_i.

The general form of the image transformation τ is a union of domain to range region transformations

$$\tau = \bigcup_{i \in c} \tau_i = \bigcup_{i \in c} s_i o \tau_i$$

where s_i is a spatial transformation, and τ_i is a gray-level transformation that processes image regions supported on D_i and maps them onto R_i.

Given an original image μ to encode, the problem consists of finding among the class of region-based image transformations defined previously: a transformation τ that leaves μ approximately invariant, i.e., which minimizes the distance $d(\mu, \tau(\mu))$.

Since the image transformations have the form $\tau = \bigcup \tau_i$

The problem is to cover the image with non-overlapping region based transformations such that the distance

$$d(\mu(R_i), \tau_i[\mu(D_i)])$$

is minimized, where D_i is the domain, and R_i is the range region of the transformation.

Since compression is inversely dependent on the number of regions, it is essential to make the regions as large as possible. In fact, there is a compression ratio – quality tradeoff in the size of the image regions.

With the block-based scheme, it is possible to perform an exhaustive search of the space to find suitable transformations. Although this always results in finding the best transformation for that class, it is a very time consuming operation. With the new class transformations (after region based partitioning), it is impractical to search entire space for the optimal solution. Due to the large search space involved, heuristic algorithms are used to perform region based transformations [34].

The decoding scheme for the region-based system is similar to the block-based system. It simply consists of iterating the transformation τ on any initial image μ_o until convergence to a final decoded image is observed. The only difference is that the transformation τ is made up of variable size and shape regions as opposed to fixed size range blocks. The region-based system achieves almost double the compression ratio over the simple block based scheme with a similar image quality. Despite being faster than the block-based scheme, the encoding still takes an order of magnitude longer than the decoding [34].

Region-based partition of the images increases the compression ratios compared to block-based partitioning. In this method, region-based transformations are used instead of fixed-size square block-based transformations. The regions can be large and irregularly shaped segments of the image. Literature survey of region-based fractal image compression indicates the following:

i) The larger the region size, larger the compression ratio. Hence it is desirable to make the regions as large as possible.

ii) It is impractical to search the entire space for the optimum solution with the class of transformation used in Region-based fractal image compression. Hence, it is necessary to develop a new search method.

A Region based FIC with quadtree segmentation proposed by Yung-Ching Chang et al. [97], quadtree segmentation by adaptive threshold is employed. The merging algorithm was developed to the resulting quadtree segmentation that combines several similar blocks into a small number of regions. To represent the shapes of the region quadtree based segmented chain code is used.

In adaptive partitioning for fractal image compression proposed by Ruhl et al. [98], neighboring ranges are iteratively merged to yield sequence of partitionings with a decreasing number of ranges. During this process in addition to maintaining a partitioning, an associated fractal code is also maintained. After each merge, next fractal code is obtained by local modification of the previous one. Merging is done by taking ranges whose merger results in the least increase of collage error.

4.4 Region-based fractal Image Compression by using Fractal dimension feature in merging of range blocks

Second generation or segmentation based image coding [68] methods take into account the human visual system and its property to differentiate between edges and texture. In segmentation based image coding, the image partition represents smooth image regions that can be sufficiently well approximated by low order polynomials.

The review of literature reveals that, fractal coding can benefit from highly flexible and image adaptive partition methods.

In this work, local fractal dimension is used as a feature for region merging and chain coding given in[50] , [100] is used to code the shape of the regions.

4.4.1 Fractal Dimension Based Image Partitioning

In any partitioning or segmentation procedure two components are important: The selection of image features and a segmentation algorithm. Among the several features used in the segmentation, the fractal dimension (FD) of the image is a powerful measure of roughness in natural images, since it has been shown that FD has a strong correlation with human judgment of surface roughness [99].

The Algorithm begins by uniformly tiling the image with square range blocks as shown in Figure 4.1 (a). Then Fractal Encoding for uniform partition is done. One of the range blocks is chosen as a seed and an attempt is made to search for a suitable domain to range block transformation in a similar way to the block based scheme. The algorithm attempts to extend the seed in all four principle directions, using fractal dimension feature. Fractal dimension for each block is found using Blanket method given in Chapter 3. Starting from the initial seed the neighboring range block is merged

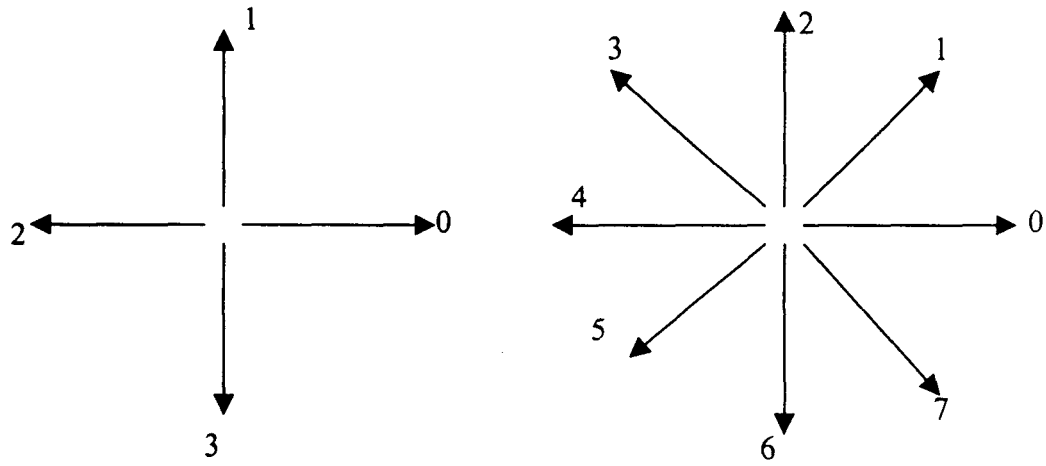
if $| D(R_i) - D(R_j) |$ T_{FD} where $D(R_i)$ is the average FD value of R_i and R_j respectively, T_{FD} is a predetermined threshold. For merged range region, search for the optimal domain is performed. It is computationally infeasible to use corresponding full domain pool. In order to obtain a matching domain block for the union of two ranges, only domains at positions that are given by the lists of domain addresses inherited from the two ranges is used. These domains have to be extended to match the size of the new range.

- Find fractal code and update partition.
- Merging is continued till the RMS error between range region and domain region is less than threshold set or required compression ratio is reached.
- Then select another seed which is not merged and repeat the process.
- Output the partition using chain coding, optimum luminance parameters and isometry options (as given in Section 3.8)

4.5 Coding of the partition information in Region-based Fractal Image Compression:

In order to efficiently encode partition obtained in region-based fractal image compression, chain coding is used. Using chain codes [2] the boundaries between the ranges are traced resulting in a stream of symbols that represent a starting point and actions like turn right, go a head and so on. The chain codes have been used to digitally represent curves [2].

4.5.1 CHAIN CODING



a) 4- directional chain-code b) 8-directional chain-code

Figure (4.2) Chain coding

Chain code consists of line segments that must lie on a fixed grid with a fixed set of possible orientations. Only starting point is represented by its location, other points on a curve are represented by successive displacements from grid point to grid point along the curve. Since the grid is uniform, direction is sufficient to characterize displacement.
The grid is usually considered to be four or eight connected and directions are assigned as in Fig(4.2). Each direction can be represented by two or three bits.

The starting point in Fig (4.3) (a) is at the dot, and the chain code for this boundary representation is shown in Fig(4.3) (b).

The chain code of a boundary depends on the starting point. For a chain code generated by starting at an arbitrary position, it is treated as a circular sequence of direction numbers and redefine the starting point so that the resulting sequence of direction numbers forms an integer of minimum magnitude. The derivative of the chain

The problem is to cover the image with non-overlapping region based transformations such that the distance

$$d(\mu(R_i), \tau_i[\mu(D_i)])$$

is minimized, where D_i is the domain, and R_i is the range region of the transformation.

Since compression is inversely dependent on the number of regions, it is essential to make the regions as large as possible. In fact, there is a compression ratio – quality tradeoff in the size of the image regions.

With the block-based scheme, it is possible to perform an exhaustive search of the space to find suitable transformations. Although this always results in finding the best transformation for that class, it is a very time consuming operation. With the new class transformations (after region based partitioning), it is impractical to search entire space for the optimal solution. Due to the large search space involved, heuristic algorithms are used to perform region based transformations [34].

The decoding scheme for the region-based system is similar to the block-based system. It simply consists of iterating the transformation τ on any initial image μ_o until convergence to a final decoded image is observed. The only difference is that the transformation τ is made up of variable size and shape regions as opposed to fixed size range blocks. The region-based system achieves almost double the compression ratio over the simple block based scheme with a similar image quality. Despite being faster than the block-based scheme, the encoding still takes an order of magnitude longer than the decoding [34].

Region-based partition of the images increases the compression ratios compared to block-based partitioning. In this method, region-based transformations are used instead of fixed-size square block-based transformations. The regions can be large and irregularly shaped segments of the image. Literature survey of region-based fractal image compression indicates the following:

i) The larger the region size, larger the compression ratio. Hence it is desirable to make the regions as large as possible.

ii) It is impractical to search the entire space for the optimum solution with the class of transformation used in Region-based fractal image compression. Hence, it is necessary to develop a new search method.

A Region based FIC with quadtree segmentation proposed by Yung-Ching Chang et al. [97], quadtree segmentation by adaptive threshold is employed. The merging algorithm was developed to the resulting quadtree segmentation that combines several similar blocks into a small number of regions. To represent the shapes of the region quadtree based segmented chain code is used.

In adaptive partitioning for fractal image compression proposed by Ruhl et al. [98], neighboring ranges are iteratively merged to yield sequence of partitionings with a decreasing number of ranges. During this process in addition to maintaining a partitioning, an associated fractal code is also maintained. After each merge, next fractal code is obtained by local modification of the previous one. Merging is done by taking ranges whose merger results in the least increase of collage error.

4.4 Region-based fractal Image Compression by using Fractal dimension feature in merging of range blocks

Second generation or segmentation based image coding [68] methods take into account the human visual system and its property to differentiate between edges and texture. In segmentation based image coding, the image partition represents smooth image regions that can be sufficiently well approximated by low order polynomials.

The review of literature reveals that, fractal coding can benefit from highly flexible and image adaptive partition methods.

In this work, local fractal dimension is used as a feature for region merging and chain coding given in[50] , [100] is used to code the shape of the regions.

4.4.1 Fractal Dimension Based Image Partitioning

In any partitioning or segmentation procedure two components are important: The selection of image features and a segmentation algorithm. Among the several features used in the segmentation, the fractal dimension (FD) of the image is a powerful measure of roughness in natural images, since it has been shown that FD has a strong correlation with human judgment of surface roughness [99].

The Algorithm begins by uniformly tiling the image with square range blocks as shown in Figure 4.1 (a). Then Fractal Encoding for uniform partition is done. One of the range blocks is chosen as a seed and an attempt is made to search for a suitable domain to range block transformation in a similar way to the block based scheme. The algorithm attempts to extend the seed in all four principle directions, using fractal dimension feature. Fractal dimension for each block is found using Blanket method given in Chapter 3. Starting from the initial seed the neighboring range block is merged

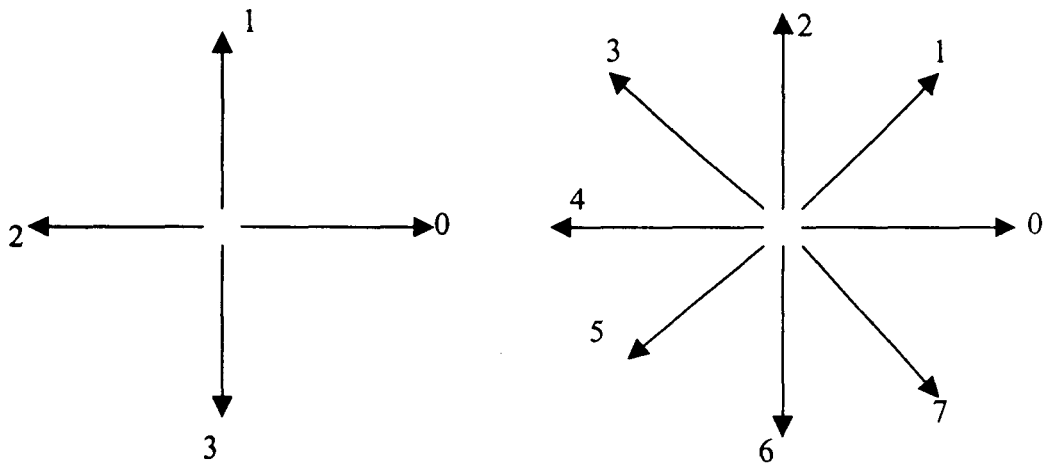
if $| D(R_i) - D(R_j) |$ T_{FD} where $D(R_i)$ is the average FD value of R_i and R_j respectively, T_{FD} is a predetermined threshold. For merged range region, search for the optimal domain is performed. It is computationally infeasible to use corresponding full domain pool. In order to obtain a matching domain block for the union of two ranges, only domains at positions that are given by the lists of domain addresses inherited from the two ranges is used. These domains have to be extended to match the size of the new range.

- Find fractal code and update partition.
- Merging is continued till the RMS error between range region and domain region is less than threshold set or required compression ratio is reached.
- Then select another seed which is not merged and repeat the process.
- Output the partition using chain coding, optimum luminance parameters and isometry options (as given in Section 3.8)

4.5 Coding of the partition information in Region-based Fractal Image Compression:

In order to efficiently encode partition obtained in region-based fractal image compression, chain coding is used. Using chain codes [2] the boundaries between the ranges are traced resulting in a stream of symbols that represent a starting point and actions like turn right, go a head and so on. The chain codes have been used to digitally represent curves [2].

4.5.1 CHAIN CODING



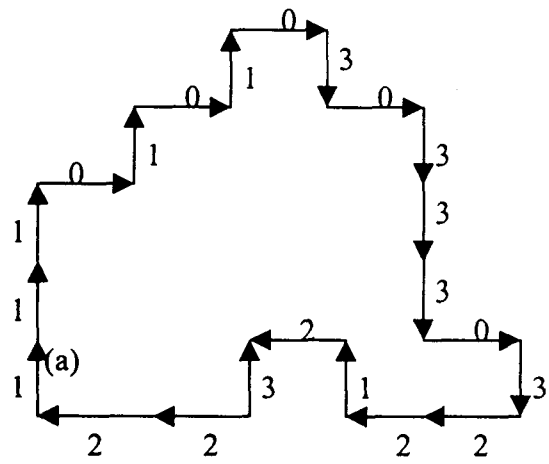
a) 4- directional chain-code b) 8-directional chain-code

Figure (4.2) Chain coding

Chain code consists of line segments that must lie on a fixed grid with a fixed set of possible orientations. Only starting point is represented by its location, other points on a curve are represented by successive displacements from grid point to grid point along the curve. Since the grid is uniform, direction is sufficient to characterize displacement.
The grid is usually considered to be four or eight connected and directions are assigned as in Fig(4.2). Each direction can be represented by two or three bits.

The starting point in Fig (4.3) (a) is at the dot, and the chain code for this boundary representation is shown in Fig(4.3) (b).

The chain code of a boundary depends on the starting point. For a chain code generated by starting at an arbitrary position, it is treated as a circular sequence of direction numbers and redefine the starting point so that the resulting sequence of direction numbers forms an integer of minimum magnitude. The derivative of the chain



Chain Code : 1110101030333032212322

Derivative: 100313133130013303113O

(b)

Figure (4.3) Chain code and its derivative for boundary shown.

code is useful because it is invariant under boundary rotation. The derivative (a first difference mod 4 or 8) is simply an another sequence of numbers indicating the relative direction of chain code segments i.e., number of left hand turns of $\pi/2$ or $\pi/4$ needed to achieve the direction of the next chain segment.

In Region-based fractal Image Compression by using Fractal dimension feature in merging of range blocks derivative chain code (DCC) given below is used [50] to store the partition information. It proceeds by tracking the range boundaries by storing the necessary movements.

METHOD 1 : The following procedure is iterated until the range boundaries are completely described. First, an unvisited point on the range boundary is selected, its position is output and it pushed on to stack. As long as the stack is not empty, the following steps are repeated .The stack is popped, and starting at the given position, the

boundary is tracked until an already visited position is reached. At every step, output a symbol showing which of the three directions (turn left, continue straight ahead, turn right) it can take next. At contour branching points, take first branch and push the end points of the other available line segments on the stack for later perusal. Since at even step the contour must continue in at least one of the three directions, $2^3-1=7$ symbols are needed for the encoding of a step.

At every step output only possible movements (one bit each) & omit the redundant information. The resulting symbol string is arithmetic entropy coded [2].

4.6 IMPLEMENTATION AND RESULTS

Region-based fractal image compression by fractal dimension feature in merging range blocks algorithm is tested for various standard and non-standard 8-bit gray images. First, fractal codes are found with uniform partition and then region merging based on fractal dimension feature is carried out. The results are given for image lenna of size 256x256.

Following parameters are chosen for testing the algorithm:

In the initialisation phase, the image is uniformly partitioned into blocks of size j x j and partitioning is performed for j=2, 4, 8. The domain pool considered for the initial fractal code consists of blocks of the uniformly partitioned image with block size 2j X 2j plus isometric versions.

Fig (4. 4) and (4.5) show variation of encoding time and PSNR with Compression ratio for Lena image range blocks basesize 2 and 8. It shows that fractal dimension based range blocks merging yields better results in terms of encoding time, compression ratio and PSNR. Table 4.1 and 4.2 shows the results for various values of bases size for lenna image

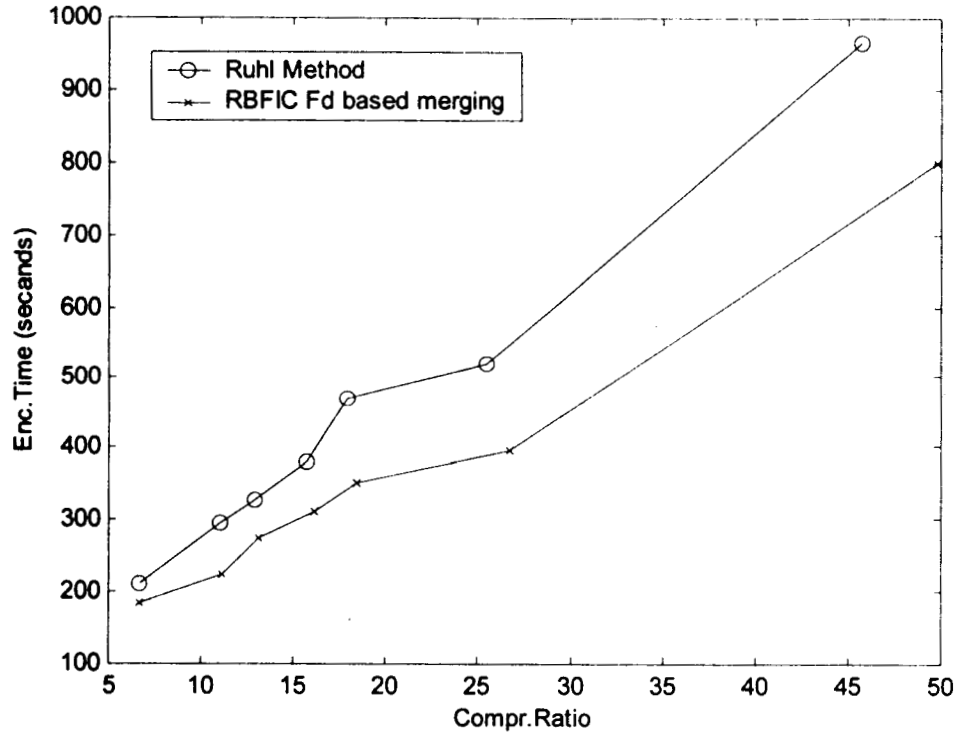
with size 256x256. Number of ranges decides the compression ratio and with smaller bases size quality of the image is good, but takes more time to encode. Results for non-standard image administrative block of CREC is given in Appendix D.

Compr. ratio	Ruhl's RBFIC Method			FD based RBFIC			Speed up Factor
	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	
30	36	11.72	26.75	29	10.88	27.4	1.24
40	40	12.07	26.5	32	11.13	27.2	1.25
60	50	13.54	25.5	43	12.78	26	1.16
80	53	15.19	24.5	49	14.34	25	1.08
100	63	16.56	23.75	55	15.72	24.2	1.15
150	77	19.79	22.2	70	18.9	22.6	1.1
180	85	21.7	21.4	78	20.73	21.8	1.09
200	91	22.73	21	85	22.08	21.25	1.07

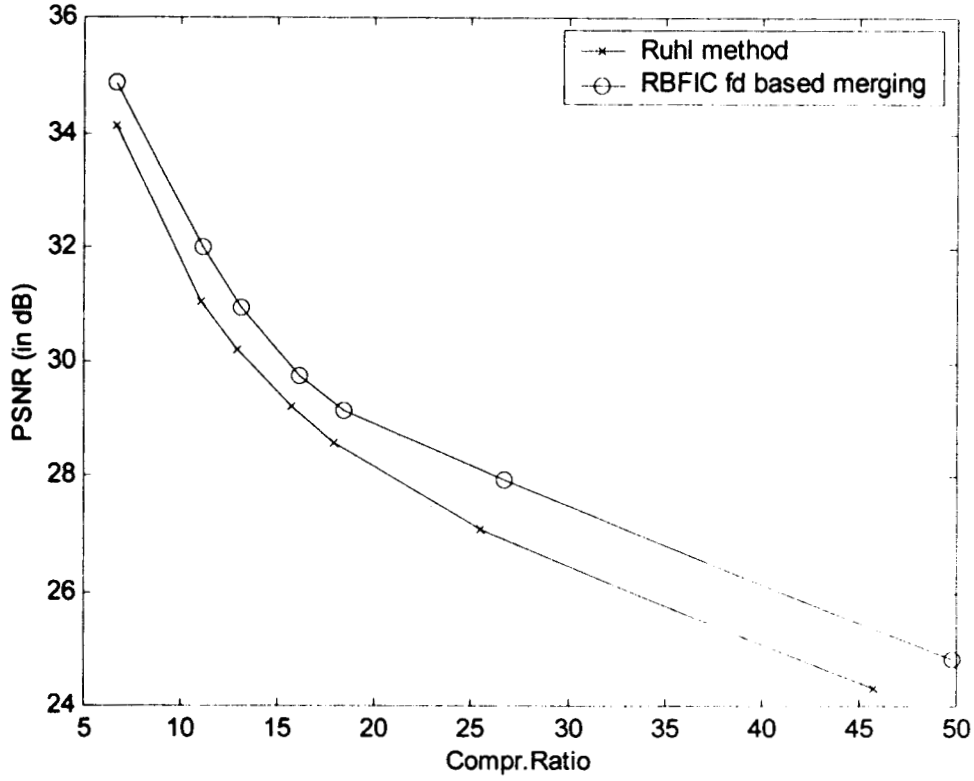
Table 4.1 Results for base size=8 for lenna image of size 256x256 Region-based fractal image compression based on fractal dimension for various values of compr.ratio

Compr. ratio	Ruhl's RBFIC Method			FD based RBFIC			Speed up Factor
	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	Enc.Time (in Secs)	RMS Error	PSNR (in dB)	
10	225	6.59	31.75	220	5.88	32.75	1.25
15	360	8.54	29.5	300	7.88	30.2	1.2
20	480	9.92	28.2	360	9.15	28.9	1.33
30	625	12.07	26.5	460	10.75	27.5	1.36
40	840	14.01	25.2	640	12.49	26.2	1.31
45	950	15.37	24.4	725	13.54	25.5	1.31

Table 4.2 Results for base size=2 for lenna image of size 256x256 Region-based fractal image compression based on fractal dimension for various values of compr.ratio



(a) Plot of Compression Ratio Vs. Encoding Time



(b) Plot of Compression Ratio Vs. PSNR

Figure (4.4) Performance graph for lenna image of size 256x256, basesize=2

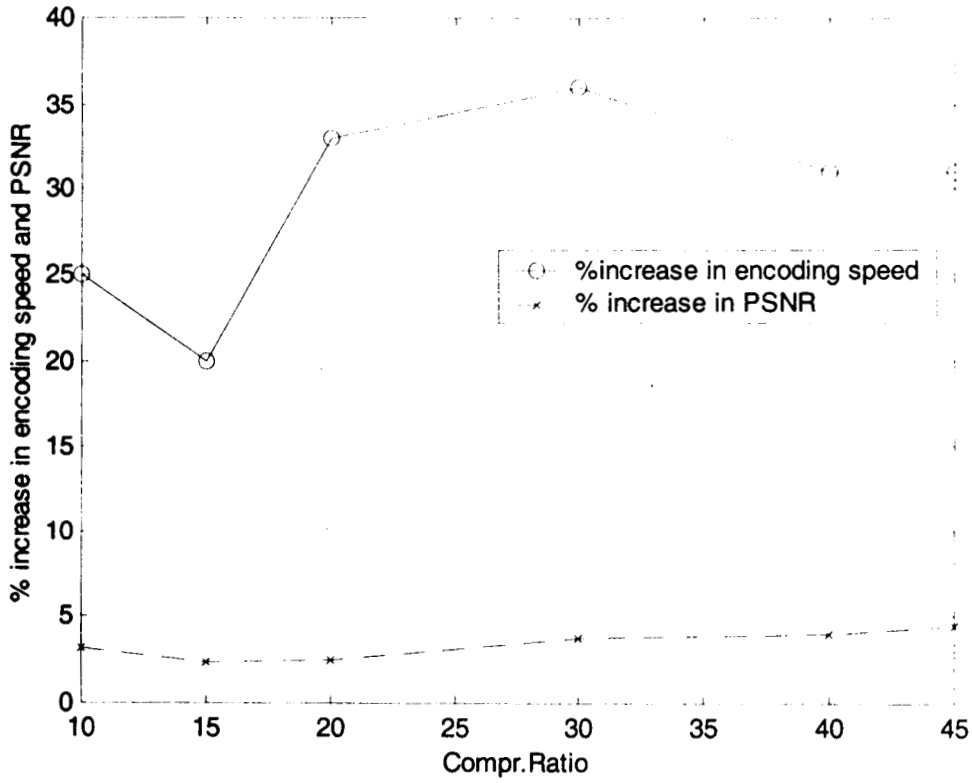
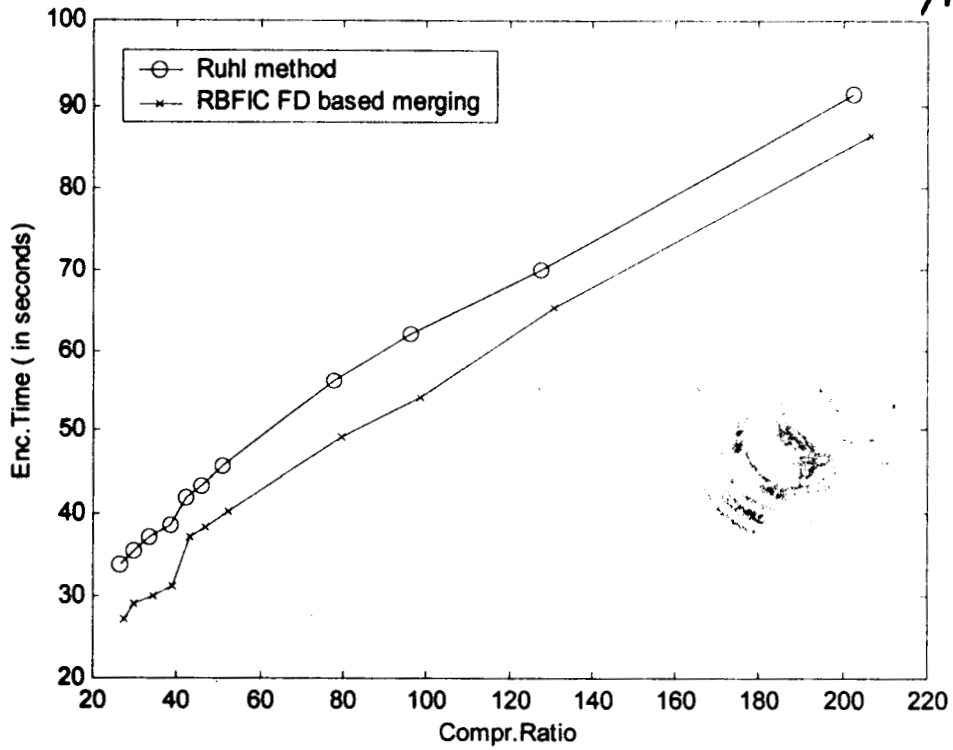
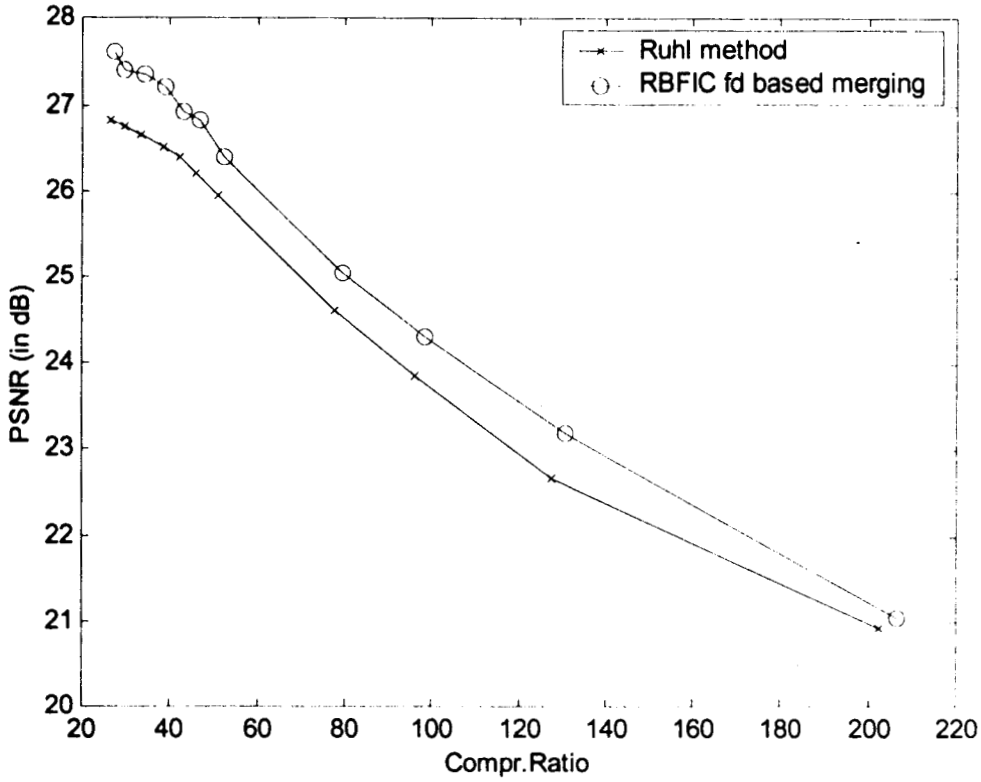


Figure (4.5) Plot of Compression Ratio Vs. % increase in encoding speed and PSNR for lenna image of size 256x256 for basesize =2(Fractal dimension based RBFIC)



(a) Compression Ratio Vs. Encoding Time



(b) Plot of Compression Ratio Vs. PSNR

Figure(4.6) Performance graph for lenna image of size 256x256, basesize =8

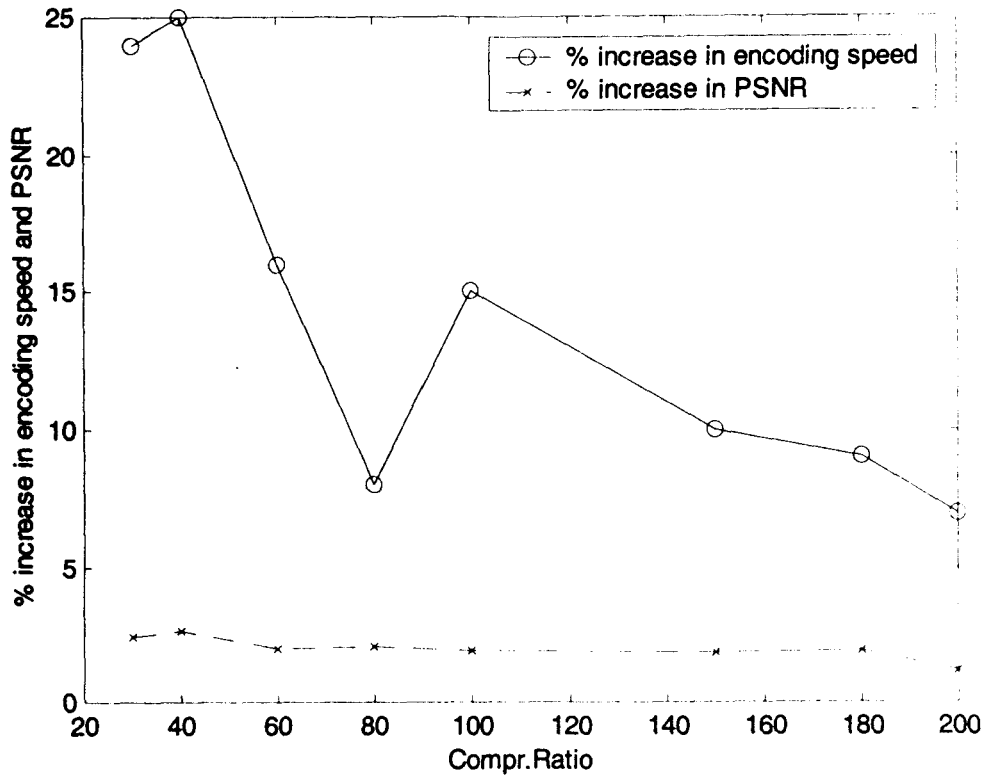


Figure (4.7) Plot of Compression Ratio Vs. % increase in encoding speed and PSNR for lenna image of size 256x256 for basesize =8 (Fractal dimension based RBFIC)

(a) Original lenna Image of size 256x256

(b)Decoded image at Compr.Ratio=40
PSNR=30.5dB using Ruhl method

(c) Decoded image at Compr.Ratio=40
PSNR=30.8dB using FD based method

Figure (4.8) Original and Decoded images for basesize 2

FRACTAL IMAGE COMPRESSION OF NOISY IMAGES

H.R. Mahadevaswamy “New approaches to image compression ” Thesis.
Department of Electronics Engineering, Regional Engineering College ,
University of Calicut, 2000

Chapter 5

FRACTAL IMAGE COMPRESSION OF NOISY IMAGES

5.1 Introduction :

In many of the image storage and transmission applications, the data to be compressed is corrupted by noise. For example, in medical imaging, emission and transmission tomography images are usually corrupted by data dependent Poisson noise. Images scanned from photographic films are corrupted by data dependent film-grain noise while scanning. Noise degrades the performance of the image compression algorithms [87] [92]. Some noisy source coding algorithms used spatial prefiltering for reducing the effect of noise [87] [93] .

In this Chapter, impact of film-grain noise on Fractal Image Compression is demonstrated. The focus of this Chapter is on Fractal Image Compression of film-grain noise corrupted images.

5.2 Effect of Noise on Image Compression :

Images are, in many cases, degraded even before they are encoded. Noise degrades the performance of image compression algorithms. There are two main reasons for this: noise reduces the interpixel correlation of an image, and extra bits are spent for coding noise which is in fact not a required information. Noise decreases the compression ratio. This is because, noise reduces interpixel correlation.

5.3 Fractal Image Compression of Noisy Images corrupted by Film-Grain Noise:

For demonstrating the effect of noise on Fractal Image Compression, images corrupted by data dependent film-grain noise are used. Film-grain type of noise is data

dependent and it occurs when scanning an image recorded on photographic films. In the exposure and development of a photographic film, silver halide grains that are exposed to a sufficient quantity of light are converted to grains of metallic silver. This process is not entirely deterministic, silver halide grains experiencing equivalent exposures are not necessarily converted to the same size and shape silver grains. Furthermore, silver grains are randomly distributed over the surface of the film. This inherent randomness in silver grain formation is called film-grain noise. This leads to a randomness or uncertainty in the amount of light passing or reflected from a print [85] .

An observed image corrupted by film-grain noise can be modeled as

$$g(x,y) = f(x,y) + f(x,y)\, n(x,y) \qquad (5.1)$$

where n(x,y) is a random zero mean data independent Gaussian noise with

$$E\{n(x_1,y_1)n(x_2,y_2)\} = \sigma_n^2(x_1-x_2,y_1-y_2), \qquad \forall (x_1,y_1),(x_2,y_2)$$

f(x,y) is the original image, and is a constant between 1/3 and 1/2.

To study the effect of noise on Fractal Image Compression, images corrupted by film-grain noise are used. Fisher's quadtree algorithm [Ch.3, 43] , Partial distance based Fractal Image Compression [90] , Section 3.10, and local Fractal dimension based classification in Fractal Image Compression algorithm given in Section 3.11 are used to study the effects of noisy images.

5.3 Results for Fractal Image Compression of film-grain noise corrupted images

The standard 8-bit images of size 256x256 ,and 512x512 corrupted by film-grain noise are used for evaluating the performance of Fractal Image Compression algorithms, viz, Fisher's quadtree algorithm [Ch.3, 43] , Partial distance based Fractal Image Compression [90] , and local Fractal dimension based classification in Fractal Image Compression algorithm(given in Section 3.10) at various bit rates. Noisy

images are obtained by using Eqn.(5.1) with σ_n^2 = {2.0, 4.0, 16.0} and = 0.5. σ_n^2 and are assumed to be known. Results are given for images lenna of size 256x256 and collie of size 256x256. The results are compared with original images. Results are tabulated in tables 5.1 –5.2 for both original and noisy image cases. Compression Ratio Versus PSNR is plotted for lenna and collie images and is shown in Figure 5.1 –5.6. It is observed that at low compression ratio, reduction in PSNR for compressed noisy image is more. It is less at high compression ratio. PSNR for noisy image when computed with respect to original image is close to the value computed for method which uses original image as input. Also it can be seen that at same PSNR ,compression ratio decreases for noisy images. Changing noise level doesn't change the general characteristics of the curves. The noisy images at various maximum levels and the decoded images are shown in Figures 5.6 – 5.8. The quality of the images can be improved by pre-filtering the input noisy image.

Compr. Ratio	PSNR (in dB)			
	Original	Noisy Image σ_n^2=2	Noisy Image σ_n^2=4	Noisy Image σ_n^2=16
4	31	29.25	28	24
10	29.25	27.5	26.25	22.1
20	26.2	25	24.1	20.75
30	24.9	23.9	23	20

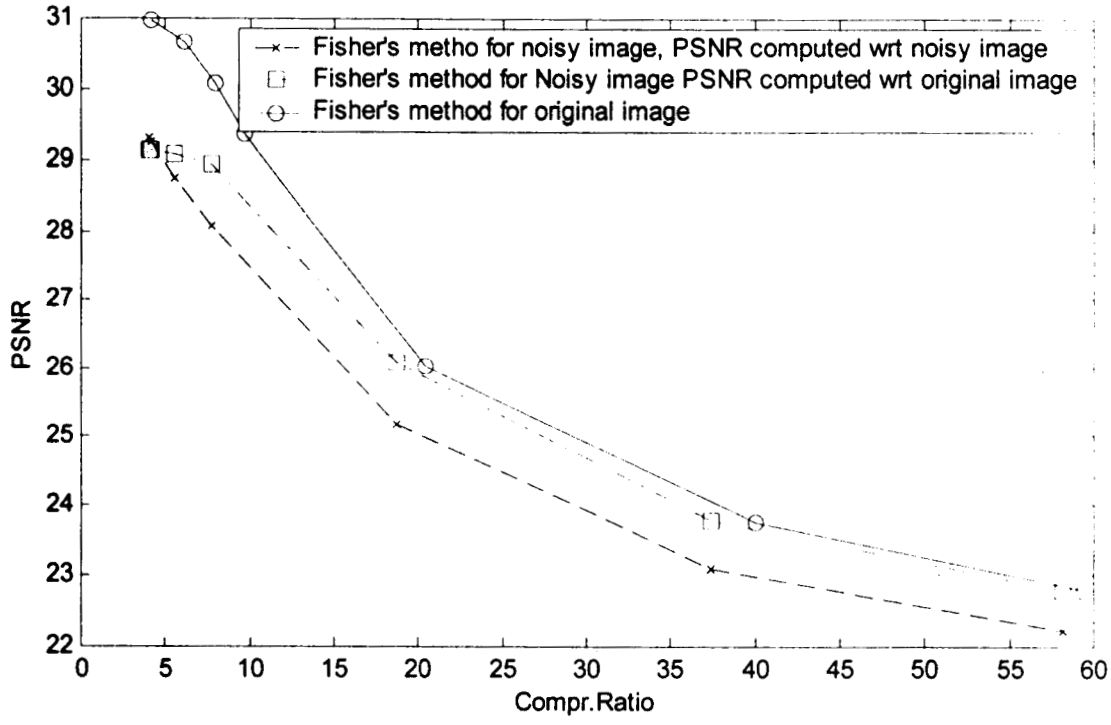
Table 5.1 Results for Noisy lenna image of size 256x256 compressed with Fisher's Quadtree method

Compr. Ratio	PSNR (in dB)			
	Original	Noisy Image σ_n^2=2	Noisy Image σ_n^2=4	Noisy Image σ_n^2=16
4	28.1	26.75	25.8	22.25
10	26.1	25.25	24.5	21.1
20	24.35	23.4	22.6	20
30	22.75	22.2	21.6	19.4

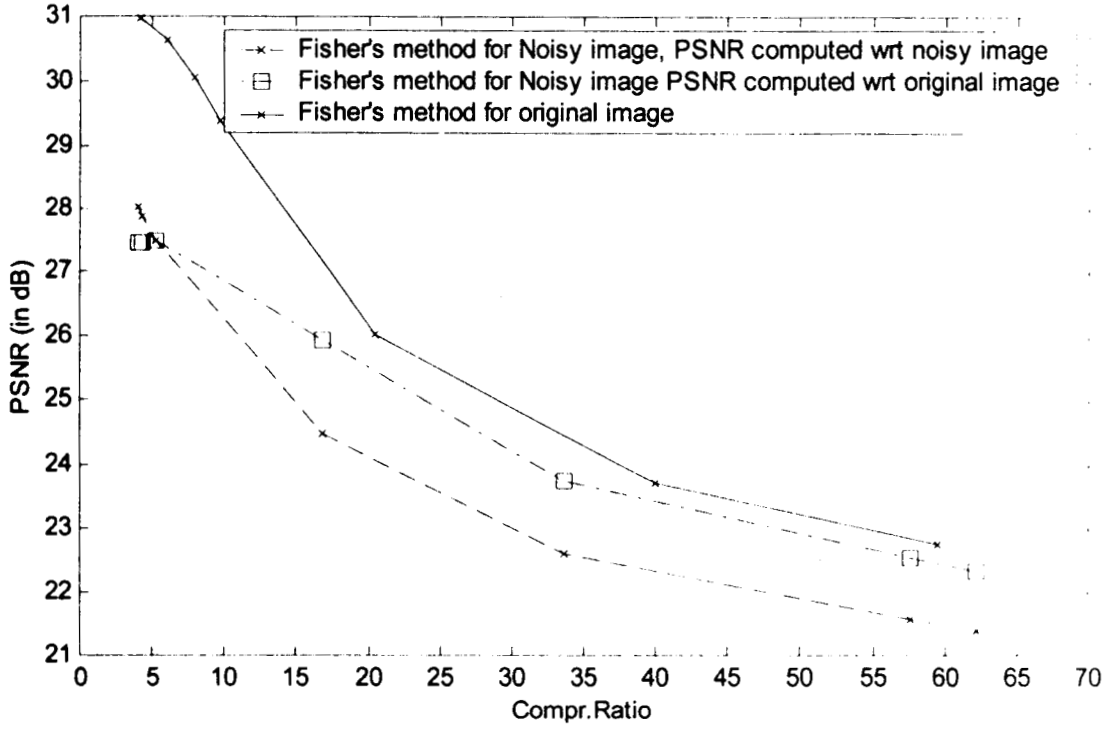
Table 5.2 Results for Noisy lenna image of size 256x256 compressed with LFD based FIC

Compr. Ratio	PSNR (in dB)			
	Original	Noisy Image σ_n^2=2	Noisy Image σ_n^2=4	Noisy Image σ_n^2=16
4	30.5	28.75	27.5	23.5
10	28.9	27	25.8	21.9
20	25.8	24.75	23.9	20.2
30	24.5	23.5	22.6	19.8

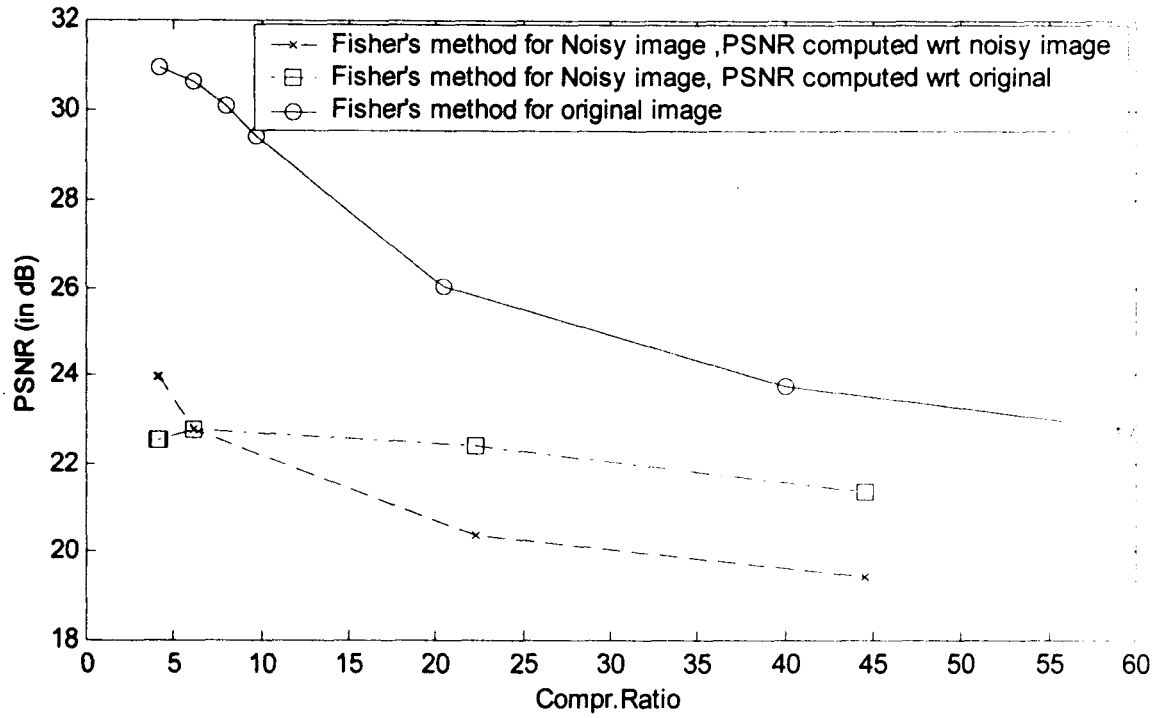
Table 5.3 Results for Noisy lenna image of size 256x256 compressed with Partial Distance FIC method



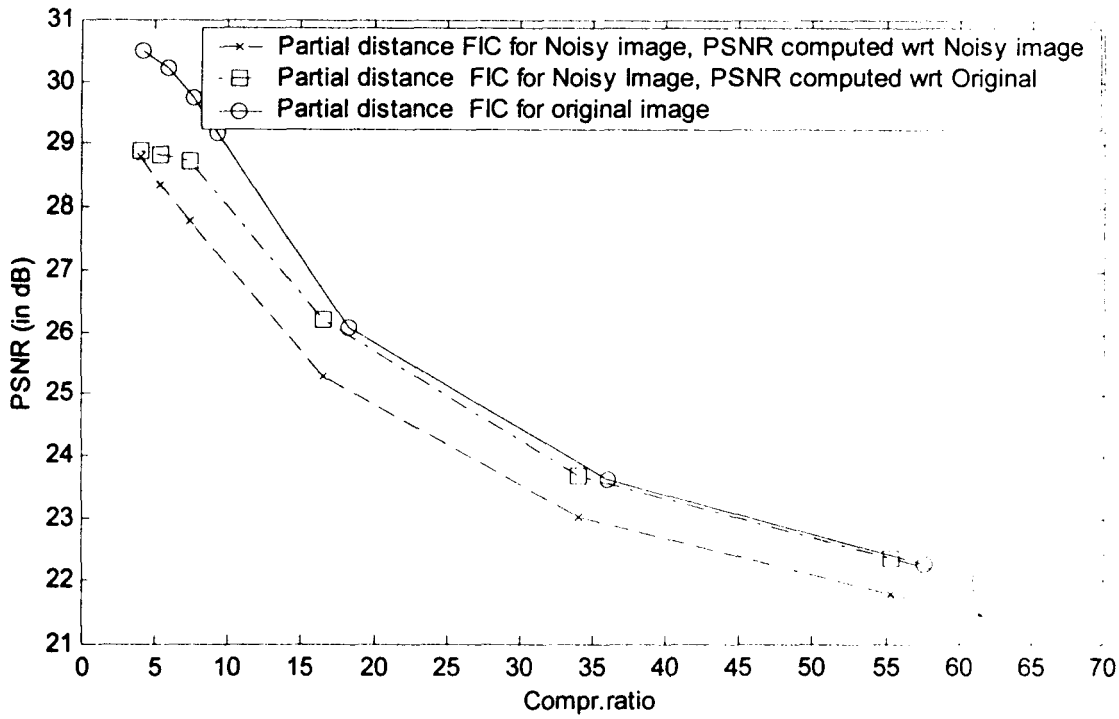
Figure(5.1) Performance graph for lenna image of size 256x256 corrupted by film-grain noise with $\sigma_n^2 = 2$ (Fisher's method)



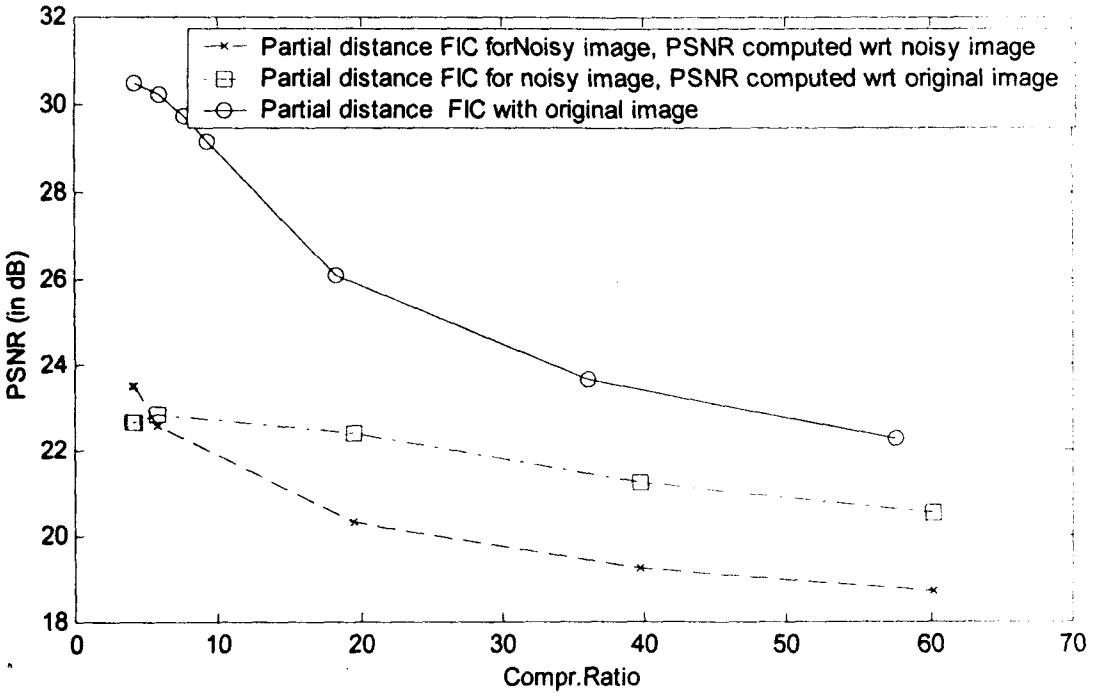
Figure(5.2) Performance graph for lenna image of size 256x256 corrupted by film-grain noise with $\sigma_n^2 = 4$ (Fisher's method)



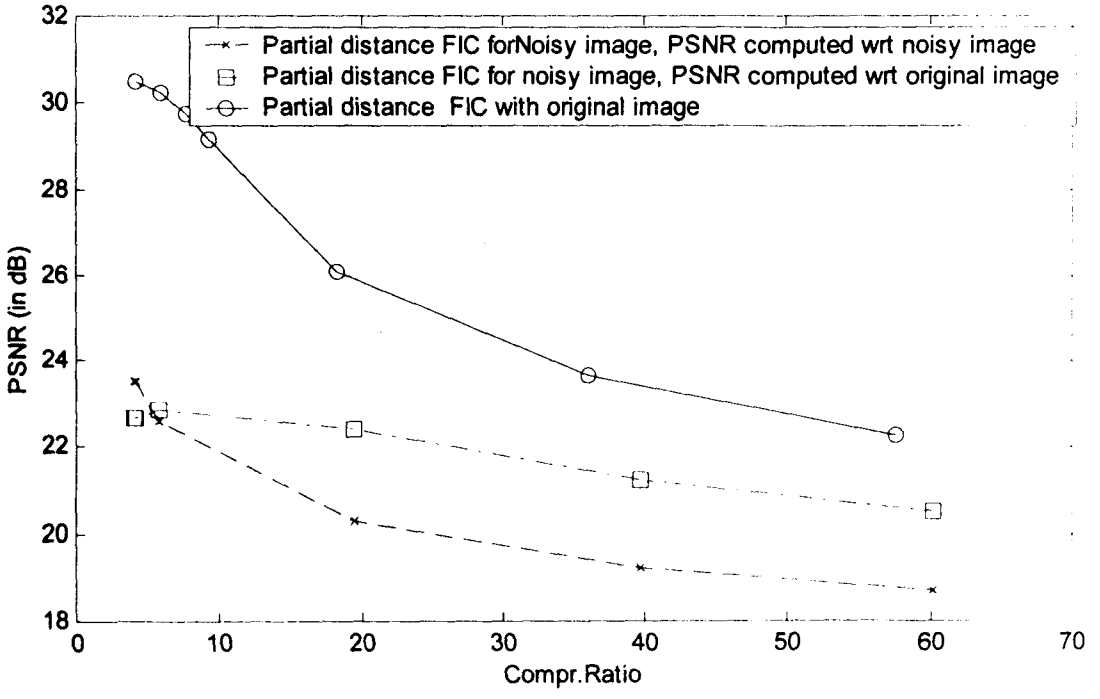
Figure(5.3) Performance graph for lenna image of size 256x256 corrupted by film-grain noise with $\sigma_n^2 = 16$ (Fisher's method)



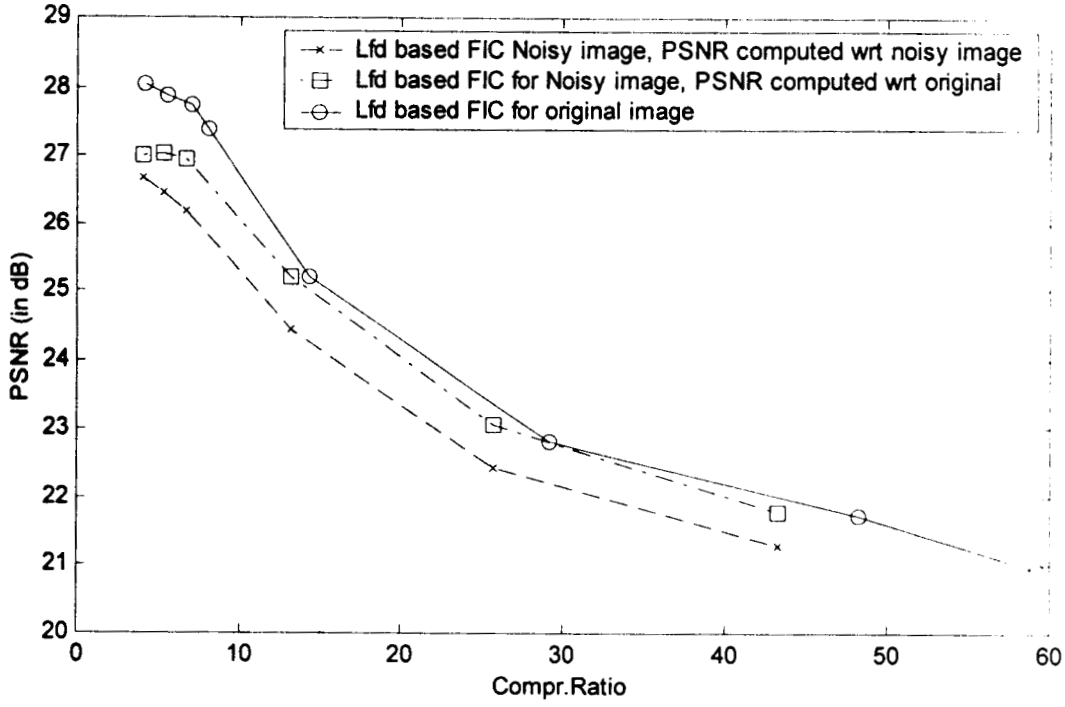
Figure(5.4) Performance graph for lenna image of size 256x256 corrupted by film-grain noise with $\sigma_n^2 = 2$ (Partial distance based FIC)



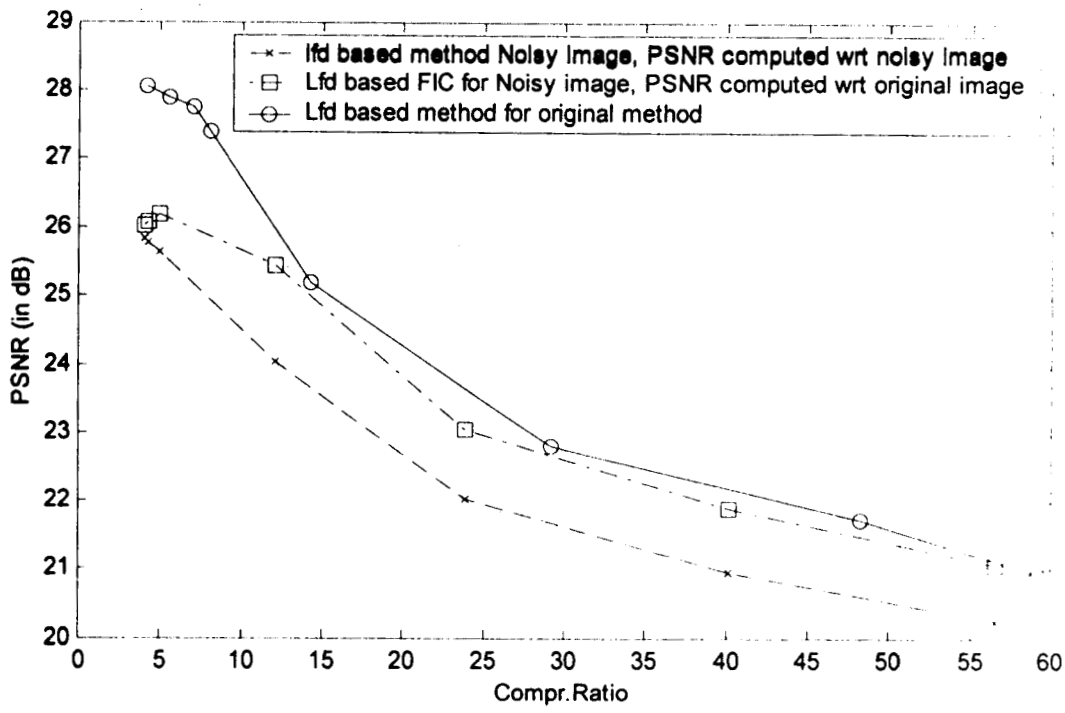
Figure(5.5) Performance graph for lenna image of size 256x256 corrupted by film-grain noise with $\sigma_n^2 = 4$ (Partial distance based FIC)



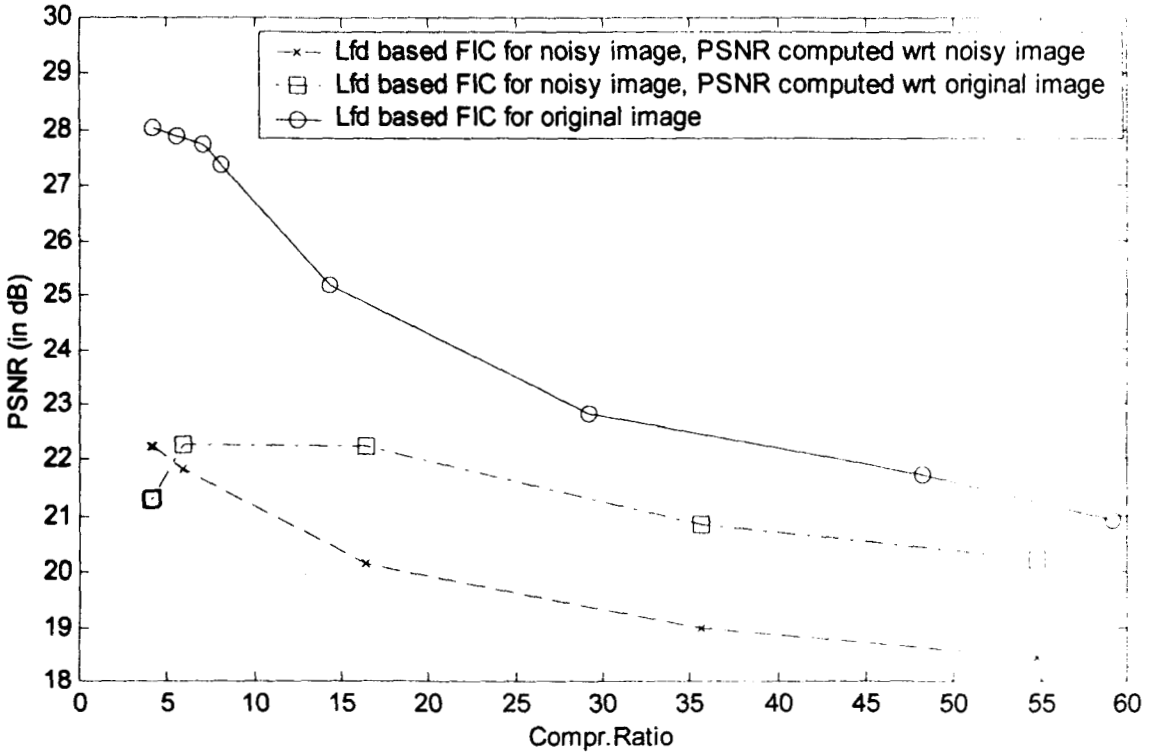
Figure(5.6) Performance graph for lenna image of size 256x256 corrupted by film-grain noise with $\sigma_n^2 = 16$ (Partial distance based FIC)



Figure(5.7) Performance graph for lenna image of size 256x256 corrupted by film-grain noise with $\sigma_n^2 = 2$ (Lfd based FIC)



Figure(5.8) Performance graph for lenna image of size 256x256 corrupted by film-grain noise with $\sigma_n^2 = 4$ (Lfd based FIC)



Figure(5.9) Performance graph for lenna image of size 256x256 corrupted by film-grain noise with σ_n^2 = 16 (Lfd based FIC)

(a) lenna image of size 256x256 corrupted by film-grain noise with σ_n^2 =16

(b) Compr.Ratio = 4.086 PSNR= 23.981 dB (computed wrt noisy image)

Figure (5.10) Lenna image corrupted by film-grain noise compressed using Fisher's quadtree method

Compr.Ratio = 4.086 PSNR= 23.981 dB (computed wrt noisy image)

Figure (5.11) Lenna image corrupted by film-grain noise compressed using Partial Distance FIC method

Compr.Ratio = 4.086 PSNR= 23.981 dB (computed wrt noisy image)

Figure (5.12) Lenna image corrupted by film-grain noise compressed using LFD Based FIC method

LOSSLESS IMAGE COMPRESSION USING WAVELETS

H.R. Mahadevaswamy “New approaches to image compression ” Thesis.
Department of Electronics Engineering, Regional Engineering College ,
University of Calicut, 2000

Chapter 6

LOSSLESS IMAGE COMPRESSION USING WAVELETS

6.1 Introduction

Wavelet transform is popular in Image Compression mainly because of its multiresolution and high energy compaction properties. Wavelet coding can be classified as sort of transform coding using the wavelet transform instead of FFT or DCT. A wide variety of wavelet transform based image coding schemes have been reported in the literature [19],[20],[18], [94]. Compression is accomplished by applying a wavelet transform to decorrelate the image data, quantizing the resulting transform coefficients and coding the quantized values. This method is called as Lossy coding. All the three steps of the above coding scheme may be performed in various ways. There are many different wavelet filters available. For example, orthogonal or biorthogonal filters [95], [17] can be used to carry out the transform part in image coding. Finding a optimal filter for a given image is difficult problem. Many papers have been published on the performance of different wavelet filters and the relevance of filter properties for image coding [16], [21], [22]. In the case of lossy wavelet coding the reconstructed image is approximation of the original.

In lossless image compression, the image can be reconstructed as in the original image. Medical images, satellite images and images scanned from manuscripts for preservation purposes typically demand lossless compression techniques. Some of the most effective methods for lossless compression are linear predictive coding [2], Differential Pulse Code Modulation (DPCM) or entropy methods like the Huffman coding[4], the arithmetic coding [11], universal codes of Elias [6], runlength coding [4], lossless

predictive coding [2] and the Lempet-ziv [9], JPEG lossless image standard [76] (which utilizes prediction as well as entropy based coding schemes). However, these methods produce a single resolution after transform, which retard the progressive transmission or recovery. The multiresolution nature of the wavelet transform is also ideal for progressive transmission. Lossless schemes generally yield compression ratios of the order 2:1 to 5:1, depending on the image data.

Theoretically, the wavelet transformation part in lossy coding method is considered lossless since the transformation is reversible mathematically. However, most transformations are lossy in practice, because all computers have only finite precision, even if floating-point calculations are used. This fact is a limitation for lossless image compression using wavelet transform based algorithms. Very few filters can be "integerized" (rounding is one way of converting the floating points to integers) to make the transform, a reversible one.

If a filter bank with floating-point coefficients is used, the subband samples will be floating-point numbers. Therefore, a mapping or a quantizer is needed to represent the samples by integers. The integer subband samples and the floating-point error values must be coded. Another possibility is to use a filter bank having integer filter coefficients which can map integer pixel values to integer subband samples. Several wavelet based integer coefficient filter banks, called binary filters can be found in [95]. Calderbank et al. [13] and Hong yang Chao et al. [14] have shown that by using lifting steps, every wavelet transform, or some subclass of general filter banks can be used to map integer input data to integer subband samples. Integer wavelet transform can be carried out with shift -and -add

operations and is fast. Moreover, the integer wavelet transform with lifting structure is suitable for both lossless and lossy image compression.

In this Chapter, several integer wavelet transforms with perfect reconstruction property are studied. In order to avoid large dynamic ranges in the subband domain, subband samples are truncated in such a way that there exists a perfect reconstruction system. Due to the non-adaptivity of the classical wavelet coding techniques the results heavily depend on the images considered. In order to overcome this difficulty adaptive wavelet coding method is developed. In this Chapter an approach to adaptive integer wavelet coding is proposed..

6.2 Wavelet Transforms

Wavelet transform provides the time-frequency representation and can be used to analyse non-stationary signals.

6.2.1 What are Wavelets ?

Wavelets are filters which form a class of general filters but having some specific properties. A unique definition does not exist, but Sweldens[96] in order to justify calling ψ a wavelet basis, the following properties would be expected:

(i) Wavelets are building blocks for general functions : In order to express any function in terms of wavelets a function f must be expressible by an infinite series of wavelets such that

$$f = \sum_{\lambda} \phi_{\lambda} \psi_{\lambda} \tag{6.1}$$

where $\psi_{\lambda} \rightarrow$ basis function

$\phi_i \rightarrow$ scaling function

(ii) **Wavelets have space-frequency localisation:** Wavelet coefficient is localised in the frequency and real space domains. Having a multiresolution representation, means that wavelet coefficients may overlap one another in spatial terms, but represent different spatial frequencies within that area. Conversely, several wavelet coefficients may represent the spatial frequency, however they cover different parts of the picture. These properties allow wavelet analysis to extract and represent picture patterns at different resolutions.

(iii) **Wavelets can be implemented using fast transform algorithms:** The wavelet transforms are evaluated in linear or log linear time in order to be usable for computer applications. This is possible using multiresolution analysis and the wavelet representation is ideally implemented in such a framework.

6.2.2 Wavelet Transform :

The discrete wavelet transform (DWT) applies different sized windows enabling high and low resolutions of the original signal to make up its representation in the wavelet domain (Figure 6.1). Extending to the two dimensional case of images, large areas of the image are covered by low frequency coefficients. The adaptability of wavelet to the natural detail of the image is very superior to the use of single sized window and enables the DWT to perform better at high compression ratios. The wavelet approach allows the use of different wavelet filters which have different computational complexities and different properties.

The mother wavelet ψ has the basic shape from which a scaling function ϕ and wavelet function ψ is created whose popular name are 'average' and 'detail' functions, respectively. For pictures these functions are superimposed on the two dimensional data matrix f to create 'average'(low-pass) and 'difference' (high-pass) data in the vertical(LH), horizontal (HL), and diagonal (HH) directions [17]. (See Figure 6.2)

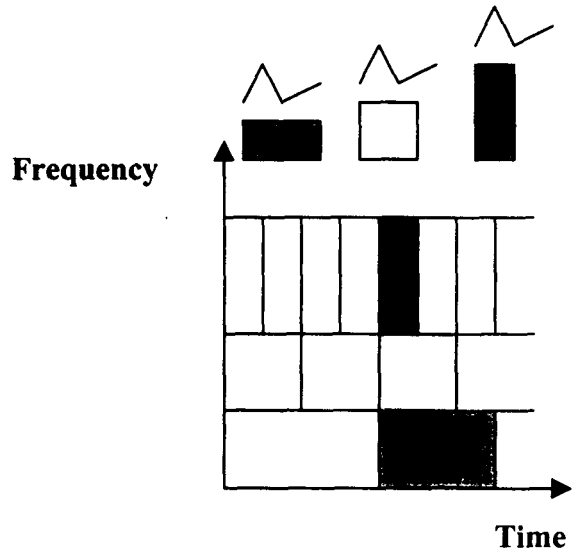


Figure (6.1) Wavelet Transform domain (one dimension)

$$LL(x,y) = \varphi(x)\ \varphi\ (y)\ f(x,y) \tag{6.2}$$

$$LH(x,y) = \varphi\ (x)\ \psi(y)\ f(x,y) \tag{6.3}$$

$$HL(x,y) = \psi(x)\ \varphi(y)\ f(x,y) \tag{6.4}$$

$$HH(x,y) = \psi(x)\ \psi f(x,y) \tag{6.5}$$

The frequency resolution of the input image is divided into two halves for every iteration of the wavelet transform making it an octave decomposition method [21]. The LL-band in Figure(6.2) is the input to the next iteration in a multiresolution analysis.

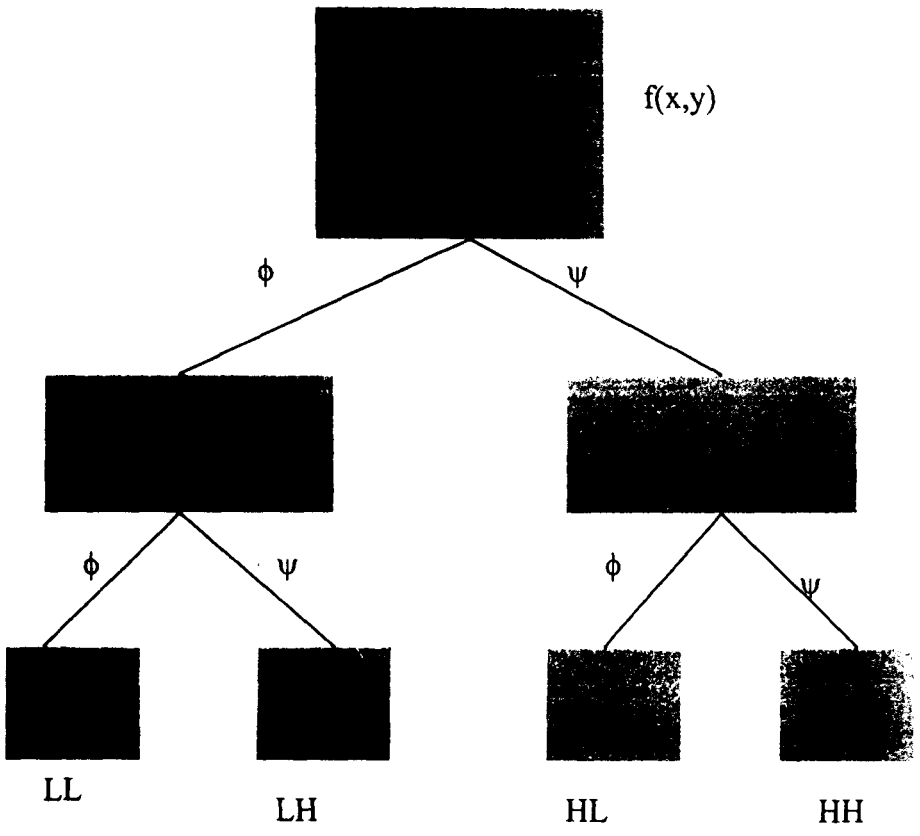


Figure (6.2) Two dimensional wavelet decomposition

6.2.3 How do we use Wavelets to extract information ?

Pictures or images are non-stationary in frequency and spatial content; hence, data representing picture content may be any where in the actual picture. Also they are spread across the spatial frequency domain with no fixed distribution. Significant visual information in pictures are sharp transitions such as edges. But features or changes in the background of a picture may also be of importance. Wavelets are suited for extracting this kind of information spread across space because of their localisation properties. If wavelets are used in a multiresolution analysis (Figure(6.2)), the representation of the

picture at different resolutions enable determination of spatial frequency as well as spatial localisation in the wavelet domain.

All the wavelets are not equally suited to perform the extraction of information decsribed above in an efficient way. As mentioned above, no formal definition exists for a wavelet and hence any wave pattern can be said to be a wavelet. However, for a wavelet to be usable in practice the wave pattern needs to possess one or more of the following properties:

(a) Orthogonality : The original signal is split into a low and a high frequency part, and filters enabling the splitting without duplicating information are said to be orthogonal. If two filters H and G are to be orthogonal the following must hold:

$$\sum_{i=1}^{N} h_i g_i = 0 \tag{6.6}$$

where N represents the length of the filters, and h_i and g_i are the filter coefficients for low (average) and high (detail) coefficients respectively. This can only happen if

$$g_i = (-1)^{i-1} h_{i-1} \tag{6.7}$$

(b) Linear Phase : To obtain linear phase, symmetric filters would have to be used. However, if a symmetric filter is used, (6.6) can never be zero and orthogonality condition cannot be fulfilled.

(c) Compact support : The magnitude response of the filter should be exactly zero outside the frequency range covered by the transform. If this property is satisfied, the transform is energy invariant, since

$$\sum_{\bullet}(< inputcoefficient >)^2 = \sum_{\bullet}(< transformcoefficient >)^2 \tag{6.8}$$

This property states that the transform is a orthonormal, and an associated advantage of this is that the error of a representation may be estimated directly in the transformed

domain, rather than performing an inverse transformation in order to estimate the distortion.

(d) Perfect reconstruction : If the input signal is transformed and inversely transformed using a set of weighted basis functions, and the reproduced sample values are identical to those of the input signal, the transform is said to have the perfect reconstruction property. If, in addition no information redundancy is present in the sampled signal, the wavelet transform is, as stated above, othronormal.

No wavelets can possess all these properties, so the choice of the wavelet is decided based on the consideration of which of the above points are important for a particular application. Haar-wavelet, Daubechies-wavelets and biorthogonal-wavelets are popular choices. These wavelets have properties which cover the requirements for a range of applications.

6.3 Lossless Image Compression Using Integer Wavelets :

In this section, some of the earlier works on integer reversible wavelet transform for lossless compression methods are described.

6.3.1 S+P Transform proposed by Said and Pearman

The oldest integer to integer transform is sequential transform (S-transform), which is the integer version of Haar transform. S+P (**S** Transform + **P**-Prediction) transform proposed by Said and Pearman [15] a multiresolution transform suited for lossless and lossy image compression. This is similar to subband decomposition, but can be computed with only integer addition and bit-shift operations. In this transform, linear prediction is performed on the lowpass coefficients a[n] to generate a new set of highpass coefficients after an S-transform. The general form of the transform is

$$d'[n] = x[2n+1] - x[2n] \tag{6.9}$$

$$a[n] = x[2n] - \lfloor d'^{(n)}/2 \rfloor \tag{6.10}$$

$$d[n] = d'[n] + \lfloor \alpha_{-1}(a[n-2] - a[n-1]) + \alpha_0(a[n-1] - a[n]) + \alpha_1(a[n] - a[n+1] - \beta_1 d'[n+1] \rfloor \tag{6.11}$$

Said and Pearman selected the predictors with coefficients

(a) $\alpha_{-1} = 0, \alpha_0 = 1/4, \alpha_1 = 1/4, \beta_1 = 0$ (b) $\alpha_{-1} = 0, \alpha_0 = 2/8, \alpha_1 = 3/8, \beta_1 = 2/8$

(c) $\alpha_{-1} = -1/16, \alpha_0 = 4/16, \alpha_1 = 8/16, \beta_1 = 6/16$. Predictor (a) has the smallest computational complexity, (b) is suitable for natural images and (c) for smooth medical images. S+P transforms gives smallest values of first order entropy compared to S-transform.

6.3.2 Algorithms proposed by Calderbank et al.

Calderbank et al. [14] proposed a method to synthesize a class of 2-channel linear filter banks and theory for implementing filter banks in non-linear fashion. The construction of integer to integer wavelet transform in this method is based upon the idea of factoring wavelet transform into lifting steps [77]. The lifting scheme is a flexible technique that has been applied to the construction of wavelets through an iterative process of updating a subband from an appropriate linear combination of the other subband. The first set of invertible integer wavelet transforms have names of the form (N,$\tilde{N}$), where N is the number of vanishing moments of the analysing highpass filter, while $\tilde{N}$ is the number of vanishing moments of the synthesising highpass filter. They are the instances of a family of symmetric, biorthogonal wavelet transforms built from the interpolating Deslauriers-Dubuc scaling functions [77]. The next transforms are built by using lifting

factorisation to the transforms S+P, D4,(9-7) [95]. It was observed that there is no filter that consistently performs better than all the other filters on all the test images.

6.3.3 Algorithm proposed by Hongyang Chao et al.

In this technique, a general way is given to create integer wavelet transformations that can be used in lossless (reversible) compression of images of arbitrary size [14]. This method is based on updating techniques such as lifting and correction and need only be calculated with integer addition and bit shift operations. In addition, the integer wavelet transforms created here possesses the property of precision preservation (PPP) and this is very useful in lossless compression, for conserving memory in both compression and decompression and for speeding up the computation process[14].

Another interesting property in these two transforms is that if the values of the signal or image pixels are represented by a finite number of bits, say one bit or one byte, we can still use the same number of bits to represent the result of the forward transform within the computer itself. This is because of the complementary code and modulo (a fixed number) arithmetic features of computers. In the reconstruction algorithm the computer will get back the exact original signal through the same features. This is called a property of Precision Preservation (PPP) for the above wavelets.

The lifting scheme [77] is a new approach for constructing biorthogonal wavelets with compact support and can be used with minor modification to create integer wavelets which are similar to biorthogonal wavelets with compact support. Here 'similar' means the decomposition results from the integer wavelets are the same as those of the corresponding biorthogonal wavelet, except for rounding error and some scaling factors.

By using another approach called correction method integer wavelets can be obtained and even some complicated filters with fast decomposition and reconstruction

abilities can be obtained. In previous approach, after the decomposition, one may not be satisfied with the result, because there may still be some correlation among the high pass components. This may be due to aliasing from the low pass components, or due to the fact that the low pass components do not carry enough of the expected information from the original signal. Hence, an improvement could be achieved by putting some correction part on the high pass components or low pass components.

6.3.4 Integer Coefficient Filter Banks For Lossless Image Compression:

Most images have integer values once they have undergone sampling both in time and magnitude (quantization). Since images have integer pixel values, it becomes natural to investigate the filter banks having integer filter values. Since the motivation is to keep the bit rate low, correct optimization criteria are needed and such algorithms should maximize the coding performance (low entropy), while restricting the filter coefficients to have only integer values [16]. In this approach by Balasingham [16], to find the integer filter coefficients in the filter banks, a discrete optimization algorithm is adapted. Here ICFB's are optimized for subband coding gain. The coding algorithm used here is based on class-wise arithmetic coding of the subband samples. While implementing ICFB's, the dynamic range of subband samples becomes very large and this will lead to higher entropy and thus higher bit-rate. Hence, it is very important to have proper scaling at the analysis ICFB which leads to an efficient implementation having an exact inverse at the synthesis ICFB.

As the images have different information content and different space and frequency characteristics, a single ICFB is not necessarily optimized for all cases. The compression ratios obtained are better compared to results based on other methods.

6.4 Lossless Image Compression using Adaptive Integer Wavelet Coding :

Review of literature on Integer wavelet transforms for lossless image compression [12] reveals the following:

i) Wavelet transforms provide multiresolution analysis.

ii) Wavelet transforms allow the reconstruction of lower resolution details first, followed by successive details. Such a mode of transmission is especially valuable in situations where bandwidth is limited, image sizes are large, and lossy compression is not desirable.

iii) There is no single integer wavelet filter that performs well on all types of images.

Due to the non-adaptivity of the integer wavelet transform, the results heavily depend on the images considered. In order to overcome this difficulty adaptive integer wavelet coding method is proposed.

6.4.1 Varying filterbanks :

In the conventional methods of lossless image compression using integer wavelets transforms, one uses a set of well chosen filter coefficients to perform a decomposition from fine to coarse scales. Commonly these coefficients do not change from one scale level to the next one. Since all the transformations at each level are performed independently, it is possible to use different filtercoefficients (which correspond to different scaling and wavelet functions) at every scale. But, the question arises, how to choose the best suited filters for different scales and directions? In this section , an algorithm is introduced for choosing the optimal filters.

6.4.2 Best filter selection :

The wavelet decomposition is obtained by using different filter pairs for different scale levels of the decomposition. For example, in Figure(6.3), filter pair G,H is used at scale level j+1 , filter pair A, B at scale level j, etc.

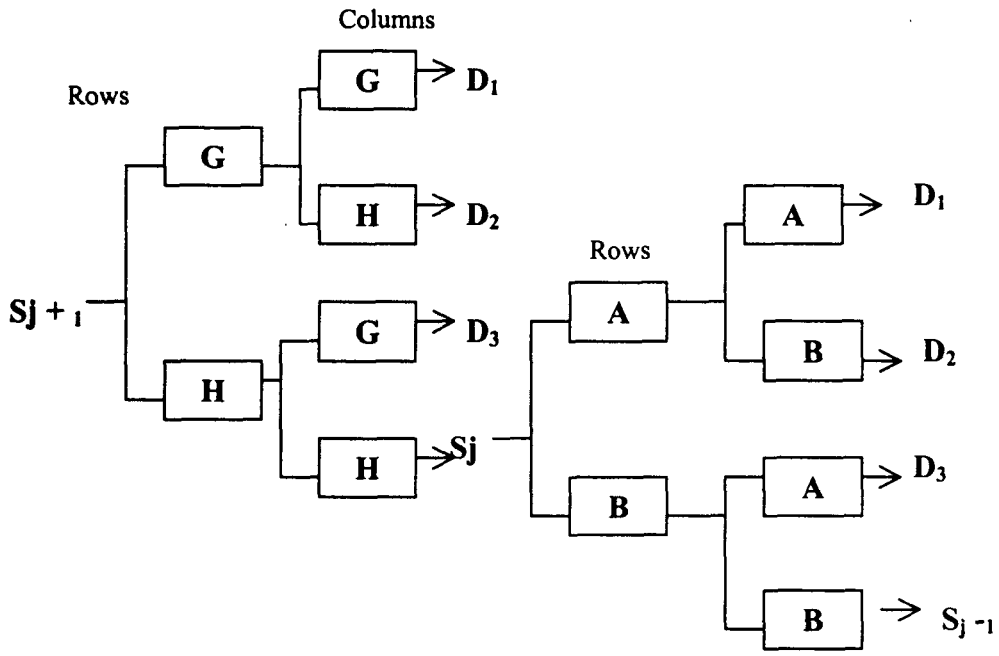


Figure (6.3) 2-D wavelet decomposition with varying filter pair

Suppose the filter library contains l pair of different integer wavelet filters and decomposition depth is fixed to be at m, then, there will be l^m different wavelet decompositions. The algorithm that selects good filter combinations from the given library is described below.

Different scale levels and the co-ordinate axes are considered independent for the choice of an optimal filter. First search for the optimal filter pair for the detail images of the first scale level. This is fixed, then search is continued for the optimal filter for the

detail images of the second scale level and so on. This reduces the number of filter combinations to be examined from l^m to ml. This is because, the combination of filters selected should be suited for compression purposes. The algorithm finds optimal combination. The optimal filters for the detail images are chosen by minimising certain cost functions on the detail images for each scale and co-ordinate axes separately. The classical additive information cost function is used. For a given library, a very efficient representation of an image is selected by minimising the information content of the detail images. An additive analogue to Shannon-Weaver entropy is used. It is given by

$$\lambda(x) = -\sum_j x_j^2 \log(x_j^2) \tag{6.12}$$

where $X=(x_0,x_1,x_2, \text{---})$ is the normalised sequence of the detail coefficients.

Algorithm :

First scale level

- Using filter pair in the library apply integer wavelet transform to the rows of the original image
- Using the filter pair in the library apply integer wavelet transform to the columns of the transformed image (rows)
- Calculate and store the coefficients of the resulting approximate and detail images for each filter pair
- Calculate and store the cost function of the resulting detail images for each filter pair
- Determine the filter pair producing the lowest cost function
- Store the resulting lowpass and three detail images.

- Proceed to next scale level

Second (and all subsequent) scale level(s):

- Using the filter pair in the library apply integer wavelet transform to the rows of the lowpass image of the previous scale
- Using the filter pair in the library apply integer wavelet transform to the columns of the transformed image (rows)
- Calculate and store the coefficients of the resulting approximate and detail images for each filter pair
- Calculate and store the cost function of the resulting detail images for each filter pair
- Determine the filter pair producing the lowest cost function
- Store the resulting lowpass and three detail images.
- Proceed to next scale level
- Code the filter choice

Reconstruction of the image is done by using appropriate filter pair at different scale.

6.4.3 Filter Library :

Filter library consists of integer wavelet transform given in [13], [14], [95]. Some of the integer wavelet transforms are given below.

(1) S or (1,1) transform :

It is the oldest integer to integer wavelet transform. It is an integer version of the Haar transform. It is also referred to as (1,1) transform.

$$d[n] = s_0[2n+1] - s_0[2n]$$

$$s_1[n] = s_0[n] - \lfloor d[n]/2 \rfloor$$

The operator $\lfloor . \rfloor$ truncates or rounds the values to the nearest integer. $s_0[], s_1[]$ and d[ ] correspond to image data, approximation(lowpass) and detail(highpass) values.

(2) (2,2) transform

$$d[n] = s_0[2n+1] - \lfloor 1/2(s_0[2n] + s_0[2n+2]) + 1/2 \rfloor$$

$$s_1[n] = s_0[2n] + \lfloor 1/4(d[n-1] + d[n]) + 1/2 \rfloor$$

(3) (4,2) Transform

$$d[n] = s_0[2n+1] - \lfloor 9/16(s_0[2n] \_ s_0[2n+2]) - 1/16(s_0[2n-2] + s_0[2n+4]) + 1/2 \rfloor$$

$$a[n] = s_0[2n] + \lfloor 1/4(d[n-1] + d[n]) + 1/2 \rfloor$$

(4) (4,4) Transform

$$d[n] = s_0[2n+1] - \lfloor 9/16(s_0[2n] + s_0[2n+2]) - 1/16(s_0[2n-2] + s_0[2n+4]) + 1/2 \rfloor$$

$$a[n] = s_0[2n] + \lfloor 9/32(d[n-1] + d[n]) - 1/32(d[n-2] + d[n+1]) + 1/2 \rfloor$$

(5) (2,2+2) Transform

One extra lifting step is used to build (2,2) transform into a transform with four vanishing moments of highpass analysing filter. This is different from (4,2) transform and is therefore called as (2,2+2) transform. First compute (2,2) yielding lowass samples a[n] and preliminary detail or highpass samples $d^{(1)}[n]$ and then use the a[n] combined with

d[n+m} (m>0) to compute $d'[n]$ as a prediction for $d^{(1)}[n]$. The final detail is $d^{(1)}[n]-d'[n]$.

$$d^{(1)}[n]=s_0[2n+1]-\lfloor 1/2(s_0[2n]+s_0[2n+2])+1/2\rfloor$$

$$a[n]=s_0[2n]+\lfloor 1/4(d^{(1)}[n-1]+d^{(1)}[n])+1/2\rfloor$$

$$d[n]=d^{(1)}[n]-\lfloor \alpha(-1/2a[n-1]+a[n]-1/2a[n+1])+\beta(-1/2a[n]+a[n+1]-1/2a[n+2])+\gamma d^{(1)}[n+1]+1/2\rfloor$$

Special cases are :

(1) $\alpha=1/6,\beta=0,\gamma=1/3$ (2) $\alpha=1/8,\beta=1/8,\gamma=0$ (3) $\alpha=1/4,\beta=-1/4,\gamma=1$

6.5 Implementation and Results :

The performance of various integer wavelet transforms are evaluated for lossless compression of standard digital images. All the images used have 256 gray levels (i.e., 8 bits per pixel) and size of 256x256. In the evaluation each image is decomposed into a maximum of six scale levels, depending on the length and width of the given image. The decomposition is terminated if the number of decomposition levels exceeds six or if the horizontal or vertical size of the lowpass-lowpass image becomes less than 4.

The effectiveness for lossless image compression is measured using the entropy H(X), given by

$$H(X)=-\sum_i P(s_i)\log(P(s_i)) \tag{6.13}$$

where X is discrete random variable taking on values s_i with probability $p(s_i)$.

For an image X(,) of size NxM, $p(s_i)$ is taken as normalised count of value s_i, i.e.,

$$p(s_i)=\left|\frac{(l,m):X(l,m)=s_i}{NM}\right| \tag{6.14}$$

The statistics in different quadrants of a wavelet transformed image are different and hence, weighted mean of the entropies in each quadrant of the transformed image is

computed. Suppose $(LL)^l$ represents the lowest frequency image after l level transformations, $(LH)^j$, $(HL)^j$, and $(HH)^j$ (j=1...l) represent the high frequency part (detail images) related to horizontal, vertical and diagonal details in the j^{th} level. The entropy is computed as

$$\frac{1}{4^l}H((LL)^l)+\sum_{j=1}^{l}\frac{1}{4^j}(H(LH)^j)+H((HL)^j)+H((HH)^j)) \qquad (6.15)$$

Table 6.1 shows entropy comparisons between various integer wavelet transform methods and adaptive integer wavelet coding method. Results for non-standard image administrative building of CREC is given in Appendix D. It is observed that no filter performs better consistently for all images. Where as proposed adaptive method perform better for all images. The proposed method of course takes more time compared to the method where in single filter pair is used for all scales. But decoding is as fast as conventional one. Compromising with time, adaptivity can be achieved.

Wavelet transform	Lena	Sanfransisco	collie	camera	bridge
S or(1,1)	4.45678	4.14495	4.11144	4.00324	4.7197
(2,2)	4.94142	4.60732	4.544125	4.6092	5.03222
(4,2)	5.26787	5.24412	5.26737	4.90523	5.27189
(4,4)	5.17532	5.01368	5.17206	4.84498	5.18174
(2,2+2)	5.21985	5.21269	5.22485	4.96109	5.2435
Adaptive filter	3.88993	3.55087	3.24195	3.45588	3.45588

Table 6.1 Entropy for various images

CONCLUSIONS AND FUTURE WORK

H.R. Mahadevaswamy “New approaches to image compression ” Thesis.
Department of Electronics Engineering, Regional Engineering College ,
University of Calicut, 2000

Chapter 7

CONCLUSIONS AND FUTURE WORK

This thesis work has focused on development of computationally efficient and effective algorithms for still image compression. Fast and efficient lossy coding algorithms using fractal image compression and lossless coding using integer wavelet are developed.

Image compression is the key technology for managing images is digital format. Fractal and wavelet transforms based image compression have been an attractive area of research from the last decade. Fractal image compression has attracted much attention mainly because of the fact that it is "resolution independent" and provides good visual quality at high compression ratio. But, its encoding stage is computationally expensive.

In this thesis, fast methods are developed to accelerate the encoding stage in Fractal image compression.

Partial distance fractal image compression method is developed to accelerate the encoding step in fractal image compression. Encoding step in fractal image compression is slow because, most of the time is spent in searching and discarding of unrelated domain blocks. Again, during this process, significant fraction of computational lost lies in the actual calculation of distances between domain and range blocks. The time required for the search is reduced by improving the efficiency of these calculations. Partial distance method is developed to achieve reduction in encoding time. With this method, while finding the best domain block for each range block, instead of calculating complete distance, only partial distance is found. If this partial distance is more than the previous value of distance, that domain is rejected before finding the complete distance. This

reduces lot of computations and hence searching time reduces. This method was tested with various images with different conditions like no symmetries, only for positive scaling and full search. Results shows that there is reduction in encoding time with little degradation in image quality compared to existing methods. While comparing the developed method with other methods, compression ratio is kept constant.

Another way to reduce the encoding time in fractal image compression is to classify the domain and range blocks based on some features. The classification of domain and range blocks based on fractal dimension feature is developed. In this technique, the domain and ranges in fractal image compression are classified into fixed number of classes based on fractal dimension of each block. Blanket method was used to find fractal dimension of the blocks. Here, while finding the fractal code for each range block, it is searched for best domain only with the class in which that range belongs. Searching time is reduced and hence encoding time in fractal image compression reduces with little degradation in image quality.

Above methods can also be used with other speed up methods to get even better gain in the reduction of encoding time.

Effect of noise on fractal image compression algorithms is studied in Chapter 5. Noisy images corrupted by data dependent film-grain noise are compressed using Partial distance based fractal image compression, fractal dimension based classification in fractal image compression and quadtree based fractal image compression algorithms. It is seen that compression ratio decreases at same PSNR for noisy images.

Generally fractal image compression is done using fixed block image partitioning methods, because it requires no overhead information for storing / transmission of corresponding image partition and easy for hardware realisation. But this neglects the image content and as a result it leads to severe blocking artefacts with increasing compression ratio. To overcome this, region-based or content based fractal image compression is developed. In this method, image is partitioned into non-overlapping regions of irregular size and shape. These regions are fractal encoded. In this method, given image is uniformly partitioned and fractal encoding is done. Then merging of blocks is carried out using fractal dimension feature. Again fractal code for merged region is found. The range partition is coded using chain coding method. Results shows that fractal dimension based region-based fractal image compression performs better compared to existing methods.

Table 7.1 shows comparison between the algorithms developed in this thesis for lenna image. Local fractal dimension based fractal image compression method takes less time to encode the given image compared to other methods. There is degradation in the quality compared to other tow methods. Depending on the requirement one can select the method to compress the given image. The parameters are compression ratio, quality and computational resources available.

Compr. Ratio	Partial Distance based FIC		Local fractal dimension based FIC		Fractal dimension based region based FIC	
	Enc.Time (in Secs)	PSNR (in dB)	Enc.Time (in Secs)	PSNR (in dB)	Enc.Time (in Secs)	PSNR (in dB)
4	500	30.5	244	28.2	-	-
8	300	29.5	180	27.4	200	34
12	255	28.2	160	26	250	31.5
20	190	25.8	140	24.3	360	29
30	150	24.4	120	22.3	450	27.4
40	120	23.4	116	22.3	800	24.8

Table 7.1 Comparison between various methods developed in this thesis for lenna image of size 256x256

Some of the applications like medical images and satellite images require lossless image compression techniques. Existing lossless compression techniques produces a single resolution after transform, which retards progressive transmission or recovery. Integer wavelet transform are used to overcome the above problem. Many integer wavelet based lossless compression methods are available, but they do not perform better on all images. The adaptive wavelet coding method is developed to overcome the above problem. Since all the transformations at each level are performed independently, it is

possible to use different filter pairs for different scale levels of the decomposition. In the method developed, optimal filter pair for the detail (high frequency) images are found by minimising the cost functions for each scale and co-ordinate axes separately. Shannon-Weaver entropy is used as cost function. The adaptive integer wavelet coding algorithm perform better compared to other integer wavelet coding methods for lossless image compression.

FUTURE WORK

Better visual modules and perception based error criteria are needed for image coding. Analysis of fractal schemes could be done in wavelet domain to study why fractal schemes work better. Using wavelet framework number of implementation issues such as bit-allocation methods, error estimation and search strategies can be studied.

To find local iterated function system in fractal encoding genetic algorithm method can be used. Genetic algorithm is found to be provide efficient searching for approximations to global optima in large and complex spaces in relatively short time.

Neural network technique can be used for feature extraction and domain classification in fractal image compression to speed up the encoding process.

Image denoising method using wavelet for noisy images could be developed. This yield better results in image compression techniques using fractal and wavelet techniques for noisy input images.

Better cost function than Shannon-Weaver entropy can be used in lossless image compression using adaptive integer wavelet method to select better filter pair from the library of filters.

REFERENCES

[1] Arun N. Netravali and Barry G. Haskell, ***"Digital Pictures - Representation, Compression, and Standards"***, Plenum Press, New York, 2nd Edition, 1994.

[2] Rafael C. Gonzalez and Richerd E. Woods, ***"Digital Image Processing"***, Addison-Wesley Publishing Company, New York, 1992.

[3] Edward R. Dougherty (Editor), ***"Digital Image Processing Methods"***, Marcel Dekker, Inc., New York, 1994.

[4] Anil K. Jain, ***"Fundamentals of Digital Image Processing"***, PHI, New Delhi, 1995.

[5] S. Golomb, ***"Run Length Encodings"***, IEEE Transactions on Information Theory,Volume IT-12, pp.399-401, 1966.

[6] P. Elias, ***"Universal Code Word Sets and Representations of the Integers"***, IEEE Transactions on Information Theory, Vol. IT-21, pp.194-203, 1975.

[7] A. Habibi, ***"Compression Of The Nth Order DPCM Encoder With Linear Transformations And Block Quantization Techniques"***, IEEE Transactions on Communication, Vol.COM-19, pp.948-956, 1971.

[8] S.K. Goyal and O' Neal, Jr., ***"Entropy Coded Differential Pulse Code Modulation For Television"***, IEEE Transactions on Communication, Vol.COM-23, pp.660-666, 1975.

[9] J.Ziv and A. Lempel, ***"A Universal Algorithm For Sequential Data Compression"***, IEEE Transactions on Information Theory, Vol.IT-23, pp.337-343, 1977.

[10] J.Ziv and A. Lempel, ***"Compression of Individual Sequences Via Variable Rate Coding"***, IEEE Transactions on Information Theory, Vol.IT-24, pp.530-536, 1978.

[11] Ian H. Witten, Radford M. Neal and John G. Cleary, ***"Arithmetic Coding For Data Compression"***, Communications of the ACM, Vol.30, pp.520-540, June 1987.

[12] H.R. Mahadevaswamy, P. Janardhanan and Y. Venkataramani, ***"Lossless Image Compression using Wavelets - A Comparative Study"***, Proceedings of National Communication Conference, NCC'99, IIT Khargpur, India, pp.345-352, Allied Publishers, NewDelhi, January 1999.

[13] A.R. Calderbank, Ingrid Dauchies, Wim Sweldens, and Boon lock Yeo, ***"Lossless Image Compression using Integer to Integer Wavelet Transforms"***, International conference on Image Processing, ICIP'97, Vol 1, No.385, pp.596-599, Oct 1997

[14] Hongyang Chao, Paul Fisher and Zeyi Hua, ***"An Approach To Integer Wavelet Transforms For Lossless Image Compression"***, http://www.mathsoft.com/wavelet.html.

[15] Amir Said, and William A. Pearman, ***"An Image Multiresolution Representation For Lossless And Lossy Compression"***, IEEE Transactions on Image Processing, Vol.5, pp.1303-1310, Sept.1996.

[16] Ilango Balasingham, ***"On Optimal Perfect Reconstruction Filter Banks For Image Compression"***, Ph.D. thesis, Norwegian University of Science and Technology, Norway, 1998.

[17] M.Vetterli and J. Kovacevic, ***"Wavelets and Subband Coding"***, Prentice Hall Signal Processing Series, Prentice Hall, Englewood Cliffs, NJ, USA, 1995.

[18] Marc Antonini, Miche, Barland, Pierre Mathieu, and Ingrid Daubechies, ***"Image Coding Using Wavelet Transform"***, IEEE Transactions on Image Processing, Vol.1, No:2, pp.205-220 ,April 1992.

[19] Amir Averbuch, Danny Lazar, and Moshe Israeli, ***"Image Compressing Using Wavelet Transform And Multiresolution Decomposition"***, IEEE Transactions On Image Processing, Vol. 5, No.8, pp. 4-15, 1996

[20] Jerome M. Shapiro, ***"Embedded Image Coding Using Zerotrees Of Wavelet Coefficients"***, IEEE Transactions on Signal Processing, Vol.41, No.12, , pp.3445-3462,December 1993.

[21] Martin Vetterli and Cormac Herley, ***"Wavelets and Filter Banks: Theory and Design"***,IEEE Transactions on Signal Processing, Vol.40, No.9, pp.2207-2232, Sept.1992.

[22] John D. Villasenor, Benjamin Belzer, and Judy Liao, ***"Wavelet Filter Evaluation For Image Compression"***, IEEE Transactions on Image Processing, Vol.4, No.8, pp.1053-1060, August. 1995.

[23] Martin Bolick, Michael J. Gormish, Edward L. Schwartz, and Alexander Keith, ***"Next Generation Image Compression And Manipulation Using CREW"***, Proceedings of International Conference on Image Processing, (ICIP'97), IEEE, 1997, http://www.crc.ricoh.com/CREW

[24] Y. Linde, A. Buzo, and R.M. Gray, ***"An Algorithm For Vector Quantizer Design"***, IEEE Transactions on Communication., Vol.COM-28, No.1, pp.84-95, 1980.

[25] B.Ramamurthi and A. Gorsho, ***"Classified Vector Quantization Of Images"***, IEEE Transactions on Communications, Vol.COM-34, pp.1105-1115, Nov. 1986.

[26] T.R. Fischer, ***"A Pyramid Vector Quantizer"***, IEEE Transactions on Information Theory, IT-32(4), pp.565-583, July 1986.

[27] T. Murakami, K. Asai, and E. Yamazaki, ***"Vector Quantizer of Video Signals"***, Electronics Letters, 18(23), pp.1005-1006, November 1982.

[28] H.M. Hang and B. Haskell, Interpolative *"Vector Quantization of Color images*, IEEE Transactions on Communication, COM-36(4), pp.465-470, 1988.

[29] M.F. Barnsley, *"Fractals Every Where"*, Academic Press, San Diego, California, 1988.

[30] M.F. Barnsley and A.D. Sloan, *"A Better Way To Compress Images"*, Byte, pp.215-223, January 1988.

[31] M.F. Barnsley and L.P. Hurd, *"Fractal Image Compression"*, A.K. Peters Ltd., Wellesley, Massachusetts, 1992.

[32] A.E. Jacquin, *"A Novel Fractal Block-Coding Technique For Digital Images"*, Proceedings IEEE International conference on Acoustic, Speech, and Signal Processing, pp.2225-2228, 1990.

[33] A.E. Jacquin, *"Image Coding Based On A Fractal Theory Of Iterated Contractive Image Transformations"*, IEEE Transactions on Image Processing, IP-1(1), pp.18-30, 1992

[34] Lester Thomas and Farzin Deravi, *"Region-Based Fractal Image Compression using Heuristic Search"*, IEEE Transactions on Image Processing, Vol.4, No.6, pp.832-838,June 1995.

[35] W.H. Chen, C.H.Smith, and S.C.Fralick, *"A Fast Computational Algorithm For The Discrete Cosine Transform"*, IEEE Transactions on Communication, COM-25(9), pp.1004-1009, September 1977.

[36] Benoit B. Mandelbrot, *"The Fractal Geometry of Nature"*, W.H.Freeman and Company, Newyork,1983.

[37] L.Lundheim, *"A Discrete Framework For Fractal Signal Modelling, in Fractal Image Compression-Theory and application,"* Y.Fisher, Ed. , Newyork, Springer-Verlog,1995.

[38] E.W.Jacobs, Y.Fisher and R.D.Boss, ***"Compression: A Study Of The Iterated Transform Method"***, Signal Processing ,Vol.29, pp 251-263, 1992.

[39] Ning Lu, ***"Fractal Imaging"***, Academic Press, San Diego, 1997.

[40] D.M.Monro, ***"Class of Fractal Transforms"*** , Electronics Letters, 29(4), pp.362-363, February 1993.

[41] D.M.Monro and F.Dudbridge, ***"Fractal Approximation Of Image Blocks"***, in Proceedings ICASSP-92, Vol.3, pp.485-488, 1992.

[42] D.M.Monro and F.Dudbridge, ***"Fractal Block Coding of Images"*** Electronics Letters, 28(11), pp.1053-1055, May 1992.

[43] Yuval Fisher(Ed.) , ***"Fractal Image Compression Theory and Applications"***, Springer-Verlag, Newyork, 1995.

[44] G.Lu and T.L.Yew, ***"Image Compression Using Quadtree Partitioned Iterated Function Systems"***, Electronics Letters, 30(1), pp.23-24, January 1994

[45] A.E.Jacquin, ***"A Novel Fractal Block-Coding technique for Digital Images"***. in Proceedings of IEEE International conference on Acoustics, speech and signal processing, Vol.4, pp.2225-2228, Alberquerque, NM, USA, April 1990.

[46] E.Reusens, ***"Overlapped Adaptive Partitioning for Image Coding based on the Theory of Iterated Function Systems"***, in Proceedings of IEEE International conference on Acoustics, speech and signal processing, Vol.5 PP.56 9-572, Adelaide, Australia, April 1994.

[47] Y.Fisher, " ***Fractal image compression, in Chaos and Fractals: New Frontiers of Science"***, Springer-Verlog, Newyork, 1992. Appendix A. Heinz-otto Peitgen, Dietmar Saupe, and Hartmut Jurgens, Authors.

[48] Y.Fisher, ***"Fractal image compression, Siggraph'92 course notes"***, Univ.of California, San Diego, http://inls.ucsd.edu/y/Fractals.

[49] Franck Davoine, Marc Antonini, Jean-March Chassery and Michel Barlaud,. ***"Fractal Image Compression based on Delaunay Triangulation and Vector Quantization"*** ,IEEE Transactions on Image processing, Vol.5, No.2, February 1996. pp.338-346.

[50] Dietmar Saupe, and Matthias Ruhl, ***"Evolutionary Fractal Image Compression"***,in Proceedings of IEEE International Conference on Image Processing, ICIP'96. Vol.I, pp.129-132, Lausanne, Switzerland, September 1996.

[51] Mathias Ruhl, Hannes Hartenstein, and Dietmar Saupe, ***"Adaptive Partitioning for Fractal Image Compression"*** ,in Proceedings of IEEE International Conference on Image Processing, ICIP '97, Santa Barbara, October 1997, ftp://ftp.informatik.uni-freiburg.de/papers/fractals.

[52] Masayuki Tanimoto, Hiroshi Ohyama, Tadahiko Kimoto, Sakae Katsuyama and Toshiaki Fujii, ***"A New Fractal Image Coding Scheme Employing Blocks Of Variable Shapes"*** ,in Proceedings of IEEE International conference on Image processing, ICIP'96, pp.137-140, Lausanne, Switzerland, September 1996.

[53] E.Reusens, *"Partitioning Complexity Issue For Iterated Functions Systems Based Image Coding"*, in Proceedings EUSIPCO '94 (European Signal Processing conference) Vol.II, pp. 878-882, Austin, TX, USA, November 1994. ftp://links.uwaterloo.ca:/pub/Fractals/Papers/External/reusens 94b.ps.gz

[54] A.E. Jacquin, ***"Fractal Image Coding: A Review"***, in Proceedings of the IEEE, Vol. 81, No.10, pp.1451-1465, October 1993.

[55] R.D.Boss and E.W.Jacobs, ***"Archetype Classifications In An Iterated Transformation Image Compression Algorithm"***, in Y.Fisher, editor, Fractal image compression-Theory and applications, chapter 4, pp.79-90, Springer-Verlag, Newyork, NY, USA, 1995.

[56] G.E.Dien and S.Lepsoy, ***"A Class Of Fractal Image Coders With Fast Decoder Convergence"***, in Y.Fisher, editor, Fractal image compression. Theory and applications, Chapter 8, pp.153-175, Springer-Verlag, Newyork, NY, USA, 1995.

[57] J .M.Beaumont, ***"Advances In Block Based Coding Of Still Pictures"***, in Proceedings IEE colloquium, the application of Fractal techniques in image processing, pp.3.1-3.6, December 1990.

[58] D.Saupe, ***"Lean Domain Pools For Fractal Image Compression"***, in Proceedings of SPIE Electronics Imaging'96, Science and Technology, still image compression II, pp.150-157, SanJose, CA,USA, January 1996, http://www.dip.ee.uct.ac.za/~brendt.

[59] B.E.Wohlberg and G.de.Jager, ***"On The Reduction Of Fractal Image Compression Encoding Time"***, in 1994 IEEE South African Symposium on communications and Signal processing (COM SIG '94) PP.158-161, Univ.of stellenbosch, October 1994. http://www.dip.ee.uct.ac.za/~brendt.

[60] Behnam Bani-Eqbal, ***"Speeding Up Fractal Image Compression"***, Proceedings from IS/&T/SPIE 1995, symposium on Electronic imaging, Vol.2418, still image compression, pp.67-74, San Jose, CA,USA, February 1995.

[61] Raouf Hamzaoui, ***"Codebook Clustering By Self-Organizing Maps For Fractal Image Compression"***, in NATO ASI conference on Fractal Image Encoding and Analysis, Trandheim, July 1995 , http://www.dip.ee.uct.ac.za/~brendt

[62] D. Saupe, ***"The Futility Of Square Isometrics In Fractal Image Compression"***, in Proceedings of IEEE International conference on Image Processing, ICIP '96, Volume- I, pp.161-164, Lausanne, Switzerland, September 1996.

[63] H.Lin and A.N. Venetsanopoulos, ***"Fast Pyramidal Search For Perceptually Based Fractal Image Compression"***, in Proceedings of IEEE International conference on Image Processing, ICIP '96, Volume I, pp.173-176, Lausanne, Switzerland, September 1996.

[64] G. Sriram, S.Vela and Hema A. Murthy, ***"New Approach To Fractal Image Compression"***, Proceedings of ADCOM '97, International Conference on Advanced Computing pp.249-254, 1997.

[65] Rauof Hamzaoui, ***"Ordered Decoding Algorithm For Fractal Image Compression"***, in Proceedings of the International Picture coding Symposium PCS '97, Berlin, September 1997. ftp:// ftp.informatik.uni-frebiburg.de/papers/fractals.

[66] Raouf Hamzaoui, ***"A New Decoding Algorithm For Fractal Image Compression"***, Electronics letters, 32(14), pp.1273-1274,1996.

[67] Raouf Hamzaoui, ***"Fast Decoding Algorithms For Fractal Image Compression"*** Institut fiir informatik - Report 86.
http: // venus.amu.edu.pl/~piotrl/compression/Hamz97~I

[68] M. Kunt, A. Ikonomopoulos and Michel Kochor, ***"Second-Generation Image-Coding Techniques"***, IEEE Proceedings, Vol.73, pp.549-574, 1985.

[69] 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, Nov.1987.

[70] H. Sanderson, and G. Crebbin, ***"Image Segmentation For Compression Of Images and Image Sequences"***, IEE Proceedings, Visual, Image Signal Processing, Vol.142, No:1, pp.15-21, February1995.

[71] Patrice Willemin, Todd R. Reed, and Murat Kant, ***"Image Sequence Coding By Split and Merge"***, IEEE Transactions on Communications, Vol.39, No:12, pp.1845-1854, December 1991.

[72] Andre Kaup, and Til Aach, ***"Coding Of Segmented Images Using Shape-Independent Basis Functions"***, IEEE Transactions on Image Processing, Vol.7, pp.937-947, July 1998.

[73] J.M. Beaumont, ***"Image Data Compression Using Fractal Techniques"***, BT Technology Journal, Vol.9, No:4, October, 1991.

[74] D. Saupe, ***"Accelerating fractal image compression by multi-dimensional nearest neighbor search"***, In J. A. Storer and M. Cohn, editors, Proceedings DCC'95 (IEEE Data Compression Conference), pp. 222-231, Snowbird, UT, USA, March 1995. http://www.dip.ee.uct.ac.za/~brendt

[75] B. Wohlberg, ***"Fractal Image Compression and the Self-Affinity Assumption: A Stochastic Signal Modelling Perspective"***, Ph.D. thesis, University of Cape Town, South Africa, August 1996, http://www.dip.ee.uct.ac.za/!brendt

[76] W.B.Pennebaker and J.L.Mitchell *"JPEG Stilt Image Data Compression Standard"*, Van Nostrand Reinhold Publishers, Newyork ,1993.

[77] Wim Sweldens, *"The Lifting Scheme: A Custom Design Construction Of Biorthogonal Wavelets"* Journal of Application and compute Harmonic analysis, Vol.3, pp.186-200, 1996.

[78] L.Daubachies, and W.Sweldens, *"Factoring Wavelet Transforms Into Lifting Steps"*, Technical Report, Bell Laboratories, Lucent Technologies, 1996.

[79] G.K.Vallace, *"The JPEG Still Picture Compression Standard"*, Communications of the ACM, Vol.43, pp.30-44.

[80] A. Said and W.A.Pearlman, *"A New Fast And Efficient Image Coding Based On Set Partitioning In Hierarchical Trees"*, IEEE Transactions on Circuits, System and Video Technology, Vol.6, pp.242-250, June 1996.

[81] K.K.Paliwal and V.Ramasubramanian, *"Effect Of Ordering On The Efficiency Of The Partial Distance Search Algorithm For Vector Quantization"*, IEEE Transactions on Communications, Vol.37, No.5, pp.538-540, May 1989.

[82] A.Nyeck, H.Mokhtari, and A.Tosser-Roussey, *"An Improved Fast Adaptive Search Algorithm For Vector Quantization By Progressive Code Book Arrangement"*, Pattern Recognition, Vol.25, No.8, pp.799-802, 1992.

[83] Chang-Da Bei, and Robert M.Gray, *"An Improvement Of The Minimum Distortion Encoding Algorithm For Vector Quantization"*, IEEE Transactions on Communications, Vol.com-33, No.10,pp.1132-1133, October, 1985.

[84] Bijit Biswas, and A.K.Mukharjee, ***"Studies On Digital Image Matching Using Ordered Hierarchical Technique"***, IETE Technical Review, Vol.15, No.3, pp.147-152, May-June 1998.

[85] William K.Pratt, ***"Digital Image Processing"*** ,John Wiley and sons, Inc, Newyork, 1991.

[86] Junji Maeda, V.Anh,* Tohru Ishizaka, and Yukinori Suzuki, ***"Integration Of Local Fractal Dimension And Boundary Edge In Segmenting Natural Images"***, Proceeding of IEEE International Conference on Image Processing ,ICIP'96, Vol.I of III, pp.845- 843, Lausanne, Switzerland, September 1996.

[87] Osma K.Al-Shaykh, and Russell M.Mersereau, ***"Lossy Compression of Noisy Images"***,IEEE Transactions on Image Processing, Vol.7, No.12, pp.1641-1652, December 1998.

[88] H.R.Mahadevaswamy, P.Janardhanan, Y.Venkataramani, ***" Advances in Fractal Image Compression"***, Proceedings of the National Seminar on Artificial Neural Networks and Cognitive Systems, ANCS'98, CUSAT, Cochin, India, pp.164-167, September 1998.

[89] Øyvind StrØmme, ***"On the Applicability of wavelet Transforms to Image and Video Compression"***, Ph.D. Thesis, University of Strathclyde, UK, February 1999

[90] H.R.Mahadevaswamy, P.Janardhanan and Y.Venkataramani, ***"Fast Fractal Image Compression"***, Proceedings of National Seminar on Applied Systems Engineering and Soft Computing, SASESC-2000, Agra, March 4th and 5th, 2000 pp.245-250

[91] S.Peleg, J.Naor, R.Hartly and D.Avnir, ***"Multiple Resolution Texture Analysis and Classification"***, IEEE Transactions on PAMI, Vol.6, No.4, pp. 518-523,1984

[92] Rusen Öktem, ***"Transform Domain Algorithms for Image Compression and Denoising"***, Ph.D. Thesis, Tampere University of Technology, May 2000.

[93] T.R.Fischer, J.D.Gibson, and B.Koo, *" Estimation and Noise Source Coding"*, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol.38, No.1, pp. 23-34,1990

[94] K.Ramachandran, M.Vetterli, and C.Herley, ***"Wavelets, Subband Coding and Best bases"***, Proc. Of the IEEE, Vol.84. No.4, pp.541-560, 1996

[95] G.Starng and T.Nguyen, ***"Wavelets and Filter Banks"***, Wellesley-Cambridge Press, 1996

[96] W.Sweldens, ***"Wavelets: What Next?"***, Proc. Of the IEEE, 85(4), pp. April 1997

[97] Yung-Ching Chang, Bin-Kai Shyu and Jia-Shung Wang, ***"Region –based Fractal Image Compression with Quadtree Segmentation"***, International Conference on Acoustics, Speech and Signal Processing, ICASSP'97, pp.3125-3128

[98] Matthais Ruhl, Hannes Hartenstein, Dietmar Saupe, ***"Adaptive Partitioning for FractalImage Compression"***, International conference on Image Processing. ICIP'97, Santa Barbara, Oct 1997

[99] A.P.Pentaland, ***"Fractal-based Description of Natural scenes"***, IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol.6, No.6, pp.661-674

[100] Dana H Ballard, Christopher M Brown, ***"Computer Vision"***, Prentice-Hall Inc., 1082

APPENDIX A

STANDARD TEST IMAGES

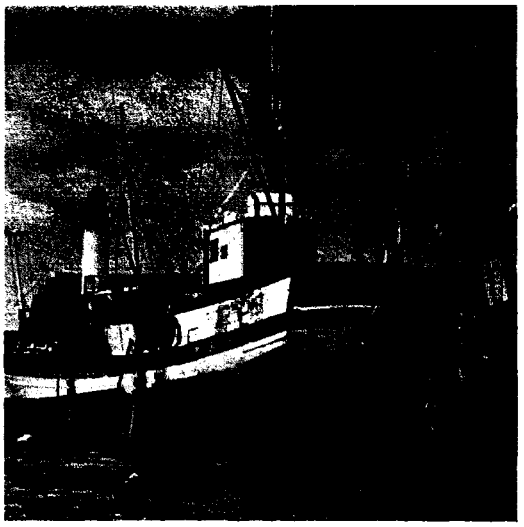


Fig. A.1 Original boat 256 x 256

Fig. A.2 Original lenna 256 x 256

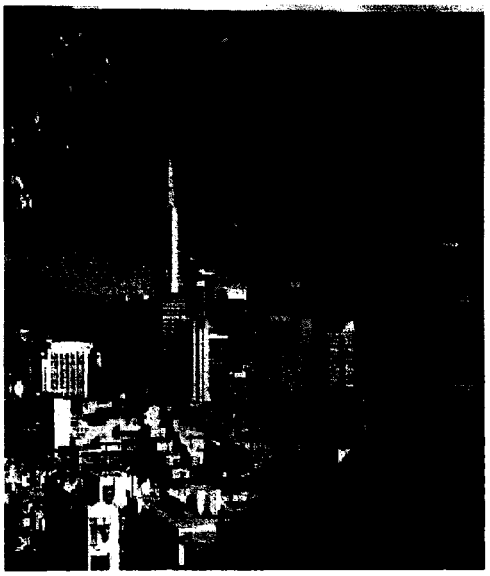


Fig. A.3 Original sanfransisco 256 x 256

Fig. A.4 Original camera 256 x 256

Figure A.5 Original peppers 256 x 256

Figure A.6 Original bridge 256 x 256

APPENDIX B

The Contractive Mapping Fixed –Point Theorem [29]

Let X be complete metric space and f: X→X be a contractive mapping. Then there exists a unique point $x_f \in X$ such that for any point $x \in X$

$$x_f = f(x_f) = \lim_{n \to \infty} f^{on}(x)$$

Such a point is called a fixed point or the attractor of the mapping f.

Proof: Select $x \in X$. Then for n>m,

$d(f^{om}(x), f^{on}(x)) < sd(f^{om-1}(x), f^{on-1}(x)) < s^m d(x, f^{on-m}(x))$

Now, using this and the triangle inequality repeatedly,

$$\begin{aligned} d(x, f^{ok}(x)) &\le d(x, f^{ok-1}(x)) + d(f^{ok-1}(x), f^{ok}(x)) \\ &\le d(x, f(x)) + d(f(x), f(f(x))) + \ldots\ldots + \\ &\quad d(f^{ok-1}(x), f^{ok}(x)) \\ &\le (1 + s + \ldots + s^{k-2} + s^{k-1}) d(x, f(x)) \\ &\le \frac{1}{1-s} d(x, f(x)). \end{aligned} \qquad \text{B.1}$$

Then writing Equation (2.2) as

$$d(f^{om}(x), f^{on}(x)) < \frac{s^m}{1-s} d(x, f(x)),$$

and since s<1, the left side can be made as small as possible, if n and m are sufficiently large. This means that the sequence x, f(x), f(f(x)),.... is a Cauchy sequence, and since X is complete, the limit point $x_f = \lim_{n\to\infty} f^{on}(x)$ is in X. Contractivity of f implies that f is continuous, and so $f(x_f) = f(\lim f^{on}(x)) = \lim f^{on+1}(x) = x_f$.

To prove uniqueness : suppose x_1 and x_2 are both fixed points, then $d(f(x_1), f(x_2)) = d(x_1, x_2)$, but we also have $d(f(x_1), f(x_2)) \leq s d(x_1, x_2)$, a contradiction.

Collage Theorem[29]

$$d(x, xf) \leq \frac{1}{1-s} d(x, f(x)),$$

Proof : This is a consequence of taking the limit as k goes to infinity in Eqn B.1

Note that it is not necessary for f to be contractive for it to have a fixed point which attracts all of X. For example, it is sufficient for some iterate of to be contractive. This leads to the following generalization of the theorem.

Definition B.1: Let f be a lipschitz function. If there is a number n such that f^{on} is contractive, then we call f eventually contractive. We call n the exponent of eventual contractivity.

Generalized Collage Theorem

Let f be eventually contractive with exponent n: then there exists a unique fixed point $x_f \in X$ such that for any $x \in X$

$$x_f = f(x_f) = \lim_{k \to \infty} f^{ok}(x)$$

In this case,

$$d(x, x_f) \leq \frac{1}{1-s} \frac{1-\sigma^n}{1-\sigma} d(x, f(x))$$

where s is the contractivity of f^{on} and σ is the Lipschitz factor of f.

Proof: Let $g = f^{on}$. To show that f^{ok} converges to x_g; that is, $f^{ok}(x)$ is arbitrarily close to x_g for all sufficiently large k. For any k, we can write $k = qn+r$, with $0 \leq r < n$. So,

$$
\begin{aligned}
d(f^{ok}(x),x_g) &= d(f^{oqn+r},x_g) \\
&\le d(f^{oqn+r}(x),f^{oqn}(x)) + d(f^{oqn}(x),x_g) \\
&= d(g^{oq}(f^{or}(x)),g^{oq}(x)) + d(g^{oq}(x),x_g) \\
&\le s^q d(f^{or}(x),x) + d(g^{oq}(x),x_g).
\end{aligned}
$$

But both of these terms can be made arbitrarily small for $0 \le r < n$ and q sufficiently large. The fixed-point condition follows from the continuity of f, and uniqueness follows from the uniqueness of x_g.

For the inequality, it is know from the Collage Theorem that:

$$
d(x,x_g) \le \frac{1}{1-s} d(x,g(x)) \qquad \text{B.2}
$$

and

$$
\begin{aligned}
d(x, g(x)) &= d(x,f^{on}(x)) \le \sum_{i=1}^{n} d(f^{oi}(x),f^{oi-1}(x)) \\
&\le d(x,f(x)) \sum_{i=1}^{n} \sigma^{i-1} \\
&\le \frac{1-\sigma^n}{1-\sigma} d(x.f(x)). \qquad \text{B.3}
\end{aligned}
$$

The result follows from eqn.B.2 and B.3.

APPENDIX C

RMS Metric

For comparison of domain and ranges, root- mean-square (rms) metric is used [43]. This allows easy computation of optimal values for s_i and o_i in Equation(3.4).

Given two square containing n pixel intensities, a_1 - - - - - -a_n(from D) and b_1- - - - b_n (from Ri), we can calculate s and o to minimize the quantity

$$R = \sum_{i=1}^{n} (s.a_i + o - b_i)^2$$

This will give contrast and brightness settings that make the affinely transformed a_i values have the least squared distance from the b_i values. The minimum R occurs when the partial derivatives with respect to s and o are zero, which occurs when

$$s = \frac{\left[n\sum_{i=1}^{n} a_i b_i - \sum_{i=1}^{n} a_i \sum_{i=1}^{n} b_i \right]}{\left[n\sum_{i=1}^{n} a_i^2 - \left(\sum_{i=1}^{n} a_i \right)^2 \right]}$$

$$o = \frac{1}{n}\left[\sum_{i=1}^{n} b_i - s\sum_{i=1}^{n} a_i \right]$$

and

In that case

$$R = \frac{1}{n}\left[\sum_{i=1}^{n} b_i^2 + s\left(s\sum_{i=1}^{n} a_i^2 - 2\sum_{i=1}^{n} a_i b_i + 2 \times o\sum_{i=1}^{n} a_i \right) + o\left(no - 2\sum_{i=1}^{n} b_i \right) \right]$$

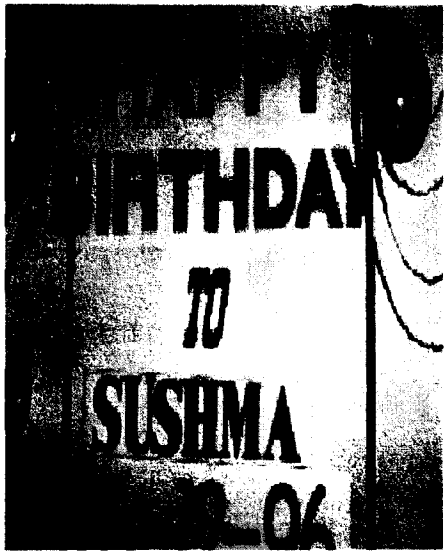
The rms error is equal to $\sqrt{R}$.

APPENDIX D

(a) Administrative block of CREC

Of size 256x256 adm256.dat

(b) image of car of size 256x256 ca256.dat

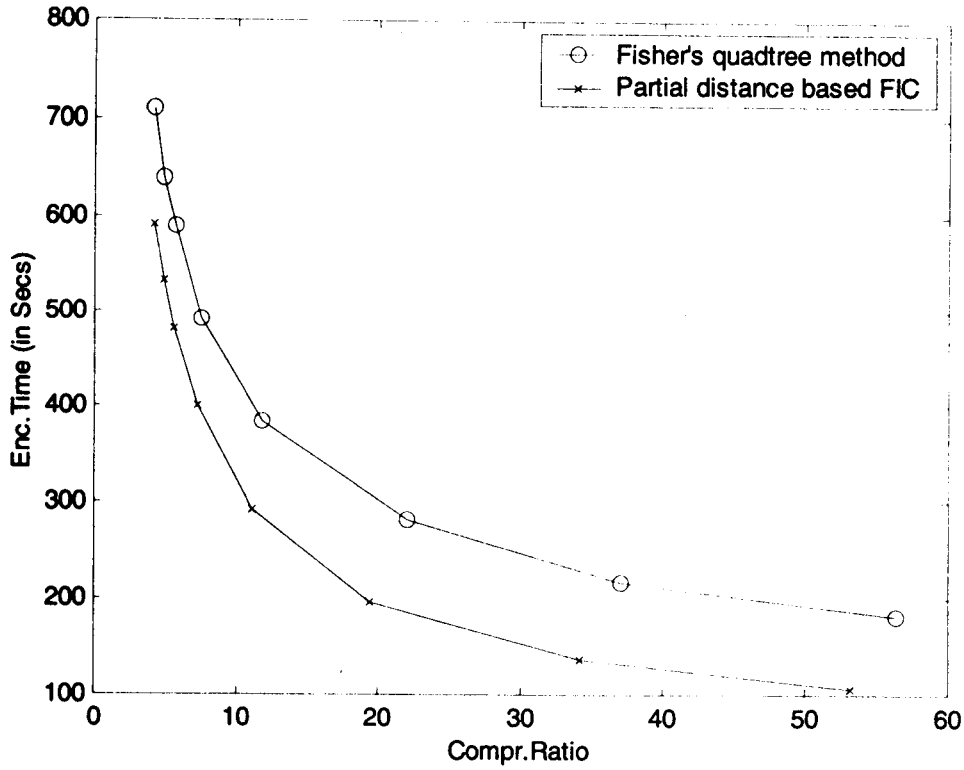


(c) text image of size 256x256 text256.dat

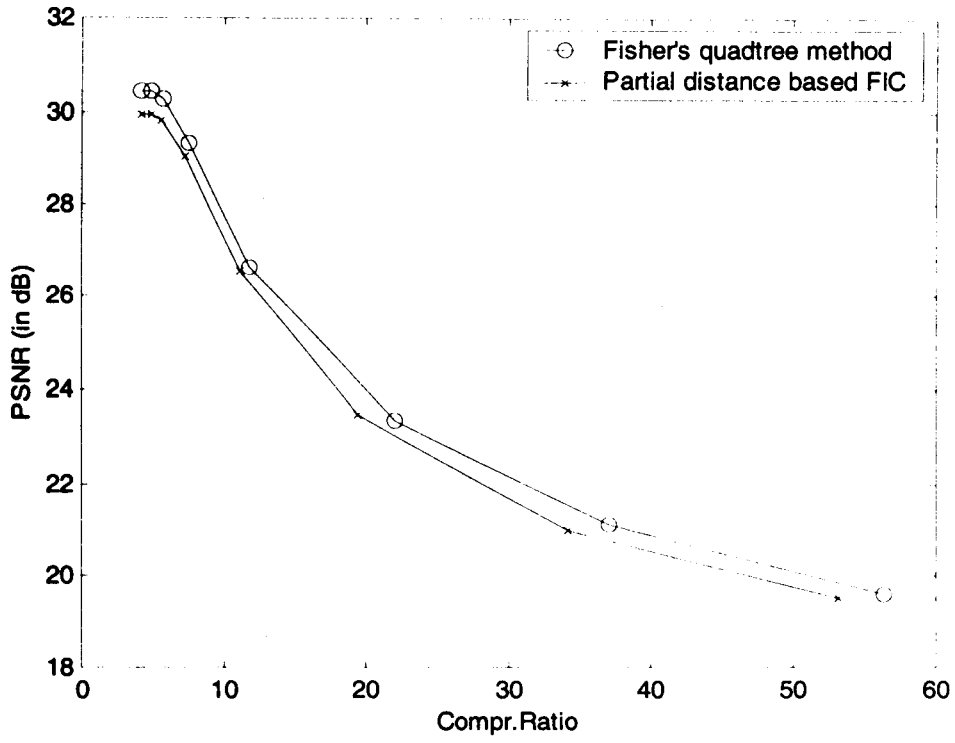
Figure(D1) Original non-standard images

Compr. Ratio	Fisher's quadtree method			Partial distance based FIC			Speed up factor
	Enc.Time (in Secs)	RMS error	PSNR (in dB)	Enc.Time (in Secs)	RMS error	PSNR (in dB)	
4	710	7.61	30.5	590	8.06	30	1.203
8	475	9.048	29	375	9.58	28.5	1.267
12	375	12.06	26.5	280	12.49	26.2	1.34
20	300	16.08	24	190	17.24	23.4	1.579
30	250	19.79	22.2	155	20.96	21.7	1.61
40	210	22.72	21	125	24.07	20.5	1.68

Table (D1) : Encoding Time and PSNR for various values of Compression Ratio for non-standard image administrative block of CREC of size 256x256 using Partial distance based fractal image compression



(a)Plot of Compression ratio Vs encoding time



(a) Plot of Compression ratio Vs. PSNR

Figure (D2) Performance graph for image adm256.dat for partial distance based FIC method

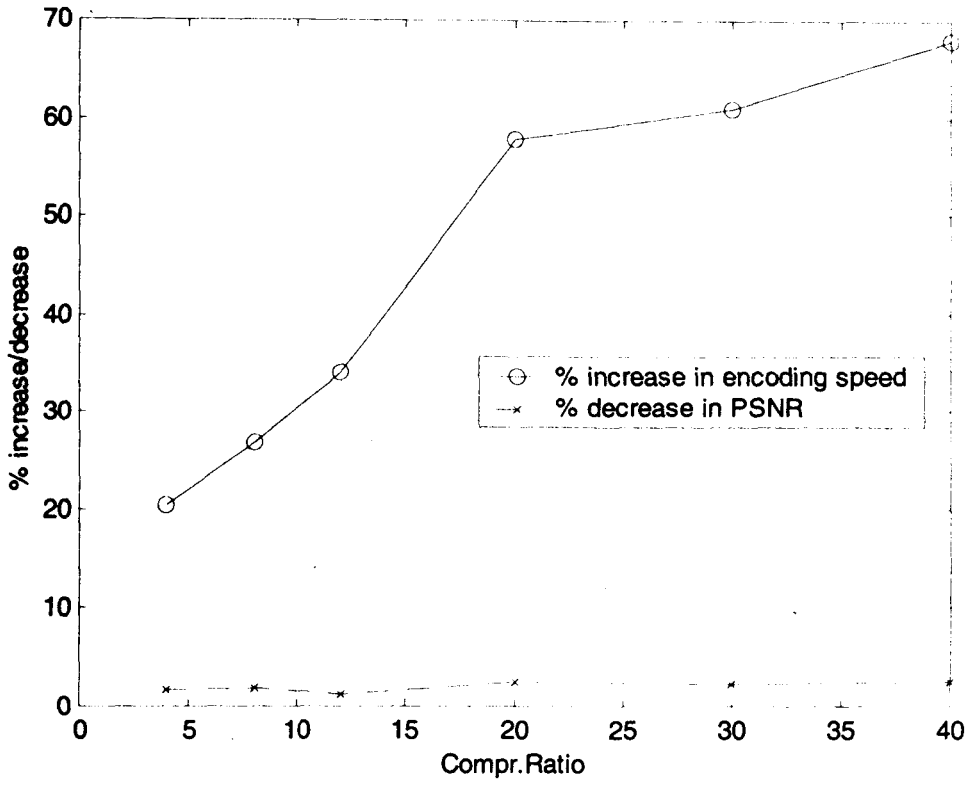
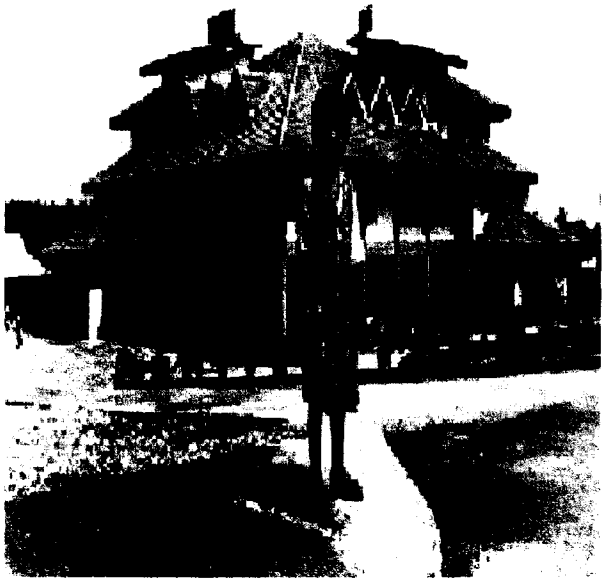
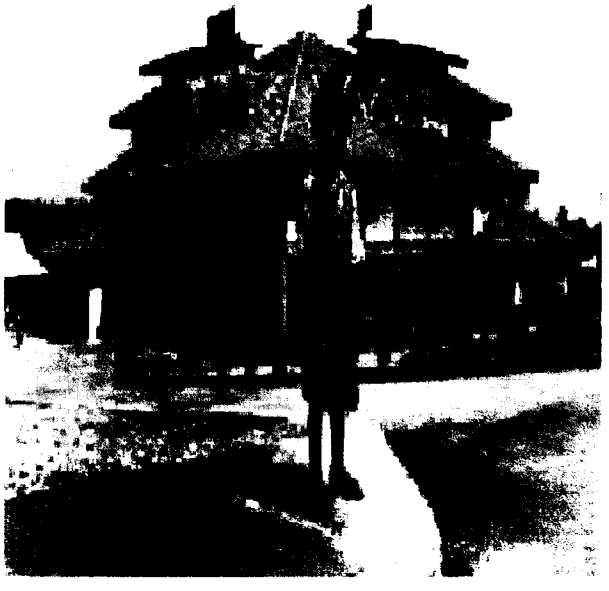


Figure (D3) Plot of Compression Ratio Vs. % increase in encoding speed and % decrease in PSNR for non_standard image adm256.dat of size 256x256 (Partial distance based FIC)



(a)Compressed and decoded image of REC Administrative building adm256.dat using Fisher;s quadtree method, Compr.Ratio=4.2, PSNR=30.43dB

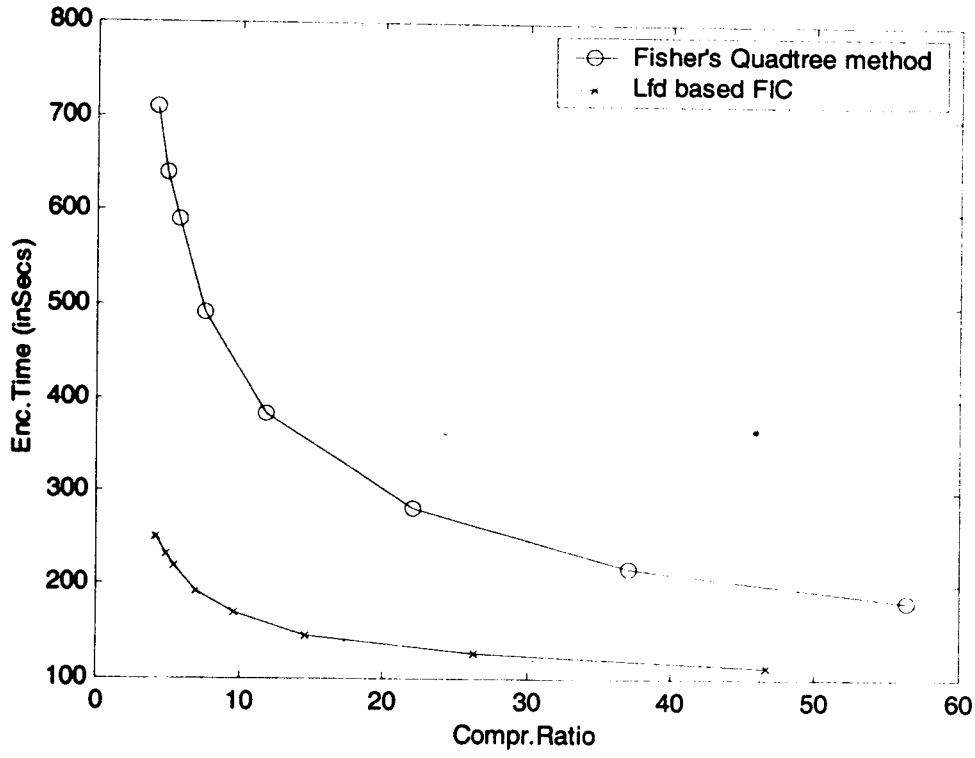


(b) compressed image at Compr.Ratio = 4.2, PSNR=29.93dB(Partial distance based FIC)

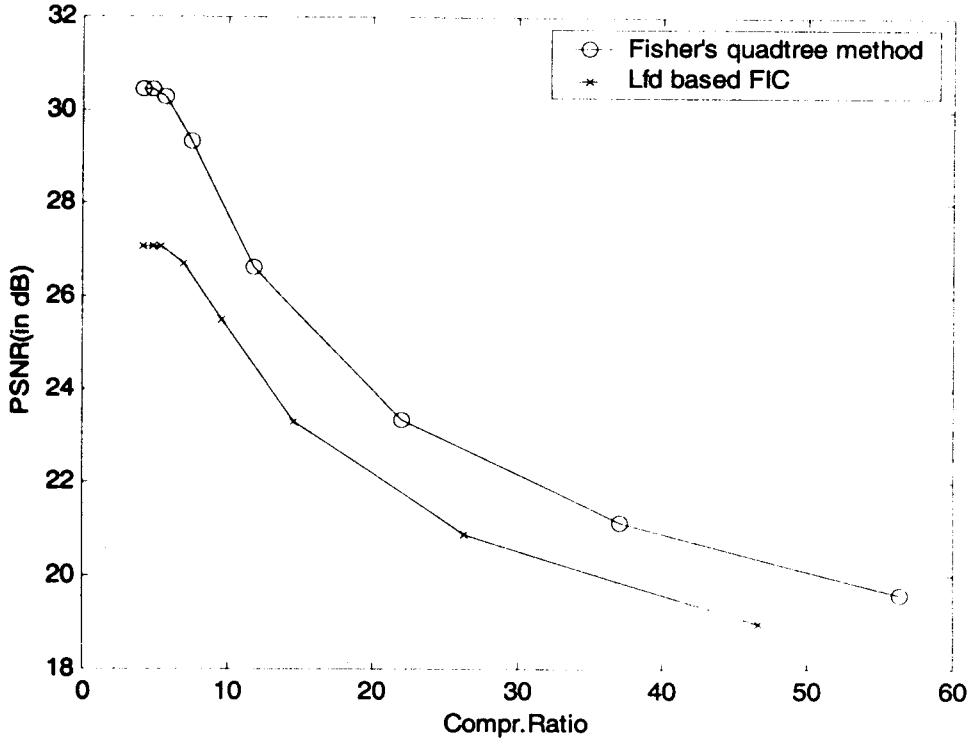
Figure (D 4) Compressed and decoded image adm256.dat using Fisher's quadtree method and Partial distance based FIC

Compr. Ratio	Fisher's quadtree method			Partial distance based FIC			Speed up factor
	Enc.Time (in Secs)	RMS error	PSNR (in dB)	Enc.Time (in Secs)	RMS error	PSNR (in dB)	
4	710	7.61	30.5	250	11.13	27.2	2.84
8	475	9.048	29	180	12.48	26.2	2.64
12	375	12.06	26.5	155	15.18	24.5	2.42
20	300	16.08	24	135	19.56	22.3	2.22
30	250	19.79	22.2	125	24.07	20.5	2
40	210	22.72	21	120	26.7	19.6	1.75

Table (D2) : Encoding Time and PSNR for various values of Compression Ratio for non-standard image administrative block of CREC of size 256x256 using Local fractal dimension based fractal image compression



(a) Compression Ratio Vs. Encoding Time



(b) Compression Ratio Vs.PSNR

Figure (D5) Performance graph for image adm256.dat for Local fractal dimension based FIC

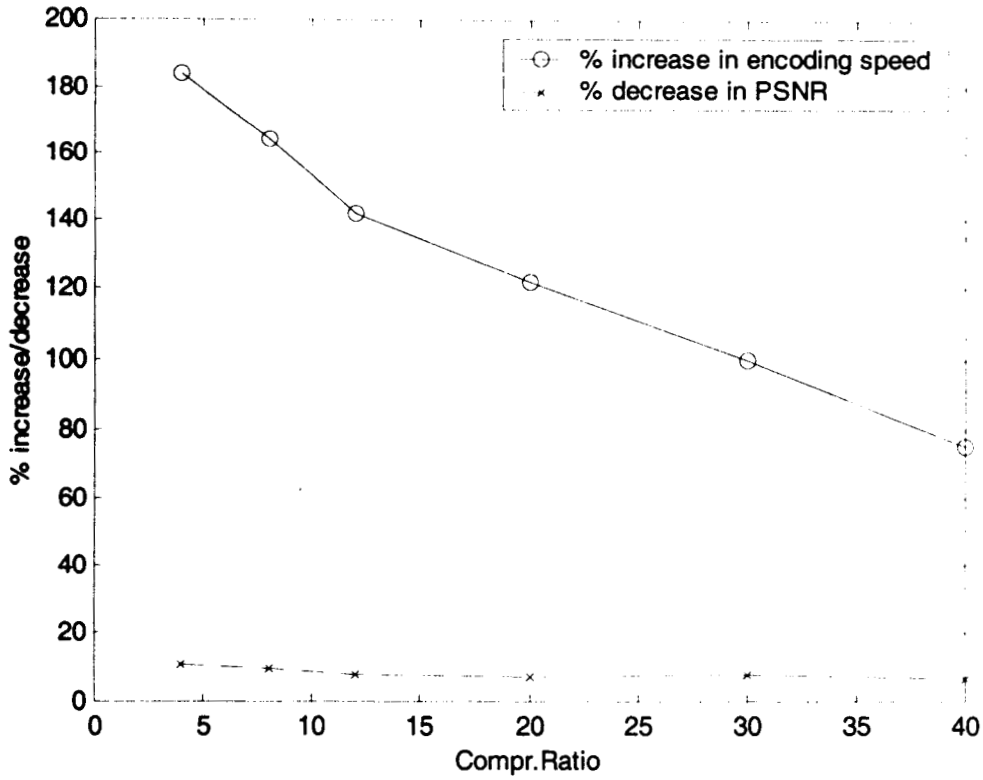
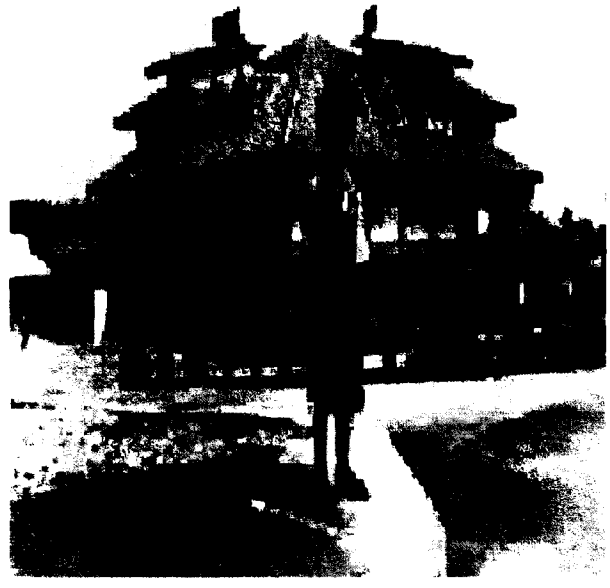


Figure (D6) Plot of Compression Ratio Vs. % increase in encoding speed and % decrease in PSNR for non_standard image adm256.dat of size 256x256 (Local fractal dimension based FIC)

(a)Compressed and decoded image of REC Administrative building adm256.dat using Fisher;s quadtree method, Compr.Ratio=4.2, PSNR=30.43dB

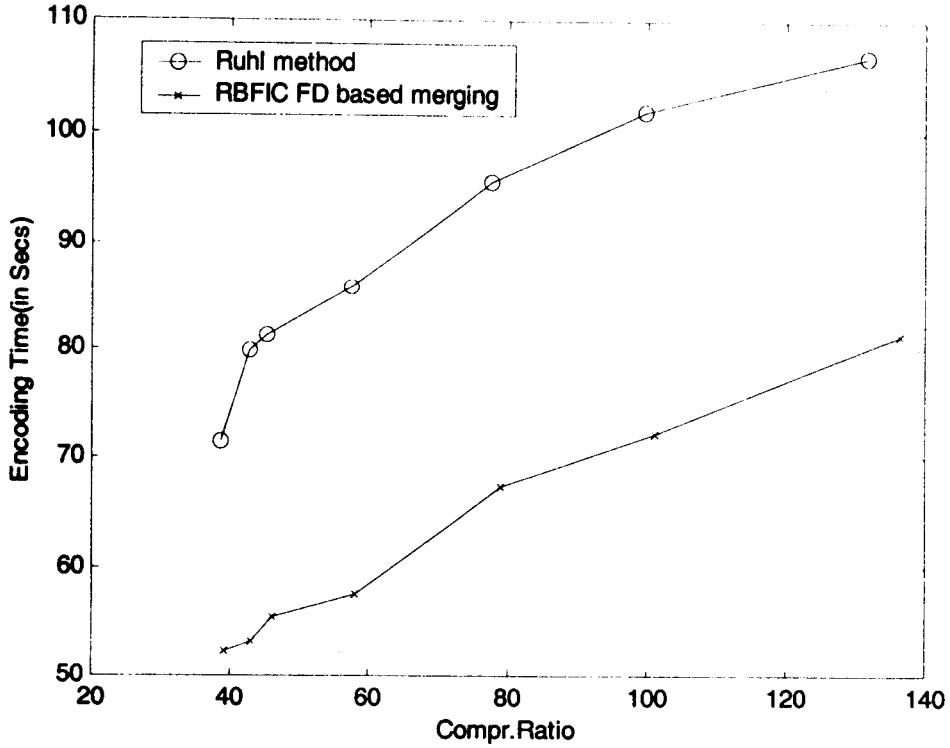


(b) compressed image at Compr.Ratio = 4.2, PSNR=29.93dB(Partial distance based FIC)

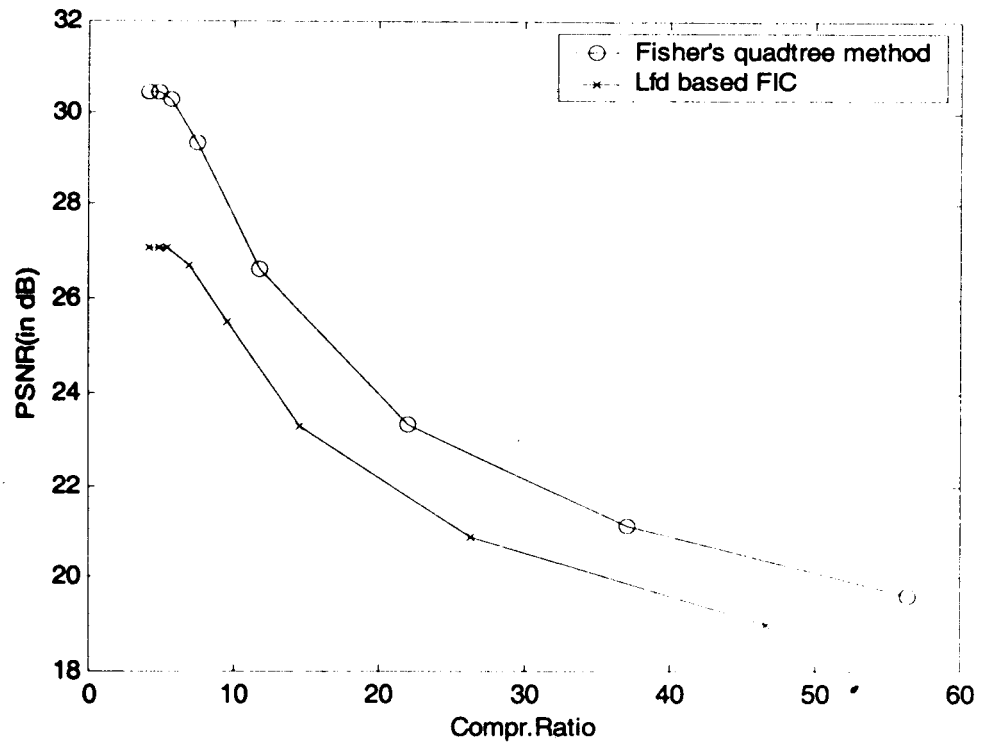
Figure (D7) Compressed and decoded image adm256.dat using Fisher's quadtree method and Partial distance based FIC

Compr. Ratio	Ruhl's RBFIC Method			FD based RBFIC			Speed up factor
	Enc.Time (in Secs)	RMS error	PSNR (indB)	Enc.Time (in Secs)	RMS error	PSNR (indB)	
40	74	19.34	22.4	52	18.47	22.8	1.423
60	87	22.2	21.2	59	21.7	21.4	1.475
80	96	23.25	20.8	68	22.72	21	1.412
100	102	25,35	20.05	72	24.21	20.45	1.417
120	105	28.61	19	77	27.16	19.45	1.364
130	106.5	30.65	18.4	80	29.28	18.8	1.331

Table (D3) Results for base size=8 for non-standard image adm256.dat of size 256x256 using Region-based fractal image compression based on fractal dimension for various values of compression ratio



(a) Compression Ratio Vs. Encoding Time



(b) Compression Ratio Vs.PSNR

Figure (D8) Performance graph for image adm256.dat for fractal dimension based Region based fractal image compression for basesize=8

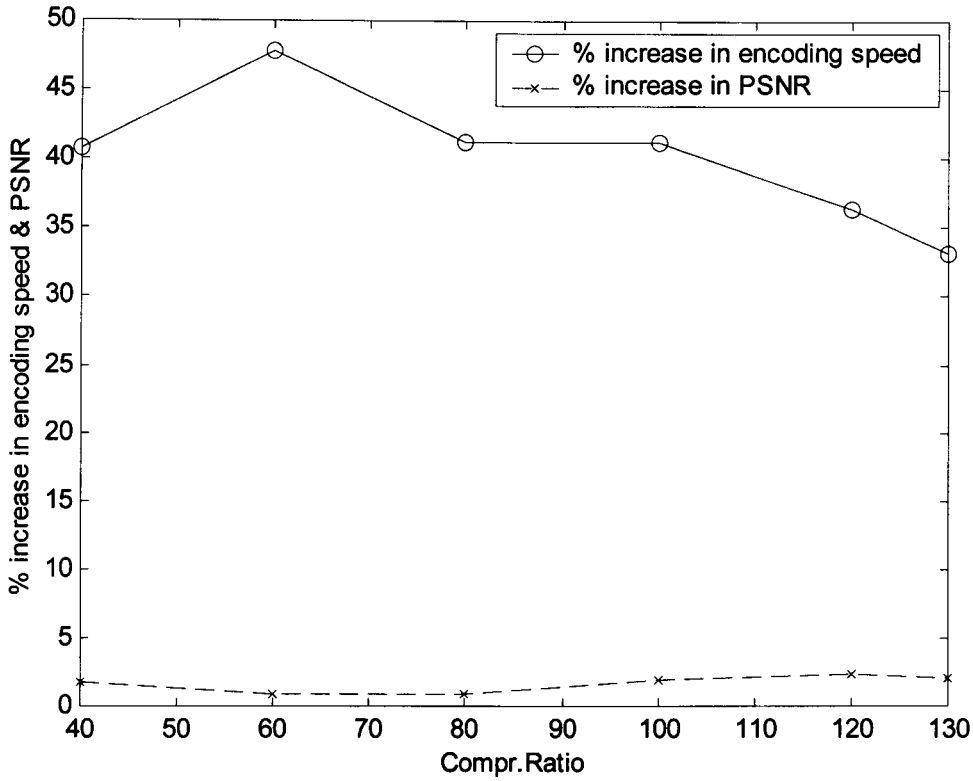
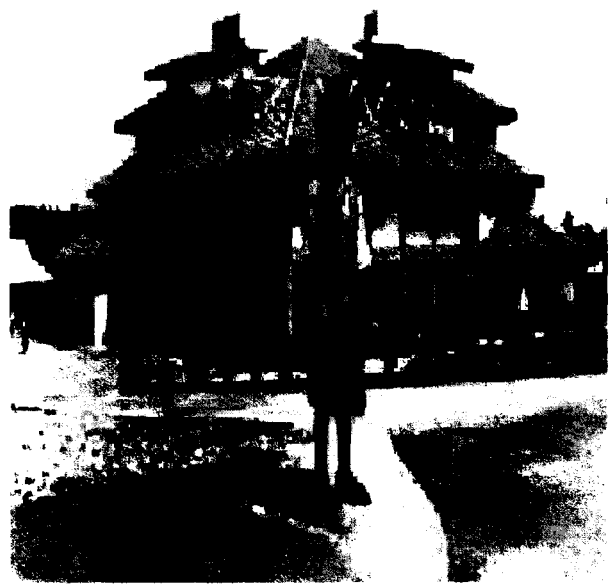
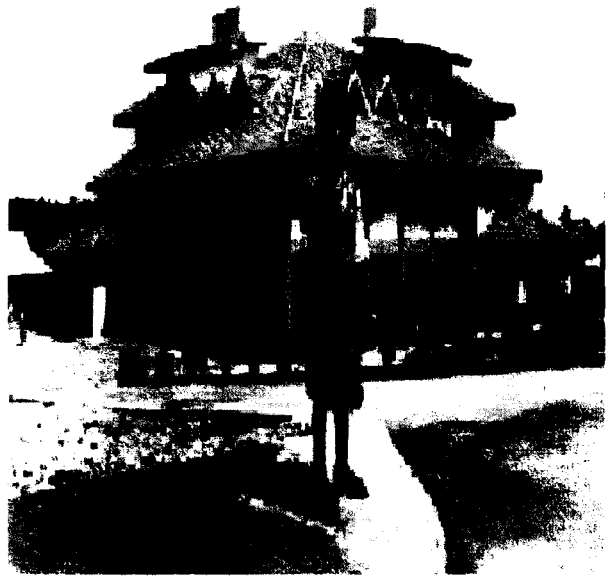


Figure (D9) Plot of Compression Ratio Vs. % increase in encoding speed and PSNR for non-standard image adm256.dat of size 256x256 for basesize =8(Fractal dimension based RBFIC)



(a) Compressed and decoded image for Compr.ratio=40, PSNR=22.4dB Using Ruhl's RBFIC method



(b)Compressed and decoded image for Compr.Ratio=40, PSNR=22.48dB using fractal dimension based RBFIC

Figure (D10) Decoded images for basesize=8

Wavelet transform type	S- transform (1,1)	(2,2)	(4,2)	(4,4)	(2,2+2)	Adaptive method
First order Entropy	4.3646	4.7487	5.101	5.109	5.1751	3.7826

Table D4 : First order Entropy for results for non-standard image administrative building of REC Calicut for Lossless Image Compression using Integer wavelets

RESEARCH PUBLICATIONS

The following papers have been published/submitted so far:

1. H.R.Mahadeavaswamy, P.Janardhanan and Y.Venkataramani, "Advances in Fractal Image Compression", Proceedings of the Networks and Seminar on Artificial Neural Networks and cognitive system", ANCS'98, CUSAT Cochin, PP.164-167, September 1998.

2. H.R.Mahadeavaswamy, P.Janardhanan and Y.Venkataramani, "Lossless Image Compression using Wavelets – A comparative study", Proceedings of 5th National Communication Conference", NCC'99, IIT, Kharagpur, PP.345-352, Allied Publishers New Delhi, January 1999.

3. H.R.Mahadevaswamy, P.Janardhanan and Y.Venkataramani, "Fast Fractal Image Compression", Proceedings of National Seminar on Applied Systems Engineering and Soft Computing, SASESC-2000, Agra, pp. 245-250, March 4 and 5, 2000

4. H.R.Mahadevaswamy, P.Janardhanan and Y.Venkataramani, "Local Fractal Dimension Based Domain Classification In Fractal Image Compression", Submitted to Journal of Circuits, System, and Signal Processing, USA

5. H.R.Mahadevaswamy, P.Janardhanan and Y.Venkataramani, "Fractal Image compression of Noisy Images", *7th National Communication Conference", NCC'2001, IIT, Kanpur* (accepted)

6. H.R.Mahadevaswamy, P.Janardhanan and Y.Venkataramani, "Lossless Image Compression Using Adaptive Integer Wavelet Coding", Submitted to National Conference on Technology Convergence for Information, Communication and Entertainment (NICE2001)

7. H.R.Mahadevaswamy, P.Janardhanan and Y.Venkataramani, "Fractal Image Compression With Reduced Encoding Time" , Submitted to IETE Journal of Research