PERFORMANCE ANALYSIS OF DIFFERENT TYPES OF SOURCE CODING TECHNIQUES

Varun Chadha*

Naresh Kumar*

Amit Prakash*

Navneet Kaur*

ABSTRACT

This research paper explains the performance analysis of source coding techniques on text and image data. The advent in communication technologies has lead to an enormous growth of human life. Internet now a day is finest example of development of Communication system. The Bandwidth requirement by users is increasing but Bandwidth is still same (practically). So, data has to be "cut down" in effective and efficient way so that we can store our message in less space and using less bandwidth. We have various Source Coding Techniques for this purpose. These source coding techniques can be applied to different types of data. In this paper, we have applied Huffman coding and RLE coding to same text data and compared the performance of both techniques. We have chosen compression ratio as performance parameter. Also, DCT compression and Wavelet compression techniques are applied to same image data and compared the performance of both techniques. Based on result of performance of these techniques pros and cons of these techniques are discussed in this paper.

Keywords: *Source Coding, RLE, Huffman, DCT Compression, Wavelet Compression, PSNR*

*Department of Electronics & Communication Engineering, Lovely Professional University, Phagwara, India

I. INTRODUCTION

Communication, in simple terms, is the sharing of information between a sender and a receiver. Communication requires a sender, a message, and an intended recipient. A complete block diagram of communication system consists of source encoder. Source coding is used for the compression of data. Compression is the process of reducing the size of a file by encoding its data information more efficiently. By doing this, the result is a reduction in the number of bits and bytes used to store the information. In effect, a smaller file size is generated in order to achieve a faster transmission of electronic files and a smaller space required for its downloading. Compression is done by using compression algorithms that rearrange and reorganize data information so that it can be stored more economically. By encoding information, data can be stored using fewer bits. It lets you store more stuff in the same space, and it lets you transfer that stuff in less time, or with less bandwidth. Text data can be compressed using Run Length Encoding (RLE) and Huffman encoding technique. Image data can be compressed using JPEG and JPEG 2000. All compression methods are compared on the basis of compression ratio and compression quality. Compression ratio is the measurement of compressed data and compression quality is determined by calculating the PSNR (Peak Signal to Noise Ratio) is one of the quantitative measures for image quality evaluation which is based on the mean square error (MSE) of the reconstructed image.

A. Types of Compression

Lossless Compression is a type of compression that can reduce files without a loss of information in the process. The original file can be recreated exactly when uncompressed. To achieve this, algorithms create reference points for things such as textual patterns, store them in a catalogue and send them along with the smaller encoded file. When uncompressed, the file is "re-generated" by using those documented reference points to re-substitute the original information. Lossless compression is ideal for documents containing text and numerical data where any loss of textual information can't be tolerated. ZIP compression, for instance, is a Lossless compression that detects patterns and replaces them with a single character. Another example, LZW compression works best for files containing lots of repetitive data. Some examples of Lossless Compression Algorithms are RLE, Huffman encoding, LZW encoding techniques

Lossy Compression reduces the size of a file by eliminating bits of information. It permanently deletes any unnecessary data. This compression is usually used with images, audio and graphics where a loss of quality is affordable. However, the original file can't be retained. For instance, in an image containing a green landscape with a blue sky, all the different and slight shades of blue and green are eliminated with compression. The essential nature of the data isn't lost-the essential colors are still there. One popular example of Lossy compression is JPEG compression that is suitable for grayscale or color images.

II. DIFFERENT COMPRESSION METHODS FOR TEXT DATA

Text data can be compressed with various source coding techniques like Run Length Encoding (RLE) and Huffman encoding technique. Compression can be achieved by using these techniques and can be compared using compression ratio.

A. RLE Compression

Run-length coding (RLE) is a very simple form of data compression in which runs of data are stored as a single data value and count, rather than as the original run. This is most useful on data that contains many such runs.

Suppose data given is: 000000001111111111000000111………If we apply the RLE data compression algorithm to the above. It could be represented as: (0,8),(1,10),(0,6),(1,3)..

COMPRESSION RATIO=20:27

Data files frequently contain the same character repeated many times in a row e.g. text files use multiple spaces to separate sentences, indent paragraphs, format tables & charts, etc. Digitized signals can also have runs of the same value, indicating that the signal is not changing. For instance, an image of the night-time sky would contain long runs of the character or characters representing the black background. Likewise, digitized music might have a long run of zeros between songs. Run-length encoding is a simple method of compressing these types of files. Advantage of using this algorithm is that it is very easy to implement and does not require much CPU horsepower. But its disadvantage is that it is not useful with files that don't have many runs as it could greatly increase the file size.

B. Huffman encoding

In computer science and information theory, Huffman coding is an entropy encoding algorithm used for lossless data compression. The term refers to the use of a variable-length code table for encoding a source symbol where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. It was developed by David A. Huffman while he was a Ph.D. student at MIT, and published in the 1952 paper "A Method for the Construction of Minimum-Redundancy Codes". A Huffman code is designed by merging together the two least probable characters, and repeating this process until there is only one character remaining. A code tree is thus generated and the Huffman code is obtained from the labeling of the code tree. An example of how this is done is shown below in fig.1

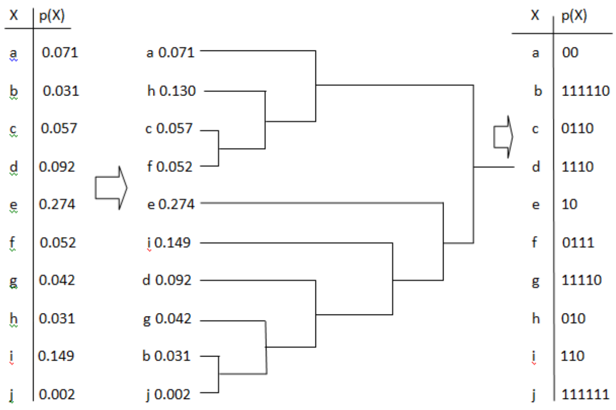

Fig. 1 Huffman tree

Steps for Huffman Coding:

1. Arrange the source symbols in a decreasing order of their probability.
2. Take the bottom two symbols and tie them together. Add probabilities of two symbols and write it on the combined node. Label the two branches with a ‘1’ or ‘0’ as shown in figure below.
3. Treat this probability as a new probability of combined symbol and repeat previous step till we have a single symbol, and whenever we tie together two probabilities we label the two branches with a ‘1’ or ‘0’.
4. This will complete the construction of Huffman tree.
5. To find out the prefix code for any symbol follow the branches from the final node back to the symbol. While tracking back the route read out the labels on the branches. This is the codeword for the symbol.

Although Huffman's original algorithm is optimal for a symbol-by-symbol coding (i.e. a stream of unrelated symbols) with a known input probability distribution, it is not optimal when the symbol-by-symbol restriction is dropped, or when the probability mass functions are unknown.

III. IMAGE COMPRESSION

Digital images are characterized by multiple parameters. The first feature of a digital image is its colour mode. A digital image can have one of three modes: binary, gray scale or colour. A binary (bi-level) image is an image in which only two possible values for each pixel. A gray scale image means that its each pixel can contain only a tint of gray colour. A digital image is a set of pixels. Each pixel has a value that defines colour of the pixel. All the pixels are composed into one array. There are several different ways in which image files can be compressed.

A. Different Compression Methods for Image data

Image compression was first used to deal with the limited memories of the first personal computers and the large file size of colour images. Compression algorithms were proposed and developed.

1) Image Compression: JPEG

JPEG is well known standard when we are concerned to images. Joint Pictures Expert Group developed a lossy image source coding technique which came into existence as JPEG in 1992. JPEG can work with grayscale as well as color images. JPEG can compress image to as little as two to five percent of its original file size, but as the degree of compression increases, there is a corresponding loss of image quality.

JPEG involves a number of steps which are as follows:

I. Discrete Cosine Transform
II. Quantization
III. Zigzag scan
IV. Encoding
 a. DPCM on DC Components
 b. RLE on AC components
V. Entropy Coding

All of the above steps ensure the compression to be fully effective, but most of the compression is done in the first three steps. In fact, DCT i.e. Discrete Cosine Transform is the most important process used in JPEG. The reason why the steps IV and V are not "so important" is because of the fact that these steps are merely compression techniques which work on some kind of redundancies/similarity in the pixels and we can use any of the coding techniques.

So our main concern, here, is the first three steps of JPEG. We can compress an image by applying DCT to the pixels of the image and then quantizing it. Although, most of the compression is done by the quantization itself but it is DCT who makes the quantization very effective in terms of image compression.

2) Discrete Cosine Transform

Discrete Cosine Transform expresses a sequence of finitely many data points, in our case pixels, in terms of cosine function oscillating at different frequencies. DCT converts the data values in term of spatial frequencies. Each spatial frequency indicates the number of times the pixel value changes in a data block, usually 8×8 block. DCT formalizes spatial frequencies by measuring how much the image content i.e. the pixel values changes according to the number of cosine cycles per block.

Discrete Cosine Transform is given mathematically as:

$$F(u,v) = \frac{1}{4}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f(i,j)C(u)C(v)\cos\frac{(2i+1)u\pi}{2N} + \mathrm{Cos}\frac{(2j+1)v\pi}{2N}$$

Where C (u) and C (v) = $\begin{cases} \sqrt{\frac{1}{N}} & for\ j,k = 0 \\ \sqrt{\frac{2}{N}} & for\ j,k \neq 0 \end{cases}$

f (i, j) is original pixel coefficient at coordinates (i, j) in the N×N block of image.

F (u, v) is the DCT coefficient at coordinates (u, v).

DCT concentrate the useful information of image in first several component of the block, usually in the upper-left part of the Discrete Cosine Transformed block.

F(0, 0) is called AC coefficient and all other coefficients are called DC coefficients.

The principal cause of using DCT in image compression is the way humans see an image. It is a proved phenomenon that human eyes are less sensitive to the high spatial frequencies. The content which is useful changes relatively slow. Usually the high spatial frequencies correspond to the borders/outlines of the objects in the image. So we can remove the high spatial frequency components, or try to lessen their quantity, to compress the image. This does not change the image significantly in terms of the quality. But it greatly helps in reducing the file size when quantization occurs.

Quantization is the main source of compression in this whole process. It will generate $F'(u,v)$ mathematically as:

$$F'(u,v) = \frac{F(u,v)}{Q(u,v)}$$

F(u,v) is the Discrete Cosine Transformed entry at coordinates (u, v), also called as DCT coefficient. Q(u,v) is a *Quantization Table* entry at coordinates (u, v).

As we can decide the value of Q (u, v), we can achieve a variable compression. Increasing the value of Q (u, v) will increase the compression and decreasing Q(u,v) will decrease the compression.

We can choose the value of the Q(u,v) by ourselves. However in JPEG it is predefined as shown in the fig. 2.

After quantization a zigzag scan is performed just to efficiently convert the DCT coefficients in a bit stream.

Based on the *PSNR* (Peak Signal to Noise Ratio) of the compressed images, we conclude the pros and cons of DCT image compression, *and jpeg too.*

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

Fig 2. Quantization matrix

Based on the *PSNR* (Peak Signal to Noise Ratio) of the compressed images, we conclude the pros and cons of DCT image compression, *and jpeg too.*

One of main advantages of DCT compression is that the fidelity of an image does not change much when the scale of quantization is within the limits.

Compression up to any level can be achieved. i.e. we can reduce the size of image even up to 2% of the original size. But as compression is increased *psnr* also increases simultaneously which means a high image noise. Although we have done analysis of DCT compression applying it on grayscale images, but it can also be applied to RGB images. On RGB, its performance will be better as compared to grayscale.

This image compression is also flexible as it can be implemented using any types of hardware and software. Also, as the output after the zigzag scan is a bit stream it will be easy to serially transmit the image to other devices.

It's a lossy compression method, so we cannot exactly reconstruct the image. So it cannot be used to compress critically important data like medical imaging.

3) Wavelet Compression

Wavelet compression is relatively new method. It is powerful tool for compressing information. This compression technology is used in the JPEG-2000 standards. Images compressed using wavelets are smaller than JPEG images and can be transferred faster. Wavelet technology can compress color images from 20:1 to 300:1 and grey scale images from 10:1 to 50:1.

Wavelet transformation is similar in principal to the DCT based compression. The main difference is the use of wavelet based mathematical technique in place of DCT transformations. In contrast to JPEG, wavelet algorithm does not divide an image into blocks, but analyze the whole image. This characteristic of wavelet compression helps to get the best compression ratio, while maintaining the quality.

Procedure: Wavelets represent pictures as wave that can be described mathematically in terms of frequency, energy and time. They have advantages over traditional Fourier and DCT methods in analyzing signals that have discontinuities and spikes. The advanced mathematics underlying wavelet transformation is very complex. Therefore, for the purpose we will introduce the concept in very simple terms without using mathematical formulations.

The steps needed to compress an image using wavelet transformation are as follow:

1) The source image is digitized into a signal, a string of numbers 's'.
2) The signal is then decomposed into sequence of wavelet coefficients 'w'.
3) Thresholding is used to modify the wavelet coefficients from w to another sequence w'.
4) Using quantization the sequence w' is converted to a sequence q.
5) Finally entropy coding is applied to compress q into a sequence e.

Example: For the purpose of understanding wavelet compression, assume we have a one-dimensional image consisting of just four pixels in a row. Assume that the four pixels in row have different shades of grey of value: 3, 5, 9, 7.

We could take the average of the first pair and then the average of the second pair to give: 4, 8. Here in effect, we have simplified the image by halving its resolution. But in the process, some information is lost. In order to restore the original values we would need to record the differences as well as the averages. These are known as the 'detail coefficients'. The detail coefficients are added to and subtracted from averages in order to reconstruct the original values.

For the case we assumed, the detail coefficients are -1 and 1; calculated as: (3 -5)/2 and (9-7)/2. Thus from the second pair we get back the original values: 4+ -1=3; 4- -1=5; 8+1=9; 8-1=7.

We can keep on decomposing the image in a similar way by taking the averages and noting the average differences. Soon we will arrive at one overall average and one set of detail coefficients. In this simple example, the image can be decomposed once more, giving the values: overall average 6 and the difference 2.

Resolution	Averages	Detail coefficients
4 (original image)	3,5,9,7	
2 (half resolution)	4,8	-1,1
1 (coarsest image)	6	2

Table1. DWT coefficients

Thus, if we recorded the overall average (6) and the detail coefficients, we would have enough information to reconstruct the image at half resolution (4,8) or full resolution (3,5,9,7). The data therefore will be: 6;2,-1,1.(Average:6; Detail coefficients:2,-1,1).

The image described by this information is lossless, since it can b reconstructed exactly as it was. It will be lossy if we keep the significant detail and throw away other detail coefficients of the image. This would produce smaller files that appear as good as original when decomposed. The result would be a lossy compressed image.

4) Discrete wavelets transform:

In discrete wavelet transformation (DWT) method an image is represented as a two dimensional array of coefficients, where each coefficient represents the brightness level of a point or pixel of that image. Most natural images have smooth color variations (low frequency variations). From compression point of view, the low frequency components are of more importance. DWT plays a very important role in extracting these low frequency components.

The DWT of a signal is obtained by passing it through two filters that are related to each other. Such filter-pair is known as analysis filter. When a digitized signal is thus filtered, half the samples can be discarded according to Nyquist's rule. So the filter outputs are down sampled by 2. The signal is passed simultaneously through a low pass filter and high pass filter. The low pass filter (LPF) gives the approximation coefficients (averages) and high pass filter (HPF) gives the detail coefficients. The approximation coefficients (outputs of the first level LPF) can be further decomposed with the second level analysis filter. Adding more and more levels, a filter bank can be constructed that can look like a binary tree.

DWT of an image: in order to obtain DWT of an image, first the lpf and HPF are applied for each row data. The procedure is repeated for all the rows and all low pass and high pass components are collected and placed side by side. Thereafter, the filtering is repeated for each column of the intermediate data. The resulting two-dimensional array of coefficients contain 4 bands of data labeled as LL(low-low), HL,LH and HH(high-high).repeated transform procedure produce the pyramidal decomposition of the image.

The LL band at the top of the pyramid is of most importance in representing the image and the degree of importance decreases as we move down.

To reconstruct the image a reverse transform of DWT is used to reassemble the various segments of data. For this reverse transformation a synthesis filter pair is used, which functions exactly as the opposite to the analysis filter. Starting from the topmost level, the filter actions are applied column-wise first and then the row-wise.the DWT of a picture can be obtained easily by software written in either Java or C language. MATLAB can also be used.

Digital images are characterized by multiple parameters. The first feature of a digital image is its colour mode. A digital image can have one of three modes: binary, gray scale or colour. A binary (bi-level) image is an image in which only two possible values for each pixel. A gray scale image means that its each pixel can contain only a tint of gray colour. A digital image is a set of pixels. Each pixel has a value that defines colour of the pixel. All the pixels are composed into one array. There are several different ways in which image files can be compressed.

IV. IMPLEMENTATION

Compression methods mentioned for text and image data are implemented in MATLAB. RLE and Huffman coding methods are applied to the same text data and DCT compression and Wavelet compression is applied to same image data.

Following fig. represents the implementation of RLE and Huffman encoding technique on text data and compression ratio is calculated.

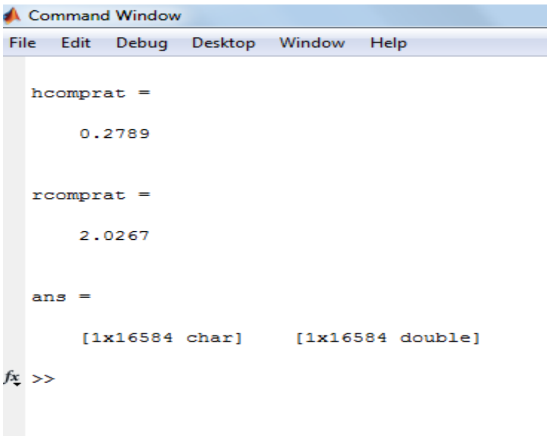

Fig. 3 Compression of text data using RLE and Huffman coding

Graphical representation of compression ratio of text data compression is shown in figure below.

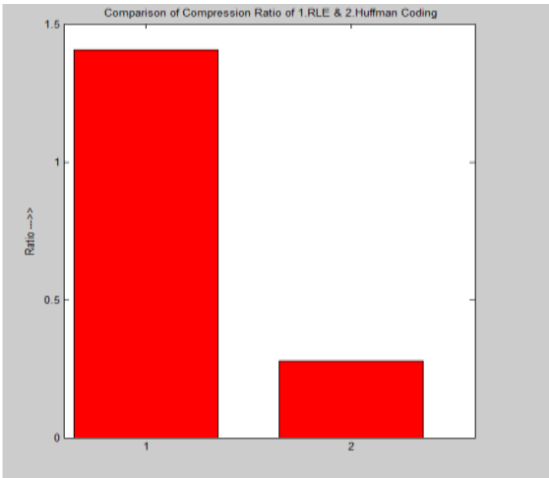

Fig.4 Comparison of Compression Ratio

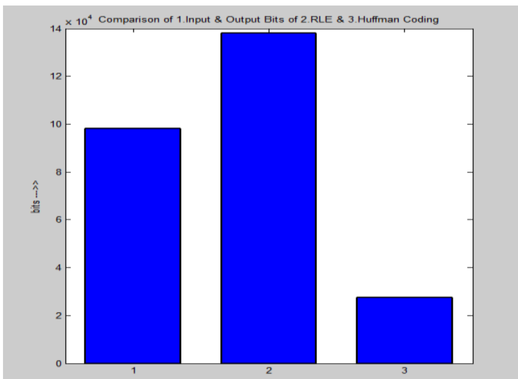

Fig. 5 Comparisons of Input and Output Bits of RLE and Huffman

Following figures represents the implementation of DCT compression technique.

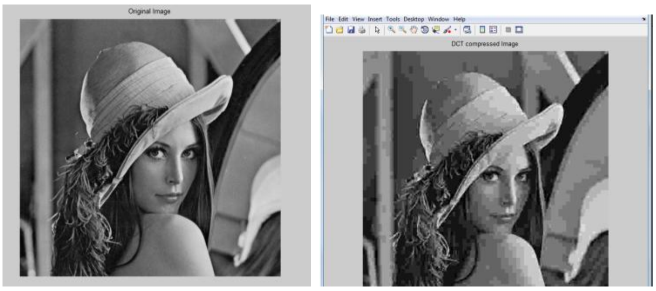

Fig. 6 Compressed image using DCT

Following figures represents the implementation of wavelet compression technique.

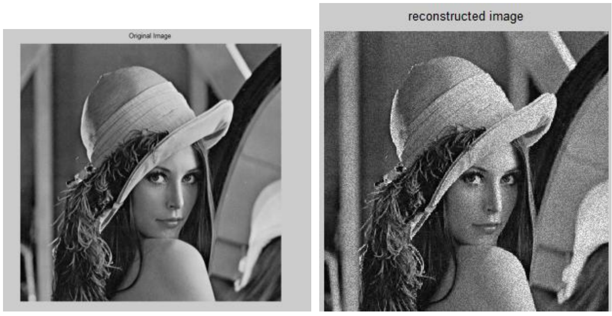

Fig. 7 Compressed image using wavelet

V. CONCLUSION

In this thesis Comparative analysis of compression methods is carried out. Comparative analysis is done on the basis of two parameters compression ratio and compression quality. Our results illustrate that we can achieve higher compression ratio for Huffman encoding. RLE compression is not better for data which does not contain long runs but Huffman encoding is showing better result with any kind of text data. Wavelet compression technique shows better result as compared to the DCT compression technique and gives better compression ratio and PSNR value.

VI. FUTURE WORK

In future this thesis can be extended by comparing compression methods that are mentioned in this thesis with the Image segmentation. Comparison can be made on the basis of memory space taken by Compressed image and Segmented image. Segmentation refers to the process of partitioning a digital image into multiple segments (sets of pixels) (Also known as super pixels). The goal of segmentation is to simplify and change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. The goal of image segmentation is to cluster pixels into salient image regions, i.e., regions corresponding to individual surfaces, objects, or natural parts of objects

REFERENCES

1. **S.** W. Golomb, "Run-Length Encoding," *IEEE Trans. Info. Theory* IT-12, 399-401 (July 1966)
2. S. Akhter and M.A. Haque. ECG Compression using Run Length Encoding. European signal processing (EUSIPCO-2010), pp. 1645-1649, August 23-27, 2010
3. R.G. Gallager, "Variations on a theme by Huffman", IEEE Trans. Information Theory, Vol. IT-24, no 6, pp 668-674, Nov. 1978
4. O. Johnsen, "On the redundancy of binary Huffman codes", IEEE Trans. Inform.. Theory, Vol. IT-26, no 2, pp 220-223, Mar 1980
5. Ashok Banerji, Ananda Mohan Ghosh "Multimedia Technologies", by Tata McGraw Hill, pp 79-81
6. X. Kavousianos, E. Kalligeros, and D. Nikolos, "Optimal Selective Huffman Coding for Text-Data Compression" IEEE Transactions on Computers, vol.56, pp. 1146-1152, Aug 2007
7. Se-Kee Kil, Jong-Shill Lee, Dong-Fan Shen, Je-Goon Ryu, Eung-Hyuk Lee, Hong-Ki Min, Seung-Hong Hong "Lossless Medical Image Compression using Redundancy Analysis" , IJCSNS International Journal of Computer Science and Network Security, Vol.6 No.1A, pp. 50-57, January 2006
8. Digital Compression and coding of Continuous-tone still images, part 1, requirements and Guidelines. ISO/IEC JTC1 Draft International Standard 10918-1, Nov. 1991
9. N. Ahmed, T. Natrajan, and K. R. Rao, "Discrete Cosine Transform", IEEE Transactions on Computers, vol. 23, July 1989
10. Lewis AS, Knowles G Image compression using the 2-d wavelet transform IEEE trans image processing vol.1 no 2 april 1992
11. Monro, D.M and Sherlock, B.G., 'Optimal quantization statergy for DWT image compression' IEEE proceedings- Vision, Image and Signal processing 1999