2002

Image compression using wavelet transforms

Ramanjit K. Sahi
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Sahi, Ramanjit K., "Image compression using wavelet transforms" (2002). *Master's Theses*. 2375.
DOI: <https://doi.org/10.31979/etd.pu6h-udtx>
https://scholarworks.sjsu.edu/etd_theses/2375

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

**ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, Mi 48106-1346 USA
800-521-0600**

UMI[®]

IMAGE COMPRESSION USING WAVELET TRANSFORMS

A Thesis

Presented to

The Faculty of the Department of Mathematics

San Jose State University

In Partial Fulfillment

of the Requirement for the Degree

Master of Science

By

Ramanjit K. Sahi

December 2002

UMI Number: 1411628

UMI[®]

UMI Microform 1411628

**Copyright 2003 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.**

**ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346**

© 2002

Ramanjit K. Sahi

ALL RIGHTS RESERVED

APPROVED FOR THE DEPARTMENT OF MATHEMATICS

Mohammad Saleem

Dr. Mohammad Saleem

R. Dodd.

Dr. Roger Dodd

A. Singh

Dr. Avtar Singh

APPROVED FOR THE UNIVERSITY

Sanjiv Datta

Abstract

Image Compression Using Wavelet Transforms

by Ramanjit K. Sahi

Image Data Compression is a science that is concerned with the reduction of the number of bits required to store, transmit, and reconstruct images without any loss of information. Though various transforms are used in the field of image compression, wavelets are the most popular, and are increasingly being applied in the field of digital images. Wavelets are a way to analyze a signal using base functions, which are localized both in time and in frequency. In this thesis, wavelet transforms are applied on a lossy image compression. Further, study of the reconstructed image is done using scalar quantization techniques. Experimental results show that the uniform quantization using 8 bits per pixel gives the best-reconstructed image, even better than the adaptive quantization.

Acknowledgements

I wish to sincerely thank Professor Mohammad Saleem for his guidance, support, and constant encouragement in the completion of this thesis. I also thank Professor Roger Dodd and Professor Avtar Singh for providing me with useful information and assistance in understanding the concept needed in the process.

I thank my parents for their constant love and encouragement. Special thanks to my brother and sister for their encouragement, whenever I seemed to lose it.

Table of Contents

1. Introduction	1
2. Image Compression	4
2.1 Compression Techniques	4
2.2 Coding	5
2.3 Transform Compression Framework	6
3. Signal Processing	9
3.1 Definitions	9
3.2 Transform Domain Analysis	11
3.3 Discrete – Time Systems	12
3.4 Sampling Theorem	13
3.5 Convolution	13
3.6 Frequency Response	14
3.7 Digital Filters: FIR and IIR	14
3.8 Decimator and Interpolator	14
4. Multiresolution Concept and Analysis	16
4.1 Axiomatic Definition	16
5. Filter Banks	20
5.1 Analysis Bank	20
5.2 Synthesis Bank	24
6. Wavelet Theory	26
6.1 Wavelet System Characteristics	26
6.2 Relevance To Image Compression	27
6.3 Haar Wavelet Transform	28
6.4 Effectiveness of Wavelet Analysis	29
7. Example Of Image Compression	30
7.1 Outline	30
7.2 Analysis	37
8. Conclusion and Future Directions	39
9. References	40
10. Appendix	42
10.1 MatLab Programs	42

List of Tables

Table 7.1	Uniform Quantization at different bits per pixel	38
Table 7.2	Adaptive Quantization (8,6,4,0) bpp	38

List of Figures

Figure 2.1	Block Diagram of Transform	6
Figure 3.1	Unit Sample Sequence	10
Figure 3.2	Unit Step Sequence	10
Figure 3.3	Exponential Function	11
Figure 3.4	Sinusoidal Sequences	11
Figure 3.5	Discrete Time System	13
Figure 3.6	Down Sampler or Decimator	15
Figure 3.7	Up Sampler or Interpolator	15
Figure 4.1	Nesting of Scaling function spaces V_j and Wavelet function spaces W_j	18
Figure 5.1	Filter Bank	20
Figure 5.2 a)	Finer Scale	20
Figure 5.2 b)	Coarser Scale	21
Figure 5.3	Two – Band Analysis Bank	23
Figure 5.4	Two Level Two Band Analysis Tree	23
Figure 5.5	Two – Band Synthesis Bank	25
Figure 6.1 a)	Scaling Function	28
Figure 6.1 b)	Wavelet Function	28
Figure 7.1	Original Lena Image	30
Figure 7.2	Image after one Haar Wavelet Transform	31
Figure 7.3	Image at 8 bpp	32
Figure 7.4	Image at 4 bpp	32
Figure 7.5	Image at 2 bpp	32
Figure 7.6	Image at .5 bpp	32
Figure 7.7	Adaptive Quantization (8, 6, 4, 0) bpp	33
Figure 7.8	Reconstructed Lena Image at 8 bpp	34
Figure 7.9	Reconstructed Lena Image at 4 bpp	34
Figure 7.10	Reconstructed Lena Image at 2 bpp	35
Figure 7.11	Reconstructed Lena Image at .5 bpp	35
Figure 7.12	Reconstructed Image after Adaptive Quantization (8, 6, 4, 0) bpp	36

Chapter 1 Introduction

Uncompressed forms of digital images require huge amount of data storage space and transmission bandwidth. Despite the rapid progress in data storage capacity, processor speed, and digital communication system performance, storage and transmission of images still pose a considerable challenge. Various image compression techniques are employed to store or transmit image files so as to save storage space without any appreciable loss of information. This compression can be lossy or lossless depending on the way the reconstruction is done. This thesis deals with image compression of gray-scale or monochrome images using wavelet transforms.

Wavelet is a versatile tool with very rich mathematical content and great potential for applications [2]. Though various transforms are used in the field of image compression, wavelets are the most popular and are increasingly being used in the field of digital images. Here, wavelet transforms are applied on a lossy image for compression. Wavelets are functions whose dilations and translations form a basis of the space of all functions with finite energy. The features of the natural images are captured well by the wavelet basis functions, as wavelet coefficients have good frequency resolution at lower frequency to capture image backgrounds as well as good time resolution at higher frequencies to capture image edges.

As compared to the Fourier based algorithms, Wavelet based algorithms provide substantial improvement in picture quality at higher compression ratios [3]. It can be seen in the way wavelets based compression techniques are fast replacing the traditional

Fourier based transforms in the form of Discrete Cosine Transform (DCT) in the field of image compression. Also, rapid advancement in wavelet technology has led to the JPEG – 2000 standard for the still image compression, which is based on Discrete Wavelet Transform (DWT) using Daubechies (9,7) biorthogonal wavelet. It has brought a new surge of interest in wavelets and also towards advancement and faster computational algorithms for storing and transmission of images. In the second stage of image compression, the transformed wavelet coefficients will be quantized. This is where the real loss of information occurs. Then the encoding of the quantized coefficients is done and after that the data is stored or transmitted. To reconstruct the image, decoding is done by first inverting the encoded sequence, then inverting the quantized coefficients and then applying the inverse wavelet transform.

In this thesis, after applying Haar Wavelet Transform (HWT), study of Scalar (Uniform and Adaptive quantization) has been done and then the quantized coefficients are inverted and inverse HWT is applied to reconstruct the image. The compression algorithm will be evaluated on how closely the reconstructed image resembles the original, and how the performance measures, in terms of peak signal to-noise-ratio (PSNR) and mean-square-error (MSE).

Chapter 1 introduces the various compression techniques used in image compression.

In Chapter 2, a brief review of various compression techniques used in image compression are discussed. Also the basic structure of the wavelet coder is studied.

Chapter 3 deals with Signal Processing. In this chapter, basic sequences, various transforms, discrete time systems, sampling theorem, FIR and IIR filters, decimator and interpolator will be defined.

How Multiresolution Concept and Analysis are related to wavelet transforms is discussed in Chapter 4.

Definition of Filter Banks and its two types, Analysis and Synthesis filters, are studied in detail in Chapter 5.

In Chapter 6, Wavelet Transform, and its features are studied. Its relevance to image compression is studied and also Haar Wavelets are discussed.

Chapter 7 deals with an example of Image compression. The standard picture used in digital signal processing i.e., the Lena image is taken. On the Lena image, wavelet transform is applied and then the image is quantized. After this an analysis is done using PSNR and MSE to find which quantization is more apt.

Future directions are discussed in Chapter 8.

Chapter 2 Image Compression

One of the main applications of digital signal processing is the compression of signals. The signals can be two-dimensional (images) or one-dimensional. Image compression refers to the reduction of the digital data to store or transmit images with less rate of distortion.

2.1 Compression Techniques

There are two ways of classifying compression techniques in the field of image compression.

2.1.1 Lossless Compression

In lossless compression, the original data can be recovered exactly from the compressed data. It is used generally for discrete data such as text, computer generated data, and certain kinds of image and video information. Lossless compression can achieve only a modest amount of compression of the data and hence it is not useful for sufficiently high compression ratios. GIF is a popular example of a lossless compression.

2.1.2 Lossy Compression

Lossy compression techniques refer to the loss of information when data is compressed. As a result of this distortion, much higher compression ratios are possible as compared to the lossless compression in reconstruction of the image. JPEG is popular example of a lossy compression technique. Lossy compression techniques are more suitable for images, as much of the detail in an image can be discarded without greatly

changing the appearance of the image. In practice, very fine details are lost in image compression, though image size is drastically reduced.

2.2 Coding

Various coding techniques are used to facilitate compression. Here pixel coding, predictive and transform coding will be discussed.

2.2.1 Pixel Coding

Each pixel in the image is coded separately. The pixel values that occurs most frequently are assigned shorter code words (i.e., fewer bits), and those pixel values that are more rare are assigned longer codes. That makes the average code word length decrease.

2.2.2 Predictive Coding

As images are highly correlated from sample to sample, predictive coding technique is relatively simple to implement. Predictive coding predicts the present values of the sample based on the past values and only encodes and transmits the difference between the predicted and the sample value. Differential Pulse Code Modulation (DPCM) is an example of a frequently used predictive coding.

2.2.3 Transform Coding

In transform coding, an image is transformed from one domain (usually spatial or temporal) to a different type of representation, using some well-known transform. Then the transformed values are coded and thus provide greater data compression. In this thesis, transforms are orthogonal so that the mapping is therefore unique and reversible. As a result, the energy is preserved in the transform domain: that is, the sum of the

squares of the transformed sequence is the same as the sum of the squares of the original sequence [9]. Thus, the image can be completely recovered by the inverse transform.

2.3 Transform Compression Framework

Transform refers to changing the coordinate basis of the original signal, such that the new signal has the whole information in few transformed coefficients. The processing of the signals in the transform domain is more efficient as the transformed coefficients are not correlated. A popular image compression framework is the Transform Based Image Compression as shown in Figure 2.1. The first step in the encoder is to apply a linear transform to remove redundancy in the data, followed by quantizing the transform coefficients, and finally entropy coding the quantized outputs. After the encoded input image is transmitted over the channel, the decoder undoes all the functionalities applied in the encoder and tries to reconstruct a decoded image as close as to the original image.

Figure 2.1 Block diagrams of Transform – based image coder: (a) Compression or Encoder and (b) Decompression or Decoder diagrams

2.3.1 Linear Transform

The first step in the compression or encoding is to transform the image from the spatial domain to the transformed domain using some known transforms like: Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Fractal Transforms, and many more. Compression of the original image is not easy, as the energy is scattered throughout the image. By transforming, a large portion of the energy can be concentrated in the low frequency-zone of the transform domain.

For the conservation of the energy from the spatial domain to the transformed domain, it is desirable for the transform to be orthogonal. Here, wavelet transforms are implemented due to its superior energy compaction and correspondence with human visual system.

2.3.2 Quantizer

Quantizer is a key component in the transform compression framework that introduces non-linearity in the system. It maps the transformed digital image to a discrete set or a discrete number. It is a lossy compression technique as it introduces an error in the image compression process i.e., it converts sequence of floating point numbers to sequence of integers. Scalar Quantization is the simplest quantization, as each input is treated separately in producing the output. In vector quantization, the input samples are clubbed together in groups called vectors, and then processed to give the output. Also, the quantization of the difference between the predicted value and the past samples is

called predictive quantization. A good quantizer is one, which represents the original signal with minimum loss or distortion.

2.3.3 Entropy Coding

After the quantization of the transformed values, entropy encoder further compresses the quantized values to give additional compression. It is a lossless compression and is also reversible. The idea is to find a reversible mapping to the quantized values such that the average number of bits or symbols is minimized. For example, the Morse code assigns short codes to the letters appearing frequently whereas long codes are assigned to the letters appearing less frequently. Entropy coding works exactly like the Morse code. The two popular entropy-coding methods are Huffman coding and arithmetic coding. This will be discussed later in chapter 8.

Chapter 3 Signal Processing

A signal is a function that conveys information, generally about the state or behavior of a physical system. Mathematically, the signals are represented as functions of one or more independent variables. In this thesis, an image is represented, as a brightness function of two spatial variables. The independent variable in the representation of the signal can be either continuous or discrete. Continuous-time signals are also referred as an analog signal. Discrete-time signals have discrete set of values and are represented mathematically as a sequence of numbers. Digital Signal Processing is the study of discrete-time systems that process a digital signal in order to modify the original signal as given in the sequence. Here, basic concepts of the digital system processing will be discussed.

3.1 Definitions

All the signals except the wavelet transforms are in a one-dimensional space.

3.1.1 Basic Sequences

A signal is a sequence of numbers that could be finite or infinite. The signals look like $x = (\dots, x_{-1}, x_0, x_1, x_2, \dots)$ or $x = \{x_n, n \in \mathbb{Z}\}$

The important sequences in the digital signal processing are

3.1.2 Unit Sample Sequence

It is defined as $\delta(n) = (\dots, 0, 0, 0, 1, 0, 0, \dots)$ or $\delta(n) = \begin{cases} 0, & n \neq 0 \\ 1, & n = 0 \end{cases}$

It is also called discrete-time impulse or impulse. Its analogous in the continuous-time domain is the “delta function” also called the Dirac delta function. The graphical representation is given in Figure 3.1

Figure 3.1 Unit Sample Sequence

3.1.3 Unit Step Sequence

It is defined as $u(n) = (\dots\dots 0,0,0,1,1,1,1\dots\dots\dots)$ or

$$u(n) = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

Figure 3.2 Unit Step Sequence

3.1.4 Exponential Sequence

The general form is of type $x(n) = A y^n$. Its graph is given in Figure 3.3

Figure 3.3 Exponential function

3.1.5 Sinusoidal Sequence

It is of the form $x(n) = A \cos(\omega n + \alpha)$ for all n with A real

Figure 3.4 Sinusoidal Sequences

3.2 Transform Domain Analysis

3.2.1 Z – Transform

The z – transform of a sequence $x(n)$ is defined as

$$X(z) = \sum_n x(n) z^{-n}, \quad -\infty < n < \infty$$

It is an infinite sum or infinite power series with z being a complex variable. It is also defined as the two-sided or bilateral z -transform

3.2.2 Fourier Transform

The Discrete Fourier transform (DFT) transforms a time domain sequence $x(n)$ to a frequency domain sequence $X(k)$. It is given by the equation

$$X(k) = \sum_n x(n) e^{-j2\pi nk/N}, \quad 0 \leq n < N \text{ and } 0 \leq k < N-1$$

There is a close relationship between the Discrete Fourier Transform and the Frequency Response of a sequence. The relationship is given by

$$X(k) = X(e^{j\theta}) \big|_{\theta=2\pi nk/N}, \quad 0 \leq k < N-1$$

3.2.3 Wavelet Transform

The continuous wavelet transform of a function $f(t) \in L^2(\mathbb{R})$ is defined as

$$f(t) = \int_{-\infty}^{\infty} \psi_{j,k}(t) f(t) dt, \quad \text{where } \psi_{j,k}(t) = (1/\sqrt{j})\psi((t-k)/j) \text{ for } j>0$$

The function $f(t)$ can be recovered from the transform by the reconstruction formula.

In this thesis, DWT have been taken into consideration. Thus, any function $g(t) \in L^2(\mathbb{R})$ can be written as

$$g(t) = \sum_k c(k) \phi_k(t) + \sum_j \sum_k d(j,k) \psi_{j,k}(t)$$

where $-\infty < k < \infty$ and $0 < j < \infty$

The coefficients $d(j,k)$ are the DWT coefficients and $\psi_{j,k}(t)$ give the discrete wavelet series.

3.3 Discrete-Time Systems

A discrete-time system is defined mathematically as a transformation that operates on an input sequence $x(n)$ to produce an output sequence $y(n)$. This is given by $y(n) = T\{x(n)\}$ and is shown in Figure 3.5

Figure 3.5 Discrete Time System

In this thesis, a Linear time invariant systems (LTI) are taken. LTI is a system, which is not only linear, but is also time invariant.

3.3.1 Linear System

A system is linear if for all signals $x(n)$ and $y(n)$, scalars a and b

$$T(ax + by) = aT(x) + bT(y)$$

3.3.2 Time Invariant System

A time invariant system is one for which a delay of the input sequence causes a corresponding delay in the output sequence.

3.4 Sampling Theorem

If the sampling frequency is at least twice as large as the highest frequency component in the signal then a signal can be represented unambiguously by its samples. Further the signal can be reconstructed from its samples by digital-to-analog conversion and then doing low pass filtering.

3.5 Convolution

Convolution refers to the output of an LTI system when its unit sample response is multiplied by the input sequence $x(n)$. It is given by the equation

$$y(n) = \sum_m h(n)x(n-m) = \sum_m x(n)h(n-m) = h(n) * x(n), \quad -\infty < n < \infty$$

where $x(n)$ represents the input, $y(n)$, the output, and $h(n)$, the unit sample response of the

system.

3.6 Frequency Response

Frequency response of a sequence $x(n)$ is given by the equation

$$X(e^{j\omega}) = \sum_n x(n)e^{-jn\omega},$$

where, ω is the digital frequency, which ranges from 0 to 2π radians corresponding to analog frequency from 0 to f_s in Hz. Frequency response is a complex continuous function of ω and provides both the magnitude and the phase response.

3.7 Digital Filters: FIR and IIR

A digital filter is a linear time operator $h(n)$ that operates on an input sequence $x(n)$ to generate a filtered output sequence $y(n)$. A general filter difference equation is given as

$$y(n) = \sum_k a_k y(n-k) + \sum_k b_k x(n-k), \quad -\infty < n < \infty$$

3.7.1 Finite Impulse Response Filter (FIR)

It is a filter whose unit impulse response $h(n)$ has finitely many non-zeros. It is a stable filter as there is no feedback. It is given as

$$y(n) = \sum_k h(k)x(n-k), \quad 0 < k < N$$

3.7.2 Infinite Impulse Response Filter (IIR)

It is a filter whose unit sample response $h(n)$ has infinite many non zeros. It is given as

$$y(n) = \sum_m a_m y(n-m) + \sum_k b_k x(n-k), \quad 0 < k < N, \quad 1 < m < L$$

3.8 Decimator and Interpolator

In DSP, decimator and interpolator are two operations concerned with the sampling rate. Decimator or down-sampler lowers the sampling rate where as

interpolator or the up-sampler raises the sampling rate. The down sampler takes a signal $x(n)$ and down samples by a factor to produce the desired output. In this thesis, down-sampling is done by a factor of two. It is shown in Figure 3.6

Figure 3.6 Down Sampler or the Decimator

Similarly, the up-sampler takes a signal and up samples by a factor of two so as to increase the number of samples. It means to insert zeros between the terms of the original sequence. The input sequence is stretched to twice its original length and zeros are inserted. It can be seen in Figure 3.7

Figure 3.7 Up Sampler or the Interpolator

Chapter 4 Multiresolution Concept and Analysis

The multiresolution concept is designed to represent signals, where a single event will be decomposed into finer and finer details. A signal is represented by a coarse approximation and finer details. The coarse and the detail subspaces are orthogonal to each other. By applying successive approximation recursively the space of the input signals $L^2(\mathbb{R})$ can be spanned by spaces of successive details at all resolutions. Mallat and Meyer pioneered this multiresolution concept, which form the basis of a mathematical framework for wavelets.

4.1 Axiomatic definition of Multiresolution Analysis

A multiresolution analysis consists of sequence of embedded closed subspaces

$$\dots V_{-3} \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \dots$$

such that

$$V_{-c} = \{0\}, \quad V_c = L^2$$

A high-resolution signal space will also contain a lower resolution signal space. Since all the spaces are scaled versions of the elements in the next space, thus

$$f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}$$

Thus, to use the idea of the multiresolution, first scaling function will be defined and from it the wavelets will be defined [1].

4.1.1 Scaling Function

A set of scaling functions are defined in terms of integer translates of the basic scaling function by the equation,

$$\phi_k(t) = \phi(t - k), \quad k \in \mathbb{Z}, \quad \phi \in L^2$$

The subspace of $L^2(\mathbb{R})$ (it is the space of all square-integrable functions) spanned by the above functions is given as

$$V_0 = \text{closure} (\text{Span}_k \{ \phi_k(t) \})$$

A two-dimensional family of functions is generated from the basic scaling function by scaling and shifting. It is given by

$$\phi_{j,k}(t) = 2^{j/2} \phi(2^j t - k)$$

whose span over k is given by

$$V_j = \text{closure} (\text{Span}_k \{ \phi_k(2^j t) \}) = \text{closure} (\text{Span}_k \{ \phi_{j,k}(t) \})$$

for all integers $k \in \mathbb{Z}$.

If $j > 0$, $\phi_{j,k}$ is narrower and is translated in smaller steps and hence represents finer details. On the other hand if $j < 0$, then $\phi_{j,k}$ is wider and is translated in larger steps, and thus represents coarser details.

If $\phi_k(t)$ is in V then by the definition of the multiresolution $\phi_k(2t)$ is also in V . This means that $\phi(t)$ can be expressed as

$$\phi(t) = \sum_n h(n) \sqrt{2} \phi(2t - n), \quad n \in \mathbb{Z}$$

where the coefficients $h(n)$ are called the scaling coefficients or the scaling filters and the $\sqrt{2}$ maintains the norm of the scaling function with the scale of two. $\phi(t)$ is expressed as a weighted sum of shifted $\phi(2t)$. This equation is called the refinement equation, the multiresolution analysis equation (MRA), or the dilation equation.

4.1.2 Wavelet Functions

The wavelet functions given by $\psi_{j,k}(t)$ spans the difference between the spaces spanned by the various scales of the scaling function. Scaling and Wavelet functions are orthogonal to each other. The orthogonal basis functions have a Parseval's theorem that allows a partitioning of the signal energy in the wavelet transform domain. W_j is the orthogonal component of V_j in V_{j+1} . The wavelet subspace spanned by W_0 is such that

$$V_1 = V_0 \oplus W_0$$

In general it is given as

$$L = V_0 \oplus W_0 \oplus W_1 \oplus \dots$$

where V_0 is the initial space spanned by the scaling function $\phi(t - k)$.

Figure 4.1 Nesting of Scaling function spaces V_j and Wavelet function spaces $W_j[l]$

The wavelet equation is given by

$$\psi(t) = \sum_n h_l(n) 2^{l/2} \phi(2^l t - n), n \in \mathbb{Z}$$

where $h_l(n)$ are some set of coefficients. In DSP, they are referenced as high pass filter coefficients. The wavelet coefficients are related to the scaling functions coefficients by [1]

$$h_l(n) = (-1)^n h_l(1-n)$$

Thus the wavelets and the scaling function span all of $L^2(\mathbb{R})$. Hence, any function $f(t)$ in $L^2(\mathbb{R})$ can be written as

$$f(t) = \sum_k c(k) \phi_k(t) + \sum_j \sum_k d(j, k) \psi_{j,k}(t)$$

The first summation gives the coarse approximation of the function $f(t)$ and the second summation adds the details with each increasing index j . These expansion functions form an orthonormal basis and hence the coefficients can be calculated by inner product as

$$c(k) = \langle f(t), \phi_k(t) \rangle = \int f(t) \phi_k(t) dt \quad \text{and}$$

$$d_j(k) = d(j, k) = \langle f(t), \psi_{j,k}(t) \rangle = \int f(t) \psi_{j,k}(t) dt$$

Chapter 5 Filter Banks

Filter bank is a set of filters. It consists of an Analysis bank and a Synthesis bank.

In this thesis, a two-channel filter bank is considered. It is shown in Figure 5.1

Figure 5.1 Filter Bank

5.1 Analysis Bank

The analysis bank has two filters: a low pass and a high pass. They separate the input signal into frequency bands. When the input signal (here it is the image) is first passed through the analysis filters, it decomposes the signal into 4 bands. They are given as LL, HL, LH, and HH. This is the first level of the decomposition and it represents the finer scale of the expansion coefficients. It is shown in Figure 5.2(a) and Figure 5.2(b).

Figure 5.2 (a) Finer Scale

Figure 5.2 (b) Coarser Scale

First a relationship between the wavelet and the scaling expansion coefficients is developed at a lower scale in terms of those at the higher scale [1]

The dilation equation is given by

$$\phi(t) = \sum_n h(n) \sqrt{2} \phi(2t - n), \quad n \in \mathbb{Z}$$

Then, scale and translate the time variable assuming that a unique solution exists

$$\phi(2^j t - k) = \sum_n h(n) \sqrt{2} \phi(2(2^j t - k) - n) = \sum_n h(n) \sqrt{2} \phi(2^{j+1} t - 2k - n)$$

Let $m = 2t + n$

$$\phi(2^j t - k) = \sum_m h(m - 2k) \sqrt{2} \phi(2^{j+1} t - m) \quad (5.1)$$

Let $V_j = \text{Span}_k \{2^{j/2} \phi(2^j t - k)\}$ and using the multiresolution property

$$f(t) \in V_{j+1} \Rightarrow f(t) = \sum_k c_{j+1}(k) 2^{(j+1)/2} \phi(2^{j+1} t - k)$$

Clearly, a signal is expressed at a scale of $j+1$. Though there are no wavelets. In order to see the details of the image, the signal has to be written at a lower resolution. At this lower resolution both the scaling and the wavelet functions are seen. Thus,

$$f(t) = \sum_k c_j(k) 2^{j/2} \phi(2^j t - k) + \sum_k d_j(k) 2^{j/2} \psi(2^j t - k)$$

where the $2^{j/2}$ term maintains the unity norm of the basis functions at various scales.

In this thesis, HWT are taken and they are orthonormal. Thus $\phi_{j,k}$ and $\psi_{j,k}$ are orthonormal, hence the j level coefficients are found by taking the inner product

$$c_j(k) = \langle f(t), \phi_{j,k}(t) \rangle = \int f(t) 2^{j/2} \phi(2^j t - k) dt$$

which by using (5.1) and interchanging the sum and integral, can be written as

$$c_j(k) = \sum_m h(m - 2k) \int f(t) 2^{(j+1)/2} \phi(2^{j+1} t - m) dt$$

$$\text{where } c_j(k) = \sum_m h(m - 2k) c_{j+1}(m) \quad (5.2)$$

and the corresponding the wavelet coefficient is given by

$$d_j(k) = \sum_m h_1(m - 2k) c_{j+1}(m) \quad (5.3)$$

From, the signal processing point of view, both (5.2) and (5.3) can be viewed as digital filtering and down-sampling that is the scale $-j$ coefficients are filtered by two FIR filters (low pass and a high pass) with coefficients $h(-n)$ and $h_1(-n)$ and then down-sampled gives the next coarser scaling and wavelet coefficients. The implementation of the equations (5.2) and (5.3) can be seen in Figure 5.3 where the down pointing arrow denotes a down-sampling by 2.

Figure 5.3 Two- Band Analysis Bank [1]

The two filters double the number of coefficients but the decimator halves it. It implies that there is a possibility of getting the original signal back. The aliasing occurring in the higher scale can be cancelled by using the signal from the lower level. This is the idea behind the perfect reconstruction filter bank. Repeating the splitting, filtering, and decimation on the scaling coefficients is called Iterating the filter bank. It is illustrated in Figure 5.4 [1]

Figure 5.4 Two Level Two Band Analysis Tree [1]

The first scale in the two-band analysis tree divides the spectrum of c_{j+1} into a lowpass and highpass bands which are shown by the scaling and the wavelet coefficient respectively at a lower scale $c_j(k)$ and $d_j(k)$. In the first level the spectrum is divided

into two equal parts. The second level again divides the lowpass band into two bands. Thus, the lower half is divided into quarters and so on. It results in a logarithmic set of bandwidths which are called “ Constant – Q ” filters, as the ratio of the bandwidth to the center frequency of the band is constant. Mostly the signal is band limited. This splitting, filtering continues till a upper scale say $j = J$ is reached, as above it the wavelet coefficients are negligibly small. Thus for $f(t) \in V_j$

$$f(t) = \sum_k c_j(k) \phi_{j,k}(t)$$

$$f(t) = \sum_k c_{j-1}(k) \phi_{j-1,k}(t) + \sum_k d_{j-1}(k) \psi_{j-1,k}(t)$$

5.2 Synthesis Bank

It is the inverse of the analysis bank. In digital signal processing there was decimation and filtering in the analysis bank, similarly there are two steps upsampling and filtering in the synthesis bank. The reconstruction of the original fine scale coefficients can be made from the combination of the scaling and the wavelet coefficients at a (lower) coarser scale.

Consider the signal $f(t) \in V_{j+1}$, it can be written in terms of the scaling coefficients as

$$f(t) = \sum_k c_{j+1}(k) 2^{(j+1)/2} \phi(2^{j+1}t - k)$$

where it can be written in terms of the coefficients at a lower scale as

$$f(t) = \sum_k c_j(k) 2^{j/2} \phi(2^j t - k) + \sum_k d_j(k) 2^{j/2} \psi(2^j t - k)$$

where c_{j+1} , c_j and d_j are related as [1]

$$c_{j+1} = \sum_m c_j(m) h(k - 2m) + \sum_m d_j(m) h_1(k - 2m)$$

The above equation is evaluated by upsampling the j scale coefficients given by $c_j(k)$, and then convolving them with $h(n)$. The same is done to the j level wavelet coefficients and the result is added to give the $j+1$ level coefficients. It is illustrated in Figure 5.5

Figure 5.5 Two-Band Synthesis Bank

This set of analysis and synthesis filters is known as the Mallat's Algorithm. Filter bank trees are mostly of two types: the full-grown tree and the Octave-band tree. In the full-grown tree every branch is further subdivided whereas in the octave-band tree only the lower branch is further subdivided. In this thesis, Octave-band tree decomposition has been taken into consideration.

Chapter 6 Wavelet Theory

Wavelets are functions defined over a finite interval having an average value of zero. The basic idea of wavelet transform is to represent any arbitrary function $f(t)$ as a superposition of a set of such wavelets or basis functions. These basic functions are obtained from a single prototype called the mother wavelet, by dilations (scaling) and translations. Thus, it is a useful tool for the analysis of transient, non-stationary, or time-varying phenomena.

Wavelet System uses a two-dimensional expansion set (usually a basis) to construct or represent one-(or higher) dimensional signals. Thus, a signal can be expressed as

$$f(t) = \sum_k \sum_j a_{j,k} \psi_{j,k}(t)$$

where both j and k are integer indices and $\psi_{j,k}(t)$ are wavelet expansion functions.

The Discrete Wavelet Transform (DWT) of the signal is given by $a_{j,k}$. As the index k changes, the location and the scaling of the wavelet moves along the time axis. As the index j changes, the shape of the wavelet changes in scale. As the scale becomes finer (j becomes larger), the time steps become smaller. Both the narrower wavelet and the smaller steps allow a representation of greater detail or resolution.

6.1 Wavelet System Characteristics

1. Wavelet functions are generated from a single scaling functions or a mother wavelet by simply scaling and translating. The two-dimensional expansion is achieved from the generating wavelet or the mother wavelet $\psi(t)$ given by

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

Their graphs are compressed by a factor of $2^{j/2}$ and translated by k .

2. Wavelet expansions localize the signals both in time and frequency. Hence, a signals energy is represented by a few expansion coefficients $a_{j,k}$.
3. Efficiently calculates the DWT. Lower resolution coefficients can be calculated by higher resolution coefficients using the tree structured algorithm known as the filter bank.
4. It satisfies the multiresolution conditions. That is, if the basic functions are made half as wide and translated in steps half as wide, they will represent a larger class of signals or functions exactly. Thus, giving a better approximation of any signal.

6.2 Relevance to Image Compression

Over the past several years, the wavelet transform has gained widespread acceptance in the image compression field. In many applications wavelet-based schemes (also referred as subband coding) outperform other schemes like DCT. Wavelet-based coding is more robust under transmission and decoding errors. Due to their inherent multiresolution nature, they are suitable for applications where scalability and tolerable degradation are important.

Wavelets are a mathematical tool for changing the coordinate system in which we represent the signal to another domain that is best suited for compression. One of the significant features of wavelets is the compactness of the bulk of the signal energy into the fewest number of transform coefficients. Thereby, we can reduce the image size without losing much of the information.

Wavelets are tools for decomposing signals, such as images, into a hierarchy of increasing resolutions. The more resolution layers, the more detailed features of the image are shown. They are localized waves that drop to zero. They come from the iteration of filters together with rescaling. Wavelets produce a natural multiresolution of every image, including the all-important edges. The output from the low pass channel is a useful compression.

6.3 Haar Wavelet Transform

These are the most basic and the oldest wavelet transforms. Alfred Haar discovered them in 1910. They came into the front in 1980's when considerable research was going in the field of Wavelet Theory. Haar scaling functions and wavelet functions are depicted in Figure 6.1 a) Scaling function and b) Wavelet function.

Figure 6.1 a) Scaling Function

b) Wavelet Function

Haar [5, 1] showed that certain square wave functions could be translated and scaled to create a basis set. The haar scaling function are given by

$$\phi(t) = \begin{cases} 1 & \text{for } 0 \leq t < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\psi(t) = \begin{cases} 1 & \text{for } 0 < t < 1/2 \\ -1 & \text{for } 1/2 < t < 1 \\ 0 & \text{otherwise} \end{cases}$$

The whole set of basis functions are derived by translations and dilations as

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad j, k \in \mathbb{Z}$$

The basis set is orthonormal and are well localized in time (as the support is finite).

6.4 Effectiveness of Wavelet Analysis

Wavelet transforms are very effective in the analysis of signals especially in the field of Image Compression. It has an unconditional basis as a result the size of the wavelet coefficients drop off rapidly with j and k for a large class of signals. The wavelet expansion coefficients represent a local component thereby making it easier to interpret. Wavelets are adjustable and hence can be designed to suit the individual applications. Its generation and calculation of DWT is well suited to the digital computer. There are only multiplications and additions in the calculations of wavelets, which are basic to a digital computer.

Chapter 7 Example Of Image Compression

7.1 Outline

In the main code an image was read firstly. In this thesis, the image taken is of Lena [4]. It is shown in Figure 7.1.

Figure 7.1 Original Lena Image

Then Haar Wavelet Transform (HWT) was applied on it, so as to transform the image into a new domain. As a result, all the energy is concentrated in a few of the wavelet coefficients. Octave-band decomposition is used here. When HWT is first applied, it divides the image into four bands and the coarser detail is given in the upper left hand corner and the rest of the three components give the detail coefficients of the transformed image. It is a loss less compression as there is no loss of data and the image can be

perfectly reconstructed as the HWT transforms are orthonormal and hence the inverse is just the transpose.

In order to have as few coefficients as possible and to have a sparse representation (that is most of the coefficients to be as small as possible or Zero), once again the HWT is applied to the upper Left corner of the image. Once again the image is decomposed into four images where the main energy is in the extreme left corner of the image and the rest give the detail. This is the next level of the HWT. Successively applying HWT to the upper left corner of the image at each level, decomposes the image further and it is continued till the extreme upper corner is a matrix of 16 by 16. HWT is applied once and it can be seen in the Figure 7.2.

Figure 7.2 Image after one Haar Wavelet Transform

After the HWT transform, Quantization is applied to it and this is where the lossy compression takes place. It is here that there is a loss of data. Scalar Quantization is then applied on the wavelet transformed coefficients. Emphasis is given on the various

Uniform Quantization applied to the given transformed coefficients. Analysis is made on the image taking 8 bpp (bits per pixel), 4 bpp, 2 bpp and .5 bpp. It is shown in the Figure 7.3, Figure 7.4, Figure 7.5, and Figure 7.6. In all the images, the image is seen on the upper left corner of the image. Clearly the various uniform quantization show that as the bit per pixel are reduced the picture's fidelity (quality) is degraded and the 8 bpp gives the best image after quantization. The various Uniform Quantized Images are given below

Figure 7.3 Image at 8 bpp

Figure 7.4 Image at 4 bpp

Figure 7.5 Image at 2 bpp

Figure 7.6 Image at .5 bpp

Also, Adaptive Quantization was applied on the image. In this all the four images are given different bits per pixel depending on the energy concentration of the image. Most of the energy is concentrated on the upper left corner and hence it was given 8 bpp, and then the other three were given 6 bpp, 4 bpp and 0 bpp. The image is given in Figure 7.7.

Figure 7.7 Adaptive Quantization. The bits used are (8, 6, 4, 0) bpp

Then the images are reconstructed. Firstly all the pixel values are got into the range from 0 - 255 . 0 represents the black color and 255 correspond to white. These images can be seen in Figure 7.8, Figure 7.9, Figure 7.10, Figure 7.11, and Figure 7.12.

Figure 7.8 Reconstructed Lena Image at 8 bpp

Figure 7.9 Reconstructed Lena Image at 4 bpp

Figure 7.10 Reconstructed Lena Image at 2 bpp

Figure 7.11 Reconstructed Lena Image at .5 bpp

Figure 7.12 Reconstructed Image after Adaptive Quantization. (8,6,4,0 bpp)

7.2 Analysis

In this thesis emphasis were given on the amount of compression used and how good the reconstruction image resembles the original. Also, analysis was done on the amount of distortion, which was calculated using important distortion measures: - mean-square-error (MSE) and peak signal to-noise ratio(PSNR) measured in decibels (dB). A good compression algorithm would reconstruct the image with low MSE and high PSNR.

The adaptive quantization gives more compression as compared to the uniform quantization. It is clear from the reconstructed image that adaptive quantization gives more distortion than the uniform quantization. When .5 bits per pixel were used in uniform quantization, there is no image; only a brief outline of the image can be faintly seen. The best image out of all the scalar quantization techniques used here was of the uniform quantization of 8 bits per pixel. Emphasis was made on how good the reconstruction image resembles the original and its performance measure in reference to MSE and PSNR (defined in equation 7.1 and 7.2 respectively). This is shown below in Table 1 and Table 2.

$$MSE=(1/n)^2*\sum((OrigA(:) - Im(:)).^2) \quad (7.1)$$

where n is the length of the transformed matrix.

$$PSNR=20*\log_{10} (255^2 / MSE) \quad (7.2)$$

7.2.1 Uniform Quantization

The compression ratio for all of the bpp was 1.7655 : 1

Bits per pixel (bpp)	MSE	PSNR (dB)
8	0.9363	96.8333
4	588.6918	40.862
2	3.6680e+003	24.973
.5	2.0646e+005	-10.0352

Table 7.1 Uniform Quantization at different bits per pixel

7.2.2 Adaptive Quantization

The compression ratio for all of the bpp was 2.1167 : 1

Bits per pixel (bpp)	MSE	PSNR (dB)
8,6,4,0	198.9442	50.287

Table 7.2 Adaptive Quantization (8,6,4,0) bpp

Chapter 8 Conclusion and Future Directions

Conclusion

Wavelet transforms was applied to the Lena image in order to study the compression using scalar quantization techniques. The compression performance was evaluated on the basis of PSNR, MSE and, as well as how the reconstructed image resembles the original image. The coding for the analysis was done on a Pentium 3, 633 MHz. From the analysis it is clear that the uniform quantization of 8 bits per pixel gives the best reconstructed image. Also, it has a high PSNR and a low MSE as compared to the other uniform quantization using different bits per pixel.

Future Directions

Image compression is a vast subject and while working on the thesis, major emphasis was made on reconstructing the image using scalar quantization techniques. In a transform framework the image can be further compressed using an encoder. So, the next step would be to further compress the image using an encoder. Huffman coding and, arithmetic coding are the two most popular entropy coding methods.

Also while reconstructing the image, only one level of Haar Wavelet Transform was used. Reconstructing image from second level of Haar Wavelet Transform was giving some trouble as the exact image was not able to be constructed. Hence, effort will be made to reconstruct from higher levels.

Finally, implementation of this image compression will be done on the TMS320C5402 DSP processor.

References

- [1] Burrus, C.S., Gopinath, R.A., and Guo, H. (1998). *Introduction to Wavelets and Wavelet Transforms A Primer*. New Jersey: Prentice Hall.
- [2] Chui, C. (1992). *An Introduction to Wavelets*, San Diego: Academic Press Inc.
- [3] Davis, G., Danskin, J., and Heasman, R. *Wavelet image compression construction kit* . < <http://www.cs.dartmouth.edu/gdavis/wavelet/wavelet.html> > (2002, March 12).
- [4] Dugad, R., and Ahuja, A. *A Fast Scheme For Image Size Change in the Compressed Domain*. < <http://vision.ai.uiuc.edu/~dugad/research/dct/index.html> > (2002, April 18).
- [5] Haar, A. (1910). Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen*, 69:331-371. Also in PHD thesis.
- [6] Hanselman, D., and Littlefield, B. (1997). *The Student Edition of MatLab Version 5 User's Guide*. The Math Works Inc. New Jersey: Prentice Hall.
- [7] Oppenheim, A.V., and Schafer, R.W. (1989). *Discrete- Time Signal Processing*. New Jersey: Prentice Hall.
- [8] Saha, S. *Image Compression – from DCT to Wavelets a review*. < <http://www.acm.org/crossroads/xrds6-3/sahaimgcoding.html> > (2002, March 2).
- [9] Sayood, K. (2000). *Introduction to Data Compression*. (2nd ed.). Morgan Kaufman Pub.
- [10] Shapiro, J.M. (December 1993). Embedded Image Coding Using Zerotrees of Wavelet Coefficients. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, SP-41:3445 - 3462.
- [11] Strang, G., and Nguyen, T. (1996). *Wavelets and Filter Banks*. New York: Wellsley- Cambridge Press.
- [12] Strum, R.D., and Kirk, D.E. (1988). *First Principles of Discrete Systems and Digital Signal Processing*. Addison-Wesley Publishing company.
- [13] Vaidyanathan, P.P. (1992). *Multirate Systems and Filter Banks*. New Jersey: Prentice Hall.

- [14] Vetterli, M. and Kovačević, J. (1995). *Wavelets and Subband Coding*. New Jersey: Prentice Hall.
- [15] Xiong,Z. and Ramchandran, K. (2000). “ Wavelet image compression,” *Handbook of Image and Video Processing*, A. Bovik, ed., Academic Press.

Appendix

10.1 MatLab Program

In this code HWT is applied and then transformed coefficients are uniformly quantized and then Inverse HWT is applied to reconstruct the image. Also, MSE and PSNR are found.

```
close all;

% Reading in the file

A=imread('lena.jpg');

n=length(A);

Im=double(A);

%Image of the original image

figure(1)

colormap(gray(256))

imagesc(Im),title('Original');

[row,col]=size(Im);      % It is 256 by 256


% Applying the HAAR Wavelet Transform at one level

while (n>64)

    Im(1:n,1:n)=haar_col(haar_row(Im(1:n,1:n)));

    n=n/2;

end
```

```

figure(2)

colormap(gray(256))

imagesc(Im),title('Image after one HWT');

A=Im;

n=length(A);

% SCALOR QUANTIZATION

% First of all find the largest coefficient and then scale all the coefficients

% in the range of -127 to 127

max_p = max (max(abs(A))) % it is 223.125

scaleA = A*127/max_p; % this is done so that all the wavelet

%coefficients have 8-bit resolution. Now all the values are between -127 to 127

%scaleA = A*7/max_p;% this is done so that all the wavelet coefficients have a 4- bit

resolution

%scaleA =A*3/max_p;% this is done so that all the wavelet coefficients have a 2 - bit

resolution

%scaleA =A*.424/max_p;% this is done so that all the wavelet coefficients have a .5 - bit

resolution

% Now quantize by simply rounding or using the floor or using the mid point of the

interval. Here midpoint of the interval is used

bits=0;

for i=1:n

    for j=1:n

```

```

    U(i,j)=(floor(scaleA(i,j))+ ceil(scaleA(i,j)))/2;

    if(U(i,j)>0)

        bits=1+bits :% to find the number of bits

    end

end

end

end

figure(3)

colormap(gray(256))

imagesc(U).title('Image after the Quantization');

Im=U;

% Reconstructing the image from the Quantized wavelet transform

% first rescaling is done

rescale=Im*max_p/127;

%rescale=Im*max_p/7;

%rescale=Im*max_p/3;

%rescale=Im*max_p/.424;

A=rescale;

N=length(A)

n2=128;

% now inverse discrete haar transform is applied

while(n2<N)

    k=haar_col_inverse(A(1:n2,1:n2));

```



```

length(k); % the length is k=64,128,256

Im(1:n2,1:n2)=ihaar_row(k);

n2=n2*2;

end

figure(4)

colormap(gray(256))

imagesc(Im),title('Reonstruted after Inverse haar');

%To test the compression and measure the error using a simplest compression scheme
% first of all the wavelet transformed ( scaled to 8bpp) and the reconstructed picture
% are written to 2 different files

imwrite(scaleA,'HaarWavelet_lena.jpg','Quality',100);

imwrite(Im,'Recons_lena.jpg','Quality',100);

% Estimate the compression

% number of bits of the original image

Orig_Image_size=8*row*col; % it is 8*65536=524288 bits

%to find the number of bits of the compressed image

bits % it is 37121

Compressed_Image_size=8*bits;

Compression=Orig_Image_size/Compressed_Image_size % it is 1.7655 : 1

% to find the Mean Square Error

MSE=(1/n)^2*sum((OrigA(:) - Im(:)).^2) % it is .9363

PSNR=20*log10(255^2/MSE) % it is 96.8333

```

```

function B = haar_col(A);

% This function performs sums and differences column-wise for the Haar Wavelet
% Transforms. The result is then placed back into the input matrix.

n = length(A);

for j = 1:n

    num = 1;

    for i = 1:2:(n-1)

        B(num,j) = (A(i,j) + A(i+1,j))/2;

        B(num + n/2,j) = (A(i,j) - A(i+1,j))/2;

        num = num + 1;

    end

end
end

```

```

function B = haar_col_inverse(A);

% This function performs sums and differences column-wise for the Haar Wavelet
% Transforms. The result is then placed back into the input matrix.

n = length(A);

for j = 1:n

    num = 1;

    for i = 1:(n/2)

        B(num,j) = (A(i,j) + A(i+n/2,j));

        B(num+1,j) = (A(i,j) - A(i+n/2,j));

        num = num + 2;

    end

end
end

```

```

function B = haar_row(A);

% This function performs sums and differences column-wise for the Haar Wavelet
% Transforms. The result is then placed back into the input matrix.

n = length(A);

for i = 1:n
    num = 1;
    for j = 1:2:(n-1)
        B(i,num) = (A(i,j) + A(i,j+1))/2;
        B(i, num + n/2) = (A(i,j) - A(i,j+1))/2;
        num = num + 1;
    end
end
end

```

```

function B = ihaar_row(A);

% This function performs sums and differences column-wise for the Haar Wavelet
% Transforms. The result is then placed back into the input matrix.

n = length(A);
for i = 1:n
    num = 1;
    for j = 1:(n/2)
        B(i,num) = (A(i,j) + A(i,j+n/2));
        B(i,num+1) = (A(i,j) - A(i,j+n/2));
        num = num + 2;
    end
end
end

```

Adaptive quantization on the image and then reconstructing it.

```
close all;

% Reading in the file
A=imread('lena.jpg');

n=length(A);

OrigA=double(A);

Im=double(A);

%Image of the original image
figure(1)

colormap(gray(256))

imagesc(Im),title('Original');


MAX=max(max(A)); % the value is 251

MIN=min(min(A)); % the value is 8

[row,col]=size(Im); % It is 256 by 256


% Applying the HAAR Wavelet Transform at one level
while(n>64)

    Im(1:n,1:n)=haar_col(haar_row(Im(1:n,1:n)));

    n=n/2;

end
```

```

figure(24)

colormap(gray(256))

imagesc(Im),title('Image after one HWT');

MAX_WT=max(max(Im)) % the value is 229

MIN_WT=min(min(Im))% the value is -69.5

A=Im;

n=length(A);

% Adaptive QUANTIZATION

% first of all find the largest coefficient and then scale all the coefficients

% in the range of -127 to 127

max_p = max(max(abs(A)));

scaleA1 = A(1:128,1:128).*127/max_p; % this is done so that all the wavelet

%coefficients have 8-bit resolution. Now all the values are between -127 to 127

% the rest of the coefficients have a 4 bit resolution

AH=A(1:128,129:256);

AV=A(129:256,1:128);

AD=A(129:256,129:256);

Max1=max(max(AH))

Max2=max(max(AV))

Max3=max(max(AD))

scaleA2 = AH.*31/max_p; % here it is 6 bpp

```

```

scaleA3= AV.*7/max_p; % here it is 4 bpp
scaleA4= AD.*0/max_p; % here it is 0 bpp
scaleA=[scaleA1,scaleA2,scaleA3,scaleA4];
size(scaleA)

% Now quantize by simply rounding or using the floor or using the mid point of the
interval. Here midpoint of the interval is used

bits=0;

for i=1:n
    for j=1:n
        U(i,j)=(floor(scaleA(i,j))+ ceil(scaleA(i,j)))/2;
        if(U(i,j)>0)
            bits=1+bits;% to find the number of bits
        end
    end
end

end

figure(34)
colormap(gray(256))
imagesc(U),title('Image after the Quantization');

Im=U;

MAX_UQ=max(max(Im)); % the value is 127

MIN_UQ=min(min(Im)); % the value is -39

```



```

% Reconstructing the image from the Quantized wavelet transforms

% Rescaling

rescale1=Im(1:128,1:128).*max_p/127;

rescale2=Im(1:128,129:256).*max_p/31;

rescale3=Im(129:256,1:128).*max_p/7;

rescale4=Im(129:256,129:256).*0;

A=[rescale1,rescale2;rescale3,rescale4];

MAX_DUQ=max(max(A)) % the value is 229

MIN_DUQ=min(min(A))% the value is -81.7857

N=length(A)

n2=128;

% Applied the inverse discrete haar transform

while(n2<=N)

    k=haar_col_inverse(A(1:n2,1:n2));

    length(k); % the length is k=64,128,256

    Im(1:n2,1:n2)=ihaar_row(k);

    n2=n2*2;

end

MAX_IWT=max(max(Im)) % the value is 277.1699

MIN_IWT=min(min(Im)) % the value is -21.1226

```

```

% this is done so that the reconstructed pixels come out between 0 to 255

L=length(Im)

for i=1:L
    for j=1:L
        if(Im(i,j)>=255)
            Im(i,j)=255;
        if(Im(i,j)<=0)
            Im(i,j)=0;
        end
    end
end

end

figure(4)

colormap(gray(256))

imagesc(Im),title('Reonstruted after Inverse haar');


%To test the compression and measure the error using a simplest compression scheme

imwrite(scaleA,'HaarWavelet_lena.jpg','Quality',100);

imwrite(Im,'Recons_lena.jpg','Quality',100);

% Estimate the compression

% number of bits of the original image

Orig_Image_size=8*row*col; % it is 8*65536=524288 bits

```

```

%to find the number of bits of the compressed image

bits % it is 30961

Compressed_Image_size=8*bits;

Compression=Orig_Image_size/Compressed_Image_size % it is 2.1167 : 1

% to find the Mean Square Error

MSE=(1/n)^2*sum((OrigA(:) - Im(:)).^2)

PSNR=20*log10(255^2/MSE)

```