© 2015 IJSRSET | Volume 1 | Issue 3 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

JPEG 2000 Still Image Data Compression

Shashikumar N*1, Choodarathnakara A L2, Prathap Kumar B R3, Ashika V B4, Raghunandan G5

[*1,2,3]Dept. of Electronics & Communication Engineering, GEC, Kushalnagar, Kodagu, Karnataka, INDIA
[4]Dept. of Computer Science & Engineering, GEC, Kushalnagar, Kodagu, Karnataka, INDIA
[5]Department of Electronics & Communication Engineering, PCE, Bangalore, Karnataka, INDIA

ABSTRACT

In the last few years technology advancement of digital processors and digital storage devices have resulted in dramatic reduction in the size and cost of digital memory and digital processing logic. The manipulation and analysis of digital pictorial information is referred to as digital image processing. Image data compression comes under the category of digital image processing. Image compression is the method of effectively coding digital images to reduce the number of bits required in representing an image. The purpose of doing so is to reduce the storage and transmission costs while maintaining good quality. There are two methods of compression, Lossy compression and Lossless compression. With the increasing use of multimedia technologies, Image compression requires higher performance as well as new features. To address this, need in the specific area of still image encoding, a new standard is used, the JPEG2000. It is interesting to note that JPEG2000 is being designed to address the requirements of a diversity of applications, e.g. internet, colour facsimile, printing, scanning, digital photography, remote sensing, mobile applications, medical imagery, digital library and E-commerce. We have implemented the basic architecture of the JPEG2000 using MATLAB. It can be used for both gray-scale and colour images.

Keywords: Digital Image Processing, Image Data Compression, Joint Photographic Experts Group (JPEG 2000), Discrete Wavelet Transform and MATLAB

I. INTRODUCTION

At the present state of technology, methods of compressing the multimedia data prior to storage and/or transmission are of significant practical and commercial interest. Original uncompressed multimedia contains huge amount of data. Image compression addresses the problem of reducing the data required to represent an image. The underlying basis of reduction process is the removal of redundant data present in the image. JPEG2000 algorithm is associated with removing/reducing the redundancies present in an image. In reducing the redundancy, capacity of Human Eye in interpreting an image plays the most important role JPEG2000 is a new image compression standard being developed by the Joint Photographic Experts Group (JPEG), part of the International Organization for Standardization (ISO). The methodology involves of both implementation compression and decompression of images using JPEG2000 algorithms. It is designed for different types of still images (bi-level, gray-level, color, multi component) 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 JPEG standards, without sacrificing performance at other points in the rate-distortion spectrum. It also provides features and functionalities that the current standard can either not address efficiently or in many cases cannot address at all. Lossless compression, Lossy compression, Progressive transmission by pixel accuracy, the Resolution, the Bit rates are some of the representative features.

JPEG2000 is designed to address the requirements of diversity of applications in colour facsimile, printing, scanning, internet, digital photography, remote sensing, picture transmission in mobiles, e-commerce and in medical field image processing plays a dominant role. From the above facts doing image compression work in image processing is apt.

II. LITERATURE SURVEY

To complete the work in a meaningful way it is necessary to do literature survey. Since, the work belongs to the area of Image Processing and Compression, it is necessary to understand the concept of compression techniques and Image processing. Hence, an extensive reference has been made to various textbooks and papers.

The basics of image processing are better understood by referring to R C Gonzalez and Woods [1]. Various transforms are used in image processing. Some of the image transforms are used in the area of image compression. They are DCT, Hadamard, Haar, Walsh, KLT and Wavelet transform. The basic concepts of image processing and multimedia application are explained by Ze-Nian Li and Mark S Drew [2].

Anil K Jain [3] gives the fundamental concepts of different Transformations used in image processing applications and the performance comparison of different image transforms are well explained. He also discusses the best image transform for a particular application. Ahmed N, Natarajan T and Rao K R have explained basics of Discrete Cosine Transforms, and its use in image compression [4].

The JPEG2000 Standard basics and algorithms are explained by Karen L. Gray [5]. The basics and implementation of Wavelet Transform Coding is explained in detail by Marc Antonini, Michel Hadamard, Barlaud, Pierre Mathieu, and Ingrid Daubechies, Touradj Ebrahimi3 [6].

Algorithms used for the compression using 2D Wavelet Transform by A. S. Lewis and G. Knowles [7]. The basic architecture implemented is explained in detail by Charilaos Christopoulos, Athanassios Skodras and Touradj Ebrahimi [8].

III. IMAGE DATA COMPRESSION

Image compression is concerned with minimizing the number of bits required to represent an image. The simplest form of image data compression is the sampling of band-limited images, where an infinite number of pixels per unit area are reduced to one sample.

A common characteristic of most images is that the neighbouring pixels are correlated and therefore contain redundant information. The foremost task then is to find less correlated representation of the image. Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver, namely the Human visual System (HVS). In general, three types of redundancy can be identified.

- **Spatial Redundancy** or correlation between neighbouring pixel values
- **Spectral Redundancy** or correlation between different colour planes or spectral bands
- **Temporal Redundancy** or correlation between adjacent frames in a sequence of images

The problem with using high quality images in data products is the amount of memory required in storage application and the amount of bandwidth required transmission application. This problem gives rise to the discovery of various image compression techniques.

In remote sensing satellites, lots of high resolution images have to be transmitted in real time that consumes a lot of bandwidth and time. In order to overcome these problems images are compressed and transmitted. At the receiving end (i.e. the ground station) they are decompressed to get back the image with some loss due to compression. It is possible to obtain high quality images using sophisticated compression algorithms.

Image compression research aims at reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies as much as possible. Since we will focus only on still image compression, we will not worry about temporal redundancy.

Different classes of compression techniques are present and broadly they are classified in two types.

A. Lossless Compression

Lossless compression techniques as the name implies, involve no loss of information. If data have been losslessly compressed, the original data can be recovered exactly from the compressed data. Lossless compression is generally used for applications that cannot tolerate any difference between original and reconstructed data. Text compression is an important area for lossless compression. It is very important that the reconstruction is identical to original text, as very small differences can result in statements with very different meaning.

If data of any kind are to be processed or enhanced later to yield more information, it is important that the integrity be preserved. For example, suppose we compressed a radiological image in lossy fashion and the difference between the reconstruction Y and the original X was visually undetectable. If the image was later enhanced, the previously undetectable difference may cause the appearance of artifacts that could seriously mislead the radiologist. Because the price for this kind of mishap may be a human life, it makes sense to be very careful about using a compression scheme that generates a reconstruction that is different from the original.

B. Lossy Compression

Unlike the error free approaches outlined, lossy compression has algorithms based on compromising the accuracy of reconstructed image in exchange for increased compression. If the resulting distortion can be tolerated, the increase in compression can be significant. The term lossy compression and irreversible information loss might sound horrible at first. But the amount of information lost in most lossy compression techniques is nothing compared to the information lost in the original scanning of the image, due to the hardware limitation.

Lossy compression techniques use the trade-off between the accuracy of the decoded image and the compression ratio to achieve greater compression ratios. In most cases the main difference between the structure of a Lossy decoder and a lossless decoder is the presence of a quantiser. As it will be discussed later the function of the quantiser is to reduce the amount of psycho visual redundant data in the image. In most cases the decoded images are not distinguishable by subjective evaluation of the human eye. The following sections describe some issues related to lossy image compression techniques.
Lossy compression thus involves some loss of information, and data that have been compressed using lossy techniques generally cannot be recovered exactly. In return for accepting this distortion in the reconstruction, we can generally obtain much higher compression ratios than is possible with lossless compression.

In many applications, this lack of exact reconstruction is not a problem. As an example, when storing or transmitting speech, the exact value of each sample is not necessary. Depending on the quality required to the reconstructed speech, varying amounts of loss of information amount the value of each sample can be tolerated. If the quality of the reconstructed speech is to be similar to that heard on the telephone, a significant loss of information can be tolerated. However, if the reconstructed speech needs to be of the quality heard on a compact disc, the amount of information loss that can be tolerated is relatively low.

The primary purpose of lossy image compression is:

- Reduce the memory required for their storage
- Reduce the effective data access time when reading the storage devices
- Reduce the bandwidth

C. Lossless vs. Lossy compression

In lossless compression schemes, the reconstructed image, after compression, is numerically identical to the original image. However Lossless compression can only achieve a modest amount of compression. An image reconstructed following Lossy compression contains degradation relative to the original. Often this because the compression scheme completely discards redundant information. However, Lossy schemes are capable of achieving much higher compression. Under normal viewing condition, no visible loss is perceived (visually lossless).

D. **Predictive vs. Transform coding**

In predictive coding, information already sent or available is used to predict future values, and the difference is coded. Since this is done in the image or spatial domain, it is relatively simple to implement and is readily adapted to local image characteristics. Differential Pulse Code Modulation (DPCM) is one particular example spatial domain representation to a different type of representation using some well-known transform and then codes the transformed values (coefficients). This method provides greater data compression compared to predictive methods, although at the expose of greater computation.

Application of data compression is primarily in transmission and storage applications. Image transmission applications are in broadband television, remote sensing via satellite, and military communication via aircraft, radar and sonar, teleconferencing, computer communications, and facsimile transmission. Image storage is required for educational and business documents, medical images that arise in Computer Tomography (CT), Magnetic Resonance Imaging (MRI) and digital radiology, satellite images, weather maps, and so on. An application of data compression is also possible in the development of fast algorithms where the number of operations required to implement an algorithm is reduced by working with the compressed data.

E. Image compression Schemes

Generally all the compression schemes can be classified into lossy and lossless compression. In lossless compression schemes, redundancy in the image is exploited for achieving compression. In the case of lossy compression algorithm a certain amount of information, which is not relevant will be lost deliberately for achieving higher compression.

A few popular lossless compression schemes are described below.

- DPCM-Huffman Rice Algorithm
- Arithmetic Coding
- LZ coding
- Run Length Coding

All the above algorithms achieve compression within the limits allowed by entropy of the source data. If the entropy of the source is high then it may not be possible to get any compression at all. The compression changes from seen to seen. Moreover, they generate variable length code, which is not conveyed for constant rate transmission applications one way or the other a rate control algorithm has to be incorporated to make them suitable for constant rate application. And these algorithms work well if the statistics of the source is well known in advance. Two pass methods, which extract the statistics before actual coding, exist but they don’t suit for real time transmission applications as far as implementation complexity and speeds are concerned. And hence only lossy coding techniques have been evaluated for the compression achievable with a tolerable distortion in the image. Here too a few algorithms have be chosen for evaluation considering the feasibility of implementation of the algorithms.

The most popular Lossy algorithms are -

- DPCM Coding
- DCT Based compression
- Vector Quantization
- Sub-band coding
- Wavelet-Transform
- Fractal Image Compression

IV. JPEG2000 IMPLEMENTATION

The block diagram of the JPEG2000 encoder is illustrated in Fig.1 (a) [9]. The discrete wavelet transform is first applied on the source image data. The transform coefficients are then quantized and entropy coded, before forming the output code-stream (bit-stream). The decoder is the reverse of the encoder (Fig.1 (b). The code stream is first entropy decoded, de-quantized and inverse discrete transformed, thus resulting in the reconstructed image data.

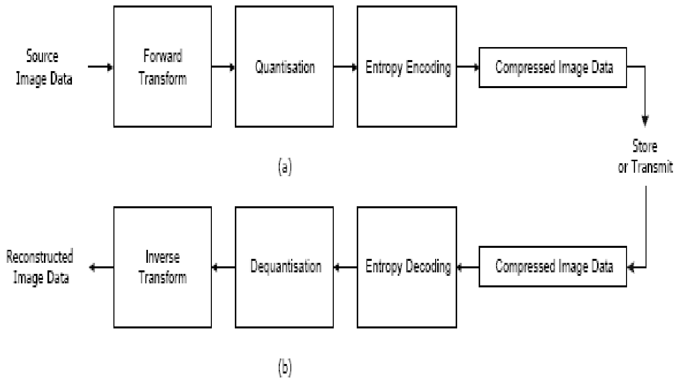

Figure.1: Block diagram of JPEG2000 (a) encoder (b) decoder

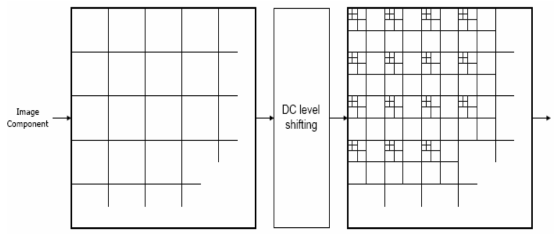

Figure 2: Tiling, DC level shifting and DWT of each image tile component.

Before proceeding with the details of each block of encoder in Fig.1, it should be mentioned that the standard works on image tiles. The term ‘tiling’ refers to the partition of the original (source) image into rectangular non-overlapping blocks (tiles), which are compressed independently, as though they were entirely distinct images (Fig 2). Prior to computation of the forward discrete wavelet transform (DWT) on each image tile, all samples of the image tile component are DC level shifted by subtracting the same quantity (i.e. the component depth). DC level shifting is performed on samples of components that are unsigned only. If colour transformation is used, it is performed prior to computation of the forward component transform. Otherwise it is performed prior to the wavelet transform.

At the decoder side, inverse DC level shifting is performed on reconstructed samples of components that are unsigned only. If used, it is performed after the computation of the inverse component transform. Huffman coding is used for the last part of the encoding process.

Encoding procedure is follows

- The source image is decomposed into components.
- The image and its components are decomposed into rectangular tiles. The tile component is the basic unit of the original or reconstructed image.
- The wavelet transform is applied on each tile. The tile is decomposed in different resolution levels.
- These decomposition levels are made up of sub-bands of coefficients that describe the frequency characteristics of local areas (rather than across the entire tile-component) of the tile component.
- The sub-bands of coefficients are quantized and collected into rectangular arrays of "code-blocks".
- The bit-planes of the coefficients in a "code-block" are entropy coded.

A. Tiling

The term 'tiling' refers to the partition of the original (source) image into rectangular non-overlapping blocks (tiles), which are compressed independently, as though they were entirely distinct images. All operations, including component mixing, wavelet transform, quantization and entropy coding are performed independently on the image tiles. Tiling reduces memory requirements and since they are also reconstructed independently, they can be used for decoding specific parts of the image instead of the whole image. All tiles have exactly the same dimensions, except maybe those at the right and lower boundary of the image. Arbitrary tile sizes are allowed, up to and including the entire image (i.e. the whole image is regarded as one tile). Components with different sub-sampling factors are tiled with respect to a high-resolution grid, which ensures spatial consistency on the resulting tile components.

B. The Wavelet Transform

For an N by N input image, the two-dimensional DWT proceeds as follows:

Step 1: Convolve each row of the image with ho[n] and h_1 [n], discard the odd-numbered columns of the resulting arrays, and concatenate them to form a transformed row.

Step 2: After all rows have-been transformed, convolve each column of the result with ho[n] and h1[n]. Again discard the odd-numbered rows and concatenate the result.

After the above two steps, one stage of the DWT is complete. The transformed image now contains four sub-bands LL, HL, LH, and HH, standing for low-low, high-low, and so on, as Fig.3 (a) shows. As in the one-dimensional transform, the LL sub-band can be further decomposed to yield yet another level of decomposition. This process can be continued until the desired number of decomposition levels is reached or the LL component only has a single element left. A two level decomposition is shown in Fig.3 (b).

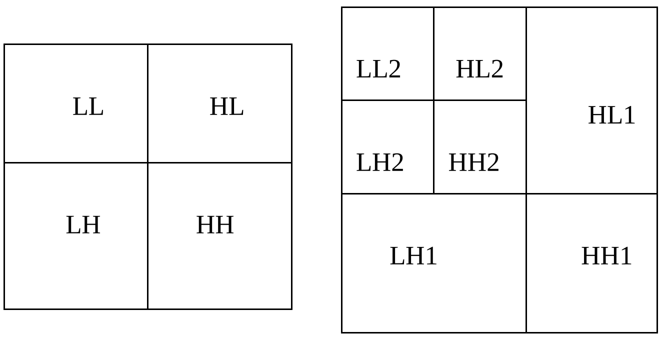

Figure 3: The two-dimensional discrete wavelet transforms:

a) One level transform (b) two-level transform

The inverse transform simply reverses the steps forward transform.

Step 1: For each stage of the transformed image, starting with the last, separate each column into low-pass and high-pass coefficients. Up sample each of the low-pass high pass arrays by inserting a zero after each coefficient.

Step 2: Convolve the low-pass coefficients with ho[n] and high-pass coefficients with h1[n] and add the two resulting arrays.

Step 3: After all columns have been processed, separate each row into low-pass and high pass coefficients and up sample each of the two arrays by inserting a zero after each coefficient.

Step 4: Convolve the low-pass coefficients with ho[n] and high-pass coefficients with h1[n] and add the two resulting arrays.

C. Quantization

After transformation, all coefficients are quantized. Quantization is the process by which the coefficients are reduced in precision. Each of the transform coefficients $a_b(u,v)$ of the sub-band b is quantized to the value $q_b(u,v)$ according to the formula

$$q_b(u,v) = \mathrm{sign}(a_b(u,v))\left\lfloor \frac{|a_b(u,v)|}{\Delta_b} \right\rfloor$$

The quantization step Δ_b is represented relative to the dynamic range R_b of sub-band b, by the exponent R_b and mantissa μ_b as

$$\Delta_b = 2^{R_b - \varepsilon_b}\left(1 + \frac{\mu_b}{2^{11}}\right)$$

The dynamic range R_b depends on the number of bits used to represent the original image tile component and on the choice of the wavelet transform. All quantized transform coefficients are signed values even when the original components are unsigned. These coefficients are expressed in a sign-magnitude representation prior to coding. For reversible compression, the quantization step size is required to be 1

D. Run Length Coding

The idea behind this approach to data compression is; If a data item d occurs n consecutive times in the input stream, replace the n occurrences of a data item are called *run length* of n, and this approach to data compression is called run length encoding.

RLE is a natural candidate for compressing graphical data. A digital image is made up of small dots called pixels. Each pixel can be either one bit indicating a black or a white dot, or several bits, indicating shades of gray. We assume that the pixels are stored in an array called a bitmap in memory, so the bitmap is the input stream for the image. Pixels are normally arranged in the bitmap in scan lines, so the first bitmap pixel is the dot at the top left corner of the image, and the last pixel is the one at the bottom right corner.

Each run of pixels of the same intensity (gray-level) is encoded as a pair (run length, pixel value). The run length usually occupies one byte following for runs up to 255 pixels. The pixel value occupies several bits, depending on the number of gray levels (typically between 4 and 8 bits).

Example: an 8 bit deep gray scale bitmap that starts with 12,12,12,12,12,12,12,12,12,35,76,112,67,87,87,87,5,5,5, 5,5,5,1….. is compressed into 9, 12, 35, 76, 112, 67, 3, 87, 6, 5, 1….

Where underlined numbers indicates the counts. The problem is to distinguish between a byte containing a grayscale value (such as 12) and one containing a count (such as 9). Here are some solutions:

- If the image is limited to just 128 gray scales, we can devote one bit in each byte to indicate whether the byte contains a gray scale value or a count.
- If the number of gray scale is 256, bit can be reduced to 255 with one value reserved as a flag to precede every byte with a count. If the flag is, say, 255, then the sequence above becomes 255,9,12,35,76,112,67,255,3,87,255,6,5,1….

In this type of coding there is always a danger of actually increasing the size of data, instead of decreasing, so usually some other pre-filtering methods are employed to ensure that the data to be compressed contains large sequences of the same values. One such method is the utilization of Colour quantization methods in lossy compression.

V. RESULTS & DISCUSSION

Results for Gray level and Colour images are presented here separately in Table 1 & Table 2. The results are taken by changing the threshold value. Mathematical parameters such as Compression ratio, Compression performance, Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) are determined for each of the tested image.

TABLE 1. RESULTS FOR GRAY-SCALE IMAGES

Images	INPUT PARAMETERS		OUTPUT PARAMETERS		
	Image-Size (bytes)	Threshold	Compression Ratio	MSE	PSNR
lena.bmp	4200	25	1.7486	115.525	27.50
cameraman.bmp	66614	35	3.2981	76.1962	29.311
moon.bmp	65670	50	4.749	116.03	27.484
mri.bmp	17462	50	2.5632	458.34	21.518
cube.tif	16570	30	2.4301	54.171	30.79
chest.tif	1996	30	2.9925	83.977	28.889
jaguar.tif	66086	30	2.765	35.840	32.58
bridge.tif	16720	40	1.7596	203.31	25.04

TABLE 2. RESULTS FOR COLOUR IMAGES

Images	INPUT PARAMETERS				OUTPUT PARAMETERS		
	Image-Size (bytes)	Threshold			Compression Ratio	MSE	PSNR
		R	G	B			
lenac.bmp	786486	50	65	85	4.021	213.431	24.838
flower.tif	198556	50	50	50	3.0581	95.2022	28.3443
girl.tif	196748	50	65	80	3.1627	167.640	25.887
baboon.tif	196662	50	60	70	2.3175	446.438	21.633

TABLE 3. RESULT FOR LENA IMAGE VARYING THE THRESHOLD VALUE

Threshold	**MSE**	**PSNR**	**Compression Ratio**
10	45.2944	31.57	1.355
20	63.39	30.11	1.854
30	86.69	28.75	2.211
40	115.69	27.49	2.517
50	150.16	26.36	2.768
60	197.77	25.16	3.002
70	260.45	23.97	3.235
80	328.48	22.96	3.492
90	402.14	22.08	3.730
100	487.75	20.24	3.980

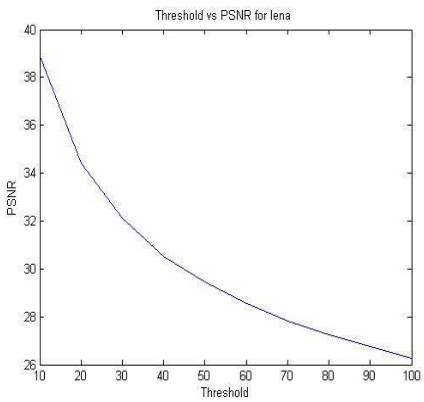

Figure 4: Threshold v/s PSNR for Lena Image

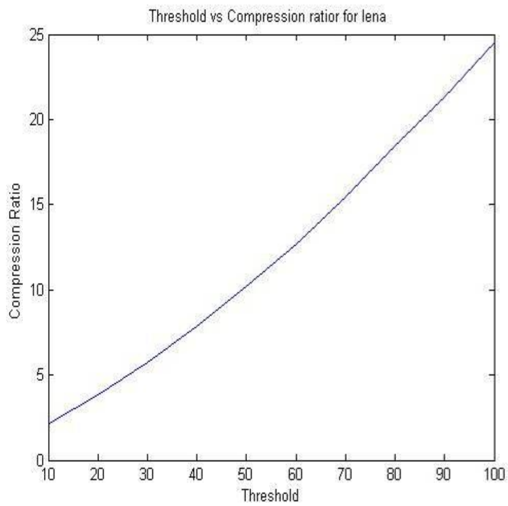

Figure 5: Threshold v/s Compression Ratio for Lena Image

VI. CONCLUSIONS

The algorithm for JPEG2000 used is fast and computationally simple. The algorithm achieves variable compression ratio for different threshold values. It has achieved good quality of reconstructed image. It outperforms the DCT based JPEG image compression algorithm.

On the whole, the implementation is tested for several images, and the output was verified to reconstruct successfully in each case. Depending on the complexity of the image and threshold used for compression, different sets of compression ratios are achieved. The simulation is done using MATLAB.

The performance measures like Mean Square Error (MSE) and Peak Signal Noise Ratio (PSNR) are calculated. The graphs of Threshold v/s Compression Ratio and Threshold v/s PSNR are plotted. From the graphs, it can be concluded that as threshold value increases compression ratio increases and PSNR decreases. To achieve higher compression ratio we have to give higher threshold values.

For simplicity of the algorithm, the Daubechies Wavelets for Forward transformation and Run Length Coding for entropy coding are used. To achieve more efficiency we can use:

- Other Wavelets like Haar, Antomini 9/7, Villa 10/18 and other wavelets can be used instead of Daubechies Wavelets and performances can be compared.
- Arithmetic Coding can be used instead of Run Length Coding but it is computationally complex

VII. REFERENCES

[1] R C Gonzalez and R.E. Woods, “Digital Image Processing”, 2nd ed,.Pearson education, 2002.

[2] Ze-Nian Li and Mark S. Drew, “Fundamentals of Multimedia”, Pearson Education, Inc 2004.

[3] Anil K Jain, “Fundamental of Image Processing”. Prentice-Hall of India 5 edition 1995.

[4] Ahmed N, Natarajan t and Rao K R, Discrete Cosine Transforms, IEEE trans, vol c-23, pp. 90-93, 1974.

[5] Karen L. Gray “The JPEG2000 Standard” IEEE transactions on image processing, vol. i no *2*. April 1992.

[6] Marc Antonini, Michel Barlaud, Pierre Mathieu, and Ingrid Daubechies, Touradj Ebrahimi3, “Image Coding Using Wavelet Transform”, IEEE Transactions on Consumer Electronics, Vol. 46, No. 4, pp. 1103-1127, November 2000.

[7] A. S. Lewis and G. Knowles “Image Compression Using the 2-D Wavelet Transform” IEEE transactions on image processing, vol. i no. 2. April 1992.

[8] Charilaos Christopoulos, Athanassios Skodras and Touradj Ebrahimi,“The jpeg2000 Still Image Coding System: An Overview” IEEE Transactions on Consumer Electronics, Vol. 46, No. 4, pp. 1103-1127, November 2000

[9] http://www.mathworks.com

[10] http://www.jpeg.org

[11] http://www.ieeexplore.com