DESIGN AND ANALYSIS OF IMAGE COMPRESSION USING DCT AND DWT

An interim minor project report submitted in
partial fulfillment of the requirements for the Degree of

Bachelor of Technology
in
Electronics and Communication Engineering
by

Ashwin Bansal (201A002)

Shivam Srivastava (201A016)

Under the Guidance of

Dr. Deepak Sharma

Department of Electronics & Communication Engineering

JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY,

A.B.ROAD, RAGHOGARH, GUNA-473 226, M.P. (INDIA)

December, 2022

Department of Electronics and Communication Engineering
Jaypee University of Engineering and Technology
A.B. Road Raghoghar, Guna-473226 (M.P.) INDIA

CERTIFICATE

This is to certify that the thesis work entitled “Design and implementation of image compression using DWT and DCT” submitted by Mr. Ashwin Bansal and Mr. Shivam Srivastava (Er. No. 201A002 and 201A016) in partial fulfillment for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering from “Jaypee University of Engineering and Technology, Guna” has carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree.

(Head of Department)
Prof. Narendra Singh
Date:___/___/_____

(Thesis Supervisor)
Dr. Deepak Sharma
Date:___/___/_____

ABSTRACT

The Joint Photographic Experts Group's image compression algorithm has been shown to be a very efficient and powerful method of compressing images. However, there are substantive information about which color space should be utilized when implementing the JPEG algorithm. Currently, the JPEG algorithm is set up for use with any three component color space. The objective of this research was to determine whether or not the color space selected will significantly improve image compression capabilities. The RGB, XYZ, YIQ, CIELAB, CIELUV, and CIELAB LCH color spaces were examined and compared. Both numerical measures and psychophysical techniques were used to assess the results. The final results indicate that the device space, RGB, is the worst color space to compress images. In comparison, the nonlinear transforms of the device space, CIELAB and CIELUV, are the best color spaces to compress images. The XYZ, YIQ, and CIELAB LCH color spaces resulted in intermediate levels of compression.

Table of Contents

LIST OF IMAGES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

Image compression is one of the most important and successful applications of the wavelet transform. Mature wavelet based image coders like the JPEG2000 standard are available, gaining popularity, and easily outperform traditional coders based on the discrete cosine transform (DCT) like JPEG. Unlike in DCT based image compression, however, the performance of a wavelet based image coder depends to a large degree on the choice of the wavelet. This problem is usually handled by using standard wavelets that are not specially adapted to a given image, but that are known to perform well on photographic images.

However, many common classes of images do not have the same statistical properties as photographic images, such as fingerprints, medical images, scanned documents and satellite images. The standard wavelets used in image coders often do not match such images resulting in decreased compression or image quality. Moreover non-photographic images are often stored in large databases of similar images, making it worthwhile to find a specially adapted wavelet for them.

Memory and bandwidth are the prime constraints in image storage and transmission applications. One of the major challenges in enabling mobile multimedia data services will be the need to process and wirelessly transmit a very large volume of data. While significant improvements in achievable bandwidth are expected with future wireless access technologies, improvements in battery technology will lag the rapidly growing energy requirements of future wireless data services.

One approach to mitigate this problem is to reduce the volume of multimedia data transmitted over the wireless channel via data compression techniques. This has motivated active research on multimedia data compression techniques such as JPEG, JPEG 2000 and MPEG. These approaches concentrate on achieving higher compression ratio without sacrificing the quality of the image. However these efforts ignore the energy consumption during compression and RF transmission. Since images will constitute a large part of future wireless data, the thesis aim on developing energy efficient and adaptive image compression and communication techniques. Based on wavelet image compression, energy efficient multi-wavelet image transform is a technique developed to eliminate computation of certain high-pass coefficients of an image.

JPEG2000 standard, defines a new image-coding scheme using state-of-art compression techniques based on wavelet technology. Its architecture is useful for many diverse applications, including internet distribution, security systems, digital photography and medical imaging. Set Partitioning in Hierarchal trees (SPIHT) is an example of progressive image compression algorithm is an extension of Shapiro's embedded zero tree wavelet (EZW) method. It has high image quality. Images from different categories tend to show different spatial domain characteristics. Example, many medical images like CT scan or ultrasound contain significant low intensity (block) regions along with image boundaries.

Generic coding schemes compressing image as a whole can produce a high compression ratio but with considerable loss of overall quality. In the ROI coding, images are segmented into ROI, which is considered important, and background, which is less important. By allowing ROI to be coded with high fidelity than the background, a high compression ratio with good quality in ROI can be achieved. The greatest benefit of ROI coding is its capability of delivering high reconstruction quality over certain spatial regions at high compression ratios. It has been observed that when we compress a variety of images of different types using a fixed wavelet filters, PSNR, SSIM and correlation vary widely from image to image. These variations can be attributed to the nature and inherent characteristics of image.

CHAPTER 2. LITERATURE SURVEY

2.1 Recent technologies in wavelet transform

The past decade has witnessed the development of wavelet analysis, a new tool that emerged from mathematics and was quickly adopted by diverse fields of science and engineering. Wavelet analysis has begun to play a serious role in a broad range of applications, including signal processing, data and image compression, solution of partial differential equations, modelling multiscale phenomena, and statistics. There seem to be no limit to the subjects where it may have utility. Our aim is to explore some additional topics that extend the basic ideas of wavelet analysis**.**

There are different wavelet transforms used in as recent technologies, some of them are :-

2.1.1 Wavelet Packet Transform

A wavelet packet transform is a simple generalization of a wavelet transform. The Discrete Fourier Transform (DFT) may be thought of in general terms as a matrix multiplication. DWT's are particularly effective in analysing waveforms which have spikes or pulses buried in noise. The noise may be more effectively removed than with FT filtering and the shape of the pulses preserved.

2.2.2 Haar Wavelet Transform

The Haar wavelet packet transform is usually referred to as the Walsh transform. It is easier to describe in the form of matrix multiplication

$$(x_k)=a_0\begin{pmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{pmatrix}+a_1\begin{pmatrix}1\\1\\1\\1\\-1\\-1\\-1\\-1\end{pmatrix}+a_2\begin{pmatrix}1\\1\\-1\\-1\\0\\0\\0\\0\end{pmatrix}+a_3\begin{pmatrix}0\\0\\0\\0\\1\\1\\-1\\-1\end{pmatrix}+a_4\begin{pmatrix}1\\-1\\0\\0\\0\\0\\0\\0\end{pmatrix}+a_5\begin{pmatrix}0\\0\\1\\-1\\0\\0\\0\\0\end{pmatrix}+a_6\begin{pmatrix}0\\0\\0\\0\\1\\-1\\0\\0\end{pmatrix}+a_7\begin{pmatrix}0\\0\\0\\0\\0\\0\\1\\-1\end{pmatrix}$$

2.2 Recent researches

There have been few research work done on image compression and wavelet transforms follows

2.2.1 Impact of DWT and median filtering on removal of salt and pepper noise in digital images .

Removes most of the noisy part from the image and maintains the visual quality. The level of wavelet decomposition is restricted to three. The renowned indexes Peak Signal to Noise Ratio

(PSNR) and Root Mean Square Error (RMSE) demonstrate marked improvement of image denoising over Gaussian method.

2.2.2 Survey on medical ımage de noising using various filters and wavelet transform.

It explained that There is a problem of high level components of noises in the Medical Images. The different medical images are Magnetic Resonance Imaging. (MRI), X-ray, Computed Tomography and Ultrasound. There are many noise reduction techniques that have been developed for removing noise. The idea behind these techniques is to get better results in terms of quality and in removal of different noises. This paper gives the review of various de noising techniques.

2.2.3 Wavelet based technıque for removal of multıple noıses sımultaneously

It presented that Denoising is important pre processing tasks for various image processing. Image noise is the random variation of brightness and color information in images produced by the scanner and digital camera. This paper presents a novel approach for simultaneously removing the speckle and Salt-pepper noise from a single image by using the median filter. This paper proposes an adaptive, data-driven threshold for image denoising via wavelet soft thresholding

2.3 Recent technologies in compression technique algorithm

Two essential and basics parts are reducing redundancy and irrelevancy. Reducing Redundancy focuses to reproduce exactly from the image. Parts of the image are omitted unnoticed by the receiver from naked eye namely Human Visual System in irrelevancy reduction. There are several image compression algorithms some of them are lossy and some are lossless such as fractal image compression, transform-based image compression (DCT, DWT)), image compression using wavelet coding, ESPIC EBCOT, embedded zero tree wavelet algorithm, SPIHT algorithm, Set Partitioned Embedded Block coding and Wavelet difference reduction algorithm. Recent compression methodologies successfully achieve high compression rate and maximum quality of perception relative to previous one

2.4 Compression technique

There are two types of compression techniques-

2.4.1 Lossless compression

Lossless compression is a class of data compression that is used for reconstructing the original data from the compressed data. Lossless compression programs process in sequence. In this sequence, the first step is to produce a statistical model for input data and the second step is used to map the input data into bit sequences to create the shorter output. There are many encoding techniques that are used to generate bit sequence like arithmetic coding and Huffman coding. Both of these two techniques arithmetic coding technique of coding is better than Huffman coding because arithmetic coding achieve better compression rates for models that is best for statistical model. There are various application where lossless Compression is used such as used in ZIP, PNG and GNU file format

2.4.2 Lossy Compression

Lossy compression is a class of data compression that is used for reconstructing the approximation of the original data to improve the compression rates. There are various application where Lossy compression is used such as TIFF and MNG. Lossy compression are mostly used for compressing digital media such as audio, video and images etc,. Lossy compression is better than lossless compression because lossy compression technique produce a smaller compressed file than lossless compression

2.5 Comparison between DCT and DWT compression technique

2.5.1 DCT (Discrete Cosine Transform)

DCT is a frequency domain image transform method that is used to reduce the storage space where we want to store the image. In DCT, the whole image is divided into n*n blocks then DCT is applied on these blocks. The changes across the width and height of the blocks are expressed as high order terms and the average value in a block is expressed as low order terms. The IDCT (Inverse discrete cosine transform) can be used to recreate the image from compressed representation. DCT is a lossy compression algorithm. The DCT lossy image compression technique gives the best result for the lossy image compression. The value of the DCT Lossy image compression PSNR value is good in high compression ratio. In the lossy compression technique the quality of the image is low. The DCT lossless image compression technique gives the average result for the lossless image compression. The value of the in the lossless compression technique DCT did not give the best result for the image compression. The PSNR value of lossless compression is good.

2.5.2 DWT (Discrete Wavelet Transform)

DWT also a frequency domain image transform method that is used to split the information of any digital media into approximation sub signal (used to show the pixel value) and detailed sub signal (used to show the vertical, horizontal and diagonal details). The main objective of DWT technique is to hide data in the form of coefficients. Dwt is used to separate the image into a pixel. DWT is used in signal and image processing especially for lossless image compression. .DWT transforms a discrete signal .L represent the low-pass filtered signal L(low frequency)allows the perfect reconstruction of original Image. H represents the high-pass filtered signal. The DWT represents the two images representing the technique to transform the DWT process. Then the DWT image will move on to the quantization process. That the process is doing again and again to get the best result. Thus the output of the DWT image compression is good. The PSNR value is also good in compression ratio.

The quality of the DWT image is also good. Now a day's DWT image compression technique is used to get the best output, and also to get the quality of the image

DWT lossy image compression technique did not give the best result because of lossy image compression. The value of the DWT image compression PSNR value is low in high compression ratio. In the Lossy compression ratio was good but average quality of the image. DWT image compression is the technique mostly used in the lossless image compression. In this technique lossless gives the best compression result. The PSNR value of Lossless Image is good quality. The lossless image compression ratio was good, and also the quality of the lossless image compression also good

Table 1: Parameter comparison based on different aspect of image

PARAMETERS	**DCT**	**DWT**
Compression Ratio	*10:1*	*16:1*
Percentage of Compression	*95%*	*90%*
PSNR	*47.1225*	*41.1225*
MSE	*0.0020*	*0.9995*

Authors Year	Techniques Proposed	Problem Addressed	Our Review
Jasmeet Kaur et. al. [2012]	DCT-DWT based video compression	To achieve the higher compression ratio on the basis of video frame and scene evaluation	The compression ratio is higher but it takes the higher execution time
Eman A. AL, hillo et. al.[2014]	Fractal compression technique for examined the size of colour image	Without any distortion in original image, image is transferred into YIQ color space and chromatic components to increase compression ratio	Colour space components increase the compression ratio , but produced the data loss content
Anil Kumar et. al.[2011]	Comparative analysis between DCT and DWT based on MSE	Finding which one is better DCT or DWT	DCT and DWT has been proved better than other surveyed

Table 2: Peoples and their works in image compression

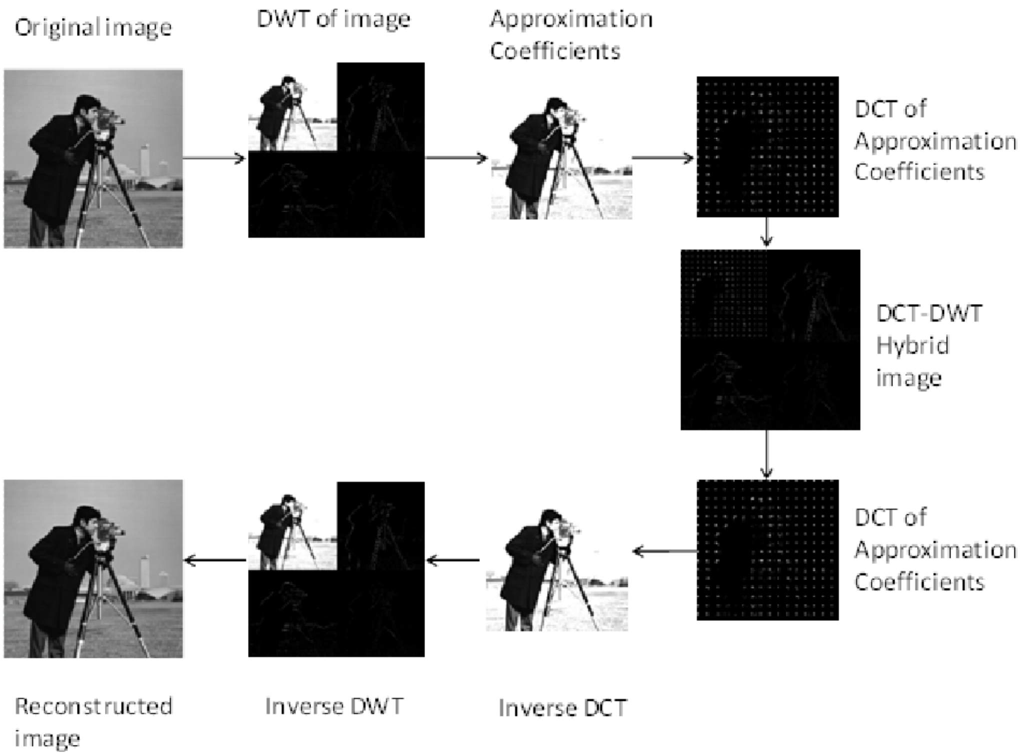

Original image
DWT of image
Approximation Coefficients
DCT of Approximation Coefficients
DCT-DWT Hybrid image
DCT of Approximation Coefficients
Reconstructed image
Inverse DWT
Inverse DCT

Fig 2.1: DCT – DWT Hybrid image compression technique block diagram

2.6 Performance parameters

Like any other system, metrics of performance of a data compression algorithm are important criteria for selection of the algorithm. The performance measures of data compression algorithms can be looked at from different perspectives depending on the application requirements: amount of compression achieved, objective and subjective quality of the reconstructed data, relative complexity of the algorithm, speed of execution, etc. We explain some of them below:-

2.6.1 Compression Ratio and Bits per Sample

The most popular metric of performance measure of a data compression algorithm is the compression ratio. It is defined as the ratio of the number of bits to represent the original data to the number of bits to represent the compressed data. Consider an image of size 256 x 256 requiring 65536 bytes of storage if each pixel is represented by a single byte. If the compressed version of the image can be stored in 4096 bytes, the compression ratio achieved by the compression algorithm will be 16:l. A variation of the *compression ratio* is *bits per sample.* This metric indicates the average number of bits to represent a single sample of the data (e.g., *bits per pixel* for image coding). If 65536 pixels of an image are compressed to 4096 bytes, we can say that the compression algorithm achieved 0.5 bits per pixel on the average. Hence the *bits per sample* can be measured by the ratio of the *number* of *bits* of a single uncompressed sample to the *compression ratio.*

It should be remembered that the achievable compression ratio using a lossless compression scheme is totally input data dependent. If the same algorithm is applied in a number of distinct data files, the algorithm will yield a different compression ratio in different files. The maximum compression ratio and hence the bits per sample that can be achieved losslessly is restricted by the entropy of the data file according to the *noiseless source coding theorem* by Shannon. Sources with less redundancy have more entropy and hence are more difficult to achieve compression. For example, it is very difficult to achieve any compression in a file consisting of mainly random data

$$Compression\ ratio = \frac{Size\ of\ the\ compressed\ image}{Size\ of\ the\ original\ image}$$

2.6.2 Compression Factor(CF)

It is the inverse of the compression ratio, which is the ratio between the size of the original image and the size of the compressed image.

$$Compression\ factor = \frac{Size\ of\ the\ original\ image}{Size\ of\ the\ compressed\ image}$$

2.6.3 Mean Squared Error

Compression of an image using a lossy image compression algorithm result in a decompressed image with loss of some information of the pixel intensity values. Those pixels are significantly different than the original image pixels. The error metrics that can be used to compare the original image and the decompressed image is MSE. It calculates the cumulative squared error between each corresponding pixel in the original and the decompressed image. The MSE is defined as follows:

$$MSE = \frac{1}{M \times N} \sum_{y=1}^{M} \sum_{x=1}^{N} [I(x, y) - I'(x, y)]^2$$

Where,

$I(x, y)$ - The intensity value in the original image

$I'(x, y)$ - The corresponding intensity value in the decompressed image

M, N - The dimensions of the images

2.6.4 Peak Signal to Noise Ratio(PSNR)

In image compression, the value of PSNR measures the ratio between the maximum possible intensity value (power) of an image (signal) and the power of distorting noise, which affects the quality of its representation as a logarithmic decibel scale. A higher PSNR value provides higher image quality and the other end of the scale, a smaller PSNR value implies high numerical differences between both the original image and the decompressed image. The PSNR is defined as follows:-

$$PSNR(dB) = 20 \times \log_{10}\left(\frac{I_{Max}}{\sqrt{MSE}}\right)$$

Where,

I_{Max} - The maximum intensity value of the image

2.6.5 Quality Metrics

This metric is not relevant for lossless compression algorithms. The quality or fidelity metric is particularly important for lossy compression algorithms for video, image, voice, etc., because the reconstructed data differ from the original ones and the human perceptual system is the ultimate judge of the reconstructed quality. For example, if there is no perceivable difference between the reconstructed data and the original ones, the compression algorithm can be claimed to achieve very high quality or high fidelity. The difference of the reconstructed data from the original ones is called the *distortion.* One expects to have higher quality of the reconstructed data, if the distortion is lower. Quality measures could be very subjective based on human perception or can be objectively defined using mathematical or statistical evaluation

CHAPTER 3. DESIGN & IMPLEMENTATION

In this chapter details of the Baseline JPEG algorithm is discussed. Then the specifics of each of the color spaces used in this research are reviewed. Finally, a computational example is presented. This example follows a sample 8x8 block of pixels from original RGB form, through the color space transformations, through the compression steps and then back out through the decompression steps. This further illustrates the discussions of the JPEG algorithm and the color space transformations. Moreover, the example provides a useful reference for the Results and Discussion section.

3.1 . Baseline JPEG

One of the major objectives of the JPEG committee was to establish a basic compression technique for use throughout industry. A method that would be compatible with all of the various types of hardware and software that would be performing image compression. To accomplish this goal the committee developed the Baseline JPEG algorithm. Additional changes could then be made to the Baseline algorithm according tb individual users preferences, but only the Baseline algorithm would be universally implemented and utilized. This algorithm was designed to accept 8 bits per pixel images. It should be noted that in compression literature, the term pixel is used loosely to mean either a single monochrome pixel or a single channel of a color pixel. The same terminology is used in this thesis and the reader should be aware that, unless otherwise specified, the word pixel will be synonymous with a single pixel channel or an individual red, green or blue pixel component.

The Baseline JPEG algorithm is composed of three major compression steps and three major decompression steps. This is shown in flowchart form in Fig. 3.1 on the following page. The first step in the process is to break the original image down into eight by eight pixel blocks. These blocks are then sent, one at a time, into the JPEG algorithm. On this chart the JPEG algorithm is represented by the large box in the figure. The steps in the compression algorithm are a transformation from a spatial representation to a frequency representation, systematic quantization and statistical encoding. These three steps would be repeated until the entire image was in a compressed form. At this point, the image can be stored or transmitted as needed. The steps in the decompression algorithm are decoding the bit stream, dequantization, and transforming from a frequency representation back to a spatial representation. Once again the basic computational unit is an eight by eight pixel block and when the last three steps have been repeated for all of the data an image will be reconstructed. The necessary color space transformations were performed before and after the JPEG algorithm. The CIELAB, CIELAB LCh, CIELUV, RGB, xyz, and YIQ spaces were selected as the color spaces to be examined during the course of this research. The following sections provide additional details about each of these steps.

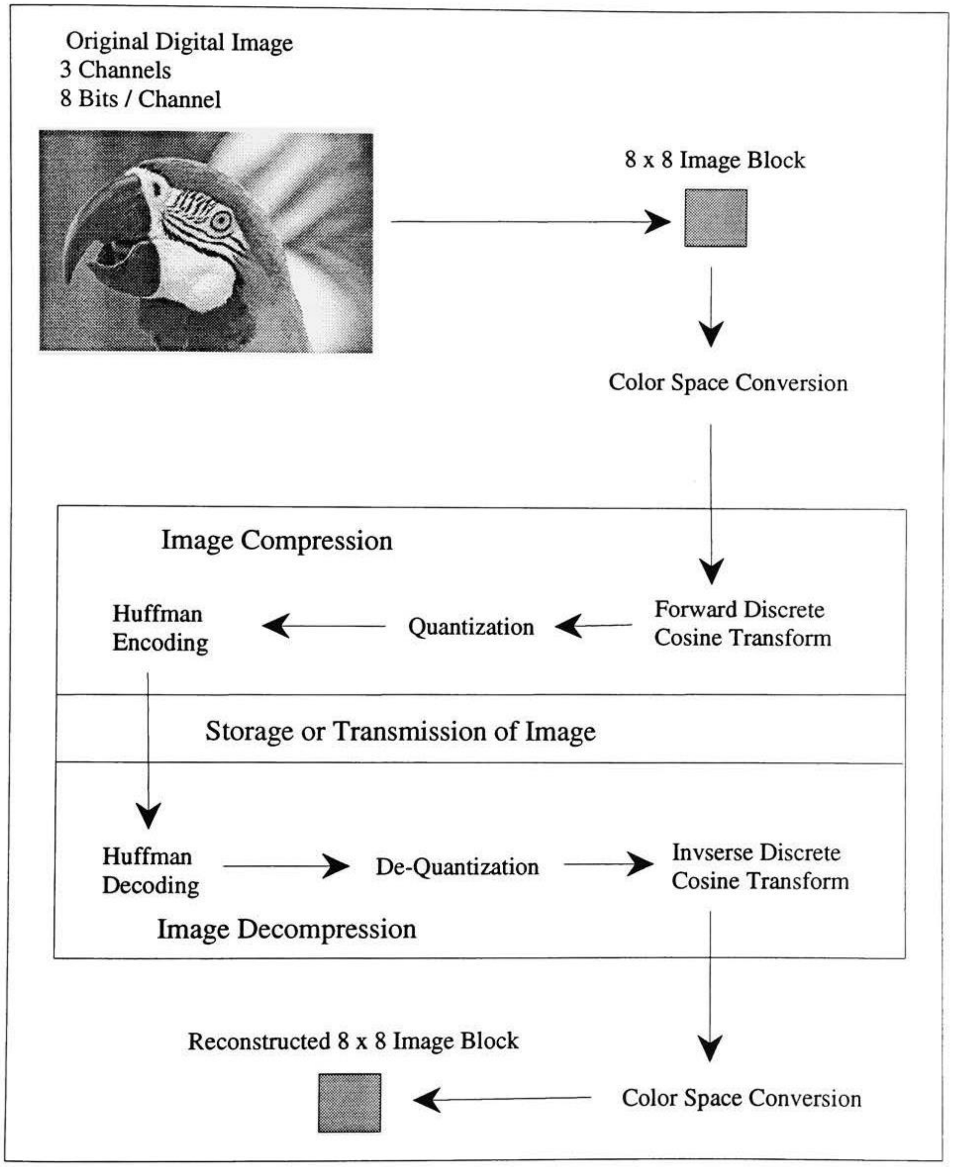


Fig 3.1 Flowchart for baseline JPEG Algorithm

3.1.1 The FDCT and IDCT

The mathematical transformation used in the JPEG algorithm for changing from a spatial representation of the pixels to a frequency representation is the Discrete Cosine Transforrn or DCT. The forward discrete cosine transform or FDCT can be thought of as a harmonic analyzer (Wallace 1992). The input to the FDCT is the 8x8 pixel block with a constant, 2(n-1) - 1, subtracted from all the values. The value n is the number of bits of per pixel and, in this case, n was equal to 8 and 128 was subtracted from all the pixels. This bipolar spatial data is then decomposed into the corresponding DCT basis functions. This process is similar to the Fourier Transform, but the DCT uses all real numbers and does not assume periodicity (Ahmed 1973). The u and v in Eqn. 3.1 are the indices for the frequency space coefficients. The x and y, on the other hand, are used to index through the original 64 pixels in their spatial representation. The two cosine terms are responsible for generating the two-dimensional sinusoidal grating patterns that are the basis functions, The actual DCT basis functions are shown in Fig. 3.2.

The equation for the FDCT is as follows:

$$F(u,v) = \frac{1}{4}C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cdot\cos\frac{(2x+1)u\pi}{16}\cdot\cos\frac{(2y+1)v\pi}{16}\right]$$

where:

$$C(u), C(v) = \frac{1}{\sqrt{2}} \text{ for } u, v = 0,$$
$$C(u), C(v) = 1 \text{ otherwise.}$$

The vertical frequency of the patterns increases from left to right and the horizontal frequency increases from top to bottom. The upper left corner is a special case because it is the lowest frequency DCT basis function and it has, effectively, no frequency. This coefficient has been given the name the DC term and the other 63 terms are commonly referred to as the AC coefficients. This figure also provides a qualitative idea of what actually happens when an eight by eight pixel block is sent through the DCT. Basically, the equation will determine which of the 64 eight by eight basis functions are needed and in what magnitude to generate the original eight by eight pixel block. The inverse discrete cosine transform, or IDCT, is just the opposite of the FDCT and it transforms an eight by eight block of DCT coefficients to a reconstructed eight by eight block of pixels. In this case, frequency information is being transformed back into a spatial representation. This process can be thought of as a harmonic synthesizer (Wallace 1992).

3.1.2. Quantization and Dequantization of the DCT Coefficients

The next step in the flow chart is the quantization of the FDCT coefficients. This step compresses the image by selectively discarding information. The JPEG scheme for quantizing the DCT coefficients attempts to model the human visual system. Specifically, the DCT terms are quantized according to their visual significance. Early on the JPEG committee recognized that color information could be quantized differently than monochrome information. The largest difference was that chrominance or color information could be quantized more coarsely than the luminance or tone information (Hunt 1988). This scheme was based on the physiological properties of the visual system. To illustrate, the human luminance and chrominance.

The equation form of the IDCT is written:

$$f(x,y) = \frac{1}{4}\left[\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)F(u,v)\cdot\cos\frac{(2x+1)u\pi}{16}\cdot\cos\frac{(2y+1)v\pi}{16}\right]$$

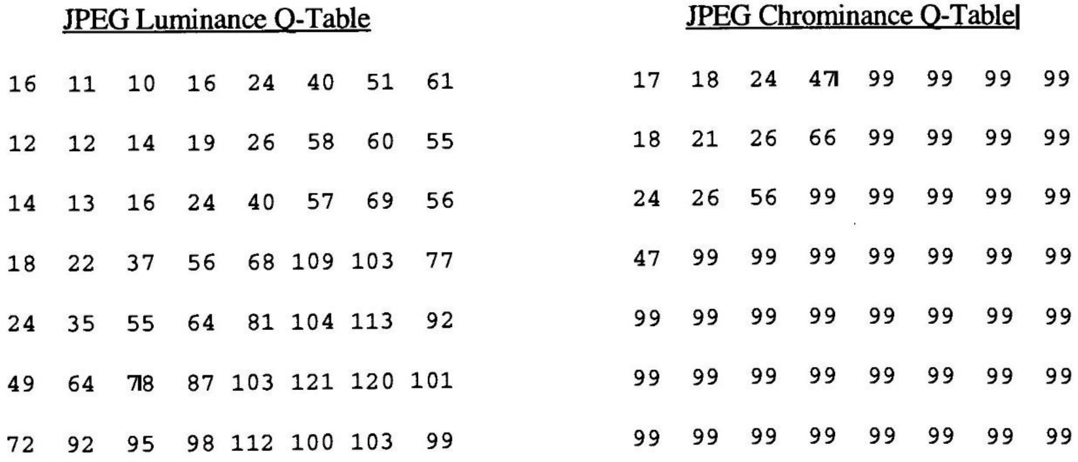
where:

$$C(u),C(v) = \frac{1}{\sqrt{2}} \text{ for } u,v = 0,$$
$$C(u),C(v) = 1 \text{ otherwise.}$$

contrast sensitivity functions are shown in Fig. 3.3. This figure shows that the luminance contrast sensitivity function is roughly bell shaped and peaks around 7 cycles per degree. The chrominance contrast sensitivity function, in comparison, is plateau shaped and peaks at a lower frequency level. The maximum sensitivity is also higher for the luminance information than for the chrominance frequencies. This demonstrates that the human visual system is more sensitive to luminance frequency changes than chrominance frequency changes (Schreiber 1991).

The JPEG quantization scheme attempts to model this relationship, as well as some of the details. The JPEG committee has not provided any quantization tables, or Q-Tables, for use as standards. For use with the algorithm and this research utilized these two tables. One table is for quantizing luminance information and the other is for chrominance quantization. It has been commented that there are some irregularities in these two tables (Klein 1992). Nevertheless, the basic concepts of quantizing chrominance information more coarsely than luminance information and of varying the rate of quantization based on the frequency of the basic functions are valid and have been supported by experimental results (Lohscheller 1984).

Another way of illustrating the quantization scheme proposed by the JPEG committee is shown in Fig. 3.5. In this figure, the 64 DCT basis functions from Fig. 3.2 are shown with varying sizes. The size of the basis functions is indicative of how that particular DCT basis function will be quantized using the luminance Q-Table shown in Fig. 3.4.



JPEG Luminance Q-Table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	718	87	103	121	120	101
72	92	95	98	112	100	103	99

JPEG Chrominance Q-Table

17	18	24	471	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Fig 3.2 Example of Luminance and Chrominance Q -Tables

The larger the basis function, the more finely the signal will be quantized and the less information will be discarded. In comparison, the smaller the basis function the more coarsely it will be quantized and the more information will be lost. The amount of quantization for each block has been scaled relative to each blocks relative size . Notice how much larger the basis functions in the upper left portion of the figure are compared to the other basis functions. The actual equation for quantization is to divide each FDCT coefficient by the corresponding.

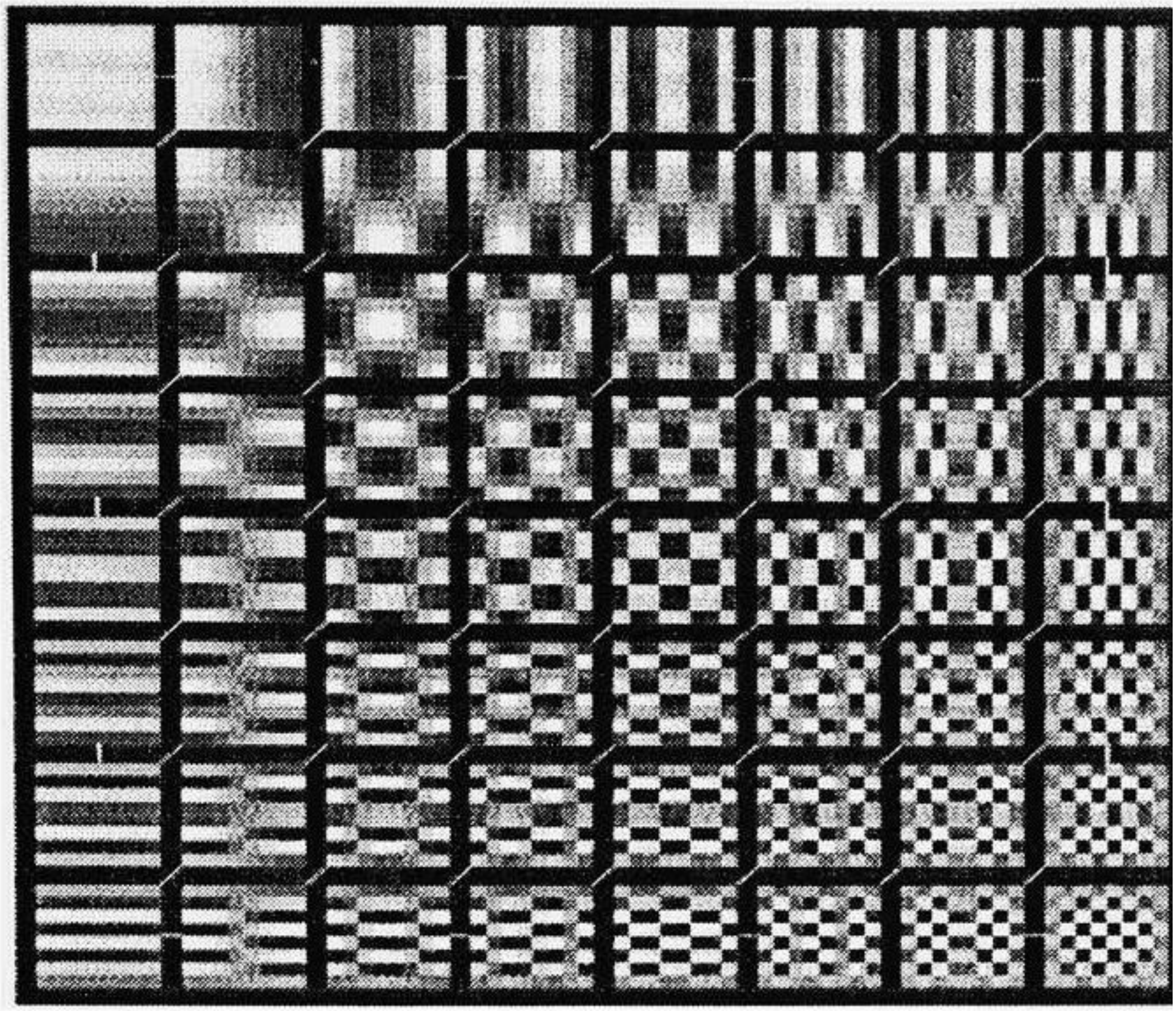


Fig 3.3 Human Luminance and Chrominance CSF's

The larger the basis function, the more finely the signal will be quantized and the less information will be discarded. In comparison, the smaller the basis function the more coarsely it will be quantized and the more information will be lost. The amount of quantization for each block has been scaled relative to each blocks relative size. Notice how much larger the basis functions in the upper left portion of the figure are compared to the other basis functions. The actual equation for quantization is to divide each FDCT coefficient by the corresponding element in the Quantization Table. This number is then rounded to the nearest integer and the entire operation can be expressed as follows:

$$F_Q(u,v) = \text{Integer Round}\left(\frac{F(u,v)}{Q(u,v)}\right)$$

where u and v are indices used to locate the specific elements in the 8x8 FDCT block and the Q-Table, F(u,v) are the FDCT coefficients and are the Q-Table values. The dequantization is the inverse operation of the quantization process. This step uses the same two quantization tables as are used for quantizing the FDCT coefficients. In this case the Q-Table values are multiplied by the DCT coefficients. This can be written in the following form:

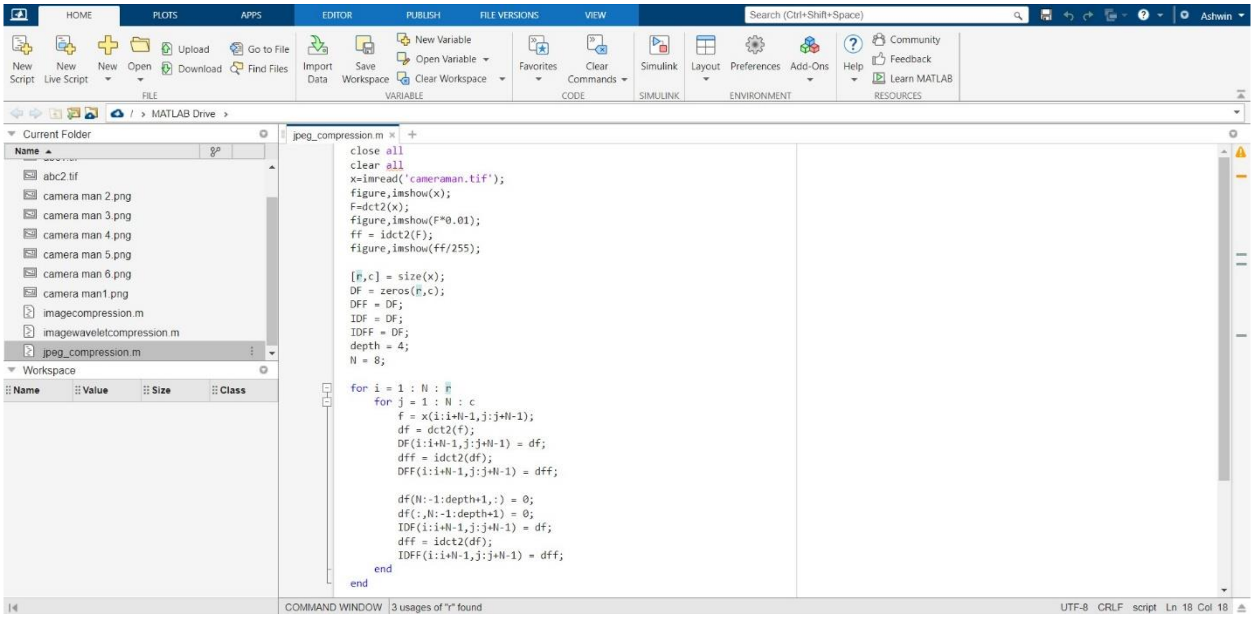
$$F_{QI}(u,v) = F_Q(u,v) \cdot Q(u,v)$$

The Q-Tables are a vital part of the JPEG algorithm, not only because of the amount of compression achieved, but because they provide a mechanism for determining how much an image will be compressed. The level of compression can be determined by scaling all of the elements in the Q-Tables by some constant factor (Wallace 1990). For instance, to quantize the signal more coarsely, all of the Q-Table values should be larger than their default values. On the other hand, to quantize the signal more finely, all of the Q-Table values should be smaller than their default values. Multiplying all of the elements by a constant greater than one will increase all of the Q-Table values and will, consequently, discard more information. Multiplying all of the elements by a constant less than one will decrease all of the Q-Table values and will discard less information.

However, if the DCT coefficients are not quantized at all, assuming no round-off error, there will be no information lost. This type of compression is referred to as lossless compression. Although lossless compression is completely error free, it is not used extensively because the resulting compression of the image is minimal. Therefore, most image compression discards information in what is often referred to as lossy compression. The JPEG Baseline algorithm, with its quantization step, is a lossy algorithm. Nevertheless, it is possible to perform lossy compression in which there is no perceptible degradation in the image. This is known as visually lossless compression and results in a compression in which a maximum data compression occurs with no distortion in the image quality (Gentile, Allebach, and Walowit 1990). The first experiment performed for this research deals with determining levels for visually lossless JPEG compression.

3.1.3 Run Length Coding

The neighboring pixels in a typical image are highly correlated to each other. Often it is observed that the consecutive pixels in a smooth region of an image are identical or the variation among the neighboring pixels is very small. Appearance of runs of identical values is particularly true for binary images where usually the image consists of runs of 0's or 1's. Even if the consecutive pixels in grayscale or color images are not exactly identical but slowly varying, it can often be preprocessed and the consecutive processed pixel values become identical. If there is a long run of identical pixels, it is more economical to transmit the length of the run associated with the particular pixel value instead of encoding individual pixel values. Run-length coding is a simple approach to source coding when there exists a long run of the same data, in a consecutive manner, in a data set. As an example, the data d = 5 5 5 5 5 5 5 19 19 19 19 19 19 19 19 19 19 19 19 0 0 0 0 0 0 0 0 8 23 23 23 23 23 23 contains long runs of 5's, 19's, 0's, 23's, etc. Rather than coding each sample in the run individually, the data can be represented compactly by simply indicating the value of the sample and the length of its run when it appears. In this manner, the data d can be run-length encoded as (5 7) (19 12) (0 8) (8 1) (23 6). For ease of understanding, we have shown a pair in each parentheses. Here the first value represents the pixel, while the second indicates the length of its run. In some cases, the appearance of runs of symbols may not be very apparent. But the data can possibly be preprocessed in order to aid run-length coding. Consider the data d = 26 29 32 35 38 41 44 50 56 62 68 78 88 98 108 118 116 114 112 110 108 106 104 102 100 98 96. We can simply preprocess this data, by taking the sample difference e(I) = d(I) - d(I - l), to produce the processed data t = 26 3 3 3 3 3 3 6 6 6 6 10 10 10 10 10 -2 -2 -2 -2 -2 -2 -2 -2 -2 -2 -2. This preprocessed data can now be easily run-length encoded as (26 1) (3 6) (6 4) (10 5) (-2 11). A variation of this technique is applied in the baseline JPEG standard for still-picture compression. The same technique can be applied to numeric databases as well. On the other hand, binary (black-and-white) images, such as facsimile, usually consist of runs of 0's or 1's. As an example, if a segment of a binary image is represented as



```
close all
clear all
x=imread('cameraman.tif');
figure,imshow(x);
F=dct2(x);
figure,imshow(F*0.01);
ff = idct2(F);
figure,imshow(ff/255);

[r,c] = size(x);
DF = zeros(r,c);
DFF = DF;
IDF = DF;
IDFF = DF;
depth = 4;
N = 8;

for i = 1 : N : r
    for j = 1 : N : c
        f = x(i:i+N-1,j:j+N-1);
        df = dct2(f);
        DF(i:i+N-1,j:j+N-1) = df;
        dff = idct2(df);
        DFF(i:i+N-1,j:j+N-1) = dff;

        df(N:-1:depth+1,:) = 0;
        df(:,N:-1:depth+1) = 0;
        IDF(i:i+N-1,j:j+N-1) = df;
        dff = idct2(df);
        IDFF(i:i+N-1,j:j+N-1) = dff;
    end
end
```

3 usages of "r" found

Fig 3.4 JPEG Compression algorithm implementation

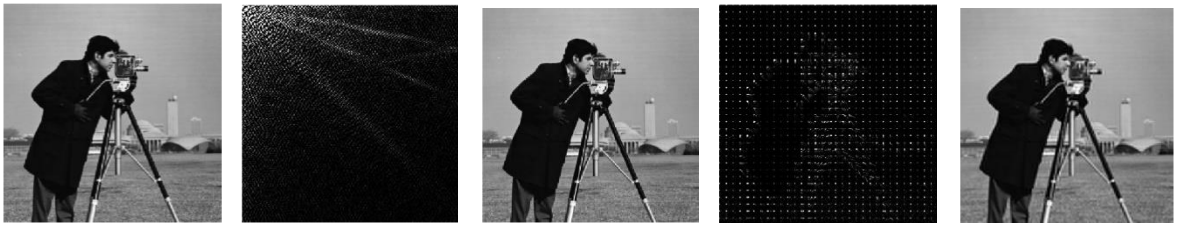


Fig 3.5 Shows results of JPEG compression over an image

d = 00000000011111111111000000000000000111000000000000010011111111111,

it can be compactly represented as c(d) = (9, 11, 15, 3, 13, 1, 2, 10) by simply listing the lengths of alternate runs of 0’s and 1’s. While the original binary data d requires 65 bits for storage, its compact representation c(d) requires 32 bits only under the assumption that each length of run is being represented by 4 bits. The early facsimile compression standard (CCITT Group 3, CCITT Group 4) algorithms were developed based on this principle.

3.1.4 Huffman Encoding and Decoding

From Shannon’s Source Coding Theory, we know that a source can be coded with an average code length close to the entropy of the source. In 1952, D. A. Huffman invented a coding technique to produce the shortest possible average code length given the source symbol set and the associated probability of occurrence of the symbols. Codes generated using this coding technique are popularly known as Huffman codes. Huffman coding technique is based on the following two observations regarding optimum prefix codes.

- The more frequently occurring symbols can be allocated with shorter codewords than the less frequently occurring symbols.
- The two least frequently occurring symbols will have codewords of the same length, and they differ only in the least significant bit.

Average length of these codes is close to entropy of the source. Let us assume that there are m source symbols {s1,s2,…..s}with associated probabilities of occurrence {p1,p2,….,pm}. Using these probability values, we can generate a set of Huffman codes of the source symbols. The Huffman codes can be mapped into a binary tree, popularly known as the Huffman tree. We describe the algorithm to generate the Huffman tree and hence the Huffman codes of the source symbols below. We show a Huffman tree in Figure 3.1.

1. Produce a set N = {N1, N2,…., N,} of m nodes as leaves of a binary tree. Assign a node Ni with the source symbol SI I = 1, 2, +…. + m and label the node with the associated probability pi. (Example: As shown in Figure 2.1, we start with eight nodes N0, N1, N2, N3, N4, N5, N6, N7 corresponding to the eight source symbols a, b, c, d, e, f, g, h, respectively. Probability of occurrence of each symbol is indicated in the associated parentheses.)
2. Find the two nodes with the two lowest probability symbols from the current node set, and produce a new node as a parent of these two nodes. (Example: From Figure 2.1 we find that the two lowest probability symbols g and d are associated with nodes N6 and N3 respectively. The new node N8 becomes the parent of N3 and N6.)
3. Label the probability of this new parent node as the sum of the probabilities of its two child nodes. (Example: The new node NS is now labeled by probability 0.09, which is the sum of the probabilities 0.06 and 0.03 of the symbols d and g associated with the nodes N3 and N6 respectively.)
4. Label the branch of one child node of the new parent node as 1 and the branch of the other child node as 0. (Example: The branch N3 to N8 is labeled by 1 and the branch N6 to N8 is labeled by 0.)
5. Update the node set by replacing the two child nodes with smallest probabilities by the newly generated parent node. If the number of nodes remaining in the node set is greater

than 1, go to Step 2. (Example: The new node set now contains the nodes N0, N1, N2, N4, N5, N7, N8 and the associated probabilities are 0.30, 0.10, 0.20, 0.09, 0.07, 0.15, 0.09, respectively. Since there are more than one node in the node set, Steps 2 to 5 are repeated and the nodes N9, N10, N11, N12, N13, N14 are generated in the next six iterations, until the node set consists only of N14.)

6. Traverse the generated binary tree from the root node to each leaf node Ni, i = 1, 2, ’.. , m, to produce the codeword of the corresponding symbol si, which is a concatenation of the binary labels (0 or 1) of the branches from the root to the leaf node. (Example: The Huffman code of symbol h is 110, formed by concatinating the binary labels of the branches N14 to N13, N13 to N11 and N11 to N7)

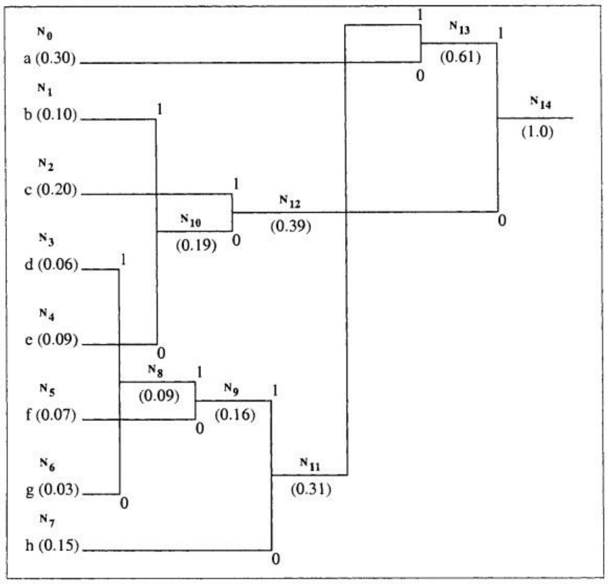


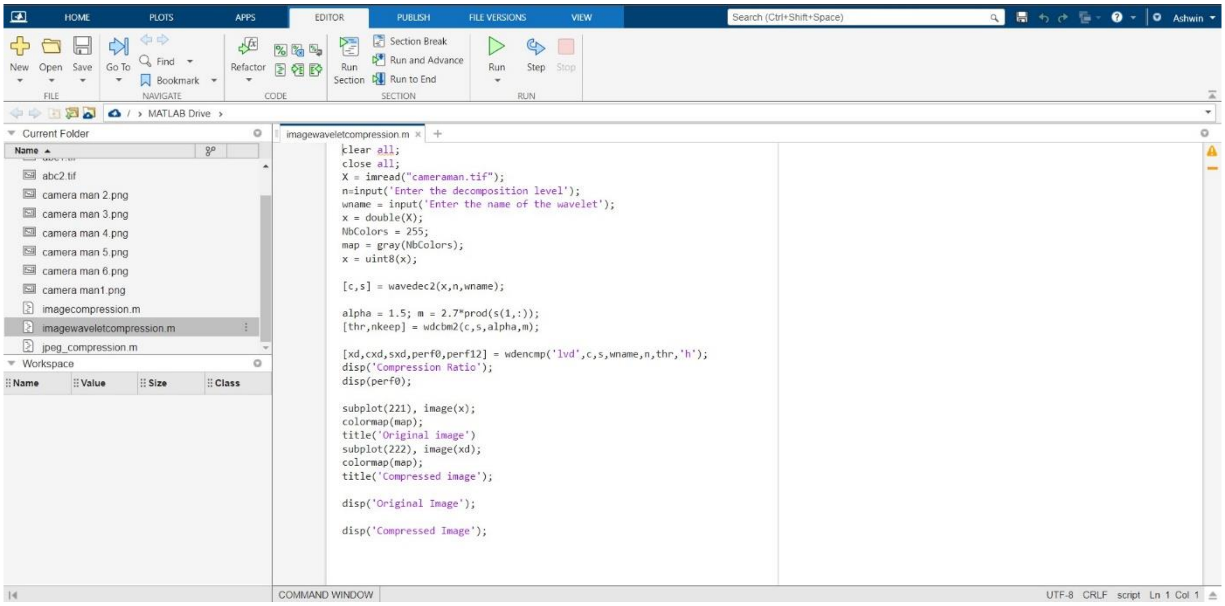
Fig 3.6 Huffman tree construction

It is needless to mention that any ensemble of binary codes, which can be mapped into a binary tree, consists of prefix codes. Hence Huffman code is HUFFMAN CODlNG 27 Symbol Probability a 0.30 b 0.10 C 0.20 d 0.06 e 0.09 f 0.07 g 0.03 h 0.15 also a prefix code. The Huffman code generation process described above is a bottom-up approach, since we perform the code construction process on the two symbols with least probabilities.

3.2 Image compression using wavelet transform

The key difference between current JPEG and JPEG2000 starts with the adoption of discrete wavelet transform (DWT) instead of the 8 x 8 block based discrete cosine transform (DCT). As we discussed earlier, the DWT essentially analyzes a tile (image) component to decompose it into a number of sub bands at different levels of resolution. The two-dimensional DWT is performed by applying the one-dimensional DWT row-wise and then column wise in each component as shown in Figure 4.4. In the first level of decomposition, four sub bands LL1, HL1, LH1, and HH1 are created. The low-pass sub band (LL1) represents a 2:l subsampled in both vertical and horizonal directions, a low-resolution version of the original component. This is an approximation of the original image in subsampled form. The other sub bands (HL1 , LH1 , HH1) represent a down sampled residual version (error because of coarser approximation) of the original image needed for the perfect reconstruction of the original image. The LL1 sub band can again be analyzed to produce four sub bands LL2, HL2, LH2, and HH2, and the higher level of decomposition can continue in a similar fashion. Typically, we don’t get much compression benefit after five levels of decomposition in natural images. However, theoretically it can go even further. The maximum number of levels of decomposition allowed in Part 1 is 32. In Part 1 of the JPEG2000 standard, only power of 2 dyadic decomposition in multiple levels of resolution is allowed.

The discrete wavelet transform (DWT) became a very versatile signal processing tool after Mallat proposed the multiresolution representation of signals based on wavelet decomposition. The method of multiresolution is to represent a function (signal) with a collection of coefficients, each of which provides information about the position as well as the frequency of the signal (function). The advantage of the DWT over Fourier transformation is that it performs multiresolution analysis of signals with localization both in time and frequency, popularly known as time-frequency localization. As a result, the DWT decomposes a digital signal into different sub bands so that the lower frequency sub bands have finer frequency resolution and coarser time resolution compared to the higher frequency sub bands. The DWT is being increasingly used for image compression due to the fact that the DWT supports features like progressive image transmission (by quality, by resolution), ease of compressed image manipulation] region of interest coding, etc. Because of these characteristics, the DWT is the basis of the new JPEG2000 image compression standard.



```
clear all;
close all;
X = imread("cameraman.tif");
n=input('Enter the decomposition level');
wname = input('Enter the name of the wavelet');
x = double(X);
NbColors = 255;
map = gray(NbColors);
x = uint8(x);

[c,s] = wavedec2(x,n,wname);

alpha = 1.5; m = 2.7*prod(s(1,:));
[thr,nkeep] = wdcbm2(c,s,alpha,m);

[xd,cxd,sxd,perf0,perf12] = wdencmp('lvd',c,s,wname,n,thr,'h');
disp('Compression Ratio');
disp(perf0);

subplot(221), image(x);
colormap(map);
title('Original image')
subplot(222), image(xd);
colormap(map);
title('Compressed image');

disp('Original Image');

disp('Compressed Image');
```

Fig 3.7 JPEG compression using Discrete Wavelet Transform

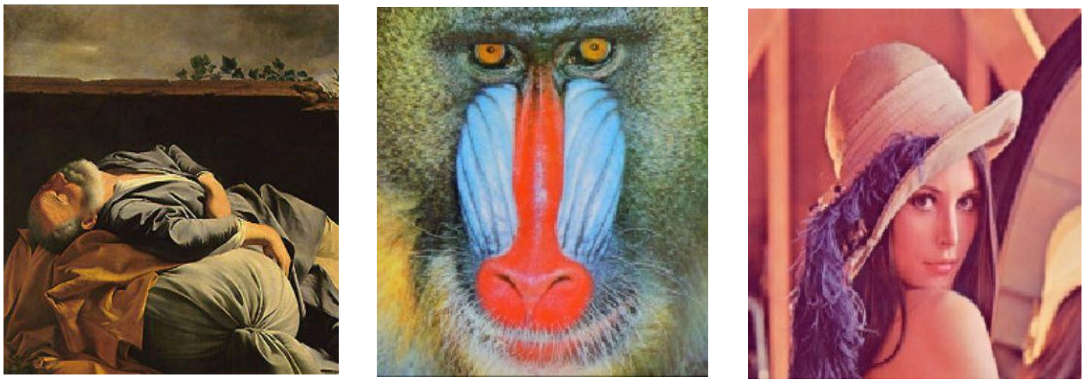


Fig 3.8 Shows image compression with DWT algorithm

Table 3: DCT and DWT PSNR and MSE result comparison

DCT (db units)	Lena	Baboon	Poor
PSNR	47.1225	47.0003	45.9991
MSE	0.0020	0.0155	0.1155

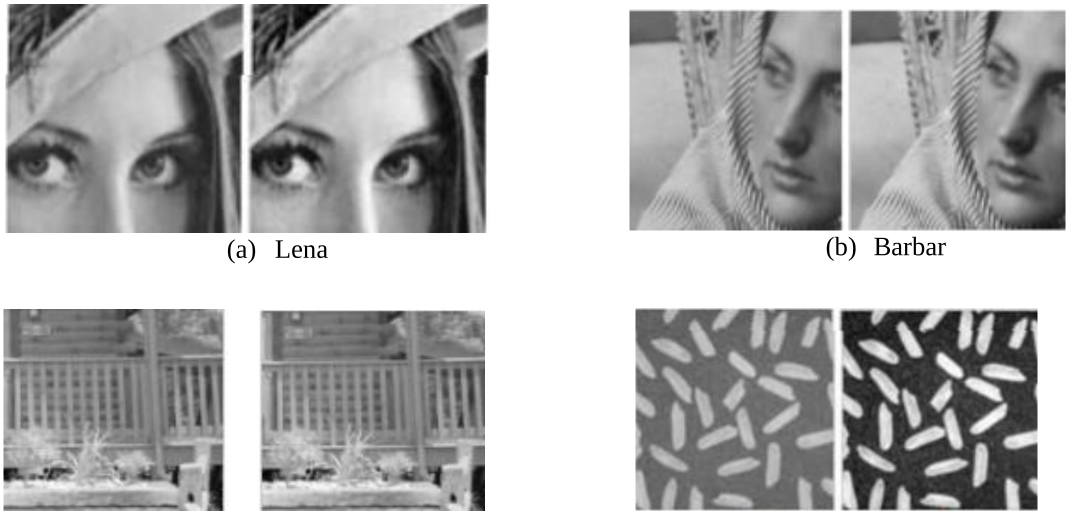
PSNR and MSE after performing DCT

DWT	Lena	Baboon	Poor
PSNR	41.1225	40.0003	39.9991
MSE	0.9995	1.1020	1.1150

PSNR and MSE after performing DWT

3.3 Results and discussion based on JPEG 2000

The JPEG 2000 standard works on the image or image tiles. Image tiles refer to the partition of the input image into nonoverlapping blocks. These image tiles are compressed and reconstructed independently as distinct images. All tiles have same size except the tiles which are located in the image boundary. Tiling used for reducing the memory requirements, and decoding specific portion of the image. Dimension of the image tiles affects the image quality. More tiling artifacts created by the smaller tiles compared to larger tiles (visually better). Every sample in the image tiles are DC level shifted by subtracting the quantity 2m-1, m is sample precision.



(a) Lena

(b) Barbar

(c) Cabin

(d) Rice

Fig 4.1 Compression results of Lena, Barbara, Cabin, Rice

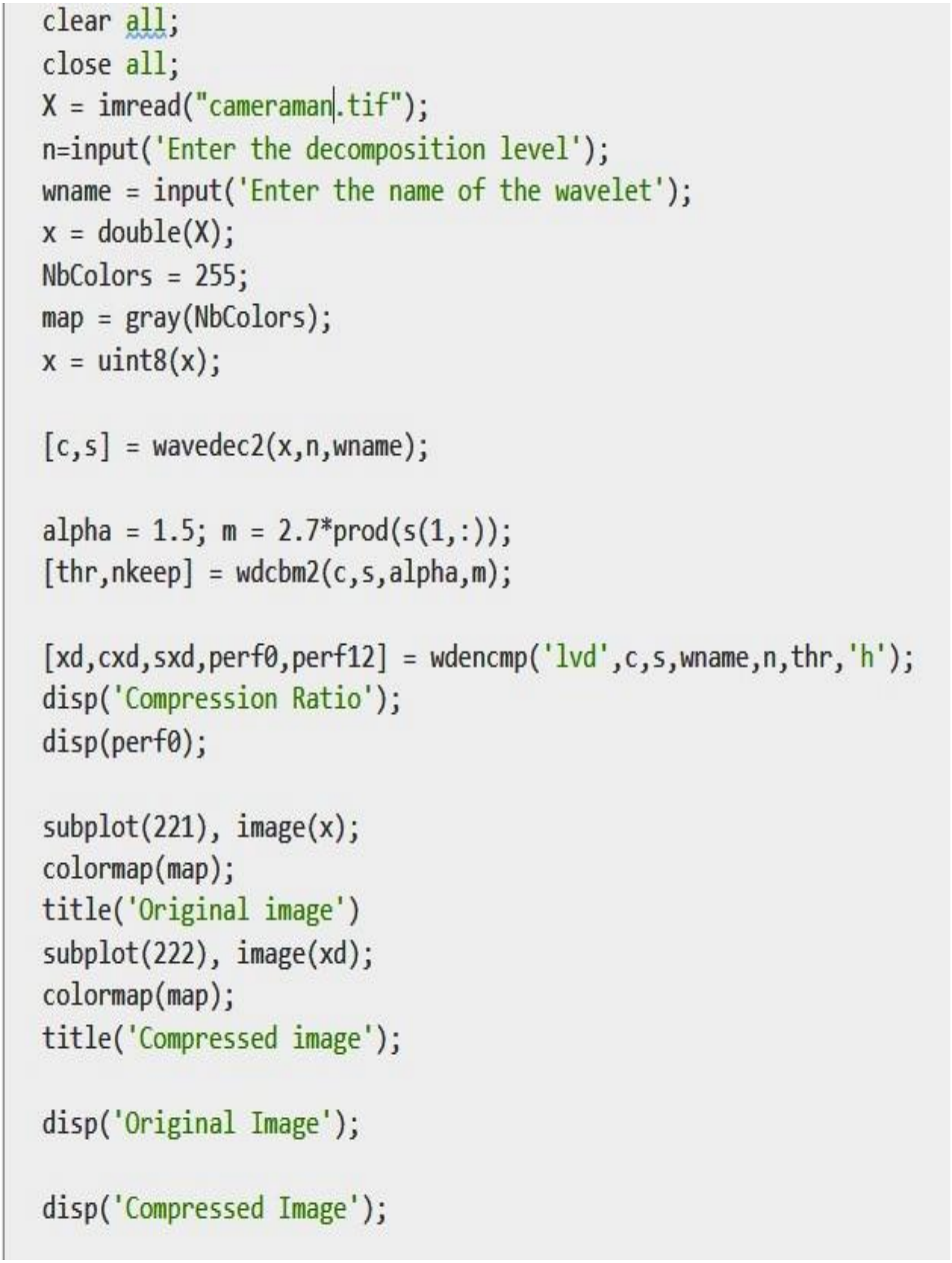
The implementation results for four different test images lena, barbara, rice and cabin of size 128*128 and 256*256 are presented. The JPEG 2000 algorithm has been applied with lifting implementation of db 5/3 wavelet, scalar quantization with bit plane coding. The results have been taken for various test images the input images and reconstructed images taken from the DSP kit. Here original image is reconstructed perfectly. Table III compares the results of JPEG 2000 algorithm, implemented in DSP hardware as well as MATLAB. It is seen that the results taken from the DSP kit is superior to the results from MATLAB. This implementation can be applied for real time image acquisition and transmission at different bit rates. Moreover, images acquired through camera can be compressed and transmitted or stored. The hardware implementation makes the above operations faster making it suitable for real time.

Test images	*Image of size 128 * 128*		*Image of size 256 *256*	
	PSNR (dB) (Hardware)	*PSNR (dB) (MATLAB)*	*PSNR (dB) (Hardware)*	*PSNR (dB) (MATLAB)*
Lena	*43.85*	*41.07*	*43.64*	*40.79*
Barbara	*44.15*	*41.32*	*44.26*	*41.92*
Rice	*43.59*	*40.06*	*43.64*	*40.10*
Cabin	*43.72*	*40.84*	*43.82*	*41.32*

Table 4: Comparison of the results in MATLAB code environment

3.4 Discrete Wavelet Transform

Wavelet transform is mathematical tool for decomposing a signal into multi-resolution representation. In DWT, image or image tile components are decomposed into different decomposition levels [2]. Decomposition levels contain a number of sub bands which consists of coefficients. These coefficients describe the horizontal, vertical and spatial frequency characteristics. The DWT can be reversible or irreversible. The 9-tap/7-tap Daubechies wavelet filter is used for irreversible (lossy) transform and 5-tap/3-tap integer wavelet filter is used for reversible (lossless) transform. The Daubechies 9/7 analysis and synthesis filter coefficients are given in Table I and Daubechies 5/3 analysis and synthesis filter coefficients are given in Table II. The JPEG 2000 standard supports convolution-based and lifting-based filtering modes. For implementing both the modes the signal should be extended periodically. Convolution-based filtering is implemented by taking a series of dot products between the filter masks and the extended input signal. Lifting-based filtering is simple filtering operations, odd sample value of the signal are updated with a weighted sum of even sample values, and even sample values are updated with a weighted sum of odd sample values. Lifting based filtering requires less memory, less computation and easily adapted to produce integer-to-integer wavelet transforms for lossless compression.



```
clear all;
close all;
X = imread("cameraman.tif");
n=input('Enter the decomposition level');
wname = input('Enter the name of the wavelet');
x = double(X);
NbColors = 255;
map = gray(NbColors);
x = uint8(x);

[c,s] = wavedec2(x,n,wname);

alpha = 1.5; m = 2.7*prod(s(1,:));
[thr,nkeep] = wdcbm2(c,s,alpha,m);

[xd,cxd,sxd,perf0,perf12] = wdencmp('lvd',c,s,wname,n,thr,'h');
disp('Compression Ratio');
disp(perf0);

subplot(221), image(x);
colormap(map);
title('Original image')
subplot(222), image(xd);
colormap(map);
title('Compressed image');

disp('Original Image');

disp('Compressed Image');
```

Fig 4.2 MATLAB code for JPEG compression using DWT

3.5 Summary

In this chapter we discussed the theoretical foundation of the discrete wavelet transform (DWT) both for convolution and lifting-based approaches. We discussed the multiresolution analysis feature of the wavelet transform, which makes it suitable for its application in image compression. We have discussed the pyramid algorithm for implementation of the DWT using the multiresolution approach. We have also discussed how the DWT is extended to two-dimensional signals as well. The multiresolution analysis-based discrete wavelet transform is the foundation of the new JPEG2000 standard. Lifting based implementation of discrete wavelet transform is new and became very popular for a number of efficient features in it. We described the underlying theory behind the lifting algorithm for DWT and showed how it is implemented via banded matrix multiplication.

CHAPTER 4. CONCLUSION

As we know image Processing refers to processing an image into digital image. Image Compression is reducing the amount of data necessary to denote the digital image. Image Compression techniques to reduce redundancy in raw Image. The encoder is used to exchange the source data into compressed bytes. The decoder decodes the compression form into its original Image sequence. Data compression is achieved by removing redundancy of Image.

The above written Image compression is basically based on two compression mechanism that are lossless compression and lossy compression. There are several image compression algorithms some of them are lossy and some are lossless such as fractal image compression, transform-based image compression (DCT, DWT)), image compression using wavelet coding. The major algorithm that we used in image compression for our project is DCT (DISCRETE COSINE TRANSFORM) and DWT (DISCRETE WAVELET TRANSFORM) based technique.

The existing JPEG standard is based on discrete Cosine Transform (DCT) scheme that is block based and leads to blocking artefacts at low bitrates. We used JPEG 2000 as our baseline coder which uses the three steps in the decompression algorithm are decoding the bit stream, dequantization, and transforming from a frequency representation back to a spatial representation including colour space conversion, chrominance downscaling, Discrete Cosine Transform, quantization, run – length and Huffman Encoding.

For our project we used DWT that is Discrete Wavelet Transform IN JPEG 2000 baseline coder model after chrominance downscaling with following process of image compression. Although DWT is a lossless based technique compression mechanism but here using in JPEG 2000 coder model the image data loss is minimum, that is giving result in more reduced PSNR and MSE, lesser or reduced size and improved compression ratio of compressed image than JPEG 2000 coder model with DCT.

CHAPTER 5. REFFERENCES

1. W. B. Pennebaker and J. L. Mitchell, JPEG Still Image Data Compression Standard. Chapman & Hall, New York, 1993
2. D. A. Huffman, “A Method for the Construction of Minimum-Redundancy Codes,” Proceedings of the IRE, Vol. 40, No. 9, pp. 1098-1101, 1952
3. Ali Bilgin , M. P. Boliek, M. W. Marcellin, M. J. Gormish ”An Overview of JPEG 2000,”Proc.of IEEE Data Compression Conference, 2000, pp. 523-541
4. David Taubman , “High Performance Scalable Image Compression with EBCOT”, IEEE Transactions on image Processing ,VOL. 9,and NO.7, July 2000, pp. 1158-1170
5. Verma O.P., Agarwal R., Dafouti D., “Performance analysis of data encryption algorithms”, ICECT, vol. 5, pp. 399-403, IEEE, 2011
6. NavitaAgarwal, Himanshu Sharma “An Efficient Pixelshuffling Based Approach to Simultaneously Perform Image Compression, Encryption and Steganography”, IJCSMC, vol. 2 issue 5,pp. 376-385, May 2013.
7. Ashwin S., “Novel and secure encoding and hiding techniques using image steganography: A survey”, ICETEEEM, vol. 1, pp. 171-177, IEEE, 2012