CONTEXTUAL COMPRESSION OF ULTRASOUND MEDICAL IMAGES

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree

of

MASTER OF TECHNOLOGY

in

ELECTRICAL ENGINEERING

(With Specialization in Measurement & Instrumentation)

By

KHER RAHULKUMAR KIRANSINH

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)

JUNE, 2006

IDNO. MT-316|2006-11|KRK-RSA

CANDIDATE'S DECLARATION

I hereby declare that the work, which is being presented in this dissertation, entitled **"Contextual Compression of Ultrasound Medical Images"** in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in Electrical Engineering with specialization in **Measurement & Instrumentation,** submitted in the Department of Electrical Engineering, **Indian Institute of Technology, Roorkee**, India is an authentic record of my own work carried out during the period from July 2005 to May 2006, under the guidance and supervision of **Dr. R.S. Anand,** Associate Professor, Department of Electrical Engineering, Indian Institute of Technology, Roorkee, India.

The matter embodied in this dissertation has not been submitted by me for the award of any other degree or diploma.

Date: 20th June, 2006

Place: Roorkee

20/6/2016

(Kher Rahulkumar Kiransinh)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

20.6.2006

(Dr. R.S. Anand)
Associate Professor,
Department of Electrical Engineering,
Indian Institute of Technology, Roorkee.
Roorkee-247 667.
Uttaranchal, India.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to **Dr. R.S. Anand,** Professor, Department of Electrical Engineering, Indian Institute of Technology, Roorkee for being constant source of motivation throughout my project. He listened patiently and authoritatively as he guided me and made his valuable suggestions. He has been very helpful in all aspects of my work. The helpful and stimulating discussions aided a lot to the success of this thesis. I owe him a great debt of gratitude.

I express my deep and sincere sense of gratitude to **Dr. S.P. Gupta**, Head of the Electrical Department, Indian Institute of Technology and to all the faculty members of Measurement & Instrumentation group.

I am indeed thankful to Shri Vibhkar Shrimali, Shri Ashish Thakur and Shri Sukhvindar Singh- Research scholars of the Electrical Engineering Department for providing valuable guidance and help as and when required.

I am thankful to all those who have helped directly or indirectly without whose support it would have been difficult to carry on this work.

These acknowledgements would be incomplete without thanking my fellow M.Tech friends for always being the source of encouragement to work hard.

Finally, I would like express my deepest gratitude to the Almighty God for showering blessings on me. I gratefully acknowledge my heartiest thanks to all my family members, especially ***my Mother*** and ***Wife***, and friends for their inspirational impetus and moral support during the course of this work.

Abstract

Medical image compression addresses the issues of larger storage and faster transmission requirements. *Contextual compression* of Ultrasound (US) medical images aims at compressing the diagnostically important region, region of interest (RoI), of an image with supreme quality as compared to rather unimportant area i.e. the background. Thus, the RoI area is compressed with less compression ratio and the background with the highest possible compression ratio in order to get better overall compression performance.

As a part of contextual compression technique four compression algorithms viz. JPEG coding, Wavelet Transform coding, JPEG 2000 coding and the SPIHT coding algorithms have been implemented in the present work. These algorithms have been implemented using MATLAB, Image Processing and Wavelet toolbox, in particular. A detailed analysis on the basis of parameters like mean square error (MSE), peak signal to noise ratio (PSNR) and correlation coefficient (CoC) has been carried out to evaluate these algorithms.

It is observed that JPEG 2000 and SPIHT compression algorithms perform better at lower bit rates (i.e. bpp below 0.20), whereas JPEG coding should not be used for lower bit rates. The Wavelet transform coding has outperformed the rest of the compression algorithms in terms of MSE, PSNR, CoC and of course, the visual quality of a compressed image. The Wavelet transform has an excellent energy retaining capability, which is responsible for its outstanding performance. In fact, the other two Wavelet-based coding schemes viz. the JPEG 2000 and SPIHT too, are not far behind in performance.

The *contextual compression* scheme has achieved a compression ratio (CR) of as high as 65.24:1 an US image, whereas a *simple or conventional compression* scheme could hardly achieve the CR of 40:1. The prime contribution of a contextual compression scheme over a conventional compression scheme is improvement in CR and visual quality of an RoI of a compressed image with modest values of MSE and PSNR.

List of Figures

List of Tables

Acronyms

US	Ultrasound
RoI	Region of Interest
CR	Compression Ratio
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
CoC	Correlation Coefficient
bpp	Bits per pixel
PACS	Picture Archiving and Communication System
WT	Wavelet Transform
DFT	Discrete Fourier Transform
DCT	Discrete Cosine Transform
PCM	Pulse Code Modulation
DPCM	Differential Pulse Code Modulation
WHT	Walsh-Hadamard Transform
EBCOT	Embedded Block Coding with Optimized Truncation
HVS	Human Visual System
DWT	Discrete Wavelet Transform
CWT	Continuous Wavelet Transform
IWT	Integer Wavelet Transform
JPEG	Joint Photographic Expert Group
JBIG	Joint Bi-level Imaging Group
STFT	Short Term Fourier Transform
MRA	Multi Resolution Analysis
CCITT	Consultative Committee of the International Telephone and Telegraph
ITU	International Telecommunication Union
ISDN	International Services Digital Network
JP2	JPEG 2000
MJP2	Motion JPEG 2000

SPIHT	Set Partitioning In Hierarchical Trees
EZW	Embedded Zero-tree Wavelet
LSP	List of Significant Pixels
LIP	List of Insignificant Pixels
LIS	List of Insignificant Sets

CONTENTS

Chapter 1

Introduction

1.1 Medical Image Compression

Medical images acquired from various modalities, such as Magnetic Resonance Imaging, Computed Tomography, ***Ultrasonography***, Nuclear Medicine, Computed Radiography, and Digital Subtraction Angiography, comprise huge amount of data rendering them impractical for storage and transmission. Since the hospitals switching over to picture archiving and communication systems (PACS) and Teleradiology, even in an average sized hospital many terabytes (10^{12}) of digital imaging data are generated every year and almost all of which have to be stored and archived [1]. The objective of medical image compression is to reduce the data volume of and to achieve a low bit rate in the digital representation of Radiologic images such as X-ray, MR, CT and ***Ultrasound*** images without perceived loss of image quality. However, the demand for transmission bandwidth and storage space in the digital radiology environment, and the proliferating use of various imaging modalities continue to outstrip the capabilities of existing technologies.

It has been observed that the amount of digital radiologic images captured per year in the United States alone is at the order of petabytes, i.e., 10^{15}, and is increasing rapidly every year. Image compression facilitates PACS as an economically viable alternative to analog film-based systems by reducing the bit size required to store and represent images, while maintaining relevant diagnostic information. It also enables fast transmission of large medical images over a PACS network to display workstations for diagnostic, review, and teaching purposes. Even over a Fiber Distributed Data Interchange (FDDI) network of 100 megabits per sec (Mb/s) and at an optimistic 20% actual transfer rate (i.e., 20 Mb/s), it would take 13 seconds to transmit a digitized chest image of 4096 x 4096, 12-bit resolution; whereas many medical applications demand the display of image in less than 2 s. This transmission problem is further aggravated in wide area network (WAN) applications which often include low-bandwidth channels, such as long distance telephone lines or an Integrated Services Digital Network (ISDN) of data rate 144 kilobits per second (Kb/s) [2].

1.2 Methods for Medical Image Compression

Technically, all image data compression schemes can be broadly categorized into two types. One is ***reversible compression,*** also referred to as "lossless." A reversible scheme achieves modest compression ratios of the order of 2 to 3, but will allow exact recovery of the original image from the compressed version. An ***irreversible scheme,*** or a "lossy" scheme, will not allow exact recovery after compression, but can achieve much higher compression ratios, e.g., ranging from 10 to 50:1 or more. Generally speaking, more compression is obtained at the expense of more image degradation, i.e., the image quality declines as the compression ratio increases [2]. For medical images, it is of utmost importance to preserve, after compression, the quality of an image, so as not to arrive at the false diagnosis of a patient on the basis of a compressed image.

Now, image compression techniques like Huffman coding, Arithmetic coding, PCM and DPCM, LZW coding, Bit-plane coding etc. are considered as lossless (reversible) techniques and can not achieve higher compression ratios (could achieve the compression ratio of only up to 3:1 for medical images), and hence could not be worth to solve the problem of the data storage capability and faster transmission. Hence lossy (irreversible) techniques, which may sometimes be referred to as ***near lossless techniques***, like discrete Fourier transform (DFT), discrete cosine transform (DCT), Walsh-Hadamard transform (WHT), Wavelet transform and the (lossless) JPEG and JPEG 2000 have been proving quite useful for the purpose.

The later mentioned techniques can achieve significant compression ratios up to 40:1 (which is acceptable in the medical community), without degrading an image to an unacceptable level. The Wavelet transform based medical image compression techniques have an edge over the other near lossless techniques as far as the factors such as image quality (based on human visual system-HVS), mean-square error (MSE), correlation coefficient (CoC), percentage rate distortion (PRD) and the peak signal to noise ratio (PSNR) are concerned. Hence the latest research in the field of Medical Image compression involves the Wavelet transform based methods such as Space frequency segmentation, set partitioning in hierarchical trees (SPIHT), embedded block coding with optimized truncation (EBCOT), embedded zero-tree wavelet (EZW) based methods. Again, the context-based methods like context-based adaptive lossless image codec (CALIC), low-complexity context-based lossless image compression (LOCO-I) (on which the latest of the JPEG standards: JPEG-LS is based), context-based adaptive predictor (CBAP), continuous-tone lossless coding with edge analysis and range amplitude detection (CLARA) etc. have been proving even more useful as their coding is based on the conditional probability distribution resulting in a compact encoding, which is better as compared to traditional encoding schemes like Huffman coding and Arithmetic coding. It is observed that these context based compression techniques can improve the performance of known compression algorithms up to 40% [3]-[6].

Thus, depending upon the type of medical images and considering various parameters like compression ratio (CR) required, complexity of compression algorithm, amount of degradation in the reconstructed image that can be tolerated (based on MSE and PSNR), transmission and reconstruction time etc. a suitable compression technique should be selected from the range of above stated methods to compress a medical image.

1.3 Performance Measuring Parameters

There are several mathematical parameters used to measure the performance of a compression algorithm: the compression ratio (CR), the mean square error (MSE), the peak signal to noise ratio (PSNR) and the correlation coefficient (CoC) to name a few. They are mathematically defined as following:

1. Compression Ratio, $CR = \dfrac{\text{Size of original image in bits}}{\text{Size of compressed image in bits}}$

2. Mean Square Error, $MSE = \dfrac{1}{NM}\sum_{x=0}^{N-1}\sum_{y=0}^{M-1}\left[\left|f(x,y)-\hat{f}(x,y)\right|^2\right]$

Where, $f(x,y)$ is the original pixel value and $\hat{f}(x,y)$ is the compressed pixel value, for an *NxM* input image.

3. Peak Signal to Noise Ratio, $PSNR = 10\log\left[\dfrac{(255)^2}{MSE}\right]$ in dB for an 8-bit image.

4. Correlation Coefficient, $CoC = \dfrac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\hat{f}(x,y)}{\sqrt{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)^2}\sqrt{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \hat{f}(x,y)^2}}$

It suggests how closely the reconstructed image is correlated with an original image, on a scale of 0-1. The nearer the value of CoC to 1 the higher the correlation of a compressed image to an original image is there and vice versa.

1.4 Contextual Compression

In *conventional compression* model, an entire image is compressed equally, i.e. *equal* or *same* level of compression is applied to the useful area as well as to the redundant area of an image. But sometimes, in medical image compression in particular, it is desired to preserve the quality of a particular portion of an image more as compared to the rest of the image. The disadvantage of a conventional compression system is that it will compress the entire image with same compression ratio. Hence we can not get a good overall compression performance in case of a conventional compression model/algorithm. And that is where a newer concept of compression, called the ***Contextual Compression***, arises where the important and unimportant areas of an image are compressed with different compression ratios.

Now, medical images like ultrasound images comprise of: clinically important areas-Region of Interest (RoI), that is used for the diagnosis and unimportant area-Background region, which comprises the patient information and is redundant. The background area in an ultrasound image is quite large, and we can compress it with quite a large compression ratio as it contains the redundant information. Again we can not compress the diagnostically important area (RoI) beyond certain CR, in order to retain quality of the reconstructed image. Hence, the *Contextual compression* aims at compressing the RoI with the ***best*** quality (and least CR) and compressing the background with ***poor*** quality (and highest CR) to attain an ***overall better compression performance.***

1.5 Organization of Thesis

Chapter 1 gives introduction to the need of medical image compression. It briefly gives idea about some of the existing but optimally used medical image compression techniques. The concept of *Contextual Compression* is introduced along with an overview of the entire thesis.

Chapter 2 briefly explains various methods to locate out an important area from an Ultrasound image. It also gives a brief idea about some of the recently developed Region of Interest (RoI) or contextual compression techniques. It starts with the explanation of embedded block coding with optimized truncation (EBCOT) coding. EBCOT-based RoI coding methods namely Maxshift method and Implicit RoI coding method are briefly introduced. Finally, it gives an overview of a proposed contextual compression technique.

Chapter 3 introduces the first of the compression algorithm implemented i.e. the JPEG coding. A detailed discussion on one of the most popular still image compression algorithm with the aid of an example is covered in this chapter. At the end of the chapter results of the JPEG coding method have been discussed. The JPEG coding algorithm has been developed using Matlab 6.5 and so are the codes of rest of the compression algorithms.

Chapter 4 gives a detailed discussion on one of the most widely used medical image compression techniques i.e. the Wavelet transform. The chapter covers the topics like scaling and wavelet functions, multi-resolution analysis (MRA) and the strength of wavelet transform, particularly for image compression. The chapter is concluded with the discussion of the results obtained with the wavelet transform coding.

Chapter 5 gives a brief introduction to the wavelet transform based still image compression standard i.e. JPEG 2000. The chapter covers the block diagram of JPEG 2000 and the distinguishing features of JPEG 2000 as compared to the DCT-based JPEG coding scheme. Finally, at the end the results obtained by JPEG 2000 have been discussed.

Chapter 6 introduces one of the recently developed image compression algorithms, particularly for Medical image compression, i.e. the set partitioning in hierarchical trees (SPIHT) algorithm. The chapter covers the key features of the SPIHT algorithm viz. list of insignificant pixels (LIP), list of significant pixels (LSP), list of insignificant sets (LIS), SPIHT sorting and refinement pass etc. with the help of an example.

Chapter 7 summarizes the work carried out in this thesis. It analytically compares the implemented compression algorithms on the basis of the performance measuring parameters like mean square error (MSE), peak signal to noise ratio (PSNR) and the correlation coefficient (CoC) and with an extensive usage of charts and graphs.

Chapter 8 gives the concluding remarks on the work presented in this thesis. Future scope of improvement and the problems associated are briefly presented for enhanced performance, for upcoming projects, in the same field.

Chapter 2

Contextual Coding Schemes

2.1 Introduction

Region-of-interest (RoI) or *contextual* coding allows for important parts of an image be coded with better quality than the rest of the image (background). In progressive transmission, RoI is transmitted and decoded before the background. RoI coding is therefore capable of delivering high reconstruction quality over user-specified spatial regions in a limited time, compared to compression of the entire image. Further, RoI coding provides an excellent trade-off between image quality and compression ratio. Most RoI coding methods are wavelet-based compression techniques. One range of popular RoI coding schemes are based on SPIHT [9]-[10].

In SPIHT, a parent–child relationship exists among the wavelet coefficients of subbands that have the same spatial location. All the coefficients corresponding to the same spatial region are organized in trees. This tree structure, representing the intersubband dependency, provides excellent rate distortion performance. The encoding process can be terminated as soon as the desired bit rate is achieved. SPIHT based RoI coding schemes allow for the modification of transmission order to place more emphasis on the RoI [10]. On the other hand, several RoI coding methods based on bit plane coding have been proposed. The Maxshift method and general scaling based methods, specified in JPEG2000 Part 1 and Part 2, respectively, scale up (shift) the wavelet coefficients contributing to the RoI reconstruction so that the bits associated with RoI are placed in the bit stream before background [11]-[12].

2.2 Selection of an RoI

Ultrasound scanners produce rectangular images containing a lot of background information. This comprises patient information, and image parameters. The actual image acquired by the probe sensor is of the form of a fan-shaped window, placed over a black rectangle, and centered and aligned with its top border. This is the diagnostically useful image area and it corresponds to only about 60% of the area of the full image in *Fig. 2.1*. Thus before applying any contextual compression technique, it is necessary to model the useful image area by means of Region of Interest approach [7].

All the extra information is added to the black frame for use in diagnosing and archiving the image. Smooth image regions contain high amounts of redundancy, and high compression can be achieved by exploiting it with an efficient algorithm. Consider now the region of the image where the extra information is stored. It contains mainly sharp edges that are expensive to code.

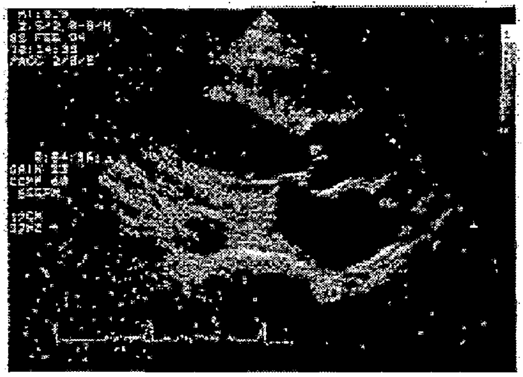


Fig. 2.1 Original ultrasound image (Heart)

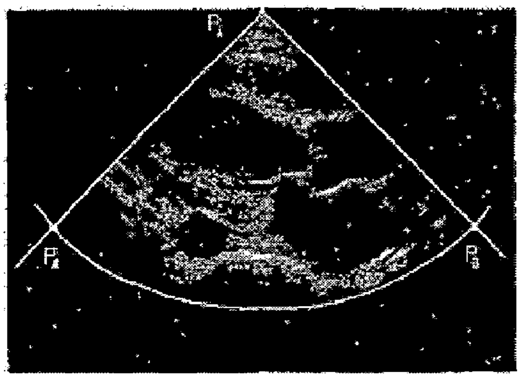


Fig. 2.2 Modeled RoI

They are a serious problem for lossy compression algorithms since they get smoothed and highly visible distortion is introduced. An easy solution to this problem would be the storage of that information in the image file header as overhead information. The original image could then be reconstructed by adding these data after the decompression step.

This approach requires an efficient way of detecting the useful image area that forms our Region of Interest (RoI). This can be achieved by automatically detecting the 3 points P_1 (uppermost), P_2 (leftmost), and P_3 (rightmost), shown in *Fig. 2.2* knowing that P_1 can be found towards the top centre point of the image. The other two are pointed to by bright markers (generated by the scanner) and are symmetrically located left and right of P_1. A raster scan of the original image will detect these points.

2.2.1 Mathematical Approach

The fan-shaped RoI can now be modeled as the sector of a circle centered at P_1 and bounded by two straight lines intersecting at *P1* and crossing the circle at P_2 and P_3 respectively. This can be expressed by the parametric equation of a straight line passing through $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$

$$P(t) = P_1 + t(P_2 - P_1) \tag{2.2-1}$$

and the parametric equation of a circle with centre at $P_1(x_1, y_1)$ and with radius r is:

$$(x - x_1)^2 + (y - y_1)^2 = r^2 \qquad (2.2\text{-}2)$$

The above stated method for locating an RoI is a mathematical approach.

2.2.2 Segmentation Approach

Segmentation is an alternative approach of separating the RoI. Sometimes it is required to locate an important structure like heart valve or any other structure. For this purpose many segmentation techniques like edge-detection or region growing segmentation method are useful. Although both techniques give good results for natural scene images and in general for smooth images, they cannot be directly applied to ultrasound images because they contain large amounts of noise induced to them during the acquisition process and which creates their patchy appearance. Edge detection produces very fine edges around these patches and fails to produce the continuous contours we are looking for. On the other hand side, region growing techniques cannot group together pixels of the same structure, due to large variations in their intensity values. To overcome this problem, the image is preprocessed prior to segmentation. This involves adaptive smoothing. The fundamental idea behind it is to iteratively convolve the image we want to smooth with a very small (3x3) averaging mask whose coefficients reflect, at each point, the degree of continuity of the image intensity function. Two effects can be observed during adaptive smoothing: one is the sharpening of the edges which will eventually become the boundaries of constant intensity regions; the other is the smoothing within each region.

In segmentation based RoI coding based on region grow technique, the aim is to group spatially connected pixels lying within a small dynamic gray level range. The region growing procedure starts with a single pixel, called the seed pixel. Each of the seed's four-connected (neighbor) pixels is checked with a region growing (or inclusion) condition. If the condition is satisfied, the neighbor pixel is included in the region. The four neighbors of the newly added neighbor pixel are then checked for inclusion in the region. This recursive procedure is continued until no spatially connected pixel meets the growing condition. A new region growing procedure is then started with the next pixel of the image which is not already a member of a region; the procedure ends when every pixel in the image has been included in one of the regions grown [32].

After locating the RoI, a suitable compression algorithm is applied in such a way that after reconstructing the image, the quality of an RoI is superior as compared to the background area. In the present work four compression algorithms namely the JPEG coding, the Wavelet transform coding, the JPEG 2000 and the SPIHT algorithm have been discussed and implemented, as a part of contextual compression technique. Their performance analysis has been evaluated on the basis of parameters like MSE, PSNR, and CoC and of course, the human visual system (HVS). As per the contextual compression scheme, the background area is compressed with a poor quality and the image area (RoI) is compressed with a superior quality with the help of above mentioned four compression algorithms.

There are few other approaches too, that are used for contextual compression e.g. Maxshift coding, Implicit coding, EBCOT based coding [8] etc. that are discussed in the next section.

2.3 EBCOT Coding

Scalability for progressive transmission in JPEG2000 is based on the discrete wavelet transform (DWT) and EBCOT. DWT is used to exploit spatial redundancy and to impart resolution scalability. DWTs used in JPEG2000 are the nonreversible wavelet transform for lossy compression and the reversible integer wavelet transform (IWT) for lossy and lossless compression [8]. In this study, we employ the lifting scheme for DWT for the purpose of efficient and low complexity implementation and IWT for lossless RoI reconstruction. The IWT is realized by rounding the output of each lifting filter to an integer. In particular, the reversible (5, 3) filter bank was used so as to achieve lossless coding of RoI. Main operations of EBCOT are described below.

2.3.1 Independent Code block

After DWT, each subband is partitioned into small (e.g. 64 x 64) blocks. The coefficients in a code block are bit plane coded independently from all other code. Instead of coding the entire bitplane in one coding pass, each bit plane is encoded in three coding passes. This produces an embedded bit stream C_j^n with provision for truncating the bit stream at the end of each coding pass. Each truncation point n_i for code-block B_j thus has corresponding length or rate $R_j^{n_i}$. Then we can estimate the weighted mean squared error (MSE) distortion $D_j^{n_i}$ associated with the truncation point n_i,

$$D_j^{n_i} = w_b \sum_{u,v \in B_j} (\hat{a}^{n_i}(u,v) - a(u,v))^2 \tag{2.3-1}$$

Where $a(u,v)$ represents the wavelet coefficient, $\hat{a}^{n_i}(u,v)$ the quantized coefficients associated with the truncation point n_i and w_b the weight for subband b containing B_j. The weight w_b compensates for the effect of biorthogonal wavelet transform. After each code block has been coded, the rate distortion slope is computed, which is defined as

$$S_j^{n_i} = \frac{\Delta D_j^{n_i}}{\Delta R_j^{n_i}} = \frac{D_j^{n_{i-1}} - D_j^{n_i}}{R_j^{n_i} - R_j^{n_{i-1}}} \tag{2.3-2}$$

The slope is a measure of the reduction in distortion for the increase in bit rate resulting from the inclusion of the coding pass. As we shall see later, the rate distortion slope plays an important role in minimizing the reconstructed image distortion at the desired bit rate.

2.3.2 Rate Control

One important issue in any image coding system is rate control (rate distortion optimization). EBCOT provides an efficient mechanism for rate control, which is referred to as post-compression rate distortion optimization. Once all the code blocks have been compressed, and the rates and slopes of coding passes in each code block are available, a post-processing operation passes over all the compressed blocks. And the number of coding passes is determined for each code block that would be included in the final bit stream to achieve the desired bit rate. Choosing the number of coding passes after entropy coding corresponds to the selection of truncation points from each code block bit stream, which essentially has the same effect as applying quantization before the entropy coding. The rate control problem is to find the truncation point for each code block that can minimize the total distortion (sum of distortion contributions from each code block), given a total bit budget for the compressed bit stream. Intuitively, if the code block has larger rate distortion slope, it contributes more coding passes to the embedded bit stream in that larger distortion reduction is achieved earlier.

2.3.3 Quality layers

To support progressive transmission, EBCOT introduces the mechanism of quality layers abstraction. Each quality layer is a collection of coding-pass compressed data from the code blocks. Coding passes that contain the most important data (in the sense of reducing distortion the most) are included in the initial layer, whereas coding passes pertaining to finer detail in the image are included in subsequent layers. During decoding, the reconstructed image quality improves incrementally with each successive layer processed. Assume the embedded bit stream consists of M layers, generated by N code blocks. Then, the ith layer can be expressed as

$$L_i = C_1^{n_{1i}} \cup C_2^{n_{2i}} \cup \ldots \cup C_N^{n_{Ni}} \tag{2.3-3}$$

The first layer L_0, or the lowest quality layer, provides the basic image quality. The subsequent layers, when added to the lower layers, generate higher quality rendition of the original image. As an example, *Fig. 2.3* illustrates the quality layer abstraction of EBCOT.

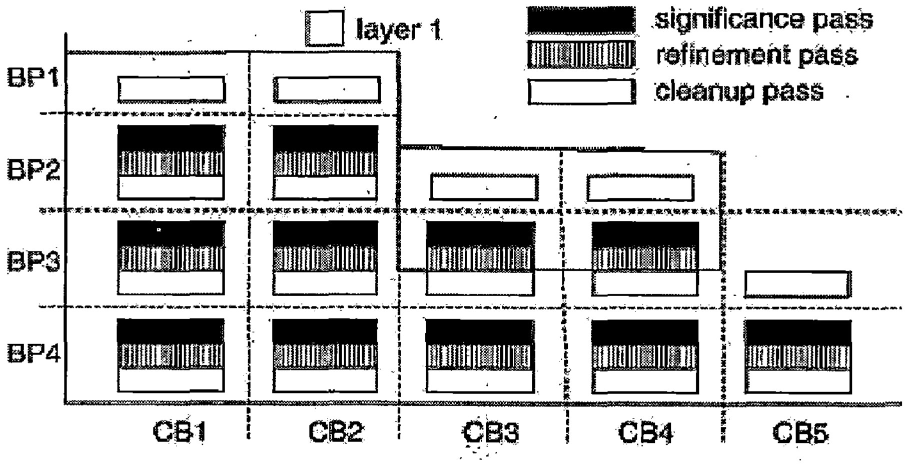


Fig. 2.3: Quality layer abstraction in EBCOT

It contains two layers and is composed of five code blocks from CB1 to CB5, which are bit plane coded in the order from the most significant bit plane BP1 to the least significant bit plane BP4. Each bit plane consists of three coding passes, except the first one. If the compressed bit stream is organized in layer progressive manner, all the coding pass contributions to layer 1 appear first, followed by the contributions to layer 2, and so on.

2.4 EBCOT based Coding Methods

When a quality progressive bit stream is transmitted to a client, the image quality of RoI is expected to improve more rapidly than the background. This is achieved with emphasis on RoI by identifying the wavelet coefficients needed for reconstruction of the spatial region of interest, and encoding them with higher priority. To identify RoI coefficients, an RoI mask is generated by tracing the inverse wavelet transform backwards. *Fig. 2.4* depicts an RoI mask for two-level wavelet decomposition. Detailed calculation of the RoI mask can be found in [11]. It is observed in *Fig. 2.4* that each RoI code block may contain RoI coefficients and background coefficients. Some of EBCOT based RoI coding techniques have been discussed below, in brief.

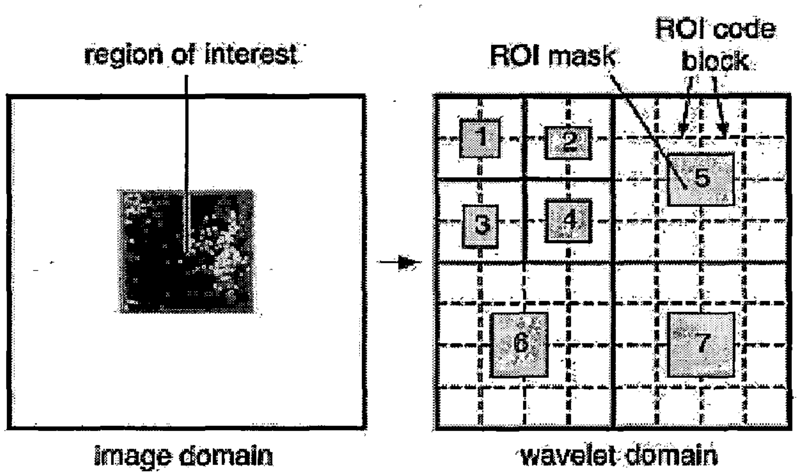


Fig. 2.4: RoI region and RoI mask

2.4.1 Maxshift Method

In the Maxshift method, prior to bitplane coding, bitplanes of the RoI coefficients are scaled up by the desired amount so that coefficients associated with the RoI are placed in higher bit planes. If the scaling value is s, the wavelet coefficient $a'(u, v)$ supplied to the bit plane entropy coder is

$$a'(u,v) = \begin{cases} a(u,v), & \mathrm{M(u,v)=0} \\ 2^s a(u,v), & \mathrm{M(u,v)=1} \end{cases} \qquad (2.4\text{-}1)$$

The scaling factor is selected to ensure there is no overlap between background and ROI bit planes, as depicted in *Fig. 2.5*. When the entropy coder encodes the code block containing RoI coefficients, the encoded RoI bits appear before the background bits. Then rate control builds a layer progressive bit stream in which information pertaining to RoI precedes that of the background. The main strength of Maxshift is its fast RoI reconstruction. It also lifts the restriction on RoI shape. Drawbacks of the Maxshift method include the increase in coding time, and background information is received only after full RoI reconstruction.

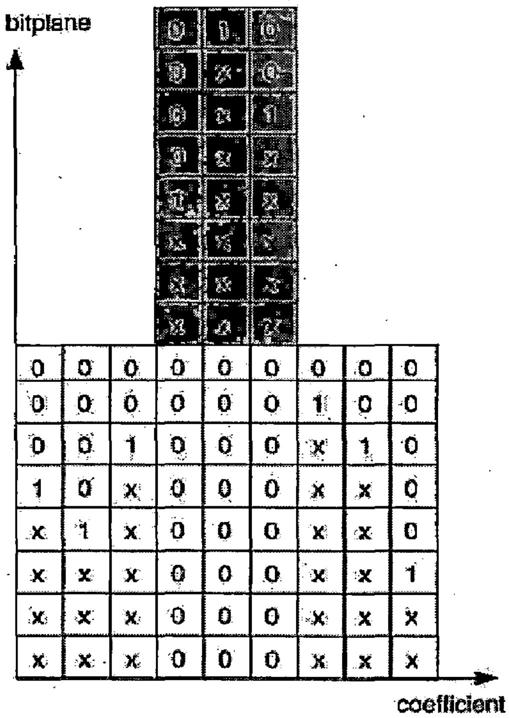


Fig. 2.5: Representation of Maxshift method

2.4.2 Implicit RoI Coding

In contrast to the Maxshift method, the implicit RoI encoding method is designed to take full advantage of EBCOT and achieves RoI emphasis in the bit stream ordering process. In EBCOT, each quality layer comprises an arbitrary contribution from the embedded bit stream of each code block of each subband. Thus RoI emphasis is possible by including relatively larger contributions from code blocks involved in the RoI reconstruction to quality layers. The implicit RoI encoding operates as follows. When given a spatial RoI, the relevant code blocks are identified and their contributions are sequenced into the embedded bit stream in a manner that effectively augments the priority of the RoI. Since EBCOT allocates the code block contributions in terms of the overall distortion minimization, the implicit RoI encoding allocates code block contributions in accordance with distortion reduction and the region of interest. As a result, the distortion measure becomes

$$D_j^{n_i} = \begin{cases} W_{ROI} w_b \sum_{u,v \in B_j} (\hat{a}^{n_i}(u,v) - a(u,v))^2, & \text{RoI code-book} \\ w_b \sum_{u,v \in B_j} (\hat{a}^{n_i}(u,v) - a(u,v))^2, & \text{otherwise} \end{cases} \tag{2.4-2}$$

Where, W_{ROI} is the weighting factor. RoI code blocks are those contain RoI coefficients.

Hence the coding passes in the RoI code block have substantially larger rate distortion slopes. And RoI code blocks contribute more to the embedded bit stream.

The main advantage of the implicit RoI encoding is its low complexity. The method itself is straightforward and easy to implement. More than that, no bitplane scaling is involved at either the encoding or the decoding side. However, the implicit RoI encoding has the disadvantage of slow RoI reconstruction. This is because the priority arrangement is made on a block-by-block basis and some RoI code blocks may contain a large amount of background information. When using Eqn. 2.4-1 to calculate the distortion, these background coefficients in RoI code blocks are assigned the same priority as RoI coefficients, and are coded in tandem.

2.5 Proposed Contextual Compression Method

In the proposed method, the diagnostically important region (RoI) and the unimportant area (background) are separated out [14]. To separate out the RoI and background the Image processing toolbox of Matlab 6.5 has been used. Particularly the functions like *roipoly* and *immultiply* have been used after performing several steps of pre-processing of an input ultrasound image. *Fig. 2.6* gives the block diagram of a proposed method.

Fig. 2.7 (a) shows an original abdominal Ultrasound image. The separated RoI and background from this image are shown in *Fig. 2.7 (b)* and *(c)* respectively. For lesser complexity purpose the entire image area has been selected as an RoI. Depending upon the type of image and need of a medico the shape and size of an RoI may vary. As has been suggested in [15]-[18] more than one RoI can also be selected and most important region can be compressed with the best quality.

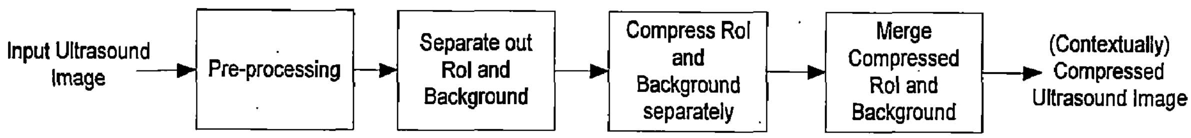


Fig. 2.6: Block diagram of a proposed contextual compression method

After separating the image area (RoI) and redundant area (background), a suitable compression algorithm is applied. The RoI is compressed with lower compression ratio to have a superior quality whereas the background area is compressed with higher compression ratio and hence poor quality.

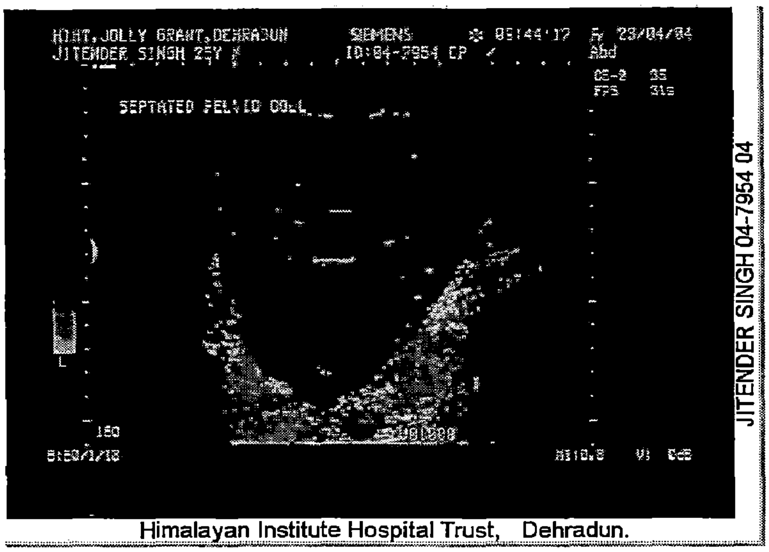
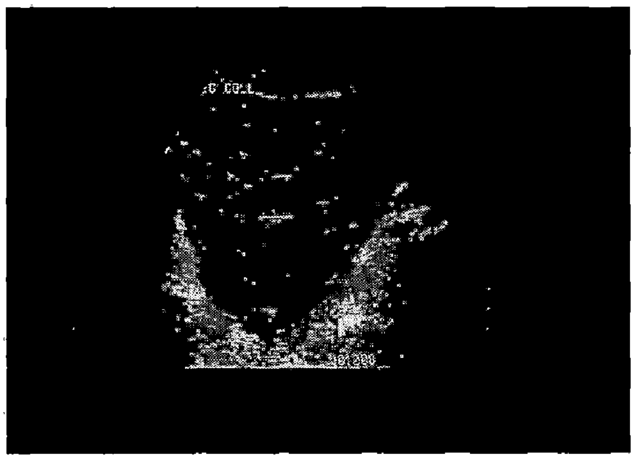


Fig. 2.7 (a): Original ultrasound image



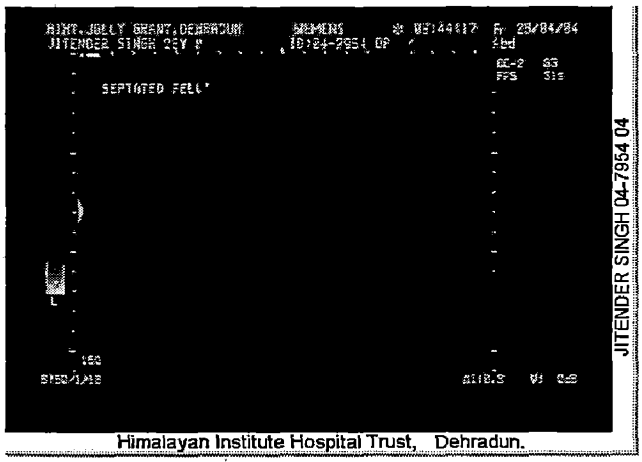


Fig. 2.7 (b): Separated image area (RoI)   **Fig. 2.7 (C): Separated background**

The compressed versions of RoI and background are then merged. In the present work four compression algorithms namely the JPEG coding, the Wavelet transform, the JPEG 2000 coding and SPIHT have been applied and their results have been compared. These algorithms have been discussed in the chapters following. The performance measure of these compression algorithms is checked with the known parameters like mean square error (MSE), peak signal to noise ratio (PSNR) and correlation coefficient (CoC) and of course, the visual quality of the compressed image using the human visual system (HVS).

Chapter 3

The JPEG Coding

3.1 Introduction

The Joint Photographic Experts Group (JPEG) is responsible for standardizing this technique for handling grayscale and full color image files. While the standard provides for lossy and lossless implementations, the most significant result of this work was the advance in compression ratios for image files with little or no visible degradation of the input image. The JPEG standard first transforms image data into the frequency domain using the discrete cosine transform (DCT), which is a lossless process. The data is then quantized, to the degree specified by the user, and then compressed for storage.

In the JPEG image compression algorithm, the input image is divided into 8-by-8 or 16-by-16 blocks, and the two-dimensional DCT is computed for each block. The DCT coefficients are then quantized, coded, and transmitted. The JPEG receiver (or JPEG file reader) decodes the quantized DCT coefficients, computes the inverse two-dimensional DCT of each block, and then puts the blocks back together into a single image. For typical images, many of the DCT coefficients have values close to zero; these coefficients can be discarded without seriously affecting the quality of the reconstructed image. A block diagram shown in *Fig. 3.1* illustrates the process involved in the JPEG compression scheme [13].

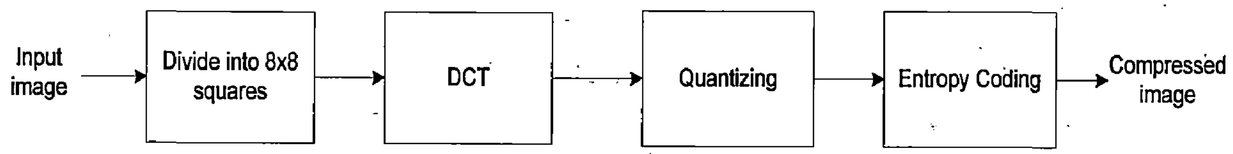


Fig. 3.1: Block diagram for JPEG coding system

The input image is decomposed into typically 8x8 blocks, and the DCT is performed by way of matrix multiplication. The resulting matrix is then processed in a quantization stage in which a user specified quality factor, usually between 1 and 100, is utilized to define quantum steps. It is in this stage that loss occurs as high frequency components of the matrix are in effect zeroed out. This stage is the significant component of this compression technique in that the user has the trade-off decision between file size and data loss. The greater the quality factor the greater the compression ratios and image degradation. Finally the quantized matrix is encoded. This encoding is a lossless procedure and utilizes any of the run length encoding methods like Huffman coding.

3.2 JPEG Coding Schemes

JPEG includes two classes of encoding and decoding processes:

- DCT-based Lossy process
- Prediction-based Lossless process

JPEG includes four modes of operation

- Sequential DCT-based mode
- Progressive DCT-based mode
- Lossless mode
- Hierarchical mode

3.2.1 Sequential DCT-based Mode

It is a baseline algorithm (heart) of JPEG coding standard. In this process an image is first partitioned into blocks of *8x8* pixels. The blocks are then processed from left to right, top to bottom. Then 2-D forward DCT of each block is taken which are then quantized. The quantized DCT coefficients are entropy coded and the compressed image is available for storage or transmission.

3.2.2 Progressive DCT-based Mode

It is similar to the sequential DCT-based mode, up to quantization stage. The quantized coefficients are first stored in a buffer. All DCT coefficients in buffer are then encoded by a multiple scanning process. In each scan, quantized DCT coefficients partially encoded either by spectral selection or successive approximation. In spectral selection, quantized DCT coefficients are divided into multiple spectral bands according to the zigzag order. In each scan, a specified band is encoded. In successive approximation, a specified number of most significant bits of quantized coefficients first encoded. In subsequent scans, less significant bits are encoded.

3.2.3 Hierarchical Mode

An image is first spatially down-sampled to a multiple layered pyramid. This sequence of frames is encoded by predictive coding. Except for the first frame, the encoding process is applied to the differential frames. Hierarchical coding mode provides a progressive presentation similar to progressive DCT-based mode but is useful in applications that have multi-resolution requirements. Hierarchical mode also provides the capability of progressive coding to a final lossless stage.

The JPEG lossless mode is nothing but Differential Pulse Code Modulation (DPCM) in spatial domain.

3.3 A Baseline JPEG Coding Example

Consider compression and reconstruction of the following 8x8 subimage with the JPEG baseline standard [13].

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	55	55	61	65	83
87	79	69	68	65	76	78	94

The original image consists of 256 or 2^8 possible gray levels, so the coding process begins by level shifting the pixels original subimage by -2^7 or -128 gray levels. The resulting shifted array is

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-62	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	- 24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-65	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

Which, when transformed according to the forward DCT Eqs. Pair (3.2-1) and (3.2-2) for $N=8$, becomes

$$T(u, v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} \frac{1}{N^2} f(x, y)\, g(x, y, u, v) \tag{3.3-1}$$

Where, $g(x, y, u, v) = \alpha(u)\,\alpha(v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$ (3.3-2)

$$\text{For } \alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u = 1,2,\ldots,N-1 \end{cases}$$

-415	-29	-62	25	55	-20	-1	3
7	-21	-62	9	11	-7	-6	6
-46	8	77	-25	-30	10	7	-5
-50	13	35	-15	-9	6	0	3
11	-8	-13	-2	-1	1	-4	1
-10	1	3	-3	-1	0	2	-1
-4	-1	2	-1	2	-3	1	-2
-1	-1	-1	-2	-1	-1	0	-1

If the JPEG recommended normalization array is used to quantize the transformed array, the scaled and truncated coefficients are

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Where, for instance, the DC coefficient is computed as

$$\hat{T} = round\left[\frac{T(0,0)}{Z(0,0)}\right]$$

$$= round\left[\frac{-415}{16}\right] = -26$$

After normalization and quantization step, the next stage is the coding stage. With the help of JPEG coefficient coding categories, JPEG default DC and AC code (luminance) tables a proper coding stream is generated. These streams are then encoded using any run-length coding algorithm like Huffman coding.

To decompress a JPEG compressed subimage, the decoder must first recreate the normalized transform coefficients that led to compressed bit stream. The next stage is to denormalize the coefficients using Eqn. $T(0,0) = \hat{T}(0,0)Z(0,0)$. The complete reconstructed image is obtained by taking the inverse DCT of the normalized array. And finally, level shifting each inverse transformed pixel by $+2^7$ (or +128) yields,

58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

Any differences between the original and the reconstructed subimage are a result of the lossy nature of the JPEG compression and decompression process. In this example, the errors range from -14 to +11 and are described below:

-6	-9	-6	2	11	-1	-6	-5
7	4	-1	1	11	-3	-5	3
2	9	-2	-6	-3	-12	-14	9
-6	7	0	-4	-5	-9	-7	1
-7	8	4	-1	11	4	3	-2
3	8	4	-4	2	11	1	1
2	2	5	-1	-6	0	-2	5
-6	-2	2	6	-4	-4	-6	10

The root mean square error (RMSE) of the overall compression and reconstruction process is approximately 5.9 gray levels. Hence the mean square error (MSE) would be 34.81.

The blocking artifacts of the JPEG technique are visible only in enlarged images, but cause problems in many image processing applications. Particular problems occur when analyzing algorithms are applied to JPEG compressed images or when the images are re-rastered for printing. When a compressed image is re-compressed (i.e. with a higher compression rate or after some processing) these additional edges further degrade the quality of the image.

3.4 Results and Discussions

The JPEG baseline coding algorithm has been implemented using Matlab 6.5 and the image processing tool box. The RoI and background are both compressed with different quality factors i.e. with different compression and not to mention that background area has been compressed with much higher compression ratio. The overall compression ratio for RoI-JPEG method ranges from 20.0 to 51.60 and corresponding bpp values are 0.40 and 0.155 respectively. These values have been selected in order to have uniformity with other compression algorithms viz. Wavelet transform, JPEG 2000 and SPIHT.

An original US image is shown in *Fig. 3.2*. The corresponding JPEG contextually compressed image is shown in *Fig. 3.3*. In this case compression ratio and PSNR are 32.0 (bpp of 0.25) and 26.86 dB, respectively.

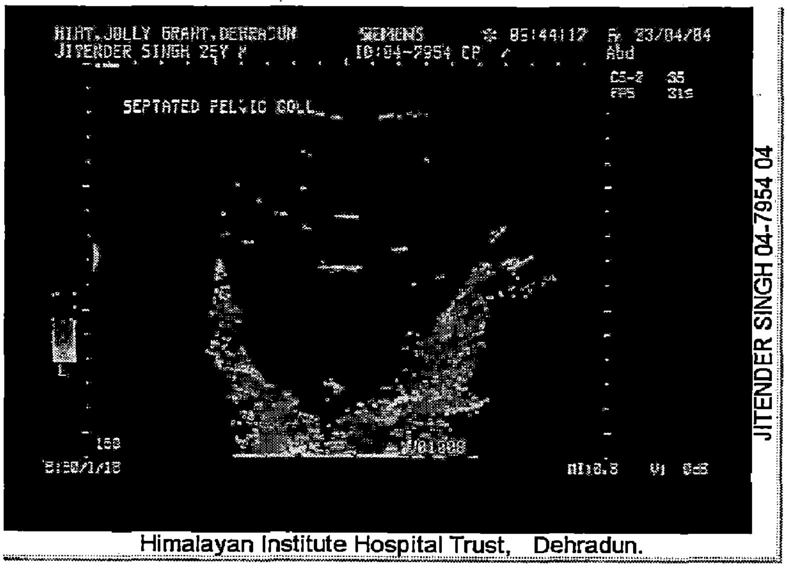


Fig. 3.2: Original US image (IMAGE 1)

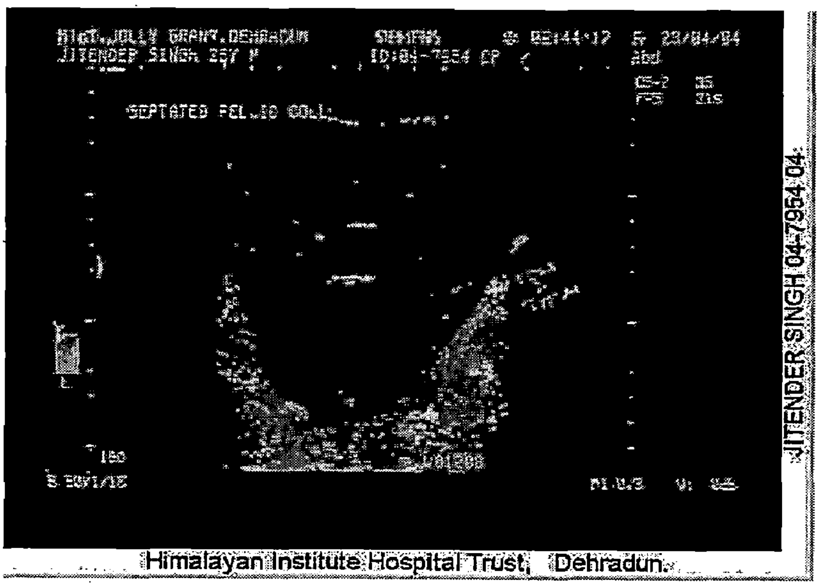


Fig. 3.3: JPEG (contextually) compressed image with CR=32.0 (0.25 bpp)

By inspecting Tables 3.1 and 3.2, it is revealed that the least MSE corresponding to bpp of 0.40 is 78.56 and the PSNR for the same is 29.18 dB; and the same for RoI (image area) are 36.56 and 32.50 dB respectively. This suggests that the image area has been compressed with much quality-preserving manner which is the prime requirement of the contextual compression technique. The highest MSE and corresponding lowest PSNR corresponding to bpp of 0.155 for the image area (RoI) are 49.62 and 31.17 dB respectively which are much better as compared to 327.05 and 22.98 dB of entire image. Another important parameter for compression performance is the CoC. The CoC values for RoI are better that those for entire image, which is a proof of superior image quality of RoI.

Table 3.1: JPEG parameters for the entire image

Sr. No.	CR	bpp	MSE	PSNR, dB	Correlation Coefficient
1.	20.0	0.40	78.56	29.18	0.9954
2.	32.0	0.25	133.75	26.86	0.9895
3.	40.0	0.20	210.69	24.89	0.9834
4.	44.38	0.18	246.95	24.21	0.9806
5.	51.60	0.155	327.05	22.98	0.9746

Table 3.2: JPEG parameters for the image area (RoI)

Sr. No.	CR	bpp	MSE	PSNR, dB	Correlation Coefficient
1.	20.0	0.40	36.56	32.50	0.9827
2.	32.0	0.25	41.59	31.94	0.9803
3.	40.0	0.20	44.06	31.69	0.9792
4.	44.38	0.18	47.69	31.34	0.9775
5.	51.60	0.155	49.62	31.17	0.9765

Fig. 3.4 and *3.5* below show the graphical comparisons of MSE v/s bpp and PSNR v/s bpp of entire image and RoI, respectively for the original US image shown in *Fig. 3.2*. It can be seen that the MSE for entire image increases almost exponentially, the highest value being 327.05; where as that for RoI it remains almost constant and the highest value is just under 50. Similarly, in case of PSNR also there is a sharp decrement for entire image, where as it is almost constant for RoI. These results speak about the superiority of RoI parameters, which is the main aim of contextual compression.

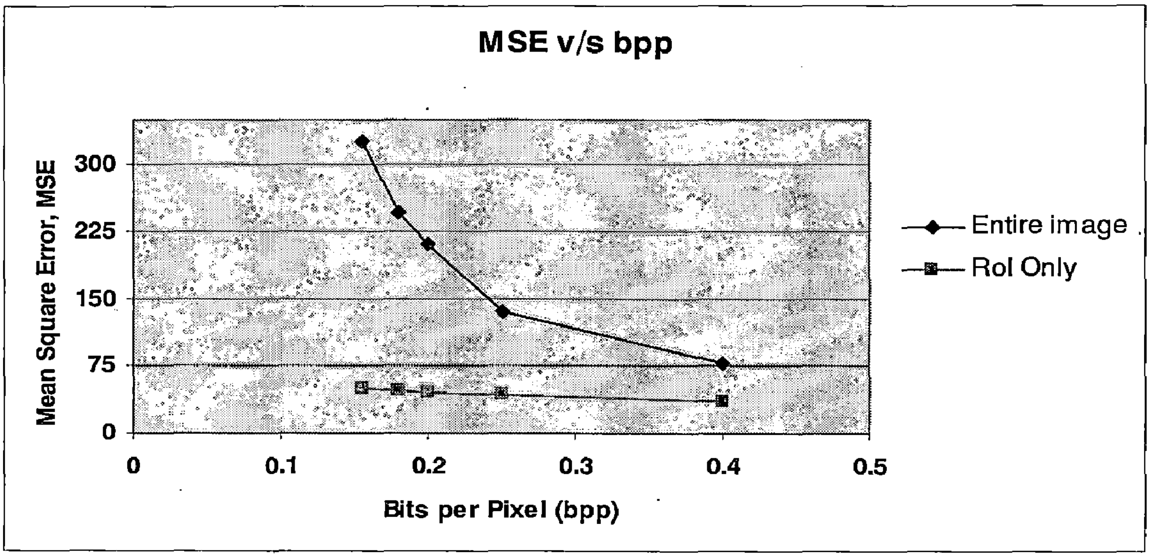


Fig. 3.4: MSE comparison for JPEG

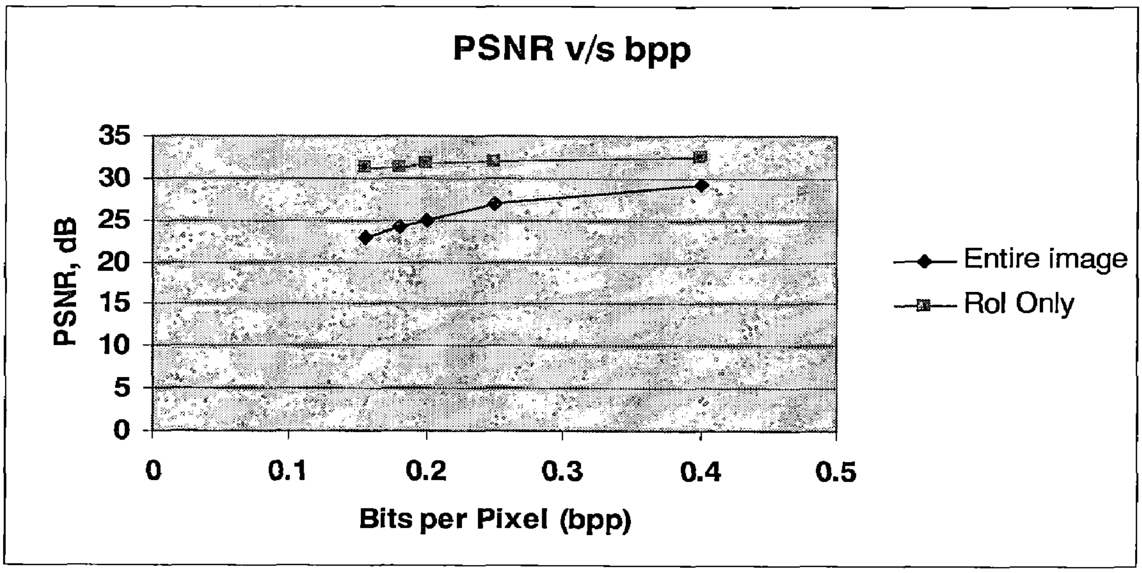


Fig. 3.5: PSNR comparison for JPEG

Chapter 4
The Wavelet Transform Coding

4.1 Introduction

The Wavelet Transform is a mathematical tool used to represent signals (or images) in a domain in which it may be manipulated more effectively. It shares many properties and is similar to the well known Fourier transform. Both the Fourier and Wavelet Transforms are based on "basis functions". Unlike the Fourier transform, whose basis functions are sinusoids, ***Wavelet Transforms*** are based on small waves, called ***Wavelets,*** of varying frequency and limited duration.

Often signals we wish to process are in the time-domain, but in order to process them more easily other information, such as frequency, is required. Mathematical transforms translate the information of signals into different representations. For example, the Fourier transform converts a signal between the time and frequency domains, such that the frequencies of a signal can be seen. However the Fourier transform cannot provide information on which frequencies occur at specific times in the signal as time and frequency are viewed independently. To solve this problem the short term Fourier transform (STFT) introduced the idea of windows through which different parts of a signal are viewed (Refer *Fig. 4.1* and *Fig. 4.2*). For a given window in time the frequencies can be viewed. However Heisenberg's Uncertainty Principle states that as the resolution of the signal improves in the time domain, by zooming on different sections, the frequency resolution gets worse. Ideally, a method of multiresolution is needed, which allows certain parts of the signal to be resolved well in time, and other parts to be resolved well in frequency. The power and magic of wavelet analysis is exactly this multiresolution.

Wavelet analysis can be used to divide the information of an image into approximation and detail subsignals. The approximation subsignal shows the general trend of pixel values, and three detail subsignals show the vertical, horizontal and diagonal details or changes in the image. If these details are very small then they can be set to zero without significantly changing the image. The value below which details are considered small enough to be set to zero is known as the threshold. The greater the number of zeros the greater the compression that can be achieved.

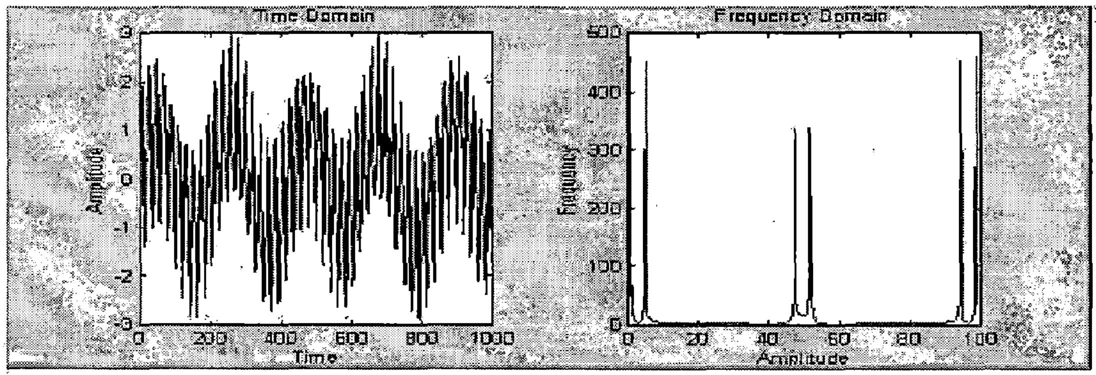


Fig. 4.1: Fourier transform- time domain and frequency domain

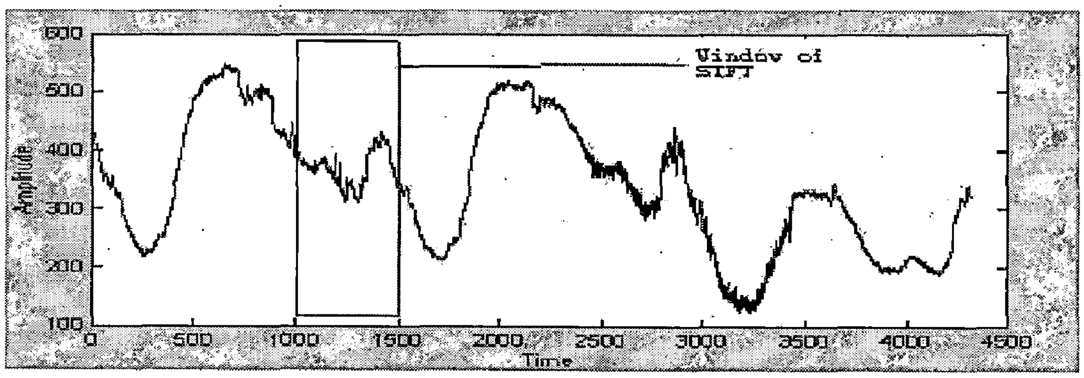


Fig. 4.2: Window of an STFT

In short, in Fourier analysis a signal is broken up into sine and cosine waves of different frequencies, and it effectively re-writes a signal in terms of different sine and cosine waves. Wavelet analysis does a similar thing, it takes a 'mother wavelet', then the signal is translated into shifted and scale versions of this mother wavelet.

4.2 MultiResolution Analysis (MRA)

The *wavelets* are considered to be the foundation of a powerful new approach to signal processing and analysis called *multiresolution* theory. As its name implies, multiresolution theory is concerned with the representation and analysis of signals (or images) at more than one resolution. The appeal of such an approach is obvious- features that might go undetected at one resolution may be easy to spot at another. Multiresolution theory incorporates and unifies techniques from a variety of disciplines, including subband coding from signal processing, quadrature mirror filtering from digital speech recognition and pyramidal image processing.

4.2.1 Scaling Functions

In MRA, a *scaling function* is used to create a series of approximations of a function or image, each differing by a factor of 2 from its nearest neighboring approximations. The *wavelets* are then used to encode the difference in information between adjacent approximations. Consider a set of expansion functions composed of integer translations and binary scalings of the real, square-integrable function $\varphi(x)$; that is, the set $\{\varphi_{j,k}(x)\}$ where

$$\varphi_{j,k}(x) = 2^{j/2}\,\varphi(2^j x - k) \tag{4.2-1}$$

for all $j, k \in Z$ and $\varphi(x) \in L^2(\mathbf{R})$. Here, k determines the position of $\varphi_{j,k}(x)$ along the x-axis, j determines $\varphi_{j,k}(x)$'s width- how broad or narrow it is along the x-axis. Because the shape of $\varphi_{j,k}(x)$ changes with j, $\varphi(x)$ is called a *scaling function*. By choosing $\varphi(x)$ wisely, $\{\varphi_{j,k}(x)\}$ can be made to span $L^2(\mathbf{R})$, the set of measurable, square-integrable functions.

4.2.2 Wavelet Functions

Given a scaling function that meets the MRA requirements, we can define a *wavelet function* $\psi(x)$ that, together with its integer translates and binary scalings spans the difference between any two adjacent scaling subspaces V_j and V_{j+1}. We define the set $\{\psi_{j,k}(x)\}$ of wavelets

$$\psi_{j,k}(x) = 2^{j/2}\,\psi(2^j x - k) \tag{4.2-2}$$

4.3 Wavelet Transforms in One Dimension

The *continuous wavelet transform*, the *discrete wavelet transform* and the *wavelet series expansion* are closely related wavelet transformations. Their counterparts in the Fourier domain are, the integral Fourier transform, the discrete Fourier transform and the Fourier series expansion, respectively. A computationally efficient implementation of the discrete wavelet transform is the *fast wavelet transform.*

4.3.1 The Continuous Wavelet Transform

Mathematically, the process of Fourier analysis is represented by the *Fourier transform*:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)\, e^{-j\omega t}\, dt \tag{4.3-1}$$

Which is the sum over all time of the signal $f(t)$ multiplied by a complex exponential. The results of the transform are the F*ourier coefficients,* which when multiplied by a sinusoid of frequency yield the constituent sinusoidal components of the original signal.

Similarly, the *continuous wavelet transform* (CWT) is defined as the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet function, . . The results of the CWT are many *wavelet coefficients* C, which are a function of scale and position. Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelets of the original signal, as shown in *Fig. 4.3*.

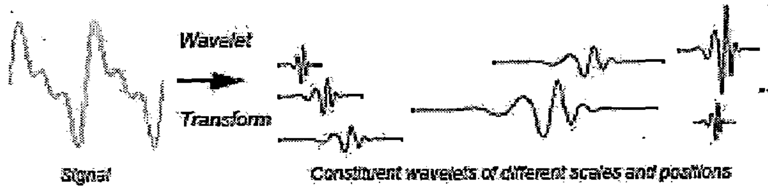


Fig. 4.3: The continuous wavelet transform

The continuous wavelet transform of a continuous, square-integrable function, $f(x)$ relative to a real-valued wavelet, $\psi(x)$, is

$$W_{\psi}(s,\tau) = \int_{-\infty}^{\infty} f(x)\psi_{s,\tau}(x)\,dx \tag{4.3-2}$$

Where

$$\psi_{s,\tau}(x) = \frac{1}{\sqrt{S}}\psi\left(\frac{x-\tau}{s}\right) \tag{4.3-3}$$

and s and τ are called *scale* and *translation* parameters, respectively. Given $W_{\psi}(s,\tau)$, $f(x)$ can be obtained using the *inverse continuous wavelet transform*

$$f(x) = \frac{1}{C_{\psi}}\int_{0}^{\infty}\int_{-\infty}^{\infty} W_{\psi}(s,\tau)\frac{\psi_{s,\tau}(x)}{s^2}\,d\tau\,ds \tag{4.3-4}$$

where

$$C_{\psi} = \int_{-\infty}^{\infty}\frac{|\Psi(u)|^2}{|u|}\,du \tag{4.3-5}$$

and $\Psi(u)$ is the Fourier transform of $\psi(x)$.

The results of the CWT are many *wavelet coefficients* C, which are a function of scale and position. Multiplying each coefficient by the appropriately scaled and shifted wavelet yields the constituent wavelets of the original signal.

4.3.2 The Discrete Wavelet Transform

Calculating wavelet coefficients at every possible scale is a fair amount of work, and it generates an awful lot of data. What if we choose only a subset of scales and positions at which to make our calculations? It turns out, rather remarkably, that if we choose scales and positions based on powers of two — so called *dyadic* scales and positions — then our analysis will be much more efficient and just as accurate. If the function being expanded is a sequence of numbers, like samples of a continuous function $f(x)$, the resulting coefficients are called the *discrete wavelet transform* (DWT) of $f(x)$. Hence DWT pair is given as

$$W_\varphi(j_0,k) = \frac{1}{\sqrt{M}} \sum_x f(x)\varphi_{j_0,k}(x) \tag{4.3-6}$$

$$W_\psi(j,k) = \frac{1}{\sqrt{M}} \sum_x f(x)\psi_{j,k}(x) \tag{4.3-7}$$

for $j \geq j_0$ and

$$f(x) = \frac{1}{\sqrt{M}} \sum_x W_\varphi(j_0,k)\varphi_{j_0,k}(x) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_k W_\psi(j,k)\psi_{j,k}(x) \tag{4.3-8}$$

Here $f(x)$, $\varphi_{j_0,k}(x)$ and $\psi_{j,k}(x)$ are functions of the discrete variable x = 0, 1, 2,..., M-1.

4.3.3 The 2-D DWT and Multiscale Image Analysis

There exist various extensions of the one-dimensional wavelet transform to higher dimensions. The extension given by Mallat for two-dimensions is described below. In two-dimensional wavelet analysis one introduces, like in the one-dimensional case, a scaling function), $\varphi(x,y)$ such that:

$$\varphi(x,y) = \varphi(x)\varphi(y) \tag{4.3-9}$$

Where, $\varphi(x)$ is a one-dimensional scaling function.

Let $\psi(x)$ be the one-dimensional wavelet associated with the scaling function. Then, the three two-dimensional wavelets are defined as:

$$\begin{aligned} \psi^H(x,y) &= \varphi(x)\psi(y) \\ \psi^V(x,y) &= \psi(x)\varphi(y) \\ \psi^D(x,y) &= \psi(x)\psi(y) \end{aligned} \tag{4.3-10}$$

where, H, V, and D stand for "horizontal", "vertical" and "diagonal" respectively.

Fig. 4.4 shows the two-dimensional wavelet transform of an image. The two-dimensional MRA decomposition is completed in two steps. First, using $\varphi(x)$ and $\psi(x)$ in the x direction, *f(x, y)* (an image) is decomposed into two parts, a smooth approximation and a detail. Next, the two parts are analyzed in the same way using $\varphi(y)$ and $\psi(y)$ in the y direction. As a result, four channel outputs are produced, one channel is A_1 *f(x, y)*, the level one smooth approximation of $f(x, y)$, through $\varphi(x)\varphi(y)$ processing, the other three channels are $D^{(H)}_1 f(x, y)$, $D^{(V)}_1 f(x, y)$ and $D^{(D)}_1 f(x, y)$, the details of the image. Level two results are obtained after decomposing $A_1 f(x, y)$ progressively. The algorithm of two-dimensional wavelet decomposition and reconstruction is the same as that for one-dimensional wavelet. Mallat pyramid scheme is also used in this case. *Fig. 4.5* is a sketch map of a one, two and three-level two-dimensional wavelet transform and *Fig. 4.6* gives the diagrammatic representation of 2-D DWT.

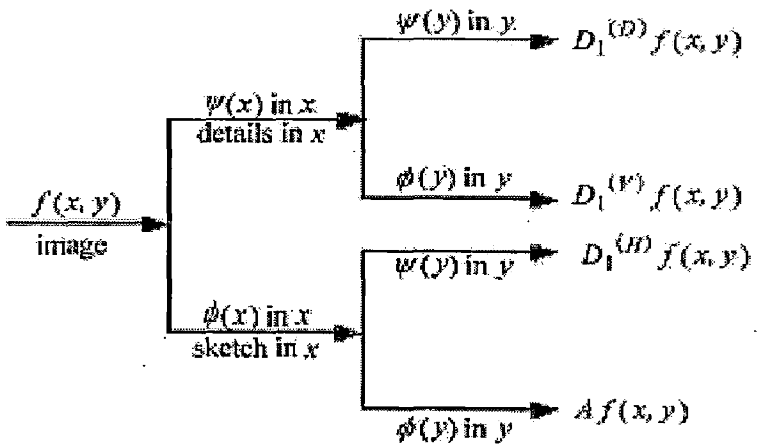


Fig. 4.4: Image decomposition with separable two-dimensional wavelets

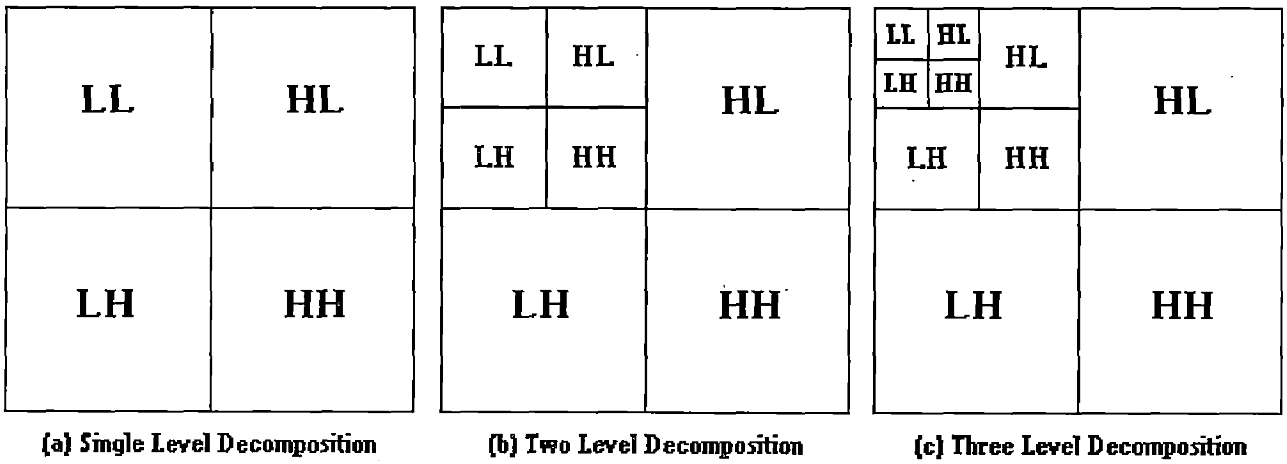


Fig. 4.5: Multi-level decomposition of an image

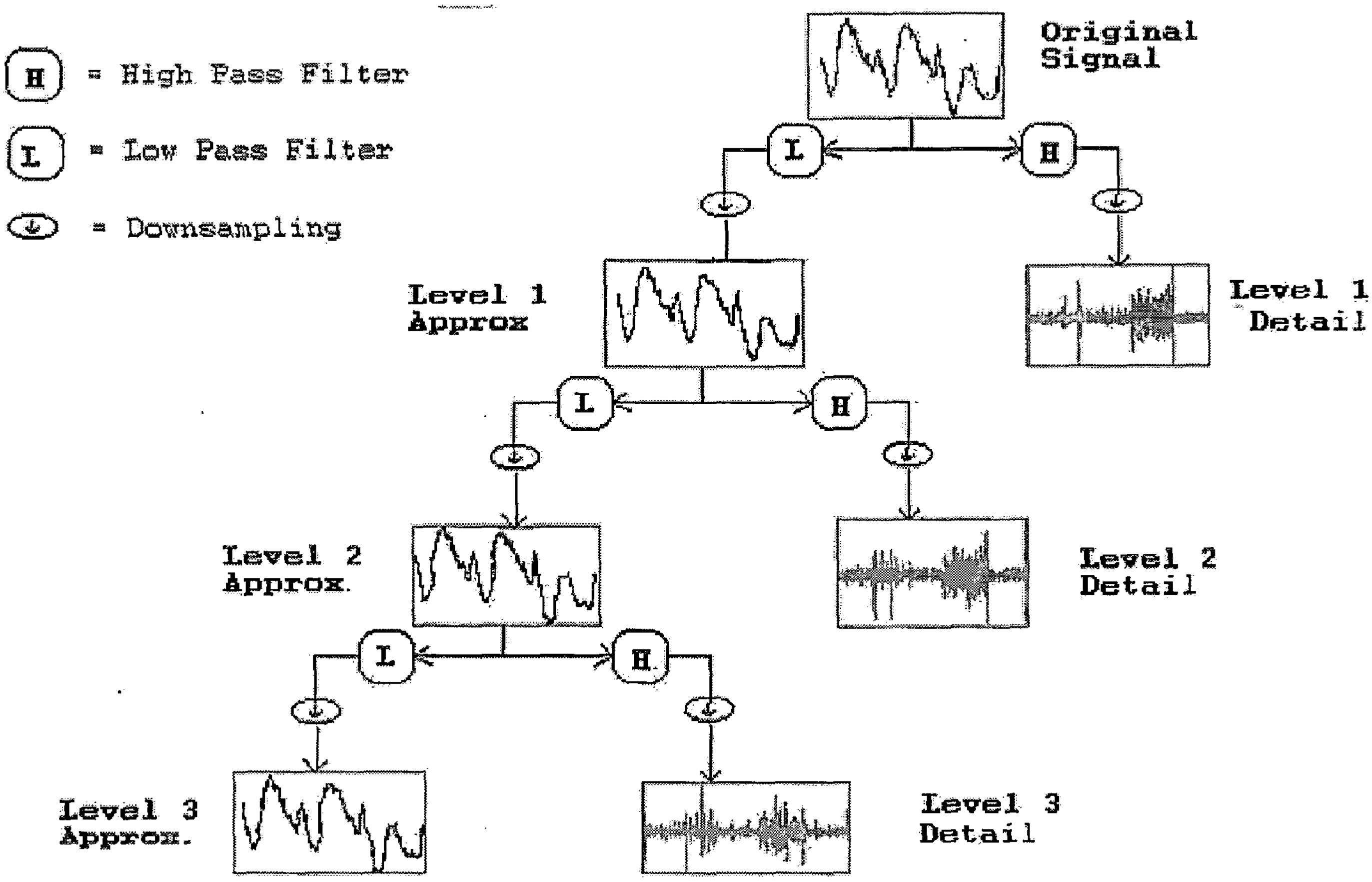


Fig. 4.6: Diagrammatic representation of 2-D DWT

4.4 Wavelet Coding

Like the transform coding techniques, the wavelet coding is based on the idea that the coefficients of a transform that decorrelates the pixels of an image can be coded more efficiently than the original pixels themselves. If the transform's basis function- wavelets- pack most of the important visual information into a small number of coefficients, the remaining coefficients can be quantized coarsely or truncated to zero with little image distortion.

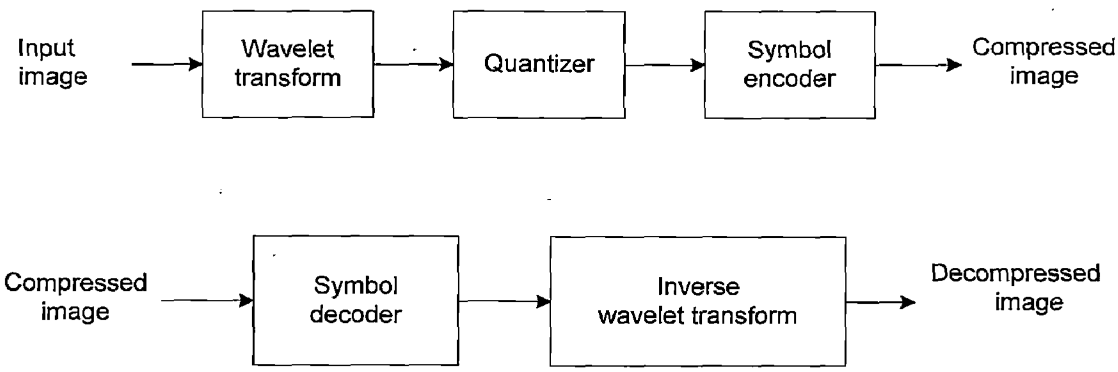


Fig. 4.7: A Wavelet Coding System

The following steps are performed to compress an image using wavelet transform:

1. Digitize the source image into a signal s, which is a string of numbers.
2. Decompose the signal into a sequence of wavelet coefficients w.
3. Use thresholding to modify the wavelet coefficients from w to another sequence w'.
4. Use quantization to convert w' to a sequence q.
5. Apply entropy coding to compress q into a sequence e.

4.4.1 Digitization

The first step in the wavelet compression process is to digitize the image. The digitized image can be characterized by its intensity levels, or scales of gray which range from 0 (black) to 255 (white), and its resolution, or how many pixels per square inch.

4.4.2 Thresholding

In certain signals, many of the wavelet coefficients are close or equal to zero. Through a method called thresholding, these coefficients may be modified so that the sequence of wavelet coefficients contains long strings of zeros. Through a type of compression known as entropy coding, these long strings may be stored and sent electronically in much less space. There are different types of thresholding.

- Hard thresholding
- Soft thresholding

In hard thresholding, a tolerance is selected. Any wavelet whose absolute value falls below the tolerance is set to zero with the goal to introduce many zeros without losing a great amount of detail. There is not a straightforward easy way to choose the threshold, although the larger the threshold that is chosen the more error that is introduced into the process.

Another type of thresholding is soft thresholding. Once again a tolerance, h, is selected. If the absolute value of an entry is less than the tolerance, than that entry is set to zero. All other entries, d, are replaced with sign (d) $||d| - h|$. Soft thresholding can be thought of as a translation of the signal toward zero by the amount h.

4.4.3 Entropy coding

Wavelets and thresholding help process the signal, but until this point, no compression has yet occurred. One method to compress the data is Huffman entropy coding. With this method, and integer sequence, q, is changed into a shorter sequence, e, with the numbers in e being 8 bit integers. The conversion is made by an entropy coding table. Entropy coding is designed so that the numbers that are expected to appear the most often in q, need the least amount of space in e.

4.4.4 Quantization

The fourth step of the process, known as quantization, converts a sequence of floating numbers w' to a sequence of integers q. The simplest form is to round to the nearest integer. Another option is to multiply each number in w' by a constant k, and then round to the nearest integer. Quantization is a lossy process because it introduces error into the process, since the conversion of w' to q is not a one-to-one function.

4.5 Results and Discussions

The Wavelet transform coding has been performed using the Matlab 6.5, Wavelet toolbox in particular. The three-level decomposition algorithm and the 'bior6.8' type of (mother) wavelet have been used to perform the compression algorithm. As has been observed in [18], the region-based wavelet compression emphasizes on the exact retrieval of RoI and this aspect has been taken care of.

An original US image is shown in *Fig. 4.8.* The corresponding contextually Wavelet compressed image is shown in *Fig. 4.9.* In this case compression ratio and PSNR are 51.6 (bpp of 0.155) and 27.06 dB, respectively.

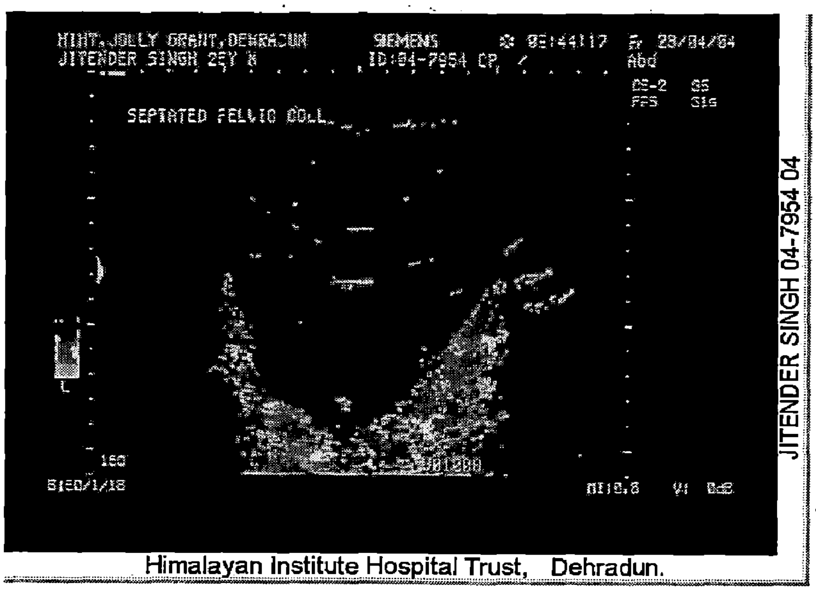


Fig. 4.8: Original US image (IMAGE 1)

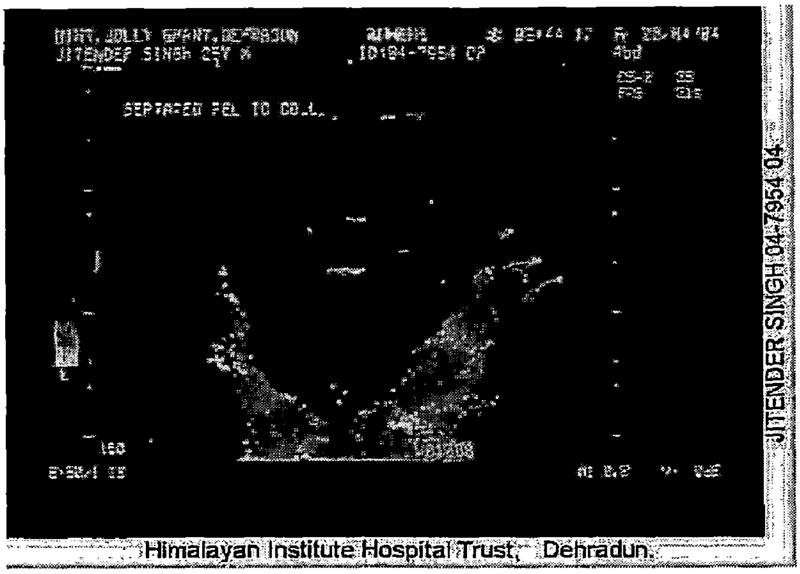


Fig. 4.9: Wavelet (contextually) compressed image with CR=51.6 (bpp=0.155)

By inspecting Tables 4.1 and 4.2, it is revealed that the least MSE corresponding to bpp of 0.40 is 81.64 and the PSNR for the same is 29.01 dB; and the same for RoI (image area) are 10.02 and 38.12 dB respectively. The highest MSE and corresponding lowest PSNR corresponding to bpp of 0.155 for the image area (RoI) are 17.93 and 35.59 dB respectively which are much better as compared to 128.02 and 27.96 dB of entire image. This suggests that the image area has been compressed with much quality-preserving manner which the prime requirement of the contextual compression technique. This fact is consolidated by the better CoC values of RoI as compared to the entire image.

Table 4.1: Wavelet parameters for the entire image

Sr. No.	CR	bpp	MSE	PSNR, dB	Correlation Coefficient
1.	20.0	0.40	81.64	29.01	0.9938
2.	32.0	0.25	95.82	28.32	0.9927
3.	40.0	0.20	118.98	27.38	0.9909
4.	44.38	0.18	124.25	27.19	0.9905
5.	51.60	0.155	128.02	27.06	0.9902

Table 4.2: Wavelet parameters for the image area (RoI)

Sr. No.	CR	bpp	MSE	PSNR, dB	Correlation Coefficient
1.	20.0	0.40	10.02	38.12	0.9954
2.	32.0	0.25	11.77	37.42	0.9944
3.	40.0	0.20	13.59	36.80	0.9935
4.	44.38	0.18	16.57	35.93	0.9921
5.	51.60	0.155	17.93	35.59	0.9915

Fig. 4.10 and Fig. *4.11* below show the graphical comparisons of MSE and PSNR values for entire image and RoI, respectively for the original US image shown in *Fig. 4.8*. It has been observed that the MSE values fall sharply for entire image, whereas for RoI they are almost constant and very low. The PSNR values of both entire image as well as of RoI do not fall sharply and are almost constant, the reason being the excellent energy retention capabilities of wavelet coding system even for higher compression ratios (and lower bit rates).

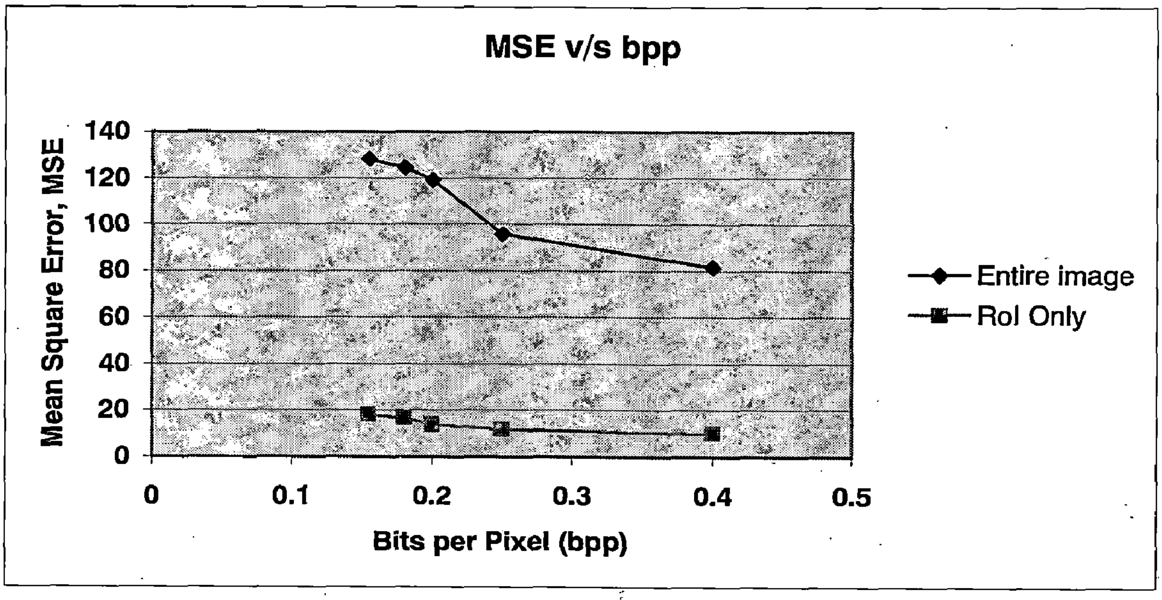


Fig. 4.10: MSE comparison for wavelet

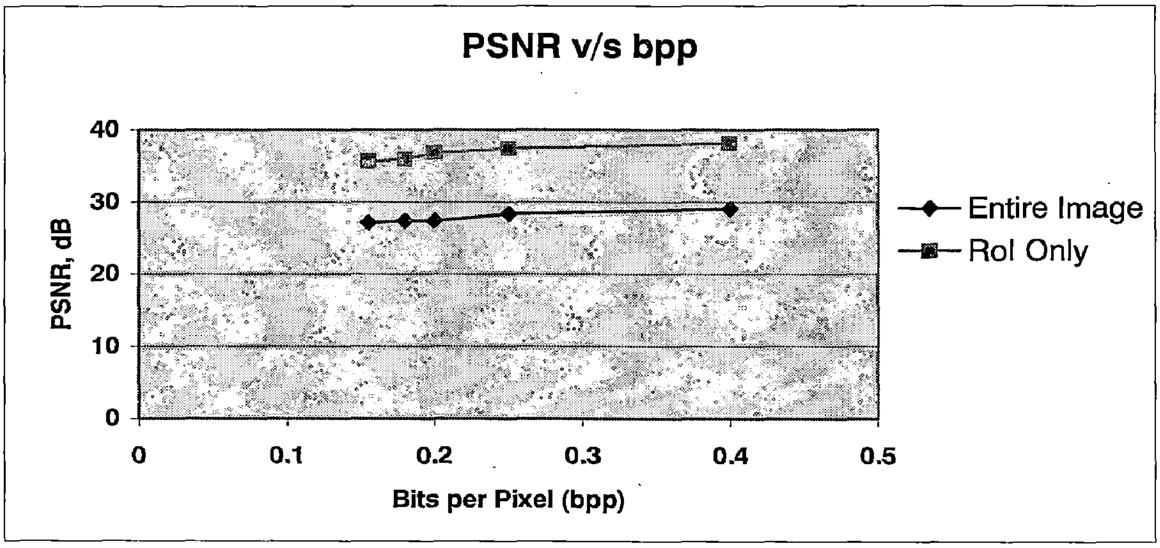


Fig. 4.11: PSNR comparison for wavelet

Chapter 5

The JPEG 2000 Coding

5.1 Introduction

JPEG 2000 is a new image compression standard being developed by the JPEG and ISO. Although not yet formally adopted, JPEG 2000 extends the initial JPEG standard to provide increased flexibility in both the compression of the continuous tone images and access to the compressed data. For example, portions of a JPEG 2000 compressed image can be extracted for retransmission, storage, display, and/or editing. The standard is based on the **Wavelet transform coding** technique. Coefficient quantization is adapted to individual scales and subbands and the quantized coefficients are arithmetically coded on a bit-plane basis [13].

JPEG 2000 is designed for different types of still images (bi-level, gray-level, color, multicomponent) allowing different imaging models (client/server, real-time transmission, image library archival, limited buffer and bandwidth resources, etc), within a unified system. JPEG2000 is intended to provide low bit rate operation with rate-distortion and subjective image quality performance superior to existing standards, without sacrificing performance at other points in the rate-distortion spectrum.

JPEG2000 addresses areas where current standards fail to produce the best quality of performance, such as:

- Low bit rate compression performance (rates below 0.25 bpp for highly-detailed gray-level images)
- Lossless and lossy compression in a single codestream
- Seamless quality and resolution scalability, without having to download the entire file. The major benefit is the conservation of bandwidth
- Large images: JPEG is restricted to 64kx64k images (without tiling). JPEG2000 will handle image sizes up to (2^32 - 1)
- Single decompression architecture
- Error resilience for transmission in noisy environments, such as wireless and the Internet

- Region of Interest coding
- Improved compression techniques to accommodate richer content and higher resolutions

5.2 Motivation for JPEG 2000

Despite the success of JPEG in the 1990s, a growing number of new applications such as high-resolution imagery, digital libraries, high-fidelity color imaging, multimedia and Internet applications, wireless, medical imaging, etc., require additional, enhanced functionalities from a compression standard that JPEG cannot satisfy due to some of its inherent shortcomings and design points that were beyond the scope of JPEG when it was developed in the previous decade. The shortcomings of JPEG can be seen in a number of areas: distortion and artifacts, ineffective handling of high-quality images, poor compression for lossless images, lack of effective color-space support, and lack of resolution scaling. Distortion and artifacts introduced by JPEG in compressing large images, especially at high compression rates, manifest in the well known blocking or tiling artifacts where each 8-by-8-pixel region developed well-defined edges, and ringing artifacts where small waves appear next to sharp edges in the image. The limitation of JPEG's 64-kilopixel sample size and the limitation to either 8- or 12-b samples have proven to be too restrictive for many new imaging applications such as medical and high-resolution imagery. The lack of effective color-space support in JPEG severely hinders its adoption in prepress and other graphics arts applications where consistent color information from image capture to editing, display, or printing has to be tightly managed. Finally, poor compression for lossless images is also seen as a limitation because the lossless mode in JPEG is accomplished by a completely different method than the lossy mode, and moving from one to the other requires completely decoding and recoding the image [19].

At the time of the completion of the JPEG series of image compression standard (ISO/IEC 10 918-1 to 4) in the early 1990s, the JPEG committee added a new project that aimed for the lossless and near-lossless compression of continuous- tone still images to address the inadequate performance of JPEG. The result of that activity was the publication of the ISO/IEC 14 495 ITU-T T.87 standard, called JPEG-LS, which offered a new improved method for coding photographic images without loss, or with well-defined distortion, which is very useful in medical imaging and others where many edits may be expected of an original image. Out of the many submissions for JPEG-LS technology, there was a wavelet-based technology submitted by Ricoh called CREW that provided, in a single compressed bit-stream, a continuous lossless to lossy compression scheme. Even though CREW was not adopted as the technology for JPEG—LS, the interesting set of features that CREW illustrated was very much appreciated by the committee. Subsequently a *new work item* for a new compression system (later called JPEG 2000) was established, thus kick-starting the JPEG 2000 project.

5.3 Organization of JPEG 2000 Standard

The JPEG 2000 family is organized into 13 parts, as shown in Table 5.1. The first 6 parts were all published as International Standards by 2004. JPEG 2000 Part 1, the core coding system, defines the core coder and is used in other parts of the standard. It defines a core coding system that is aimed at minimal complexity while satisfying 80% of the applications. Part 1 specifies a simple file format, JP2, which meets the needs of most (but not all) standard applications. JP2 provides a basic level of interoperability for applications that wish to interchange JPEG 2000 data. Applications that choose to implement JP2 must conform to all of the relevant provisions in the standard in order to be able to claim compliance. JPEG 2000 Part 2, the extensions, extends the core Part 1 decoder by introducing a range of advanced features including extended file format support and improved compression efficiency using more complex algorithms. It is intended to serve those applications where maximal interchange is less important than meeting specific requirements.

JPEG 2000 Part 3 defines Motion JPEG 2000 (MJP2) and is primarily based on the technology in Part 1 with the addition of a file format. It results in an encoder that is significantly less complex than the MPEG family of standards (because motion estimation is not used) and provides full random access to the individually coded frames. It is intended for applications such as digital sill cameras with burst capture mode, video editing in postproduction environments, and digital cinema archive and distribution. JPEG 2000 Part 4 defines conformance testing to ensure high-quality implementations of the standard. JPEG 2000 Part 5 defines two reference software implementations for Part 1. One is a Java implementation by the JJ2000 group. The other is a C implementation called JasPer. JPEG 2000 Part 6 defines a compound image file format for document scanning and raster graphics. The main feature of Part 6 is the ability to combine different image layers using transparency masks and layouts. This handles the compression of documents where scanned images are first segmented into homogeneous parts and then individually compressed and transmitted together with mask information indicating how to combine the individual parts. The first six parts of JPEG 2000 form the major portion of the development work by the JPEG committee during the period 1997–2001. In 2002, the committee introduced four new parts, Parts 8–11, to address new applications areas.

- Part 8—JPSEC is concerned with security in JPEG 2000 applications.
- Part 9—JPIP defines an advanced set of network protocol for distributed application involving JPEG 2000 contents.

Table 5.1: Parts of JPEG 2000 standard family

Part	Title
1	Core Coding System
2	Extensions
3	Motion JPEG 2000
4	Conformance Testing
5	Reference Software
6	Compound Image File Format
7	Cancelled (not used)
8	JSPEC: Secure JPEG 2000
9	JPIP: Interactivity Tools, APIs and Protocols
10	JP3D: 3-D and Floating Point Data
11	JPWL: Wireless
12	ISO Base Media File Format
13	An Entry Level JPEG 2000 Encoder

- Part 10—JP3D is concerned with compression of three–dimensional (3-D) and floating point data.
- Part 11—JPWL deals with wireless applications using JPEG 2000.

These four new parts are under development by the JPEG committee, with Part 9—JPIP expected to reach International Standard status by the end of 2004.

Part 12, which was formerly an amendment of Part 3, is a common activity with the MPEG committee working toward a common file format with MPEG-4. Part 13 is a most recently established new part (March 2004) with the aim of standardization of an entry-level JPEG 2000 encoder [19].

5.4 Overview of JPEG 2000 Technology

The key incentive behind the development of JPEG 2000 was not just to provide higher compression performance compared to JPEG, but to provide a new image coding system with a rich set of features, all supported within the same compressed bit-stream. We first highlight the key features of JPEG 2000 and follow with a summary the key technologies in JPEG 2000 that made these features possible. Details of various aspects of JPEG 2000 technology can be found in [20]-[24].

5.4.1 JPEG 2000 Features

JPEG 2000 Part 1 addresses the shortcomings of JPEG by supporting the following set of features, all within single, tightly integrated compression architecture and code-stream syntax.

- Superior compression performance: At high bit rates, where artifacts become just imperceptible, JPEG 2000 has a compression advantage over JPEG by roughly 20% on average. At lower bit rates, JPEG 2000 has a much more significant advantage over certain modes of JPEG. The compression gains over JPEG are attributed to the use of DWT and more sophisticated entropy encoding scheme.
- Multiple resolution representation: JPEG 2000 provides seamless compression of image components each from 1 to 16 bits per component sample. With tiling, it handles arbitrary large image size in one single codestream.
- Progressive transmission by pixel and resolution accuracy (progressive decoding and signal-to-noise ratio (SNR) scalability): JPEG 2000 provides efficient codestream organizations which are progressive by pixel accuracy or by quality (SNR) and also by resolution or size.
- Lossless and lossy compression: JPEG 2000 provides both lossless and lossy compression from single compression architecture with the use of a reversible (integer) wavelet transform.
- Random codestream access and processing (region of interest): JPEG 2000 codestreams offer several mechanisms to support spatial random access or region of interest access at carrying degrees of granularity.

- Error resilience (robust to bit errors): JPEG 2000 is robust to bit errors introduced by noisy communication channels such as wireless. This is accomplished by the inclusion of resynchronization markers, the coding of data in relatively small independent blocks, and the provision of mechanisms to detect and conceal errors within each block.
- Sequential buildup capability: JPEG 2000 allows for encoding of an image from top to bottom in a sequential fashion without the need to buffer an entire image.
- Flexible file format: the JP2 and JPX file formats allow for handling of color-space information, metadata, and for interactivity in networked applications as developed in the JPEG Part 9—JPIP protocol.

5.4.2 JPEG 2000 Technology

Fig. 5.1 illustrates the basic building blocks of JPEG 2000: a preprocessing step which typically consists of tiling, dc level shifting and multicomponent transform (used for RGB color input), then a DWT, followed by a quantizer (for lossy compression only), then an entropy coder, and, finally, a bit stream organization step to prepare the final codestream of the compressed image.

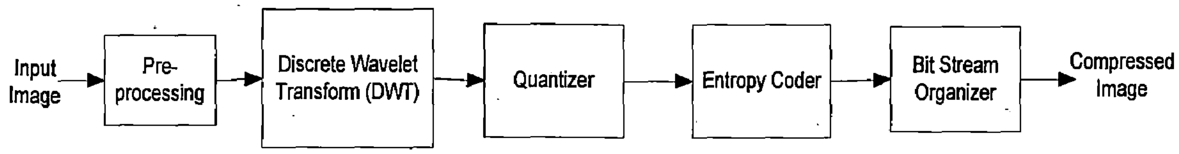


Fig. 5.1: JPEG 2000 building blocks

1) Preprocessing: *Tiling, DC Level Shifting, and Multicomponent Transform:* The image samples are mapped as regularly spaced points on a high-resolution reference grid. The reference grid is divided up into regular rectangular, nonoverlapping tiles of equal size (except for those tiles at image borders). The tile size is arbitrary and can be as large as the original image itself. Each tile is compressed independently using its own set of specified compression parameters. Unsigned image samples are dc level shifted so that their range is centered on zero. Signed samples are not level shifted. The multicomponent transform is a linear transformation that maps RGB to YCbCr color spaces.

2) DWT: The DWT is applied independently to the image components and decorrelates the image into different scale sizes, preserving much of its spatial correlation. A one-dimensional (1-D) DWT consists of a low (L) and high (H) pass filter splitting a line of pixels into two lines of half the size. Application of the filters to two-dimensional (2-D) images in horizontal and vertical directions produces four subbands (LL, LH, HL, and HH). The LL subband is a lower resolution representation of the original image, and the missing details are filtered into the remaining subbands. The subbands contain the horizontal (LH), vertical (HL), and diagonal (HH) edges on the scale size defined by the wavelet. Each of these subbands can be decomposed further by subsequent wavelet transforms. JPEG 2000 core coding system specifies a choice of two wavelet filters: the Daubechies 9/7 or the integer Daubechies 5/3. The 9/7 filter (given as floating point numbers) is primarily used for high visual quality compression. The shorter integer 5/3 filter can be implemented in integer arithmetic and the associated DWT is reversible, enabling the lossless compression.

3) Codeblock and Precinct Partitioning: Each tile-component- resolution level is partitioned into codeblocks and precincts, in a similar manner to the partitioning of the original image into tiles. The codeblock is the unit within which bit-plane coding is done. Precincts are made up from codeblocks and are a higher level partitioning used for construction of the data packets that make up the final codestream.

4) Quantization: Before bit-plane coding, the transform coefficients within each subband are quantized. JPEG 2000 core system adopts a simple linear dead zone quantizer where the dead zone has exactly twice the step size of the remaining quantizing steps so as to obtain an optimal embedded structure to achieve SNR scalability.

5) Entropy Coding: *Bit-Plane Coding:* To create embedded bit-streams for the compressed image data, bit-plane encoding of the quantized wavelet coefficients is used.

A unique feature of JPEG 2000 in the approach to bit-plane that makes it different from the usual approach of exploiting the correlation between subbands is to encode each subband independently of other subbands. Moreover, a block coding paradigm is used where each subband is partitioned into small rectangular blocks, called codeblocks, and each codeblock is encoded independently. This approach, known as embedded block coding with optimized truncation (EBCOT), offers many advantages, including localized random access into the image, parallelization, improved cropping and rotation functionality, improved error resilience, efficient rate control, and maximum flexibility in arranging progression orders. By not exploiting the intersubband redundancies, it may seem that coding efficiency is affected. However, this is more than compensated by the finer scalability results from multiple-pass encoding of the codeblock bit-planes. By using an efficient rate control strategy that independently optimizes the contribution of each codeblock to the final bit stream, the JPEG 2000 Part 1 encoder achieves a compression efficiency that is superior to other existing approaches.

6) Bit Stream Organization: JPEG 2000 offers significant flexibility in the organization of the compressed bit stream to enable such features as random access, region of interest coding and scalability. The data representing a specific tile, layer, component, resolution, and precinct appears in the codestream in a continuous segment called a packet.

5.5 Results and Discussions

The JPEG 2000 algorithm has been implemented using Matlab 6.5. The indigenous functions like *wavefilter, wavefast, wavecut, wavecopy, wavepaste and waveback* have been developed for various purposes like wavelet decomposition and reconstruction, multilevel 2-D forward and inverse wavelet transform and zeroing coefficients in wavelet decomposition structure. These functions have been used as and when required during the encoding and decoding process of JPEG 2000 algorithm. The provision for either explicit or implicit quantization has been kept. Instead of Arithmetic coding the well known Huffman coding algorithm has been used for the simplicity purpose.

Fig. 5.2 and *5.3* show the original and JPEG 2000 contextually compressed US images. The compression ratio and the PSNR of compressed image are 64.0 (bpp of 0.125) and 23.41dB, respectively.

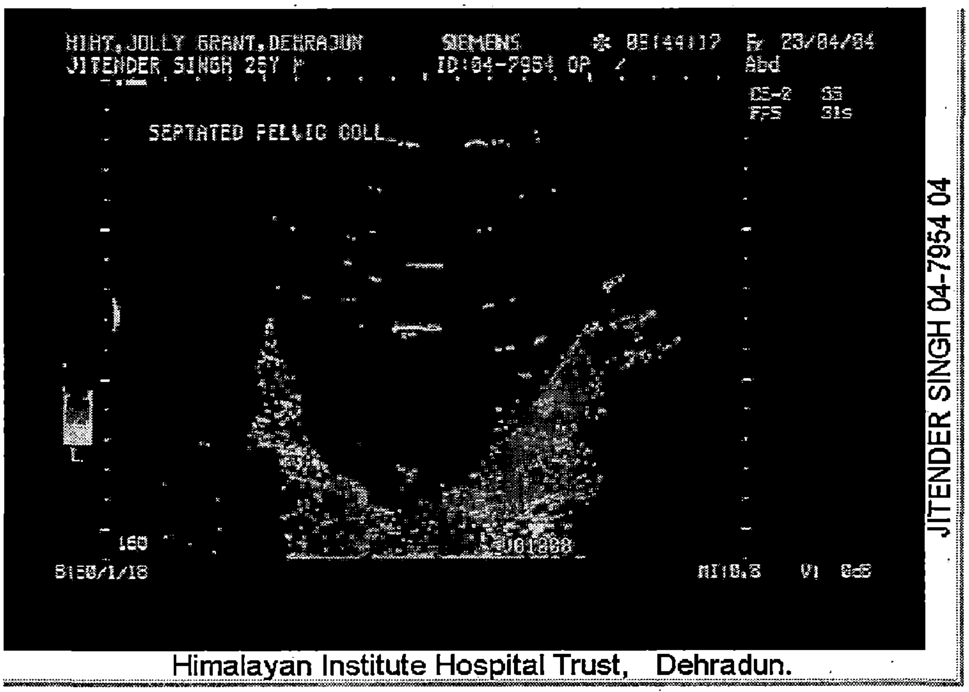


Fig. 5.2: Original US image (IMAGE 1)

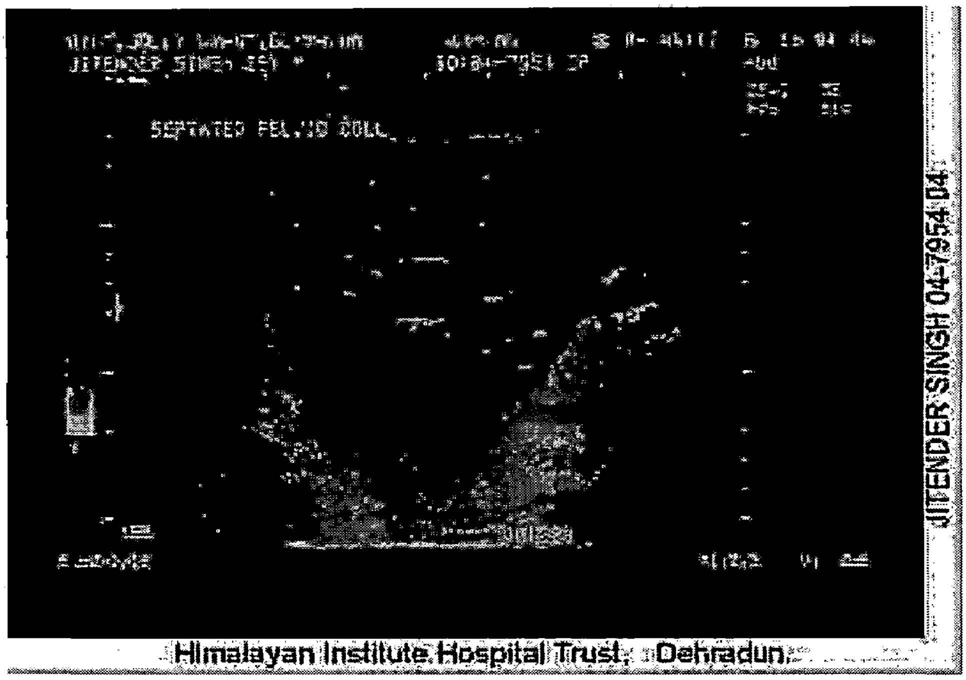


Fig. 5.3: JPEG 2000 (contextually) compressed image with CR=64.0 (bpp = 0.125)

By inspecting Tables 5.2 and 5.3, it is revealed that the least MSE corresponding to bpp of 0.40 is 63.39 and the PSNR for the same is 30.11 dB; and the same for RoI (image area) are 13.29 and 36.89 dB, respectively. The highest MSE and corresponding lowest PSNR corresponding to bpp of 0.125 for the image area (RoI) are 28.04 and 33.65 dB, respectively which are much better as compared to 296.21 and 23.41 dB of entire image. This is a yardstick of the quality preserved in RoI, which is the prime requirement of the contextual compression technique.

Table 5.2: JPEG 2000 parameters for entire image

Sr. No.	CR	bpp	MSE	PSNR, dB	Correlation Coefficient
1.	20.0	0.40	63.39	30.11	0.9951
2.	32.0	0.25	97.75	28.23	0.9924
3.	40.0	0.20	159.11	26.11	0.9876
4.	44.38	0.18	233.78	24.44	0.9817
5.	51.60	0.155	277.52	23.69	0.9782
6.	64.0	0.125	296.21	23.41	0.9767

Table 5.3: JPEG 2000 parameters for image area (RoI)

Sr. No.	CR	bpp	MSE	PSNR, dB	Correlation Coefficient
1.	20.0	0.40	13.29	36.89	0.9968
2.	32.0	0.25	20.66	34.87	0.9942
3.	40.0	0.20	22.62	34.58	0.9892
4.	44.38	0.18	22.97	34.52	0.9891
5.	51.60	0.155	23.52	34.42	0.9888
6.	64.0	0.125	28.04	33.65	0.9866

Further, the JPEG 2000 could achieve the highest CR of 64.0 (bpp of 0.125). At this CR the PSNR and CoC obtained for the entire image are 23.41 dB and 0.9767 , respectively. The same parameters for image area (RoI) are 33.65 dB and 0.9866, respectively, which are considered as quite remarkable and signify the usefulness of JPEG 2000 at lower bit rates (i.e. bpp below 0.25).

Fig. 5.4 and *5.5* below show the graphical comparisons of MSE v/s bpp and PSNR v/s bpp of entire image and RoI, respectively. The superiority of RoI parameters is clearly visible for these graphs. It has been observed that for entire image the background ha been compressed worstly which is responsible for high MSE. But the image area (RoI) since the quality is a prime requirement; it has been compressed with the least compression ratio which yields in remarkable reduction in MSE and improvement in PSNR as compared to entire image.

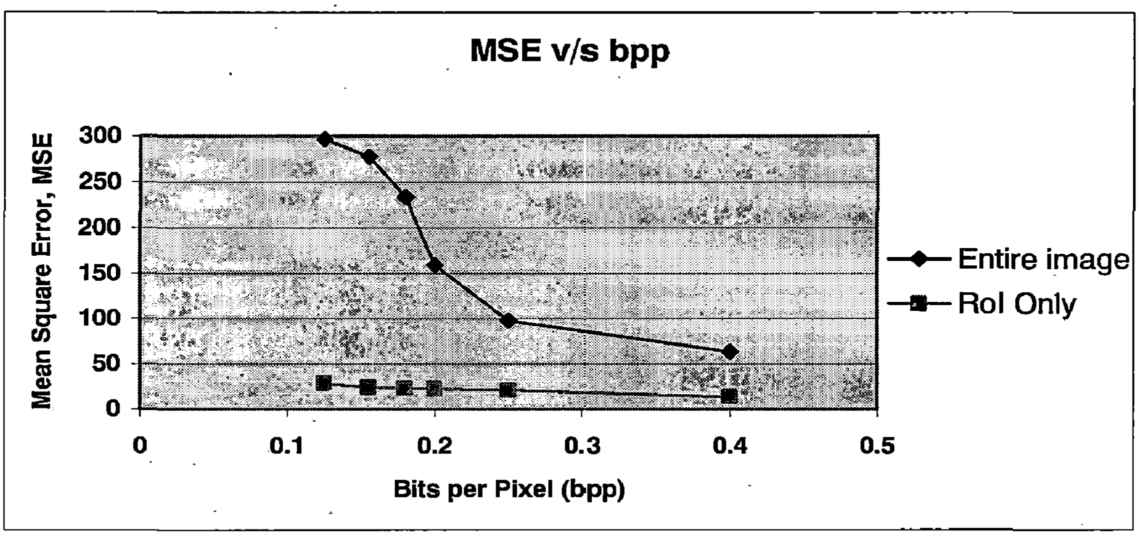


Fig. 5.4: MSE comparison for JPEG 2000

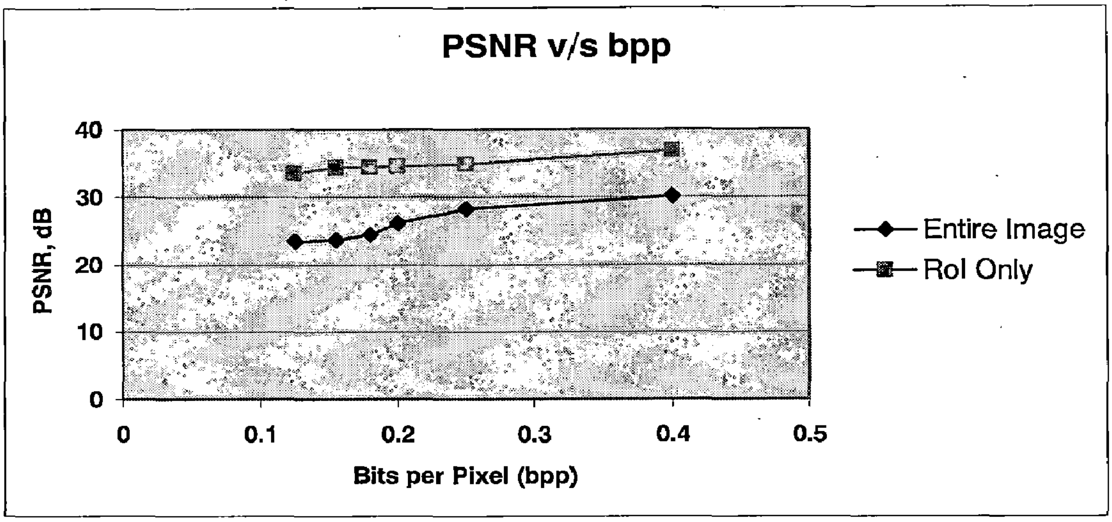


Fig. 5.5: PSNR comparison for JPEG 2000

Chapter 6
The SPIHT Coding

6.1 Introduction

Set partitioning in hierarchical trees *(SPIHT)*, an example of a progressive image compression algorithm is an extension of Shapiro's embedded zerotree wavelet (EZW) method [25]. These two new algorithms are a significant breakthrough in lossy image compression in that they give substantially higher compression ratios than prior techniques including JPEG, vector quantization, and the discrete wavelet transform combined with quantization. In addition, the algorithms allow for progressive transmission (meaning coarse approximations of an image can be reconstructed quickly from beginning parts of the bit stream), require no training, and are of low computational complexity.

The SPIHT algorithm uses the 9/7-tap biorthogonal ('bior 9.7') filter in the discrete wavelet transform. To take advantage of the selfsimilarity among wavelet coefficient magnitudes in different scales, the coefficients are grouped into tree structures called *zerotrees.* The organization of wavelet coefficients into a zerotree is based on relating each coefficient at a given scale (parent) to a set of four coefficients with the same orientation at the next finer scale (children). Zerotrees allow the prediction of insignificance of the coefficients across scales (that is, if the parent is insignificant with respect to a given threshold, its children are also likely to be insignificant) and represent this efficiently by coding the entire tree at once [26].

SPIHT groups the wavelet coefficient trees into sets and orders coefficients by the highest bit plane of the magnitude. The ordering information is encoded with a set partitioning algorithm. This algorithm is fully reproduced at the decoder. The SPIHT algorithm transmits the wavelet coefficients in bit plane order with most significant bit plane first. For each bit plane there are two passes. In the first pass, called the *dominant pass,* coefficients which are significant with respect to the current threshold are found and coded using the set partitioning method. In the second pass, the *subordinate pass,* the precision of all previously significant coefficients is increased by sending the next bit from the binary representation of their values. Such refinement allows for progressive-approximation quantization and produces a fully embedded code, i.e., the transmission of the encoded bit stream can be stopped at any point and a lower rate image can still be decompressed and reconstructed. Additionally, a target bit rate or target distortion can be met exactly.

6.2 SPIHT Features

The SPIHT method is not a simple extension of traditional methods for image compression, and represents an important advance in the field. Different compression methods were developed specifically to achieve at least one of the objectives mentioned below. But what makes SPIHT really outstanding is that it yields all these qualities **simultaneously**. The method deserves special attention because it provides the following:

- Good image quality, high PSNR, especially for color images
- Optimized for progressive image transmission
- Produces a fully embedded coded file
- Simple quantization algorithm
- Fast coding/decoding (nearly symmetric)
- Has wide applications, completely adaptive
- Can be used for lossless compression
- Can code to exact bit rate or distortion

6.2.1 Image Quality

Extensive research has shown that the images obtained with wavelet-based methods yield very good visual quality. At first it was shown that even simple coding methods produced good results when combined with wavelets. SPIHT belongs to the next generation of wavelet encoders, employing more sophisticated coding. In fact, SPIHT exploits the properties of the wavelet-transformed images to increase its efficiency. The advantage of SPIHT is even more pronounced in encoding color images, because the bits are allocated automatically for local optimality among the color components, unlike other algorithms that encode the color components separately based on global statistics of the individual components. It is quite normal for SPIHT to obtain visually lossless color compression at compression ratios from 100-200:1.

6.2.2 Progressive Image Transmission

The problem with some widely used schemes that use progressive transmission employ a very primitive progressive image transmission method. On the other extreme, SPIHT is a state-of-the-art method that was *designed for* optimal progressive transmission (and still beats most non-progressive methods!). It does so by producing a fully embedded coded file in a manner that at any moment the quality of the displayed image is the best available for the number of bits received up to that moment. So, SPIHT can be very useful for applications where the user can quickly inspect the image and decide if it should be really downloaded, or is good enough to be saved, or need refinement.

6.2.3 Optimized Embedded Coding

A strict definition of the embedded coding scheme is: if two files produced by the encoder have size M and N bits, with $M > N$, then the file with size N is *identical* to the first N bits of the file with size M. Let's see how this abstract definition is used in practice. Suppose if we need to compress an image for three remote users. Each one has different needs of image reproduction quality, and we find that those qualities can be obtained with the image compressed to at least 8 Kb, 30 Kb, and 80 Kb, respectively. If we use a non-embedded encoder (like JPEG), to save in transmission costs (or time) we must prepare one file for each user. On the other hand, if we use an embedded encoder (like SPIHT) then we can compress the image to a single 80 Kb file, and then send the first 8 Kb of the file to the first user, the first 30 Kb to the second user, and the whole file to the third user.

Surprisingly, with SPIHT *all* three users would get (for the same file size) an image quality comparable or superior to the most sophisticated non-embedded encoders available today. SPIHT achieves this feat by optimizing the embedded coding process and always coding the most important information first. An even more important application is for progressive image transmission, where the user can decide at which point the image quality satisfies his needs, or abort the transmission after a quick inspection, etc.

6.2.4 Encoding/Decoding Speed

The SPIHT process represents a very effective form of entropy-coding using two forms of coding: binary-uncoded and context-based adaptive arithmetic coded. Surprisingly, the difference in compression is small, showing that it is not necessary to use slow methods. A fast version using Huffman codes can also be implemented. A straightforward consequence of the compression simplicity is the greater coding/decoding speed. The SPIHT algorithm is nearly symmetric, i.e., the time to encode is nearly equal to the time to decode. Complex compression algorithms tend to have encoding times much larger than the decoding times.

6.2.5 Rate or Distortion Specification

Almost all image compression methods developed so far do not have precise rate control. For some methods you specify a target rate, and the program tries to give something that is not too far from what you wanted. For others you specify a "quality factor" and wait to see if the size of the file fits your needs. The embedded coding property of SPIHT allows *exact* bit rate control, without any penalty in performance (no bits wasted with padding). The same property also allows exact mean squared-error (MSE) distortion control. Even though the MSE is not the best measure of image quality, it is far superior to other criteria used for quality specification.

6.2.6 Lossless Compression

SPIHT codes the individual bits of the image wavelet transform coefficients following a bit-plane sequence. Thus, it is capable of recovering the image perfectly (every single bit of it) by coding all bits of the transform. However, the wavelet transform yields perfect reconstruction only if its numbers are stored as infinite-precision numbers. In practice it is frequently possible to recover the image perfectly using rounding after recovery, but this is not the most efficient approach. For lossless compression an integer multiresolution transformation, similar to the wavelet transform, also called S+P transform, has been used. It solves the finite-precision problem by carefully truncating the transform coefficients *during* the transformation (instead of after). In other words, the property that SPIHT yields progressive transmission with practically no penalty in compression efficiency applies to lossless compression too.

6.3 SPIHT Coder Overview

The success of the SPIHT is in part due to the organization of the wavelet coefficients into the spatial orientation trees. *Fig. 6.1* shows the spatial orientation tree in a wavelet pyramid, which has the tree root in the subband at the coarsest level of the pyramid and branches successively into the subbands at the finer level in the same spatial orientation. The SPIHT coder uses three types of sets: *D(i, j)* denoting the set of all descendants of a node *(i, j)*, *O(i, j)* representing the set of all offspring of the node *(i, j)*, and *L(i, j)* representing the set of all descendants excluding the immediate four offspring (called non-child descendants) of the node *(i, j)*, that is, $L(i, j)= D(i, j) - O(i, j)$. A wavelet coefficient at coordinate *(i, j)*, $c_{i,\,j}$ is called *significant* with respect to a given threshold T ($T = 2^n$ at resolution n) if $c_{i,\,j} \geq T$; otherwise, it is called *insignificant*.

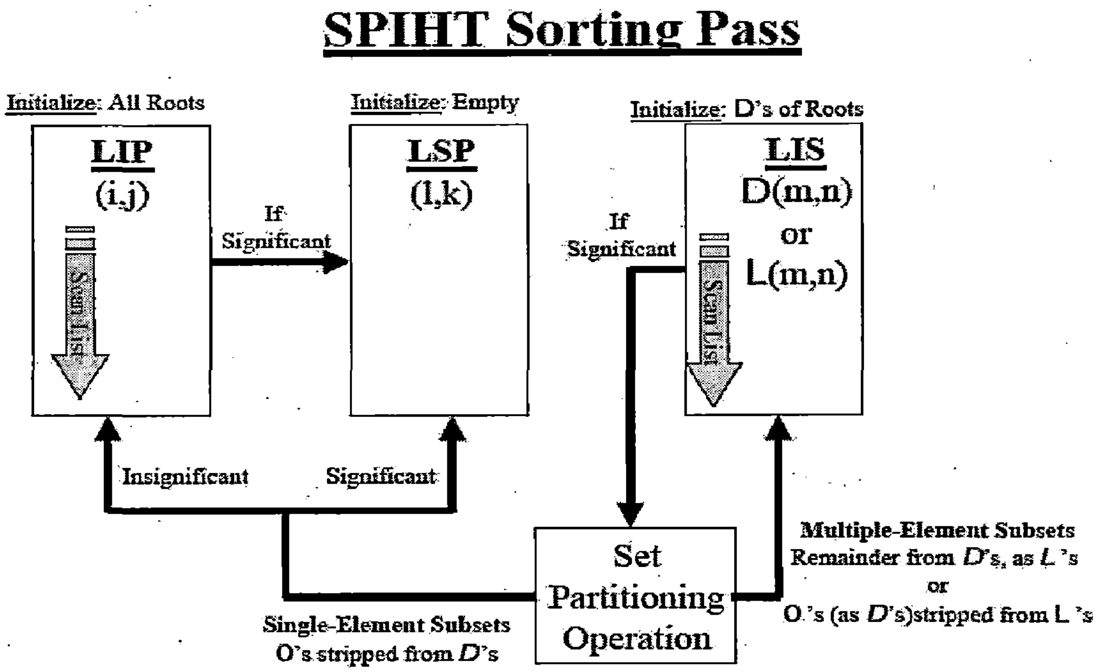
O(i,j): set of coordinates of all <u>offspring</u> of node (i,j); *children only*
D(i,j): set of coordinates of all <u>descendants</u> of node (i,j); *children, grandchildren, great-grand, etc.*
H(i,j): set of all tree roots (nodes in the highest pyramid level); *parents*
L(i,j): **D**(i,j) – **O**(i,j) (all descendents except the offspring); *grandchildren, great-grand, etc.*

```
Initialization
      n = ⌊log2 (max |coeff|)⌋
      LIP = All elements in H
      LSP = Empty
      LIS = D's of Roots
Significance Map Encoding ("Sorting Pass")
      Process LIP
            for each coeff (i,j) in LIP
                  Output Sn(i,j)
                  If Sn(i,j)=1
                        Output sign of coeff(i,j): 0/1 = -/+
                        Move (i,j) to the LSP
                  Endif
            End loop over LIP
```

$$S_n(\Gamma)=\begin{cases}1, & \max\limits_{(i,j)\in\Gamma}\left\{\left|c_{ij}\right|\right\}\geq 2^n\\ 0, & \text{otherwise}\end{cases}$$

```
      Process LIS
            for each set (i,j) in LIS
                  if type D
                        Send Sn(D(i,j))
                        If Sn(D(i,j))=1
                              for each (k,l)∈ O(i,j)
                                    output Sn(k,l)
                                    if Sn(k,l)=1, then add (k,l) to the LSP and output sign of coeff: 0/1 = -/+
                                    if Sn(k,l)=0, then add (k,l) to the end of the LIP
                              endfor
                        endif
                  else (type L )
                        Send Sn(L(i,j))
                        If Sn(L(i,j))=1
                              add each (k,l) ∈ O(i,j) to the end of the LIS as an entry of type D
                              remove (i,j) from the LIS
                  end if on type
            End loop over LIS
Refinement Pass
      Process LSP
            for each element (i,j) in LSP – except those just added above
                  Output the nth most significant bit of coeff
            End loop over LSP
Update
      Decrement n by 1
      Go to Significance Map Encoding Step
Adaptive Arithmetic Code (Optional)
```

Fig. 6.1: SPIHT algorithm



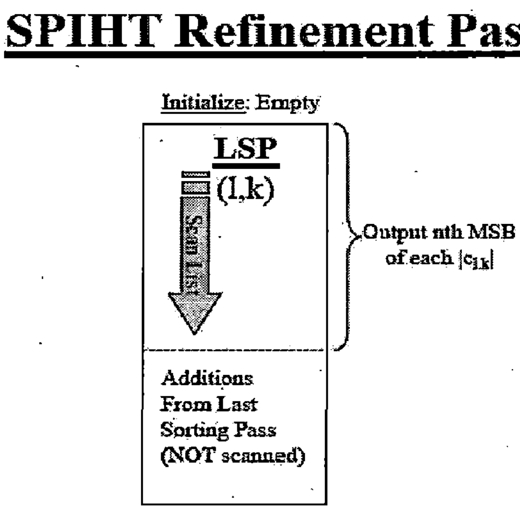


Fig. 6.2: SPIHT sorting and refinement pass

In a practical implementation, the significant information is stored in three ordered lists: a list of insignificance sets *(LIS)*, a list of insignificant pixels *(LIP)*, and a list of significant pixels *(LSP)*. In all lists, each entry is identified by a coordinate *(i, j)*, which represents individual pixels in the LIP and LSP, and indicates either the set *D (i, j)* (a set of type A) or *L (i, j)* (a set of type B) in the LIS. At the initialization step, the coefficients in the highest pyramid level (the coarsest level) are added to LIP and only those with descendants also to LIS as type A entries. The LSP is set as an empty list. The SPIHT coder starts with the most significant bit plane. At every bit plane, it tests the tree lists in order, starting with LIP, followed by LIS and LSP. The coefficients in the LIP are coded. Those that become significant are moved to the end of the LSP and their signs are coded.

Similarly, sets *D (i, j)* and *L (i, j)* are sequentially coded following the LIS order, and those that become significant are partitioned into subsets. The set partitioning rules are as follows: (1) If *D (i, j)* is significant, then it is partitioned into *L (i, j)* plus the four single-element sets with *(k, l)*∈*O(I, j)*. (2) If *L(i, j)* is significant, then it is partitioned into four sets *D(k, l)* with *(k, l)*∈*O(I, j*. Newly formed sets *L(i, j), D(k, l)* are added to the end of LIS to be coded again before the same sorting pass ends. Finally, each coefficient in LSP except the ones added in the last sorting pass is refined in each refinement pass. The algorithm then repeats the above procedure for the next resolution, and then stops at desired bit rates. The decoding algorithm can be obtained by duplicating the encoder's execution path. An additional task done by the decoder is to update the reconstructed image [27]. The entire discussion is explained graphically in the *Fig. 6.1*. *Fig. 6.2* below gives the graphical explanation of Sorting and Refinement pass.

6.4 An Example of SPIHT

26	6	13	10
-7	7	6	4
4	-4	4	-3
2	-2	-2	0

Initialization

LIP

(0,0) → 26
(0,1) → 6
(1,0) → -7
(1,1) → 7

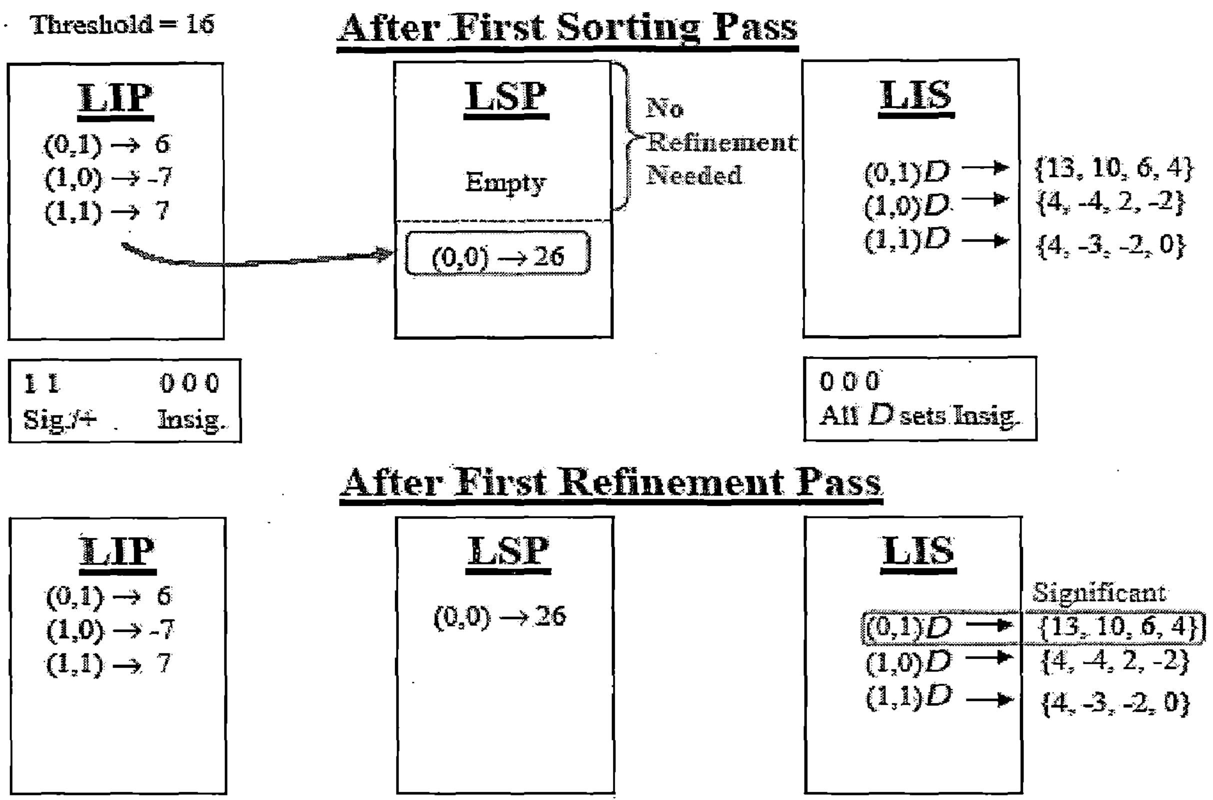
LSP

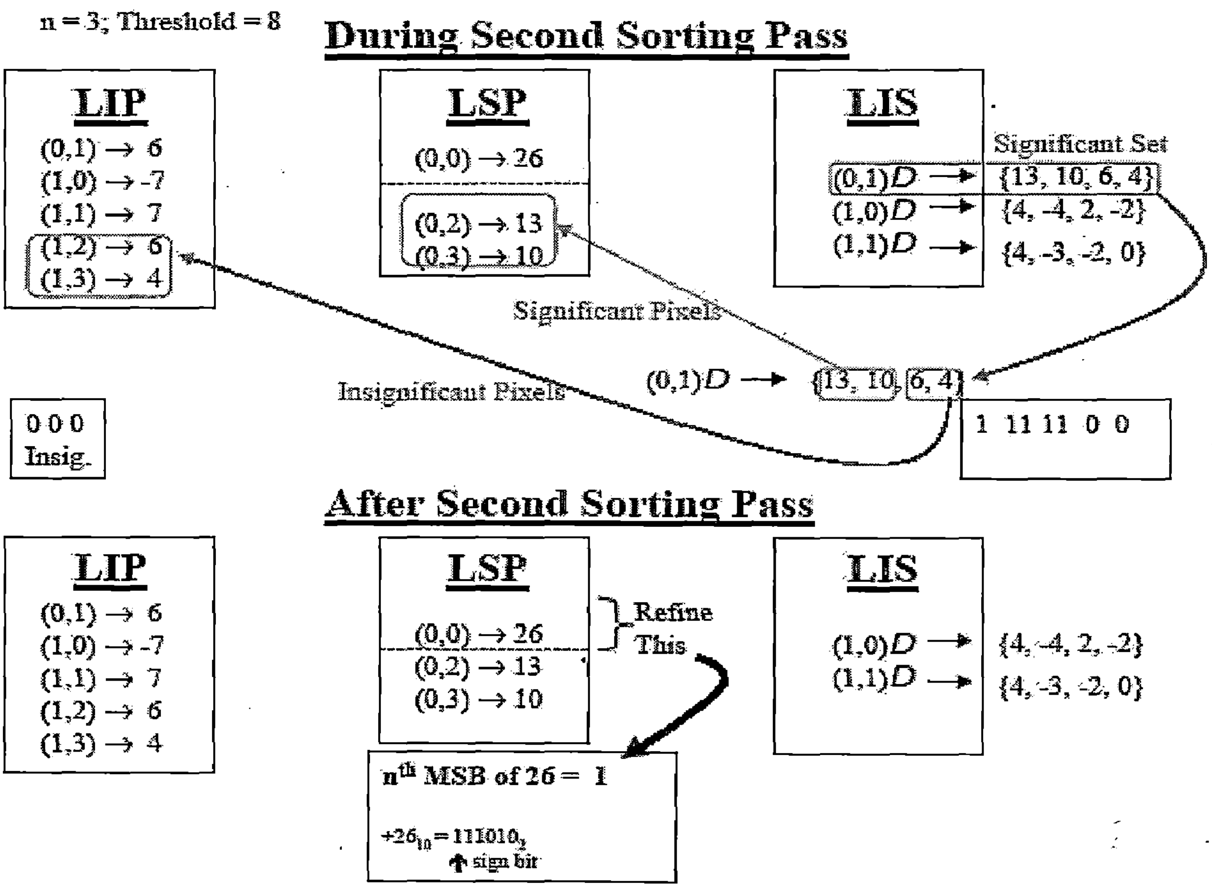
Empty

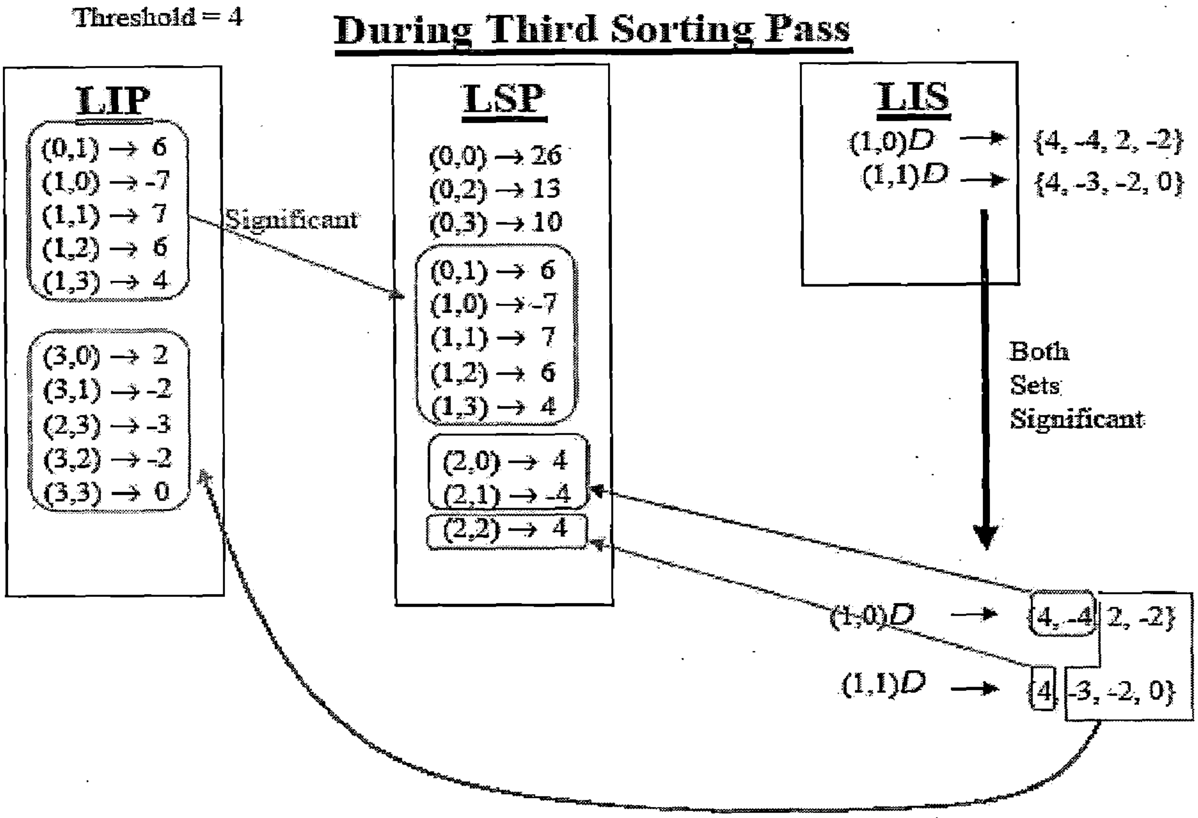
$n = \lfloor \log_2(26) \rfloor = 4$

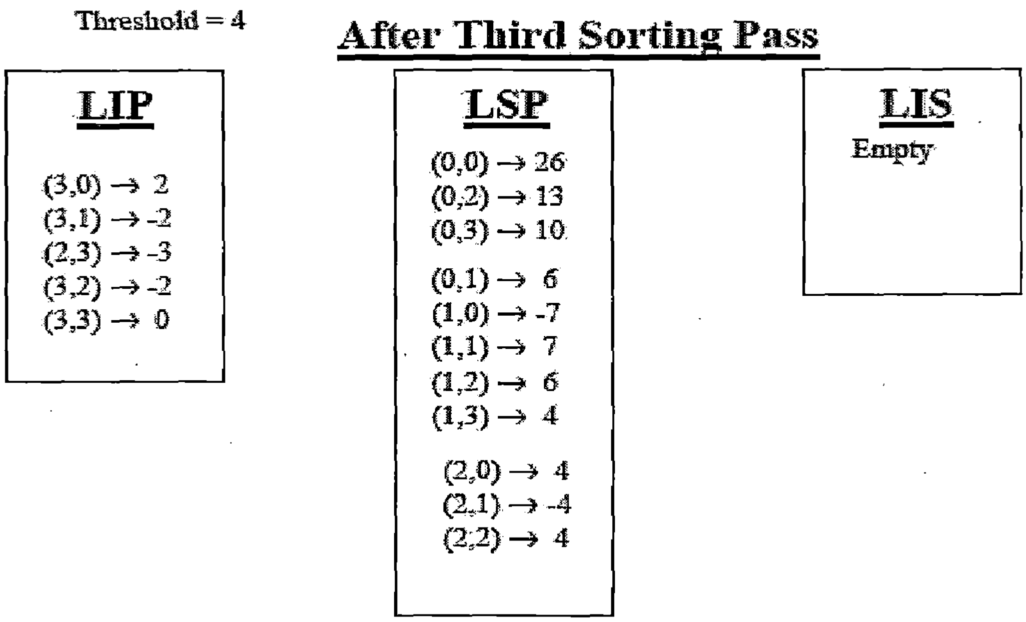
LIS

(0,1)*D* → {13, 10, 6, 4}
(1,0)*D* → {4, -4, 2, -2}
(1,1)*D* → {4, -3, -2, 0}


Threshold = 16
After First Sorting Pass
LIP
(0,1) → 6
(1,0) → -7
(1,1) → 7
LSP
Empty
(0,0) → 26
No Refinement Needed
LIS
(0,1)D → {13, 10, 6, 4}
(1,0)D → {4, -4, 2, -2}
(1,1)D → {4, -3, -2, 0}
1 1 Sig./+
0 0 0 Insig.
0 0 0 All D sets Insig.
After First Refinement Pass
LIP
(0,1) → 6
(1,0) → -7
(1,1) → 7
LSP
(0,0) → 26
LIS
Significant
(0,1)D → {13, 10, 6, 4}
(1,0)D → {4, -4, 2, -2}
(1,1)D → {4, -3, -2, 0}


n = 3; Threshold = 8
During Second Sorting Pass
LIP
(0,1) → 6
(1,0) → -7
(1,1) → 7
(1,2) → 6
(1,3) → 4
LSP
(0,0) → 26
(0,2) → 13
(0,3) → 10
LIS
Significant Set
(0,1)D → {13, 10, 6, 4}
(1,0)D → {4, -4, 2, -2}
(1,1)D → {4, -3, -2, 0}
Significant Pixels
Insignificant Pixels
(0,1)D → {13, 10, 6, 4}
0 0 0 Insig.
1 11 11 0 0
After Second Sorting Pass
LIP
(0,1) → 6
(1,0) → -7
(1,1) → 7
(1,2) → 6
(1,3) → 4
LSP
(0,0) → 26
(0,2) → 13
(0,3) → 10
Refine This
LIS
(1,0)D → {4, -4, 2, -2}
(1,1)D → {4, -3, -2, 0}
nth MSB of 26 = 1
+26₁₀ = 111010₂
sign bit





There have been few advancements over the years in the SPIHT technology since its inception in1996. They are partial SPIHT (P-SPIHT), enhanced partial SPIHT and Vector SPIHT (VSPIHT) as suggested in [28]-[30].

6.5 Results and Discussions

The SPIHT algorithm- encoder and decoder- has been developed using Matlab 6.5. The functions like *dwt*, *idwt*, *wavedec2*, *waverec2* etc. have been extensively used for developing SPIHT encoder and decoder. The initialization sequence, sorting pass, refinement pass are all developed using Matlab functions.

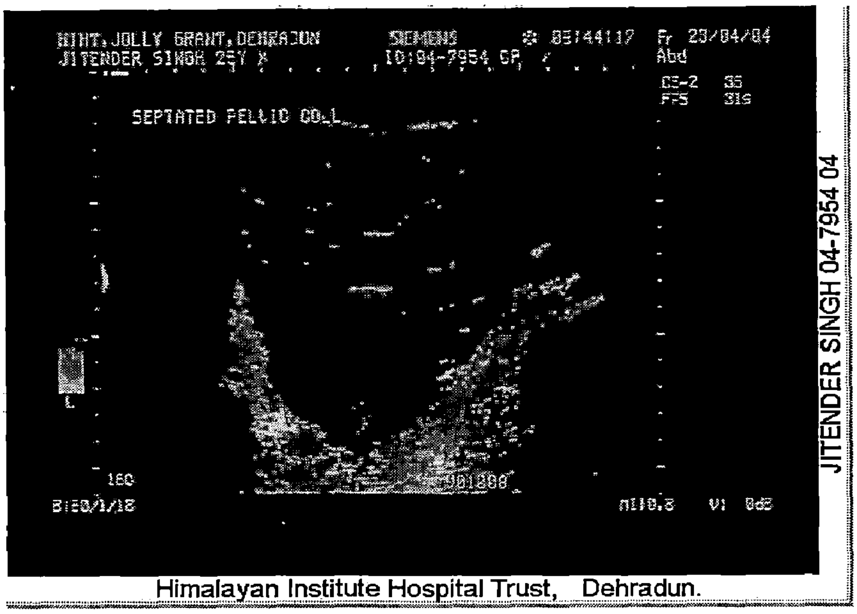


Fig. 6.3: Original US image (IMAGE 1)

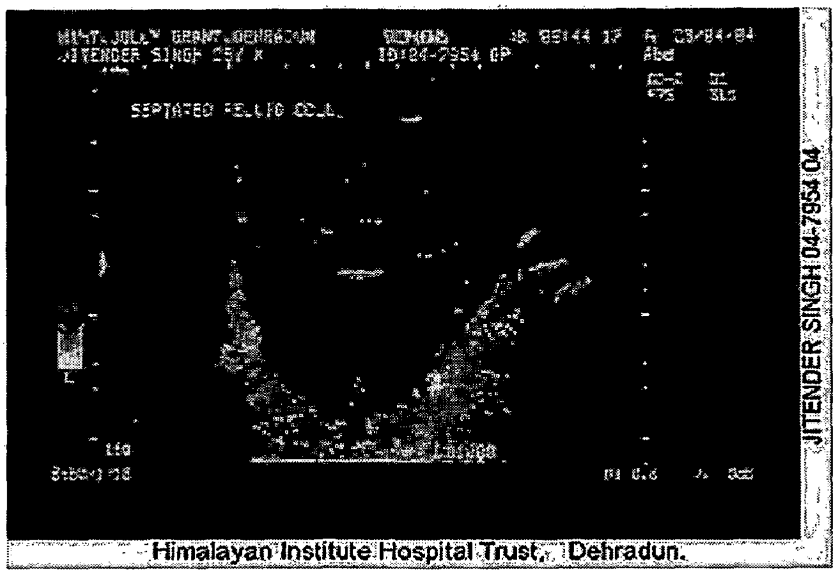


Fig. 6.4: SPIHT (contextually) compressed image with CR=65.24 (bpp =0.123)

Tables 6.1 and 6.2 summarize the compression parameters for SPIHT for the original US image shown in *Fig. 6.3*. The compressed image is shown in *Fig. 6.4*, having a CR of 65.24 and PSNR of 23.69 dB, respectively. By inspecting Table 6.1 and 6.2 it is revealed that the least MSE corresponding to bpp of 0.40 is 258.19 and the PSNR for the same is 24.01 dB; and the same for RoI (image area) are 12.98 and 36.99 dB, respectively. This signifies that the image area has been compressed with much quality-preserving manner which is the prime requirement of the contextual compression technique.

Table 6.1: SPIHT Parameters for Entire Image

Sr. No.	CR	bpp	MSE	PSNR, dB	Correlation Coefficient
1.	20.0	0.40	258.19	24.01	0.9798
2.	32.0	0.25	261.99	23.95	0.9795
3.	40.0	0.20	268.66	23.84	0.9790
4.	44.38	0.18	276.81	23.71	0.9783
5.	51.60	0.155	277.70	23.69	0.9783
6.	65.24	0.123	280.06	23.65	0.9781

Table 6.2: SPIHT Parameters for Image Area (RoI)

Sr. No.	CR	bpp	MSE	PSNR, dB	Correlation Coefficient
1.	20.0	0.40	12.98	36.99	0.9938
2.	32.0	0.25	14.68	36.46	0.9930
3.	40.0	0.20	16.09	36.07	0.9923
4.	44.38	0.18	16.33	36.0	0.9922
5.	51.60	0.155	17.28	35.75	0.9918
6.	65.24	0.123	19.75	35.17	0.9906

Also, the SPIHT could achieve the highest CR of 65.24 (bpp of 0.123). At this CR the PSNR and CoC obtained for the entire image are 23.65 dB and 0.9781 respectively. This suggests that the SPIHT algorithm is much effective at lower bit rates (below 0.25 bpp). The same parameters for image area (RoI) are 35.97 dB and 0.9906 respectively, which are considered as quite remarkable.

Fig. 6.5 and *6.6* below show the graphical comparisons of MSE v/s bpp and PSNR v/s bpp of entire image and RoI respectively. The superiority of RoI parameters is clearly visible from these graphs. It is observed that although the MSE for entire image at lower bpp_s are very high but at higher bpp_s (below 0.18 bpp) they are quirt reasonable. For RoI even this is not the case and they are quite low for all CRs (and bpp_s). Thus there is not a sharp fall in the MSE values even at lower bit rates and they are almost constant. Hence, the SPIHT coding can aptly be used when lower bit rate is a prime criteria. Similarly, PSNR values too are almost constant for both entire image and RoI for all bpp values. Further, the PSNR values for RoI are quite large as compared to the entire image.

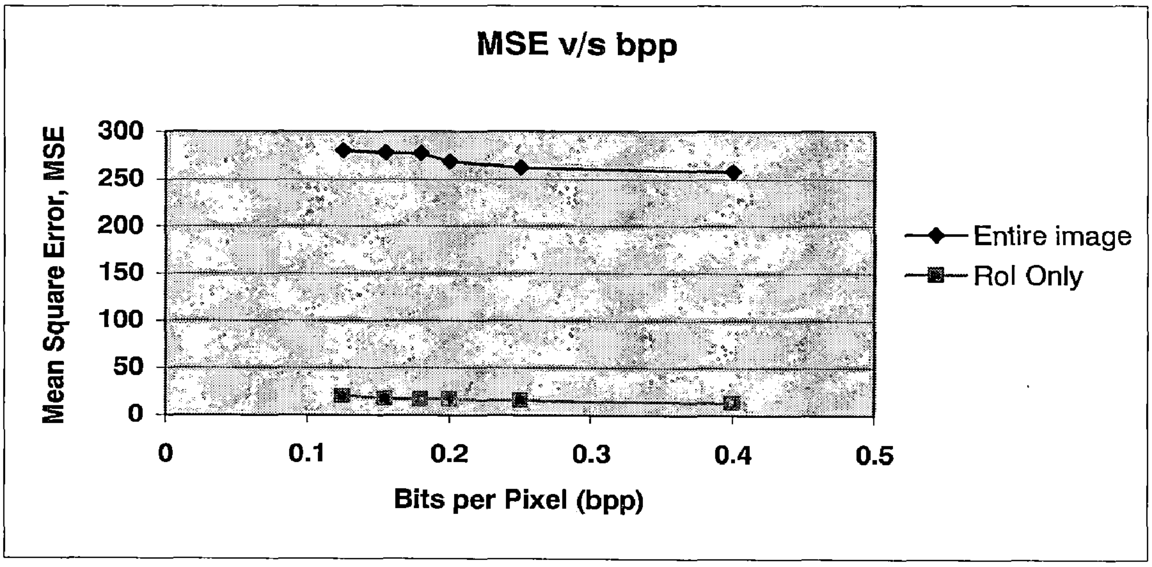


Fig. 6.5: MSE comparison for SPIHT

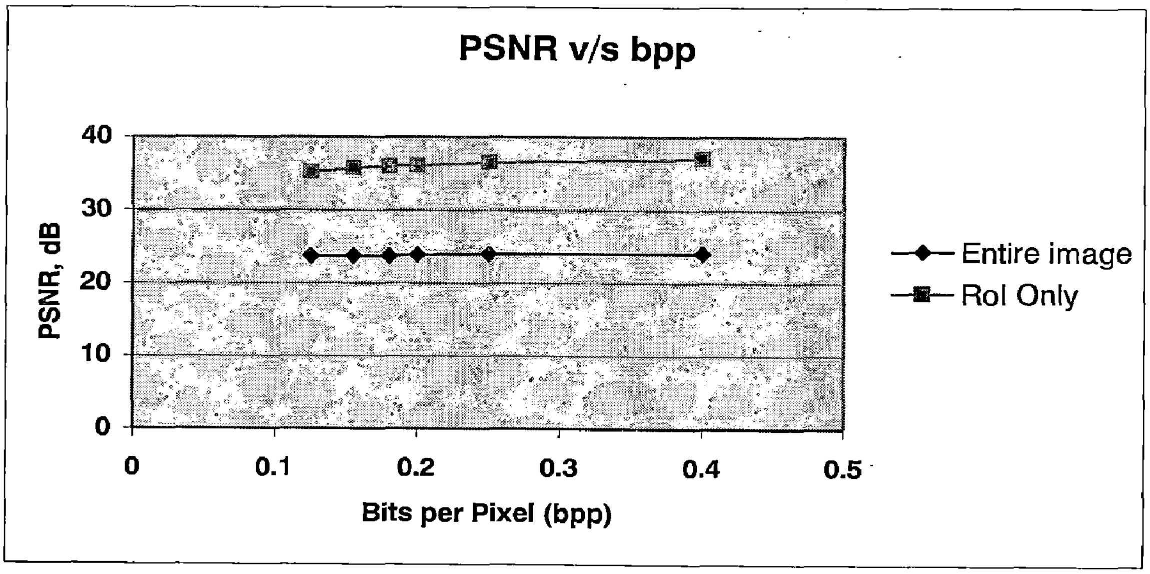


Fig. 6.6: PSNR comparison for SPIHT

Chapter 7

Comparative Analysis

7.1 Tabular Representation

In the present work, four compression algorithms viz. the JPEG coding, the Wavelet transform coding, the JPEG 2000 coding and the SPIHT coding have been implemented. Table 7.1 and 7.2 present the comparative representation of MSE and PSNR values of the image compressed by these methods, respectively for entire image. All the results shown are for five levels of compression ratios ranging from 20.0 (bpp of 0.4) to 51.6 (bpp of 0.155). The prime contribution of contextual compression technique over the conventional compression techniques is that it give higher compression ratios while maintaining the supreme quality of diagnostically important regions (RoI).

From Table 7.1 it is observed that JPEG 2000 compressed image has achieved the least MSE of 63.39 corresponding to CR of 20.0 (bpp of 0.4). Whereas JPEG compressed image has the highest amount of MSE of 327.05 indicating that JPEG is not much suitable for lower bit rates (i.e. bpp below 0.25). The Wavelet transform coded image has the MSE of 128.02 even at the CR of 51.6 (bpp of 0.155). The SPIHT coded image has very high MSEs at higher bit rates but quite reasonable MSEs at lower bit rates. Thus, SPIHT compression algorithm performs better at lower bit rates and so does the JPEG 2000 coding.

Table 7.1: Overall MSE comparison for entire image

Sr. No.	CR	bpp	MSE			
			JPEG	JPEG 2000	Wavelet	SPIHT
1.	20.0	0.4	78.56	**63.39**	81.64	258.19
2.	32.0	0.25	133.75	97.75	95.82	261.99
3.	40.0	0.2	210.69	159.11	118.98	268.66
4.	44.4	0.18	246.95	233.78	124.25	276.81
5.	51.6	0.155	327.05	277.52	128.02	277.70

It is observed from Table 7.2, that JPEG 2000 compressed image has achieved the highest PSNR corresponding to CR of 20.0 (bpp of 0.4). The Wavelet transform coded image has got the PSNR of 27.06 dB even at the CR of as high as 51.6 (bpp 0f 0.155). Even though SPIHT coding could not achieve much higher PSNR for higher bit rates, the difference in PSNR corresponding to lowest and highest CR is minor, indicating that it has performed very steadily.

Table 7.2: Overall PSNR comparison for entire image

Sr. No.	CR	bpp	PSNR			
			JPEG	**JPEG 2000**	**Wavelet**	**SPIHT**
1.	20.0	0.4	29.18	**30.11**	29.01	24.01
2.	32.0	0.25	26.86	28.23	28.32	23.95
3.	40.0	0.2	24.89	26.11	27.38	23.84
4.	44.4	0.18	24.21	24.44	27.19	23.71
5.	51.6	0.155	22.98	23.69	27.06	23.69

Table 7.3 compares the four methods on the basis of correlation coefficient (CoC). CoC is a measure of how closely the compressed image is correlated to the original image. JPEG 2000 has got the highest CoC Of 0.9951 at CR of 20.0, while Wavelet has achieved the CoC of 0.9902 at CR of 51.6 which is quite outstanding.

Table 7.3: Overall CoC comparison for entire image

Sr. No.	CR	bpp	CoC			
			JPEG	**JPEG 2000**	**Wavelet**	**SPIHT**
1.	20.0	0.4	**0.9954**	0.9951	0.9938	0.9798
2.	32.0	0.25	0.9895	0.9924	0.9927	0.9795
3.	40.0	0.2	0.9834	0.9876	0.9909	0.9790
4.	44.4	0.18	0.9806	0.9817	0.9905	0.9783
5.	51.6	0.155	0.9746	0.9782	0.9902	0.9783

Table 7.4 and 7.5 compare the four methods on the basis of MSE and PSNR for image area (RoI). It is observed that Wavelet transform coded image has achieved the least MSE of 10.02 (as compared to 81.64 for entire image) for CR of 20.0. This signifies the quality presentation in the RoI compression. For Wavelet and SPIHT, the MSEs are in the tens which are less than one tenth of MSEs of entire image.

The PSNR values for RoI are also quite high for all four methods with Wavelet leading with values of 38.12 dB, highest among all. This value is almost 10 dB higher compared to its entire image value of 29.01 dB.

Table 7.4: Overall MSE comparison for image area (RoI)

Sr. No.	CR	bpp	MSE			
			JPEG	**JPEG 2000**	**Wavelet**	**SPIHT**
1.	20.0	0.4	36.56	13.29	**10.02**	12.98
2.	32.0	0.25	41.59	20.66	11.77	14.68
3.	40.0	0.2	44.06	22.62	13.59	16.09
4.	44.4	0.18	47.69	22.97	16.57	16.33
5.	51.6	0.155	49.62	23.52	17.93	17.28

Table 7.5: Overall PSNR comparison for image area (RoI)

Sr. No.	CR	bpp	PSNR			
			JPEG	**JPEG 2000**	**Wavelet**	**SPIHT**
1.	20.0	0.4	32.50	36.89	**38.12**	36.99
2.	32.0	0.25	31.94	34.87	37.42	36.46
3.	40.0	0.2	31.69	34.58	36.80	36.07
4.	44.4	0.18	31.34	34.52	35.93	36.0
5.	51.6	0.155	31.17	34.42	35.59	35.75

Table 7.6 compares all the methods on the basis of CoC obtained for image area (RoI). Here, too, Wavelet leads with the value of 0.9954 for CR of 20.0. The values of 0.9918 and 0.9915 for SPIHT and Wavelet suggest the outstanding correlation between the compressed and original images even at the CR of 51.6.

Table 7.6: Overall CoC comparison for image area (RoI)

Sr. No.	CR	bpp	CoC			
			JPEG	JPEG 2000	Wavelet	SPIHT
1.	20.0	0.4	0.9827	0.9938	**0.9954**	0.9938
2.	32.0	0.25	0.9803	0.9912	0.9944	0.9930
3.	40.0	0.2	0.9792	0.9892	0.9935	0.9923
4.	44.4	0.18	0.9775	0.9891	0.9921	0.9922
5.	51.6	0.155	0.9765	0.9888	0.9915	0.9918

7.2 Graphical Presentation

Fig. 7.1 through *7.4* show the graphical comparison of MSE and PSNR of the four methods for entire image and RoI for IMAGE 1. The Wavelet transform coding has got the MSEs as half as those for SPIHT. The MSE values for JPEG and JPEG 2000 are quite comparable with each other. Again for JPEG and JPEG 2000, the MSE values have large difference for lowest and highest CRs, whereas for Wavelet and SPIHT they do not vary much and have little difference. The MSE values for RoI in case of JPEG are much high as compared to other three methods.

Similarly, the PSNR values for JPEG and JPEG 2000 vary much for lowest and highest CRs as compared to Wavelet and SPIHT. Also, the PSNR values of JPEG are quirt low in comparison with the other methods.

Thus, in nutshell, it can be stated that overall the Wavelet transform based compression method has outperformed the other three methods considering the factors like MSE, PSNR, CoC and the visual quality of the compressed image. Although, the other Wavelet based methods JPEG 2000 and SPIHT are also not far behind in performance. It can also be concluded that JPEG should not be used for lower bit rates i.e. bpp less than 0.25 as it has given quite high values of MSE both for entire image as well as RoI. Again, the JPEG 2000 and SPIHT should be used whenever lower bit rate are the prime requirements.

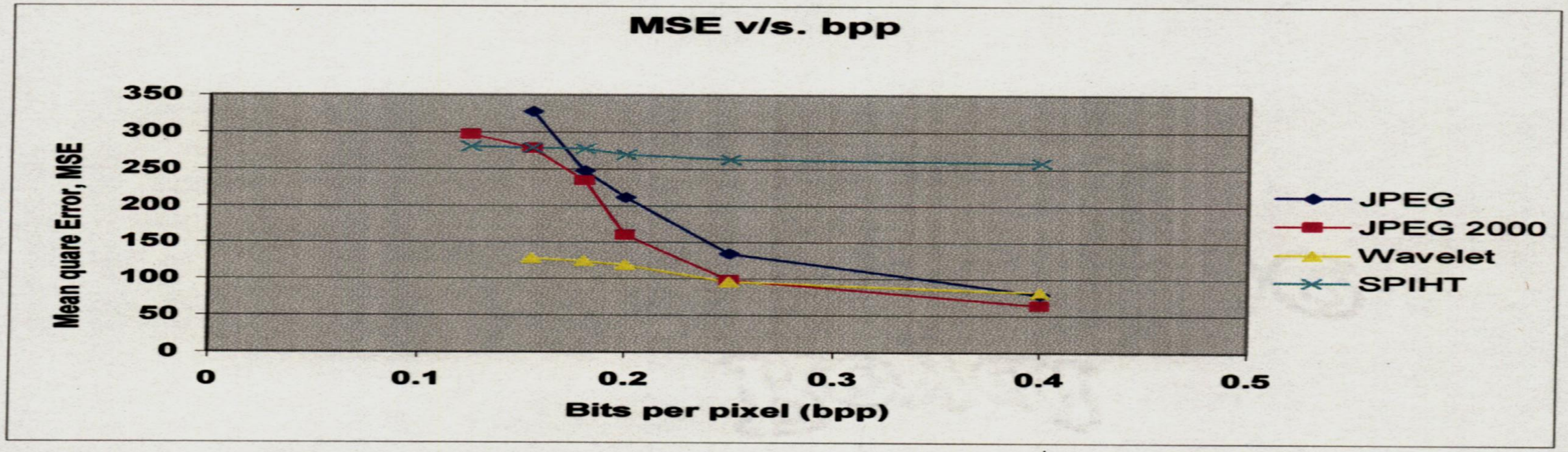


Fig. 7.1: MSE comparison for entire image

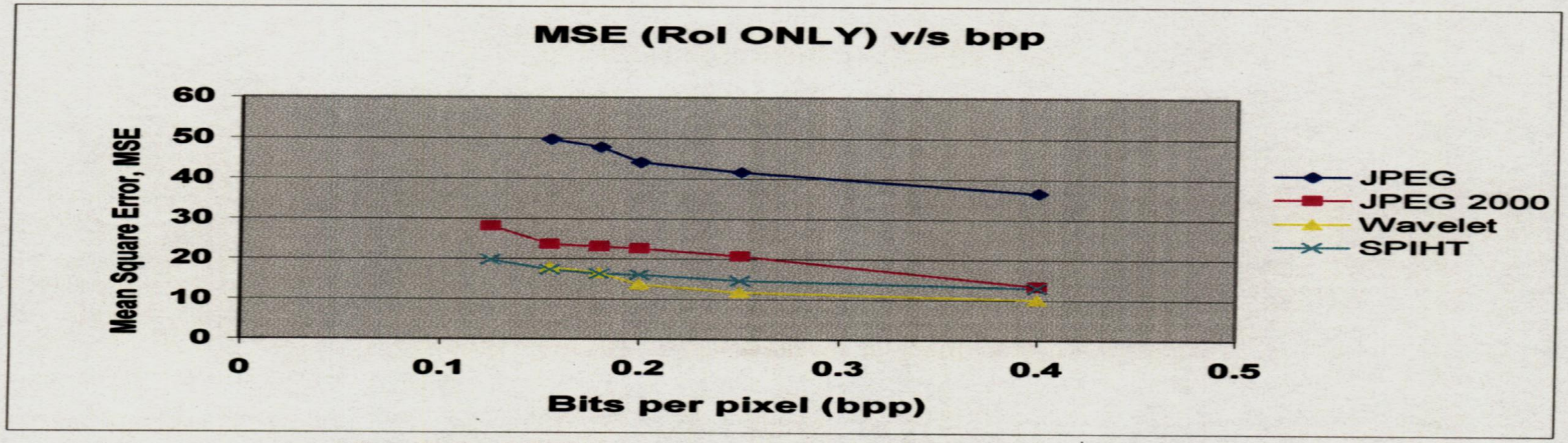


Fig. 7.2: MSE comparison for image area (RoI)

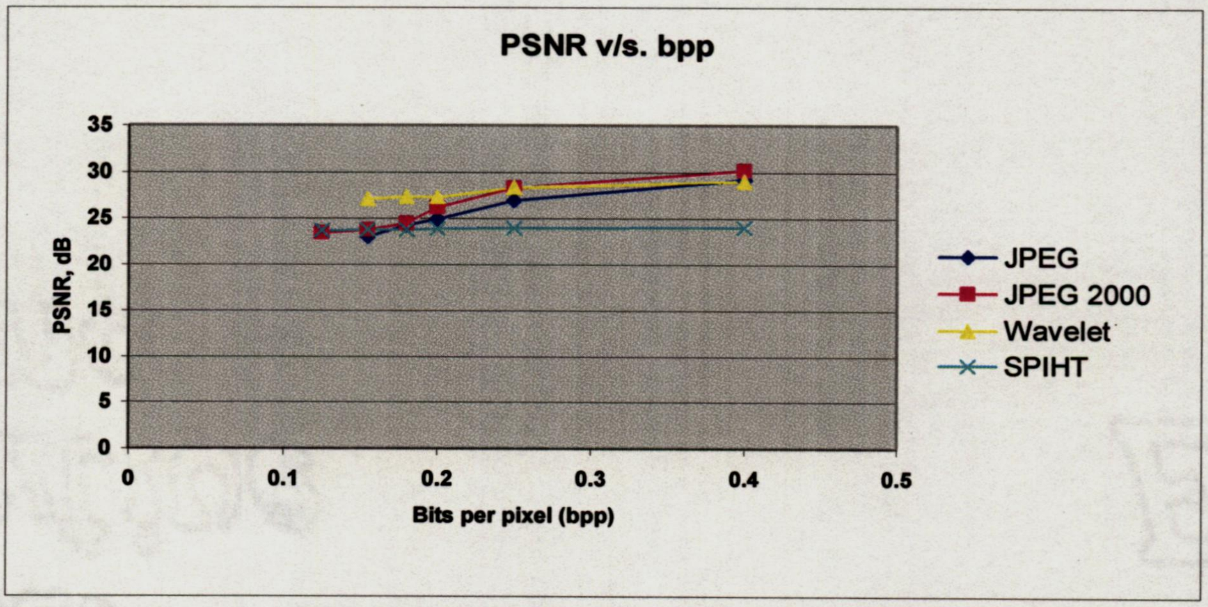


Fig. 7.3: PSNR comparison for entire image

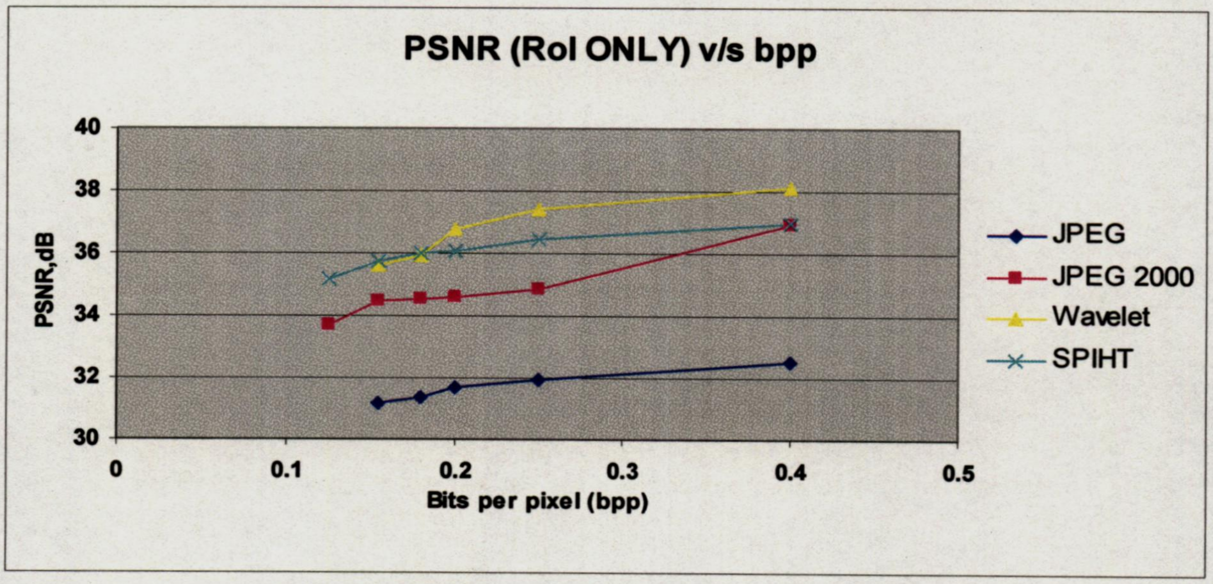


Fig. 7.4: PSNR comparison for image area (RoI)

Chapter 8
Conclusions and Scope for Future Work

8.1 Conclusions

The work in this thesis has addressed the concept of *Contextual Compression* in which the diagnostically important region of an Ultrasound image, called the *Region of Interest (RoI)*, has been compressed with the supreme quality as compared to rather unimportant area, the background. In the present work four compression algorithms viz. the JPEG coding, the Wavelet transform, the JPEG 2000 coding and the SPIHT coding have been implemented using Matlab 6.5 and Image Processing toolbox as a part of the contextual compression method. All the compression algorithms have been applied on the various Ultrasound medical images.

There has been few RoI-based compression techniques- mainly based on bitplane coding- proposed by the researchers [7] - [10], [14], [15]. These techniques are quite complex in the sense the separation algorithm (i.e. separating an RoI from background) is very complicated. In the present work, rather simpler approach of separating an RoI from the background has been adopted mainly using the Image Processing toolbox (of Matlab 6.5) instructions like *roipoly* and *immultiply*. The separated RoI and background are then compressed using the four compression algorithms, as a part of *contextual coding* scheme. The contextual compression scheme has been able to achieve a highest compression ratio of 65.24:1 (bpp 0.123) with modest values of MSE and PSNR, which is not possible with conventional compression schemes.

The main emphasis has been given on the compression algorithms and analysis parts. Analyzing the compression algorithms implemented, it reveals that the Wavelet Transform coding has outperformed as far as the parameters like MSE, PSNR and CoC are concerned. In fact the Wavelet transform based algorithms- JPEG 2000 and SPIHT have given quite encouraging results in addition to the Wavelet Transform itself.

The SPIHT coding could achieve the highest CR of 65.24 (bpp of 0.123) with a reasonable PSNR of 23.65 dB among all the methods. For JPEG 2000 these parameters are 64.0 (bpp of 0.125) and 23.41 dB, respectively. The other two methods could achieve the highest CR of 51.6 (bpp of 0.155). Beyond these CRs the visual quality of RoI as well as background is extremely degraded that could lead to the false diagnosis. Both JPEG 2000 and SPIHT have performed quite well at lower bit rates (i.e. bpp below 0.18), which is what the main cause was behind the innovation of these compression techniques. At lower bit rates these two methods have comprehensively outperformed the JPEG method. But the Wavelet transform method has beaten all the methods in all respects. Thus, it can be said that the Wavelet Transform method still leads the list of compression techniques.

The visual quality of an RoI is the prime requirement and of utmost importance in the contextual compression technique. In case of *conventional compression* schemes the equal loss of information will occur for important area as well as for redundant area, as they are compressed with equal CR. But in *contextual compression* scheme, the visual quality of important area (RoI) will be quite better due to less information loss of RoI area as compared to redundant area (background). Thus, the prime contribution of the *contextual compression* technique over the *conventional compression* techniques is immensely higher compression ratio with slightly deteriorated MSE and PSNR but extremely improved visual quality of image area (RoI).

8.2 Scope for Future Work

1. In the present work the compression algorithms have been applied on the grayscale Ultrasound images. But in practice, the true color Ultrasound scanners are already in use. Hence the present work can be extended for the true color Ultrasound images. Moreover, there are many other types of medical imaging modalities being practiced like X-ray, computer tomography (CT), and magnetic resonance imaging (MRI) etc. The work carried out in this thesis may be implemented on these images as well [18], [31] and [32].
2. Further, in the present work for the purpose of lesser complexity only one RoI has been selected. For obtaining even better performance and accuracy the work can be extended to multi-region (two, three or even more) aspect, as stated in [14] and [17].
3. Segmentation is one of the important methods for locating and separating out an RoI. Hence the segmentation based approaches as depicted in [33] - [34] can also be implemented for contextual compression.

Appendix A

A.1 Other Ultrasound Images Used

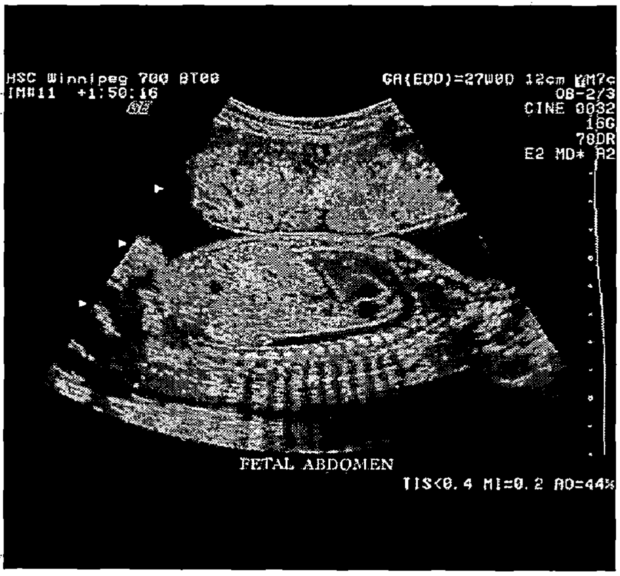


Fig. A.1: IMAGE 2(Fetal Abdomen)

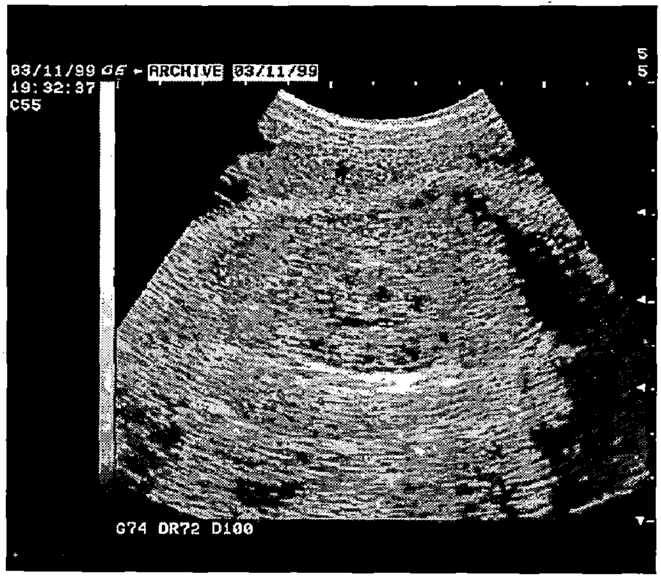


Fig. A.2: IMAGE 3(Kidney)

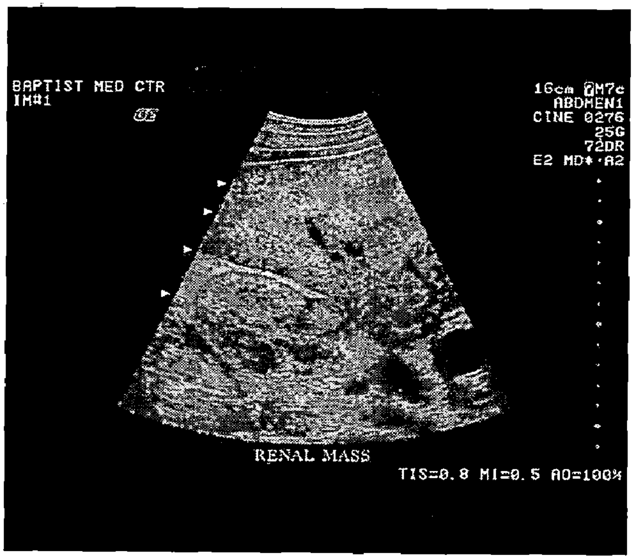


Fig. A.3: IMAGE 4 (Renal Mass)

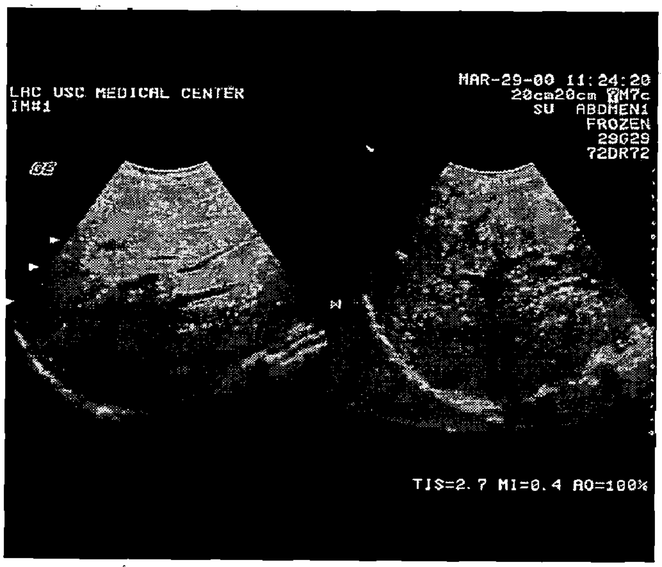


Fig. A.4: IMAGE 5 (Abdomen)

A.2 Results of Other Ultrasound Images

A.2.1 For IMAGE 2 (Fetal Abdomen):

i. Contextual JPEG:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	20.74	0.386	62.15	15.92	30.20	36.11	0.9909	0.9969
2.	30.48	0.262	127.44	25.18	27.08	34.12	0.9813	0.9951
3.	36.97	0.216	226.22	29.19	24.59	33.48	0.9699	0.9944
4.	40.41	0.198	263.27	35.74	23.93	32.60	0.9650	0.9931
5.	44.75	0.178	351.86	41.62	22.66	31.93	0.9562	0.9920

ii. Contextual JPEG 2000:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	19.42	0.412	56.62	24.23	30.60	34.29	0.9917	0.9953
2.	30.1	0.266	103.81	44.57	27.97	32.61	0.9848	0.9921
3.	34.0	0.235	142.97	45.44	26.58	31.57	0.9790	0.9912
4.	39.73	0.201	198.57	46.32	25.15	31.47	0.9707	0.9910
5.	44.13	0.181	233.64	47.84	24.45	31.33	0.9655	0.9908
6.	52.29	0.153	266.65	61.60	23.87	30.24	0.9655	0.9881

iii. Contextual Wavelet:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	21.24	0.377	64.06	13.97	30.06	36.68	0.9907	0.9973
2.	29.63	0.27	77.04	16.63	29.24	35.92	0.9886	0.9968
3.	38.54	0.208	101.70	20.19	28.06	35.08	0.9851	0.9961
4.	42.65	0.188	109.39	26.76	27.74	33.86	0.9840	0.9949
5.	50.31	0.16	11.93	30.05	27.41	33.35	0.9827	0.9942

iv. Contextual SPIHT:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	15.15	0.528	172.32	22.24	25.77	34.66	0.9746	0.9957
2.	24.1	0.332	178.68	30.74	25.61	33.26	0.9736	0.9941
3.	30.17	0.265	182.59	36.07	25.52	32.56	0.9731	0.9930
4.	33.65	0.238	184.84	40.19	25.46	32.09	0.9727	0.9922
5.	39.15	0.204	198.17	41.28	25.16	31.97	0.9707	0.9920
6.	46.41	0.172	200.28	52.83	25.11	30.90	0.9704	0.9898

A.2.2 For IMAGE 3 (Kidney):

i. Contextual JPEG:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	19.80	0.404	56.23	22.42	30.63	34.62	0.9971	0.9986
2.	28.60	0.28	108.01	39.98	27.80	32.11	0.9944	0.9975
3.	36.80	0.218	176.30	46.84	25.70	31.42	0.9918	0.9971
4.	39.65	0.202	195.45	57.31	25.22	30.55	0.9909	0.9965
5.	42.80	0.187	272.98	69.39	23.77	29.72	0.9884	0.9957

ii. Contextual JPEG 2000:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	19.55	0.41	66.61	21.68	29.90	31.63	0.9965	0.9972
2.	27.30	0.293	133.0	36.22	26.89	29.21	0.9931	0.9948
3.	32.125	0.25	145.87	38.77	26.50	28.88	0.9924	0.9947
4.	37.70	0.212	182.61	45.91	25.52	28.78	0.9905	0.9945
5.	40.60	0.197	200.12	57.16	25.12	28.60	0.9896	0.9940
6.	46.70	0.171	243.77	61.43	24.26	27.23	0.9873	0.9924

iii. Contextual Wavelet:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	21.24	0.38	57.09	21.92	30.56	34.72	0.9970	0.9987
2.	28.45	0.28	71.05	26.55	29.62	33.92	0.9963	0.9984
3.	34.16	0.23	104.85	33.02	27.93	32.95	0.9945	0.9980
4.	39.80	0.2	114.12	45.12	27.56	31.59	0.9941	0.9972
5.	46.11	0.173	123.50	51.11	27.21	31.05	0.9936	0.9969

iv. Contextual SPIHT:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	15.15	0.53	138.65	28.01	26.72	31.12	0.9928	0.9969
2.	24.10	0.33	163.07	31.34	26.00	29.42	0.9915	0.9954
3.	30.17	0.265	176.76	36.87	25.66	28.70	0.9908	0.9946
4.	33.65	0.24	182.86	39.06	25.51	28.46	0.9905	0.9943
5.	39.15	0.20	205.50	47.81	25.00	27.52	0.9893	0.9929
6.	43.65	0.18	220.50	54.29	24.70	26.98	0.9985	0.9920

A.2.3 For IMAGE 4 (Renal Mass):

i. Contextual JPEG:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	25.60	0.3125	50.42	14.28	31.10	36.59	0.9909	0.9967
2.	36.63	0.218	99.28	21.89	28.16	43.73	0.9820	0.9949
3.	43.33	0.185	186.53	25.22	25.42	34.11	0.9697	0.9941
4.	47.42	0.169	218.99	31.21	24.72	33.18	0.9644	0.9927
5.	51.50	0.155	299.11	35.06	23.37	32.68	0.9547	0.9918

ii. Contextual JPEG 2000:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	25.00	0.32	43.34	20.37	31.77	35.04	0.9922	0.9952
2.	38.01	0.21	78.93	30.78	29.16	33.03	0.9857	0.9914
3.	44.01	0.181	106.41	36.71	27.86	32.48	0.9806	0.9914
4.	51.05	0.157	147.52	37.19	26.44	32.43	0.9731	0.9910
5.	57.0	0.141	177.19	38.36	25.65	32.29	0.9676	0.9907
6.	67.03	0.119	197.46	47.29	25.18	31.38	0.9639	0.9889

iii. Contextual Wavelet:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	24.72	0.32	50.84	12.12	31.07	37.29	0.9908	0.9972
2.	39.12	0.204	60.93	14.31	30.28	36.57	0.9890	0.9967
3.	45.16	0.177	81.57	17.39	29.01	35.73	0.9852	0.9959
4.	52.05	0.154	87.79	22.79	28.70	34.55	0.9841	0.9947
5.	57.65	0.14	95.61	25.52	28.33	34.06	0.9826	0.9940

iv. Contextual SPIHT:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	15.15	0.53	131.46	20.91	26.94	34.93	0.9760	0.9951
2.	24.10	0.33	134.88	24.35	26.83	34.26	0.9754	0.9943
3.	30.17	0.265	136.44	25.79	26.78	34.02	0.9751	0.9940
4.	33.65	0.24	137.30	26.27	26.75	33.93	0.9750	0.9938
5.	39.15	0.20	140.61	28.57	26.65	33.57	0.9743	0.9933
6.	43.65	0.18	145.62	33.72	26.50	32.85	0.9734	0.9921

A.2.4 For IMAGE 5 (Abdomen):

i. Contextual -JPEG:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	23.81	0.34	54.41	14.37	30.77	36.56	0.9898	0.9963
2.	35.15	0.23	103.96	21.28	27.96	34.85	0.9804	0.9945
3.	42.40	0.19	196.81	24.37	25.19	34.26	0.9671	0.9937
4.	46.36	0.172	229.16	29.62	24.53	33.42	0.9615	0.9923
5.	51.50	0.155	310.22	33.06	23.21	32.94	0.9518	0.9914

ii. Contextual JPEG 2000:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	22.62	0.35	43.63	17.37	31.73	35.73	0.9918	0.9956
2.	34.91	0.23	77.48	34.16	29.24	33.80	0.9855	0.9932
3.	40.5	0.198	110.76	30.17	27.69	33.33	0.9791	0.9922
4.	47.14	0.17	160.48	30.61	26.08	33.27	0.9696	0.9921
5.	51.24	0.156	197.43	31.54	25.18	33.14	0.9625	0.9918
6.	60.51	0.132	221.12	38.12	24.69	32.32	0.9579	0.9901

iii. Contextual Wavelet:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	21.43	0.37	55.77	12.74	30.66	37.08	0.9896	0.9967
2.	32.12	0.25	66.35	15.09	29.91	36.34	0.9875	0.9961
3.	40.16	0.20	84.86	17.81	28.84	35.62	0.9840	0.9954
4.	46.21	0.173	88.85	22.08	28.64	34.69	0.9833	0.9943
5.	51.06	0.156	95.22	23.98	28.35	34.33	0.9821	0.9938

iv. Contextual SPIHT:

Sr. No.	CR	bpp	MSE		PSNR, dB		CoC	
			Entire Image	RoI	Entire Image	RoI	Entire Image	RoI
1.	15.15	0.53	134.02	15.99	26.86	36.09	0.9747	0.9959
2.	24.10	0.33	137.40	19.25	26.75	35.28	0.9740	0.9950
3.	30.17	0.265	139.71	21.16	26.68	34.88	0.9736	0.9945
4.	33.65	0.24	140.55	21. 57	26.65	34.79	0.9734	0.9944
5.	39.15	0.20	142.95	23.98	26.57	34.33	0.9730	0.9938
6.	43.65	0.18	146.14	27.20	26.48	33.79	0.9721	0.9930

Appendix B

B.1 Important Matlab Functions Used

appcoef2	A = appcoef2(C,S, 'wname',N)	It computes the approximation coefficients at level N using the wavelet decomposition structure [C,S]
blkproc	B = blkproc(A,[m n],fun)	It processes the image A by applying the function fun to each distinct m-by-n block of A, padding A with zeros if necessary. fun is a function that accepts an m-by-n matrix, x, and return a matrix, vector, or scalar y.
col2im	A = col2im(B,[m n],[mm nn], block_type)	It rearranges matrix columns into blocks. A = col2im (B,[m n],[mm nn],'distinct') rearranges each column of B into a distinct m-by-n block to create the matrix A of size mm-by-nn.
dctmtx	D = dctmtx(n)	It returns the n-by-n DCT (discrete cosine transform) matrix. D*A is the DCT of the columns of A and D'*A is the inverse DCT of the columns of A (when A is n-by-n).
dwt2	[cA,cH,cV,cD] = dwt2(X, 'wname')	The dwt2 command performs single-level two-dimensional wavelet decomposition. It computes the approximation coefficients matrix cA and details coefficients matrices cH, cV, and cD (horizontal, vertical, and diagonal, respectively), obtained by wavelet decomposition of the input matrix X. The 'wname' string contains the wavelet name.
detcoef2	D = detcoef2(O,C,S,N)	It extracts from the wavelet decomposition structure [C,S], the horizontal, vertical or diagonal detail coefficients for O = 'h'(or 'v' or 'd', respectively), at level N.

floor	B = floor(A)	It rounds the elements of A to the nearest integers less than or equal to A. For complex A, the imaginary and real parts are rounded independently.
hist	N = hist(y)	It bins the elements of Y into 10 equally spaced containers and returns the number of elements in each container. If Y is a matrix, HIST works down the columns.
histc	n = histc(x, edges)	It counts the number of values in vector x that fall between the elements in the edges vector (which must contain monotonically non-decreasing values).
im2double	I2 = im2double(I1)	It takes an image as input, and returns an image of class double. If the input image is of class double, the output image is identical to it. If the input image is of class logical, uint8 or uint16, im2double returns the equivalent image of class double, rescaling or offsetting the data as necessary.
immultiply	Z = immultiply(X,Y)	Z = immultiply(X,Y) multiplies each element in the array X by the corresponding element in the array Y and returns the product in the corresponding element of the output array Z.
ifft2	B = ifft2(A)	B = ifft2 (A) returns the two-dimensional inverse fast Fourier transform of matrix A. If A is a vector, B has the same orientation as A.
roipoly	BW = roipoly (I)	BW = roipoly (I) displays the image I on the screen and lets you specify the polygon using the mouse. If you omit I, ROIPOLY operates on the image in the current axes. Use normal button clicks to add vertices to the polygon. Pressing <BACKSPACE> or <DELETE> removes the previously selected vertex. A

		shift-click, right-click, or double-click adds a final vertex to the selection and then starts the fill; pressing <RETURN> finishes the selection without adding a vertex.
rgb2gray	I2 = rgb2gray (I1)	RGB2GRAY converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.
wcodemat	Y = wcodemat(X, NBCODES, OPT, ABSOL)	It returns a coded version of input matrix X. if ABSOL = 0, or ABS(X) if ABSOL is nonzero, using the first NBCODES integers. Coding can be done row-wise (OPT = 'row' or 'r'), column wise (OPT = 'col' or 'c'), or globally (OPT = 'mat' or 'm').
wavedec2	[C,S] = wavedec2(X,N, 'wname')	It is a two-dimensional wavelet analysis function. [C,S] = wavedec2(X,N,'wname') returns the wavelet decomposition of the matrix X at level N, using the wavelet named in string 'wname'
wrcoef2	X = wrcoef2('type',C,S, 'wname',N)	It computes the matrix of reconstructed coefficients of level N, based on the wavelet decomposition structure [C, S].
wdencmp	[XC, CXC, LXC, PERF0, PERFL2] = Wdencmp ('gbl', X, 'wname', N, THR, SORH, KEEPAPP)	It returns a de-noised or compressed version XC of input signal X (one- or two-dimensional) obtained by wavelet coefficients thresholding using global positive threshold THR. Additional output arguments [CXC, LXC] are the wavelet decomposition structure of XC. PERF0 and PERFL2 are L2-norm recovery and compression score in percentage.
wfilters	[Lo_D,Hi_D,Lo_R,Hi_R] = wfilters ('wname')	It computes four filters associated with the orthogonal or biorthogonal wavelet named in the string 'wname'.
		waverec2 performs a multilevel 2-D wavelet

waverec2	X = waverec2(C, S,'wname')	reconstruction using either a specific wavelet ('wname') or specific reconstruction filters (Lo_R and Hi_R). X = waverec2(C, S,'wname') reconstructs the matrix X based on the multi-level wavelet decomposition structure [C, S].

References

[1] M. Nadir Kurnaz, Z. Dokur and T. Olmez: "Compression of MR and Ultrasound Images by Using Wavelet Transform", Proceedings of the Second Joint EMBS/BMES IEEE Conference, Houston, pp 1021-1022, October 23-26, 2002.

[2] Stephen Wong et al.: "Radiologic Image Compression- A Review", IEEE Transactions on Medical Imaging, Vol. 83, No. 2, February 1995.

[3] Xiaolin Wu and Nasir Memon: "Context-Based, Adaptive, Lossless, Image Coding", IEEE Transactions on Communications, Vol. 45, No.4, April 1997.

[4] Marcelo J. Weinberger, Gadiel Seroussi and Guillermo Sapzro: "LOCO-I: A Low Complexity, Context-Based, Lossless Image Compression Algorithm", IEEE Transactions on Image Processing, Vol. 6, No. 10, 1996.

[5] V. N. Ramaswamy, K. R. Namuduri, and N. Ranganathan: "Context-Based Lossless Image Coding Using EZW Framework", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 4, April 2001.

[6] Farshid Golchin and Kuldip K. Paliwal:" "A Context-Based Adaptive Predictor for Use in Lossless Image Coding", IEEE TENCON - Speech and Image Technologies for Computing and Telecommunications, 1997.

[7] V.Vlahakis and R.I. Kitney: "Wavelet-Based, Inhomogeneous, Near-Lossless Compression of Ultrasound Images of the Heart", IEEE Transactions on Computers in Cardiology, 1997.

[8] H. Yang, M. Long and H.-M. Tai: "Region-of-interest image coding based on EBCOT", IEE Proceedings on Visual Image Signal Processing, Vol. 152, No. 5, October 2005.

[9] Atsumi E. and Farvardin N.: "Lossy/lossless region-of-interest image coding based on set partitioning in hierarchical trees" Proceedings of IEEE International Conference of Image Processing, pp. 87-91, Chicago, USA, 1998.

[10] Park K. and Park H.W.: "Region-of-interest coding based on set partitioning in hierarchical trees", IEEE Transactions on Circuits Systems Video Technology, pp. 106–113, Vol.12, No. 2, 2002.

[11] Christopoulos C., Askelof J. and Larsson M.: "Efficient methods for encoding regions of interest in the upcoming JPEG 2000 still image coding standard", IEEE Signal Processing Letters, pp. 247–249, Vol. 7, No. 9, 2000.

[11] Christopoulos C., Askelof J. and Larsson M.: "Efficient methods for encoding regions of interest in the upcoming JPEG 2000 still image coding standard", IEEE Signal Processing Letters, pp. 247–249, Vol. 7, No. 9, 2000.

[12] Askelof J., Carlander M., and Christopoulos C.: "Region of interest coding in JPEG 2000", Signal Processing Image Communication, pp. 105–111, Vol.17, 2002.

[13] R.C. Gonzalez and R.E. Woods: "Digital Image Processing", Pearson Education, Second Edition.

[14] V.Vlahakis and R.I. Kitney: "Multi-Region Semi-Automatic JPEG Based Medical Image Compression", 18th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Amsterdam, pp 1220-1221, 1996.

[15] Xu Yan et al.: "The Coding Technique of Image with Multiple RoIs using Standard Maxshift Method", The 30th Annual Conference of the IEEE Industrial Electronics Society, Busan, Korea, pp 2077-2080, 2004.

[16] Kazuhiko Hamamoto and Toshihiro Nishimura: "Basic Investigation on Medical Ultrasonic Echo Image Compression by JPEG 2000- Availability of Wavelet Transform and RoI Method", Proceedings of the 23rd Annual EMBS International Conference, Istanbul, pp 2049-2052, 2001.

[17] Andras Cziho et al.: "Medical Image Compression Using Region-of-Interest Vector Quantization", Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp 1277-1280, Vol. 20, No.3, 1998.

[18] V.Vlahakis and R.I. Kitney: "ROI Approach to Wavelet Based, Hybrid Compression of MR Images", IEE International Conference on Image and Pattern Analysis, pp 833-837, 1997.

[19] Daniel T. Lee: "JPEG 2000: Retrospective and New Developments", IEEE Transactions on Image Processing, Vol. 93, No.1, January 2005.

[20] M. Rabbani and R. Joshi: "An Overview of the JPEG 2000 Still Image Compression Standard", Signal Processing Image Communication, Vol. 17, pp 3-48, January 2002.

[21] M. Gormish, D. Lee and M. W. Marcellin: "JPEG 2000: Overview, architecture and applications", Proceedings of the IEEE International Conference of Image Processing, Vancouver, September 2000.

[22] M. W. Marcellin et al.: "An Overview of JPEG 2000", Proceedings of the Data Communication Conference, pp 523-541, 2000.

[23] D. Santa-Cruz and T. Ebrahimi: "An analytical study of JPEG 2000 functionalities", Proceedings of the IEEE International Conference on Image Processing, Vol. 2, pp 49-52, 2000.

[24] C. Christopoulos, A. Skodras and T. Ebrahimi: "The JPEG 2000 still image coding system: An overview", IEEE Transactions on Consumer Electronics, Vol. 46, No. 4, pp 1103-1127, November 2000.

[25] A. Said and W. A. Pearlman: "A new, fast and efficient image codec based on set partitioning in hierarchical trees", IEEE Transactions on Circuits Systems Video Technology, Vol. 6, pp 243-250, January 1996.

[26] Alexander E. Mohr, A. Riskin and Richard E. Lander: "Unequal Loss Protection: Graceful Degradation of Image Quality over Packet Erasure Channels Through Forward Error Correction", IEEE Journal on Selected Areas in Communications, Vol. 18, No. 6, June 2000.

[27] Ki-Lyung Kim and Sung Woong Ra: "Performance Improvement of the SPIHT Coder", Signal Processing: Image Communication, Vol. 19, pp 29-36, 2000.

[28] Ahmed Abu-Hajar and Ravi Sankar: "Wavelet Based Lossless Image Compression Using Partial SPIHT and Bit plane Based Arithmetic Coding", ICASSP 2002, Vol.4, pp 3497-3500, Orlando, Florida, 2002.

[29] Ahmed Abu-Hajar and Ravi Sankar: "Enhanced Partial-SPIHT for Lossless and Lossy Image Compression", ICASSP 2003, Vol.3, pp 253-256, 2003.

[30] Debargha Mukherjee and Snajit K. Mitra: "Vector SPIHT for Embedded Wavelet Video and Image Coding" IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 3, March 2003.

[31] Salih Barah Gokturk et al.: "Medical Image Compression Based on Region of Interest, With Application to Colon CT Images", IEEE Transactions on Image Medical Imaging, 2001.

[32] Jacob Strom and Pamela C. Cosman: "Medical image compression with lossless regions of interest", Elsevier Signal Processing, Vol. 59, pp 155-171, 1997.

[33] Liang Sheng and Rangraj M. Rangayyan: "A Segmentation-based Lossless Image Coding Method for High Resolution Medical Image Compression", IEEE Transactions on Medical Imaging, Vol. 16, No.3, June 1997.

[34] Lin Z.: "3-D Magnetic Resonance Image Segmentation and Medical Image Compression", Ph.D. Thesis of University of New South Wales, Sydney, http://www.unsw.edu

Web Resources:

[35] http://www.jpeg.org/apps/medical.html

[36] http://www.jpeg.org/jpeg2000/index.html

[37] http://www.cs.sfu.ca/CourseCentral/365/li/material/notes/chap4.html

[38] http://www.pascalzine.amirmelamad.co.il/Graphics/JPEG/JPEG.htm

[39] http://www.164.241.2.51/ntb/2002SICS/05NextGen.pdf

[40] http://www.wikipedia.org/wiki/JPEG2000

[41] http://www.math.wustl.edu/wavelet/

[42] http://www.amara.com/current/wavelet.html

[43] http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html

[44] http://gvsu.edu/math/wavelets/student_work/EF

[45] http://www.cipr.rpi.edu/research/SPIHT

List of Publications

1. Rahul K. Kher: "Comparative Study of Wavelet and JPEG Methods of Image Compression for Ultrasound Medical Images", Proceedings of the National Conference on Information and Emerging Technologies (NCIET 2006), Instt. of Engg. & Tech, Bhaddal, Ropar, February 17-18, 2006.
2. Rahul K. Kher: "Compression of Ultrasound Medical Images", Proceedings of the National Conference on Emerging Trends in Computer and Electronics Engineering for Rural Development, Chhatrapati Shivaji Instt. of Tech, Durg, March 4-5, 2006.

Selection Awaited:

3. Rahul K. Kher, R.S. Anand: "Contextual Compression of Ultrasound Medical Images", National Conference on Signal Processing and Communication, Jawaharlal Nehru National Engineering College, Shimoga, Karnataka, July 7-8, 2006.
4. Rahul K. Kher and R.S. Anand: "Ultrasound Medical Image Compression Using Contextual Approach", 9th International Conference on Information Technology (CIT 2006), Bhubaneswar, December 18-21, 2006.