IMAGE COMPRESSION USING DISCRETE COSINE TRANSFORM AND WAVELET BASED TRANSFORM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology
in
Computer Science Engineering

By

ANSHUMAN, Roll No : 10306001
GAURAV JAISWAL, Roll No : 10306004
ANKIT RAI, Roll No : 10206028

Department of Computer Science Engineering
National Institute of Technology, Rourkela

May, 2007

IMAGE COMPRESSION USING DISCRETE COSINE TRANSFORM AND WAVELET BASED TRANSFORM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology

In

Computer Science Engineering

By

ANSHUMAN, Roll No : 10306001

GAURAV JAISWAL, Roll No : 10306004

ANKIT RAI, Roll No : 10206028

Under the Guidance of

Prof. R. Baliarsingh

Department of Computer Science Engineering

National Institute of Technology, Rourkela

May,2007

National Institute of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled **"Image compression using discrete cosine transform and wavelet transform and performance comparison''** Submitted by **Anshuman**, **Roll No: 10306001, Gaurav Jaiswal, Roll No: 10306004 & Ankit Rai, Roll No: 10206028** in the partial fulfillment of the requirement for the degree of **Bachelor of Technology** in **Computer Science Engineering**, National Institute of Technology, Rourkela, is being carried out under my supervision.

To the best of my knowledge the matter embodied in the thesis has not been submitted to any other university/institute for the award of any degree or diploma.

Professor R. Baliarsingh
Date Department of Computer Science Engineering
National Institute of Technology
Rourkela-769008

Acknowledgment

We avail this opportunity to extend our hearty indebtedness to our guide **Professor R. Baliarsingh**, Computer Science Engineering Department, for their valuable guidance, constant encouragement and kind help at different stages for the execution of this dissertation work.

We also express our sincere gratitude to **Dr. S.K.JENA**, Head of the Department, Computer Science Engineering, for providing valuable departmental facilities.

Submitted by:

Anshuman
Roll No: 10306001
Computer Science Engineering
National Institute of Technology
Rourkela

Gaurav Jaiswal
Roll No: 10306004
Computer Science Engineering
National Institute of Technology
Rourkela

Ankit Rai
Roll No: 10206028
Computer Science Engineering
National Institute of Technology
Rourkela

CONTENTS

ABSTRACT

Image compression deals with reducing the size of image which is performed with the help of transforms. In this project we have taken the Input image and applied wavelet techniques for image compression and have compared the result with the popular DCT image compression. WT provided better result as far as properties like RMS error, image intensity and execution time is concerned. Now a days wavelet theory based technique has emerged in different signal and image processing application including speech, image processing and computer vision. In particular Wavelet Transform is of interest for the analysis of non-stationary signals. In the WT at high frequencies short windows and at low frequencies long windows are used. Since discrete wavelet is essentially sub band–coding system, sub band coders have been quit successful in speech and image compression. It is clear that DWT has potential application in compression problem.

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 BACKGROUND

Uncompressed graphics, audio and video data require considerable storage capacity and transmission bandwidth. Despite rapid progress in mass storage density, processor speeds and digital communication system performance, demand for data storage capacity and data transmission bandwidth continues to out strip the capabilities of the available technologies. The recent growths of data intensive digital audio, image, and video based (multimedia) web applications, have sustained the need for more efficient ways. With the growth of technology and the entrance into the Digital Age, the world has found itself amid a vast amount of information. Dealing with such enormous amount of information can often present difficulties. Digital information must be stored and retrieved in an efficient manner in order to put it to practical use. Wavelet compression is one way to deal with this problem. For example, the FBI uses wavelet compression to help store and retrieve its fingerprint files. The FBI possesses over 25 million cards, each containing 10 fingerprint impressions. To store all of the cards would require over 250 Terabytes of space. Without some sort of compression, sorting, storing and searching for data would be nearly impossible. Typically television image generates data rates exceeding 10million bytes/sec. There are other image sources that generate even higher data rates. Storage and transmission of such data require large capacity and bandwidth, which could be expensive. Image data compression technique, concerned with the reduction of the number of bits required to store or transmit image without any appreciable loss of information. . Using wavelets, the FBI obtains a compression ratio of about 1: 20

1.2 NEED FOR COMPRESSION

The amount of data associated with visual information is so large that its storage would require enormous storage capacity. Although the capacities of several storage media are substantial, their access speeds are usually inversely proportional to their capacity. Typical television images generate data rates exceeding 10 million bytes per second. There are other image sources that generate even higher data rates. Storage and/or transmission of such data require large capacity and/or bandwidth, which could be very expensive. Image data compression techniques are concerned with reduction of the number of bits required to store or transmit images without any appreciable loss of information. Image transmission applications are in broadcast television; remote sensing via satellite, aircraft, radar or sonar;

teleconferencing; computer communications; and facsimile transmission. Image storage is required most commonly for educational and business documents, medical images used in patient monitoring systems, and the like. Because of their wide applications, data compression is of great importance in digital image processing.

The figures in the Table below show the qualitative transition from simple text to full-motion video data and the disk space needed to store such uncompressed data.

Multi-media data	Size/duration	Bits/pixel Bits/sample	Uncompressed size
Page of text	11”x8.5”	Varying resolution	16-32 kbits
Telephone quality speech	1 sec	8 bps	64 kbit
Grayscale image	512x512	8 bpp	2 mbits
Color image	512x512	24 bpp	6.29 mbits
Full motion video	640x480,10 sec	24 bpp	2.21 gbits

Table 1.1 **:** Multimedia data types and uncompressed storage space required

The examples above clearly illustrate the need for large storage space for digital image, audio and video data. So at the present state of technology, the only solution is to compress these multimedia data before its storage and transmission, and decompress it at the receiver for playback. With a compression ratio of 16:1, the space requirement can be reduced by a factor of 16 with acceptable quality.

1.3 PRINCIPLES OF COMPRESSION

The amount of data associated with visual information is so large that its storage would require enormous storage capacity. Although the capacities of several storage media are substantial, their access speeds are usually inversely proportional to the capacity.

Typical television image generate data rates exceeding 10 million bytes per second. There are other image sources that generate even higher data rates. Storage and transmission of such data require large capacity and bandwidth which could be very expensive.

Image data compression techniques are concerned with reduction of the number of bits required to store or transmit images without any appreciable loss of information. The underlying basis of the reduction process is the removal of redundant data, i.e. the data that either provides no relevant information or simply restate that which is already known. Data redundancy is the central issue in digital image compression. If n1 and n2 denote the number of information carrying units in two data sets that represent the same information, then the compression ratio is defined as below:

$$\mathbf{C_R = n_1 / n_2}$$

In this case, relative data redundancy RD of the first data set can be defined as follows:

$$\mathbf{R_D = 1 - 1/ C_R}$$

When n2=n1 then CR=1 and hence RD=0. It indicates that the first representation of the information contain no redundant data.
When n2<<n1 then CR->∞ and hence RD->1. It implies significant compression and highly redundant data.
In the final case when n1<<n2 then CR->0 and hence RD->-∞, indicating that the second data set contains much more data than the original representation.

Various methods can be used for the compression of the image that contains redundant data. Here we use the *Discrete Cosine Transform* (DCT) method to get a compressed image of an original image.

A common characteristic of most images is that the neighboring pixels are highly correlated and therefore contain highly redundant information. The foremost task is to find an image representation in which the image pixels are decorrelated. Redundancy and irrelevancy reductions are two fundamental approaches used in compressions. Where as redundancy reduction aims at removing redundancy from the signal source (image or video), irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver. In general three types of redundancy in digital images and video can be identified:

- **Spatial redundancy** or correlation between neighboring pixel values.
- **Spectral redundancy** or correlation between different color planes or spectral bands.
- **Temporal redundancy** or correlation between adjacent frames in a sequence of energies.

Image compression aims at reducing the number of bits needed to represent the image by removing the spatial and spectral redundancies as much as possible.

1.4 COMPRESSION TECHNIQUES

There are different ways of classifying compression techniques. Two of this would be mentioned here.

1.4.1 LOSSLESS VS LOSSY COMPRESSION

The first categorization is based on the information content of the reconstructed image. They are *lossless compression* and *lossy compression* scheme. In lossless compression, the reconstructed image after compression is numerically identical to the original image on a pixel by pixel basis. However, only a modest amount of compression is achievable in this technique. In lossy compression, on the other hand, the reconstructed image contains degradation relative to the original, because redundant information is discarded during compression. As a result, much higher compression is achievable and under normal viewing conditions no visible loss is perceived (visually lossless).

1.4.2 PREDICTIVE VS TRANSFORM CODING

The second categorization of various coding schemes is based on the space where the compression method is applied. These are *predictive coding* and *transform coding*. In predictive coding, information already sent or available is used to predict future values and the differences are 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 of predictive coding. Transform coding, on the other hand, first transforms the image from its spatial domain representation to a different type of representation using some well known transforms mentioned later, and codes the transform values (coefficient). The primary advantage is that it provides greater data compression as compared to the predictive method, although at the expense of greater computation.

1.5 AN INTRODUCTION TO IMAGE

Before talking about different types of images and their applications lets first examine the sampling mechanism by which the image is converted to data and the limitations of this process.

1.5.1 SAMPLING AND QUANTIZATION

Sampling is the process of examining the values of continuous functions at regular intervals.

Quantization is the process of limiting the value of function at any sample to one of a predetermined number of permissible values, so that it cam be represented by a finite no. of bits in the digital world.

1.5.2 SAMPLING RATE AND ALIASING

When a signal is sampled, it has values only at specific points in time or space. Between the samples, there is no knowledge about what has happened.

In fact, the maximum bandwidth of a sampled waveform is determined exactly by its sampling rate, the max. frequency representable in a sampled waveform is termed its Nyquist Frequency, and is equal to one half the sampling rate. Thus, for ex, a waveform sampled at 16,000 Hz cam represent all frequencies upto its Nyquist Frequency of 8,000 Hz. A problem called aliasing occurs whaen a signal o be sampled contains energy at frequencies abobve the sampling Nyquist frequency. When the sampling rate is much too low for the frequency of an input signal.

Obviously, Aliasing has the effect of producing sounds of lower frequency that are higher in frequency than the Nyquist Frequency. Once aliasing has occurred, it is absolutely impossible to distinguish a component generated by alisasing from one that was actually present in the input signal. This effect is one of the mmost common sourceds of distortion in digitzed waveforms. Fortunately, most modern computer hardware for digitizing sound has built in filters which are tuned to remove sound energy at frequencies beyond the nyquist frequency for whatever sampling rate is being used.

1.5.3 TWO-DIMENSIONAL SAMPLING

If we have image, rather than just a waveform, we need to sample it in two dimensions, along two axes usually designated as X and Y. Generally the image can be represented by the smallest no. of samples if the row sampling axes are orthogonal, horizontal and vertical. For any sampling direction, Aliasing can be avoided only if it obeys Nyquist theorem. Generally, in image processing the sampling rate is the square, or approximately so. In olther words sampling in the X direction are spaced the same, or nearly the same as those in the Y direction.

1.6 QUALITY MEASURES IN IMAGE CODING

In order to measure the quality of the image or video data at the output of gthe decoder, mean sq error (MSE) and peak to signal to noise ratio(PSNR ratio) are often used. The MSE is often called quantization error variance $\sigma^2 q$. The MSE between the original image f and the reconstructed image g at decoder is defined as

$$MSE = \sigma^2_q = 1/N \sum (f[j,k] - g[j,k])^2$$

Where the sum over j,k denotes the sum over all pixels in the image and N is the no. of pixels in each image. The PSNR between two images having 8 bits per pixels aor samples in term of decibels(dBs) is given by:

$$PSNR = 10\log_{10} (255^2 / MSE)$$

Generally when PSNR is 40 dB or greater, than the original and the reconstructed images are virtually indistinguishable by human observers.

Signal to noise ratio(SNR) ratio is also a measure, bbut it is mostly used in telecommunications. However, one can calculate SNR for an image in terms of decibels(dBs) as : SNR =$10\log_{10}$(Encoder input image energy or variance/Noise energy or variance)

1.7 IMAGE COMPRESSION THEORY

Underlying basis of the reduction process is the removal of redundant data i.e., the data that either provides no relevant information or simply restart that which is already known. Data redundancy is the central issue in digital image compression. If n1 and n2 denote the number of information carrying units in two data sets that represent the same information, then the compression ration C_R is defined as below.

$$C_R = n1/n2 \qquad (1.3)$$

In this case relative data redundancy RD of the first data set can be defined as follows.

$$R_D = 1 - 1/C_R \qquad (1.4)$$

When n2 = n1, then $C_R = 1$ and hence $R_D = 0$. It indicates that the first representation of the information contains no redundant data.

When n2 << n1, then $C_R \to \infty$ and $R_D \to 1$. It implies significant compression and highly redundant data. In the final case when n2 << n1, then $C_R \to 0$ and $R_D \to -\infty$, indicating that the second data set contains much more data than the original representation. Various methods can be used for the compression of the image that contains redundant data.

1.8 A TYPICAL IMAGE CODER

How does a typical image coder look like? A typical lossy image compression system shown in figure, consist of three closely connected components: (a) Source Encoder or Linear Transforms (b) Quantizer and (c) Entropy Encoder

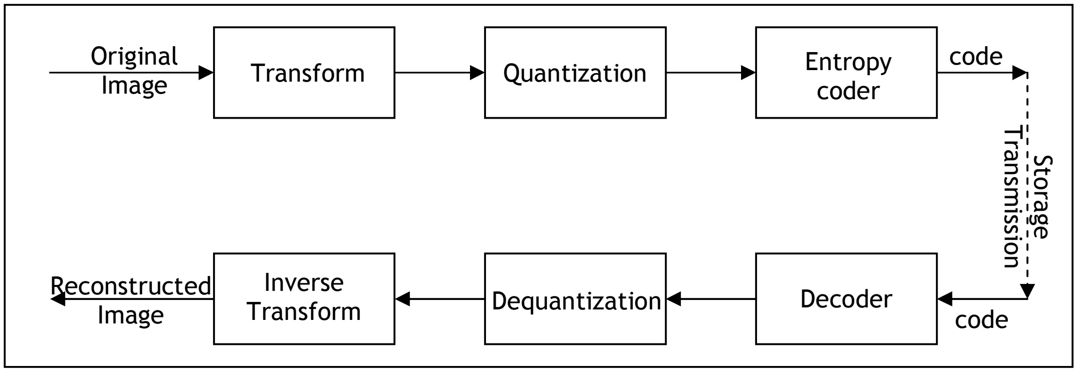

Fig 1.1 : A Typical Image Coder

A Quantizer simply reduces the number of bits needed to store the transformed coefficients by reducing the precision of those values. Since this is a many-to-one mapping, it's a lossy process and is the main source of compression in an encoder. Quantization can be performed on each individual coefficient, which is known as Scalar Quantization (SQ). Quantization can also be performed on a group of coefficients together, and this is known as Vector Quantization (VQ). Both, uniform and non-uniform quantizer can be used depending on problem at hand.

An Entropy Encoder further compresses the quantized values losslessly to give better overall compression. Most commonly used entropy encoders are the Huffman encoder and the Arithmetic encoder, although for applications requiring fast execution, simple run-length coding has proven very effective. A properly designed quantizer and entropy are absolutely necessary along with optimum signal transformation to get best possible compression.

Over the years a variety of linear transforms have been developed which include Discrete Fourier Transform (DFT),Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and many more, each with its own advantages and disadvantages.

The Discrete Cosine Transform is one of many transforms that takes the input and transforms it into a linear combination of weighted basis functions. These basis functions are commonly the frequency, like sine waves. The 2D Discrete Cosine Transform is just a one dimensional DCT applied twice, once in the x direction, and the second in the y direction.

More recently, wavelet transform has become a cutting edge technology for image compression research. It is seen that, wavelet-based coding provides substantial improvement in picture quality at higher compression ratios mainly due to the better energy compaction property of wavelet transforms. Over the past few years, a variety of powerful and sophisticated wavelet-based schemes for image compression have been developed and implemented. Because of the many advantages, the top contenders in the upcoming JPEG-2000 standard are all wavelet-based compression algorithms.

Chapter 2

THE DISCRETE COSINE TRANSFORM

2.1 INTRODUCTION

The discrete cosine transform is a fast transform that takes a input and transforms it into linear combination of weighted basis function, these basis functions are commonly the frequency, like sine waves.

It is widely used and robust method for image compression, it has excellent energy compaction for highly correlated data, which is superior to DFT and WHT. Though KLT minimizes the MSE for any input image, KLT is seldom used in various applications as it is data independent obtaining the basis images for each sub image is a non trivial computational task, in contrast DCT has fixed basis images. Hence most practical transforms coding systems are based on DCT which provides a good compromise between the information packing ability and computational complexity.

Compared to other independent transforms it has following advantages, can be implemented in single integrated circuit has ability to pack most information in fewer number of coefficients and it minimizes the block like appearance, called blocking artifact that results when the boundary between sub images become visible.

One dimensional DCT is defined as

$$c(u) = a(u) \sum_{x=0}^{N-1} f(x) \cos [(2x+1)u\pi/2N]$$

where u=0,1,2,…….,N-1

Inverse DCT is defined as

$$f(x) = \sum_{x=0}^{N-1} a(u)\, c(u) \cos [(2x+1)u\pi/2N]$$

where x=0,1,2,…….,N-1

$$a(u) = \sqrt{1/N} \quad \text{for } u = 0$$

$$a(u) = \sqrt{1/N} \quad \text{for } u=1,2,3\ldots N-1$$

The correlation between different coefficient of DCT is quite small for most of the image sources and since DCT processing is Asymptotically Gaussian. Those transformed coefficients are treated as they are mutually independent.

In general, DCT correlates the data being transformed so that most of its energy is packed in a few of its transformed coefficient’s.

The goal of the transformation process is to decorrelate the pixels of each sub images or to pack as much information as possible into the smaller number of transform coefficients.

The Quamtization stage then selectively eliminates or more coarsely quantizes the coefficients that carry the least information.these coefficients have the smallest impact on the reconstructed sub image quality.the encoding process terminates by coding the quantized coefficients

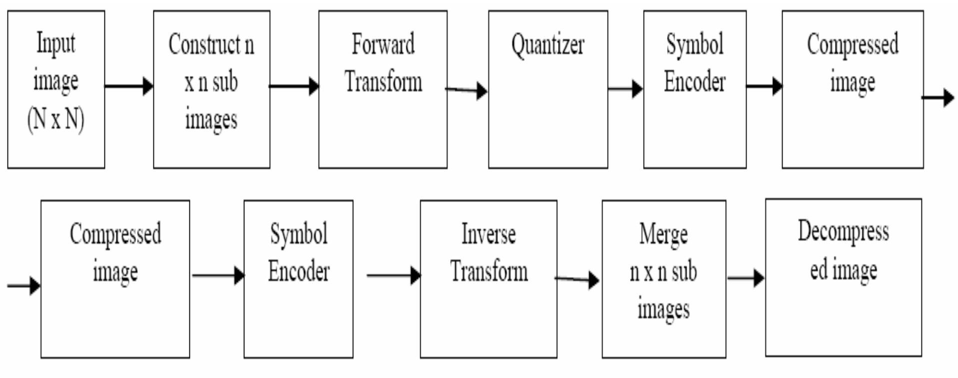

Fig 2.1 : Steps for DCT compression

2.2 COMPRESSION PROCEDURE

For a given image , you can compute the DCT of, say each row, and discard all values in the DCT that are less then a certain threshold. We then save only those DCT coefficients that are above the threshold for each row, and when we need to reconstruct the original image, we simply pad each row with as many zeroes as the number of discarded coefficients, and use the inverse DCT to reconstruct each row of the original image. We can also analyze image at the different frequency bands, and reconstruct the original image by using only the coefficients that are of a particular band. The steps for compression are as follows:

Step 1: Digitize the source image into a signal s, which is the string of numbers.

Step 2: Decompose the signal into a sequence of transform coefficients w.

Step 3: Use threshold to modify the transform coefficients from w to another sequence w′.

Step 4: Use quantization to convert w' to a sequence q.

Step 5: Apply entropy coding to compress q into a sequence e.

The detail compression steps are as follows:

Step 1 : DIGITIZATION

The first step in the image compression process is to digitize the image. The digitized image can be characterized by its intensity levels or scales of gray which range from 0(black) to 255(white), or its resolution, or how many pixels per square inch. Each of the bits involved in creating an image takes up both time and money, so a tradeoff must be made.

Step 2 : TRANSFORM

Apply DCT transform to each of the pixel values to get a set of transform coefficients. The basic motive behind transforming the pixels is to concentrate the image data spread over many pixels to a lesser number of pixels and then the pixels that do not contain and relevant data can be discarded, hence reducing the image size. Typically transforms applied are any functions that are invertible so that we can regenerate the transformed values and should be capable of concentrating the image data over a lesser area. The well known Discrete Cosine Transform and Discrete Wavelet Transform are few examples. The upcoming JPEG 2000 uses the Discrete Wavelet Transform for its compression.

Step 3 : THRESHOLDING

In certain signals, many of the transform coefficients are zero. Through a method called threshold, these coefficients may be modified so that the sequence of transform coefficients contain long strings of zeros. Through a type of compression known as entropy coding, these long strings may be stored and sent electronically in much less space. There are different types of threshold. In *hard threshold*, a tolerance is selected. Any transform coefficient whose absolute value falls below the tolerance is set to zero with the goal to introduce many zeros without losing a great amount of detail. There is not a straightforward easy way to choose the threshold, although the larger the threshold that is chosen, the more error that is introduced into the process. Another type of threshold is *soft threshold*. Once again a tolerance h is selected. If the absolute value of an entry is less than the tolerance then that entry is set to zero. All other entries, d, are replaced with sign(d)||d|-h|. Soft threshold can be thought of as a translation of the signal toward zero by the amount h. A third type of threshold is *quantile threshold*. In this method a percentage p of entries to be eliminated are selected. The smallest (in absolute value) p percent of entries are set to zero.

Step 4: QUANTIZATION

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

Step 5: ENTROPY CODING

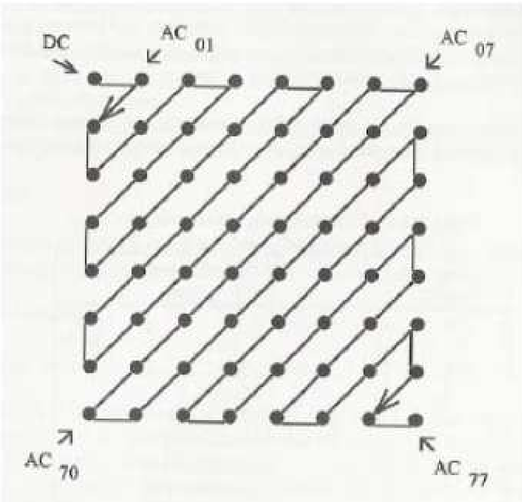

Fig 2.2 : Zigzag scan

Transforms and threshold help process the signal, but up until this point, no compression has yet occurred. One method to compress the data is Huffman entropy coding. With this method, an integer sequence, q is changed into a shorter sequence, e, with the numbers in e being 8-bit integers. The conversion is made by an entropy coding table. Strings of zeros are coded by the numbers 1 through 100, 105 and 106, while the non-zero integers in q are coded by 101 through 104 and 107 through 254. In Huffman entropy coding, the idea is to use two or three numbers for coding, with the first being a signal that a large number or zero sequence is coming. Entropy coding is designed so that the numbers that are expected to appear the most often in q need the least amount of space in e.

2.3 FORMULAES USED IN DCT COMPUTATION

The NxN cosine transform matrix **C={c(k,n)}**, also called the discrete cosine transform(DCT), is defined as

$$1/\sqrt{N}, \qquad k=0,\ 0<=n<=N-1$$
$$\sqrt{(2/N)}\cos((\prod(2n+1)k)/(2N)), \qquad 1<=k<=N-1,\ 0<=n<=N-1$$

The one-dimensional DCT of a sequence { u(n), 0<=n<=N-1 } is defined as

$$v(k)=\alpha(k) \sum u(n) \cos[(\prod(2n+1)k)/(2N)], \qquad 0<=k<=N-1$$

where

$$\alpha(0)= \sqrt{(1/N)}, \quad \alpha(k)=(2/N) \text{ for } 1<=k<=N-1$$

The inverse transformation is given by

$$u(n)= \sum \alpha(k)v(k)\cos[(\prod(2n+1)k)/(2N)], \qquad 0<=n<=N-1$$

Note that many coefficients are small, i.e. most of the data is packed in a few transform coefficients.

The two-dimensional cosine transform pair is obtained by

$$\mathbf{v(k,l) = \sum\sum a(k,m)u(m,n)a(l,n) \leftrightarrow V=CUC'} \qquad \text{eq. 1}$$

$$\mathbf{u(m,n) = \sum\sum a^*(k,m)v(k,l)a^*(l,n) \leftrightarrow U=C'VC} \qquad \text{eq. 2}$$

where C´ is the transpose of C and $\{a_{k,l}(m,n)\}$, called image transform, is a set of complete orthonormal discrete basis functions satisfying the properties

Orthonormality: $\sum\sum \mathbf{a}_{k,l}\mathbf{(m,n)a^*}_{k'l'}\mathbf{(m,n)=\delta(k-k',l-l')}$

Completeness: $\sum\sum \mathbf{a}_{k,l}\mathbf{(m,n)a^*}_{k,l}\mathbf{(m',n')= \delta(m-m',n-n')}$

The elements v(k,l) are called the transform coefficients and V={v(k,l)} is called the transformed image. The orthonormality property assures that any truncated series expansion of the form

$$\mathbf{u}_{P,Q}\mathbf{(m,n)} = \sum \sum \mathbf{v(k,l)a^*}_{k,l}\mathbf{(m,n)}, \qquad \mathbf{P<=N, Q<=N}$$

will minimize the sum of squares error

$$\boldsymbol{\sigma^2=\sum\sum [u(m,n)-u}_{P,Q}\mathbf{(m,n)]^2}$$

where the coefficients v(k,l) are given by the eqn. 1 and 2.

The completeness property assures that this error will be zero for P=Q=N.

CHAPTER 3

WAVELET BASED IMAGE COMPRESSION

3.1 WHAT IS A WAVELET TRANSFORM?

Wavelets are functions defined over a finite interval and having an average value of zero. The basic idea of the wavelet transform is to represent any arbitrary function *(t)* as a superposition of a set of such wavelets or basis functions. These basis functions or baby wavelets are obtained from a single prototype wavelet called the mother wavelet, by dilations or contractions (scaling) and translations (shifts). The Discrete Wavelet Transform of a finite length signal *x(n)* having *N* components, for example, is expressed by an *N* x *N* matrix.

Wavelets are mathematical functions that were developed by scientists working in several different fields for the purpose of sorting data by frequency. Translated data can then be sorted at a resolution which matches its scale. Studying data at different levels allows for the development of a more complete picture. Both small features and large features are discernable because they are studied separately. Unlike the Discrete Cosine Transform, the wavelet transform is not Fourier-based and therefore wavelets do a better job of handling discontinuities in data. In this section we would be employing Haar wavelet transform for image compression.

The Haar wavelet operates on data by calculating the sums and differences of adjacent elements. The Haar wavelet operates first on adjacent horizontal elements and then on adjacent vertical elements. The Haar transform is computed using:

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

One nice feature of the Haar wavelet transform is that the transform is equal to its inverse. As each transform is computed the energy in the data in relocated to the top left hand corner; i.e. after each transform is performed the size of the square which contains the most important information is reduced by a factor of 4.

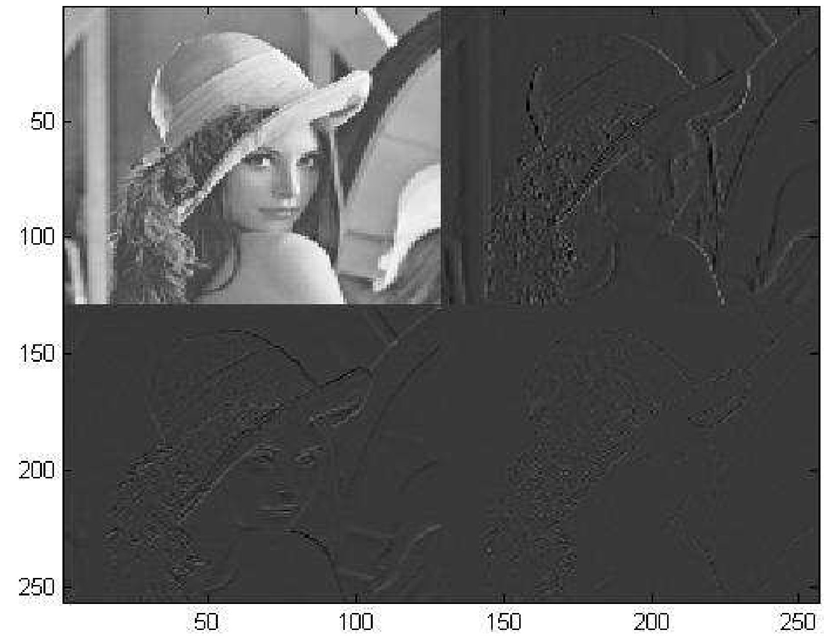

Fig 3.1 : The image “lena” after one Haar wavelet transform

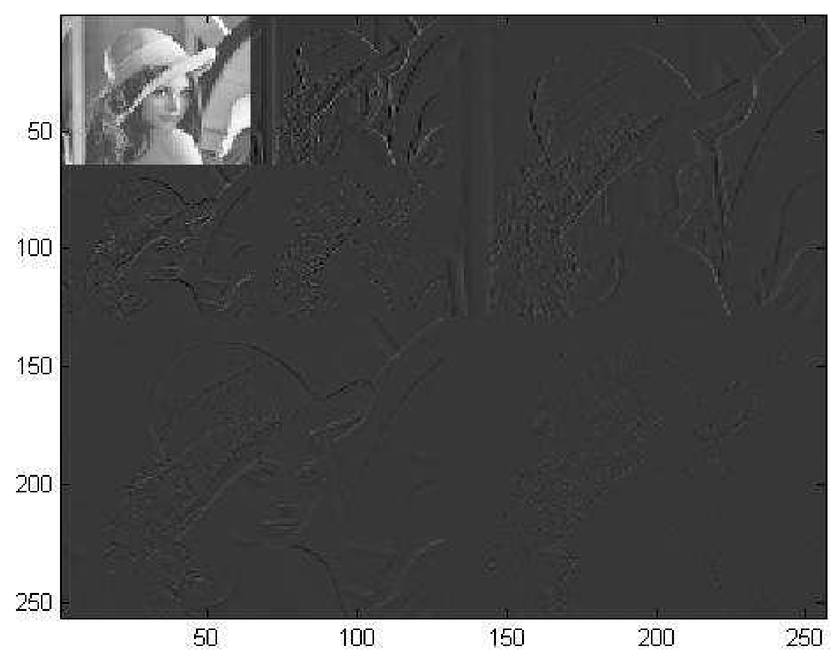

Fig 3.2 : The image “lena” after two Haar wavelet transform

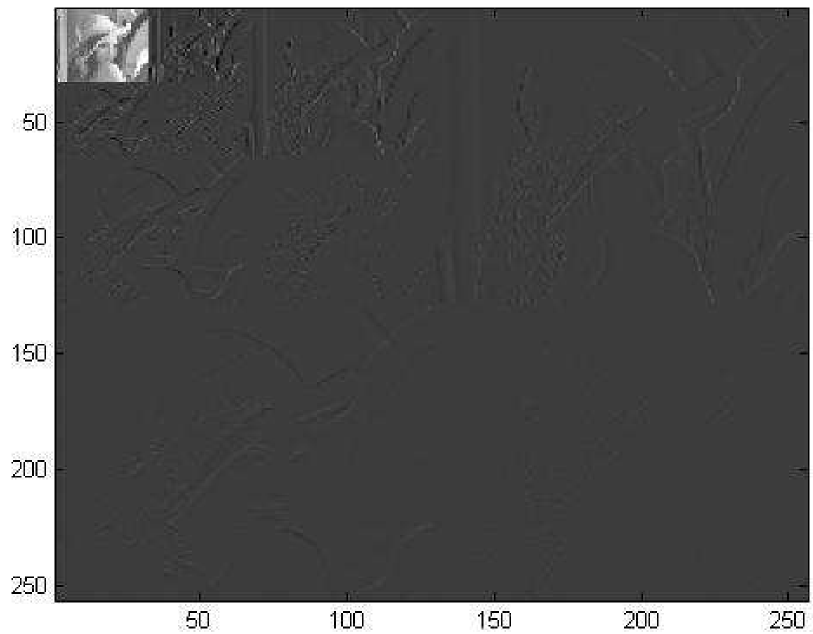

Fig 3.3 : The image “lena” after three Haar wavelet transform

3.2 WHY WAVELET-BASED COMPRESSION?

Despite all the advantages of JPEG compression schemes based on DCT namely simplicity, satisfactory performance, and availability of special purpose hardware for implementation, these are not without their shortcomings. Since the input image needs to be ``blocked," correlation across the block boundaries is not eliminated. This results in noticeable and annoying ``blocking artifacts" particularly at low bit rates. Lapped Orthogonal Transforms (LOT) attempt to solve this problem by using smoothly overlapping blocks. Although blocking effects are reduced in LOT compressed images, increased computational complexity of such algorithms do not justify wide replacement of DCT by LOT.

3.3 UNDERSTANDING THE HAAR WAVELET TRANSFORM

3.3.1 METHOD OF AVERAGING AND DIFFERENCING

The method of “Averaging and Differencing” (otherwise known as “The Haar Wavelet Transform”), by Colm Mulcahy, Ph.D, to the 8×8. To understand “Averaging and

Differencing" strip off the first row of the 8 × 8 matrix. Now form a new row by averaging each pair of numbers in the original row. This will yield a new row only half the length of the original row. Fill the remaining positions by subtracting the averages from the corresponding first element of each pair. Continue this process until all the original numbers are averaged down into one number. The remaining numbers will be subtraction differences also called "detail coefficients."

$$\begin{bmatrix} 576 & 704 & 1152 & 1280 & 1344 & 1472 & 1536 & 1536 \\ 640 & 1216 & 1408 & 1536 & -64 & -64 & -64 & 0 \\ 928 & 1472 & -288 & -64 & -64 & -64 & -64 & 0 \\ 1200 & -272 & -288 & -64 & -64 & -64 & -64 & 0 \end{bmatrix}$$

Notice that with this 1 × 8 row, three steps are needed to complete the process.

This is the idea of "Averaging and Differencing." To complete this process on the 8×8 matrix, though, the process must be applied to every row and then to every column of the new matrix. This would require repeating the previous operations 15 times. This is a lot of work, and of course linear algebra simplifies the process greatly.

Imagine an 8 × 8 matrix that could perform these operations for us. The following matrix will actually complete the first step of our process for each row.

$$A_1 = \begin{bmatrix} 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & -1/2 \end{bmatrix}$$

Refer to the original matrix as P, and the new matrix as A1. By multiplying matrix P on the right by matrix A1 the first step is completed for each row. Notice that multiplying our original first row by the matrix A1 yields the same results as shown before.

(576, 704, 1152, 1280, 1344, 1472, 1536, 1536)A1= (640, 1216, 1408, 1536, −64, −64, −64, 0)

A similar 8×8 matrix will perform the second step to each row. It will take the averages and differences of the left side of the rows and leave the right sides (detail coefficients) unchanged. Thinking in terms of block multiplication, a new matrix is easily constructed.

$$A_2 = \begin{bmatrix} 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & -1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Note the similarity between matrix A2 and matrix A1. Also notice the differences, particularly the identity matrix that is found in lower right. This is the portion of the matrix that leaves the detail coefficients unchanged. Carrying on from our previous example this point is illustrated:

(640, 1216, 1408, 1536, −64, −64, −64, 0)A2

= (928, 1472, −288, −64, −64, −64, −64, 0)

A third and last 8 × 8 matrix will complete the averaging and differencing process for the rows from the original matrix P. This last matrix, A3, will take the average and difference of the remaining two entries and leave the detail coefficients unchanged.

$$A_3 = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Again, note the size of the identity matrix in the lower right. The larger size makes sense because there are more elements in the rows are to be left unchanged. Again carrying through with the example the point is illustrated.

(928, 1472, −288, −64, −64, −64, −64, 0)A3

= (1200, −272, −288, −64, −64, −64, −64, 0)

The Averaging and Differencing will be complete when the original matrix P is multiplied on the right by A1, A2, and A3. Repeat the process on the columns of the resulting matrix by multiplying on the left by AT1, AT2, and AT3. This process, although quicker than the original, still involves a lot of plugging and chugging. Here again linear algebra simplifies the mathematics.

By multiplying A1, A2, and A3 together, a new matrix W is created.

$$W = A_1 A_2 A_3 = \begin{bmatrix} 1/8 & 1/8 & 1/4 & 0 & 1/2 & 0 & 0 & 0 \\ 1/8 & 1/8 & 1/4 & 0 & -1/2 & 0 & 0 & 0 \\ 1/8 & 1/8 & -1/4 & 0 & 0 & 1/2 & 0 & 0 \\ 1/8 & 1/8 & -1/4 & 0 & 0 & -1/2 & 0 & 0 \\ 1/8 & -1/8 & 0 & 1/4 & 0 & 0 & 1/2 & 0 \\ 1/8 & -1/8 & 0 & 1/4 & 0 & 0 & -1/2 & 0 \\ 1/8 & -1/8 & 0 & -1/4 & 0 & 0 & 0 & 1/2 \\ 1/8 & -1/8 & 0 & -1/4 & 0 & 0 & 0 & -1/2 \end{bmatrix}$$

The matrix W will perform the same operations as A1, A2, and A3, but will greatly simplify this process. Similarly, the transpose of matrix W will be equal to the product of AT1, AT2, and AT3. So, by multiplying the original matrix P by W on the right and WT on the left the Averaging and Differencing process is completed and a new matrix T is created.

$$T = W^T P W \qquad \ldots\ldots(1)$$

Applying this process to matrix P produces the new transformed matrix T:

$$T = W^T P W = \begin{bmatrix} 1212 & -306 & -146 & -54 & -24 & -68 & -40 & 4 \\ 30 & 36 & -90 & -2 & 8 & -20 & 8 & -4 \\ -50 & -10 & -20 & -24 & 0 & 73 & -16 & -16 \\ 82 & 38 & -24 & 68 & 48 & -64 & 32 & 8 \\ 8 & 8 & -32 & 16 & -48 & -49 & -16 & 16 \\ 20 & 20 & -56 & -16 & -16 & 32 & -16 & -16 \\ -8 & 8 & -48 & 0 & -16 & -16 & -16 & -16 \\ 44 & 36 & 0 & 8 & 80 & -16 & -16 & 0 \end{bmatrix}$$

Notice that the top left entry represents an overall average, and the other entries are all detail coefficients.

3.3.2 IMPLEMENTING THRESHOLDS

Equation (1) creates a new matrix T. Using the following method matrix P is reconstructed from T.

$$
\begin{aligned}
T &= W^T P W \\
(W^T)^{-1} T W^{-1} &= (W^T)^{-1} W^T P W W^{-1} \\
(W^T)^{-1} T W^{-1} &= I P I
\end{aligned}
$$

This leads to the following reconstruction of matrix P.

$$(W^T)^{-1} T W^{-1} = P$$

Clearly equation (2) merely un-does the operations done by equation (1). However, this will not achieve the desired results. In lieu of using matrix T in equation (2), replace it with a close approximation matrix, N. This matrix N is constructed by implementing a threshold (replacing every element in T whose absolute value is less than or equal to a specified value with zero) on matrix T. Consider again, matrix T.

$$
T = \begin{bmatrix}
1212 & -306 & -146 & -54 & -24 & -68 & -40 & 4 \\
30 & 36 & -90 & -2 & 8 & -20 & 8 & -4 \\
-50 & -10 & -20 & -24 & 0 & 73 & -16 & -16 \\
82 & 38 & -24 & 68 & 48 & -64 & 32 & 8 \\
8 & 8 & -32 & 16 & -48 & -49 & -16 & 16 \\
20 & 20 & -56 & -16 & -16 & 32 & -16 & -16 \\
-8 & 8 & -48 & 0 & -16 & -16 & -16 & -16 \\
44 & 36 & 0 & 8 & 80 & -16 & -16 & 0
\end{bmatrix}
$$

Implement a threshold of 50 (let 0 replace every number in matrix T whose absolute value is less than or equal to 50)

$$
N = \begin{bmatrix}
1212 & -306 & -146 & -54 & 0 & -68 & 0 & 0 \\
0 & 0 & -90 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 72 & 0 & 0 \\
82 & 0 & 0 & 68 & 0 & -64 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -56 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 80 & 0 & 0 & 0
\end{bmatrix}
$$

3.4 STEPS IN DWT

DWT can be used to reduce the image size without losing much of the resolution. For a given image, you can compute the DWT of, say each row, and discard all values in the DWT that are less then a certain threshold. We then save only those DWT coefficients that are above the threshold for each row and when we need to reconstruct the original image, we simply pad each row, with as many zeros as the number of discarded coefficients, and use the inverse DWT to reconstruct each row of the original image. We can also analyze the image at different frequency bands, and reconstruct the original image by using only the coefficients that are of a particular band. The steps needed to compress an image are as follows:

1. Decompose the signal into a sequence of wavelet coefficients w.

2. Use threshold to modify the wavelet coefficients from w to another sequence w'.

3. Use Quantization to convert w' to a sequence q.

5. Apply entropy coding to compress q into a sequence e.

3.4.1 THRESHOLDING

In certain signals, many of the wavelet coefficients are close or equal to zero. Through a method called threshold, these coefficients may be modified so that the so sequence of wavelet coefficients contains long strings of zeros. Through a type of compression known as entropy coding these long strings may be stored and sent electronically in much less space. There are different types of threshold. In hard threshold, a tolerance is selected. Any wavelet whose absolute value falls below the tolerance is set to zero with the goal to introduce many zeros without losing a great amount of detail. There is not a straightforward easy way to choose the threshold. Although the larger the threshold that is chosen the more error that is introduced into the process. Another type of threshold is soft threshold. Once again a tolerance, h, is selected. If the absolute value of an entry is less than the tolerance, than that entry is set to zero. All other entries, d, are replaced with sign (d)□□d □- h□. Soft threshold can be thought of as a translation of the signal toward zero by the amount h. A third type of threshold is quartile threshold. In this method a percentage p of entries to be eliminated are selected. The smallest (in absolute value) p percent of entries are set to zero.

3.4.2 QUANTIZATION

The fourth step of the process, known as Quantization, converts a sequence of floating numbers w' to a sequence of integers q. The simplest form is to round to the nearest integer. Another option is to multiply each number in by a constant k, and then round to the nearest integer. Quantization is called Lossy because it introduces error into the process, since the conversion of w' to q is not a one-to-one function. In FT, the kernel function, allows us to obtain perfect frequency resolution. Because the kernel itself is a window of infinite length. If we use a window of infinite length, we get the FT, which gives perfect frequency resolution but no time information. Furthermore, in older to obtain the stationarity, we have to have a short enough window in which the signal is stationary. The narrower we make the window, the better the time resolution and better the assumption of stationarity but poorer the frequency resolution. The Wavelet transform (WT) solves the dilemma of resolution to a certain extent.

3.4.3 ENTROPY CODING

Wavelets and threshold help process the signal but up until this point, no compression has yet occurred. One method to compress the data is Huffman entropy coding. With this method, and integer sequence, q, is changed into a shorter sequence, e, with the numbers in e being 8 bit integers. An entropy-coding table makes the conversion. Strings of zeros are coded by the numbers I through 100, 105, and 106, while the non-zero integers in q are coded by 101 through 104 and 107 through 254. In Huffman entropy coding, the idea is to use two or three numbers for coding, with the first being a signal that a large number or long zero sequence is coming. Entropy coding is designed so that the numbers that are expected to appear the most often in q need the least amount of space in e.

3.5 SIMULATION

The algorithm for image compression using WT uses averaging and differencing to form the wavelet. Then we use the threshold technique to reduce the number of coefficients. Inverse transform is then applied to get the compressed mage.

3.5.1 ALGORITHM

1. W=s1*s2*s3 where s1, s2, s3 are obtained by using the averaging and differencing techniques
2. T=W'AW where W' is the transpose of the matrix W.
3. Now T is compressed to T*.We select a certain threshold value and all the coefficients below that particular value are neglected.
4.$(W^{-1})' T^{*} W^{-1}$=A*.
5. A* is a matrix approximate to the original matrix A.

3.6 RECONSTRUCTING AN IMAGE

As equation (2) shows, matrix P can be reconstructed very easily. If matrix N is substituted for matrix T a close approximation of matrix P will result. Thus:

$$(W^T)^{-1}NW^{-1} = R$$

The new approximation matrix R:

$$R = \begin{bmatrix} 670 & 670 & 1147 & 1137 & 1464 & 1464 & 1572 & 1572 \\ 670 & 670 & 1147 & 1137 & 1464 & 1464 & 1572 & 1572 \\ 614 & 614 & 1057 & 1339 & 1464 & 1464 & 1572 & 1572 \\ 726 & 726 & 945 & 1227 & 1464 & 1464 & 1572 & 1572 \\ 932 & 932 & 912 & 1176 & 1614 & 1614 & 1586 & 1586 \\ 932 & 932 & 912 & 1176 & 1614 & 1614 & 1586 & 1586 \\ 848 & 688 & 876 & 884 & 1314 & 1314 & 1558 & 1558 \\ 688 & 848 & 876 & 884 & 1314 & 1314 & 1558 & 1558 \end{bmatrix}$$

Although matrix R is an approximation of matrix P, the images are very similar. As mentioned previously, the differences between the reconstructed image and the original image are slight, and barely noticeable to a human eye. Keep in mind that these images are 8 × 8, a small portion of an actual image.

3.7 APPLYING THE HAAR WAVELET TRANSFORM TO FULL SIZE IMAGES

Now that the Haar Wavelet Transform is understood for 8×8 matrices, it's time to apply these ideas to full size images. This is done by first "normalizing" (multiplying by p2)

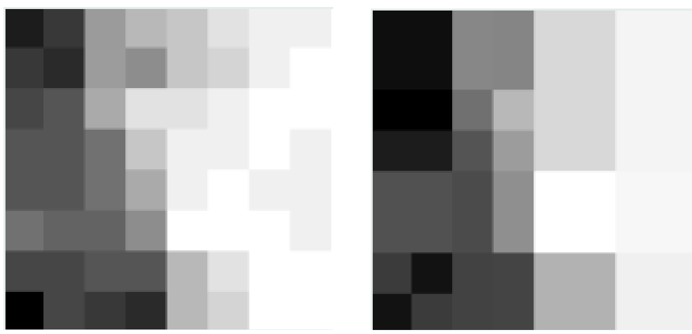

Fig 3.4 : Original image(P) and New image(R)

Original Image on Left represented by matrix P, New Image on Right represented by matrix R matrix A1, matrix A2, matrix A3, and matrix W. The result is quite interesting.

$$A_1 = \sqrt{2}A_1 = \begin{bmatrix} 1/\sqrt{2} & 0 & 0 & 0 & 1/\sqrt{2} & 0 & 0 & 0 \\ 1/\sqrt{2} & 0 & 0 & 0 & -1/\sqrt{2} & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & 0 & 0 & 0 & 1/\sqrt{2} & 0 & 0 \\ 0 & 1/\sqrt{2} & 0 & 0 & 0 & -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 0 & 0 & 0 & 1/\sqrt{2} & 0 \\ 0 & 0 & 1/\sqrt{2} & 0 & 0 & 0 & -1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 1/\sqrt{2} & 0 & 0 & 0 & 1/\sqrt{2} \\ 0 & 0 & 0 & 1/\sqrt{2} & 0 & 0 & 0 & -1/\sqrt{2} \end{bmatrix}$$

By normalizing matrix A1 a new matrix A1 is created. This new matrix has the property that its transpose acts as its inverse. This happens because the columns are orthogonal to one another. With denominators of p2 the multiplication of AT1 and A1, creates an identity matrix. Thus, it may be stated that

$$A^T = A^{-1}.$$

When matrix A2, matrix A3, and matrix W are normalized the same properties arise. Therefore,

$$W^T = W^{-1} \quad (4)$$

Now equation (2) can be simplified knowing that

$$(W^T)^{-1} T W^{-1} = P$$
$$(W^T)^T T W^T = P$$

This leads to the following result:

$$WTW^T = P \quad (5)$$

If a threshold is again implemented on matrix T, a new matrix N will again be constructed. Therefore equation (3) can also be re-written:

$$WNW^T = R \quad (6)$$

Matrix N still takes up less memory, and matrix R still is an approximation of matrix P.

In order to apply the new matrix W to a full size image it must be as large as the matrix it will be multiplied by. With linear algebra any matrix W is found by creating large matrices similar to A1 and following similar procedures to find A2, A3, A4, . . . , An, where the number n is determined by the size of the image. By multiplying these matrices together a new matrix W is created. The following 256 × 256 pixel images were generated using this procedure. Compare the compressed images to the original image. Pay attention to the change in quality as the threshold increases; when threshold is small–quality is retained, when threshold is large–quality suffers.

CHAPTER 4

EXPERIMENTAL RESULTS

WT Compression Result

DCT compression result

Performance comparison : DCT vs WT

4.1 WT COMPRESSION RESULT

The algorithm for image compression using WT uses averaging and differencing to form the wavelet. Then we use the threshold technique to reduce the number of coefficients. Inverse transform is then applied to get the compressed mage.

Fig 4.1 : Image compression using WT

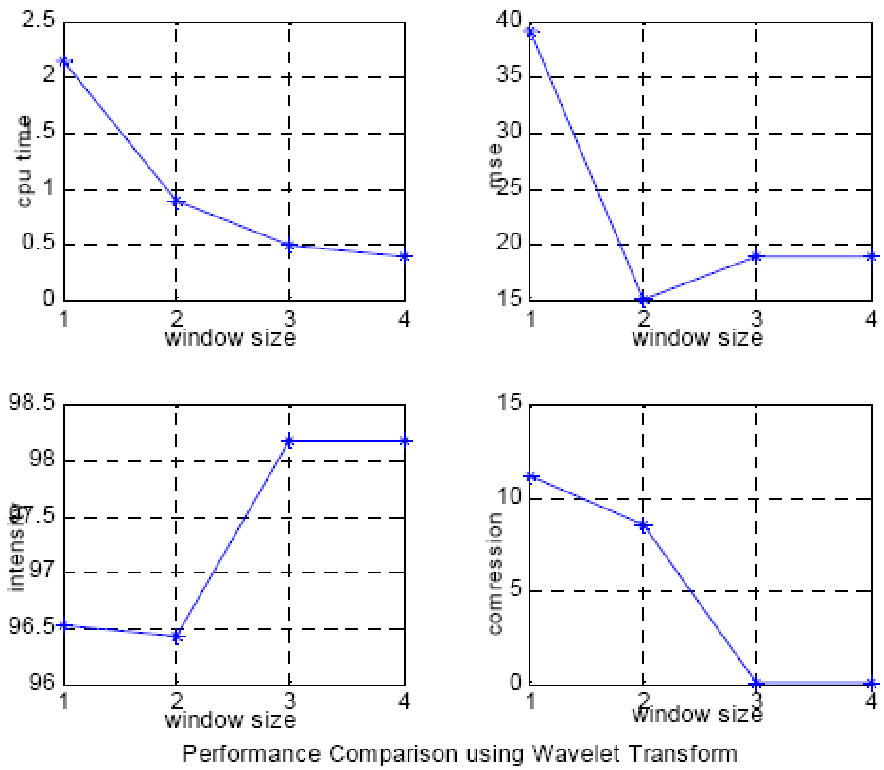

Fig 4.2 : The Intensity, CPU Time, Compression Ratio and Mean Square Error for WT

4.2 DCT COMPRESSION RESULT

Here we have taken the standard image LENA for our study purpose. We have subdivided the whole image into 3 x 3 sub images. The forward 2D-DCT-transformation is applied to all the pixels of each sub image. Next the pixels that carry least information eliminated. So the values of the pixels, which have values less than the threshold value, are set to zero. In our experiment we have chosen the threshold value equals to 20. So all the pixels having value less than 20 are assumed to be having value equals to zero. Then the inverse Discrete Cosine Transformation equation is applied to all the transformed pixels of the sub image. The same procedure is followed for all the sub images. It has been found that the energy retained by the compressed image is equal to 98.16%. The compression using Wavelet Transform gave a better performance than the 2D DCT. The image intensity was around 96.4%, the MSE is 12 dB. The time taken for the program execution was reduced to around 0.9. Also the compression was 8.5. The figure shows the performance comparision of 2D DCT image compression of CPU time, MSE, intensity, and compression for different window size.

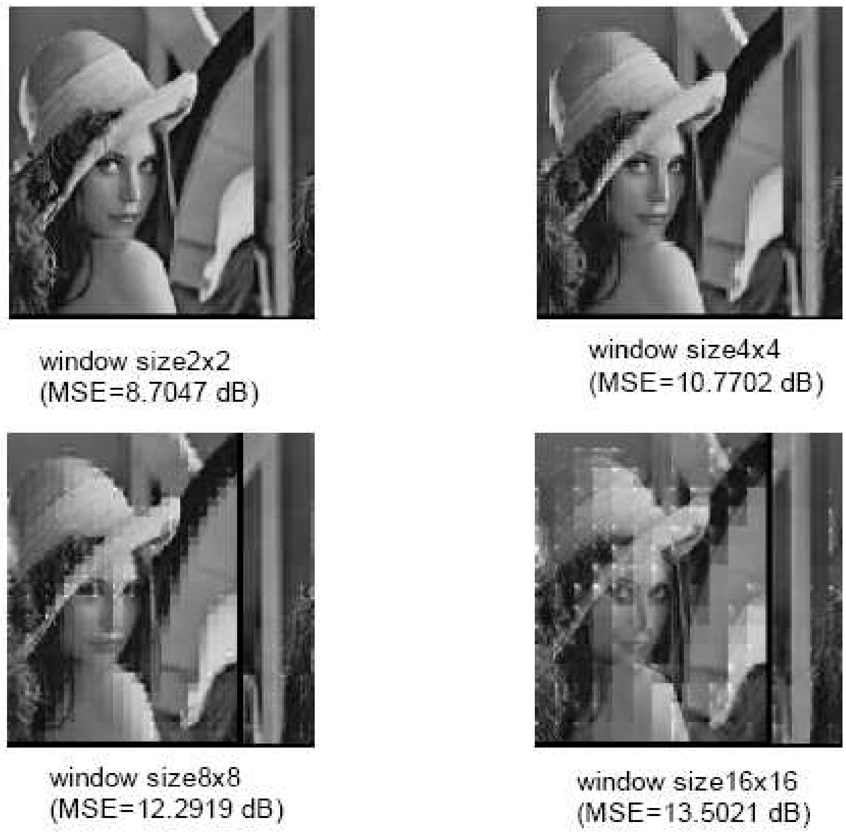

Fig 4.3 : Image compression using DCT

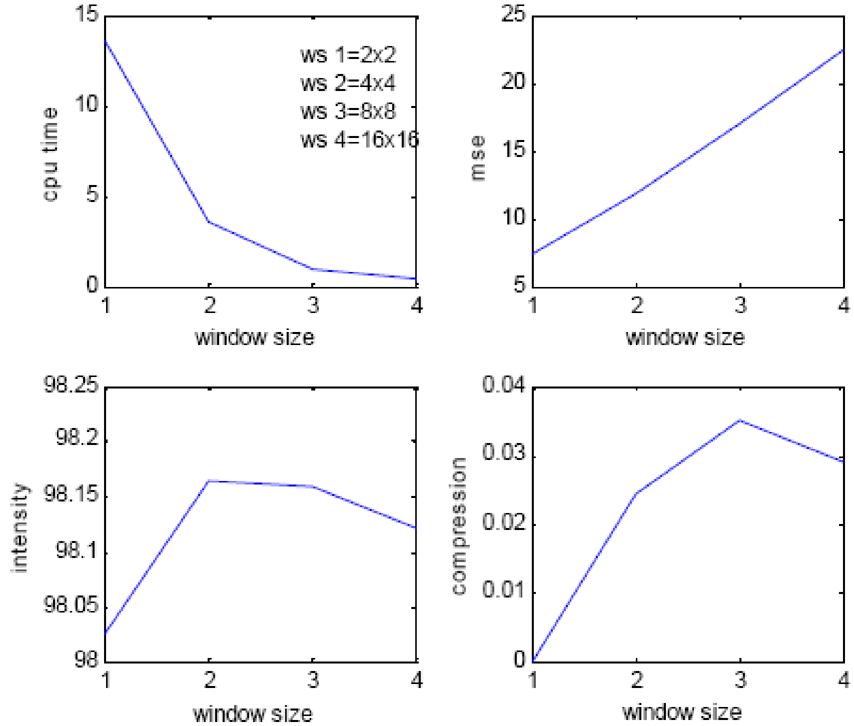

Fig 4.4 : The intensity, CPU Time, Compression Ratio and Mean Square Error for DCT

4.3 PERFORMANCE COMPARISON : DCT VS WT

COMPARISON OF WAVELET TRANSFORM AND 2D DCT

Property	2D DCT	Wavelet transform
Image Intensity	98.16%	96.4%
MSE in dB	8	12
CPU time/Exec. Time	3.8	0.9
Compression	0.025	8.5

Table 4.1 : Result comparison for window size (4 x 4)

CONCLUSION

Even if Discrete Cosine Transform is a widely adapted and robust method used for compression of digital image as it has the ability to carry the most of the information in smallest number of pixels compared to other method, the Wavelet based Transform provided better result as far as properties like RMS error, image intensity and execution time is concerned. So Wavelet based Transform is widely used.

REFERENCES

1. Proakis John G, Manolokis Dimitris G, “Digital Signal Processing Principles Algorithm and Applications”, San Diego, Prentice-Hall,1996

2. Jain Anil K., “Fundamentals of Digital Image Processing, Englewood Cliffs”, NJ, Prentice Hall, 1989, p. 439

3. Gonzalez Rafel C, Woods Richard E., “Digital Image Processing” , Addison Wesley

4. Gabor D. "Theory of Communications", J.I.E.E.E.,. Vol. 93, (1946), p. 429-459

5. Oppenheim A. V. and Schafer R. W., “Discrete Time Signal Processing”, New Delhi: PHI, India

6. Averbuch A., Lazar Danny and Israeli Moshe, “Image Compression using WT and Multiresolution Decomposition”, IEEE Trans. On Image Processing., Vol. 5, No. 1 (1996) , Jan 31

7. Baliarsingh R. and Jena G., “Gabor Function: An Efficient Tool for Digital Image Processing,” Intl. Conf., SRKR Engg college (JNTU), vol. 1, (Oct 2005) p 98-101,