PERFORMANCE BASED IMAGE COMPRESSION

A Dissertation Submitted In Partial Fulfillment of the Requirement for the Award of the Degree of

Master of Engineering

In

Computer Technology and Applications

By

SUBMITTED BY

ABHISHEK BHARDWAJ

(ROLL NO. 8541)

UNDER THE ESTEEMED GUIDANCE OF

MR. VINOD KUMAR

(ASSISTANT PROFESSOR, COMPUTER ENGINEERING)

DEPARTMENT OF COMPUTER ENGINEERING

DELHI COLLEGE OF ENGINEERING

UNIVERSITY OF DELHI, DELHI

(2009-2011)

CERTIFICATE

Date: ………………..

This is to certify that the major report entitled as "**PERFORMANCE BASED IMAGE COMPRESSION**" is being submitted by **ABHISHEK BHARDWAJ** University Roll No. 8541, in partial fulfillment for the award of "Master of Engineering Degree in Computer Engineering" in Delhi College of Engineering, Delhi University, and Delhi. He has worked under my guidance and supervision and has fulfilled the requirements for the submission of this report, which has reached the requisite standard.

I wish him success in all his endeavors.

MR. VINOD KUMAR

Assistant Professor and Project Guide
Department of Computer Engineering
Delhi College of Engineering,
University of Delhi, India

ACKNOWLEDGEMENT

It is my distinct pleasure to express my deep sense of gratitude and indebtedness to my learned supervisor Mr. Vinod Kumar, Department of Computer Engineering, Delhi College of Engineering, for his invaluable guidance, encouragement and patient reviews. His continuous inspiration only has made me complete this project. He kept on boosting me time to time for putting an extra ounce of effort to realize this work.

I would also like to thank Dr. Daya Gupta, HOD, Department of Computer Engineering, for her valuable suggestions.

I would also like to take this opportunity to present my sincere regards to all the teachers in the Department of Computer Engineering for their support and encouragement.

I would also like to thank my classmates for their unconditional support and motivation during this work.

Lastly, I would like to thank Almighty GOD for his countless blessings.

ABHISHEK BHARDWAJ
Master in Engineering
(Computer Technology & Application)
College Roll No. - 01/CTA/09
University Roll No. - 8541
Department of Computer Engineering
Delhi College of Engineering, Delhi-110042.

ABSTRACT

The rapid growth of digital imaging applications, including desktop publishing, multimedia, teleconferencing, and high-definition television (HDTV) has increased the need for effective and standardized image compression techniques. The purpose of image compression is to achieve a very low bit rate representation, while preserving a high visual quality of decompressed images.

As use and reliance on computers continues to grow, so does the need for efficient ways of storing large amount of data. For example, someone with a web page or online catalog that uses dozens or hundreds of images will more than likely need to use some form of image compression to store those images. The purpose of image compression is to achieve a very low bit rate representation, while preserving a high visual quality of decompressed images.

Compression reduces the storage and finds its potency and limitations. Transmission burdens of raw information by reducing the ubiquitous redundancy without losing its entropy significantly. The image manipulation that occupies a significant position in multimedia technology necessitated the development of JPEG compression technique, which has proved its usefulness Until recently, to minimize the blocking artifact, inherently present in JPEG at higher compression ratios, JPEG2000 is devised that makes use of wavelet function.

In this work, a new approach to JPEG compression technique is proposed that enhanced the compression performances in comparison with aforesaid JPEG techniques. The new technique considers both Discrete Cosine Transform (DCT) based (DCT, SVD, BTC) and Discrete Wavelet Transform (DCT) based (PYRAMID, EZW) methods in the transformation and reconstruction sides for best performed algorithm. A rigorous comparison of the various compressions through quality components (PSNR, MSE).

The proposed Algorithm select the best possible algorithm based on the decision parameter for image to achieve low mean square error (MSE), better peak signal to noise ratio (PSNR), a high Compression ratio (CR), while preserving good fidelity of decompressed image.

MATLAB codes have been developed for all the possible combinations, separately.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CR	Compression Ratio
DCT	Discrete Cosine Transform
FT	Fourier Transform
HT	Hartley Transform
PSNR	Peak Signal to Noise Ratio
PCM	Pulse Code Modulation
RLE	Run Length Encoding
RMSE	Root Mean Square Error
JPEG	joint photographic expert group.
LAR	Locally Adaptive Resolution
MRA	Multi Resolution Analysis
STFT	Short time Fourier transforms
STD	Standard Deviation
SVD	Singular value decomposition
BTC	Block truncation code
EZW	Embedded Zerotree Wavelet
ZTR	Zerotree root

LIST OF FIGURES

CHAPTER –1

INTRODUCTION

Now a day, the usage of digital image in various applications is growing rapidly. Video and television transmission is becoming digital and more and more digital image sequences are used in multimedia applications.

A digital image is composed of pixels, which can be thought of as small dots on the screen and it becomes more complex when the pixels are colored. An enormous amount of data is produced when a two dimensional light intensity function is sampled and quantized to create a digital image. In fact, the amount of data generated may be so great that it results in impractical storage, processing and communications requirements [1].

1.1 Fundamentals of Digital Image

An image is a visual representation of an object or group of objects. When using digital equipment to capture, store, modify and view photographic images, they must first be converted to a set of numbers in a process called digitization or scanning. Computers are very good at storing and manipulating numbers, so once the image has been digitized it can be used to archive, examine, alter, display, transmit, or print photographs in an incredible variety of ways. Each pixel of the digital image represents the color (or gray level for black & white images) at a single point in the image, so a pixel is like a tiny dot of a particular color. By measuring the color of an image at a large number of points, we can create a digital approximation of the image from which a copy of the original image can be reconstructed. Pixels are a little grain like particles in a conventional photographic image, but arranged in a regular pattern of rows and columns [1, 2]. A digital image is a rectangular array of pixels sometimes called a bitmap. It is represented by an array of N rows and M columns and usually N=M. typically values of N and M are 128, 256, 512 and 1024 etc.

1.2 Types of Digital Image

For photographic purposes, there are two important types of digital images: color and black & white. Color images are made up of colored pixels while black & white images are made of pixels in different shades of gray.

1.2.1 Black & White Images

A black & white image is made up of pixels, each of which holds a single number corresponding to the gray level of the image at a particular location. These gray levels span the full range from black to white in a series of very fine steps, normally 256 different grays [1]. Assuming 256 gray levels, each black and white pixel can be stored in a single byte (8 bits) of memory.

1.2.2 Color Images

A color image is made up of pixels, each of which holds three numbers corresponding to the red, green and blue levels of the image at a particular location. Assuming 256 levels, each color pixel can be stored in three bytes (24 bits) of memory. Note that for images of the same size, a black & white version will use three times less memory than a color version.

1.2.3 Color Models

The purpose of a color model is to facilitate the specification of colors in some standard generally accepted way. In essence, a color model is a specification of a 3-D coordinate system and a subspace within that system where each color is represented by a single point. Each industry that uses color employs the most suitable color model. For example, the RGB color model is used in computer graphics, YUV or YCbCr are used in video systems, PhotoYCC is used in PhotoCD production and so on. Transferring color information from one industry to another requires transformation from one set of values to another. Intel IPP provides a wide number of functions to convert different color spaces to RGB and vice versa.

1.2.3.1 RGB Color Model

In the RGB model, each color appears as a combination of red, green, and blue. This model is called additive, and the colors are called primary colors. The primary colors can be added to produce the secondary colors of light (see Figure "Primary and Secondary Colors for RGB and CMYK Models") - magenta (red plus blue), cyan (green plus blue), and yellow (red plus green). The combination of red, green, and blue at full intensities makes white.

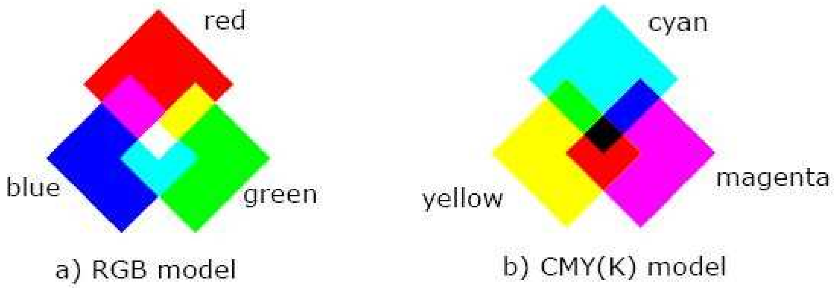


FIGURE1.1 Primary **and Secondary Colors for RGB and CMYK Models**

The color subspace of interest is a cube shown in Figure "RGB and CMY Color Models" (RGB values are normalized to 0…1), in which RGB values are at three corners; cyan, magenta, and yellow are the three other corners, black is at their origin; and white is at the corner farthest from the origin. The gray scale extends from black to white along the diagonal joining these two points. The colors are the points on or inside the cube, defined by vectors extending from the origin. Thus, images in the RGB color model consist of three independent image planes, one for each primary color. As a rule, the Intel IPP color conversion functions operate with non-linear gamma-corrected images R'G'B'. The importance of the RGB color model is that it relates very closely to the way that the human eye perceives color. RGB is a basic color model for computer graphics because color displays use red, green, and blue to create the desired color. Therefore, the choice of the RGB color space simplifies the architecture and design of the system. Besides, a system that is designed using the RGB color space can take advantage of a large number of existing software routines, because this color space has been around for a number of years.

1.2.4 Binary Images

Binary images use only a single bit to represent each pixel. Since a bit can only exist in two states- ON or OFF, every pixel in a binary image must be one of two colors, usually black or white. This inability to represent intermediate shades of gray is what limits their usefulness in dealing with photographic images.

1.3 Image Compression

Image compression addresses the problem of reducing the amount of data required to represent a digital image. It is a process intended to yield a compact representation of an image, thereby reducing the image storage/transmission requirements.

1.3.1 Need for compression

The following example illustrates the need for compression of digital images.

- To store a color image of a moderate size, e.g. 512×512 pixels, one needs 0.75 MB of disk space.
- A 35mm digital slide with a resolution of 12µm requires 18 MB.
- One second of digital PAL (Phase Alternation Line) video requires 27 MB.

To store these images, and make them available over network (e.g. the internet), compression techniques are needed. Image compression addresses the problem of reducing the amount of data required to represent a digital image. The underlying basis of the reduction process is the removal of redundant data. According to mathematical point of view, this amounts to transforming a two-dimensional pixel array into a statistically uncorrelated data set. The transformation is applied prior to storage or transmission of the image. At receiver, the compressed image is decompressed to reconstruct the original image or an approximation to it. The initial focus of research efforts in this field was on the development of analog methods for reducing video transmission bandwidth, a process called bandwidth compression. The example below clearly shows the importance of compression [1].

An image, 1024 pixel×1024 pixel×24 bit, without compression, would require 3 MB of storage and 7 minutes for transmission, utilizing a high speed, 64 kbits/s, and ISDN line. If the image is compressed at a 10:1 compression ratio, the storage requirement is reduced to 300 KB and the transmission time drop to 6 seconds.

Table 1.1: Multimedia data types and uncompressed storage space, transmission bandwidth, and transmission time required

Multimedia Data	**Size/Duration**	**Bits/Pixel or Bits/Sample**	**Uncompressed Size (B for bytes)**	**Transmission Bandwidth (b for bits)**	**Transmission Time (using a 28.8K Modem)**
A page of text	11" x 8.5"	Varying resolution	4-8 KB	32-64 Kb/page	1.1 - 2.2 sec
Telephone quality speech	10 sec	8 bps	80 KB	64 Kb/sec	22.2 sec
Grayscale Image	512 x 512	8 bpp	262 KB	2.1 Mb/image	1 min 13 sec
Color Image	512 x 512	24 bpp	786 KB	6.29 Mb/image	3 min 39 sec
Medical Image	2048 x 1680	12 bpp	5.16 MB	41.3 Mb/image	23 min 54 sec
SHD Image	2048 x 2048	24 bpp	12.58 MB	100 Mb/image	58 min 15 sec
Full-motion Video	640 x 480, 1 min	24 bpp	1.66 GB	221 Mb/sec	5 days 8 hrs

1.3.2 Principle behind compression

A common characteristic of most images is that the neighboring pixels are correlated and therefore contain redundant information. The foremost task then is to find less correlated representation of the image. Two fundamental components of compression are redundancy and irrelevancy reduction.

Redundancies reduction aims at removing duplication from the signal source (image/video).

Irrelevancy reduction omits parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System.

In an image, which consists of a sequence of images, there are three types of redundancies in order to compress file size. They are:

- **Coding redundancy:** Fewer bits to represent frequent symbols.
- **Interpixel redundancy:** Neighboring pixels have similar values.
- **Psychovisual redundancy:** Human visual system cannot simultaneously distinguish of all colors.

1.3.3 Types of compression

Compression can be divided into two categories, as Lossless and Lossy compression. In lossless compression, the reconstructed image after compression is numerically identical to the original image. In lossy compression scheme, the reconstructed image contains degradation relative to the original. In the case of video, compression causes some information to be lost; some information at a detail level is considered not essential for a reasonable reproduction of the scene. This type of compression is called **lossy compression**. Audio compression on the other hand, is not lossy, it is called **lossless compression**. An important design consideration in an algorithm that causes permanent loss of information is the impact of this loss in the future use of the stored data. Lossy technique causes image quality degradation in each compression/decompression step.

Careful consideration of the human visual perception ensures that the degradation is often unrecognizable, though this depends on the selected compression ratio. In general, lossy techniques provide far greater compression ratios than lossless techniques.

The following are the some of the lossless and lossy data compression techniques:

Lossless coding techniques

- Run length encoding
- Huffman encoding
- Arithmetic encoding
- Entropy coding
- Area coding

Lossy coding techniques

- Predictive coding
- Transform coding (FT/DCT/Wavelets)

1.3.3.1 Lossless versus Lossy compression: In lossless compression schemes, the reconstructed image, after compression, is numerically identical to the original image. However lossless compression can only a achieve a modest amount of compression. Lossless compression is preferred for archival purposes and often medical imaging, technical drawings, clip art or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. An image reconstructed following lossy compression contains degradation relative to the original. Often this is because the compression scheme completely discards redundant information. However, lossy schemes are capable of achieving much higher compression. Lossy methods are especially suitable for natural images such as photos in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences can be called visually lossless [3].

Predictive versus Transform coding: In predictive coding, information already sent or available is used to predict future values, and the difference is coded. Since this is done in the image or spatial domain, it is relatively simple to implement and is readily adapted to local image characteristics. Differential Pulse Code Modulation (DPCM) is one particular example 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 transform and then codes the transformed values (coefficients). This method provides greater data compression compared to predictive methods, although at the expense of greater computational requirements.

FIGURE 1.2

Image Compression model

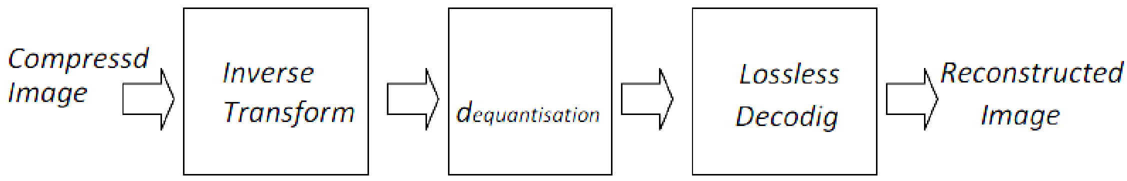


FIGURE 1.3

Image Decompression model

Image compression model shown here consists of a Transformer, quantizer and encoder.

1.3.3.2Transformer: It transforms the input data into a format to reduce interpixel redundancies in the input image. Transform coding techniques use a reversible, linear mathematical transform to map the pixel values onto a set of coefficients, which are then quantized and encoded. The key factor behind the success of transform-based coding schemes is that many of the resulting coefficients for most natural images have small magnitudes and can be quantized without causing significant distortion in the decoded image. For compression purpose, the higher the capability. of compressing information in fewer coefficients, the better the transform; for that reason, the Discrete Cosine Transform (DCT) and Discrete Wavelet Transform(DWT) have become the most widely used transform coding techniques.

Transform coding algorithms usually start by partitioning the original image into subimages (blocks) of small size (usually 8 × 8). For each block the transform coefficients are calculated, effectively converting the original 8 × 8 array of pixel values into an array of coefficients within which the coefficients closer to the top-left corner usually contain most of the information needed to quantize and encode (and eventually perform the reverse process at the decoder's side) the image with little perceptual distortion. The resulting coefficients are then quantized and the output of the quantizer is used by symbol encoding techniques to produce the output bitstream representing the encoded image. In image decompression model at the decoder's side, the reverse process takes place, with the obvious difference that the dequantization stage will only generate an approximated version of the original coefficient values e.g., whatever loss was introduced by the quantizer in the encoder stage is not reversible.

1.3.3.3 Quantizer: It reduces the accuracy of the transformer's output in accordance with some pre-established fidelity criterion. Reduces the psychovisual redundancies of the input image. This operation is not reversible and must be omitted if lossless compression is desired. The quantization stage is at the core of any lossy image encoding algorithm. Quantization at the encoder side, means partitioning of the input data range into a smaller set of values. There are two main types of quantizers: scalar quantizers and vector quantizers. A scalar quantizer partitions the domain of input values into a smaller number of intervals. If the output intervals are equally spaced, which is the simplest way to do it, the process is called uniform scalar quantization; otherwise, for reasons usually related to minimization of total distortion, it is called non uniform scalar quantization. One of the most popular non uniform quantizers is the Lloyd- Max quantizer. Vector quantization (VQ) techniques extend the basic principles of scalar quantization to multiple dimensions.

1.3.3.4 Symbol (entropy) encoder: It creates a fixed or variable-length code to represent the quantizer's output and maps the output in accordance with the code. In most cases, a variable-length code is used. An entropy encoder compresses the compressed values obtained by the quantizer to provide more efficient compression. Most important types of entropy encoders used in lossy image compression techniques are arithmetic encoder, huffman encoder and run-length encoder.

1.3.4 Applications

Over the years, the need for image compression has grown steadily. Currently it is recognized as an "enabling technology." It plays a crucial role in many important and diverse applications [1, 2] such as:

- Business documents, where lossy compression is prohibited for legal reasons.
- Satellite images, where the data loss is undesirable because of image collecting cost.
- Medical images, where difference in original image and uncompressed one can Compromise diagnostic accuracy.
- Televideoconferencing.
- Remote sensing.
- Space and hazardous waste control applications.
- Control of remotely piloted vehicles in military.
- Facsimile transmission (FAX).

Image compression has been and continues to be crucial to the growth of multimedia computing. In addition, it is the natural technology for handling the increased spatial resolutions of today's imaging sensors and evolving broadcast television standards.

CHAPTER 2

Image Compression using Discrete Cosine Transform

2.1. Introduction

JPEG stands for the Joint Photographic Experts Group, a standards committee that had its origins within the International Standard Organization (ISO).JPEG provides a compression method that is capable of compressing continuous-tone image data with a pixel depth of 6 to 24 bits with reasonable speed and efficiency.JPEG may be adjusted to produce very small, compressed images that are of relatively poor quality in appearance but still suitable for many applications. Conversely, JPEG is capable of producing very high-quality compressed images that are still far smaller than the original uncompressed data.

Transform coding constitutes an integral component of contemporary image/video processing applications. Transform coding relies on the premise that pixels in an image exhibit a certain level of correlation with their neighboring pixels. Similarly in a video transmission system, adjacent pixels in consecutive frames2 show very high correlation. Consequently, these correlations can be exploited to predict the value of a pixel from its respective neighbors. A transformation is, therefore, defined to map this spatial (correlated) data into transformed (uncorrelated) coefficients. Clearly, the transformation should utilize the fact that the information content of an individual pixel is relatively small i.e., to a large extent visual contribution of a pixel can be predicted using its neighbors.

A typical image/video transmission system is outlined in Figure2. 1. The objective of the source encoder is to exploit the redundancies in image data to provide compression. In other words, the

Source encoder reduces the entropy, which in our case means decrease in the average number of bits required to represent the image. On the contrary, the channel encoder adds redundancy to the output of the source encoder in order to enhance the reliability of the transmission. Clearly, both these high-level blocks have contradictory objectives and their interplay is an active research area ([4], [5], [6], [7], [8]). However, discussion on joint source channel coding is out of the scope of this document and this document mainly focuses on the *transformation* block in the source encoder. Nevertheless, pertinent details about other blocks will be provided as required.

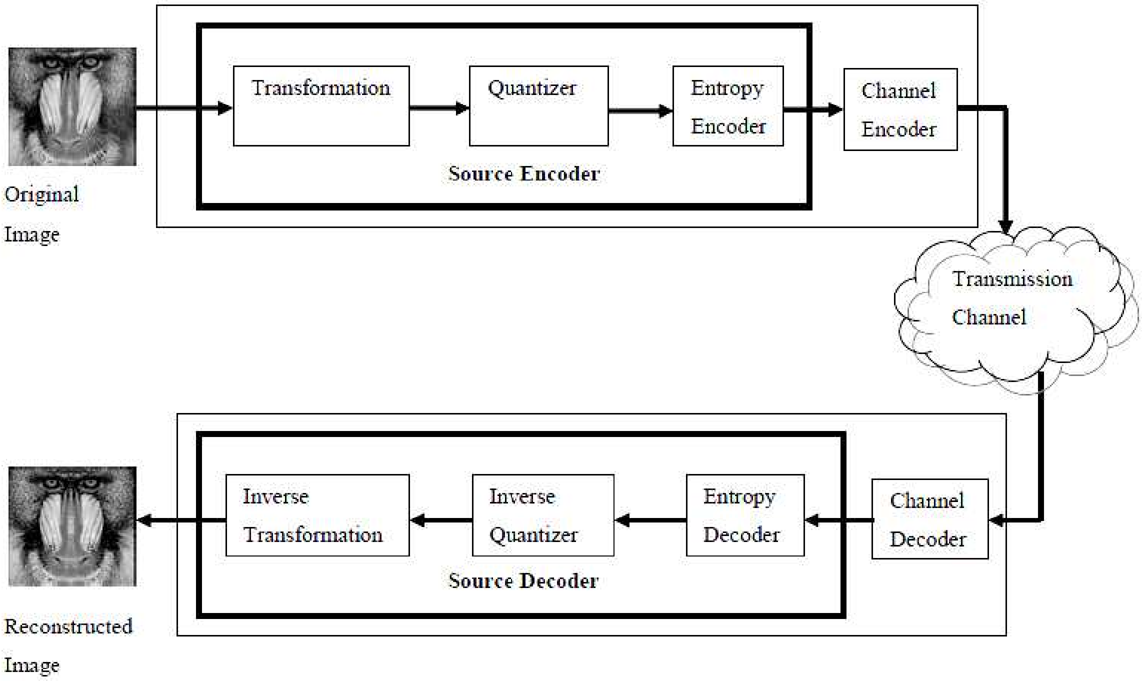
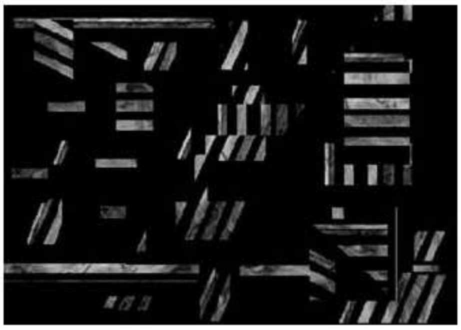


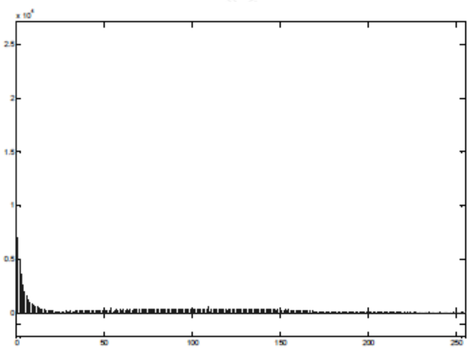
FIGURE 2.1 Components of typical image /video transmission system.

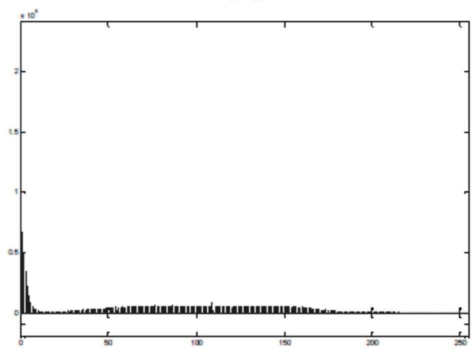
As mentioned previously, each sub-block in the source encoder exploits some redundancy in the image data in order to achieve better compression. The transformation sub-block decorrelates the image data thereby reducing (and in some cases eliminating) *interpixel redundancy3* [11]. The two images shown in Figure 2.2 (a) and (b) have similar histograms (see Figure 2.2 (c) and (d)). Figure 2.2 (f) and (g) show the normalized autocorrelation among pixels in one line of the respective images. Figure 2.2 (f) shows that the neighboring pixels of Figure 2.2 (b) periodically exhibit very high autocorrelation. This is easily explained by the periodic repetition of the vertical white bars in Figure 2.2(b). This example will be will be employed in the following sections to illustrate the decorrelation properties of transform coding. Here, it is noteworthy that transformation is a lossless operation; therefore, the inverse transformation renders a perfect reconstruction of the original image.

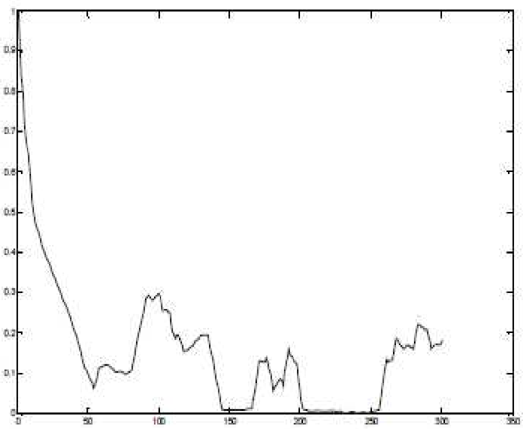


(a)

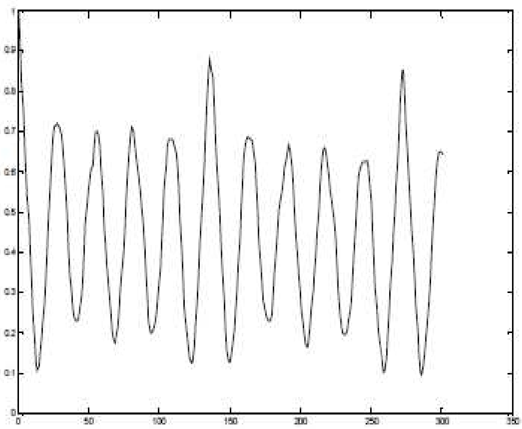
(b)







(e)



(f)

FIGURE 2.2 (a) First Image (b) Second Image (c) Histogram of First image (d) Histogram of Second image (e) Normalized autocorrelation of one line of first image (f) normalized autocorrelation of one line of Second image .

The quantizer sub-block utilizes the fact that the human eye is unable to perceive some visual information in an image. Such information is deemed redundant and can be discarded without introducing noticeable visual artifacts. Such redundancy is referred to as psychovisual redundancy [10]. This idea can be extended to low bit rate receivers which, due to their stringent bandwidth requirements, might sacrifice visual quality in order to achieve bandwidth efficiency. This concept is the basis for rate distortion theory, that is, receivers might tolerate some visual distortion in exchange for bandwidth conservation.

Lastly, the entropy encoder employs its knowledge of the transformation and quantization processes to reduce the number of bits required to represent each symbol at the quantizer output. In the last decade, Discrete Cosine Transform (DCT) has emerged as the de-facto image transformation in most visual systems. DCT has been widely deployed by modern video coding standards, for example, MPEG, JVT etc. This document introduces the DCT, elaborates its important attributes and analyzes its performance using information theoretic measures.

2.2. The Discrete Cosine Transform

Like other transforms, the Discrete Cosine Transform (DCT) attempts to decorrelate the image data. After decorrelation each transform coefficient can be encoded independently without losing compression efficiency. This section describes the DCT and some of its important properties.

2.2.1. The One-Dimensional DCT

The most common DCT definition of a 1-D sequence of length N

$$C(u)=\alpha(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{\pi(2x+1)u}{2N}\right]. \quad (1)$$

For u = 0, 1, 2… $N-1$. Similarly, the inverse transformation is defined as

$$f(x)=\sum_{u=0}^{N-1}\alpha(u)C(u)\cos\left[\frac{\pi(2x+1)u}{2N}\right]. \quad (2)$$

for $x = 0,1,2,\ldots,N-1$. In both equations **(1)** and **(2)** α*(u)* is defined as

$$\alpha(u)=\begin{cases}\sqrt{\frac{1}{N}} & for \quad u=0\\ \sqrt{\frac{2}{N}} & for \quad u\neq 0.\end{cases} \quad (3)$$

It is clear from **(1)** that for $u=0$, $C(u=0)=\sqrt{\frac{1}{N}}\sum_{x=0}^{N-1} f(x)$. . Thus, the first transform coefficient is

the average value of the sample sequence. In literature, this value is referred to as the *DC Coefficient*. All other transform coefficients are called the *AC Coefficients.*

To fix ideas, ignore the *f*(x) and *α(u)* component in **(1)**. The plot of $\sum_{x=0}^{N-1}\cos\left[\frac{\pi(2x+1)u}{2N}\right]$

For N = 8 and varying values of u is shown in Figure 3. In accordance with our previousobservation, the first the top-left waveform (u = 0) renders a constant (DC) value, whereas, all other waveforms (u = 1,2,…,7) give waveforms at progressively increasing frequencies [13].

These waveforms are called the *cosine basis function*. Note that these basis functions are orthogonal. Hence, multiplication of any waveform in Figure 3 with another waveform followed by a summation over all sample points yields a zero (scalar) value, whereas multiplication of any waveform in Figure 3 with itself followed by a summation yields a constant (scalar) value. Orthogonal waveforms are independent, that is, none of the basis functions can be represented as a combination of other basis functions [14].

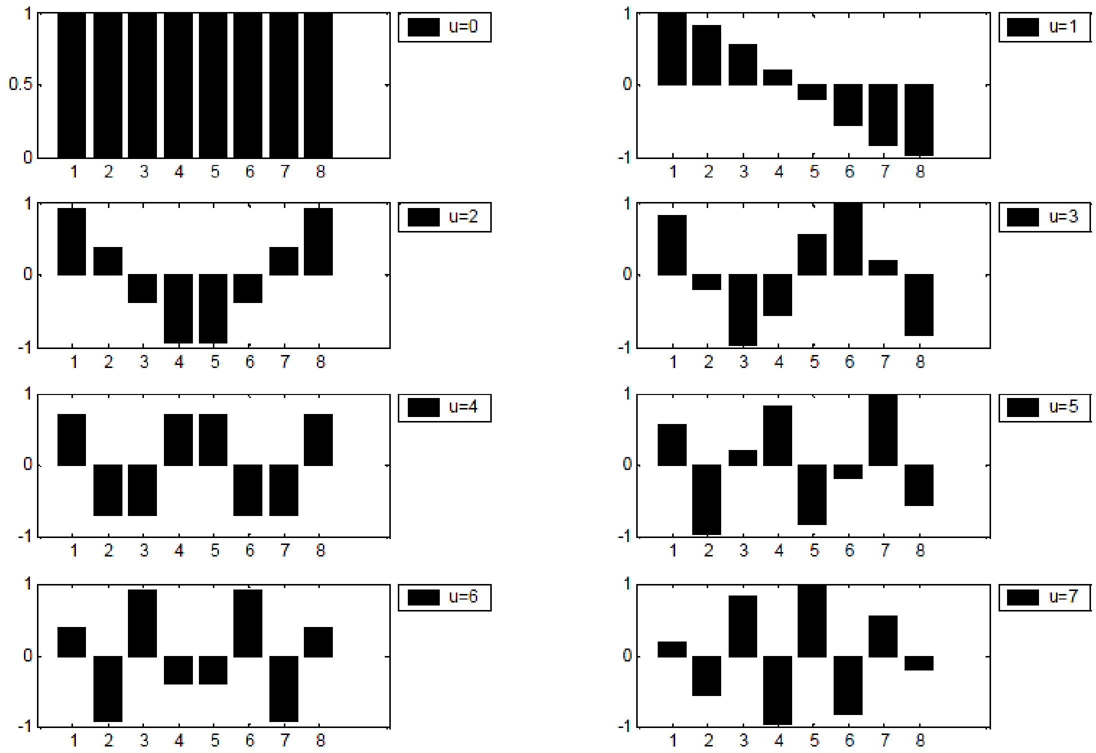


FIGURE 2.3 ONE DIMENSIONAL COSINE BASIS FUNCTION (N=8)

If the input sequence has more than N sample points then it can be divided into sub-sequences of length N and DCT can be applied to these chunks independently. Here, a very important point to note is that in each such computation the values of the basis function points will not change. Only the values of $f(x)$ will change in each sub-sequence. This is a very important property, since it shows that the basic functions can be pre-computed offline and then multiplied with the sub-sequences. This reduces the number of mathematical operations (i.e., multiplications and additions) thereby rendering computation efficiency.

2.2.2. The Two-Dimensional DCT

2.2.2.1 Process

The following is a general overview of the JPEG process. Later we will take the reader through a detailed tour of JPEG's method so that a more comprehensive understanding of the process may be acquired.

1. The image is broken into 8x8 blocks of pixels.
2. Working from left to right, top to bottom, the DCT is applied to each block.
3. Each block is compressed through quantization.
4. The array of compressed blocks that constitute the image is stored in a drastically reduced amount of space.
5. When desired, the image is reconstructed through decompression, a process that uses the Inverse Discrete Cosine transform (IDCT).

2.2.2.2 The DCT Equation

The DCT equation (Eg. 1) computes the i, j^{th} entry of the DCT of an image.

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \qquad 1$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{if } u > 0 \end{cases} \qquad 2$$

P(x, y) is the x, yth elements of the image represented by the matrix p.N is the Size of the block that the DCT is done on. The equation calculates one entry of the transformed image from the Pixel value Of Original image Matrix. For the Standard 8x8 block that JPEG Compression uses N equals 8 and x & y range from 0 to 7.

$$D(i,j) = \frac{1}{4} C(i)C(j) \sum_{x=0}^{7} \sum_{y=0}^{7} p(x,y) \cos\left[\frac{(2x+1)i\pi}{16}\right] \cos\left[\frac{(2y+1)j\pi}{16}\right] \qquad 3$$

Because the DCT use cosine functions, the resulting matrix depends on the horizontal, diagonal, and vertical frequencies. Therefore and image black with a lot of change in frequency has a very random looking resulting matrix, while and image matrix of just one color, has a resulting matrix of a large value for the first element and zeroes for the other elements.

$$T_{i,j} = \begin{cases} \frac{1}{\sqrt{N}} & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{(2j+1)i\pi}{2N}\right] & \text{if } i > 0 \end{cases} \qquad 4$$

For an 8x8 block it results in this matrix:

$$T = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix}$$

.

2.2.2.3 Doing the DCT on an 8x8 Block

The first row (i =1) of the matrix has all the entries equal to $1/\sqrt{8}$ as expected from Equ (4) .The columns of T an orthogonal set, so T is an Orthogonal matrix .When Doing the inverse DCT the orthogonality of T is Important ,as the inverse of T is T' Which is easy to calculate .

Before we begin, it should be noted that the pixel values of a black & white image range from 0 to 255 in step of 1, where pure black is represented by 0 and pure white by 255. Thus it can be seen how a photo, illustration etc. can be accurately represented by these 256 shades of gray. Since an image comprises hundreds or even thousands of 8x8 blocks of pixels , the following Description of what happens to one 8x8 block is a microcosm of JPEG Process.

Now, let's start with a block of image pixel values. This particular block was chosen from the very upper left hand corner of an image.

$$\text{Original} = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

Because the DCT is designed to work on the pixel values ranging from -128 to 127, the original block is leveled off by subtracting 128 from each entry .This results in the following matrix.

$$M = \begin{bmatrix} 26 & -5 & -5 & -5 & -5 & -5 & -5 & 8 \\ 64 & 52 & 8 & 26 & 26 & 26 & 8 & -18 \\ 126 & 70 & 26 & 26 & 52 & 26 & -5 & -5 \\ 111 & 52 & 8 & 52 & 52 & 38 & -5 & -5 \\ 52 & 26 & 8 & 39 & 38 & 21 & 8 & 8 \\ 0 & 8 & -5 & 8 & 26 & 52 & 70 & 26 \\ -5 & -23 & -18 & 21 & 8 & 8 & 52 & 38 \\ -18 & 8 & -5 & -5 & -5 & 8 & 26 & 8 \end{bmatrix}$$

We are now ready to perform the discrete cosine Transform, which is accomplished by matrix multiplication.

$$D = TMT' \qquad 5$$

In Equation (5) matrix M is first multiplied on left by the DCT matrix T from the previous section; This Transforms the rows .The columns are then transformed by multiplying on the right by the transpose of the DCT matrix. This yields the following matrix.

$$D = \begin{bmatrix} 162.3 & 40.6 & 20.0 & 72.3 & 30.3 & 12.5 & -19.7 & -11.5 \\ 30.5 & 108.4 & 10.5 & 32.3 & 27.7 & -15.5 & 18.4 & -2.0 \\ -94.1 & -60.1 & 12.3 & -43.4 & -31.3 & 6.1 & -3.3 & 7.1 \\ -38.6 & -83.4 & -5.4 & -22.2 & -13.5 & 15.5 & -1.3 & 3.5 \\ -31.3 & 17.9 & -5.5 & -12.4 & 14.3 & -6.0 & 11.5 & -6.0 \\ -0.9 & -11.8 & 12.8 & 0.2 & 28.1 & 12.6 & 8.4 & 2.9 \\ 4.6 & -2.4 & 12.2 & 6.6 & -18.7 & -12.8 & 7.7 & 12.0 \\ -10.0 & 11.2 & 7.8 & -16.3 & 21.5 & 0.0 & 5.9 & 10.7 \end{bmatrix}$$

This block matrix now consists of 64 DCT Coefficients, C_{ij} where i and j range from 0 to 7 .The top-left coefficient, C_{00}, correlates to the low frequencies of the Original Image Block .As we move from C_{00} in all directions, the DCT Coefficients Correlate to Higher and higher frequencies of the image block, where C_{77} corresponds to the highest Frequency. It is important to note that the human eye is most sensitive to low Frequencies, and results from the quantization step will reflect this fact.

2.2.2.4 Quantization

Our 8x8 block Of DCT coefficient is now ready for compression by quantization .A remarkable and Highly Useful Feature of the JEPG process is that in this step, varying levels of image compression and quality are obtainable through selection of specific quantization matrices. This enables the user to decide on quality levels ranging from 1 to 100.where 1 gives the poorest image quality and highest compression, where as 100 gives the best quality and lowest compression. As a results, the quality/compression ratio can be tailored to suit different needs.

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Subjective experiments involving the human visual system have resulted in the JPEG Standard quantization matrix .with a quality level of 50, this matrix renders both high compression and excellent decompression image quality.

However, another level of quality and compression is desired, scalar multiples of the JPEG standards quantization matrix may be used. For a quality level greater than 50(less compression, more image quality), the standards quantization matrix is multiplied by (100-quality levels)/50. For a quality level less than 50(more compression, lower image quality), the standards quantization matrix is multiplied by positive integer value ranging from 1 to 255. For example, the following quantization matrices yields quality levels of 10 and 90.

$$Q_{10} = \begin{bmatrix} 80 & 60 & 50 & 80 & 120 & 200 & 255 & 255 \\ 55 & 60 & 70 & 95 & 130 & 255 & 255 & 255 \\ 70 & 65 & 80 & 120 & 200 & 255 & 255 & 255 \\ 70 & 85 & 110 & 145 & 255 & 255 & 255 & 255 \\ 90 & 110 & 185 & 255 & 255 & 255 & 255 & 255 \\ 120 & 175 & 255 & 255 & 255 & 255 & 255 & 255 \\ 245 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 & 255 & 255 & 255 \end{bmatrix}$$

Quantization is achieved by dividing each element in transformed image matrix D by the corresponding elements in the quantization matrix and then rounding to the nearest integer value. For following steps, quantization matrix Q_{50} is used

$$C_{i,j} = \text{round}\,(D_{i,j} / Q_{i,j}) \qquad 6$$

$$C = \begin{bmatrix} 10 & 4 & 2 & 5 & 1 & 0 & 0 & 0 \\ 3 & 9 & 1 & 2 & 1 & 0 & 0 & 0 \\ -7 & -5 & 1 & -2 & -1 & 0 & 0 & 0 \\ -3 & -5 & 0 & -1 & 0 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Recall that the coefficients situated near the upper-left corner correspond to the lower frequencies-to which the human eye is most sensitive –of the image block .In addition, the zero represents the important, higher frequencies that have been discarded, giving rise to the lossy part of compression. As mentioned earlier, only the remaining non Zero coefficients will be used to reconstruct the image .It is also Interesting to note the effect of different quantization matrices; use of Q_{10} would give C significantly more Zeros, while Q_{90} would results in very few zeros.

2.2.2.5 Coding

The quantized matrix C is now ready for the final step of compression .Before storage; all coefficients of c are converted by an encoder to a stream of binary data (0110110011……). After the quantization, it is quite common for the most coefficients equal to zero. Jpeg takes advantage of this encoding quantized coefficient in zig –zag sequence shown in fig.the advantage lies in the consolidation of relatively large runs of zeros, which compress very well.

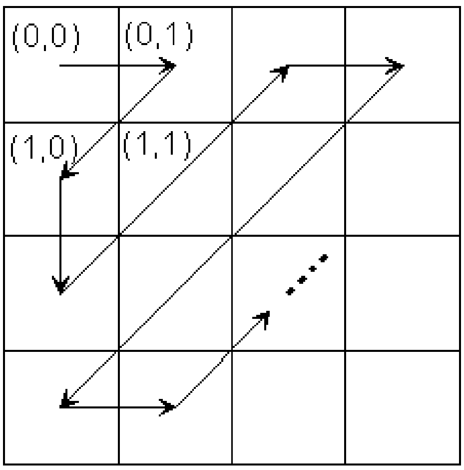


FIGURE 2.3 quantized coefficients encoding in zig-zag

2.2.2.6 Decompression

Reconstruction of our image begins by decoding the bit stream representing the quantized matrix C. Each element of C is then multiplied by the corresponding element of the quantization matrix originally used.

$$R_{i,j} = Q_{i,j} \times C_{i,j} \qquad 7$$

$$R = \begin{bmatrix} 160 & 44 & 20 & 80 & 24 & 0 & 0 & 0 \\ 36 & 108 & 14 & 38 & 26 & 0 & 0 & 0 \\ -98 & -65 & 16 & -48 & -40 & 0 & 0 & 0 \\ -42 & -85 & 0 & -29 & 0 & 0 & 0 & 0 \\ -36 & 22 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The IDCT is next applied to matrix R, which is rounded to the nearest integer. Finally, 128 is added to each element of that result, giving us the decompressed JPEG version N of our original 8x8 image block M.

$$N = \text{round}(T' \; RT) + 128 \qquad 8$$

2.2.2.7 Comparison of Matrices

Let us now see how the jpeg version of our original pixel block compares.

$$\text{Original} = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

$$Decompressed = \begin{bmatrix} 149 & 134 & 119 & 116 & 121 & 126 & 127 & 128 \\ 204 & 168 & 140 & 144 & 155 & 150 & 135 & 125 \\ 253 & 195 & 155 & 166 & 183 & 165 & 131 & 111 \\ 245 & 185 & 148 & 166 & 184 & 160 & 124 & 107 \\ 188 & 149 & 132 & 155 & 172 & 159 & 141 & 136 \\ 132 & 123 & 125 & 143 & 160 & 166 & 168 & 171 \\ 109 & 119 & 126 & 128 & 139 & 158 & 168 & 166 \\ 111 & 127 & 127 & 114 & 118 & 141 & 147 & 135 \end{bmatrix}$$

This is a remarkable result, considering that nearly 70% of DCT coefficients were discarded prior to image block decompression /reconstruction .Given that Similar results will occur with the rest of the block that constitute the entire image, it Should be no surprise that JPEG image will be scarcely distinguishable from the original .Remember there are 256 possible shades of Gray in a black & white picture and a difference of say 10, is barely noticeable to the human eye.

Chapter 3

Singular value decomposition image compression

3.1 INTRODUCTION

It is well known that the images, often used in variety of computer applications, are difficult to store and transmit. One possible solution to overcome this problem is to use a data compression technique where an image is viewed as a matrix and then the operations are performed on the matrix. Image compression is achieved by using Singular Value Decomposition (SVD) technique on the image matrix. The advantage of using the SVD is the property of energy compaction and its ability to adapt to the local statistical variations of an image. Further, the SVD can be performed on any arbitrary, square, reversible and non reversible matrix of m x n size.

The mechanics of singular value decomposition, especially as it relates to some techniques in natural language processing. It's written by someone who knew zilch about singular value decomposition or any of the underlying maths before he started writing it, and knows barely more than that now. Accordingly, it's a bit long on the background part, and a bit short on the truly explanatory part, but hopefully it contains all the information necessary for someone who's never heard of singular value decomposition before to be able to do it.

3.1.1 POINTS AND SPACE

A point is just a list of numbers. This list of numbers, or coordinates, specifies the point's Position in space. How many coordinates there are determines the dimensions of that space.For example, we can specify the position of a point on the edge of a ruler with a single Coordinate. The position of the two points 0:5cm and 1:2cm are precisely specified by single Coordinates. Because we're using a single coordinate to identify a point, we're dealing with Points in one-dimensional space, or 1-space.

The position of a point anywhere in a plane is specified with a pair of coordinates; it takes three coordinates to locate points in three dimensions. Nothing stops us from going beyond points in 3-space. The fourth dimension is often used to indicate time, but the dimensions can be chosen to represent whatever measurement unit is relevant to the objects we're trying to describe.

Generally, space represented by more than three dimensions is called hyperspace. You'll also see the term n-space used to talk about spaces of different dimensionality (e.g. 1-space, 2-space... n-space).

3.1.2 VECTORS

For most purposes, points and vectors are essentially the same thing1, that is, a sequence of numbers corresponding to measurements along various dimensions.

Vectors are usually denoted by a lower case letter with an arrow on top, e.g. $\vec{x}$. The numbers comprising the vector are now called components, and the number of components equals the dimensionality of the vector. We use a subscript on the vector name to refer to the component in that position. In the example below, $\vec{x}$ is a 5-dimensional vector, x1 = 8, x2 = 5, etc.

$$\vec{x} = \begin{pmatrix} 8 \\ 6 \\ 7 \\ 5 \\ 3 \end{pmatrix}$$

Vectors can be equivalently represented horizontally to save space, e.g. $\vec{x}$ = [8, 6, 7, 5, 3] is the same vector as above. More generally, a vector $\vec{x}$ with n-dimensions is a sequence of n Numbers, and component xi represents the value of $\vec{x}$ on the i^{th} dimension.

3.2 Process of SVD

The use of Singular Value Decomposition (SVD) in image compression has been widely studied [15, 16 and 17]. If the image, when considered as a matrix, has low rank, or can be approximated sufficiently well by a matrix of low rank, then SVD can be used to find this approximation, and further this low rank approximation can be represented much more compactly than the original image.

Singular Value Decomposition (SVD) is said to be a significant topic in linear algebra by many renowned mathematicians. SVD has many practical and theoretical values; Special feature of SVD is that it can be performed on any real (m, n) matrix. Let’s say we have a matrix A with m rows and n columns, with rank r *and* $r \leq n \leq m$. Then the A can be factorized into three matrices:

$$A = USV^{T} \qquad (1)$$

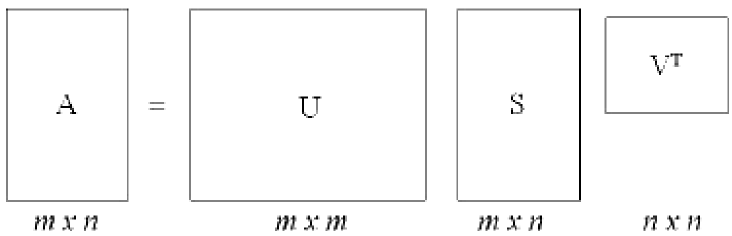


FIGURE 3.1 Illustration of Factoring *A* to *USV*^{t}

Where Matrix U is an $m \times m$ orthogonal matrix

$$U = [\mathrm{u}_1, \mathrm{u}_2, \ldots \mathrm{u}_r, \mathrm{u}_{r+1}, \ldots, \mathrm{u}_{\mathrm{m}}] \qquad (2)$$

Column vectors u_{i} ,for i = 1, 2… m, form an orthogonal set:

$$\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij} = \begin{cases} 1,,, i = j \\ 0,,, i \neq j \end{cases} \quad (3)$$

And matrix V is an $n \times n$ orthogonal matrix

$$V = [\mathrm{v}_1, \mathrm{v}_2, \ldots \mathrm{v}_r, \mathrm{v}_{r+1}, \ldots, \mathrm{v}_n] \quad (4)$$

Column vectors v_i for i = 1, 2… n, form an orthogonal set:

$$\mathrm{v}_i^T \mathrm{v}_j = \delta_{ij} = \begin{cases} 1,,, i = j \\ 0,,, i \neq j \end{cases} \quad (5)$$

Here, S is an $m \times n$ diagonal matrix with singular values (SV) on the diagonal. The Matrix S can be showed in following:

$$S = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \sigma_{r+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & \sigma_n \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} \quad (6)$$

For i = 1, 2… n, σ_i are called Singular Values (SV) of matrix A. It can be proved that

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 \text{, and}$$
$$\sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_N = 0 . \qquad (7)$$

For i = 1, 2… n, σ_i are called Singular Values (SVs) of matrix A. The v_i 's and u_i's are Called right and left singular vector of A.

3.3 Properties of the SVD

There are many properties and attributes of SVD; here we just present parts of the properties that we used in this project.

1. The singular value $\sigma_{1,}$ $\sigma_{2,}$ $\sigma_{3,}$ σ_4............σ_n are unique, however, the matrices *U and V* are not unique;

2. Since $A^T A = VS^TSV^T$, so V diagonalizes $A^T A$, it follows that the $\mathbf{v}_j$*'s* are the eigenvector of $A^T A$.

3. Since $AA^T = USS^T U^T$, so it follows that U diagonalizes AA^T and that the u_i *'s* are the eigenvectors of AA^T.

4. The rank of matrix A is equal to the number of its nonzero singular values.

3.4 METHODOLOGY OF SVD APPLIED TO IMAGE PROCESSING

3.4.1 SVD Approach for Image Compression

Image compression deals with the problem of reducing the amount of data required to represent a digital image. Compression is achieved by the removal of three basic data redundancies: 1) coding redundancy, which is present when less than optimal; 2) Interpixel redundancy, which results from correlations between the pixels; 3) Psycho visual redundancies, which is due to data that is ignored by the human visual. [14].

The property 4 of SVD in section 3.3 tells us “the rank of matrix A is equal to the number of its nonzero singular values”. In many applications, the singular values of a matrix decrease quickly with increasing rank. This propriety allows us to reduce the noise or compress the matrix data by eliminating the small singular values or the higher ranks.

When an image is SVD transformed, it is not compressed, but the data take a form in which the first singular value has a great amount of the image information. With this, we can use only a few singular values to represent the image with little differences from the original. To illustrate the SVD image compression process, we show detail procedures:

$$A = USV^T = \sum_{i=1}^{r} \sigma_i \mathrm{u}_i \mathrm{v}_i^T$$

That is *A* can be represented by the outer product expansion:

$$A = \sigma_1 \mathrm{u}_1 \mathrm{v}_1^{\ T} + \sigma_2 \mathrm{u}_2 \mathrm{v}_2^{\ T} + \cdots + \sigma_r \mathrm{u}_r \mathrm{v}_r^{\ T}$$

When compressing the image, the sum is not performed to the very last SVs; the SVs with small enough values are dropped. (Remember that the SVs are ordered on the diagonal.) The closet matrix of rank *k* is obtained by truncating those sums after the first *k* terms:

$$A_k = \sigma_1 \mathrm{u}_1 \mathrm{v}_1^{\ T} + \sigma_2 \mathrm{u}_2 \mathrm{v}_2^{\ T} + \cdots + \sigma_k \mathrm{u}_k \mathrm{v}_k^{\ T}$$

The total storage for A_k will be k (m+n+1).

The integer *k* can be chosen confidently less then *n*, and the digital image corresponding to A_k still have very close the original image. However, chose the different *k* will have a different corresponding image and storage for it.

For typical choices of the k, the storage required for A_k will be less the 20 percentage.

3.4.2 Image Compression Measures

To measure the performance of the SVD image compression method, we can computer the compression factor and the quality of the compressed image. Image compression factor can be computed using the Compression ratio:

$$CR = m*n/ (k (m + n + 1))$$

To measure the quality between original image *A* and the compressed image A_k, the measurement of Mean Square Error (MSE).

$$MSE = \frac{1}{mn}\sum_{y=1}^{m}\sum_{x=1}^{n}(f_A(x,y) - f_{A_k}(x,y))$$

3.5 Application of the SVD: Compression and Pseudoinverse

3.5.1 Low rank approximation

One use of the SVD is to approximate a matrix by one of low rank. One way of looking at the product $U\Sigma V^T$ gives:

$$\begin{bmatrix} \vdots & & \vdots \\ \mathbf{u}_1 & \cdots & \mathbf{u}_m \\ \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \ddots \end{bmatrix} \begin{bmatrix} \cdots & \mathbf{v}_1 & \cdots \\ & \vdots & \\ \cdots & \mathbf{v}_n & \cdots \end{bmatrix} = \begin{bmatrix} \vdots & & \vdots \\ \mathbf{u}_1 & \cdots & \mathbf{u}_m \\ \vdots & & \vdots \end{bmatrix} \begin{bmatrix} \cdots & \sigma_1 \mathbf{v}_1 & \cdots \\ & \vdots & \\ \cdots & \sigma_n \mathbf{v}_n & \cdots \end{bmatrix}$$

Which multiples out in the column-times-row picture as

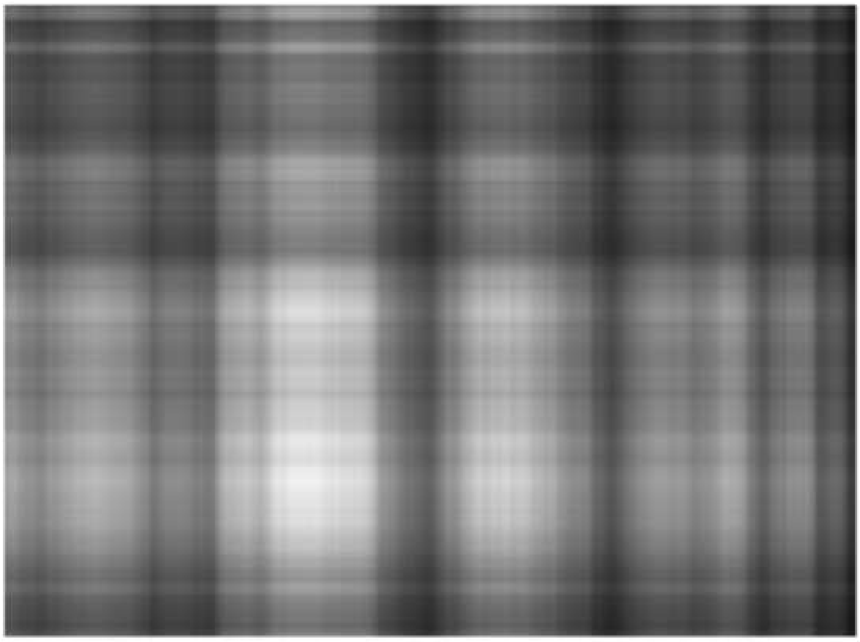
$$\sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \cdots + \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^T$$

Where r is the rank of A (the σ beyond this are all zero). This is a sum of rank one matrix. Now since the σ_i are in decreasing order, and the **u**'s and **v**'s are all unit vectors, these rank one matrices are written in decreasing order of "size" (at least in one way of measuring size of a matrix). So if we want a low-rank approximation to A we should just stop this sum after a few terms. The first term tells us about the single direction that gets magnified the most by the matrix A; the second tells us about the direction that gets magnified second most, and so on.

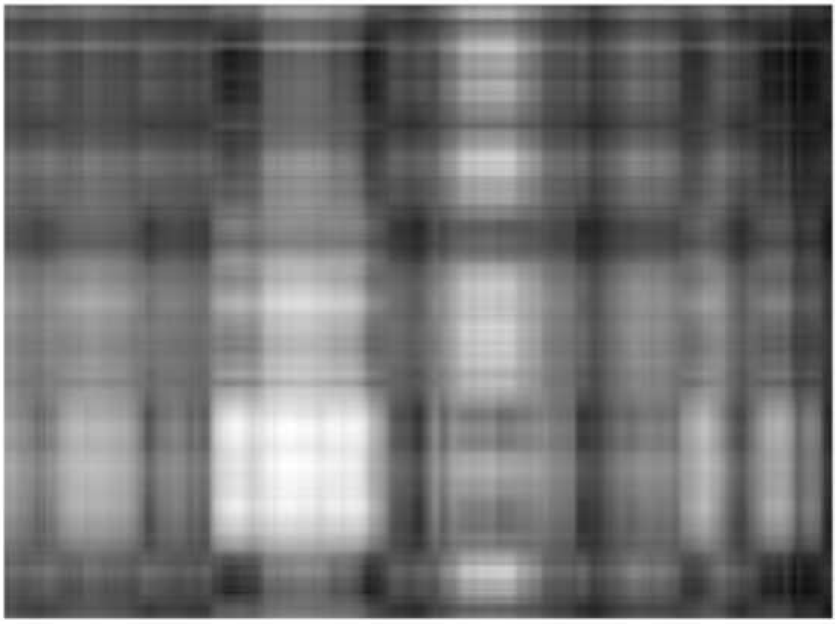
3.5.2 SVD image compression

A digitized picture is essentially just a big matrix of numbers. For instance, these could be the gray-levels in a black-and-white image, or the color levels in a color image. Let's say our image is 1000 x 2000 pixels. That requires 2 million numbers. But if the picture could be accurately approximated by, say, a ten term SVD decomposition we would only have to store 10 **u**'s (10000 numbers), 10 **v**'s (20000 numbers) and 10 σ's (10 numbers). Our storage costs drop to just over 30000, for a compression ratio of over 650: 1. Here is an example: (original image 480*640, 256 gray levels, and 307,200 bytes)

Mathematic to compute the singular value decomposition of the underlying matrix of numbers. Keep just one term—approximating the matrix with a rank one matrix, here is the result (480 + 640 bytes for the vectors, 4 bytes for the singular value (which will be a real number), total 1124 bytes for a compression ratio of over 273: 1)

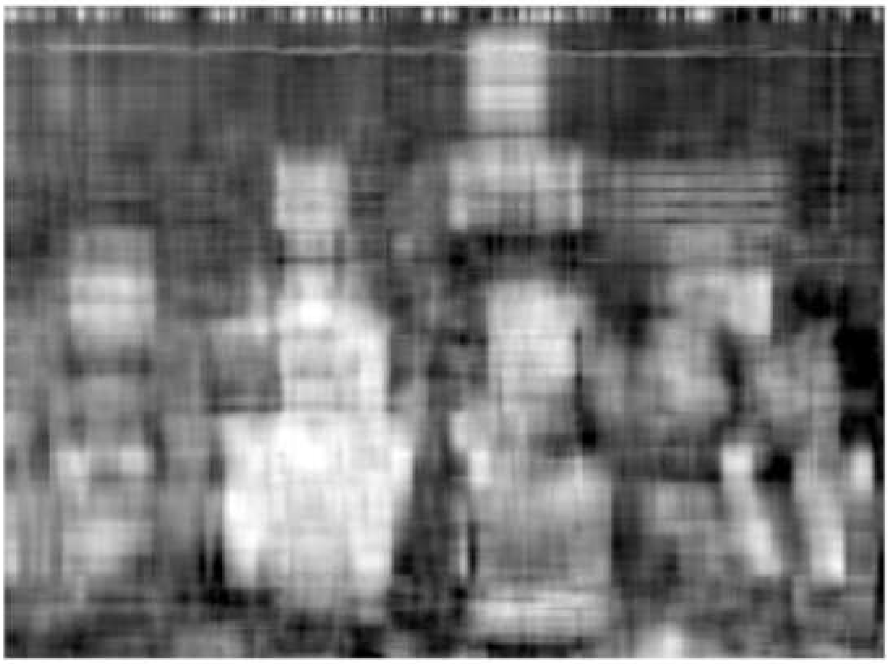


Can’t really see much, although you can see the light and dark patches where the children are wearing lighter-colored clothes. Let’s try two terms (compression ratio over 136: 1)



Still no good, really. It’s like looking through distortion glass. We can look at the error (true picture – approximation) and it looks like the image on the right. In the difference, you can clearly make out the children even though they’re kind of ghostly (appropriate for the Halloween picture, no?). So we’ll up the number of terms we use

Trying ten terms (compression ratio now about 27: 1) gives us



Now the picture is clearly discernable, though still quite distorted. The error picture is getting harder to make out, but still clearly shows the kids’ outlines. Next, we try 30 terms (compression ratio of about 9: 1)

The error picture is now almost all black, so is very hard to see. The picture looks pretty good a little blocky as if taken by a low-resolution camera, but clearly discernable. Finally, we try 90 terms (compression ratio 3: 1)

Now we're nearly perfect. Compression is small (only 3: 1) but we get a very good picture and still some substantial savings. Note that the bar of light color across the top is not an error or artifact of the technique go back and check that it was in the original picture.

Now while this is a nifty idea, it is not as good a picture compression scheme as some other techniques out there. The discrete cosine transform (DCT) is related to the fast Fourier transform (FFT) so can be done fast, is used for JPEG image files and it can achieve compression ratios of around 20: 1 with very good image quality. Wavelets are used in other image formats (JPEG2000). Some wavelet schemes can produce compression of upward of 100: 1 without perceptible image distortion.

3.5.3 Web searching

Search engines like Google use enormous matrices of cross-references—which pages link to which other pages, and what words are on each page. When you do a Google search, the higher ranks usually go to pages with your key words that have lots of links to them. But there are billions of pages out there, and storing a billion by billion matrix is trouble, not to mention searching through it.

Here is where SVD shines. In searching, you really only care about the main directions that the Web is taking. So the first few singular values create a very good approximation for the enormous matrix, can be searched relatively quickly (just a few billion entires) and provide compression ratios of millions to one. The proof is in the pudding—Google works.

3.5.4 The pseudoinverse

In an entirely different direction, the SVD can give us the "best" we can do toward inverting an arbitrary, even non-square, matrix. Note that A sends the row space to the column space in an invertible fashion, while the nullspace gets sent to 0. The best we could hope for is to send the column space back to where it came from in the row space, and perhaps send the left nullspace to 0. But that is easy with the SVD!

Chapter 4

IMAGE COMPRESSION USING BLOCK TRUNCATION CODING

4.1 Introduction

The amount of image data grows day by day. Large storage and bandwidth are needed to store and transmit the Images, which is quite costly. Hence methods to compress the image data are essentially now-a-days. The image Compression techniques are categorized into two main classifications namely Lossy compression techniques and Lossless compression techniques [1]. Lossless compression ratio gives good quality of compressed images, but yields only less compression whereas the lossy compression techniques [2] lead to loss of data with higher compression ratio. JPEG [1] and Block Truncation Coding [18] is a lossy image compression techniques .It is a simple technique which involves less computational complexity. BTC is a recent technique used for compression of monochrome image data. It is one-bit adaptive moment-preserving quantizer that preserves certain statistical moments of small blocks of the input image in the quantized output. The original algorithm of BTC preserves the standard mean and the standard deviation [20]. The statistical overheads Mean and the Standard deviation are to be coded as part of the block. The truncated block of the BTC is the one-bit output of the quantizer for every pixel in the block.

4.2 BTC ALGORITHM

Block Truncation Coding (BTC) is a well-known compression scheme proposed in 1979 for the grayscale images. It was also called the moment-preserving block truncation [18]-[19] because it preserves the first and second moments of each image block. The BTC algorithm involves the following steps:

• **Step1**: The given image is divided into non overlapping rectangular regions. For the sake of simplicity the blocks were let to be square regions of size m x m.

• **Step 2**: For a two level (1 bit) quantizer, the idea is to select two luminance values to represent each pixel in the block. These values are the mean *x* and standard deviation σ.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (1)$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x}_i)^2} \qquad (2)$$

Where xi represents the i^th pixel value of the image block and n is the total number of pixels in that block.

• **Step3**: The two values *x* and σ are termed as quantizers of BTC. Taking $\bar{x}$ as the threshold value a two-level bit plane is obtained by comparing each pixel value x_i with the threshold. A binary block, denoted by B, is also used to represent the pixels. We can use "1" to represent a pixel whose gray level is greater than or equal to $\bar{x}$ and "0" to represent a pixel whose gray level is less than $\bar{x}$

$$B = \begin{cases} 1 & x_i \geq \bar{x} \\ 0 & x_i \prec \bar{x} \end{cases} \qquad (3)$$

By this process each block is reduced to a bit plane. For example, a block of 4 x 4 pixels will give a 32 bit compressed data, amounting to 2 bit per pixel (bpp).

· **Step 4**: In the decoder an image block is reconstructed by replacing '1's in the bit plane with H and the '0's with L, which are given by:

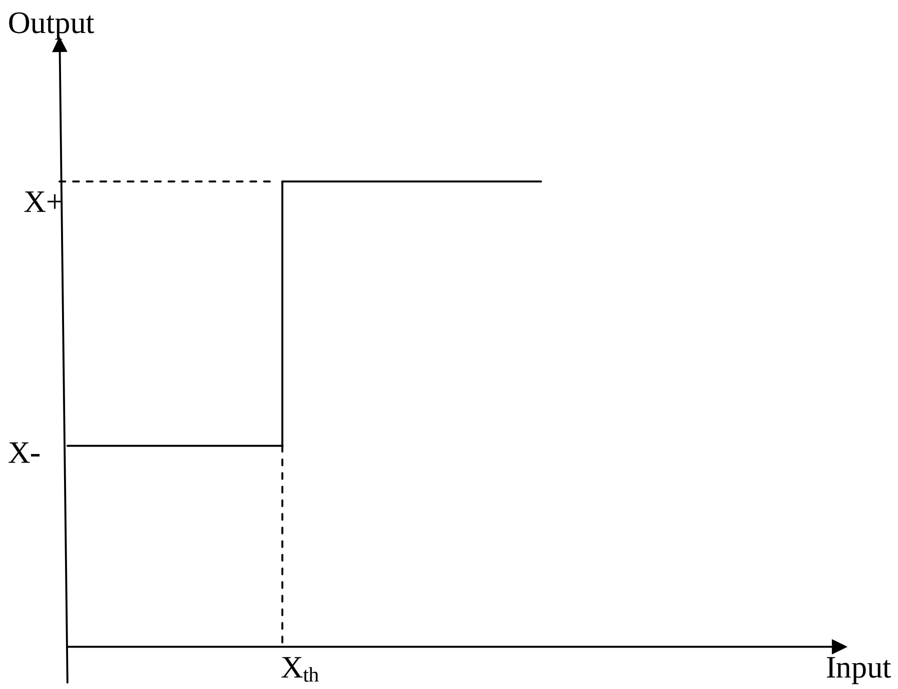
$$H = \bar{x} + \sigma\sqrt{\frac{p}{q}} \qquad (4)$$

$$L = \bar{x} - \sigma\sqrt{\frac{q}{p}} \qquad (5)$$

Where p and q are the number of 0's and 1's in the compressed bit plane respectively.

4.2.1Two Level (binary) Quantizer

The rest of the information is in the Mean and the Standard deviation (SD) of



4.3Advantages of BTC

•Small complexity (faster than TC).

•Preserving edges.

•Each block can be compressed separately according to its variance.

•Fixed and Adaptive bit-allocation optional.

4.4 BTC Encoding

•Assume a 512x512 image with 256 gray levels.

•The threshold will be the mean value (x_{ave}).

•For each block we transmit bit-level matrix, x_{Sd} and x_{ave}.

•The levels X^+ and X^- can be determined by setting up the expressions that equate (preserve) the moments before and after quantization.

4.5 BTC FOR COLOR IMAGE

Liquid-crystal displays (LCD) panels have rapidly occupied the display market share because of their attractive characteristics. However, motion blur of fast moving objects caused by the slow respond time of LCD is an inevitable problem. To reduce the motion blur, the overdrive technique has been proposed [21]. However, this technique requires the frame buffer to store a previous frame as shown in Fig. 1. In general, a large size of the frame buffer increases the cost of products. Therefore, simple image compression algorithms such as block truncation coding (BTC) [22] are employed to reduce the frame buffer. Since the conventional BTC method is a nonoverlapping since the conventional BTC method is a nonoverlapping block compression algorithm based on a two level quantizer, one bitmap containing the quantization level of each pixel and two representative values for each color component are generated from each block. In order to increase the compression, vector quantization BTC (VQBTC) algorithm for color images is proposed in [23]. In VQBTC, all color pixels are represented by RGB vectors and classified into two classes by using the vector quantization method in [24]. Although the compression ratio required for LCD overdrive can be achieved by VQ-BTC, the blocking artifacts at block boundaries are inevitable because VQBTC is a block-based method. To solve this problem, we propose a novel color compression method based on VQBTC. The proposed method first reduces the number of bits used for encoding the representative vectors. Then, using these remaining bits, the representative vectors are precisely refined to preserve more edge information.

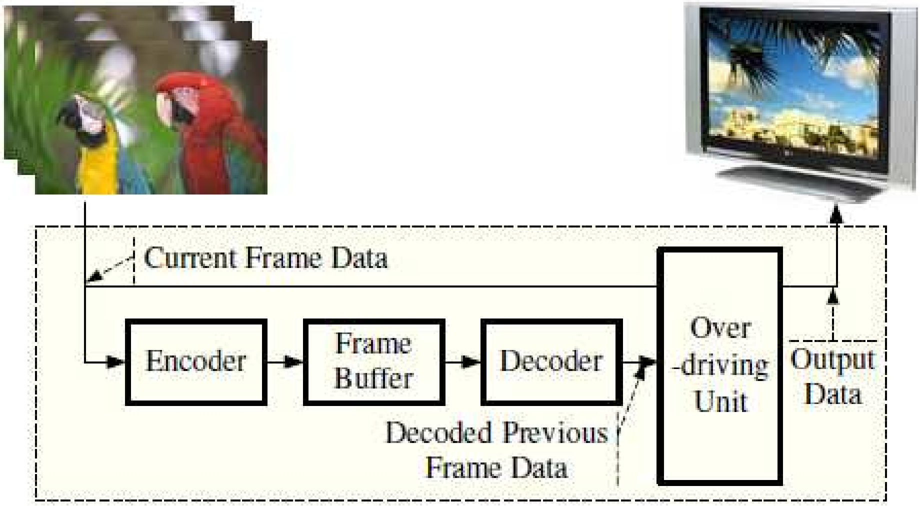


FIGURE 4.1 Schematic diagrams of simple overdriving techniques

4.5.1. ALGORITHM

In the conventional VQ-BTC algorithm, a color image is divided into non-overlapping blocks which are individually coded by a two-level adaptive vector quantization. Two representative vectors in VQ-BTC consisting of R, G, and B components are defined as

$$V_1 = \begin{bmatrix} R_1 \\ G_1 \\ B_1 \end{bmatrix}, \quad V_2 = \begin{bmatrix} R_2 \\ G_2 \\ B_2 \end{bmatrix}.$$

In case of the 4x4 block which consists of 8-bit resolution color components, the total number of bits required for encoding each block is equal to 64 since 48 bits for two representative vectors and 16 bits for the bitmap are required. As a consequence, this VQ-BTC method achieves the 1/6 compression ratio. However, annoying blocking artifacts occurs at the block boundary. In the proposed algorithm, we introduce a new method that can effectively alleviate the blocking artifacts. The method consists of two stages. In the former, the number of bits used for encoding the representative vectors is decreased. These remaining bits can be used for preserving edge Information in the latter stage.

CHAPTER 5

WAVELET TRANSFORMS

This chapter discusses the fundamental basics of wavelet. It also contains a brief Overview of the types of wavelet transforms used in this work.

5.1 Introduction to Wavelet

The concept of wavelet was hidden in the works of mathematicians even more than a Century ago. In 1873, Karl Weirstrass mathematically described how a family of functions can be constructed by superimposing scaled versions of a given basis function.

The term wavelet was originally used in the field of seismology to describe the disturbances that emanate and proceed outward from a sharp seismic impulse [25]. Wavelet means a "small wave". The smallness refers to the condition that the window function is of finite length (compactly supported) [26]. A wave is an oscillating function of time or space and is periodic. In contrast, wavelets are localized waves. They have their energy concentrated in time and are suited to analysis of transient signals. While Fourier Transform and STFT use waves to analyze signals, the Wavelet Transform uses wavelets of finite energy [25].

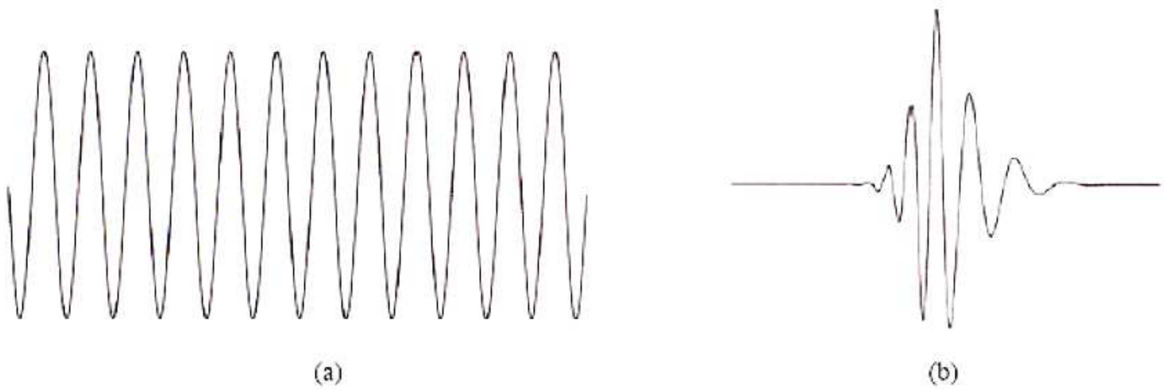


FIGURE 5.1 Difference between Wave and Wavelet (a) wave (b) wavelet.

In wavelet analysis the signal to be analyzed is multiplied with a wavelet function and then the transform is computed for each segment generated. The Wavelet Transform, at high frequencies, gives good time resolution and poor frequency resolution, while at low frequencies; the Wavelet Transform gives good frequency resolution and poor time resolution. An arbitrary signal can be analyzed in terms of scaling and translation of a single mother wavelet function (basis). Wavelets allow both time and frequency analysis of signals simultaneously because of the fact that the energy of wavelets is concentrated in time and still possesses the wave-like (periodic) characteristics. As a result, wavelet representation provides a versatile mathematical tool to analyze transient, time-variant (non stationary) signals that are not statistically predictable especially at the region of discontinuities-a feature that is typical of images having discontinuities at the edges [27].

5.2 Mathematical Representation of Wavelet

Wavelets are functions generated from one single function (basis function) called the prototype or mother wavelet by dilations (scaling) and translations (shifts) in time (frequency) domain. If the mother wavelet is denoted by $\Psi(t)$, the other wavelets $\Psi_{a,b}$ (t) can be represented as

$$\psi_{a,b}(t) = (1 * \psi((t-b)/a))/\sqrt{|a|} \quad (1)$$

Where a and b are two arbitrary real numbers. The variables 'a' and 'b' represent the parameters for dilations and translations respectively in the time axis.

The mother wavelet can be essentially represented as

$$\Psi(t) = \Psi_{0,1}(t) \quad (2)$$

For any arbitrary $a \neq 1$ and $b = 0$, we can derive that

$$\psi_{a,0}(t) = (1 * \psi(t/a))\sqrt{|a|} \quad \text{... (3)}$$

As shown above, $\Psi_{a,0}$ (t) is nothing but a time-scaled (by a) and amplitude-scaled version of the mother wavelet function Ψ(t). The parameter 'a' causes contraction of Ψ(t), in the time axis when a < 1 and expansion or stretching when a > 1. That's why the parameter 'a' is called the dilation (scaling) parameter. For a < 0, the function $\Psi_{a,b}$ (t) results in time reversal with dilation. Mathematically, we can substitute 't' in equation by (t – b) to cause a translation or shift in the time axis resulting in the wavelet function $\Psi_{a,b}$ (t) as shown in equation 1.

The function $\Psi_{a,b}$ (t) is a shift of $\Psi_{a,0}$ (t) in right along the time axis by an amount b when b > 0 whereas it is a shift in left along the time axis by an amount b when b < 0. That's why the variable b represents the translation in time (shift in frequency) domain [25].

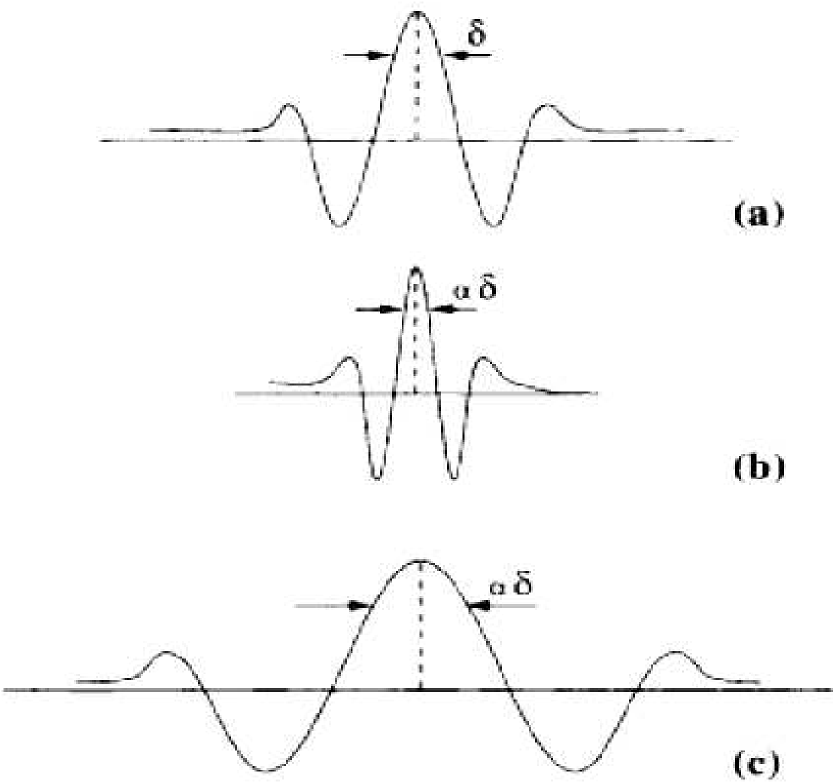


FIGURE 5.2 a) Mother wavelet, Ψ(t), ;b) Ψ(t/α); o<α<1;c) Ψ(t/α); α >1

5.3 Translation and Scale in WT

TRANSLATION is related to the location of the window, as the window is shifted through the signal. It corresponds to time information in the transform domain [26]. It simply means delaying (or hastening) its onset.

Mathematically, delaying a function $\Psi(t)$, by k is represented by $\Psi(t - k)$ [28].

Wavelet function $\Psi(t)$,   shifted Wavelet function $\Psi(t-k)$,

FIGURE 5.3 Translations

However, we do not have a frequency parameter, (as in STFT). Instead, we have scale Parameter which is defined as inverse of frequency.

SCALE is a parameter in the WAVELET analysis that is quite similar to the scale used in maps. In case of maps, high scales correspond to a non-detailed global view and low scales correspond to a detailed view.

Similarly in case of frequency, low frequencies (high scales) correspond to a global information of a signal (that usually spans the entire signal), whereas high frequencies (low scales) correspond to a detailed information of a hidden pattern in the signal (that usually lasts for a relatively short time).Scaling, as a mathematical operation, either dilates or compresses a signal. Larger scales correspond to dilated (or stretched out) signals and small scales correspond to compressed signals.

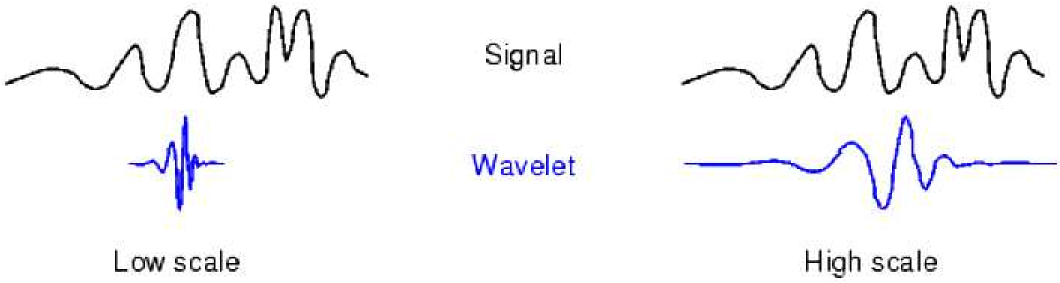


FIGURE 5.4 Scaling

If f (t) is a given function, then f (st) corresponds to a contracted (compressed) version of f (t) if s>1 and to an expanded (dilated) version of f (t) if s<1.However, in WT, the Scaling term is used in the denominator and hence s>1 dilates the signal and s<1 Compresses the signal [26].

5.4 Multi-Resolution Analysis in WT

MULTI-RESOLUTION ANALYSIS, as the name itself suggests, analyzes the signal at different frequencies with different resolutions. Here, every spectral component is not resolved equally as was the case in the STFT.

MRA provides an alternative approach to analyze any signal, although the TIME and FREQUENCY resolution problems are results of a phenomenon (the Heisenberg's Uncertainty Principle) and exist regardless of the transform used. MRA is designed to give good time resolution and poor frequency resolution at high Frequencies and good frequency resolution and poor time resolution at low frequencies. This approach makes sense especially when the signal at hand has high frequency Components for short durations and low frequency components for long durations. In Practical applications as well, we face such problems [26, 31].

5.5 Properties of Wavelet

• 'Regularity' defined as: if r is an integer and a function is r-time continuously Differentiable at x_0, then the regularity is r. If r is not an integer, let *n* be the integer Such that n <r<n+1, then function has a regularity of r in x_0 if its derivative of order *n*

resembles (x− x_0)$^{r-n}$ locally around x_0. This property is useful for getting nice features, Such as smoothness, of the reconstructed signals.

• The support of a function is the smallest space-set (or time-set) outside of which Function is identically zero.

• The number of vanishing moments of wavelets determines the order of the polynomial that can be approximated and is useful for compression purposes.

• The wavelet symmetry relates to the symmetry of the filters and helps to avoid dephasing in image processing. Among the orthogonal families, the Haar wavelet is the only symmetric wavelet. For biorthogonal wavelets it is possible to synthesize wavelet Functions and scaling functions that are symmetric or antisymmetric [25].

5.6 Types of Wavelet Transforms

There are mainly two types of Wavelet Transforms-

• Continuous Wavelet Transformation (CWT)

• Discrete Wavelet Transformation (DWT)

DWT stands for Discrete Wavelet Transformation. It is the Transformation of sampled data, e.g. transformation of values in an array, into wavelet coefficients.

IDWT is Inverse Discrete Wavelet Transformation: procedure converts wavelet coefficients into the original sampled data.

Here the case of square images has been considered. Let us take an N by N image.

5.6.1 Decomposition Process

To start with, the image is high and low-pass filtered along the rows and the results of each filter are down- sampled by two. Those two sub-signals correspond to the high and low frequency components along the rows and are each of size N by N/2. Then each of these sub-signals is again high and low-pass filtered, along the column data. The results are again down-sampled by two.

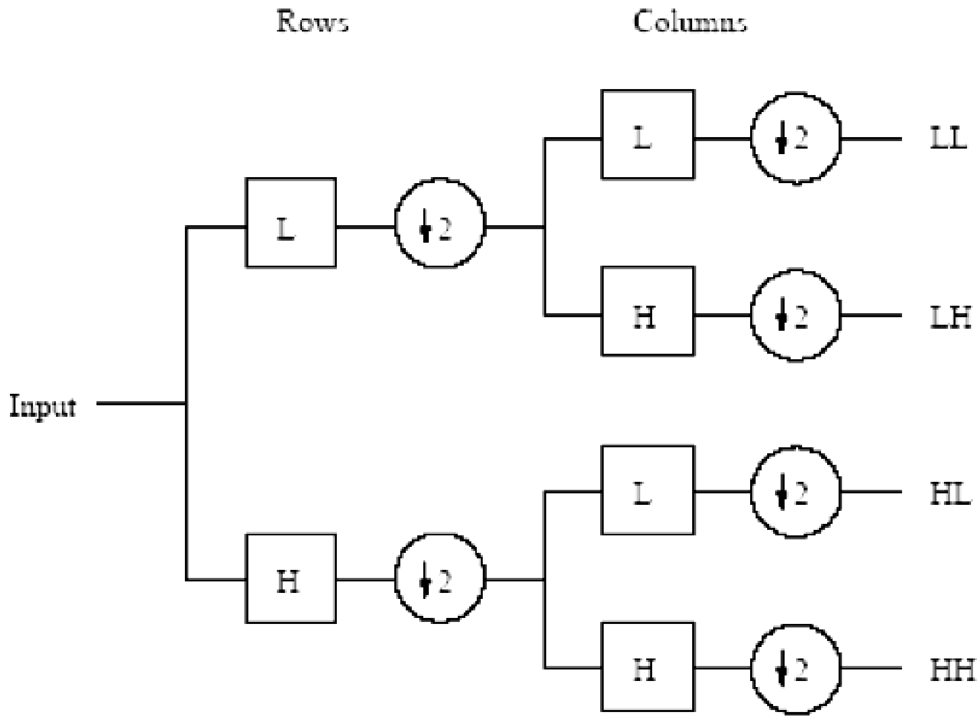


FIGURE 5.5 One decomposition step of the two dimensional image.

As a result the original data is split into four sub-images each of size N/2 by N/2 Containing information from different frequency components. Figure 5.5 shows the level one decomposition step of the two dimensional grayscale images. Figure 5.6 shows the four sub bands in the typical arrangement.

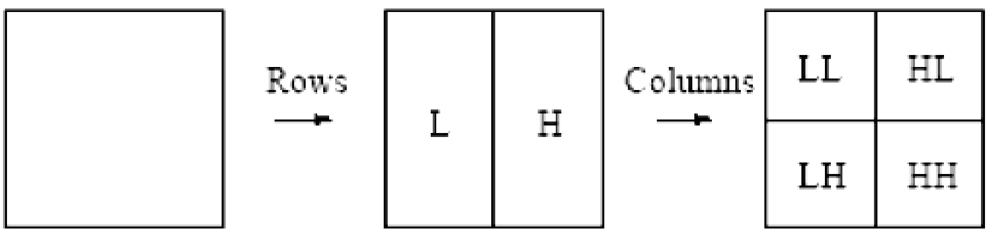


FIGURE 5.6 One DWT decomposition step

The LL subband is the result of low-pass filtering both the rows and columns and it contains a rough description of the image as such. Hence, the LL subband is also called the approximation subband. The HH subband is high-pass filtered in both directions and contains the high-frequency components along the diagonals as well. The HL and LH images are the result of low-pass filtering in one direction and high-pass filtering in another direction. LH contains mostly the vertical detail information that corresponds to horizontal edges. HL represents the horizontal detail information from the vertical edges. All three subbands HL, LH and HH are called the detail subbands, because they add the high-frequency detail to the approximation image.

5.6.2 Composition Process

The inverse process is shown in Figure 5.7. The information from the four sub-images is up-sampled and then filtered with the corresponding inverse filters along the columns. The two results that belong together are added and then again up-sampled and filtered with the corresponding inverse filters. The result of the last step is added together in order to get the original image again. Note that there is no loss of information when the image is decomposed and then composed again at full precision.

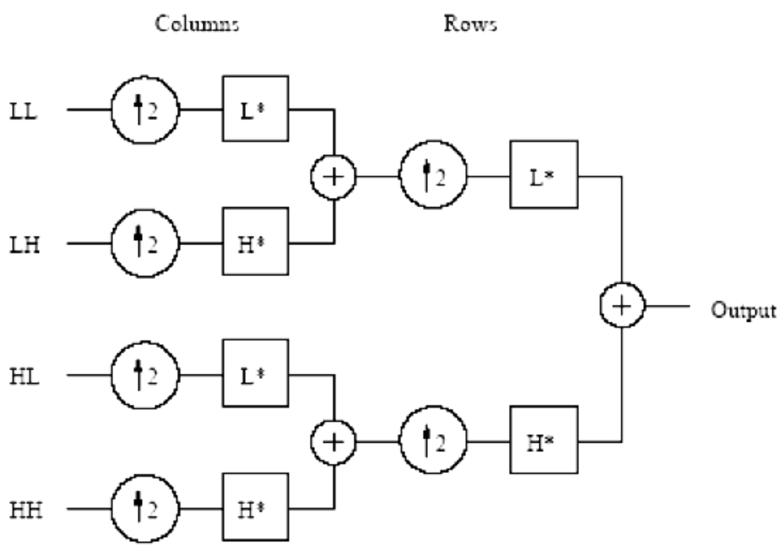


FIGURE 5.7 One composition step of the four sub images

With DWT we can decompose an image more than once. Decomposition can be continued until the signal has been entirely decomposed or can be stopped before by the application at hand. Mostly two ways of decomposition are used. They are:

i.) Pyramidal decomposition

ii.) Packet decomposition

5.6.3 Pyramidal Decomposition

Pyramidal decomposition is the simplest and most common form of decomposition used.For the pyramidal decomposition we only apply further decompositions to the LL subband. Figure 5.8 shows a systematic diagram of three decomposition steps. At each Level the detail subbands are the final results and only the approximation subband is further decomposed [28].

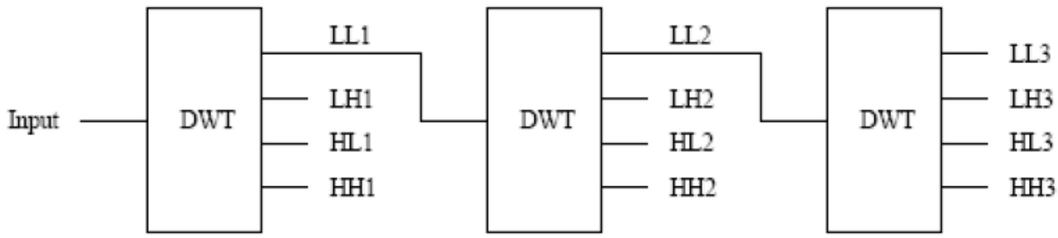


FIGURE 5.8 Three decomposition steps of an image using Pyramidal Decomposition

Figure 5.9 shows the pyramidal structure that result from this decomposition. At the lowest level there is one approximation subband and there are a total of nine detail subbands at the different levels. After L decompositions, a total of D (L) = 3 * L + 1 subbands are obtained.

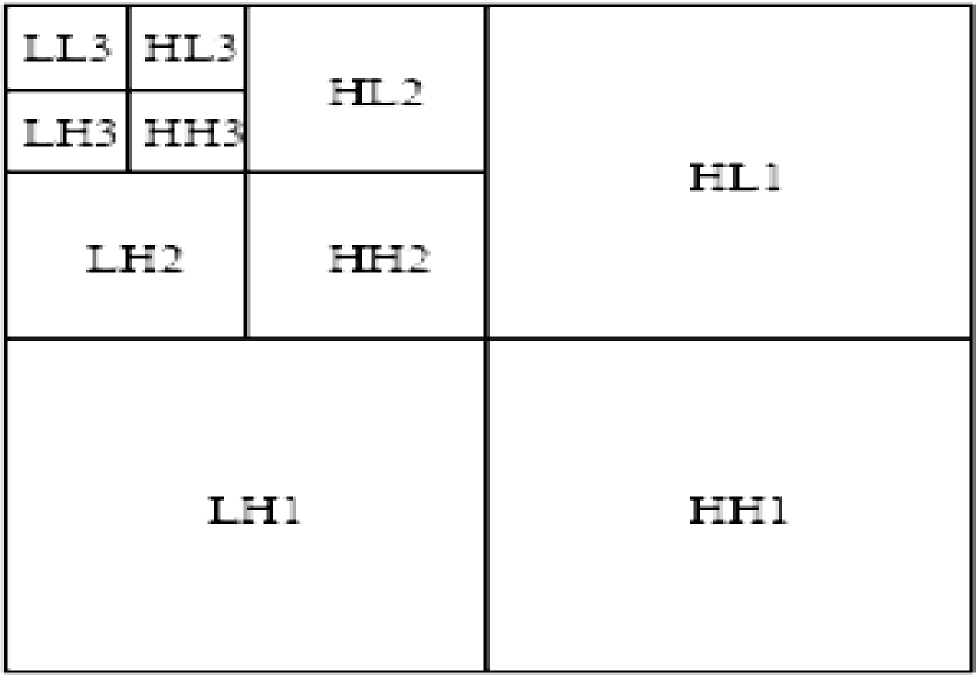


FIGURE 5.9 Pyramid after three decomposition steps

Figure 5.10 is an example of this decomposition process. It shows the “Lena" image after one, two and three pyramidal decomposition steps [30].

FIGURE 5.10 Pyramidal decomposition of Lena image (1, 2 and 3 times)

5.6.4 Wavelet Packet Decomposition

For the wavelet packet decomposition, the decomposition is not limited to the approximation subband only but a further wavelet decomposition of all subbands on all levels is considered. In figure 5.11, the system diagram for a complete two level wavelet packet decomposition has been shown.

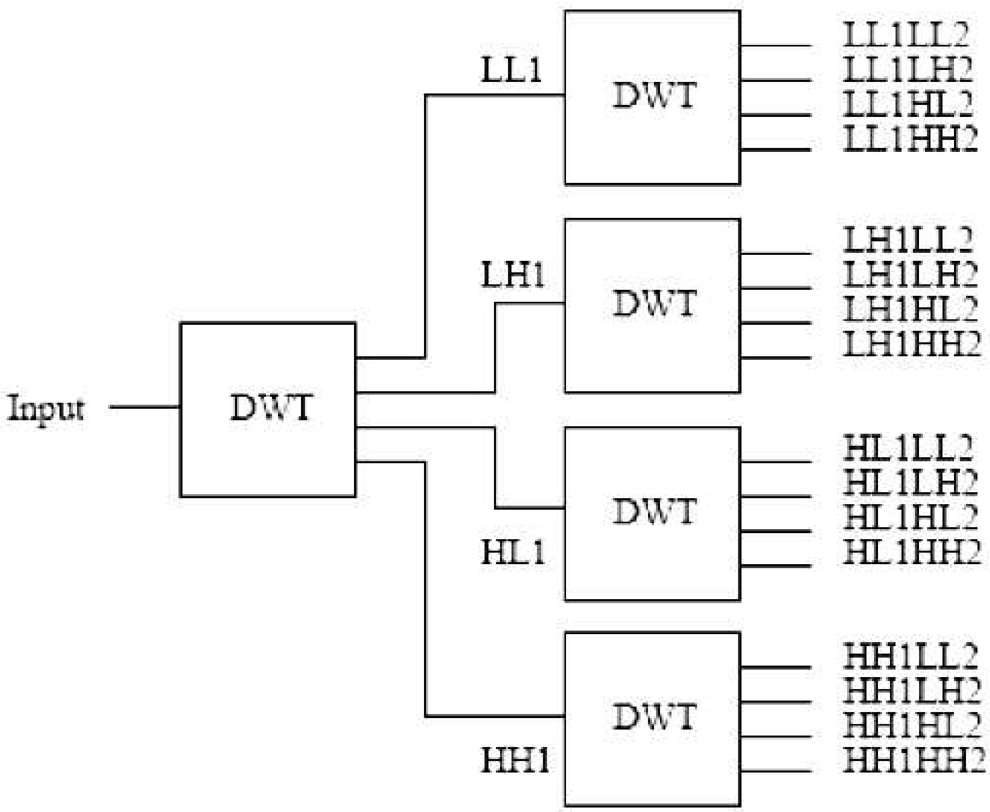


FIGURE 5.11 Two complete decomposition steps using wavelet packet decomposition

In figure 5.12, the resulting subband structure is on display. Again the simple decomposition step from 5.5 is used as a basic building block. The composition step is equivalent to the pyramidal case. All four subbands on one level are used as input for the inverse transformation and a resultant in the subband on the higher level is obtained. This process is repeated again and again until the original image is reproduced.

LL1LL2	LL1HL2	HL1LL2	HL1HL2
LL1LH2	LL1HH2	HL1LH2	HL1HH2
LH1LL2	LH1HL2	HH1LL2	HH1HL2
LH1LH2	LH1HH2	HH1LH2	HH1HH2

Figure 5.12 Subband structure after two level packet decomposition.

The discrete wavelet transform is very efficient from the computational point of view. Its only drawback is that it is not translation invariant. Translations of the original signal lead to different wavelet coefficients. In order to overcome this and to get more complete characteristic of the analyzed signal the undecimated wavelet transform was proposed.
The general idea behind it is that it doesn't decimate the signal. Thus it produces more precise information for the frequency localization. From the computational point of view the undecimated wavelet transform has larger storage space requirements and involves more computations [28].

5.7 Undecimated Wavelet Transform

UDWT is based on the idea of no decimation. It applies the wavelet transform and omits both down-sampling in the forward and up-sampling in the inverse transform. More precisely, it applies the transform at each point of the image and saves the detail coefficients and uses the low-frequency coefficients for the next level. The size of the coefficients array does not diminish from level to level. By using all coefficients at each level, we get very well allocated high-frequency information. From level to level there is very small step in the width of the scaling filter - instead of 8 pixels at the third level of

DWT; here its width is 5 pixels. Generally, the step is not a power of 2 but a sum with 2. This property is good for noise removal because the noise is usually spread over small number of neighboring pixels. With this transform the number of pixels involved in computing a given coefficient grows slower and so the relation between the frequency and spatial information is more precise. In the ideal case, this means removal of the noise only at the places that it really exists, without affecting the neighboring pixels. It gives the best results in terms of visual quality (less blurring for larger noise removal) [29].

5.8 Wavelet Families

There are a number of basic functions that can be used as the mother wavelet for Wavelet Transformation. Since the mother wavelet produces all wavelet functions used in the Transformation through translation and scaling, it determines the characteristics of the resulting Wavelet Transform. Therefore, the details of the particular application should be taken into account and the appropriate mother wavelet should be chosen in order to use the Wavelet Transform effectively.

Figure 5.13 illustrates some of the commonly used wavelet functions. Haar wavelet is one of the oldest and simplest wavelet. Therefore, any discussion of wavelets starts with the Haar wavelet. Daubechies wavelets are the most popular wavelets. They represent the foundations of wavelet signal processing and are used in numerous applications. These are also called Maxflat wavelets as their frequency responses have maximum flatness at frequencies 0 and R. This is a very desirable property in some applications. The Haar, Daubechies, Symlets and Coiflets are compactly supported orthogonal wavelets. These wavelets along with Meyer wavelets are capable of perfect reconstruction. The Meyer, Morlet and Mexican Hat wavelets are symmetric in shape. The wavelets are chosen based on their shape and their ability to analyze the signal in a particular application [28].

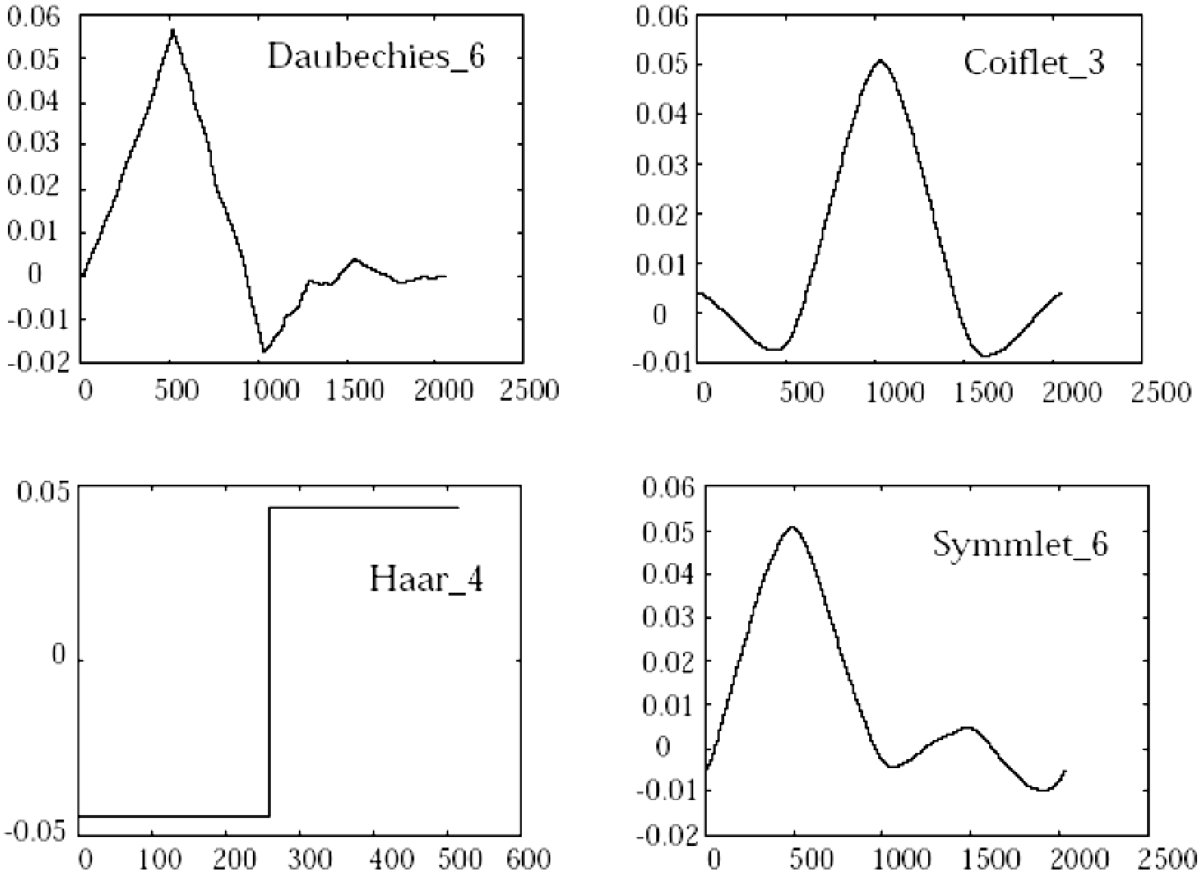


Figure 5.13 several different families of wavelets

CHAPTER 6

IMAGE COMPRESSION USING PYRAMID DECOMPOSITION

6.1 Introduction

An image may be represented by its Fourier transform, with operations applied to the transform coefficients rather than to the original pixel values. This is appropriate for some data compression and image enhancement tasks, but inappropriate for others. The transform representation is particularly unsuited for machine vision and computer graphics, where the spatial location of pattern elements is critical.

Recently there has been a great deal of interest in representations that retain spatial localization as well as localization in the spatial—frequency domain. This is achieved by decomposing the image into a set of spatial frequency band pass component images. Individual samples of a component image represent image pattern information that is appropriately localized, while the band passed image as a whole represents information about a particular fineness of detail or scale. There is evidence that the human visual system uses such a representation, and multiresolution schemes are becoming increasingly popular in machine vision and in image processing in general.

The importance of analyzing images at many scales arises from the nature of images themselves. Scenes in the world contain objects of many sizes, and these objects contain features of many sizes. Moreover, objects can be at various distances from the viewer. As a result, any analysis procedure that is applied only at a single scale may miss information at other scales. The solution is to carry out analyses at all scales simultaneously. Convolution is the basic operation of most image analysis systems, and convolution with large weighting functions is a notoriously expensive computation. In a multiresolution system one wishes to perform convolutions with kernels of many sizes, ranging from very small to very large. And the computational problems appear forbidding. Therefore one of the main problems in working with multiresolution representations is to develop fast and efficient techniques. Members of the Advanced Image Processing. Research Group have been actively involved in the development of multiresolution techniques for some time. Most of the work revolves around a representation known as a "pyramid," which is versatile, convenient, and efficient to use. We have applied pyramid-based methods to some fundamental problems in image analysis, data compression, and image manipulation.

6.2 Image pyramids

The task of detecting a target pattern that may appear at any scale can be approached in several ways. Two of these, which involve only simple convolutions, are illustrated in Fig. 1.

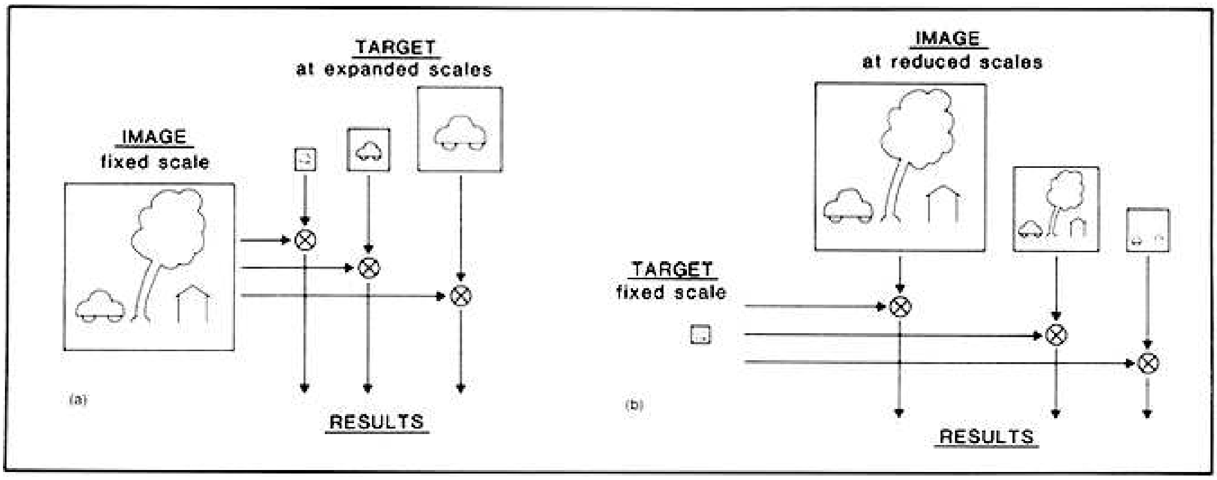


FIGURE6.1. Two methods of searching for a target pattern over many scales. In the first approach, (a), copies of the target pattern are constructed at several expanded scales, and each is convolved with the original image. In the second approach, (b), a single copy of the target is convolved with copies of the image reduced in scale. The target should be just large enough to resolve critical details the two approaches should give equivalent results, but the second is more efficient by the fourth power of the scale factor (image convolutions are represented by 'O').

Several copies of the pattern can be constructed at increasing scales, and then each is convolved with the image. Alternatively, a pattern of fixed size can be convolved with several copies of the image represented at correspondingly reduced resolutions. The two approaches yield equivalent results, provided critical information in the target pattern is adequately represented. However, the second approach is much more efficient: a given convolution with the target pattern expanded in scale by a factor s will requires more arithmetic operations than the corresponding convolution with the image reduced in scale by a factor of s. This can be substantial for scale factors in the range 2 to 32, a commonly used range in image analysis. The image pyramid is a data structure designed to support efficient scaled convolution through reduced image representation. It consists of a sequence of copies of an original image in which both sample density and resolution are decreased in regular steps. These reduced resolution levels of the pyramid are themselves obtained through a highly efficient iterative algorithm. The bottom, or zero level of the pyramid, G_0, is equal to the original image. This is low pass- filtered and subsampled by a factor of two to obtain the next pyramid level, G_1. G_1 is then filtered in the same way and subsampled to obtain G_2. Further repetitions of the filter/subsample steps generate the remaining pyramid levels. To be precise, the levels of the pyramid are obtained iteratively as follows. For $0 < l < N$:

$$G_{l(I,j)} \sum_{m} \sum_{n} w(m,n) G_{l-1}(2i+m,2j+n) \qquad (1)$$

However, it is convenient to refer to this process as a standard REDUCE operation, and simply write G_l = REDUCE [G_{l-1}]. We call the weighting function $w_{(m,n)}$ the "generating kernel." For reasons of computational efficiency this should be small and separable.

Fig6.2. Equivalent weighting functions. The process of constructing the Gaussian (lowpass) pyramid is equivalent to convolving the original image with a set of Gaussian-like weighting functions, then sub sampling, as shown in (a). The weighting functions double in size with each increase in 1. The corresponding functions for the Laplacian pyramid resemble the difference of two Gaussians, as shown in (b).

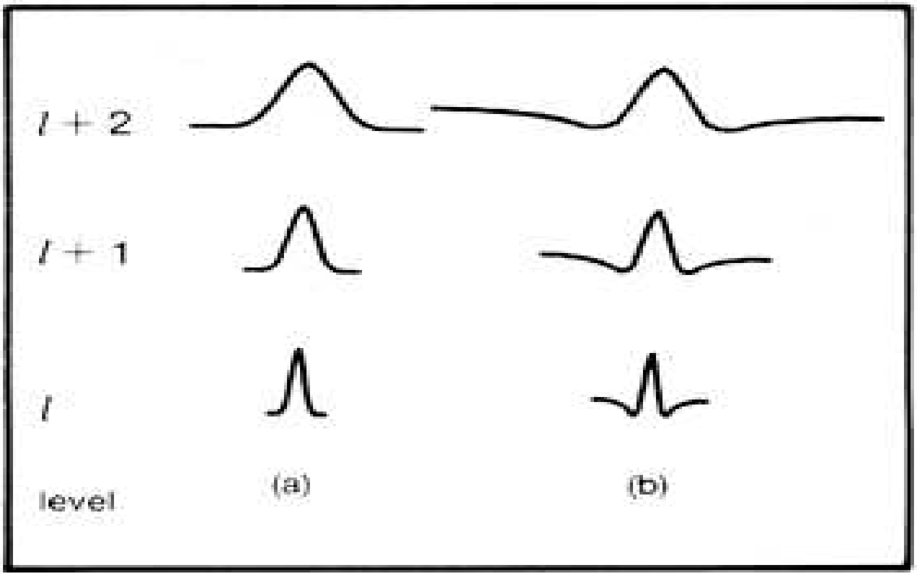


FIGURE 6.2 Laplacian pyramid

Pyramid construction is equivalent to convolving the original image with a set of Gaussian-like weighting functions. These "equivalent weighting functions" for three successive pyramid levels are shown in Fig6. 2a. Note that the functions double in width with each level. The convolution acts as a lowpass filter with the band limit reduced correspondingly by one octave with each level. Because of this resemblance to the Gaussian density function we refer to the pyramid of lowpass images as the "Gaussian pyramid."Bandpass, rather than lowpass, images Bandpass, rather than lowpass, images are required for many purposes. These may be obtained by subtracting each Gaussian (lowpass) pyramid level from the nextlower level in the pyramid.

6.3 Image analysis

Pyramid methods may be applied to analysis in several ways. Three of these will be outlined here. The first concerns pattern matching and has already been mentioned: to locate a particular target pattern that may occur at any scale within an image, the pattern is convolved with each level of the image pyramid. All levels of the pyramid combined contain just one third more nodes than there are pixels in the original image. Thus the cost of searching for a pattern at many scales is just one third more than that of searching the original image alone. The complexity of the patterns that may be found in this way is limited by the fact that not all image scales are represented in the pyramid. As defined here, pyramid levels differ in scale by powers of two, or by octave steps in the frequency domain. Power-of-two steps are adequate when the patterns to be located are simple, but complex patterns require a closer match between the scale of the pattern as defined in the target array, and the scale of the pattern as it appears in the image. Variants on the pyramid can easily be defined with square root-of-two and smaller steps. However, these not on]y have more levels, but many more samples, and the computational cost of image processing based on such pyramids is correspondingly increased[32]. A second class of operations concerns the estimation of integrated properties within local image regions. For example, a texture may often be characterized by local density or energy measures. Reliable estimates of image motion also require the integration of point estimates of displacement within regions of uniform motion. In such cases early analysis can often be formulated as a three-stage sequence of standard operations. First, an appropriate pattern is convolved with the image (or images, in the case of motion analysis). This selects a particular pattern attribute to be examined in the remaining two stages. Second, a nonlinear intensity transformation is performed on each sample value. Operations may include a simple threshold to detect the presence of the target pattern, a power function to be used in computing texture energy measures, or the product of corresponding samples in two images used in forming correlation measures for motion analysis. Finally the transformed sample values are integrated within local windows to obtain the desired local property measures.

Pattern scale is an important parameter of both the convolution and integration stages. Pyramid-based processing may be employed at each of these stages to facilitate scale selection and to support efficient computation. A flow diagram for this three stage analysis is given in Fig. 6. Analysis begins with the construction of the pyramid representation of the image. A feature pattern is then convolved with each level of the pyramid (Stage 1), and the resulting correlation values may be passed through a nonlinear intensity transformation (Stage2). Finally, each filtered and transformed image becomes the bottom level of a new Gaussian pyramid. Pyramid construction has the effect of integrating the input values within a set of Gaussian-like windows of many scales (Stage 3).

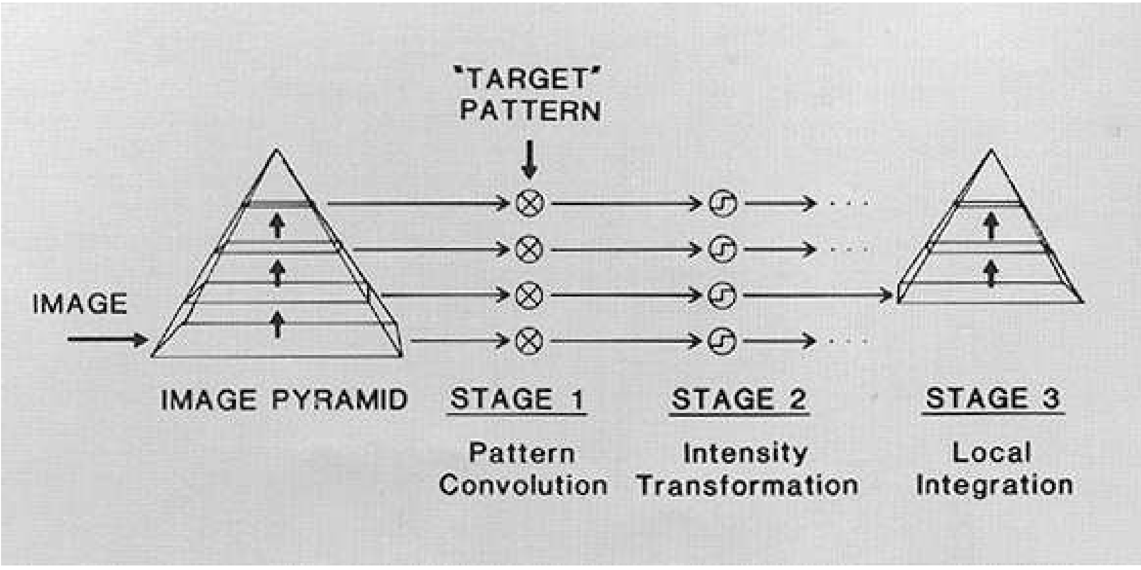


FIGURE.6.3 Efficient procedure for computing integrated image properties at many scales. Each level of the image pyramid is convolved with a pattern to enhance an elementary image characteristic, step 1. Sample values in the filtered image may then be passed through a nonlinear transformation, such as a threshold or power function, step 2. Finally, a new "integration" pyramid is built on each of the processed image pyramid levels, step 3. Node values then represent an average image characteristic integrated within a Gaussian-like window.

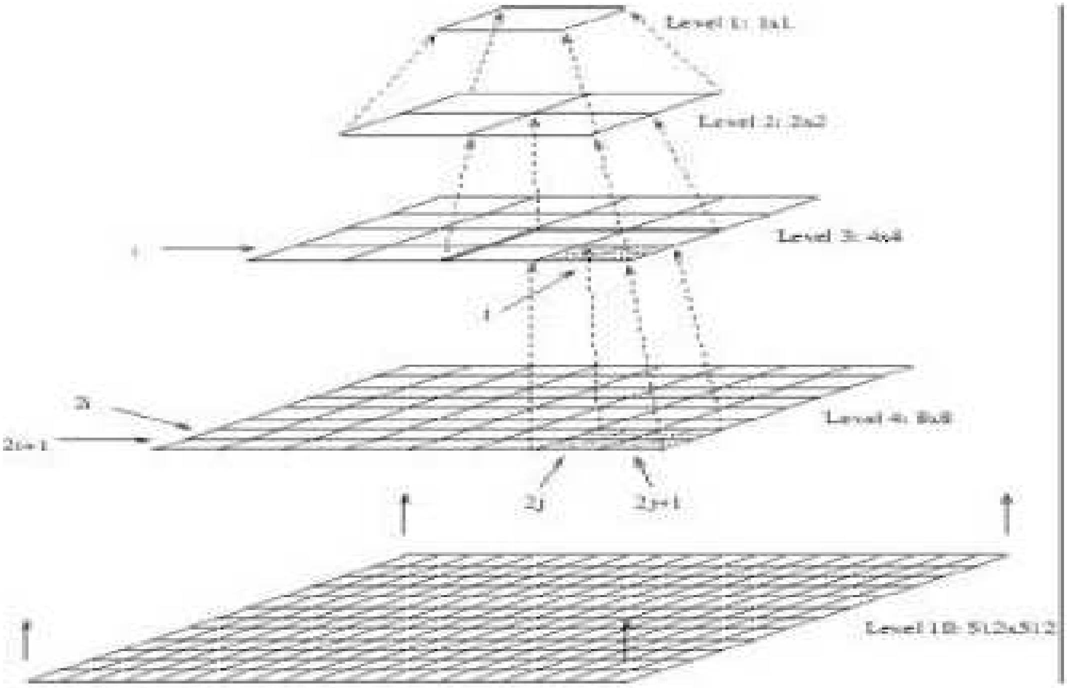


FIGURE 6.4 pyramid of image block

CHAPTER 7

IMAGE COMPRESSION USING EMBEDDED ZEROTREE WAVELET

7.1 Introduction

An embedded wavelet coding technique, known as Embedded Zerotree Wavelet (EZW) coding that effectively exploits the self-similarity between subbands and the fact that the high-frequency subbands mostly contain insignificant coefficients. First, we define the relationship between the subbands, based on the spatial locations and then define a data structure in the form of a hierarchical tree that includes spatially related coefficients across different subbands. The tree defines a parent-child relationship of DWT coefficients across subbands. The concept of a zerotree is introduced which identifies the parts of a tree that have all the DWT coefficients insignificant starting with a root. Since, DWT coefficients are generally insignificant at higher frequency subbands, occurrences of zerotrees are expected to be frequent and the zerotree roots can be encoded with a special symbol. The EZW algorithm is based on successive approximation quantization and this facilitates the embedding algorithm. Based on the concepts we are going to present in this lesson, the students should be able to design a complete wavelet coder, which can be suited to the desired bit-rate of the channel.

7.2 Embedded Coding

In embedded coding, the coded bits are ordered in accordance with their importance and all lower rate codes are provided at the beginning of the bit stream. Using an embedded code, the encoder can terminate the encoding process at any stage, so as to exactly satisfy the target bit-rate specified by the channel. To achieve this, the encoder can maintain a bit count and truncate the bit-stream, whenever the target bit rate is achieved. Although the embedded coding used in EZW is more general and sophisticated than the simple bit-plane coding, in spirit, it can be compared with the latter, where the encoding commences with the most significant bit plane and progressively continues with the next most significant bit-plane and so on. If target bit-rate is achieved before the less significant bit planes are added to the bit-stream, there will be reconstruction error at the receiver, but the "significance ordering" of the embedded bit stream helps in reducing the reconstruction error at the given target bit rate.

7.3 Relationship between subbands

In a hierarchical subband system, which we have already discussed in the previous lessons, every coefficient at a given scale can be related to a set of coefficients at the next finer scale of similar orientation. Only, the highest frequency subbands are exceptions, since there is no existence of finer scale beyond these. The coefficient at the coarser scale is called the parent and the coefficients at the next finer scale in similar orientation and same spatial location are the children. For a given parent, the set of all coefficients at all finer scales in similar orientation and spatial locations are called descendants. Similarly, for a given child, the set of coefficients at all coarser scales of similar orientation and same spatial location are called ancestors.

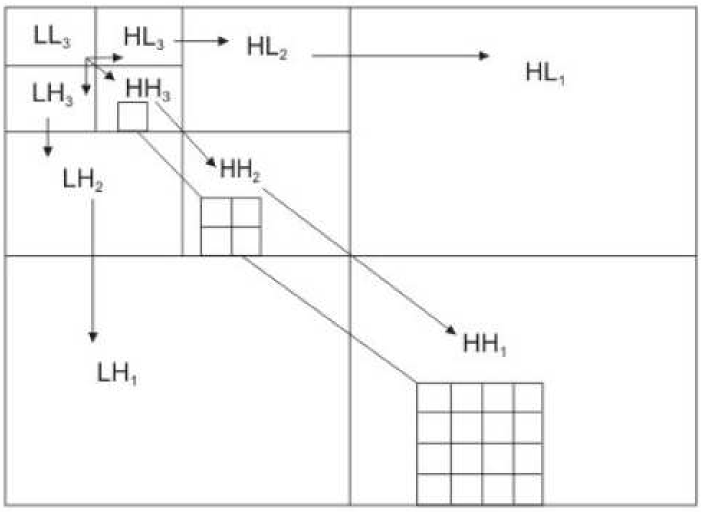


FIGURE 7.1 Parent –child dependencies of subbands

Fig.7.1 illustrates this concept, showing the descendants of a DWT coefficient existing in HH3 subband. Note that the coefficient under consideration has four children in HH2 subband, since HH2 subband has four times resolution as that of HH3. Likewise, the coefficient under consideration in HH3 subband has sixteen descendants in subband HH1, which in this case is a highest-resolution subband. For a coefficient in the LL subband, that exists only at the coarsest scale (in this case, the LL3), the hierarchical concept is slightly different. There, a coefficient in LL3 has three children – one in HL3, one in LH3 and one in HH3, all at the same spatial location. Thus, every coefficient at any subband other than LL3 must have its ultimate ancestor residing in the LL3 subband.

The relationship defined above best depicts the concept of space-frequency localization of wavelet transforms. If we form a descendant tree, starting with a coefficient in LL3 as a root node, the tree would span all coefficients at all higher frequency subbands at the same spatial location.

7.4 Significance of DWT coefficients

Before we can exploit the hierarchical subband relationship concept for efficient encoding of DWT coefficient, it is necessary to introduce a very simple concept of significance. We say that a DWT coefficient of magnitude lXl *is significant* with respect to a given threshold *T* if lXl>T and is *insignificant* otherwise. In the embedded coding adopted in EZW, the significance of DWT coefficients are first examined with the highest value of threshold in the first pass and then progressively, the threshold is decreased by a factor of 2 in subsequent passes. Before we start, all coefficients are assumed to be insignificant and progressively, more and more significant coefficients will be detected and by the end of the final pass, all coefficients would assume significance at some pass. At each pass, there is a significance map that tells about the significance of the DWT coefficients and this map requires to be encoded efficiently. The significance map has an entry of zero if the coefficient is insignificant with respect to a threshold and is one if significant. It should be noted that the significance is decided only with respect to the magnitude and hence the sign of the significance (positive or negative) must be included in the encoding process.

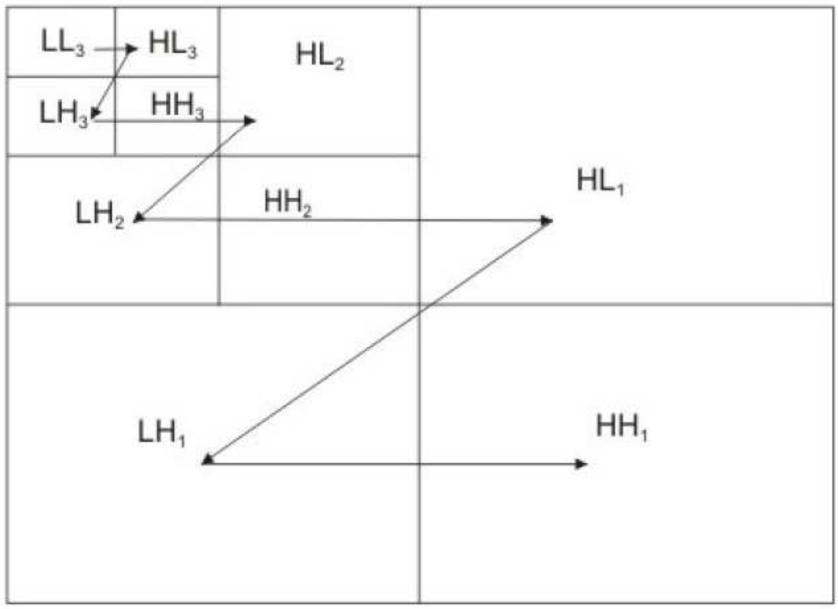


FIGURE 7.2 scanning order of sub bands

The coefficients are scanned for significance in a manner illustrated in fig.7.2 for 3-level subband decomposition. It starts with the lowest frequency subband, designated as LL_N where N s the number of levels. Following the scanning of all the coefficients in this subband, all the coefficients in subband HL_N are scanned. This is followed by HL_N and HH_N . Then the scanning proceeds to the next finer level N-1 in the same order HL, LH and HH. It continues till the highest frequency subbands are covered. This ensures that no child node is scanned before its parent.

7.5 **Encoding the Significance map**

We are now going to examine how to efficiently encode the significance map at any pass. For this, the hierarchical relationship of coefficients presented in Section-7.2 is utilized. A data-structure, called zerotree is defined as a tree-like data structure that includes an insignificant coefficient into it, provided all the descendants of that coefficient are also insignificant. A zerotree must therefore have a root, which itself is insignificant, but its parent is significant at that threshold. If all the ancestors till the coarsest frequency LL subband form the zerotree, then the ancestor at LL subband is declared as the zerotree root. The zerotree concept is based on the hypothesis that if a DWT coefficient at a coarse scale is insignificant with respect to a given threshold, then all its higher frequency descendants are likely to be insignificant with respect to the same threshold. Although, this may not be always true, but these are generally true. It may however be noted that all insignificant coefficients may not be a part of zerotree. It is possible that a coefficient is insignificant, but has some significant descendants. These coefficients are called *isolated zero.* Four symbols are used to encode the significance map, namely

- Zerotree root (ZTR).
- Positive significance (PS).
- Negative significance (NS).
- Isolated Zero (IZ).

The encoding of the coefficients into one of the above four symbols is illustrated as

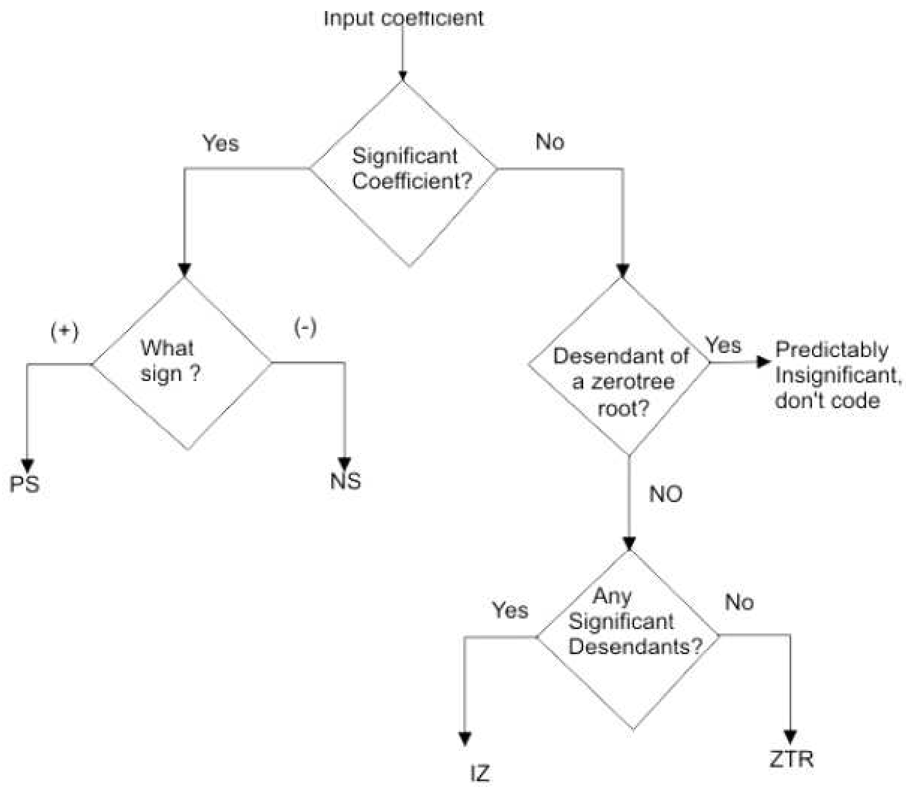


FIGURE 7.3 Flow chart for encoding significant map

Zerotree coding reduces the cost of encoding the significance map using self-similarity. Even though DWT essentially decorrelates the coefficients, occurrences of insignificant coefficients are not independent events. It is easier to predict insignificance, rather than predicting significant details across the scales and zerotree coding exploits the redundancies that the insignificant coefficients offer.

7.6 Successive Approximation Quantization (SAQ)

Successive Approximation Quantization (SAQ) performs encoding of magnitudes of DWT coefficients in successive stages. An initial threshold T_0 to examine the significance is first set up such that $T_0 > |X_{max}| /2$, where X_{max} is the maximum of all DWT coefficients. In each stage of encoding, it reduces the threshold by half and examines the significance once more. The sequence of thresholds that get applied in successive stages $T_{0,}$ $T_{1,}$ $T_{2,}T_{3,}$ ………. T_{N-1} are where *N* is the number of passes and $T_i = T_{i-1/2}$ for i= 1,2,3,4………N-1 .Each stage consists of two passes – a *dominant pa*ss and a *subordinate pass*.

7.6.1Dominant pass

A dominant pass is used to encode those coefficients that have not yet (that is, till the previous stage of encoding) been found to be significant with respect to a threshold T_i. The significant coefficients identified during this pass in the same scanning order, as illustrated earlier in fig.7.2 are encoded in zerotree structures, discussed in Section-7.4 and their magnitudes are appended to a list, known as subordinate list. At the same time, the coefficient in the DWT array is set to zero such that during the next dominant passes at lower thresholds, the coefficient is treated as insignificant and can be included as a part of zerotree.

7.6.2 Subordinate pass

A dominant pass is followed by a subordinate pass in which the coefficients found to be significant in the subordinate list are scanned and their magnitudes are refined with an added bit of precision, splitting the uncertainty region of encoding into two halves. For each magnitude in the subordinate list, this refinement can be encoded using a binary symbol, “0” if it falls in the lower half of the uncertainty region and “1” if it is in the other half. The string of symbols generated from during the subordinate pass is entropy coded. After the completion of a subordinate pass, the magnitudes on the subordinate list are sorted in decreasing amplitude, to the extent that the decoder also should be able to carry out the same sorting. The encoding process alternates between dominant pass and subordinate pass and the threshold is halved after each dominant pass. The encoding stops when some target bit rate is achieved. The ability to truncate the encoding or decoding anywhere is extremely useful in systems that are rate-constrained or distortion-constrained.

7.7 An encoding example

127	69	24	73	13	5	-8	5
-37	-18	-18	8	-6	7	15	4
44	-87	-15	21	8	-11	14	-3
55	18	29	-56	0	-2	3	7
34	38	-18	17	3	-9	-2	1
-27	-41	11	-5	0	-1	0	-3
6	17	5	-19	2	0	-3	1
32	26	-7	5	-1	-5	7	4

FIGURE 7.4 Example DWT coefficient array for 3-level on an 8x8 image.

The basic principles of EZW coding described so far can be best understood by considering an example array of DWT coefficients, as shown in fig.7.4. The example shows a 3-level DWT coefficient array of an 8 x 8 image, split into 10 subbands. It may be observed that the magnitude of the highest DWT coefficient is 127. The initial threshold may be set anywhere in the range (63.5, 127]. We set the initial threshold T_0 as 64. Before we begin the first dominant pass, all the coefficients in this array were treated to be insignificant. With respect to the initial threshold, the dominant pass picks up the following significant coefficients in the scanning order illustrated in fig.7.2.

- Coefficient value 127 in LL3. This will be encoded as "PS", since the coefficient is of positive value. After decoding this signal, the decoder knows that the coefficient lies in the interval [64,128) and its reconstruction value is the centre of this interval, i.e., 96.

- Coefficient value 69 in HL3. This will also be encoded as "PS". As before, its reconstruction value is also 96.
- Coefficient value 73 in HL2. This will also be encoded as "PS" with a reconstruction value of 96.
- Coefficient value -87 in LH2. This will be encoded as "NS", since the coefficient is of negative value. The decoder knows that the magnitude of the coefficient lies in the interval [64,128) and its reconstruction value will be -96.

All remaining coefficients are insignificant in the first dominant pass. The first dominant pass scanning will identify the following zerotree root (coded as "ZTR") and isolated zeros (coed as "IZ"):

- Coefficient value of -37 in LH3 is insignificant, but it has significant coefficient value of -87 in its descendants in LH2. Thus, this coefficient will be encoded as isolated zero (IZ).
- Coefficient value of -18 in HH3 is insignificant and all its descendants in HH2 and HH1 are insignificant. Thus, this coefficient qualifies to be a zerotree root (ZTR) and will be encoded accordingly.
- The reader may verify that the following coefficients are also zerotree root (ZTR):
    - 24, -18 and 8 in HL2.
    - 44, 65 and 18 in LH2.
- Also observe that the following coefficients are zeros, but not a part of any zerotree root:

- -8, 5, 15 and 4 in HL1.
- -18, 17, 11 and -5 in LH1.
-

For these highest frequency subbands, ZTR and IZ may be merged into a common symbol of “Zero” (Z). At the end of the first dominant pass, the subordinate list will contain only the four significant coefficients identified. The first subordinate pass will refine the magnitudes of the significant coefficients and categorize them into one of the two uncertainty intervals, viz., [64, 96) and [96,128). Thus, only the LL3 coefficient of magnitude 127 will belong to the latter interval and will be encoded with symbol 1, whereas the remaining three significant coefficients will belong to the former interval and encoded with symbol 0. The first coefficient will have a reconstruction value of 112 and the remaining coefficients will have a reconstruction value of 80 at the middle of the uncertainty interval. In this case, the first coefficient only is encoded as 1 and the remaining as 0s and no re-ordering in subordinate list is necessary.

The first dominant and the first subordinate pass complete the first stage of processing. Now, the second dominant pass starts with threshold set to $T_1= T_0 /2=32$ .During this pass, the coefficients which are yet to be found as significant will only be scanned. All the coefficients previously found to be significant are set to zero so that they could be included as a part of zerotree in this, as well as latter passes. However, the subordinate list is still maintained and the second and subsequent passes will only append the significant coefficients found in that pass to the subordinate list.

As an exercise, the student is advised to complete the second dominant and second subordinate pass encoding. The processing alternately continues between the dominant pass and the subordinate pass and can be stopped at any time.

7.8 **Order of importance in the bit-stream**

The embedded bit-stream, which the EZW algorithm generates inherently, performs an ordering of bit-stream according to the importance. The importance follows the order of precision, magnitude, scale and spatial location according to the initial dominant list. The first importance is assigned to the numerical precision of the coefficients. All the coefficients in a pass are encoded with the same numerical precision and it is only after a dominant pass that the numerical precision is refined by a factor of two. The next in importance is the magnitude. Prior to a pass, all coefficients are assumed to be insignificant and the dominant pass picks up all significant coefficients, having magnitudes greater than those of the insignificant coefficients. During the subordinate pass, the magnitudes are sorted in a descending order of the centres of uncertainty intervals. Scale is the next factor of importance. It follows the ordering of subbands on the initial dominant list. The coarser scales are covered before the finer or the high-frequency coefficients. The final factor is the spatial location. It simply means that two coefficients, which cannot be distinguished by precision, magnitude and scale, have their relative importance decided arbitrarily by the initial scanning ordering of the two coefficients within a subband.

CHAPTER 8

PURPOSED WORK

8.1 Introduction

In this work, a new approach to image compression technique is proposed that enhanced the compression performance. The new technique considers both discrete cosine Transform and Discrete Wavelet Transform.

In this approach we select the compression technique on the decision parameter of the image .In this work the decision parameter of image is its Standard Deviation (STD).

In this work, SVD associates DCT and BTC in JPEG as a baseline coding. This technique considers these there DTC, SVD and BTC jpeg based compression technique. The incorporation of SVD with nearest neighborhood approach has improved the compression performance significantly. And this work consider two compression technique based on wavelet transformation i.e. pyramid decomposition using Gaussian filter, and EZW.

8.2 Proposed technique

The proposed algorithm incorporates both DCT and SVD transform coding instead of DCT only in baseline coding. Depending on the image properties a simple decision making criterion is o choose used the transform to be employed. The decision making criterion is based on the observation of standard deviation (STD) of the source SD of an image is larger when it has many abrupt changes in intensity than when the image has Smoothly varying intensity. The technique is based on the following algorithm.

1. Input an image which is processed according to the selected compression technique selected based on the STD of the image.

2. The STD of current image is computed and for the compression technique belonging to jpeg compression (other than wavelet) if STD is lesser than the decision making

parameter, the DCT is used to compute the transform coefficients. Conversely if the standard deviation is more than the decision making parameter the SVD transform is used. However if the STD is lying between 35 to 45 then BTC is used even though the results of compression for BTC depends on the block size of image here we are considering to be 8x8. This is done because DCT is computationally efficient and achieves good performance for images characterized by high correlation. In contrast, SVD provides optimal energy packing efficiency for less correlated images.

3. For the compression technique belongs to wavelet technology pyramid and EZW are incorporated such that if STD is lesser the decision parameter, the EZW is used for the compression .conversely if the standard deviation is more the decision making parameter the pyramid work better even though the its compressed image quality depends on the number of levels used for compression.

4. The image quality factors are given as the output.

8.3 Image quality measurements

Image quality measures play important roles in various images processing application .Once image compression System has been designed and implemented, it is important to be able to evaluate its performance. This evaluation should be done in such a way to be able to compare results against other image compression techniques. The image quality metrics can be broadly classified into two categories, subjective and objective. Subjective image quality is a method of evaluation of images by the viewers read images directly to determine their quality. In objective measures of image quality metrics, some statistical indices are calculated to indicate the image quality. In our work we will focus in objective measures such as Peak Signal to Noise Ratio (PSNR) and mean square error (MSE).

The PSNR is most commonly used as a measure of quality of reconstruction of lossy compression .It is an attractive measure for the loss of image quality due to its simplicity and mathematical convenience .Peak signal-to-noise ratio (PSNR) is a qualitative measure based on the mean-square-error of the reconstructed image .If the reconstructed image is close to the original image, then MSE is small and PSNR takes a large value .PSNR is dimensionless and is expressed in decibel .Peak Signal-to-Noise Ratio (PSNR) avoids this problem by scaling the MSE according to the image range .PSNR is defined as follow:

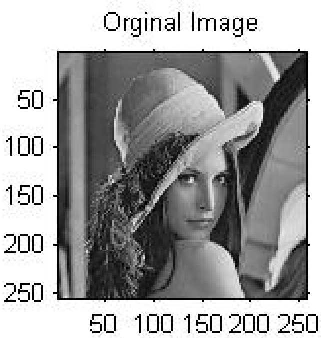
$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left[y(i,j) - x(i,j) \right]^2$$

$$PSNR = 10 \log \left(\frac{L^2}{MSE} \right)$$

Where L is the dynamic range of the pixel values (255 for 8-bit grayscale images).

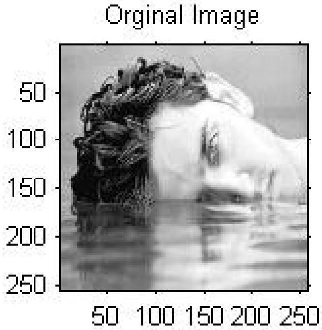
8.4 Experimental Results

8.4.1. Input image-lena.bmp



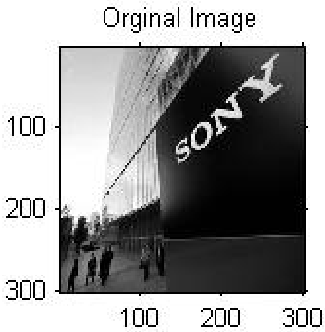
Algorithm	PSNR	MSE	SD
DCT	30.70271	55.31073	52.59425
SVD	33.97248	26.0515	52.59425
BTC	30.08119	63.82063	52.59425
EZW	30.25315	61.34305	52.59425
Pyramid	32.75411	34.48814	52.59425

8.4.2. Input image-Hrithik.bmp



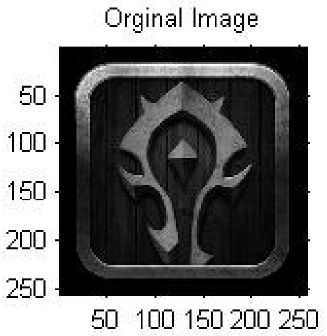
Algorithm	PSNR	MSE	SD
DCT	31.86982	42.27641	69.02297
SVD	38.79564	8.580566	69.02297
BTC	30.73738	54.87093	69.02297
EZW	27.23152	123.0065	69.02297
Pyramid	33.16875	31.3477	69.02297

8.4.3. Input image-sony.bmp



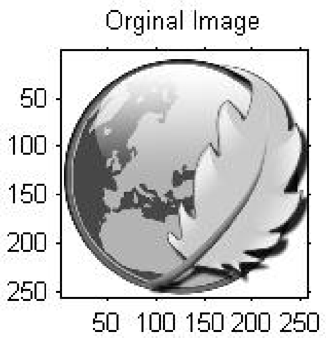
Algorithm	PSNR	MSE	SD
DCT	29.53501	72.37341	88.9451
SVD	32.66404	35.21088	88.9451
BTC	30.07144	63.96414	88.9451
EZW	14.50748	2303.206	88.9451
Pyramid	34.09747	25.31241	88.9451

8.4.4. Input image-Square.bmp



Algorithm	PSNR	MSE	SD
DCT	32.81447	34.01212	47.78622
SVD	37.70188	11.03804	47.78622
BTC	32.03844	40.66646	47.78622
EZW	34.66862	22.19313	47.78622
Pyramid	34.20711	24.68137	47.78622

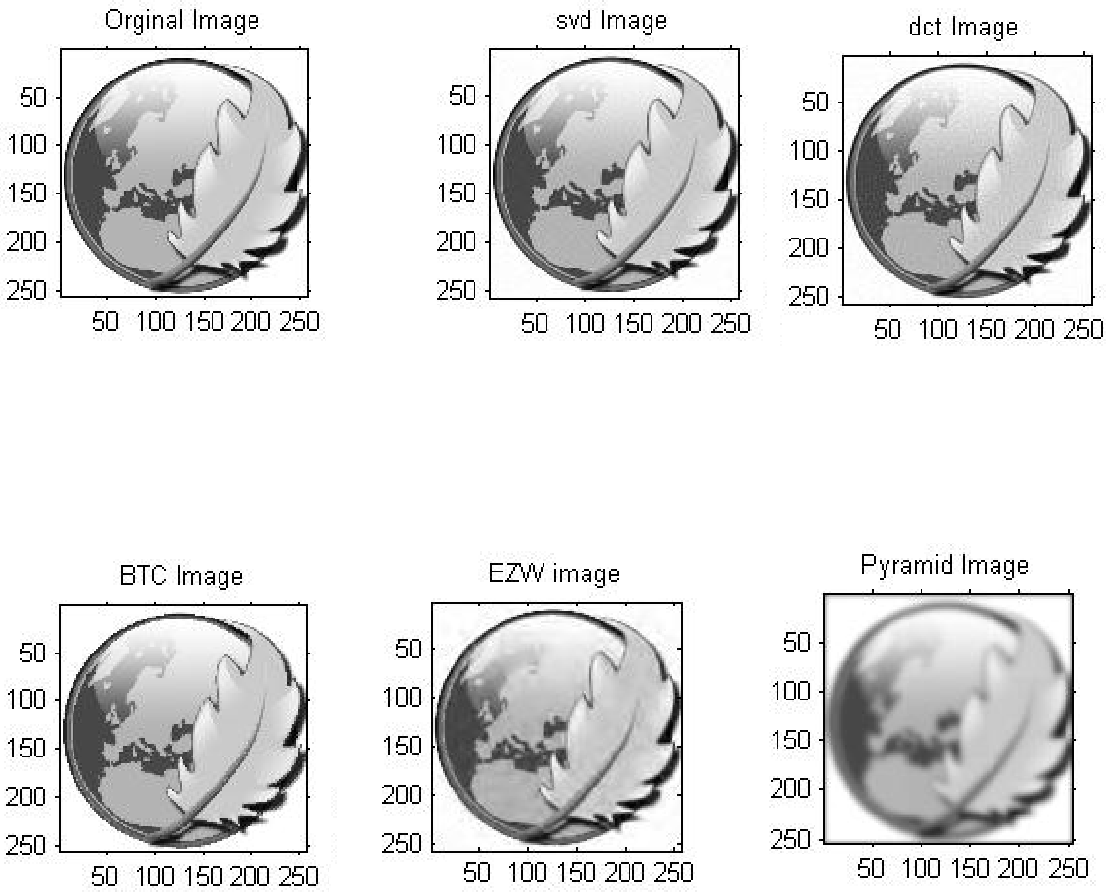
8.4.5. Input image-Circle.bmp



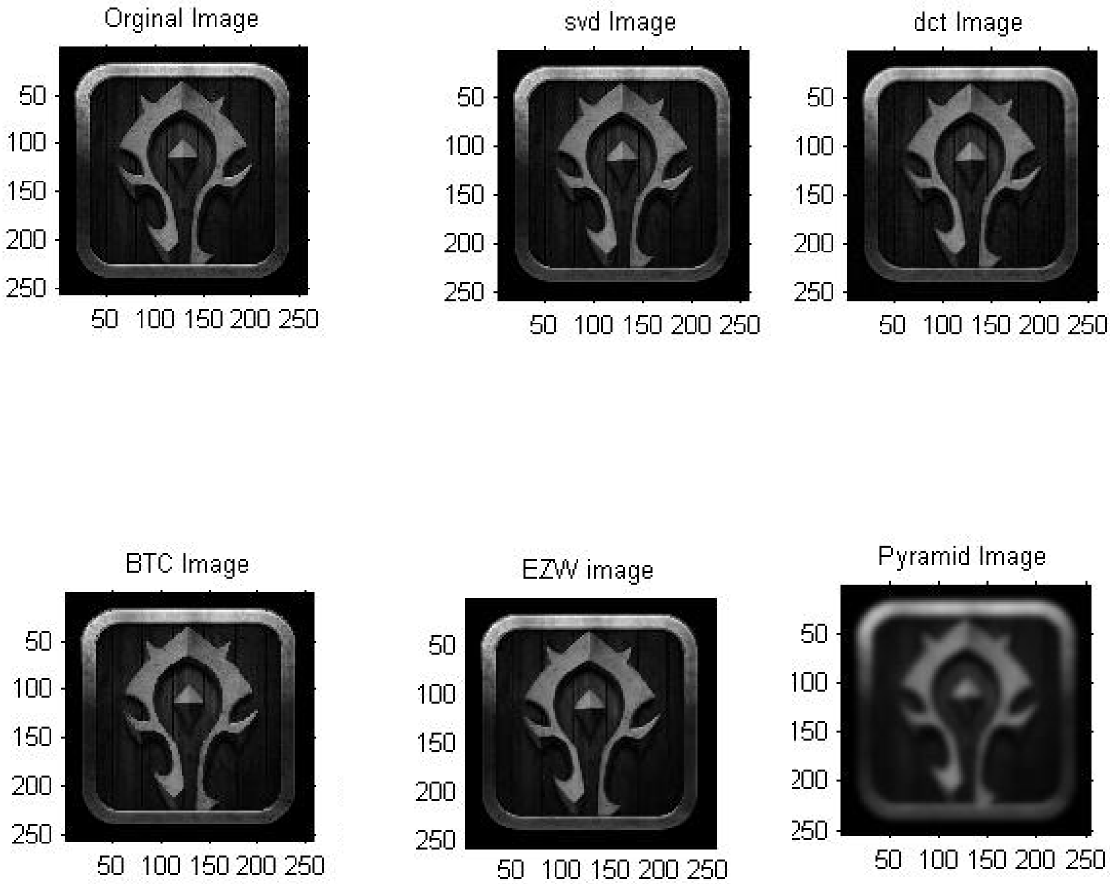
Algorithm	PSNR	MSE	SD
DCT	29.3267	75.92943	66.34601
SVD	33.07188	32.05476	66.34601
BTC	28.8524	84.69154	66.34601
EZW	27.19073	124.1672	66.34601
Pyramid	32.25451	38.69276	66.34601

8.5 Following are the screen shot of the output for various input images with different standard deviation.

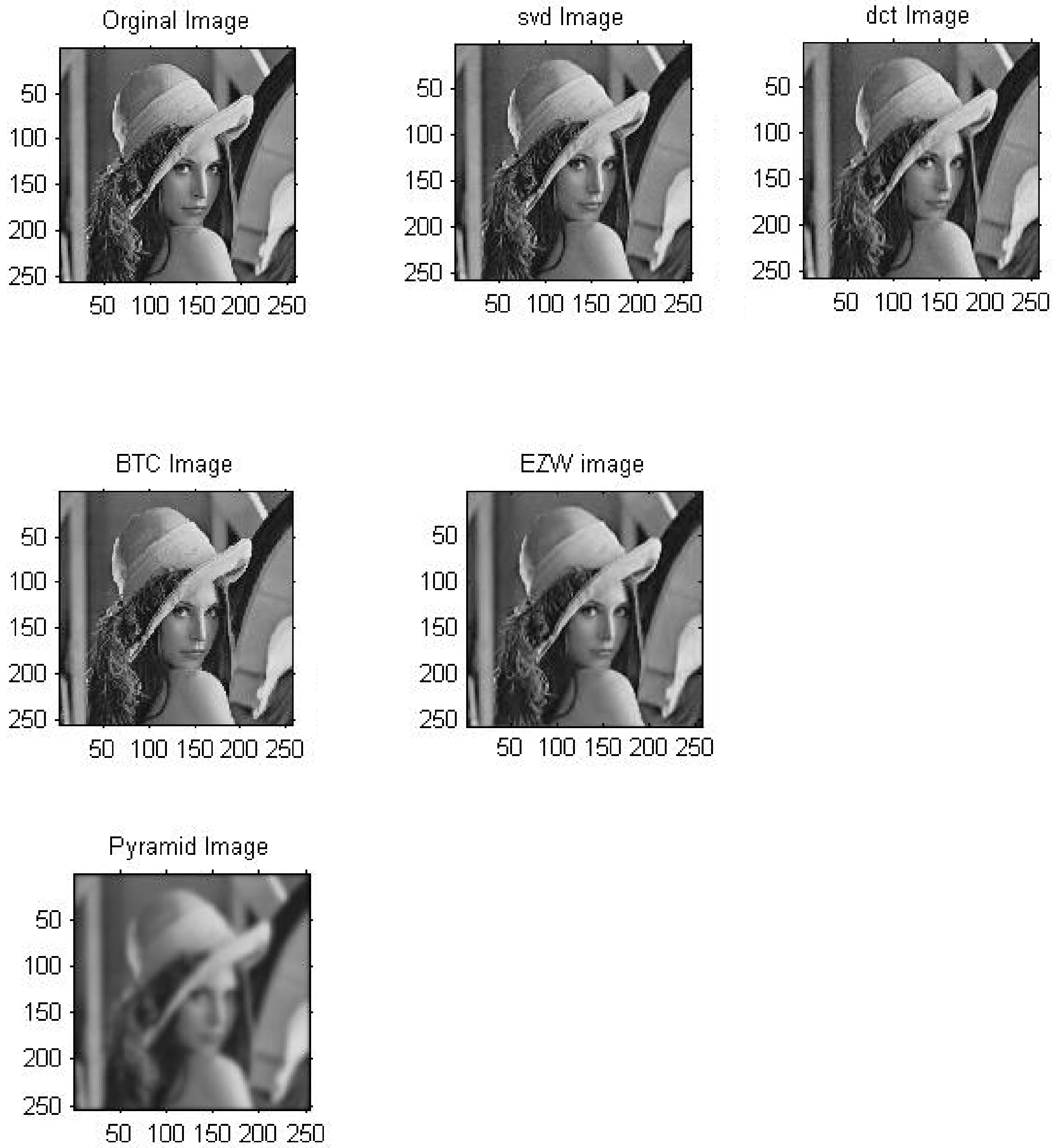
8.5.1. Images with STD 66.34



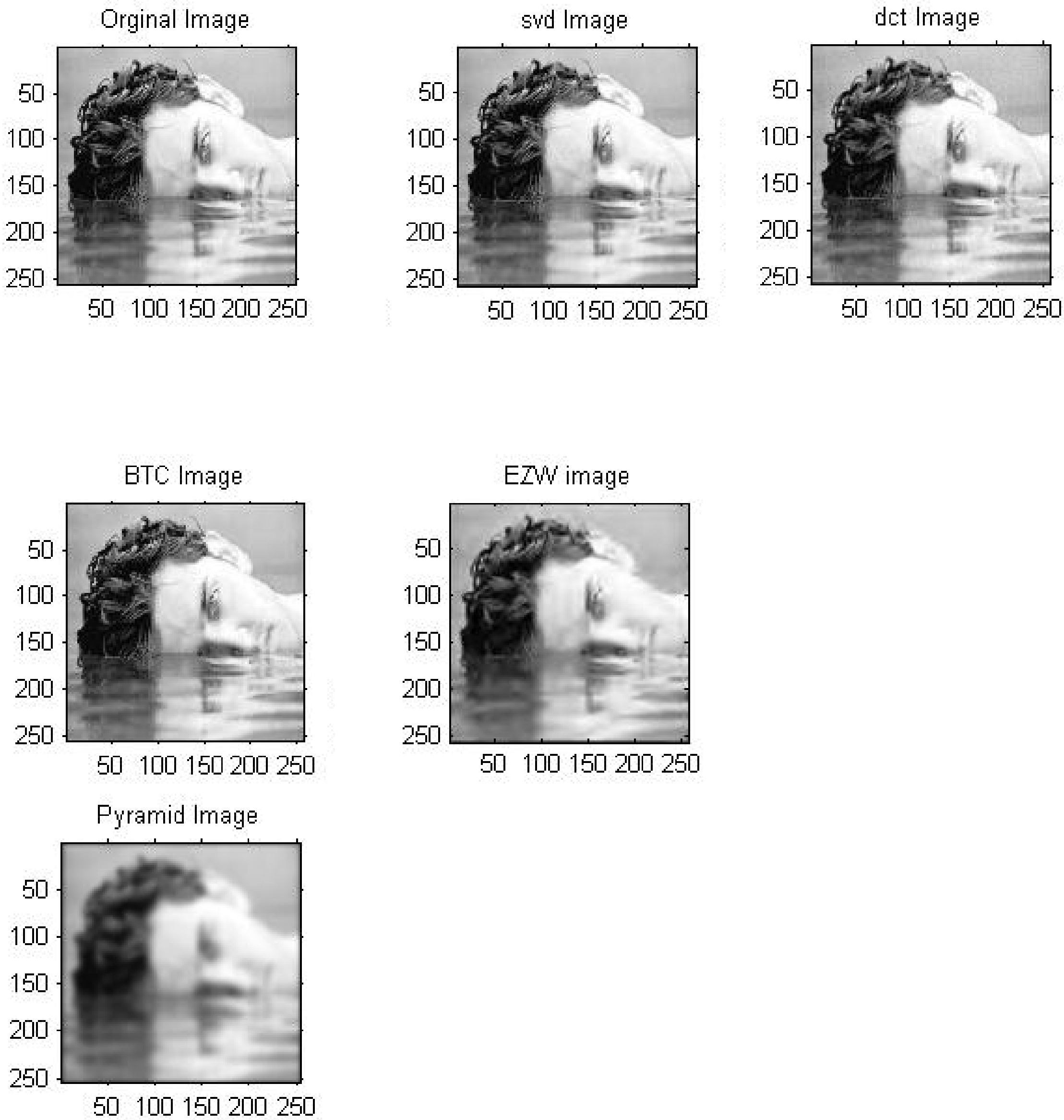
8.5.2. Image with STD 47.78



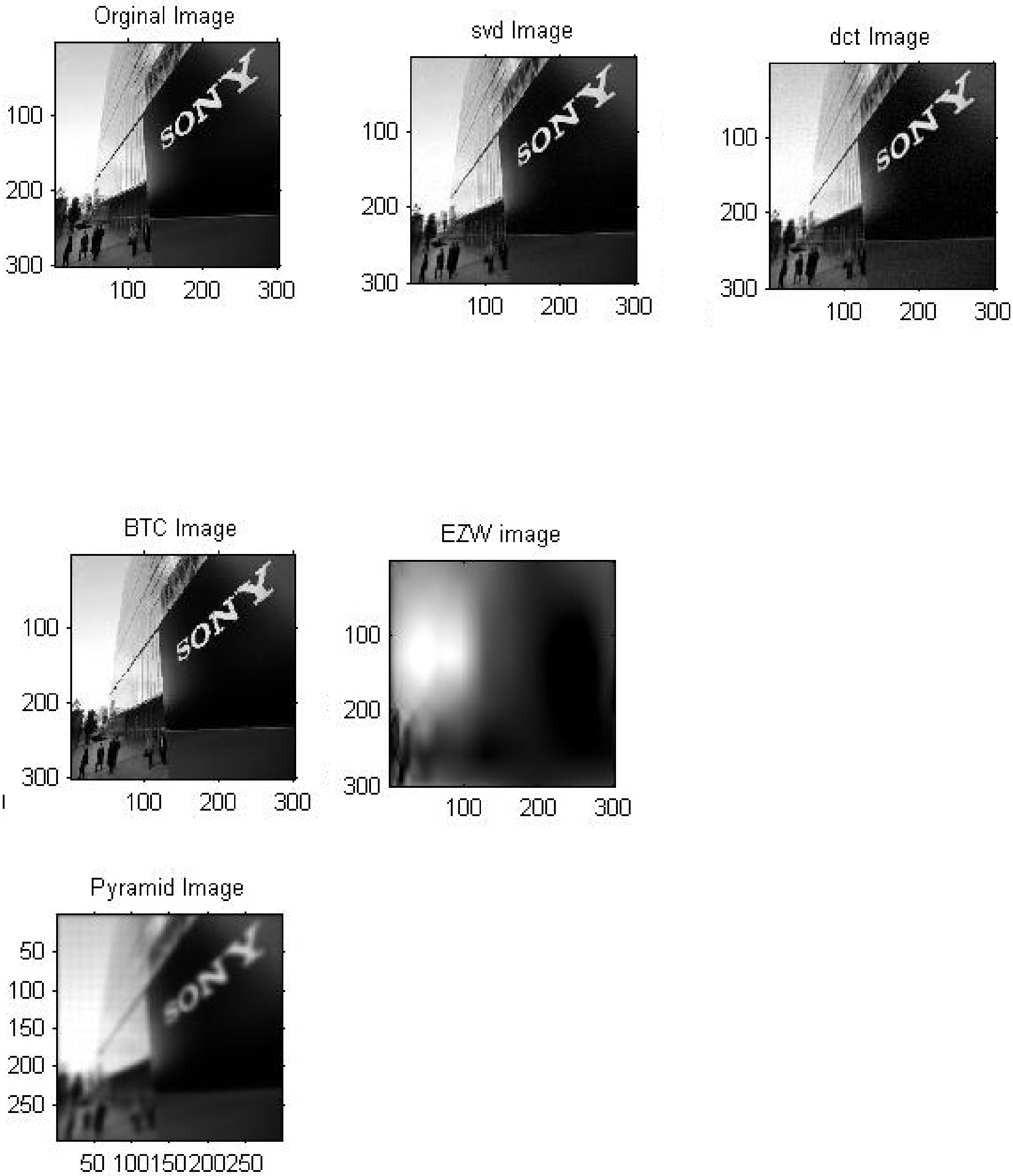
8.5.3. Image with STD 52.59



8.5.4. Image with STD 69.02



8.5.5. Image with STD 88.94



CHAPTER 9

CONCLUSION AND FUTURE SCOPE

This chapter concludes the work in this thesis in terms of the various input and output parameters that have been considered while compressing images using Different compression techniques.
It also provides with a look up in the future scope of our work area.

9.1 Conclusion

This thesis presents a comparative analysis of various image compression techniques using wavelet transforms and discrete cosine Transformation. A lot of combinations have been applied in order to find the best method.

The analysis, of all the obtained experimental results, demonstrates that the incorporation of SVD and BTC in image compression along with DCT in an adaptive manner enhances the compression performance significantly. The proposed technique perform perform the best technique in terms of PSNR and MSE.

But it requires slightly longer time that makes it suitable for large bandwidth channel only.

In this research compression technique is selected on the basis of its standard deviation used as decision parameter for compression. Compression techniques other then wavelet transformation can be divided as SVD for the image having large standard deviation (greater than 45). If the standard deviation lies between 40 to 45 BTC compression is selected.and lesser standard deviation DCT is selected.

And for wavelet based compression technique we select EZW for image with lesser standard deviation and pyramid for image with larger standard deviation.

9.2 Future scopes

The field of image processing has been growing at a very fast pace. The day to day emerging technology requires more and more revolution and evolution in the image processing field. The well known saying “A picture says a thousand words” can be taken as the main motive behind the need of image processing.

The work proposed in this thesis also portrays a small contribution in this regard. The proposed compression technique can provide a good platform for further research work in this respect.

This work can be further enhanced to by selecting other decision parameter other than of standard deviation of the images. It will provide a good add on to the already existing compression techniques used for images compression.

Moreover, for future work we can train our algorithm using various AI techniques like fuzzy logic or neural network, in order to attain the best output without performing calculations for each and every combination. Use of AI techniques will lead to the optimal solution directly, with more efficiency and less tedious work.

REFERENCES

1 R.C. Gonzalez & R. E. Woods, “Digital Image Processing”, Addison-Wesley Co., 1991.

2 A.K.Jain, “Fundamentals of Digital Image Processing”, Prentice-Hall Inc, Englewood Cliffs, 1989.

3. http://en.wikipedia.org/Image_compression.

4. K. Ramchandran, A. Ortega, K. Metin Uz, and M. Vetterli, “Multiresolution broadcast for digital HDTV using joint source/channel coding.” *IEEE Journal on Selected Areas in Communications*, 11(1):6−23, January 1993.

5. M. W. Garrett and M. Vetterli, “Joint Source/Channel Coding of Statistically Multiplexed Real Time Services on Packet Networks,” *IEEE/ACM Transactions on Networking*, 1993.

6. S. McCanne and M. Vetterli, “Joint source/channel coding for multicast packet video,” International Conference on Image Processing (Vol. 1)-Volume 1, October 1995, Washington D.C**.**

7. K. Sayood and J. C. Borkenhagen, “Use of residual redundancy in the design of jointsource/channel coders,” *IEEE Transactions on Communications*, 39(6):838-846, June 1991.

8. J. Modestino, D.G. Daut, and A. Vickers, “Combined source channel coding of images using the block cosine transform,” *IEEE Transactions on Communications*, vol. 29, pp.1261-1274, September 1981.

9. Q. Zhang, Z. Ji, W. Zhu, J. Lu, and Y.-Q. Zhang, “Joint power control and source-channel Coding for video communication over wireless,” *IEEE VTC’01*, October 2001, New Jersey.

10 Hayder Radha, “Lecture Notes: ECE 802 - Information Theory and Coding,” January 2003.

11 R. C. Gonzalez and P. Wintz, “Digital Image Processing,” Reading. MA: Addison-Wesley,
1977.

12 N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete cosine transform,” *IEEE Transactions on Computers*, vol. C-32, pp. 90-93, Jan. 1974.

13 W. B. Pennebaker and J. L. Mitchell, “JPEG – Still Image Data Compression Standard,”
Newyork: International Thomsan Publishing, 1993.

14. H. C. Andrews and C. L. Patterson. Singular Value Decomposition (SVD) Image Coding. *IEEE Transactions on Communications*, 24:425–432, April 1976.

15. C. S. M. Goldrick, W. J. Dowling, and A. Bury. Image coding using the singular value decomposition and vector quantization. In *Image Processing and Its Applications*, pages 296–300. IEE, 1995
.

16. P. Waldemar and T. A. Ramstad. Image compression using singular value decomposition with bit allocation and scalar quantization. In *Proceedings of NORSIG Conference*, pages 83–86, 1996.

17. J.-F. Yang and C.-L. Lu. Combined Techniques of Singular Value Decomposition and Vector Quantization for Image Coding. *IEEE Transactions on Image Processing*, 4(8):1141–1146, August 1995.

18. E. J Delp., Saenz, M., and Salama, P., 2000, Block Truncation Coding (BTC), Handbook of Image and Video Processing, edited by Bovik A. C., Academic Press, pp. 176-181.

19. C. C. Tsou, S. H. Wu, Y. C. Hu, "Fast Pixel Grouping Technique for Block Truncation Coding," 2005 Workshop on Consumer Electronics and Signal Processing (WCEsp 2005), Yunlin, Nov. 17-18, 2005.

20. A. M. Eskicioglu and P.S. Fisher, “Image quality measures and their performance,” IEEE Trans. Communications, vol. 34, pp. 2959-2965, Dec. 1995.

21 J. Someya, N. Okuda, and H. Sugiura, “The suppression of noise on a dithering Image in LCD overdrive,” *IEEE Trans Consumer Electron.*, vol. 52, no. 4, pp. 1325-1332, Nov. 2006.

22 E. J. Delp and O. R. Mitchell, “Image compression using block truncation coding,” *IEEE Trans. Commun.*, vol. COM-27, no. 9, pp. 1336-1342, Sep. 1979.

23 T. Kurita and N. Otsu, “A method of block truncation coding for color image compression,” *IEEE Trans. Commun.*, vol. 41, no. 9, pp. 1270-1274, Sep. 1993.

24 Y. Linde, A. Buzo, and R. M. Gray, “An algorithm for vector quantizer design,” *IEEE Trans. Commun.*, vol. COM-28, no. 1, pp. 84-95, Jan. 1980.

25 Tinku Acharya, Ajoy.K.Ray, “IMAGE PROCESSING –Principles and Applications”, Hoboken, New Jersey, A JOHN WILEY & SONS, MC. , Publication, 2005.

26 Detail information about the Multiresolution Analysis and The Continuous Wavelet Transform6, < http://users.rowan.edu/~polikar/WAVELETS/WTpart3.html>

27 Detail information about the Fundamental Concepts of Transformations, http://users.rowan.edu/~polikar/WAVELETS/WTpart1.html

28. Detail information about Wavelet Analysis http://www.mathworks.com/access/helpdesk/help/toolbox/wavelet

29 Aglika Gyaourova, Chandrika Kamath and K. Fodor, “Undecimated wavelet transforms for image de-noising”, November 19, 2002.

30. S.Kother Mohideen, Dr. S. Arumuga Perumal, Dr. M.Mohamed Sathik, “Image denoising Using discrete wavelet transform”, *International Journal of Computer Science and Network Security*, vol.8 no.1, January 2008

31. Detail information about Multiresolution Analysis: The Discrete Wavelet Transform8, < http://users.rowan.edu/~polikar/WAVELETS/WTpart4.html>

32. P Burt and E. Adelson, "The Laplacian Pyramid as a Compact Image Code," *IEEE Transactions on Communication*, COM-31 pp. 532-540, 1983a.

33. S. Mallat. “*A Wavelet Tour of Signal Processing*”. Academic Press, New York, NY, 1998.

34. Nusrat Ahmed Surobhi, "Amalgamation of SVD to JPEG for Enhanced performance, "*International conference on Information & communication Technology-2007*".

35. S.P.Raja, “Performance Evaluation on EZW","*IEEE Transactions on Communication-2010*".

36. Doaa Mohammed,"Image compression using BTC", "*Multidisciplinary Journal in Science & technology, JSAT, Feb 2011"*.