Saurashtra University
Re – Accredited Grade 'B' by NAAC
(CGPA 2.93)

Dudhagara, Chetan R., 2013, "A comparative study, analysis and modelling of various image processing algorithms", thesis PhD, Saurashtra University

http://etheses.saurashtrauniversity.edu/1033

Copyright and moral rights for this thesis are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author.

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

Saurashtra University Theses Service
http://etheses.saurashtrauniversity.edu
repository@sauuni.ernet.in

© The Author

A COMPARATIVE STUDY, ANALYSIS AND MODELLING OF VARIOUS IMAGE PROCESSING ALGORITHMS

A Thesis Submitted To

SAURASHTRA UNIVERSITY

For the award of the degree of

DOCTOR OF PHILOSOPHY

IN

COMPUTER SCIENCE

In the Faculty of Science

: Submitted By :

CHETANKUMAR RAVJIBHAI DUDHAGARA

(Ph.D. Reg. No. : 4298 & Date : 01/01/2011)

: Under the Guidance of :

DR. K. H. ATKOTIYA

J. H. BHALODIA WOMEN'S COLLEGE
RAJKOT

MARCH – 2013

CERTIFICATE FOR PRE Ph.D. PRESENTATION

This is to certify that ***Mr. Chetankumar Ravjibhai Dudhagara*** has made Pre Ph.D. presentation as per UGC Guideline "University Grant Commission (Minimum Standard and Procedure for award of Ph.D. Degree) Regulation-2009" and Saurashtra University Ordinance for Ph.D. Programme (O.Ph.D. 6.2), on the research work entitled *"**A Comparative Study, Analysis and Modelling of Various Image Processing Algorithms**"* in the department of Computer Science on 21/02/2013 before all the faculty members and students of the Department for getting feedback and comments.

I also certify that the Research work was appreciated by all who remain present and there was no comments made for this research work.

Place : Rajkot   Dr. K. H. Atkotiya

Date :   J. H. Bhalodia Women's College

Rajkot.

CERTIFICATE

This is to certify that the contents of the doctorate degree thesis entitled "***A Comparative Study, Analysis and Modelling of Various Image Processing Algorithms***" is the original research work of ***Mr. Chetankumar Ravjibhai Dudhagara*** carried out under my supervision and guidance.

I further certified that the work has not been submitted either partially or fully to any other University/Institute for the award of any degree according to best of my knowledge.

Place : Rajkot

Date :

Dr. K. H. Atkotiya
J. H. Bhalodia Women's College
Rajkot.

STATEMENT

I hereby declare that the work incorporated in the present doctorate degree thesis is original and has not been submitted to any University/Institution for the award of the Degree.

I further declare that the result presented in the doctorate degree thesis, considerations made there-in, contribute in general the advancement of knowledge in education and particular to "***A Comparative Study, Analysis and Modelling of Various Image Processing Algorithms***" in the context of Image Processing Technology.

Place : Rajkot
Date :

Dudhagara Chetankumar Ravjibhai

ACKNOWLEDGEMENT

I extend my sincere thanks to **Dr. K. H. Atkotiya** – my research guide, who has permitted to do work under his guidance. I take this opportunity to convey my deepest gratitude to him for his valuable advice, comfort encouragement, keen interest, constructive criticism, scholarly guidance & wholehearted support.

I take opportunity to express my deep sense of gratitude to **Dr. M. K. Padalia**, Vice-Chancellor of the Saurashtra University for his consistent encouragement to the research and development.

I am thankful to **Dr. C. K. Kumbharana, Dr. A. M. Gonsai** and all teaching and non-teaching staff members of Computer Science Department, Saurashtra University, constantly took interest in my pursuit and facilitated it in every possible way. Their genuine interest in my career and warm wishes constitute a great part my efforts.

I express my special thanks to **Dr. P. V. Virpariya**, Professor & Head, Computer Science Department, Sardar Patel University for his constant encouragement, advice and for providing the required moral support during the work.

I express my special thanks to **Dr. D. R. Kathiriya**, Director of IT, Anand Agricultural University for his constant encouragement, advice and for providing the required moral support during the work.

I express my special thanks to my friends **Sanjay Bhanderi, Hasamukh Patel, Hitesh Solanki** and **Navtej Bhatt** for his constant support and help during the work.

I must express my indebtedness to my parents, my brothers, wife and my lovely son. Without their kind and silent support this work could not have been completed.

I specially thankful to my co-research scholars for their kind co-operation and all my relatives and friends who constantly inspired me.

Finally I wish to make devout supplication to the Divine Spirit without whose blessings this task would not have been accomplished.

- C. R. Dudhagara

To my parents, my wife and my loving son

CONTENTS

LIST OF FIGURE

LIST OF TABLE

CHAPTER – 1

RESEARCH SURVEY AND INTRODUCTION

1.1 INTRODUCTION

Image processing techniques were first developed in 1960, with the help and collaborative work of wide range of academicians and scientists. The main objective and focus of their work was to develop the medical imaging, character recognition and create high quality images at the microscopic level. At that time, the hardware / equipment and the processing cost were most important factors to the research work.

The financial constraints play a serious role on the depth and breadth of technology development. In year 1970s, the cost of the computing equipment was dropped substantially and used of making digital image processing are more realistic. Film and software companies invested significant funds into the development and enhancement of image processing, creating a new industry.

Image processing is a rapidly growing area of computer science and Information Technology. Its growth has been fueled by technological advances in digital imaging, computer processors and mass storage devices. Fields which traditionally used analog imaging are now switching to digital systems, for their flexibility and affordability. The examples are medicine technology, film and video production, photography, remote sensing, and security monitoring. These all industries produce a very huge amount of digital data or images every day.

Generally all the image processing techniques involve treating the image as a two dimensional signal. It applies the standard signal processing techniques to that image. Image processing is generally referred to as a digital image processing, but analog image processing are also possible.

Digital Image Processing is a set of different techniques for manipulation of digital images by computers. The raw data received from the imaging sensors on the satellite platforms contains flaws and deficiencies. To overcome these flaws and deficiencies in order to get the originality of the data, it needs to perform some steps of processing. These steps of processing vary from different images, because it depends on the different types of image format, initial condition of the image and the information of interest and the composition of the image scene.

Digital image processing is primarily concerned with extracting useful information from images which is done by the computer.

Image processing algorithms may be placed at three levels such as:

- Lowest Level
- Middle Level
- Highest Level

Lowest Level: - At this level, we can put those techniques which deal directly with the raw, possibly noisy pixel values, with denoising and edge detection being good examples.

Middle Level: - At this level, we can include those algorithms which utilize low level results for further means, such as segmentation and edge linking.

Highest Level: - At this level, we can focus on those methods which attempt to extract semantic meaning from the information provided by the lower levels, for example, handwriting recognition.

The literature abounds with algorithms for achieving various image processing tasks. However, there does not appear to be any unifying principle guiding many of them. Some are one dimensional signal processing techniques which have been extended to two dimensions. Others apply methods from alternative disciplines to image data in a somewhat inappropriate manner. Many are the same basic algorithm with parameter values tweaked to suit the problem at hand.

The local segmentation principle states that the first step in processing a pixel should be to segment explicitly the local region encompassing it. In a local scale processing has the effect to make clear which pixels belong together, and which pixels do not. The segmentation process results in a local approximation of the underlying image, effectively separating the signal from the noise. Higher level algorithms can operate directly on the signal without risk of amplifying the noise. Local segmentation process can provide a common framework for constructing image processing algorithms.

Many existing image processing algorithms have already been making partial use of the local segmentation concept. It is possible to examine these algorithms with respect to the local segmentation model they use. This helps to make their strengths and weaknesses more apparent. Even popular techniques, such as linear and rank filters, can be framed in terms of their application of local segmentation.

Image denoising is particularly suited to demonstrating the utility of local segmentation. Denoising is the process of removing unwanted noise from an image. A denoised image is an approximation to the underlying true image, before it was contaminated. A good denoising algorithm must simultaneously preserve structure and remove noise. Obviously, to do this the algorithm must be able to identify what structure is present. Local segmentation specifically attempts to separate structure from noise on a local scale. Denoising would therefore be a good application with which to test different approaches to local segmentation.

Local regions only contain a small number of pixels. It is unlikely that there would be more than a few segments present at such a scale, so unconnected, homogeneous groups of pixels are likely to part of the same global segment.

1.2 SELECTION OF TITLE

Image processing is the application of different areas like computer science, electronics, mathematics, engineering etc. It is used to solve different types of problems related to these areas. Image processing was highly introduced only a few years ago. This field has a long history with its roots going back more years.

The use of information technology and mathematics has become a critical part of the image processing.

"In computer science and electrical engineering, image processing is any form of signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image".

The title of my research is: ***“A Comparative Study, Analysis and Modelling of Various Image Processing Algorithms”***

1.3 STATEMENT OF THE PROBLEM

In this research work, various image processing techniques are studied and analyzed. From these image processing techniques we performed this study only for one process. In this work, we selected Image Compression from various image processing techniques such as Image Enhancement, Image Restoration, Image Fusion, Image Registration, Image Segmentation, Image Transformations, Image Compression, Image Watermarking etc.

1.4 GENERAL OBJECTIVE OF THE STUDY

The aim and objectives of this research work is:

1. To study and analysis of various types of images.
2. To study various characteristics of image.
3. To study different processes related to image like Image Compression, Image Filtering, Image Enhancement, Image Restoration, Image Fusion, Image Registration, Image Segmentation, Image Transformations, and Image Watermarking etc…
4. To study various Image Processing Algorithms.
5. To analyzed various Image Processing Algorithms.

6. To perform a comparative study for any one Image Process.
7. To represent a model for this Image Processing Algorithm i.e. image compression.

Image Processing Software provides a comprehensive set of reference standard algorithms and graphical tools for image processing, analysis, visualization, and algorithm development.

1.5 RESEARCH METHODOLOGY

In order to write this thesis for research work, we have performed a literature review for a better understanding of previously used techniques and approaches. We conducted an experiment to find comparable values of Compression Ratio (CR) to support our research. In our research methodology (Qualitative and Quantitative) we have adopted the following methodologies in this study [1].

1.5.1 Qualitative Methodology - Literature Review

We performed a variety of literature review of various image compression algorithms proposed by different researchers in this area. Literature review is useful to study the past and current research status related to this specific area. It is useful to do our research in proper way.

At that time, when you select the research topic, you must have familiar with the topic whether it is researchable or not. It belongs to your area of study and interest. Researchers do literature review that helps them to relate to the previous study and to current study in that specific area. It provides framework for comparing results. Literature helps to explore not only research problems. Quantitative study not only gives some facts and figures that support qualitative study but also suggest possible questions and hypothesis related to area [1].

1.5.2 Quantitative Methodology - Experiments

Practical and experiments work are appropriate when we want to control over the situation and want to come on conclusion. In the quantitative part of the research, we made an attempt to evaluate and improve the Compression Ratio (CR).

There are two processes involved in the experiment.

(i) Compression
(ii) Decompression

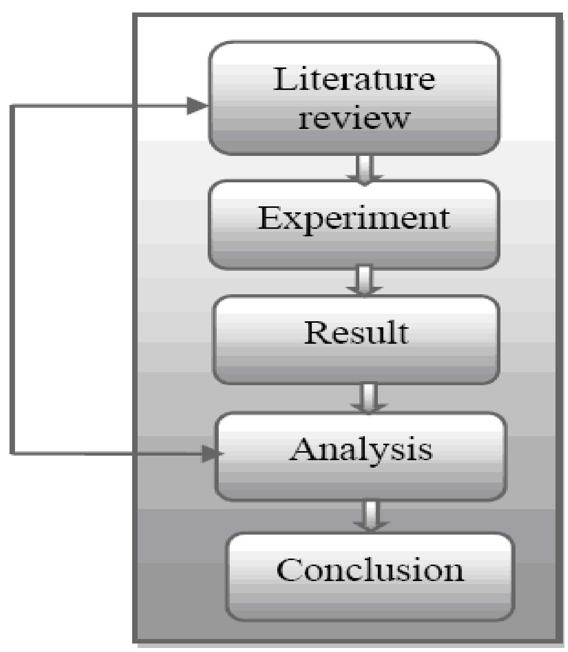


Figure – 1.1 Research Methodologies

Compression: - In image compression process, there are two types of variables: Independent and Dependent. The figure given below shows the compression process [1].

Figure – 1.2 Input output compression process

Independent Variables are:

(i) Curve Order
(ii) Block size
(iii) Quantization Number
(iv) Original Image

Dependent Variables are:

(i) Compression Ratio
(ii) Compressed Image
(iii) Response Time

Decompression: - Image decompression process is the reverse process of the compression process. It has also two types of variables: Independent and Dependent. The figure given below shows the decompression process [1].

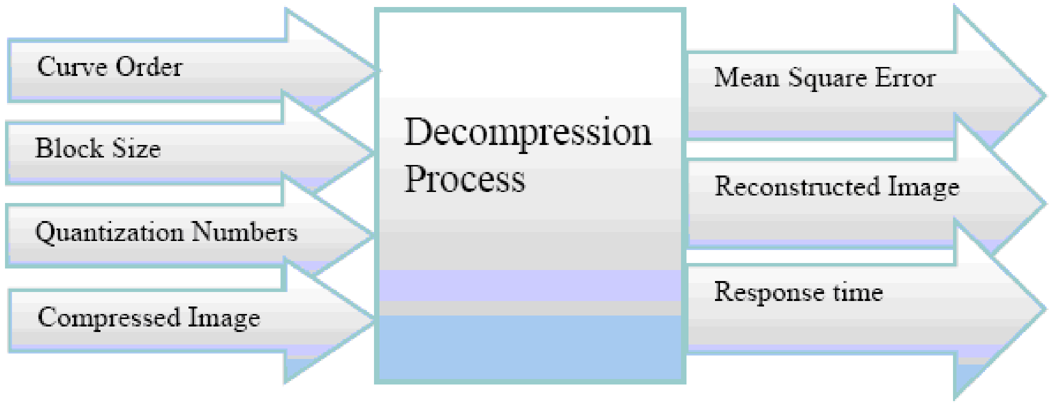


Figure – 1.3 Input output decompression process

Independent Variables are:

(i) Curve Order
(ii) Block Size

Dependent Variables are:

(i) Mean Square Error
(ii) Reconstructed Image
(iii) Response Time

1.6 SCOPE OF THE STUDY

The scope of the research work is to perform a comparative study, analysis and modelling of various image processing algorithms and to try develop a model for suitable image processing algorithm. In this research work, we select image compression from different image processing.

In this research work, we would perform a comparative study and analysis of image compression algorithm. Here we would mainly carry out our study on four image compression algorithms such as Wavelet, JPEG, Vector

Quantization and Fractal. This work is also aimed at trying to suggest a high performance algorithm. Our research work can be useful for researchers of different areas such as computer science, mathematical science, engineering etc.

1.7 LIMITATION OF THE STUDY

The limitation of the research work is that we would perform a comparative study, analysis and modelling of only one operation of image processing among the different operation such as Image Compression, Image Filtering, Image Enhancement, Image Restoration, Image Fusion, Image Registration, Image Segmentation, Image Transformations or Image Watermarking.

The limitation of this research work is that we would perform our study only for image compression. In this work we select only four image compression algorithms such as Wavelet, JPEG, Vector Quantization and Fractal.

The other limitation of this research work is that here we would perform the image compression only for selected images.

1.8 LITERATURE REVIEW OF THE STUDY

The main objective of the present study is: ***"To perform a comparative study and analysis of various image processing algorithms".***

The researcher had studied lots of related literature to identify the different works done in the related area of image processing and mainly image compression. The study had given us a more clear vision in the task. In this report, the brief information about the related research works for the present study is given.

Initial computers were designed just to do simple calculations. Researchers and scientists have been trying to endow the computer with more and more power and intelligence. That is to make the digital machine do things, which require human-like intelligence. Thus the term Artificial Intelligence (AI) was coined in the early 1950s. Gradually scientists came to know the immense power of the device and its limitations. More and more systems were developed as a consequence of incremental advances in computer science and engineering.

Nowadays computer systems are available for all applications such as from civil amenities to medical diagnosis; from home entertainment to space programs; from simple calculations to complex mathematical modelling, advertisement, commercial use, education, online selling and marketing, research etc.

Image processing is a physical process. It converts the image signal into the physical image. The image signal may be analog signal or digital signal. The real output of the process can be an actual physical image or the some characteristics of the image.

The general type of image processing is photography. In this process, the image is captured by using camera to create a digital or analog image. To produce or develop a physical image or picture, the appropriate technology based on the input source type is applied to the image.

In a digital photography, the image is stored in the computer system as an image file. The image file is translated by using photographic software to generate an actual or physical image. The shading and colors all are captured at the time of taking a photograph. The software then translates this information into the image.

When the image is created using analog photography, the image is burned into a film which uses a chemical reaction triggered by controlled exposure to light signal. This image is produced in a dark room, by using special chemicals for creating an actual or physical image. This process is not more popular due to the advent of digital photography, which requires less effort and special training to product images.

There are many operations, which are performed in image processing as per the requirement. This field of digital image has created many new applications and tools for image processing. These applications and tools were impossible previously. Medical image processing, biometric related applications and remote sensing, all are possible due to the development of digital image processing. Special computer programs and software are used to correct and enhance the images.

There are three major benefits of digital image processing:

- The consistent high quality of the image.
- The low cost of processing.
- The ability to manipulate all aspects of the process is all great benefits.

Day by day the speed of computer processing is continuously increasing. Also, the cost of storage memory and devices are continuously decreasing. As a result, the fields of image processing have grown to very high speed.

The applications of image processing are:

- Image processing (document and medical imaging, computer vision and industrial applications, Remote sensing and space applications, Military applications, etc….)
- Image Fusion (image classification, aerial and satellite imaging, medical imaging, robot vision, concealed weapon detection, etc…)
- Image transmission and storage applications (broadcast television, teleconferencing, etc…)
- Microscope image processing (medicine, biological research, cancer research, drug testing, etc…)
- Medical image processing
- Enhancement of cell images from microscope
- Image synthesis for cartoon making or fashion design
- Face detection
- Feature detection
- Geographical mapping
- Prediction of agricultural crops
- Prediction of urban growth

- Study pollution patterns from aerial and satellite imagery
- Flood and fire control
- Recognition and analysis of objects on images from space missions
- Robot vision for industrial automation
- Detection and recognition of various types of targets (radar or sonar images)
- Guidance of aircraft or missile systems

Image or data compression is one of the most important application areas of image processing. It reduces the redundancy of image data to store it efficiently. Multimedia data such as graphics, audio or video, which are uncompressed, need storage capacity and transmission bandwidth. Now days, there is rapid progress in mass storage density and digital communication system performance. The future of multimedia based web applications is data intensive, so we need to have efficient ways to encode signal and images.

Compression is achieved by the removal of one or more of three basic data redundancies:

(i) **Coding Redundancy:** This is present when less than optimal (i.e. the smallest length) code words are used.

(ii) **Inter Pixel Redundancy:** This result from correlations between the pixels of an image.

(iii) **Psycho Visual Redundancy:** This is due to data that is ignored by the human visual system.

The techniques used to compress or decompress a single gray level image are expected to be easily modified to encode or decode color images and image sequences. The goal of image compression is to save storage space and to reduce transmission time for image data. It aims at achieving a high Compression Ratio (CR) while preserving good fidelity of decoded images. Recent compression methods can be briefly classified into following four categories:

i. Wavelet
ii. JPEG/DCT
iii. VQ
iv. Fractal Methods

These Compression Methods are discussed by different peoples are briefly reviewed as below.

Chaur-Chin Chen

In this study he has only reviewed and summarized the characteristics of four up-to-date image coding algorithms based on Wavelet, JPEG/DCT, VQ, and Fractal approaches. Experimental comparisons on four 256×256 commonly used images, Jet, Lenna, Mandrill, Peppers, and one 400×400 fingerprint image suggest a recipe described as follows. Any of the four approaches is satisfactory when the 0.5 bits per pixel (bpp) is requested. Hence for practical applications, he concluded that wavelet based compression algorithms are strongly recommended [2].

Chin-Chen Chang

In this study the proposed model does the data embedding into the cover image by changing the coefficients of a transform of an image such as discrete cosine transform. The high compression rate is one of the advantages of fractal image compression. Main advantage is the good image quality, after enough iteration for decompression. But the computation time required to encode an image might be very long due to an exhaustive search for the optimal code. And DES encryption is used to provide the security to the data, but it is unable to protect the Stego-Image from subterfuge attack. Which is nothing but, the attacker not only detect a message, but also render it useless or even worse, modify it to the opponent's favour[3].

Chin-Chen Chang

In this study he proposed a scheme to embed an image compressed via fractal image compression into the DCT domain of the cover image. Due to the high compression rate of fractal compression, also it can embed a secret image larger than the cover image itself. Moreover, the more decompression iterations will be done, the better decompressed secret image quality will get. Also, these compression codes of fractal compression must not be lost, or the embedded message cannot be extracted. Thus some modification on the bit streams of the modified coefficients to prevent the information loss caused by discrete cosine transformation is needed. As for security, the compressed data should be encrypted via DES so that it can be prevent the eavesdroppers from getting the secret image [3].

Sachin Dhawan

According to him, the hybrid VQ algorithm is explained as follows; first the correlation in the test image is taken away by wavelet transform. Wavelet transform is employed in the course of lifting scheme. The wavelet employed during lifting scheme is HAAR wavelet. Then it will lead with the primary level of decomposition and gives way four components namely LL, LH, HL and HH correspondingly. Multistage VQ is implemented to LL band. The pointed coefficients in the sub band LH, HL and HH are pyramidal vector quantized by capturing vectors of stated measurement. Finally an entropy coding algorithm like Huffman coding is implemented as the final stage of the compression system to code the indices. He proposed that an algorithm groups the benefit of lessening of codebook searches and storage complexity which is intrinsic to MSVQ. Additionally, pyramid vector quantization may not need great codebook storage having simple encoding and decoding algorithm. Hence high compression ratio can be accomplished by including PVQ along with MSVQ [4].

K. B. Raja

In this study the proposed model which uses LSB has been focused upon. But LSB provides poor security and DCT for converting objects in spatial domain to frequency domain. This model uses only raw images because of subterfuge attack. In the JPEG, BMP and GIF image formats, the header contains most of the image information. This leads to the problem of insecurity and therefore the payloads from such images can be easily identified [5].

Hairong Qi

In this study the proposed model the technique BCBS (Blind Consistency Based Steganography) is used. The advantage of the BCBS approach is that the decoding process can be operated blindly without access to the cover image, which enhances the imperceptibility and that it not only decodes the message exactly, but it also detects if the message has been tampered without using any extra error correction. The main drawback is that the amount of data to hide is very less because he has selected only one row or column to hide with one bit per pixel. It also provides the security only for the Stegano-Image not for the data that can be improved by using DES [6].

Niels Provos

In this study it is found that the PSNR (Peak Signal to Noise Ratio) to be a good indicator of finger and face recognition matching scores in the case of JPEG2000 and SPIHT. Both wavelet based algorithms perform exceptionally well in terms of rate-distortion performance and matching scores of all recognition systems considered. While PSNR exactly predicts the poor matching scores of fractal compression in the case of fingerprint images, the relatively high PSNR results for face images suggest fractal compression to perform superior to JPEG for this biometric modality. The opposite is true – despite the low PSNR results, JPEG performs quite well in face recognition applications for high and medium bit rate applications [7].

N. F. Johnson

In this study he brought the Tree Structured VQ techniques in order to decrease the table storage needed for encoding and decoding along with unstructured vector quantization (UVQ) or Tree-Structured Vector Quantization (TSVQ). Particularly, a low storage secondary quantizer is employed to squeeze the code vectors of the primary quantizer. The absolute benefits of uniform and non-uniform secondary quantization are examined. Quantization levels are put up on a binary or quad tree structure (suboptimal) [8].

It is to set vectors in different quadrant. Signs of vectors are only needed to be evaluated. This will reduce the number of links by 2^L for L-d vector problem. It will work fine for symmetric distribution [8].

1.9 PLANNING OF THE REMAINING CHAPTERS

The thesis entitled "**A Comparative Study, Analysis and Modelling of Various Image Processing Algorithms"** contains mainly six sections. It is organized as follows:

Chapter – 1: Research Survey and Introduction

In this section of the thesis overview of research work has been provided. To be particular, in this chapter, problem domain, selection of research title, survey of research, objective of study, scope of the study and planning of the remaining chapters have been highlighted.

Chapter – 2: Introduction to Image Processing Technology

In this section of the thesis, various types of images and image processing technology have been discussed. Also, various types of images, image representation formats and fundamental study related to image processing have been highlighted. We have also discussed (1) image enhancement and restoration such as color models, chromaticity diagram, histogram processing, negation processing, spatial and domain filters, brightness enhancement, contrast enhancement, thresholding, gamma correction, highlight and shadow, edge detection, (2) Geometric transformation such as scale, resize, zoom, rotation, stitching, copy, cut, paste, curve and surface fitting, (3) Image analysis such as skew detection and correction, OCR, ICR and OMR. Additionally, image registration and image segmentation are also highlighted in this chapter.

Chapter – 3: Lossless and Lossy Image Compression Techniques

In this section of the thesis, the basic concepts related to image compression process have been highlighted. Moreover, various lossless and lossy image compression algorithms are studied. A lossless compression algorithm includes Run Length Encoding (RLE), Huffman Encoding, Lempel-Ziv-Welch (LZW) Coding and Area Coding. A lossy compression algorithm includes Transform Coding, Vector Quantization, Fractal Coding, Block Truncation Coding, Sub band Coding, Segmented Image Coding, Spline Approximation Method.

Chapter – 4: Experimental Studies and Comparison of Various Image Compression Algorithms

In this section of the thesis, we have performed study of various image compression algorithms. Also, various lossless image compression algorithms are Wavelet and Vector Quantization and lossy image compression algorithms are JPEG/DCT and Fractal are performed on various images. Wavelet based image compression are performed in Haar Wavelet, Daubechies (db4) Wavelet, Biorthogonal Wavelet, Symlets Wavelet and Coiflet Wavelet.

Chapter – 5: Image Compression Model

In this section of the thesis, we have presented the image compression model. In this model three levels of image compression are performed such as lowest compression, average compression and highest compression. Here we set the 10% compression as lowest level, 50% compression as average level and 100% compression as highest level. In this model many other operations are also performed on images.

Chapter – 6: Contributions of the Research Work and Suggested Future Work

In this section of the thesis, all research work conclusions have been highlighted. This chapter shows the comparison between various image compression algorithms and also displays the result of the comparison. Finally the conclusion of the study and suggestions for the future work are also highlighted.

CHAPTER – 2

INTRODUCTION TO IMAGE PROCESSING TECHNOLOGY

2.1 INTRODUCTION TO IMAGE

An image may be defined as a two dimensional function *f(x,y)*, where *x* and *y* are spatial or plane coordinates, and the amplitude of *f* at any pair of coordinates *(x, y)* is called intensity of the image at that point.

A digital image is an image *f(x, y)* that has been discredited both in spatial coordinates and brightness.

The elements of such a digital array are called image elements or pixels.

A Pixel is Pix Element or Picture Element. It refers to the representation of a single point within an image. A pixel may be simply a bit (for black & white images), or a much larger data structure.

2.2 IMAGE REPRESENTATION FORMATS

Nowadays images are very important documents for everyone. To work with the image in some applications, it needs to be compressed, more or less depending on the purpose of the different application. There are some algorithms that perform this compression in different ways; some are lossless and keep the same information as the original image, some others loss information when compressing the image. Some of these compression methods are designed for specific kinds of images, so they will not be so good for other kinds of images. Some algorithms even let you change parameters they use to adjust the compression better to the image.

There are main two different types of image representations formats, which are as follows:

- Lossless Image Representation Formats
- Lossy Image Representation Formats

2.2.1 Lossless Image Representation Formats

There are mostly two lossless image file representations formats, which are as follows:

2.2.1.1 TIFF

Tagged Image File Format (TIFF) is a very flexible file format. It can be lossless or lossy. TIFF is used almost exclusively as a lossless image storage format [9].

This file format is used mainly for storing images, including photographs and line art. It is one of the most popular and flexible current public domain raster file formats.

This file format was created by the Aldus Company, jointly with Microsoft, for use with PostScript printing. This format is a very popular file format for high color depth images. It is widely supported by image manipulation applications, by scanning, faxing, word processing, optical character recognization, and other applications [9].

This image file format is not used with web images because they produce big files, and its large amount requires much storage space on web. Most web browsers will not display these file formats.

2.2.1.2 PNG

Portable Network Graphics (PNG) Development Group has created PNG file format in year 1995 as an alternative to the Graphics Interchange Format (GIF). It is a bitmap image file format. It employs lossless data compression. It was mainly created for improvement and replacement of GIF file format.

PNG file format supports palette based, gray scale and RGB images. PNG was designed for distribution of images on the internet not for professional graphics and such other color spaces.

PNG is mainly used in the following two applications:

- If you have an image with large areas of exactly uniform color, but contains more than 256 colors. PNG is very useful at that time. It is similar to the GIF, but it supports 16 million colors, while GIF support only 256 colors.
- If you want to display a photograph exactly without loss on the web, it is very useful for that purpose. Most web browsers support the PNG file format. It is the only lossless format which is supported by web browsers.

It is superior to the GIF file format. It produces small files in size and also allows more colors. It also supports partial transparency. Partial transparency can be used for many purposes, such as fades and antialiasing of text.

In GIF compression, the first step is to "index" the image's color palette. It decreases the number of colors in your image to a maximum of 256 (8-bit color). The smaller number of colors in the palette, the greater the efficiency of the algorithm is generated.

2.2.2 Lossy Image Representation Formats

There are mostly two lossy image file representations formats, which are as follows:

2.2.2.1 JPEG

Joint Photographic Experts Group (JPEG) is an algorithm designed to compress images with 24 bits depth or gray scale images. It is a lossy compression algorithm. It is a very flexible algorithm. One of the characteristics of this algorithm is that the compression rate is adjustable. That means the user can adjust the compression ratio as per his/her requirement. If we compress on a high ratio, the quality of image will be low and more information will be lost from image, but the result image size will be smaller. With a smaller compression rate we obtain a better quality of image, but the size of the resulting image will be larger [9].

This compression consists in making the coefficients in the quantization matrix bigger when we want more compression, and smaller when we want less compression. The algorithm is based in two visual effects of the human visual system [9].

- Humans are more sensitive to the luminance than to the chrominance.
- Humans are more sensitive to changes in homogeneous areas, than in areas where there is more variation (higher frequencies).

JPEG is the most used format for storing and transmitting images in Internet [9].

2.2.2.2 JPEG 2000

Joint Photographic Experts Group 2000 (JPEG 2000) is a wavelet based image compression standard. It was created by the Joint Photographic Experts Group committee with the intention of superseding their original Discrete Cosine

Transform (DCT) based JPEG standard[9].

JPEG 2000 has higher compression ratios than JPEG. It does not suffer from the uniform blocks, so characteristics of JPEG images with very high compression rates [9].

But it usually makes the image more blurred than that by JPEG.

2.2.3 Other Image Representation Formats

There are some other image file representations formats, which are as follows:

2.2.3.1 GIF

Graphics Interchange Format (GIF) is a very widely known image file format. It was created by CompuServe in 1987. It was revised in 1989. The main disadvantage of this file format is that, it used limited number of colors (256 color table) and saves using 8 bit quality only. It allows saving images with only a few colors very efficiently.

It is often picked for images which are to be displayed on web pages that involve transparency or image animation. It is also the only format absolutely universally understood by all web browsers.

The GIF format can save multiple images to form an animation sequence, and for this purpose also saves the image canvas size and page information.

This file format is well known to be good for graphics that contain text, computer generated art, large areas of solid color, small images of cartoons, line drawing, small icons, etc…

2.2.3.2 RAW

RAW is an image output option available on some digital cameras. It produces the very small file size than the TIFF files formats. It creates the factor of three or four smaller than TIFF files of the same image. The main disadvantage of this file format is that there is a different RAW format for each manufacturer. So you have to require that particular manufacturer's software to view the images. Some graphics applications can be read by some manufacturer's RAW formats only.

2.2.3.3 BMP

The BMP (Microsoft Windows Bitmap) is a bitmapped graphics file format used internally by the Microsoft Windows graphics subsystem. It is used commonly as a simple graphics file format on that platform. This file format is a most commonly used file format on IBM PC Compitable computers. It is an uncompressed format [9].

2.2.3.4 PSD / PSP

It is a Photoshop's native file format and sometimes it is also called PDD file format. It is a widely accepted file format. It supports all available image modes such as RGB, Gray scale, Bitmap, CMYK, indexed color, Multichannel, etc…

Photoshop's files are saved as using PSD extension of image file, while Paint Shop Pro files use PSP extension of image file. These are proprietary image file formats used by graphics programs. These are the preferred working formats. It is proprietary file format, so you can edit images in the particular software only, and it has all the editing power of the programs. These packages use layers. If you use non-proprietary file format such as JPG and TIFF to build complex images, at that time it may be lost of same layer information.

2.3 INTRODUCTION TO IMAGE PROCESSING

In the field of computer science and electrical engineering, image processing is any form of signal processing. In which the input is an image, such as photographs or frames of video. The output of image processing can be either an image or a set of characteristics or parameters related to the image.

2.4 IMAGE PROCESSING – AN OVERVIEW

Image processing is an important component of modern technologies because human depends so much on the visual information than other creatures. Image is better than any other information form for us to perceive. Image processing has traditionally been an area in engineering community. The basic tools are Fourier analysis with a long history and wavelet analysis which has become popular in the engineering community since 1980's. In the past a few decades ago several advanced mathematical approaches or statistical methods were applied to image processing and it has now become an important tool for theoretical image processing.

Digital image processing is an ever expanding and dynamic area with applications reaching out into our everyday life such as medicine, space exploration, surveillance, authentication, automated industry inspection and many more areas. Applications such as these involve different processes like image compression, registration, filtering, segmentation, fusion, enhancement or object detection.

Implementation of these types of applications on a general purpose computer can be very easy, but not every time efficient due to additional constraints on memory and other peripheral devices. Application specific hardware implementation offers much greater speed than a software implementation. With advances of the VLSI (Very Large Scale Integrated) technology hardware implementation has become an attractive alternative.

A digital image is a discrete two-dimensional function, *f(x,y)*, which has been quantized over its domain and range. Without loss of generality, it will be assumed that the image is rectangular, consisting of *Y* rows and *X* columns. The resolution of such an image is written as *X x Y*. By convention, *f(0, 0)* is taken to be the top left corner of the image, and *f(X-1, Y–1)* is the bottom right corner. This is represented in the figure given below [10]:

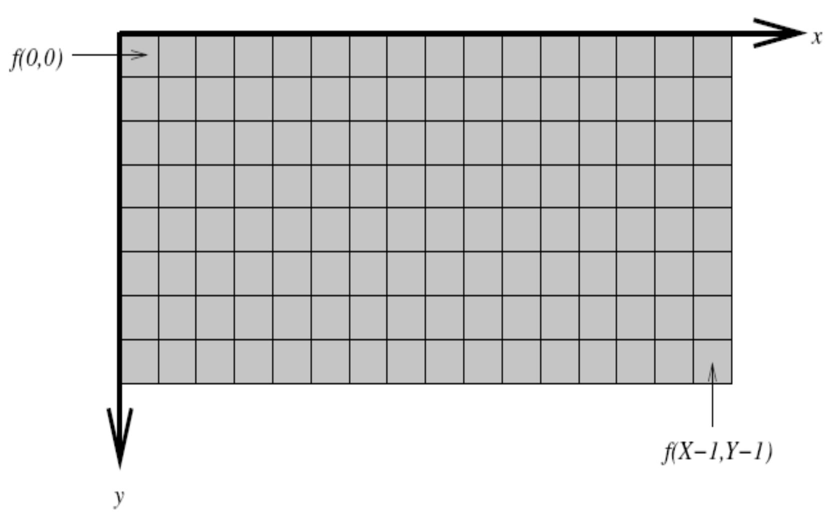


Figure – 2.1 A rectangular digital image of resolution 16 X 8 [10].

Each distinct coordinate in an image is called a pixel, which is short for “picture element”. The nature of the output of f(x, y) for each pixel is dependent on the type of image. Most images are the result of measuring a specific physical phenomenon, such as light, heat, distance or energy. The measurement could take any numerical form [10].

A gray scale image measures light intensity only. Each pixel is a scalar proportional to the brightness. The minimum brightness is called black, and the maximum brightness is called white. A typical example is given in below figure [10].

Figure – 2.2 Lena gray scale image of resolution 512 X 512

A color image measures the intensity and chrominance of light. Each color pixel is a vector of color components. Common color spaces are RGB (Red, Green and Blue), HSV (Hue, Sturation, Value), and CMYK (Cyan, Magenta, Yellow, Black), which are used in the printing industry. Pixels in a range image measure the depth of distance to an object in the scene. Range data is commonly used in machine vision applications [11] [12].

For storing purposes, pixel values need to be quantized. The brightness in gray scale images is usually quantized to Z levels, so f (x, y) Є {0, 1... Z–1}. If Z has the form 2^L the image is referred to as having L bits per pixel. Many common gray scale images use 8 bits per pixel, giving 256 distinct gray levels. Medical scans often use 12 – 16 bits per pixel, because their accuracy could be more important. Those images to be processed predominantly by machine may often use higher values of Z to avoid loss of accuracy throughout processing. Images not encoding visible light intensity, such as range data, may also require a larger value of Z to store sufficient distance information [10].

There are many other types of pixels. Some measure bands of the electromagnetic spectrum such as infrared or radio, or heat, in the case of thermal images. Volume images are actually three dimensional images, with each pixel being called a voxel. In some cases, volume images may be treated as adjacent of two dimensional image slices [10].

2.4.1 Components of Image Processing

The basic components of image processing system are as follows:

- In the simplest case, this image is capture using a CCD camera, a flatbed scanner, or a video recorder in an image acquisition system.
- Image captured from image acquisition system is converted into digital image and is stored. The device is used for the process known as frame grabber.
- The digital image is processed by using a personal computer or a workstation.
- Image processing software provides the tools for manipulates and analyzes the images.

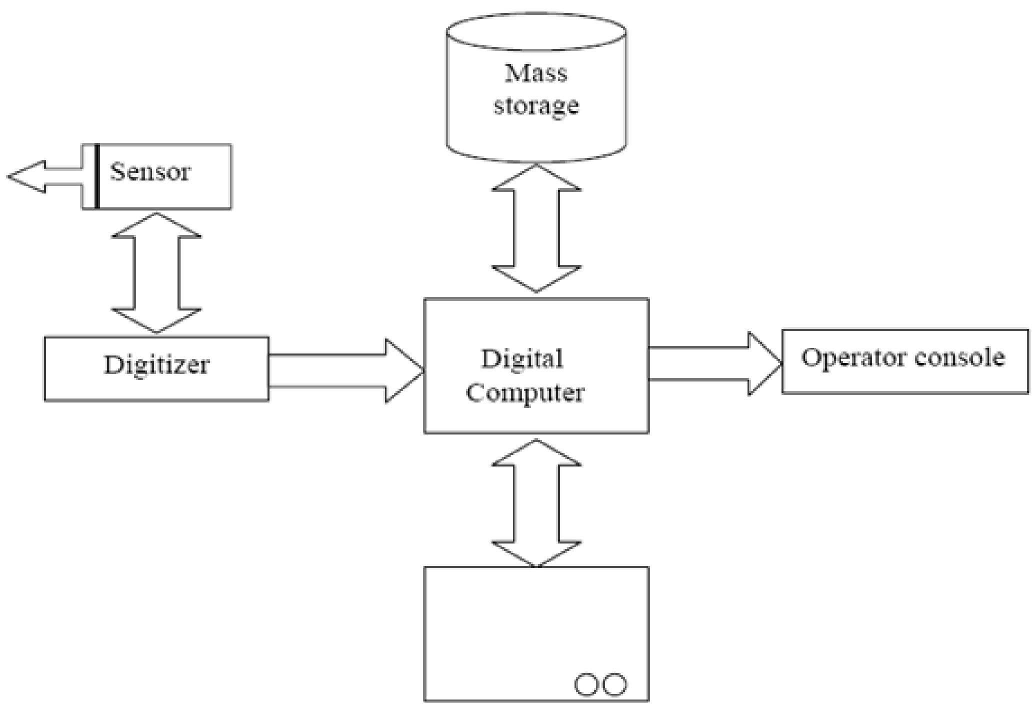


Figure – 2.3 General purpose image processing system

2.4.2 Operations Performed on Image Processing

The following operations can be performed on various types of images.

- Image Compression
- Image Registration
- Image Segmentation
- Image Filtering
- Image Enhancement
- Image Restoration
- Image Fusion
- Image Transformation

2.4.3 Application Areas of Image Processing

In a digital image processing, there are mainly two application areas:

- Improvement
- Processing

Improvement: - Improvement of pictorial information for human interpretation.

Processing: - Processing is screen data for autonomous machine perception.

In processing application area, interest focuses on procedures for extracting from image information in a form suitable for computer processing.

Examples include automatic character recognization, industrial machine vision for product assembly and inspection, military recognizance, automatic processing of fingerprints etc.

2.4.4 Fundamental Steps of Image Processing

The following steps are to be performed in image processing:

- Image Acquisition
- Image Preprocessing
- Image Segmentation
- Image Representation
- Image Description
- Image Recognization
- Image Interpretation

Image Acquisition: To acquire a digital image from the various sources.

Image Preprocessing: To improve the image in ways that increases the chances for success of the other processes.

Image Segmentation: To part an input image into its constituent part or objects.

Image Representation: To convert the input data to a form suitable for computer processing.

Image Description: To extract features from images and the result in some quantitative information of interest or features that are basic for differentiating one class of objects from another.

Image Recognization: To assign a label to an object based on the information provided by its descriptors.

Image Interpretation: To assign meaning to an ensemble of recognized objects.

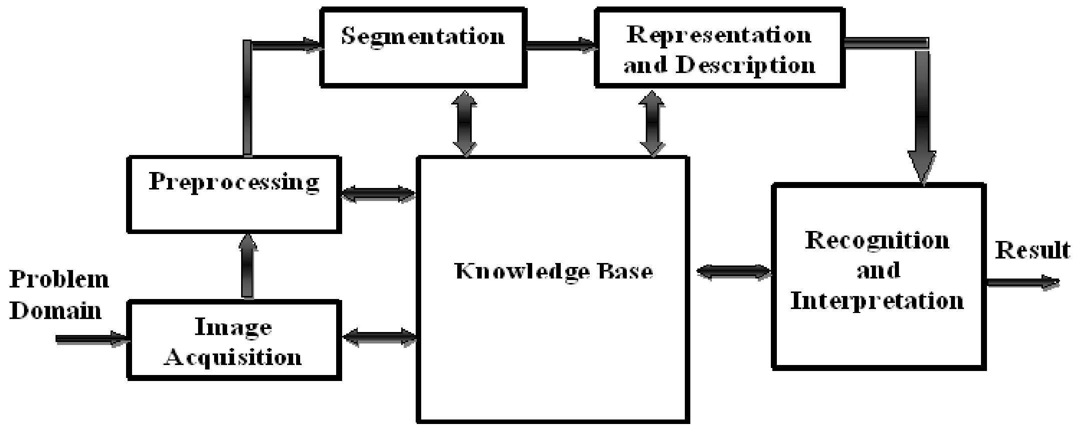


Figure – 2.4 Fundamental steps of image processing

2.4.5 Elements of Image Processing System

The following are basic operations to be performed in a digital image processing systems:

- Image Acquisition
- Image Storage
- Image Processing
- Image Communication
- Image Display

Image Acquisition: Image is acquired from various sources. Various image acquisition devices are used such as camera, scanner, video, etc…

Image Storage: Acquired Images are stored in various data storage devices such as Optical disks, Tap, Video Tape, Magnetic disk, etc…

Image Preprocessing: To improve the image in image processing unit such as computer, workstation, etc…

Image Communication: Various communication channels are used for communication and data transfer in image processing system. Cables are used as communication channel in computer system.

Image Display: Display units are used to display or represent the data or image. Display units are TV Monitos, Printers, Slide Projectors, etc...

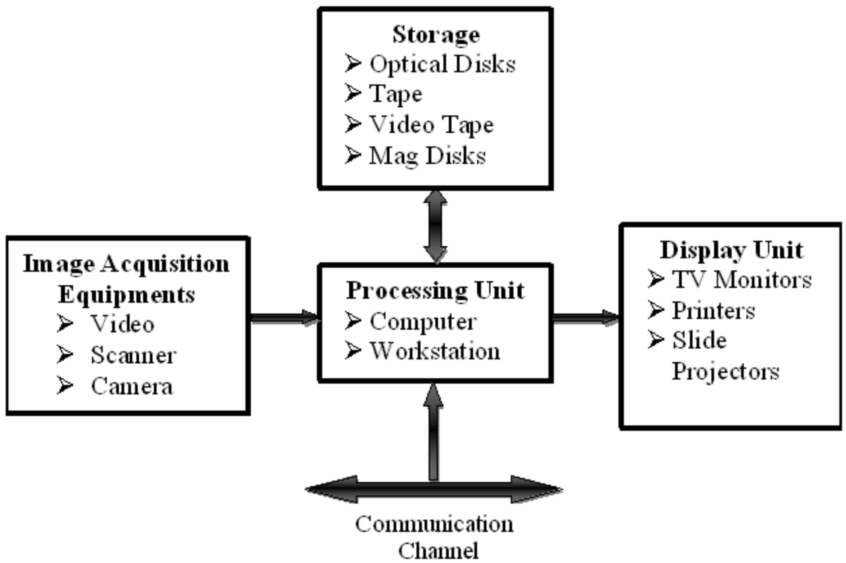


Figure - 2.5 Elements of image processing

2.5 IMAGE PROCESSING TECHNOLOGY

Image Processing is the most important component of any Document Image Management or Workflow System. This area of the software technology, as the name suggests, deals with enhancement and manipulation of the digitized images primarily to extract, refine and evaluate information. Development of image processing software involves a broad range of practical and theoretical understanding and knowledge. The image processing tasks or operations are extremely taxing of computer power at least in two respects [13].

First, the amount of raw data in an image is very high; for instance a visibly small image of 1024 X 1024 eight-bit pixels requires 1 MB of storage in its simplest form.

Second, it is computationally expensive, with typical non-trivial image processing tasks requiring between 100 and 10,000 machine cycles per pixel. Unless the design and implementation of the image processing algorithms are fully optimized, the software is likely to run unacceptably slowly due to inherent complexities of the image processing tasks.

In a typical document management system, image processing tasks begin at the time of image acquisition and continue to be used throughout the lifetime of the document image. Image processing operations are performed not only when the image is in the primary memory or is being displayed, but also each time it is being archived, retrieved, and modified or analyzed [13].

The most basic image processing software solutions are available in the form of API functions that belong to various libraries. We have developed 16-bit and 32-bit Dynamic Link Libraries that are collectively termed as NIPL (Newgen Image Processing Libraries). For a core image processing function, input is always an image or a sequence of images along with other control parameters, and output is essentially an image or a sequence of images and/or information extracted from the input image. The image processing functions might affect pixel-depth, color shades, dimensions, file format, compression scheme, and other attributes of the input image, depending upon the type of operation desired.

All the image processing tasks can be widely categorized in the following four groups from the viewpoint of Software

Development [13]:

- Image Enhancement and Restoration
- Geometric Transformations
- Image Analysis
- File Formats and Compression Schemes

2.6 IMAGE ENHANCEMENT AND RESTORATION

Image enhancement refers to those image processing operations that improve the quality of input image in order to overcome the weakness of the human visual system. These operations in turn enable more information extraction when the output is presented to a human subject or an algorithm for further processing. Image restoration uses similar image improvement techniques mainly to correct errors introduced during acquisition and transmission of the images [13].

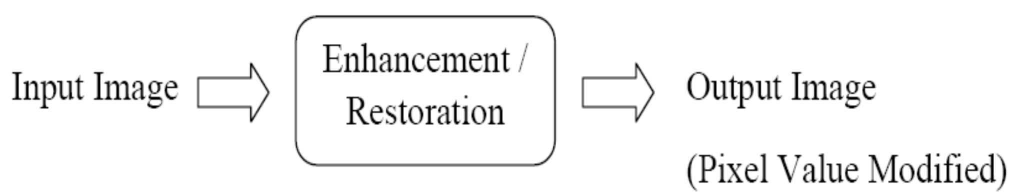


Figure – 2.6 Image enhancement or restoration [13]

There is an extensive set of image enhancement and restoration functions available in NIPL. The following basic components are commonly used to subgroup the tasks:

2.6.1 Color Models

The purpose of a color model is to facilitate the specification of colors in some standard. The various types of color models are used. Generally RGB, YcbCr, HSI, and CMY are used. It can convert the image in one color model to another one [13].

2.6.1.1 RGB Model

In this model each color appears in its primary spectral components of Red, Green and Blue. Images in the RGB Model consist of three independent image planes which combine to form a composite color image. Most color digitizers are used for acquiring digital images and devices for displaying images utilize the RGB format, making the most widely used model. This is the simplest color model and is also used in processing multi-spectral image data [13].

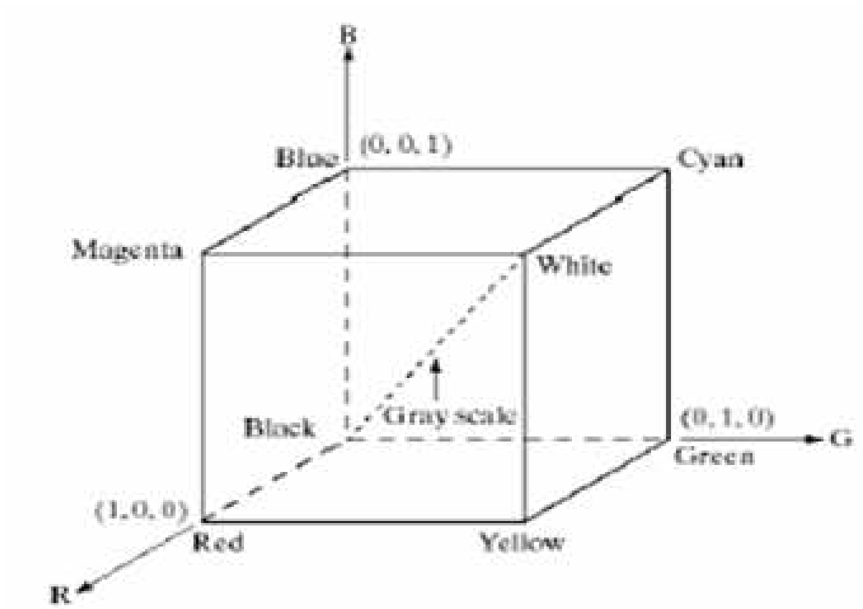


Figure – 2.7 Schematic of the RGB color cube

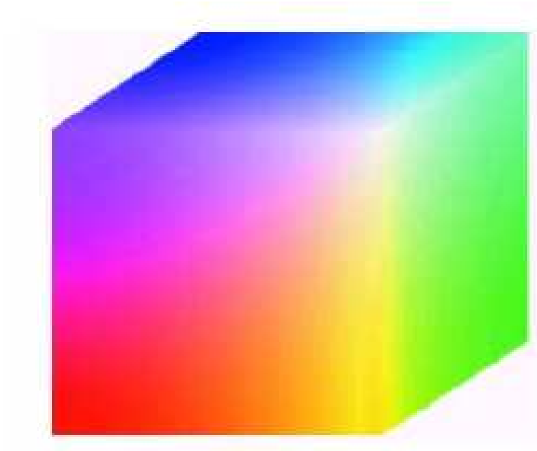


Figure – 2.8 24-bit RGB color cube

The RGB color model relates very closely to the way we perceive color with the r, g and b receptors in our retinas. RGB uses additive color mixing. It is the basic color model used in television or any other medium that projects color with light. It is the basic color model used in computers and for web graphics, but it cannot be used for print production [13].

The secondary colors of RGB – cyan, magenta, and yellow – are formed by mixing two of the primary colors (red, green or blue) and excluding the third color. Red and green combine to make yellow, green and blue to make cyan, and blue and red form magenta. The combination of red, green, and blue in full intensity makes white [13].

In a different layers of the image, it will make the intensities mix together according to the additive color mixing model. This is analogous to stacking slide images on top of each other and shining light through them.

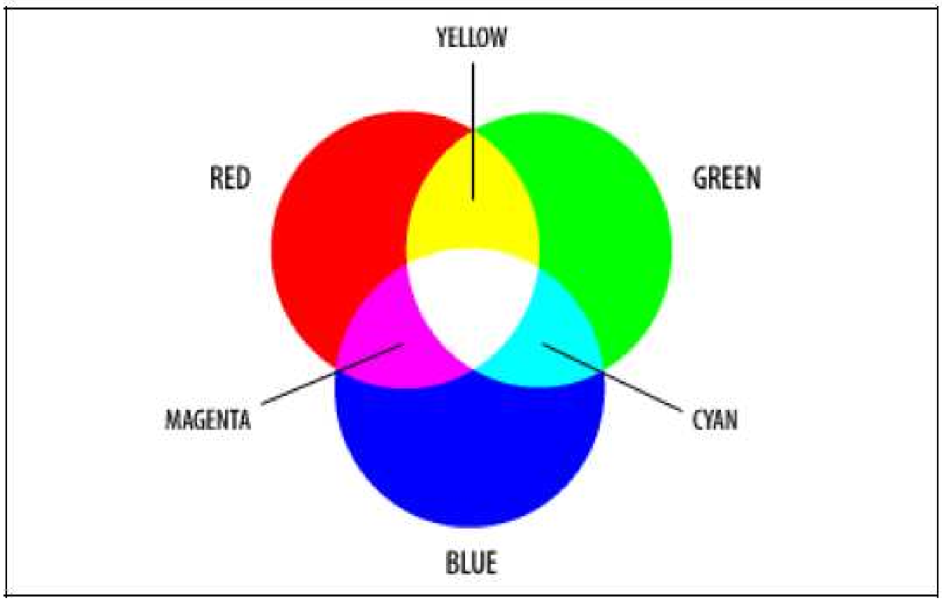


Figure – 2.9 Additive model of RGB

2.6.1.2 YCbCr Model

This model is characterized by Luminance (Y) and color information determined by Chrominance (Cb, Cr). Luminance is proportional to the amount of light perceived by human eye. The important feature in this model is that the color information is decoupled to the luminance. As a result it is useful in maintaining compatibility between color and monochrome TV broadcasting standard by changing the luminance, which does not alter the color content of the image [13].

2.6.1.3 HSI Model

Hue, Saturation, and Intensity model is useful for following principal fact.

- The Intensity component is decoupled from the color information in the image.

The Hue and Saturation components are intimately related to the way the human being perceive color [13].

In this system Hue is the color as described by wavelength, for instance distinction between red and yellow. Saturation is the measure by which the pure color is diluted by white light. Intensity is the amount of light present, for example distinction between a dark red and light red or a dark gray and light gray. HSI model has a wide range of usefulness in design of imaging systems and color matching.

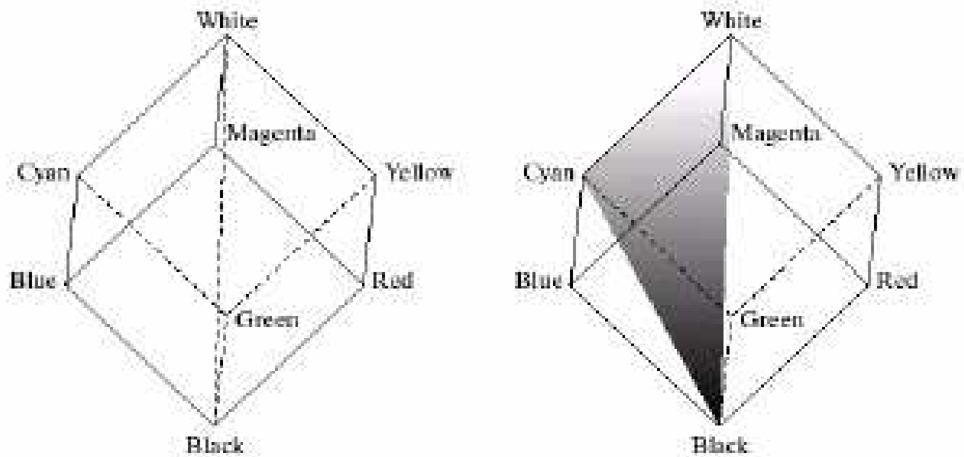


Figure – 2.10 Conceptual relationships between the RGB and HIS color model

HSI model can be obtained from the RGB model. The diagonal of the joining Black and White in the RGB cube is the intensity axis. HSI color model based on a circle are shown in the figure given below. The circle and the triangle are perpendicular to the intensity axis.

In the figure given below, the dot is an arbitrary color point. The enable from the red axis gives the Hue, and the length of the vector is the Saturation. The intensity of all colors in any of these planes is given by the position of the plane on the vertical intensity axis.

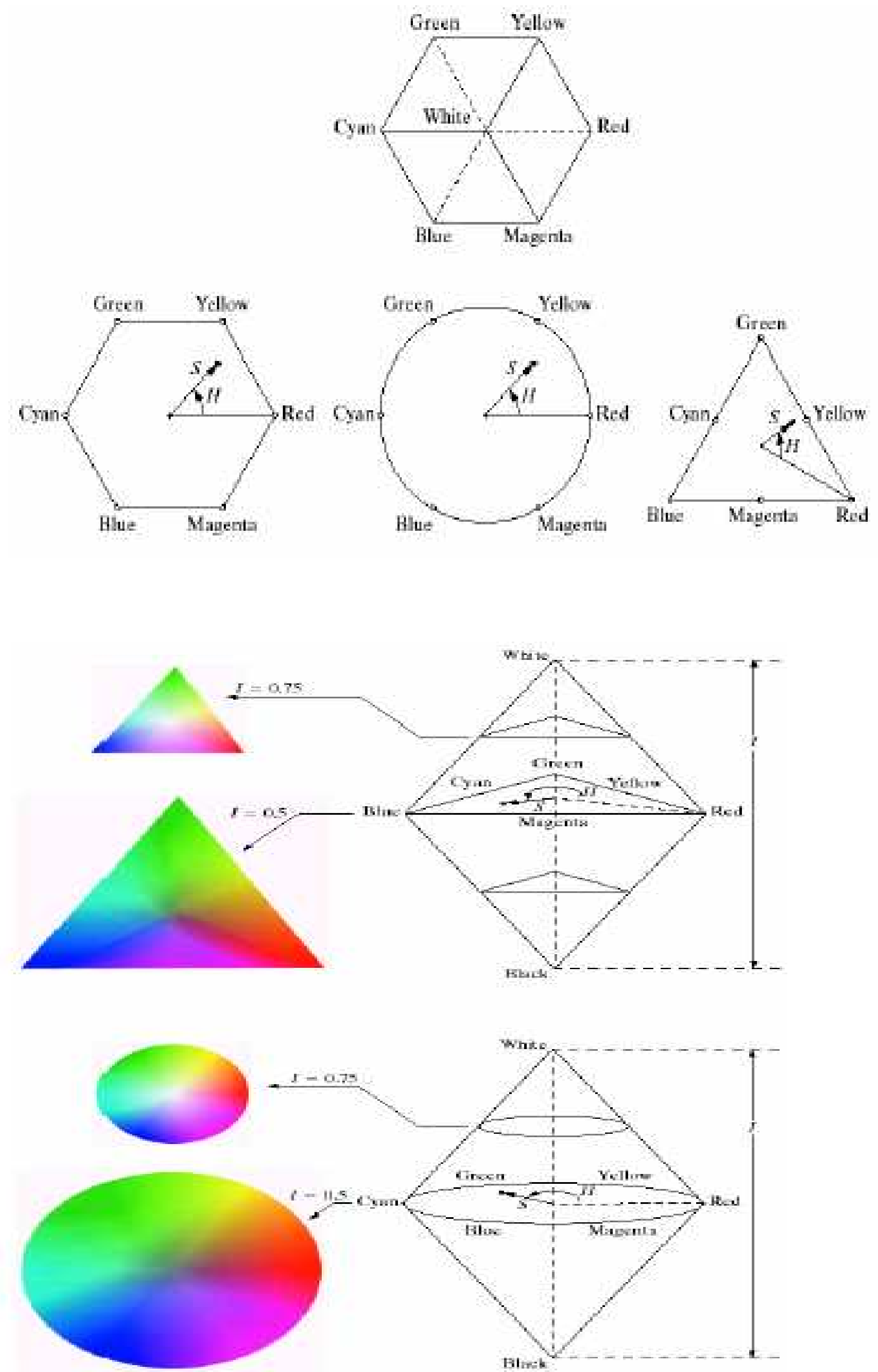


Figure – 2.11 Hue, Saturation in the HSI color model

2.6.1.4 CMY and CMYK Model

This model is made up of three secondary colors of light - Cyan, Magenta and Yellow. This is the reverse of RGB model as it does not emit Red, Green or Blue light when illuminated under white light. CMY data input is required in color printers and photocopiers that often use one more color Black (K), thus making it a CMYK system. Additional black is required since CMY alone cannot produce a true black and rather gives a muddy grayish brown in practical printing, owing to impurities in the inks [13].

The 4-color CMYK model used in printing lays down overlapping layers of varying percentages of transparent cyan (C), magenta (M) and yellow (Y) inks. In addition a layer of black (K) ink can be added. The CMYK model uses the subtractive color model.

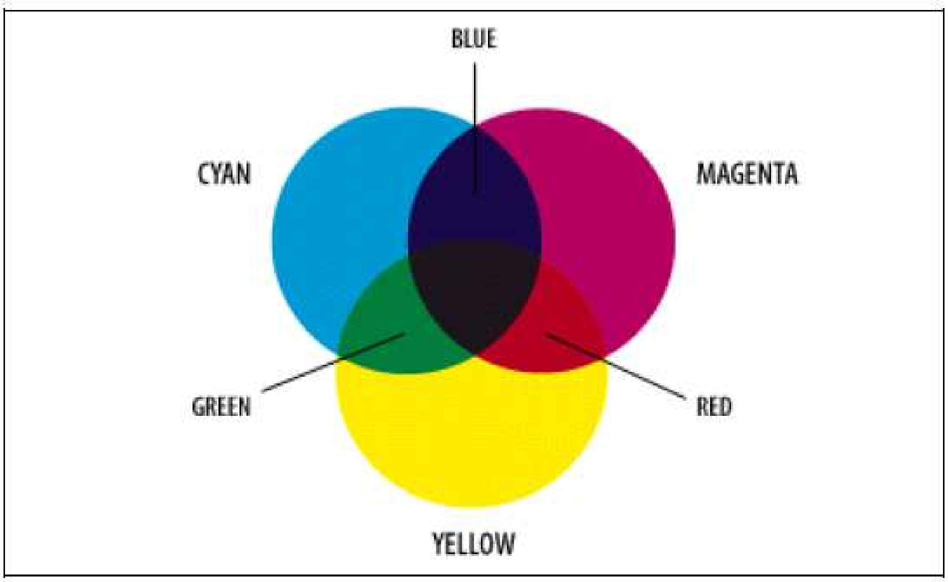


Figure – 2.12 Subtractive model of CMYK

Subtractive color space that relates to RGB is represented as follows:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

It will produce black color when

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

2.6.2 Chromaticity Diagram

This chromaticity diagram shows color composition as a function of x and y. The triangle in the diagram given below shows the color gamut for a typical RGB system plotted as the XYZ system.

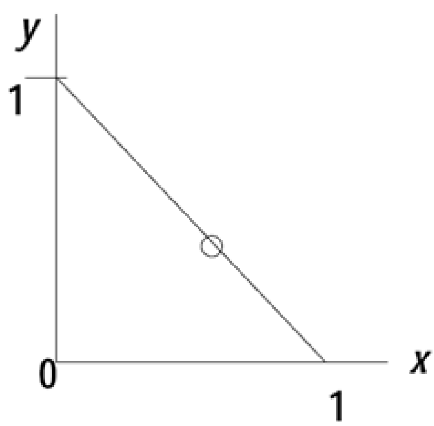


Figure – 2.13 Color gamuts for a typical RGB system

The axes extend from 0 to 1. The origin corresponds to BLUE. The extreme points on the axes correspond to Red and Green. The point corresponding to x = y = 1/3 (marked by the spot) corresponds to White.

The positions of various spectrum colors from violet to red are indicated around the boundary. These are pure color. Any inside point represents mixture of spectrum colors. A straight line joining a spectrum color point to the equal energy point shows all the different shades of the spectrum color.

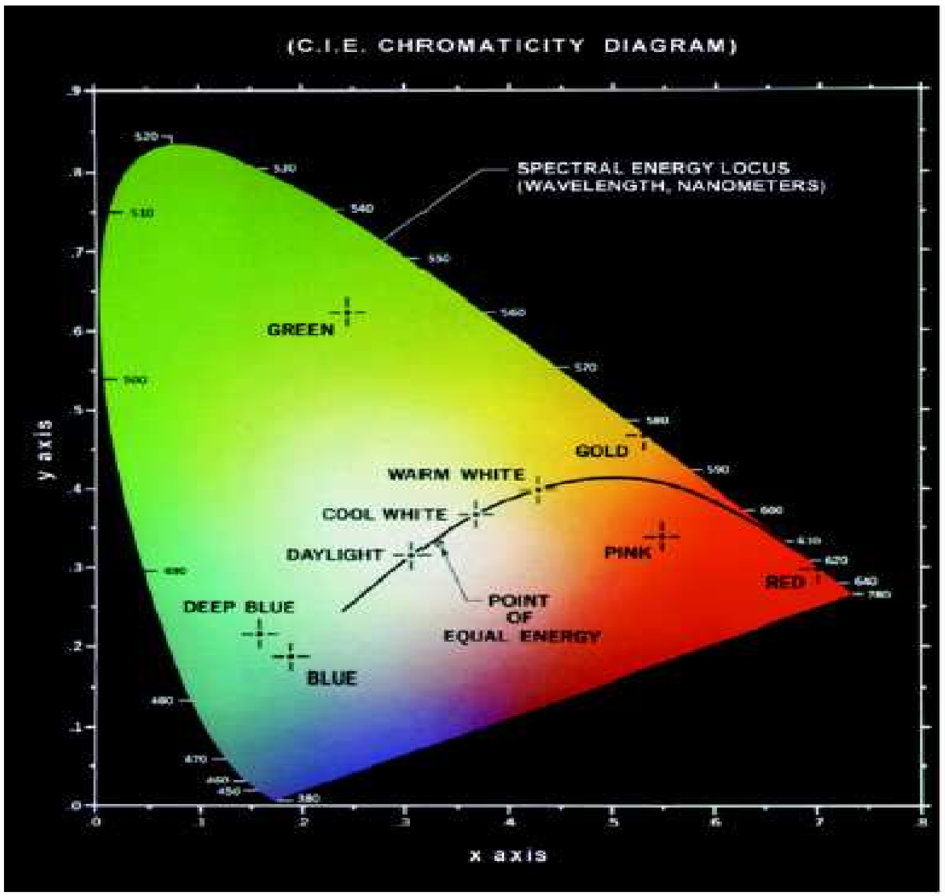


Figure – 2.14 Chromaticity diagram

Any color in the interior of the “Horse Shoe” can be achieved through the linear combination of two pure spectral colors. A straight line joining any two points shows all the different colors that may be produced by mixing the two colors corresponding to the two points. The straight line connecting Red and Blue is referred to as the line of

Purples. RGB primaries form a triangular color gamut. The white color falls in the center of the diagram, which is represented clearly in the figure given below:

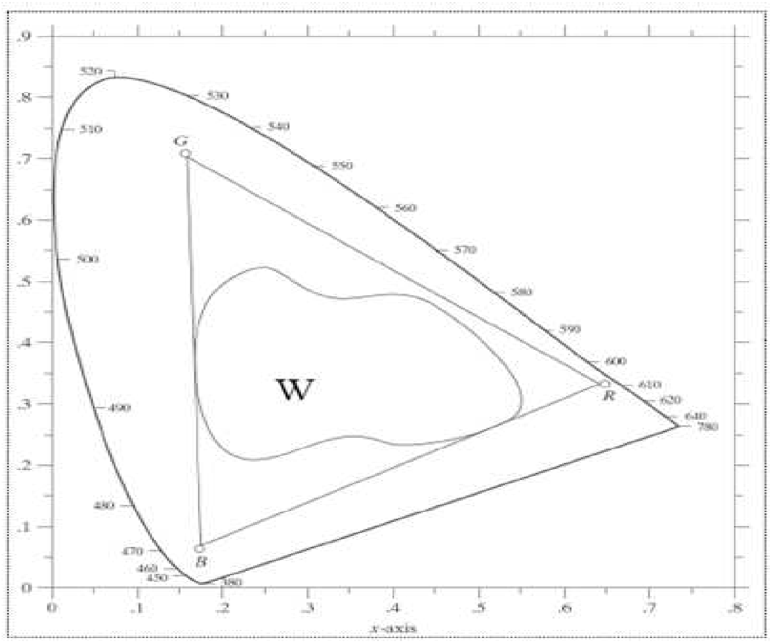


Figure – 2.15 Chromaticity diagram with triangle

2.6.3 Color Quantization

Color Quantization is the reduction in the number of colors in a color image depending on the importance of each color in describing the image. This technique can be used to scale down the pixel-depth of, say, a true color image (which supports 1 million colors) to a fixed color image supporting 256 or 16 colors [13].

Quality consideration is an important aspect of color quantization algorithms as it involves selection of most important colors. This process also finds application in displaying true or high color images on video display cards that have the capability of displaying on 256 colors or 16 colors from their palette [13].

2.6.4 Histogram Processing

A Histogram is a discrete function representing count of pixels of each gray shade in a gray scale image. It provides global description of appearance of the image, mainly the overall intensity and contrast. The processing of histogram of an image relies upon some decisions being made about the whole image, based upon its statistics. Histogram equalization is a type of histogram processing technique used to achieve an optimal contrast improvement by redistribution of pixels in order to produce a uniform or equalized histogram. Histogram specification is a variation of this technique where something other than a uniform histogram is identified as goal. This may be used to compensate for the non-linear response of the human visual system. The figure given below shows the example of histogram of gray scale and color Lena image [13].

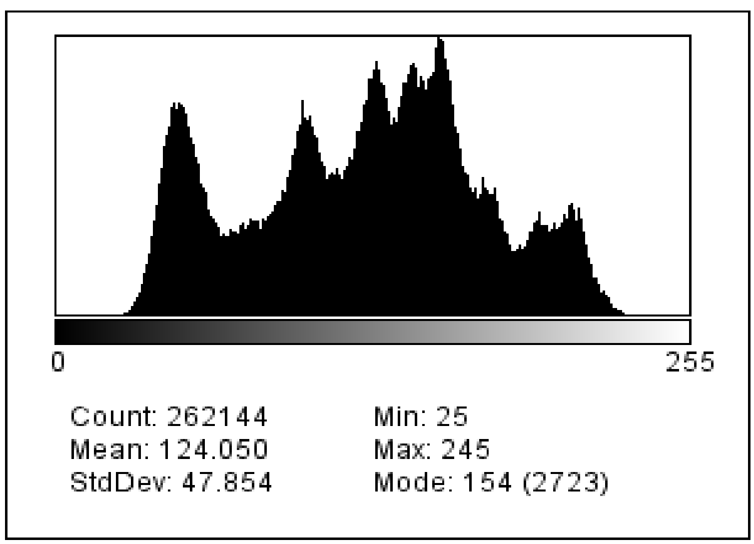


Figure – 2.16 Histogram of Lena gray scale image

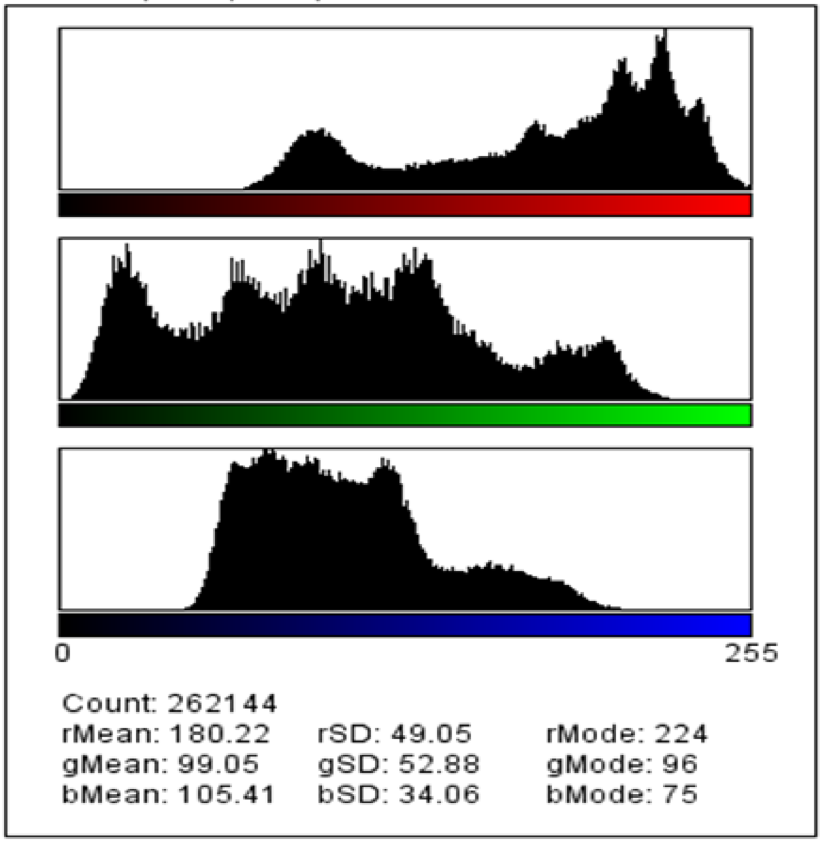


Figure – 2.17 Histogram of Lena RGB color image

Adaptive histogram equalization is yet another histogram processing technique which divides the image into sub-images and performs separate equalization operations on each; in some cases it yields better results, but it can be very slow [13].

2.6.5 Negation

Negation of an image is creation of negatives of the image by using a transformation function to reverse the order of gray level, so that intensity of the output image increases as the intensity of the input image decreases. The figure given below shows the example of negation of Lena image [13].

Figure – 2.18 Negative of gray scale Lena image

Negatives are useful in displaying medical images and photographing a screen with monochrome positive film with the idea of using the resulting negatives as normal slides [13].

2.6.6 Spatial Domain Filters

Spatial domain refers to the aggregate of pixels composing an image, and spatial domain methods are procedures that operate directly on these pixels. The main approach is defining a neighborhood about an image pixel by using a square or rectangular sub image area centered at that position. The center of sub image is moved from pixel to pixel from top left corner to the bottom right corner of the image. A spatial mask is applied for image enhancement at each location. Spatial mask is a matrix applied to an image pixel and its neighborhood to achieve desirable results. The following filters are available in spatial domain [13].

- Low Pass Filter
- High Pass Filter
- Band-Pass and Band-Reject Filter

Low Pass Filter: - It removes the high frequency components from the image giving a smoothening effect to the image. It finds application in removal of noise from the image and as an intermediate stage in image enhancement.

Figure – 2.19 Low pass filter of Lena image

High Pass Filter: - It removes all low frequency components from an image giving a sharpening effect to the image. This is useful in enhancing outlines in an image [13].

Figure – 2.20 High pass filter of Lena image

Band-Pass and Band-Reject Filter: - It can be used to pass on or reject frequencies in a certain range. One selectively performs the filter operations on a small region of the image.

Band-Pass and Band-Reject Filters can be used to pass on or reject frequencies in a certain range. One selectively performs the filter operations on a small region of the image [13].

Figure – 2.21 Band pass filter of Lena image

2.6.7 Frequency Domain Filters

Frequency domain techniques are based on convolution theorem which involves translation of image pixels into respective frequency components. Translation to the frequency domain is represented by the following formula:

$$g(x, y) = h(x, y) * f(x, y)$$

That is convolution in spatial domain. Here f(x, y) represents the original image, h(x, y) is the frequency or optical transfer function and g(x, y) is the final convoluted image [13].

$$g(x,y) = h(x,y) * f(x,y) \Longleftrightarrow G(u,v) = H(u,v)\, F(u,v)$$

Enhancement in frequency domain is done in three steps:

- Computing Fourier transform of the input image.
- Multiply the resulting output by filter transfer function.
- Apply inverse Fourier transform to obtain the transformed image in spatial domain.

The advantage of frequency domain filters is that in finding solution too many problems not easily addressable by spatial domain techniques since the features of the image are directly related to the frequency. Edge detection and highlighting can be easily done by emphasizing high frequency components in the frequency domain. Similarly low-pass and other filters can be used to bring about a desired cosmetic or smoothing effect [13].

2.6.8 Brightness Enhancement

Brightness is an overall attribute of the image. The need for enhancing (i.e., increasing or decreasing) the brightness content of the image can be easily confirmed by looking at its histogram or simply its overall visibility. If many pixels are concentrated towards one end of the histogram abscissa, brightness can be applied to pull them towards the other end. In the simplest form increasing brightness can be thought of as the addition of a constant to all pixel intensity values stored in the image array. Clearly brightness can be decreased by subtraction of a constant value. Sophisticated brightness enhancement algorithms use appropriate non-linear transfer functions depending upon the kind and degree of change desired. Brightness can be also affected on color imagery using appropriate color models [13].

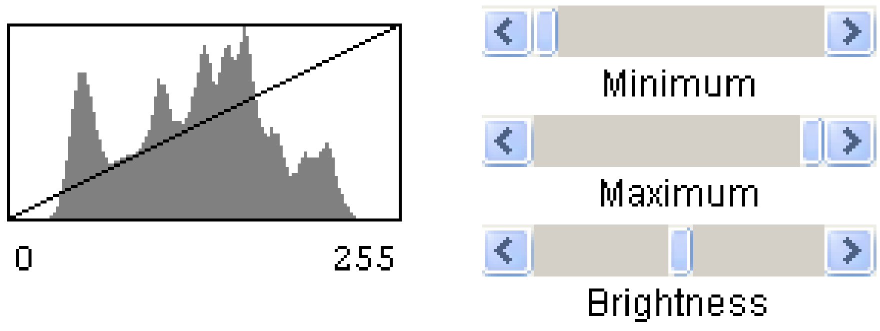


Figure – 2.22 Brightness Enhancement

2.6.9 Contrast Enhancement

If the image is such that it does not make full use of the available gray scale range it is said to lack contrast. Contrast enhancement is the process of stretching the used range of the gray scale values to a desired level, thereby altering the distribution of the pixel intensities in the histogram. There are many ways in which contrast enhancement can be achieved. One simple way to stretch the contrast is to simply multiply the pixel values with $2^n/P_{max}$, where P_{max}, is the maximum gray level value of significance in the original image. Contrast enhancement can be also effected on color imagery using appropriate color models [13].

A combination of contrast modification and brightness adjustment is used for image restoration to make up for inappropriate camera work, and also to achieve optimal usage of the available gray levels. However, it can equally well be used for image enhancement, for example of radiograms, where one part of the gray scale is emphasized at the expense of other parts, to reveal specific application-dependent information [13].

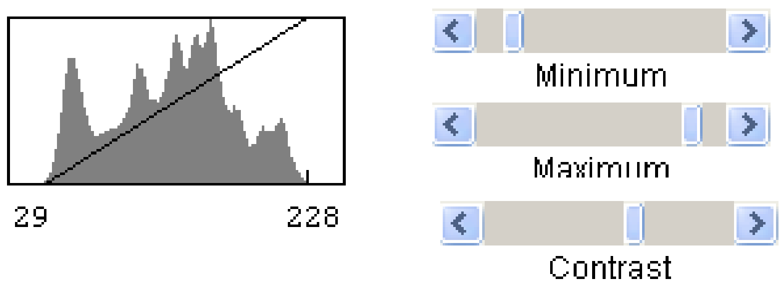


Figure – 2.23 Contrast Enhancement

2.6.10 Thresholding

Thresholding is a common technique of converting any higher pixel depth image into binary. This operation might be intended for the situation where special effect is to be shown, or when display/printing device require the image in the binary form, or when an image is required to occupy little memory space. At times thresholding is also reckoned as an image analysis technique since it divides the image into two parts - the foreground and the background. Several methods exist for thresholding of images. The name thresholding comes from the fact that all the methods first arrive at a value 'T', and then binaries the image such that all pixels with gray shades above this value are turn ON and below this value are turned OFF [13].

Depending upon the selection of threshold there are various thresholding techniques like global thresholding technique where the entire image is analyzed and T is chosen, local or adaptive thresholding technique where the image is divided

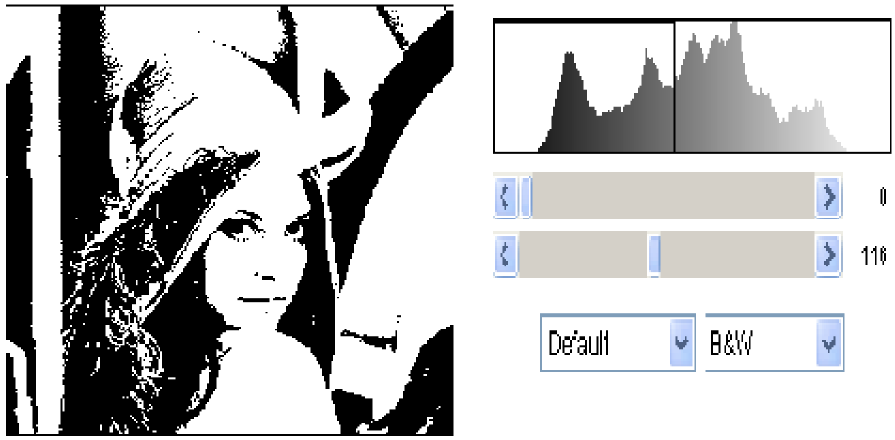
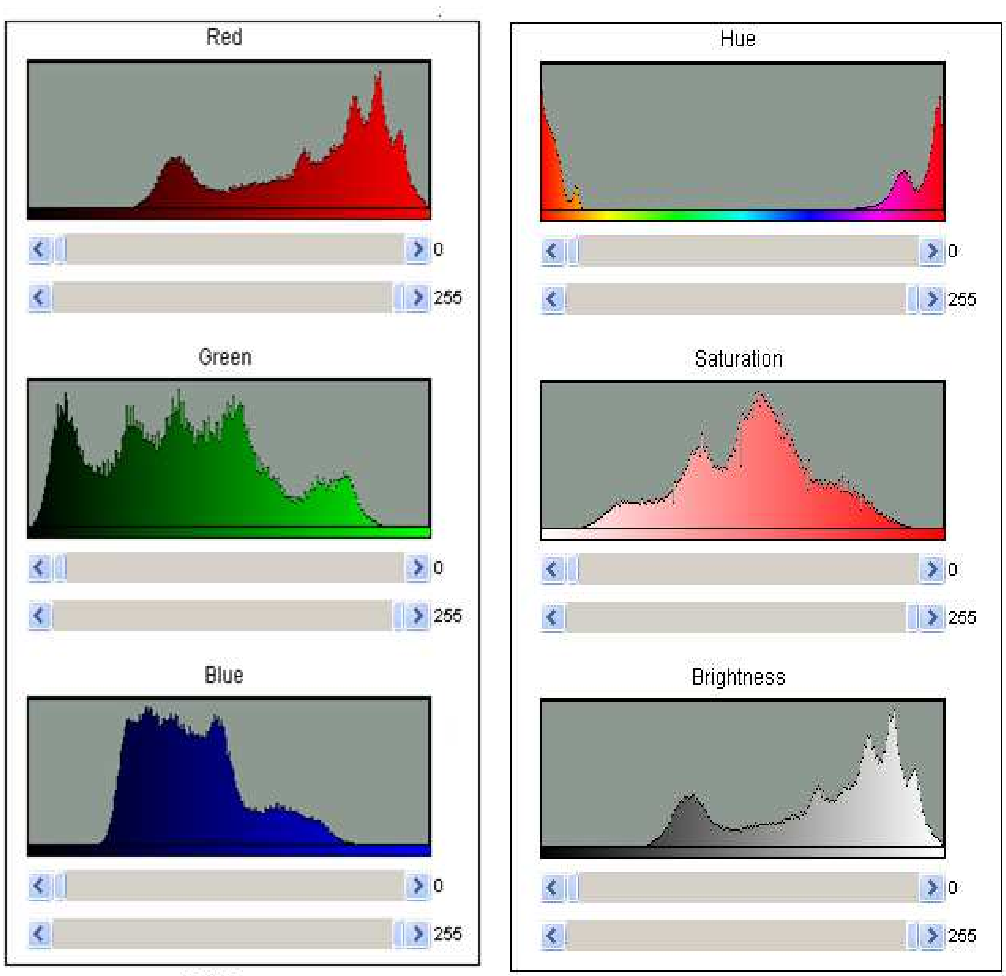


Figure – 2.24 Thresholding of gray scale Lena image



RGB HSB

Figure – 2.25 Color thresholding of Lena image

into many smaller regions and each region is analyzed separately and threshold by taking a local threshold value. Although originally meant for gray scale image, thresholding is extendable for operation on color images [13].

2.6.11 Gamma Correction

Gamma correction is required in almost all practical imaging applications to compensate for the non-linear responses of the devices that capture or display images, including the response of the human eye. For example CRTs do not produce equal change in the intensity of pixels with equal change in gray level of the pixels. Unless the overall effect of all non-linear devices is neutralized by the Gamma Correction operation, the images with higher pixel-depth would appear quite different from what they originally are [13].

Figure – 2.26 Gamma correction of gray scale Lena image

One should note that Gamma correction is recommended only for neutralizing the non-linear effect and not for merely effecting enhancement on the image [13].

2.6.12 Highlight and Shadow

Highlight and shadow are the processes for adjusting the brightness of certain pixels. These operations can also be used to clip some of the gray levels so that the rest is given more significance for easy analysis [13].

Different types of shadow are: North, East, South, West, Northeast, Southeast, Northwest, and Southwest.

The figure given below shows the North Shadow of gray scale Lena image.

Figure – 2.27 North shadow of gray scale Lena image

2.6.13 Edge Detection

Edge Detection is an important pixel-based method of segmentation where the image is divided into meaningful regions, which corresponds to part of, or the whole of, objects within a compound image. In edge detection edges that characterize object boundaries are identified, which is much useful in analyzing and identification of objects in the scene. An edge is the boundary between two regions with relatively distinct gray level properties. To find edges, edge detection operators called as masks are used which are convolved with the image to identify an edge. Masks are further divided into two types Gradient operators and compass operators depending upon their direction in which they detect an edge. Gradient operators are capable of detecting edges only in two orthogonal directions and some of the widely used operators of this type are Roberts, Prewitt, etc. Compass operators measure gradients in a selected number of directions and widely used compass operators are Kirsch, and Sobel [13].

Figure – 2.28 Edges of gray scale Lena image

2.7 GEOMETRIC TRANSFORMATION

In this category of the image processing operations, the spatial distribution of pixels is deliberately changed to achieve the desired results. Geometric transformations might affect the dimension, orientation or position of a part or the whole image [13].

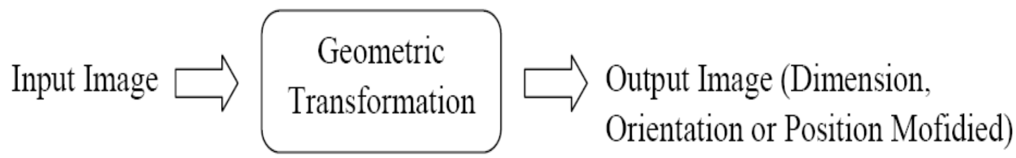


Figure – 2.29 Geometric Transformation[13]

The following types of transformations are geometric:

- Scale / Resize / Zoom
- Rotation
- Stitching
- Copy / Cut / Paste
- Curve / Surface Fitting

2.7.1 Scale / Resize / Zoom

There are three most primary geometric operations that are performed in almost all the documents of image.

- Scale
- Resize
- Zoom

The above operations are allowed to change the X scale, Y scale, Height, Width, Interpolation, Zoom In, Zoom Out, Set Zoom, etc…

In most cases the scanning resolution of the documents is much higher than the resolution of the video display devices. So if the document is visualized as on the video monitor, it does not accommodate the whole image; for instance on the traditional 640 X 480 screen resolution VGA modes only a fraction of A4 size document will be visible, if it is scanned at 300 dpi (dots per inch). In order to get the overall view of the image document, it is first scaled and then sent to the display device. Sometimes a small portion of the image is required to be enlarged in order to inspect or edit it with greater comfort. Programs like scanner and printer device drivers make effective use of these routines to meet the rigid requirements of the hardware devices [13].

Naturally information loss occurs when an image is condensed, and additional information need be created when it is enlarged

Examples of Scale, Resize and Zooming function on gray scale Lena image are shown as follows:

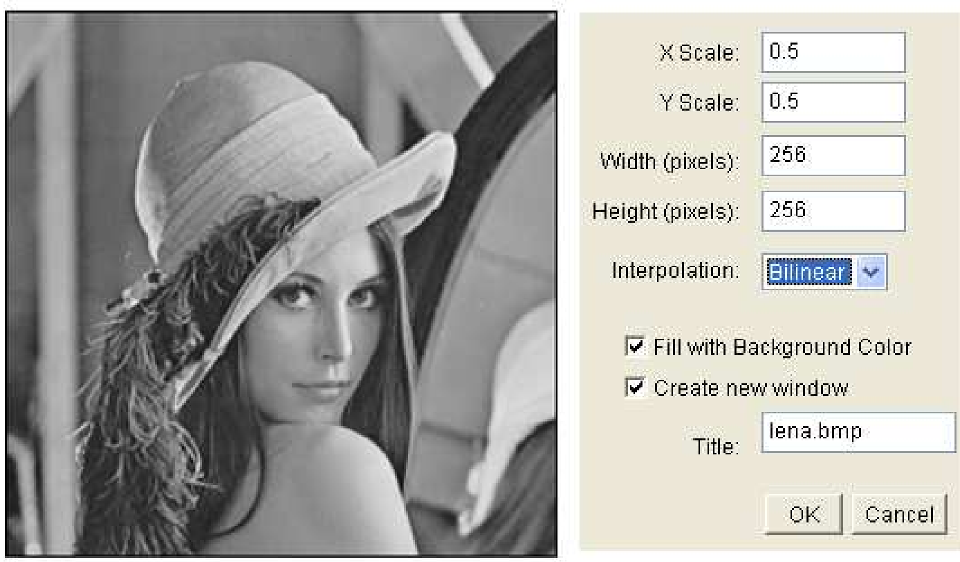


Figure – 2.30 Scale function of Lena image

Figure – 2.31 Resize function of Lena image

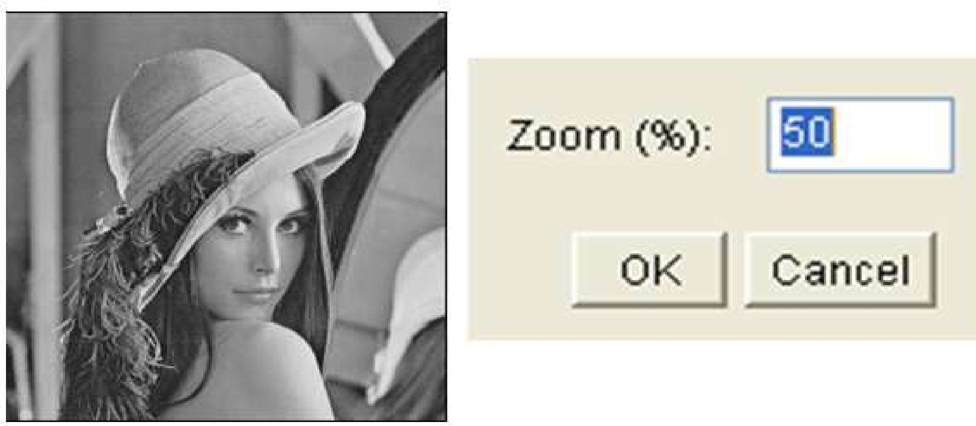


Figure – 2.32 Zooming function of Lena image

2.7.2 Rotation

Rotation is used to changing the orientation of the image about its center or a predefined center of rotation. Three rotation methods are available. Rotation of an image is a simple matrix transposition if the angle of rotation is 90, 180 or 270 degrees. However, customized rotation for any angle other than these may be a computationally expensive exercise.

The quality of rotated gray scale and color images can be enhanced by using interpolation option to the rotate functions. Rotation in general is a very important task as it is internally used in many operations. The figure given below shows the rotation 50 degree of gray scale Lena image with 5 grid lines [13].

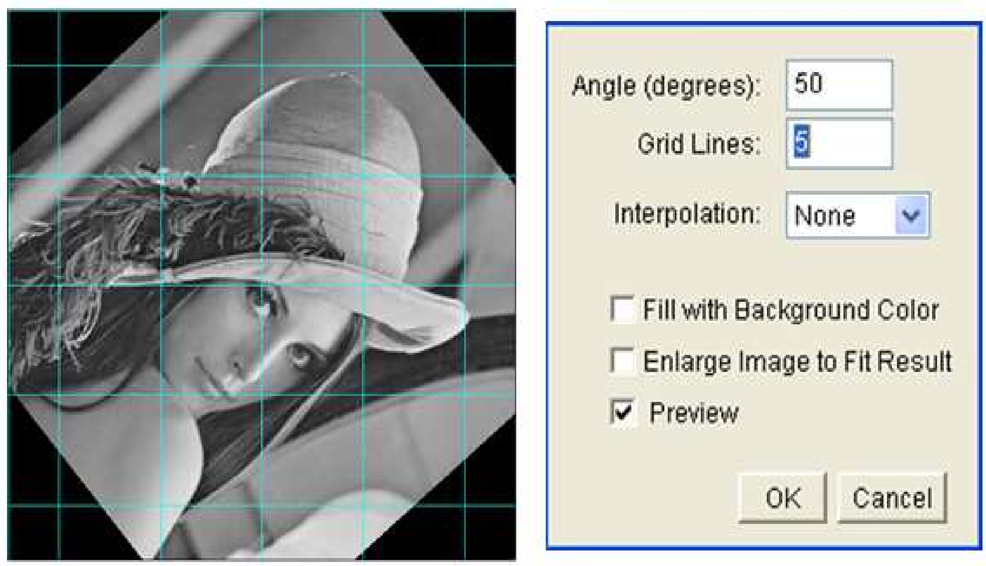


Figure – 2.33 Rotation of gray scale Lena image

2.7.3 Stitching

Stitching is the process of appending an image to the top, left, bottom or right edges of another image, resulting in a large space-wise contiguous image. This is useful for big documents images that cannot be scanned completely in one go due to limitation of scanner. In such cases document is scanned in two or more parts and the parts are stitched together to obtain the original continuous image [13].

2.7.4 Copy / Cut / Paste

Following are three basic and simple geometric operations to be performed in image.

- Copy
- Cut
- Paste

It can copy or cut the selected specific rectangular portion of the image. It can also paste the copied or cut portion of the image to the specific place of the same image document or any other image document or any other documents. These operations are useful for various image editing operations that are performed in various types of image and image application software [13].

2.7.5 Curve / Surface Fitting

It implies generation of a curve/surface for fitting into a specific imaging situation. For example free-hand curve drawing may be required to allow the user define some non-rectangular graphic for annotations, etc. There are many standard methods available for fitting curve and surfaces. Straight Line, 2nd, 3rd and 4th degree Polynomial, Exponential, Power, Log, Gamma Variant, Gaussian etc are examples of various types of curves. The figure given below shows the Gaussian curve fitting of gray scale Lena image [13].

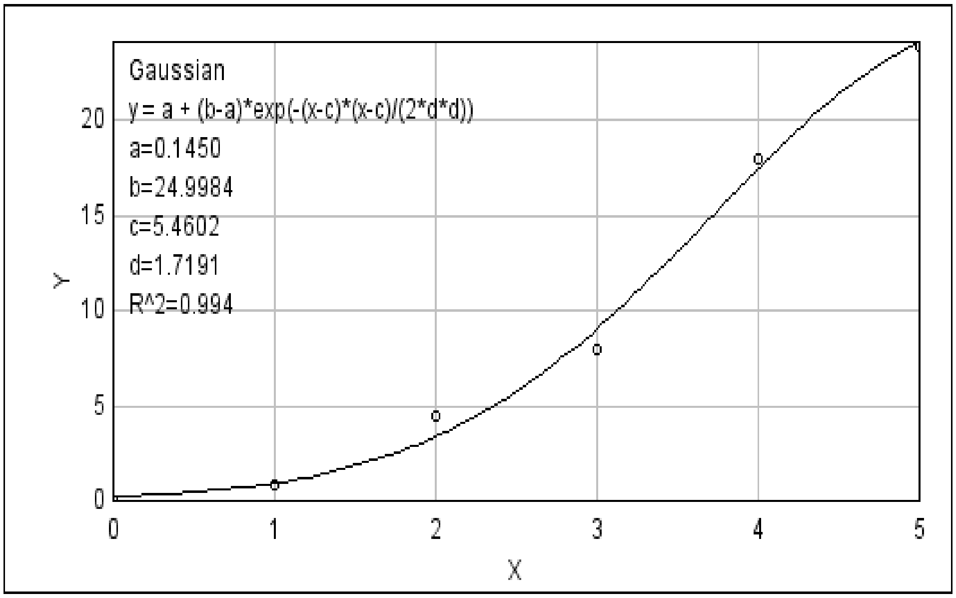


Figure – 2.34 Gaussing curve fitting of gray scale Lena image

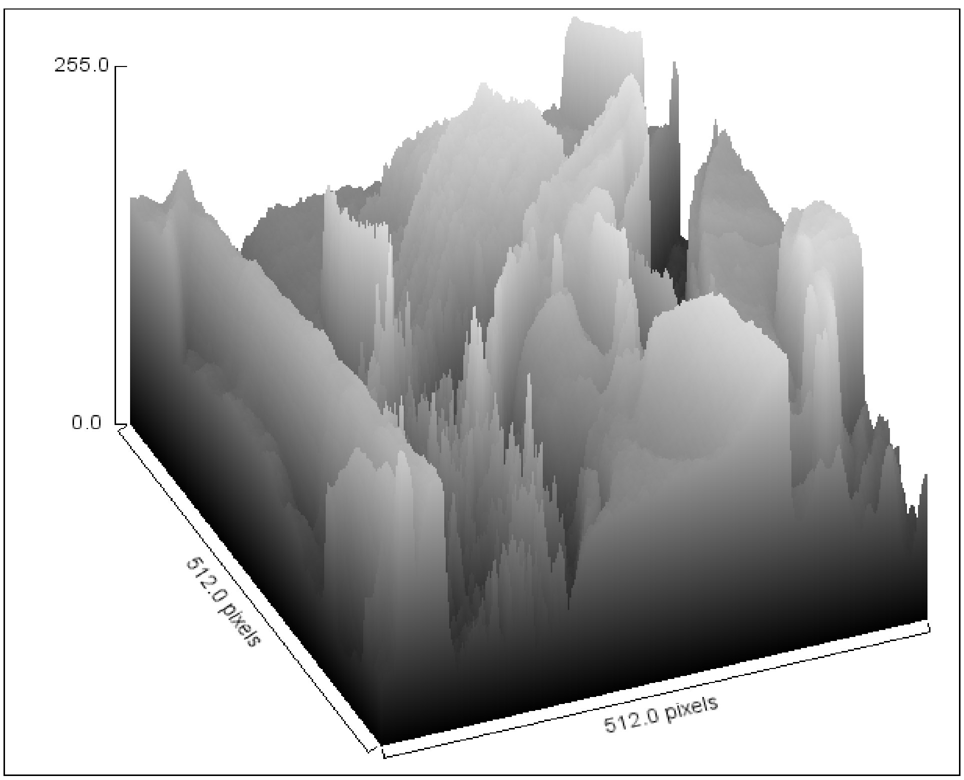


Figure – 2.35 Surface fitting of gray scale Lena image

2.8 IMAGE ANALYSIS

Image Analysis deals with interpretation and recognization of image contents. It is the most difficult part of the image processing technology and often the most time consuming one. The long term goal of image processing technology is to develop an ultimate system that will sense its surrounding through its visual faculties i.e. sensors and will function autonomously as the human beings do. However in the present state of arts, the world uses image analysis to decipher the graphical text or bar symbologies, to analyze and quantify the skew present in the document images, to perform layout analysis of the document, to segment the image, and so on[13].

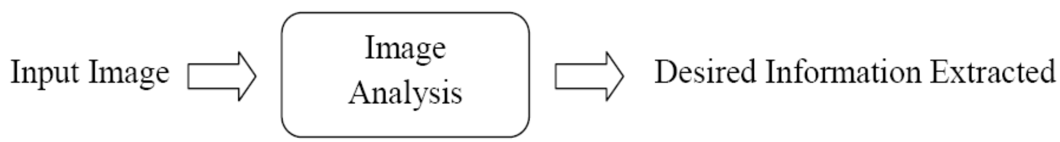


Figure – 2.36 Image analysis[13]

2.8.1 Skew Detection and Correction

Optical Character Recognization (OCR), Optical Mark Recognization (OMR), Barcode Recognization or simple visualization require the input image that is free from effects such as skew that gets introduced at the time of scanning. For further processing on images, it must be corrected the skew. If the skew is not corrected, further processing will be fail. Normally a skew in the range -10 to +10 degree creeps in on the scanners with or without ADF. Many people find the document visually displeasing if the skew happens to be greater than ±1 degree [13].

2.8.2 Optical Character Recognization (OCR)

It is a technology to the document management system. It is a means by which the text objects within a document are converted from a bit-mapped image to a text representation such as ASCII (American Standard Code for Information Interchange). Those groups of pixels which form alphanumeric characters are “read” by the OCR software, and translated into proper machine-readable character code.

This technology could support only a limited number of popular typewriter fonts. The introduction of systems that could be taught different fonts as necessary broadened the appeal of OCR to the desktop publishing and text retrieval markets. Pattern Recognization and Feature Extraction has replaced the OCR technique. Template matching has become the most commonly used technology. It allows recognization of virtually all characters based on their unique features. These new technologies comprise of a new set of recognization add-ons, commonly referred to as Intelligent Character Recognization [13].

2.8.3 Intelligent Character Recognization (ICR)

One of the most easily confused terms in OCR is Intelligent Character Recognization (ICR). It is used in different ways by different software. ICR is commonly accepted as the general term for OCR using lexical analysis. ICR has been extended to include handwriting recognization, and is often called Handwriting ICR [13].

2.8.4 Optical Mark Recognization (OMR)

In OMR Technology, filled forms are scanned using optical mark readers, scanners or any other image acquisition devices. Scanned forms are processed for mark recognization and the result is returned in the form of an ASCII file. This file can be used by various Word processors, Spread sheets and Databases [13].

2.8.5 Layout Analysis

It is mostly required to find or extract out some portion of the document. For example, it is required to locate and extract the bar code symbols present in the document, or it may be the text region for that matter. To perform this operation, it first divides the document into components, and then analyzes each of it to declare the regions that contain the bar symbols or text.

Layout analysis is a technique that is applied to formatted, machine printed pages. Form recognization, a type of layout analysis is applied to machine printed or handwritten text occurring within the delineated blocks on a printed form. It comprises of methods for analyzing the contents of the document [13].

Layout analysis is performed in the following two different ways:

- Structural Layout Analysis
- Functional Layout Analysis

Structural Layout Analysis: - It is to get physical segmentation of groups of document components.

Depending on the document format segmentation can be performed to isolate words, text lines and structural blocks such as paragraphs, tables etc… [13].

Functional Layout Analysis: - It is also known as logical layout analysis. It uses domain dependent information consisting of layout rules of a particular page to perform labeling of the structural blocks giving some indication of the function of the block. For example functional labeling of the first page of an article would indicate title, an author, abstract, keywords, paragraphs of the text body etc… [13].

2.9 IMAGE REGISTRATION

The images need to be geometrically aligned for better observation. This procedure of mapping points from one image to corresponding points in another image is called Image Registration. Advances in computer science have led to reliable and efficient image processing methods useful in medical diagnosis, treatment planning and medical research. In clinical diagnosis using medical images, integration of useful data obtained from separate images is often desired.

It is a spatial transform. The reference and the referred image could be different because were taken

- At different times
- Using different devices like MRI, CT, PET, SPECT etc.
- From different angles in order to have 2D or 3D perspective

Registration is mostly used in medical imaging applications. A common practice of these applications could be found in fusion of multimodality images when patients have to undergo epilepsy surgery. Registration and fusion of MR and PET images will benefit the surgeon. Besides multimodality registration, there exist important application areas in monomodality registration.

2.9.1 Introduction to Image Registration

The registration problem is to find the optimal transformation T* which best aligns the images. For reference image I and floating image J, image registration can be defined as follows [14]:

$$T^* = arg\max_T \rho\left(I, T\left(J\right)\right)$$

Where ρ refers to similarity measure.

Therefore, different registration methods can be derived from different similarity measures and different search strategies. For example in MI-based methods, ρ is mutual information of the images.

The most common transformations applied to register medical images are rigid and affine. An affine transformation includes transition, rotation, scaling and shearing where it maps parallel lines to parallel lines. Rigid is a special kind of affine transformation where only transitions and rotations are allowed. In rigid transformation, the objects retain their relative shape and size. It is generally used for brain images. A three-dimensional rigid transformation is the product of rotation R and transition D matrices where the rotation matrix is the product of three matrices representing rotation around x, y and z axes [14].

$$
\begin{aligned}
T &= D \times R \\
R &= R_x \times R_y \times R_z
\end{aligned}
$$

Transition matrix includes three parameters {dx, dy, dz} as:

$$
D = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

And rotation matrices representing three angles {φx, φy, φz} are expressed as:

$$
R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\varphi_x) & -sin(\varphi_x) & 0 \\ 0 & sin(\varphi_x) & cos(\varphi_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
R_y = \begin{bmatrix} 0 & cos(\varphi_y) & -sin(\varphi_y) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & sin(\varphi_y) & cos(\varphi_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
R_z = \begin{bmatrix} 0 & cos(\varphi_z) & -sin(\varphi_z) & 0 \\ 0 & sin(\varphi_z) & cos(\varphi_z) & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Consequently, a rigid transformation requires six independent parameters to be determined [14].

2.9.2 Steps of Image Registration Process

Due to the diversity of images to be registered and due to various type of degradations it is impossible to design a universal method applicable to all registration tasks. Every method should take into account not only the assumed type of geometric deformation between the images but also radiometric deformations and noise corruption, required registration accuracy and application dependant data characteristics [15].

Mostly all image registration processes follow the below four different steps:

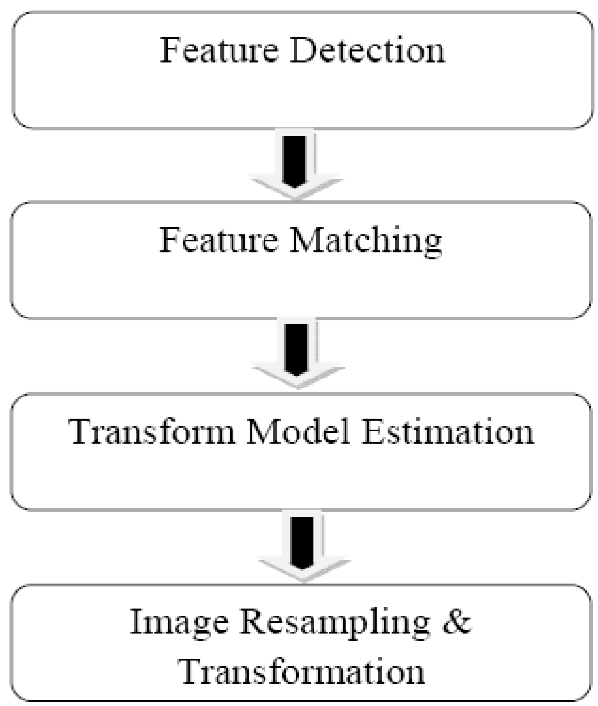


Figure – 2.37 Steps for image registration process [15]

Feature detection: - This step is based on the extraction of salient features and structures in the image. Significant regions, lines or points are considered.

Feature matching: - In this step, the correspondence between the features detected in the sensed image and those detected in the reference image is established.

There are two major categories, Area based and Feature based methods, are further classified into subcategories according to different methods and advantage and disadvantage.

Transform model estimation: - The type and parameters of the so-called mapping functions, aligning the sensed image with the reference image, are estimated.

After the feature correspondence has been established the mapping function is constructed. It should transform the sensed image to overlay it over the reference one.

Image re-sampling and transformation: - The sensed image is transformed by means of the mapping functions. Image values in non-integer coordinates are computed by the appropriate interpolation technique.

The mapping functions constructed during the previous step are used to transform the sensed image and thus to register the images. The transformation can be realized in a forward or backward manner. Each pixel from the sensed image can be directly transformed using the estimated mapping functions. This approach, called a forward method, is complicated to implement, as it can produce holes and/or overlaps in the output image (due to the discretization and rounding). Hence, the backward approach is usually chosen.

2.9.3 Application of Image Registration

Registration is required in many fields. Following are the various applications in various fields:

- Remote sensing
- Multi-spectral classification
- Environmental monitoring
- Weather forecasting
- Creating super resolution images
- Geographical information system (GIS)
- Medical image analysis
- Labeling and segmentation.
- Target localization
- Automatic quality control
- Shape reconstruction
- Motion tracking
- Stereo mapping
- Character recognization

2.9.4 Classification of Image Registration

The classification of image registration process is based on nine basic criteria [15].

The classification of nine basic criteria is as follows [15]:

1. Dimensionality
   - 2 D
   - 3 D
2. Nature of registration basis
   - Extrinsic
   - Intrinsic
   - Non image based

3. Nature of transformation
    - Rigid
    - Affine
    - Projective
    - Curved
4. Domain of transformation
    - Local
    - Global
5. Interaction
    - Interactive
    - Automatic
    - Semiautomatic
6. Optimization procedure
    - Parameters computed directly
    - Parameters searched for
7. Modalities involved
    - Mono model
    - Multimodal
    - Modality to model
    - Patient to modality
8. Subject
    - Inter subject
    - Intra subject
    - Atlas
9. Object
    - Head
    - Thorax
    - Abdomen
    - Pelvis
    - Limbs

The above classification is based on the following features [15]:

Dimensionality: - 2D/2D, 2D/3D, 3D/3D Image registrations are possible. Sometimes time could be the fourth dimension.

Domain of transformation: - It could be global or local depending on whether the whole image or its part is to be registered.

Type of transformation: - The transformation could be rigid, affine, projective or nonlinear.

Tightness of feature coupling: - The transformation can be interpolating (features of the objects in one image are exactly transferred into features in the other image) or approximating.

Measure of registration quality: - Various measures are applied depending on the data features or data itself.

Method of parameter determination: - The parameters of the transformation can be found out using direct or search oriented methods.

Subject of registration: - If the two images contain the same subject it is intra subject registration. If the subject in the two images differs it is inter subject registration.

Type of data: - It can be raw data; features extracted from data or introduced markers in data.

Source of features: - Features explicitly present in the data are called intrinsic features where as those introduced from outside are called as extrinsic features.

Automization level: - This can be automatic or semiautomatic depending on user intervention level.

2.9.5 Image Registration Approach

Image Registration approach is given by different scientists as per the following:

- Transformations using Fourier Analysis
- Cross correlation approach using Fourier Analysis
- Sum of squares search technique
- Eigen value decomposition
- Moment matching techniques
- Warping techniques
- Procedural approach
- Anatomic atlas
- Internal landmarks
- External landmarks

2.9.6 Image Registration Methods

Following are different methods of image registration process:

- Extrinsic registration methods
- Curve methods
- Surface methods
- Moment and Principal Axes methods
- Correlation methods
- Atlas methods
- Mutual information-based methods
- Wavelet-based methods
- Soft Computing based methods
    - Artificial Neural Networks
    - Genetic Algorithm
    - Fuzzy sets
    - Rough Sets

2.10 IMAGE SEGMENTATION

Image segmentation is an important process and its results are used in many image processing applications. However, there is no general way to successfully segment all images. Color images have more information than gray scale images, and this information can be used to create higher quality segmentation. It does, however, increase the complexity of the problem. A way of handling this complexity is to use a directed search method, such as genetic algorithms. Genetic algorithms, which have many qualities that make them well suited to the problem of image segmentation, such as the ability to forego a local optimum to reach a global optimum and the ability too efficiently find an optimal solution from within a large search space.

The main uses of genetic algorithms in image segmentation are for the modification of parameters in existing segmentation algorithms and pixel-level segmentation. Various algorithms that successfully apply genetic algorithms to image segmentation have been developed. Though these results are promising, none of them has yet been able to solve this open problem.

2.10.1 Introduction to Image Segmentation

Image segmentation is the process of dividing an image into homogeneous regions. This is equivalent to finding the boundaries between the regions. It is the primary or first step for higher level image processing. It includes medical imaging, locating object through satellite images, shape recognization, road sign recognization, face detection, etc…

Positions of image segmentation are as follow:

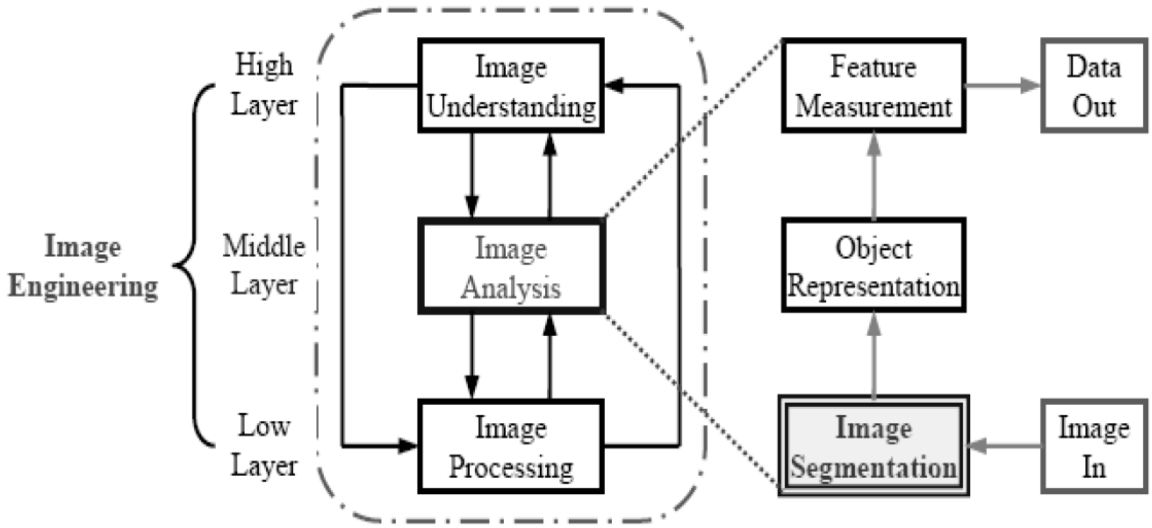


Figure – 2.38 Position of image segmentation

2.10.1.1 Color Segmentation

Until recently, most image segmentation has been performed on gray scale images. Processing color images requires much more computation than the processing of gray scale ones, but now with the increasing speed and decreasing cost of computation; color image processing has been much researched in the last decade.

Color images contain far more information than monochrome images. Each pixel in a color image has information about brightness, hue and saturation. There are many models to represent the colors, including RGB (red, green, blue), CMY (cyan, magenta, yellow), HSV (hue, saturation, and intensity), YIQ, HSI and many others. Several color spaces have been used for image segmentation and no general advantage of one color space has yet been found.

Many of the color image segmentation algorithms are derived from methods of gray scale image segmentation. However, color creates a more complete representation of an image and exploiting this fact can result in a more reliable segmentation. Specialized techniques suited to the nature of color information have been devised.

2.10.1.2 Methods of Image Segmentation

Image segmentation is an old and important problem, and there are numerous image segmentation methods. Most of these methods were developed to be used on a certain class of images and therefore aren't general image segmentation methods.

The image segmentation algorithms are divided into following three major categories:

- Edge Based
- Region Based
- Clustering Based

Edge Based Techniques

Edge detection involves the detection of boundaries between different regions of the image. These boundaries correspond to discontinuities between pixels of the chosen feature (e.g. color, texture, intensity).

Region Based Techniques

Region based techniques are further divided into following four categories:

1. Region Splitting
2. Region Growing

3. Region Merging
4. Region Splitting and Merging

Region Splitting: - It is an image segmentation method whereby pixels are classified into regions. Each region corresponds to a range of feature values, with thresholds being the delimiters. The choice of these thresholds is very important, as it greatly affects the quality of the segmentation. This method tends to excessively split regions, resulting in over segmentation.

Region Growing: - It joins neighboring pixels with similar characteristics to form larger regions. This continues until the termination conditions are met. Most of the region growing algorithms focus on local information, making it difficult to get good global results. This method tends to excessively add to regions, resulting in under segmentation.

Region Merging: - It recursively merges similar regions. It is similar to region growing, except that two whole regions are combined, rather than one region combining with individual pixels.

Region Splitting and Merging: - It tries to overcome the weaknesses of region growing and region splitting by combining the two techniques. Initially the image is divided into arbitrary regions. Region splitting and region merging occur until the termination conditions are met.

Clustering Based Techniques

Clustering separates the image into various classes without any prior knowledge. This method is based on the assumption that objects within each class should have a high degree of similarity, while those in different classes should be dissimilar. It is considered an unsupervised image segmentation technique.

2.10.1.3 Difficulties with Image Segmentation

Image segmentation is easy when objects have distinct colors and are well separated, but can be a problem if there are many complex objects with less distinct color. Gradual variation in color, illumination, shading and textures are also possible problems.

A brute force method of dealing with image segmentation would be to enumerate all possible partitions of the image and evaluate each one. This creates an extremely large search space, and so this method is not feasible.

Even once a segmentation method has been chosen; there are usually many parameters that need to be tuned to create high quality segmentation. For most methods, it is not feasible to perform an exhaustive search of these parameters.

Despite the many methods for image segmentation, there is no general algorithm that works well for all images. Because of the wide variety of images, a general algorithm needs to be adaptable. Only then can a segmentation algorithm cope with a wide variety of images.

Many adaptive methods have been used for image segmentation, including genetic algorithms, neural networks, self-adaptive regularization, ant colony optimization, fuzzy clustering and simulated annealing.

2.10.2 Segmentation Evaluation

Methods of evaluating image segmentation are divided into two main categories:

- Analytical Method
- Empirical Method
  - Goodness Method
  - Discrepancy Method

Analytical Method: - It looks at the actual segmentation algorithm itself, rather than its results, while empirical methods evaluate the segmentation algorithm by looking at its results. In this case, it is the empirical methods that we are more interested in.

Empirical Method: - It can be further sub divided into goodness method and discrepancy method.

- **Goodness Method:** It evaluates the quality of the segmentation by looking at its desirable properties, and does not compare it to any other segmentation. It is this type of method that we wish to use as a fitness function for the genetic algorithm, because they don't need any prior knowledge of the image segmentation or ground truth segmented image and so are advantageous for an unsupervised algorithm.
- **Discrepancy Method:** It compares the segmentation results to an ideally segmented image, to see how much its segmentation differs from the target segmentation. It is this class of method that we will use to evaluate the results of our segmentation algorithm during experimentation.

CHAPTER – 3

LOSSLESS AND LOSSY IMAGE COMPRESSION TECHNIQUES

3.1 INTRODUCTION

An image is generally a 2-D signal processed by the human visual system. The signals representing images are usually in analog form. However, for processing, storage and transmission by computer applications, they are converted from analog to digital form. A digital image is basically a 2-Dimensional array of pixels. Images from the significant part of data, particularly in remote sensing, biomedical and video conferencing applications. The use of and dependence on information and computers continue to grow, so too does our need for efficient ways of storing and transmitting large amounts of data. Below figure shows the digital image.

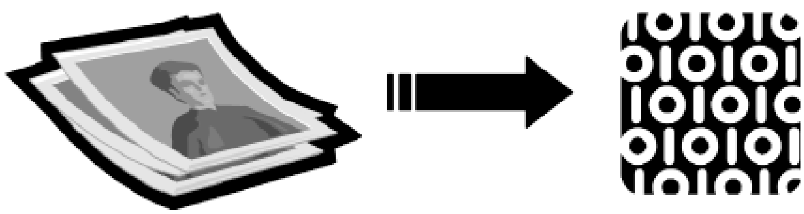


Figure – 3.1 Digital image representations

Digital image and video compression is now essential. The Internet has resulted in a network by which almost any form of communication is possible. All internet users are communicated through text, images or video. The simplest and easiest way is through text and it is quick as simple to use as sending text. The problem is to send and receive images and videos require large amount of bandwidth. Compression is an important component of the solutions available for creating file sizes of manageable and transmittable dimensions. Increasing the bandwidth is another method, but the cost sometimes makes this a less attractive solution. Therefore there is a need to decrease the size of the image or video sent or received.

The aim of data compression techniques is to reduce the amount of data needed to accurately represent an image, such that this image can be economically transmitted or archived. The easiest way to reduce the size of the image file is to reduce the size of the image itself. By shrinking the size of the image, fewer pixels need to be stored and consequently the file will take less time to load. The problem with this is that if an image file is reduced the quality of the image is also reduced.

The overall goal of compression is to represent an image with the smallest possible no. of bits, so speeding transmission and minimizing storage requirements. There are different techniques is used for compress image. They are broadly classified into two classes called: Lossless and Lossy compression techniques. The name suggests that in lossless compression techniques, no information regarding the image is lost. So, the reconstructed image from the compressed image is identical to the original image. Whereas in lossy compression techniques, some information related to image may be lost, i.e. the reconstructed image from the compressed image is similar to the original image but not identical to it.

3.2 IMAGE COMPRESSION – AN OVERVIEW

Nowadays, the use of digital media is rapidly increases. Every printed media is converting into digital form. It is necessary to compress images or videos due to the growing amount of visual data to make efficient transfer and storage of data. Visual data is stored in form of bits which represents pixels. The gray scale image requires 8-bits to store one pixel and a color image requires 24-bits to store one pixel [16] [17].

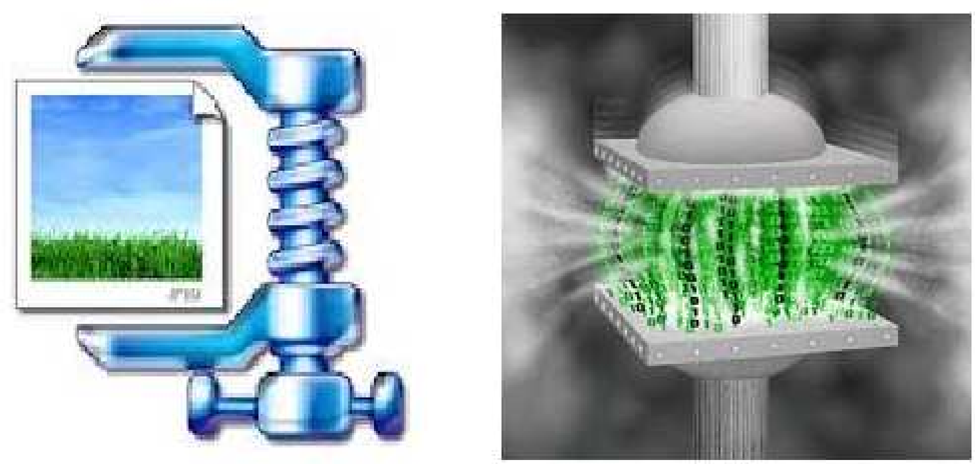


Figure – 3.2 Image compression

Access rate and bandwidth are also very important factors; uncompressed images take high amounts of bandwidth. For example, 512x512 pixels frames require 188.7 Mbit/sec bandwidth for colored video signals composed of 30 frames/sec without compression. Due to the dramatic increase of bandwidth requirement for transmission of uncompressed visual data i.e. images and videos, there is a need to

- Reduce the memory required for storage
- Improve the data access rate from storage devices
- Reduce the bandwidth and/or the time required for transfer loss communication channels

An image often contains redundant and/or irrelevant data. Redundancy is the statistical properties of an image and irrelevancy is the subject/viewer perception of an image. Both redundancy and irrelevancy are divided into following three different types:

- Spatial
- Spectral
- Temporal

Spatial Redundancy: - Correlation between neighboring pixel values causes spatial redundancy.

Spectral Redundancy: - Correlation between different colors curves causes spectral redundancy.

Temporal Redundancy: - It is related to video data and it has relationship between frames in a sequence of images.

These redundancies are reduced through different techniques during the compression. The main purpose of compression is to reduce the number of bits as much as possible. It also maintains the visual quality of the reconstructed image close to the original image. The decompression is the process of restoring the original image from the compressed image.

Typically there are three basic steps for image compression which are implemented in following order [16][17]:

(1) Image Decomposition or Transformation

(2) Quantization

(3) Entropy Encoding

Specific transformations are performed in the de-correlation step. The de-correlation step involves, segmentation into 4x4 or 8x8 blocks and applying curve fitting routine on each block and finally storing the parameters. The purpose is to produce uncorrelated coefficients that are quantized in the quantization step to reduce the allocated bits. Quantization is a lossy process. It cannot recover the lost data during de-quantization. At the end an entropy coding scheme is applied to further reduce the number of allocated bits. It is a lossless compression technique.

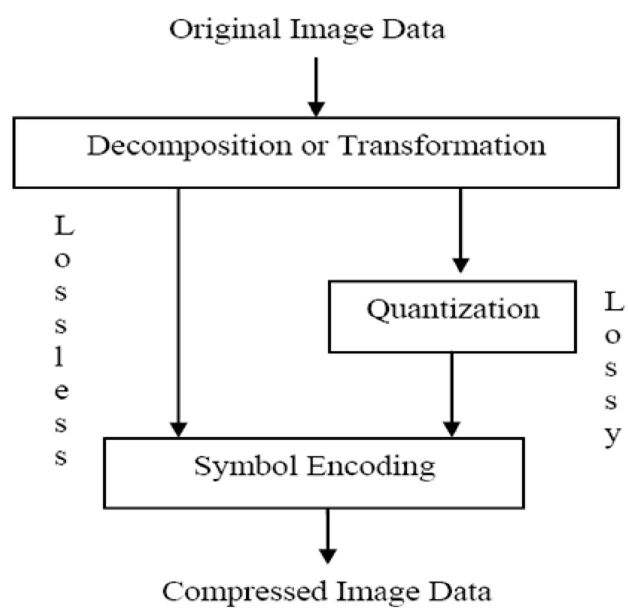


Figure – 3.3 General compression frameworks [17]

3.3 BACKGROUND STUDY OF IMAGE COMPRESSION

Image compression is divided into four subgroups:

1. Pixel Based
2. Block Based
3. Sub band Coding
4. Region Based

3.3.1 Pixel Based: - In this technique, every pixel is coded using its spatial neighborhood pixels. It is ensured that the estimated 'P_e' pixel value is very similar to the 'P_o' original pixel value [18].

3.3.2 Block Based: - In this technique is very efficient in some cases. Block based techniques are main two types [19] [20]:

- **Training: -** It includes Vector Quantization (VQ) and Neural Network (NN).
- **Non-Training:-** It includes Block Truncation, Transform Coding. Transform Coding also include Discrete Cosine Transform (DCT), Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), and Wavelet Transform etc...

3.3.3 Sub band Coding: - This technique is used for encoding audio and video data. In this technique each signal is divided into many frequency bands and then each band is treated separately in this process. Incomplete polynomial transforms quantize the coefficients to fit 4x4 blocks through sub band coding [20] [21].

3.3.4 Region Based: - This technique is computational intensive and slow. While block based compression technique is fast but it achieves limited compression.

The pictures can contain more than one image, text and graphics patterns. Due to the compound nature of an image there are three basic segmentation schemes [22]:

- Object Based
- Layered Based
- Block Based

3.3.4.1 Object based: - This segmentation scheme of an image is divided into different regions. This all region contains graphical objects such as photographs, graphical objects and text etc. By using this segmentation scheme it is difficult to get good compression because edges of the objects take extra bits in memory. So it gives less compression and increase the complexity [23].

3.3.4.2 Layered based: - This is simple technique as compare to object based. In this technique, an image is divided into layers such as graphical objects (pictures, buildings etc), text and colors. Every layer is treated with different method for compression [23].

3.3.4.3 Block based: - This is more efficient technique as compare to object and layer based. The main advantage of this technique is easy segmentation. Every block is treated with same compression technique and less redundancy [23].

In this image compression scheme, every image is divided into N x N (rows and columns) blocks and then quantization and entropy coding is applied for compression. It is supposed that data are in a matrix form, in two dimensions consisting of rows and columns. An image is divided into non-overlapping N x N (4 x 4 and 8 x 8) blocks. In 1st order curve, we find the parameters 'a_0', 'a_1' and 'a_2' for the equation $f(x, y) = a_0 + a_1x + a_2y$ such that the Mean Square Error (MSE) is minimized for the blocks. Here we can take sixteen values for 4x4 blocks and sixty four values for 8x8 blocks. Two dimensional representations of 8x8 blocks are shown in below figure:

X-axis

Y-axis

	a	b	c	d	e	f	g	h
i	1:1	1:2	..					
j	2:1	..						
k	..							
l								
m								
n								
o								
p								8:8

Figure – 3.4 2D representations of 8 x 8 blocks

Polynomial Curve fitting is used in different computer programs for different purpose [24]. The 2nd order Polynomial equations is $f(x, y) = a_0 + a_1x + a_2y + a_4x^2 +$

a_5y^2. The 1st order curve needs less computational time as compare with higher order curve. Large size blocks increase the value of the Mean Square Error (MSE) as compare with smaller size blocks [25].

$$\mathbf{MSE} = \frac{\textbf{Total Squared Errors}}{\textbf{Number of data points}}$$

Here 'Total Squared Errors' is equal to the square of the difference of the pixels between original and reconstructed image [25].

Main objective of the research is to get maximum compression with minimum loss of image quality. The error is minimal for higher order curves using small block size. To analyze the compression performance of the algorithm the Compression Ratio (CR) is calculated as [25]:

$$\mathbf{CR} = \frac{\textbf{No. of bits in original image}}{\textbf{No. of bits in compressed image}}$$

3.3.5 Error Metrics

There are following two different types of the error metrics is used to compare the various image compression techniques.

- Mean Square Error (MSE)
- Peak Signal to Noise Ratio (PSNR)

MSE is the cumulative squared error between the compressed and the original image.

PSNR is a measure of the peak error. The mathematical formulae for the two are:

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

PSNR = 20 * log10 (255 / sqrt(MSE))

Where I(x, y) is the original image, I'(x, y) is the approximated version (which is actually the decompressed image) and M, N are the dimensions of the images.

A lower value for MSE means less error. There are inverse relationship between the MSE and PSNR. So it translates to a high value of PSNR. Generally, a higher value of PSNR is good because it means that the ratio of Signal to Noise is higher. In this study the 'signal' is nothing but the original image. The 'noise' is the error in reconstruction. So, if you find a compression scheme with lower MSE and a high PSNR, that’s better one.

3.4 PRINCIPAL BEHIND COMPRESSION

A common characteristic of most images is that the neighboring pixels are correlated and therefore contain redundant information. The main task is that to find the less correlated representation of the image [26].

There are two fundamental components of compression are as follow:

- Redundancy Reduction
- Irrelevancy Reduction

Redundancy Reduction: - The aim of the redundancy reduction is removing duplication from the signal source such as image or video [26].

Irrelevancy Reduction: - It removes the parts of the signal that will not be noticed by the signal receiver, namely the Human Visual System [26].

Compression is achieved by the removal of one or more of the three basic data redundancies [26]:

1. **Coding Redundancy: -** Fewer bits to represent frequently occurring symbols.

2. **Inter pixel Redundancy: -** Neighboring pixels have almost same value.

3. **Psycho visual Redundancy: -** Human visual system cannot simultaneously distinguish all colors

Image compression technique is reducing the number of bits required to represent an image by using the advantages of different redundancies. The reverse process is applied to the compressed data to get the reconstructed image is called decompression. The objective of compression is to reduce the number of bits as much as possible and keeping the resolution and the visual quality of the reconstructed image as close to the original image as possible [26].

Image f(x,y) is fed into the encoder, which creates a set of symbols form the input data and uses them to represent the image. If n1 and n2 are the number of bits in two datasets that represent the same image, the Compression Ration

(CR) can be defined as:

$$\mathbf{CR} = \frac{\mathbf{n1}}{\mathbf{n2}}$$

And Relative Data redundancy (RD) of the original image can be defined as:

$$\mathbf{RD} = \frac{\mathbf{1}}{\mathbf{CR}}$$

There are three possibilities arise here:

(1) If n1=n2, then CR=1 and hence RD=0 which implies that original image do not contain any redundancy between the pixels.

(2) If n1<n2, then CR→∞ and hence RD>1 which implies considerable amount of redundancy in the original image.

(3) If n1>n2, then CR>0 and hence RD→-∞ which indicates that the compressed image contains more data than original image.

3.4.1 Coding Redundancy

A gray level image having n pixels is considered. The number of gray levels in the image is L (i.e. the gray levels range from 0 to L-1) and the number of pixels with gray level rk is nk. Then the probability of occurring gray level is rk is Pr(rk). If the number of bits used to represent the gray level rk is l(rk), then the average number of bits required to represent each pixel is.

$$L_{avg} = \sum_{K=0}^{L-1} l(r_k) P_r\ (r_k)$$

Where

$$P_r(r_k) = \frac{n_k}{n}$$

k = 0, 1, 2, ..., L-1

So, the number of bits required to represent the whole image is n x L_{avg}. Maximal compression ratio is achieved when L_{avg} is minimized. In such a way to write coding the gray levels Lavg is not minimized.

Generally coding redundancy is presented when the codes assigned to a gray level don't take full advantage of gray level’s probability (Pr (rk)). Therefore it almost always presents when an image's gray levels are represented with a straight or natural binary code. A natural binary coding of their gray levels assigns the same number of bits to both the most and least probable values, thus failing to minimize the above equation and resulting in coding redundancy.

3.4.2 Inter Pixel Redundancy

Another important form of data redundancy is inter pixel redundancy. It is directly related to the inter pixel correlations within an image because the value of any given pixel can be reasonable predicted from the value of its neighbors, the information carried by individual pixels is relatively small. Much of the visual contribution of a single pixel to an image is redundant; it could have been guessed on the basis of its neighbor's values. A variety of names, including spatial redundancy, geometric redundancy, and inter frame redundancies have been coined to refer to these inter pixel dependencies.

In order to reduce the inter pixel redundancies in an image, the 2-D pixel array normally used for human viewing and interpretation must be transformed into a more efficient but usually non-visual format. For example, the differences between adjacent pixels can be used to represent an image. Transformations of this type are referred to as mappings. They are called reversible if the original image elements can be reconstructed from the transformed data set.

To reduce the inter pixel redundancy various techniques are use, such as:

1. Run length coding
2. Delta compression
3. Constant area coding
4. Predictive coding

3.4.3 Psycho Visual Redundancy

Human perception of the information in an image normally does not involve quantitative analysis of every pixel or luminance value in the image. In general, an observer searches for distinguishing features such as edges or textural regions and mentally combines them into recognizable groupings. The brain then correlates these groupings with prior knowledge in order to complete the image interpretation process. So eye does not give proper respond with equal sensitivity related to all visual information. Some information has less relative importance than other information in normal visual processing. This information is said to be psycho visually redundant.

It can be eliminated without significantly impairing the quality of image perception. Psycho visual redundancies are fundamentally different from the coding Redundancy and inter pixel redundancy. Psycho visual redundancy is associated with real or quantifiable visual information. The elimination of psycho visual redundant data results in a loss of quantitative information. So it is an irreversible process.

To reduce psycho visual redundancy we use Quantizer. Since the elimination of psycho visually redundant data results in a loss of quantitative information. It is commonly referred to as quantization. As it is an irreversible operation i.e. visual information is lost, quantization results in lossy data compression.

3.5 NEEDS FOR IMAGE COMPRESSION

The following example described 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 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 example below clearly shows the importance of compression.

An image, 1024 pixel x 1024 pixel x 24 bit, without compression, would require 3 MB of storage space and 7 minutes for transmission, utilizing a high speed, 64 Kbps ISDN line. If the image is compressed at a 10:1 compression ratio, the storage requirement is reduced to nearby 300 KB and the transmission time drops to nearby 6 seconds. Seven 1MB images can be compressed and transferred to a floppy disk in less time than it takes to send one of the original files.

3.6 IMAGE COMPRESSION CONSIDERATION

Generally computer graphics related applications are generating digital photographs and other complex color images. It can generate very large file sizes. To solve the problem related to storage space, and the need to rapidly transmit image data across networks and over the Internet, has led to the development of a range of image compression techniques in order to reduce the physical size of files [27].

Most compression techniques are independent of specific file formats – indeed, many formats support a number of different compression types. They are an essential part of digital image creation, use, and storage [27].

There are a number of factors to be considered when using compression algorithms:

3.6.1 Efficiency

Most algorithms are particularly suited to specific circumstances; these must be understood if they are to be used effectively. For example, are more efficient at compressing monochrome images, whilst others yield better results with complex color images [28].

3.6.2 Lossiness

Graphics compression algorithms fall into two categories:

Lossy Compression: - It achieves the effect at the cost of a loss in image quality, by removing some image information[28].

Lossless Compression: - It reduces the size whilst preserving all of the original image information, and without degrading the quality of the image [28].

3.7 IMAGE COMPRESSION MODEL

Below figure represent the general image compression model:

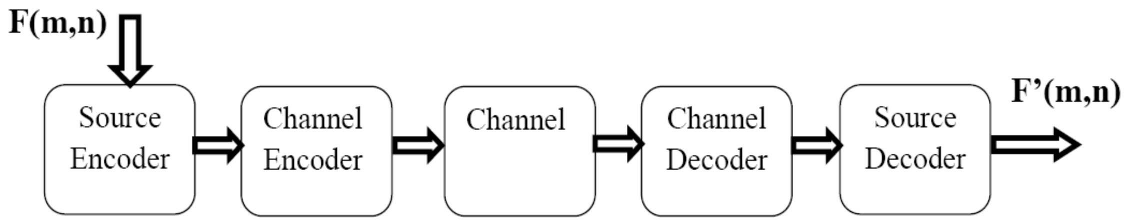


Figure – 3.5 Image compression model [29]

The image compression models are differing for the different image compression method. There are many general features that can be described which represent most image compression algorithms [29].

The following are the main steps for image compression:

1. **Source Encoder: -** It is used to remove the redundancy in the input image.
2. **Channel Encoder: -** It is used as overhead in order to combat channel noise. Parity bit is the example. By using this overhead, a certain level of immunity is gained from noise that is inherent in any storage or transmission system.
3. **Channel: -** It may be communication link or a storage system in this model.
4. **Channel Decoder: -** It performs the opposite work of channel encoder.
5. **Source Decoder: -** It performs the opposite work of source encoder.

3.7.1 Fidelity Criterion

In psycho visual redundancy data results in a loss of quantitative visual information. The loss of information is highly desirable. It is necessary required to measure the amount if data lost due to the compression process. This measurement is called Fidelity Criterion. There are two main categories of fidelity criterion as follow [29]:

1. Objective
2. Subjective

Objective Fidelity Criterion: - It involves a quantitative approach to error criterion. Root Mean Square Error (RMS) is the most common example of objective fidelity criterion. Objective fidelity criteria may be useful in analyzing the amount of error involved in a compression scheme [29].

Let *f(x,y)* represent an image and *f'(x,y)*denote an approximation of *f(x,y)* after compressing and decompressing. For any value of *x,y* the error *e(x,y)* between *f(x,y)* and *f'(x,y)* can be defined as:

$$e(x,y) = f'(x,y) - f(x,y)$$

so that the total error between the two images is

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]$$

Where the images are *M* x *N.* The Root Mean Square Error, e_{rms} between *f(x,y)* and *f'(x,y)* then is the square of the squared error averaged over the *M* x *N* array, or

$$e_{rms} = [\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x,y) - f(x,y)]^2]^{\frac{1}{2}}$$

A very much related measure of this criterion is the mean square signal to noise ratio of the compressed and decompressed image. If *f'(x,y)* is considered as sum of the original image *f(x,y)* and noise signal as *e(x,y)*, the root man squared signal to noise ratio of the output image denotes SNR_{rms} define as:

$$SNR_{rms} = \sqrt{\frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[\hat{f}(x,y) - f(x,y)]^2}}$$

Subjective Fidelity Criterion: - It involves a quality approach to error criterion based on a human observer. This can be done by showing a typical decompressed image to an appropriate group of viewer and then averaging their evaluations [29].

The ratings are given after the n evaluation of a compression scheme. There are two compression scale is given as follow:

1. **Absolute Comparison Scale: -** It is based solely on the decompressed image.
2. **Relative Comparison Scale: -** It involves viewing the original and decompressed images side by side in comparison.

Below table represent the different rating related to both the comparison scale.

The problem may be arises in subjective fidelity is that, it may vary from person to person. One person see a Passable, another person may view as a Fine.

Value	Rating	Description
1	Excellent	An image of extremely high quality. As good as desired.
2	Fine	An image of high quality, providing enjoyable viewing.
3	Passable	An image of acceptable quality.
4	Marginal	An image of poor quality, one wishes to improve it.
5	Inferior	A very poor image, but one can see it.
6	Unusable	An image so bad, one can't see it.

Table – 3.1 Absolute comparison scales [29]

Value	Rating
-3	Much Worse
-2	Worse
-1	Slightly Worse
0	Same
1	Slightly Better
2	Better
3	Much Better

Table – 3.2 Relative comparison scales [29]

3.7.2 Information Theory

This theory provides the theoretical basis for most data compression techniques. It is useful to answer the questions such as [29]:

- What is the minimum amount of data is needed to represent an image without loss of information?
- Theoretically what is the best compression possible?

The generation of information may be viewed as a probabilistic process. The input or source is viewed to generate one of N possible symbols from the source alphabet set A= {a, b, c..., z), {0, 1}, {0, 2, 4..., 280}, etc. in unit time. The source output can be denoted as a discrete random variable E, which is a symbol from the alphabet source along with a corresponding probability (z). When an algorithm scans the input for an occurrence of E, the result is a gain in information denoted by I(E), and quantified as[29]:

$$\mathbf{I\ (E) = \log\ (1/\ P\ (E))}$$

This relation indicated that the amount of information attributed to an event is inversely related to the probability of that event. For example in certain event (P (E) = 1) leads to I (E) = 0. This makes sense, since as we know that the event is certain, observing its occurrence adds nothing to our information. On the other hand, when a highly uncertain event occurs, a significant gain of information is the result [29].

An important concept called the entropy of a source. It is defined as the average amount of information gained by observing a particular source symbol. It allows an algorithm to quantize the randomness of a source. The amount of randomness is quite important because the more random a source is the more information that is needed to represent it. It turns out that for a fixed number of source symbols, efficiency is maximized when all the symbols are equally likely. It is based on this principle that code words are assigned to represent information [29].

3.8 BENEFITS OF COMPRESSION

Many benefits are provided by the compression. Some of them are as follow:

- It provides a potential cost savings associated with sending less data over networks where cost is really based on its duration.
- It not only reduces storage requirements but it also reduces the overall execution time.
- It also reduces the probability of transmission errors because fewer bits are transferred.
- It also provides a level of security against illicit monitoring.

3.9 IMAGE COMPRESSION TECHNIQUES

The image compression techniques are broadly classified into two categories depending on reconstruction of original image from the compressed image. These are:

1. Lossless Technique
2. Lossy Technique

3.9.1 Lossless Compression Techniques

When we are reduced the image data, one can except that the quality of image will automatically reduced. The loss of information is totally avoided in this lossless compression. When image data are reduced, image information is totally preserved.

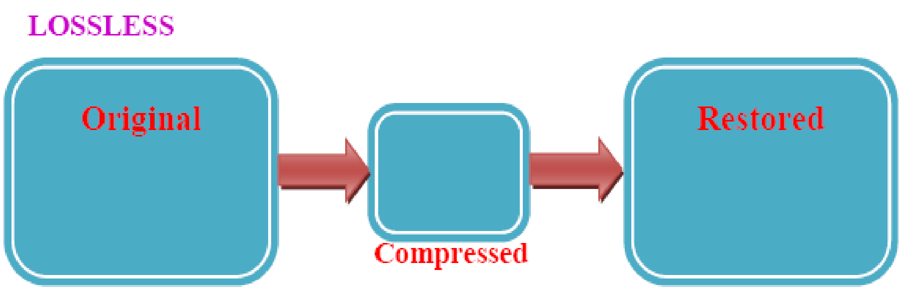


Figure – 3.6 Lossless image compression

In lossless compression techniques, the original image can be perfectly recovered from the compressed encoded image. These are also called noiseless since they do not add noise to the signal or image. It is also known as entropy coding since it use statistics/decomposition techniques to eliminate/minimize the redundancy.

Lossless methods yield lower compression ratios but preserve every pixel in the original image. The main method lossless compressions techniques are to that allow an image to be encoded into a smaller size and then decoded into the original format. The common lossless compression methods are Run-Length Encoding (RLE) and Lempel-Ziv-Welch (LZW). These methods use a process that replace sequences of repeating numbers in a graphics file with two numbers: one that specifies the length of the run (the number of times the value is repeated), and another that specifies the value itself.

The main principle behind reducing the size of the image is coding redundancy. Coding redundancy is based upon the idea that in an image there are some colors that are used frequently and other that are used infrequently. If there are repeated colors this soon would cause “repetition” and therefore a certain amount of redundancy would appear within the system. As such short codes are assigned to the frequently used colors and longer codes to the ones that are used less.

Following techniques are included in lossless compression:

1. Run Length Encoding (RLE)
2. Huffman Encoding
3. Lempel-Ziv-Welch (LZW) Coding
4. Area Coding

3.9.1.1 Run Length Encoding

It is a very simple compression method used for sequential data. It is very useful in case of repetitive data. This technique replaces sequences of identical symbols (pixels) called runs by shorter symbols. The run length code for a gray scale image is represented by a sequence {Vi, Ri} where Vi is the intensity of pixel and Ri refers to the number of consecutive pixels with the intensity Vi as shown in the figure[30].

25	25	25	68	68	68	68	68	73	73	73	73

{ 25, 3 }	{ 68, 5 }	{ 73, 4 }

The idea with run length encoding is rather than store:

XXXYYYYYYZZZZZZZZZ

We store: 3X6Y9Z

This gives good compression (3:1), but unfortunately where we have rapid changes

ABCDEF

We get: 1A1B1C1D1E1F

A negative compression (1:2)

RLE, better known as the ZIP file format, is a simple way to compress runs of the same byte. A number is used to indicate multiple times the same data. Fewer bytes are needed to express the same thing. RLE is a simple lossless compression scheme, but it is limited. RLE is most efficient with fewer levels of gray i.e. text [30].

3.9.1.2 Huffman Encoding

This algorithm is developed by D.A. Huffman. This is a general technique for coding symbols based on their statistical occurrence frequencies (probabilities). The pixels in the image are treated as symbols. The symbols that occur more frequently are assigned a smaller number of bits, while the symbols that occur less frequently are assigned a relatively larger number of bits. Huffman code is a prefix code. This means that the code of any symbol is not the prefix of the code of any other symbol[30].

This algorithm builds up a weighted binary tree according to their rate of occurrence. New code word is assigned to each element. Here the length of the code word is determined by its position in a tree. Input stream generate certain tokens occur than others. The token is most frequent. It becomes the root of the tree is assigned the shortest code. Each less common element is assigned a longer code word. The least frequent element is assigned a code word. It is twice as long as the input token [30].

The compression ratio achieved by Huffman encoding uncorrelated data becomes something like 1:2. On slightly correlated data, as on images, the compression rate may become much higher, the absolute maximum being defined by the size of a single input token and the size of the shortest possible output token as follow:

$$\textbf{Max. Compression} = \frac{\textbf{Token Size [bits]}}{\textbf{2 [bits]}}$$

While standard images with a limit of 256 colors may be compressed by 1:4 if they use only one color, more typical images give results in the range of 1:1.2 to 1:2.5[30].

3.9.1.3 LZW (Lempel – Ziv – Welch) Coding

This process used dictionary based coding. Dictionary based coding can be as follow:

1. **Static: -** In this coding, the dictionary is fixed during the encoding and decoding processes.
2. **Dynamic: -** In this coding, the dictionary is updated on fly.

Nowadays, the modified Lempel/Ziv codings are available. These all algorithms have a common way of working. The coder and the decoder both build up an equivalent dictionary of meta-symbols, each of which represents a whole sequence of input tokens. If a sequence is repeated after a symbol was found for it, then only the symbol becomes part of the coded data and the sequence of tokens referenced by the symbol becomes part of the decoded data later. As the dictionary is build up based on the data, it is not necessary to put it into the coded data. LZW is widely used in computer industry and is implemented as compress command on UNIX.

3.9.1.4 Area Coding

Area coding is an enhanced form of run length coding, reflecting the two dimensional character of images. This is a significant advance over the other lossless methods. For coding an image it does not make too much sense to interpret it as a sequential stream, as it is in fact an array of sequences, building up a two dimensional object. The algorithms for area coding try to find rectangular regions with the same characteristics. These regions are coded in a descriptive form as an element with two points and a certain structure [30].

The possible performance of this coding method is limited mostly by the very high complexity of the task of finding largest areas with the same characteristics. Practical implementations use recursive algorithms for reducing the whole area to equal sized subrectangles until a rectangle does fulfill the criteria defined as having the same characteristic for every pixel. This type of coding can be highly effective but it bears the problem of a nonlinear method, which cannot be implemented in hardware [30].

3.9.2 Lossy Compression Techniques

Lossy compressed methods give higher compression ratios, but less ability reproduces the original, uncompressed pixel for pixel. JPEG is the best known lossy compression standard and widely used to compress images. It is considerably more complicated than RLE. It produces higher compression ratios – even for images containing little or no redundancy. Except where every piece of information of a scan is critical – for example, scientific data – a scan must only provide enough information to meet the needs of the reproduction process and the viewer. The idea behind JPEG compression is to segregate the information in an image by level of importance, and then discard the less important information to reduce the overall quantity of data that must be stored. It does so by transforming a matrix of color values into a matrix of amplitude values corresponding precise frequencies in the image. JPEG standard was developed by a group that called upon experts from the printing, prepress, photographic, and design industries to identify at what point the compressed printed image differed from an uncompressed image. After few years, they examined thousands of printed pieces, which led the current standards of JPEG. The eye doesn't perceive all the subtle color shifts in a typical bitmapped image, so some of the detail can be discarded without affecting the overall information content.

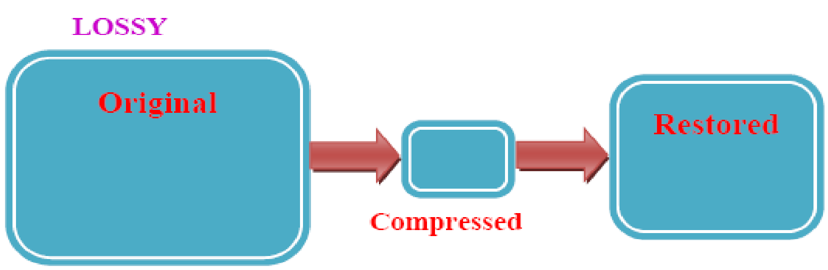


Figure – 3.7 Lossy image compression

In transform encoding, each image has run the mathematical transformation. It is just similar to Fourier Transformation. It separates the image information on gradual spatial variation of brightness from information with faster variation of brightness at edges of the image. In the next step, the slower changes in information are transmitted essentially lossless, but information on faster local changes is communicated with lower accuracy. The second step is called quantization. This step cannot be reversed when decompressing the data or image. The overall compression is 'lossy' or 'irreversible'.

Lossy schemes provide much higher compression ratios than lossless schemes. Lossy schemes are widely used since the quality of the reconstructed images is adequate for most applications. By this scheme, the decompressed image is not identical to the original image, but reasonably close to it.

In this prediction – transformation – decomposition process is completely reversible. The quantization process results in loss of information. The entropy coding after the quantization step, however, is lossless. The decoding is a reverse process. Entropy decoding is applied to compressed data to get the quantized data. Then de-quantization is applied to it and finally the inverse transformation to get the reconstructed image.

Major performance considerations of a lossy compression scheme include:

1. Compression ratio
2. Signal - to – noise ratio
3. Speed of encoding and decoding

Lossy compression techniques includes following schemes:

1. Transformation Coding (FT/DCT/Wavelets)
2. Vector Quantization
3. Fractal Coding
4. Block Truncation Coding
5. Sub band Coding
6. Segmented Image Coding
7. Spline Approximation Method

3.9.2.1 Transformation Coding

In this coding scheme, transforms such as DFT (Discrete Fourier Transform) and DCT (Discrete Cosine Transform) are used to change the pixels in the original image into frequency domain coefficients. These coefficients have several desirable properties. One is the energy compaction property that results in most of the energy of the original data being concentrated in only a few of the significant transform coefficients. This is the basis of achieving the compression. Only those few significant coefficients are selected and the remaining is discarded. The selected coefficients are considered for further quantization and entropy encoding. DCT coding has been the most common approach to transform coding [30].

A general transform coding scheme involves subdividing an NxN image into smaller nxn blocks. After that it performs the unitary transform of each sub-image. The main objective of the transform is to de-correlate the original signal. This de-correlation generally results in the signal energy being redistributed among only a small set of transform coefficients. In this way, many coefficients may be discarded after quantization and prior to encoding. Visually lossless compression can often be achieved by incorporating the Human Vision System (HVS) contrast sensitivity function in the quantization of the coefficients [30].

Generally four stages are performed in transform coding are as follow:

1. Image Subdivision

2. Image Transformation
3. Coefficient Quantization
4. Huffman Encoding

For a transform coding scheme, logical modelling is done in two steps:

Segmentation: - In which the image is subdivided in bi-dimensional vectors.

Transformation: - In which the chosen transform (e.g. KLT, DCT, and Hadamard) is applied.

Quantization can be performed in several ways. Most classical approaches use the following:

- Zonal Coding
- Threshold Coding

Zonal Coding: - It consists in the scalar Quantization of the coefficients. It belongs to a predefined area.

Threshold Coding: - It consist the choice of the coefficients of each block characterized by an absolute value exceeding a predefined threshold. Another possibility, that leads to higher compression factors, is to apply a vector Quantization scheme to the transformed coefficients.

3.9.2.2 Vector Quantization

The basic idea in this technique is to develop a dictionary of fixed-size vectors, called code vectors. A vector is usually a block of pixel values. A given image is then partitioned into non-overlapping blocks called image vectors. The index in the dictionary is used as the encoding of the original image vector. Thus, each image is represented by a sequence of indices that can be further entropy coded [30].

A vector quantizer can be defined mathematically as a transform operator T from a K-dimensional Euclidean space R^K to a finite subset X in R^K made up of N vectors. This subset X becomes the vector codebook, or, more generally, the codebook [30].

The choice of the set of vectors is of major importance. The level of distortion due to the transformation T is generally computed as the most significant error (MSE) between the "real" vector x in R^K and the corresponding vector x' = T(x) in X. This error should be such as to minimize the Euclidean distance d.

3.9.2.3 Fractal Coding

The essential idea in this technique is to decompose the image into segments by using standard image processing techniques such as color separation, edge detection, and spectrum and texture analysis. Then each segment is looked up in a library of fractals. The library actually contains codes called Iterated Function System (IFS) codes, which are compact sets of numbers. Using a systematic procedure, a set of codes for a given image is determined. IFS codes are applied to a suitable set of image blocks yield an image that is a very close approximation of the original. This scheme is highly effective for compressing images that have good regularity and self similarity.

Dr Michael Barnsley reported the use of "Iterated Function System" for image compression and synthesis. Using sets of affine transformations developed for a given image, and a principal result known as the "collage theorem", intra frame compressions in excess of 10,000:1 and inter frame compression in excess of 1,000,000:1 were reported. The collage theorem states that if an image can be covered with compressed affine transformations of itself, then the image can be reconstructed by computing the attractor of this set of affine transformations.

3.9.2.4 Block Truncation Coding

In this scheme, the image is divided into non overlapping blocks of pixels. Threshold and reconstruction values are determined for each block. The threshold is usually the mean of the pixel values in the block. A bitmap of the block is derived by replacing all pixels whose values are greater than or equal to the threshold by 1. Then for each segment (group of 1s and 0s) in the bitmap, the reconstruction value is determined. This is the average of the values of the corresponding pixels in the original block.

3.9.2.5 Sub band Coding

In this scheme, the image is analyzed to produce the components containing frequencies in well-defined bands, the sub bands. Subsequently, quantization and coding is applied to each of the bands. The advantage of this scheme is that the quantization and coding well suited for each of the sub bands can be designed separately.

3.9.2.6 Segmented Image Coding

Segmented image coding is a region-oriented coding, which considers images to be composed of regions of slowly varying image intensity, separated by image edges. The texture inside each region is approximated by a bi-variate polynomial. The coefficients of these polynomials and the location of the contours separating the image regions constitute the contour-texture image model [30].

The visually extremely important image contours are always well retained by this technique, which guarantees a reasonable image quality, even at very low bit rates. In contrast to some conventional compression techniques, region oriented techniques are very well suited for progressive image transmission, in the reconstruction phase; the image is gradually built up. This is important in, e.g., database applications, where the gradual image build-up allows the user to quickly decide whether or not the transmitted image is the correct one required [30].

3.9.2.7 Spline Approximation Methods

This method is fall in more general category of image reconstruction or sparse data interpolation. The basic concept is to interpolate data from a set of points coming from original pixel data or calculated in order to match some error criteria [30].

In order to apply this kind of technique to image coding, a good interpolant must be used to match visual criteria. Spline interpolation provides a good visual interpolant, not with standing its requiring a great computational effort.

Bilinear interpolation is easier to implement, while maintaining a very good visual quality. Regularization involves the minimization of an energy function in order to obtain an interpolant which presents some smoothness constraints; it can be combined with non-continuities along edges in order to preserve contour quality during reconstruction. Generally all interpolants computations require the solution of very large linear equation sets, even if related to very sparse matrices. This leads to the use of recursive solution such as relaxation or to the use of gradient descent algorithm[30].

The use of an interpolation algorithm for image coding is more interesting when related to techniques such as two source decomposition, where the image is modeled as the sum of two sources as follow:

Stationary Part: - It is coded by means of a prediction scheme that can be one of the previous described interpolants.

Residual Content: - It coming from non-stationeries such as edges. It can be coded through the use of a classical coding method.

Moreover two source decomposition is a very effective coding scheme as far as it shows a low tile effect that affects all block coding techniques when compression factors become higher [30].

CHAPTER – 4

EXPERIMENTAL STUDIES AND COMPARISON OF VARIOUS IMAGE COMPRESSION ALGORITHMS

4.1 WAVELET COMPRESSION ALGORITHM

4.1.1 Introduction

Wavelet compression is also known as Discrete Wavelet Transforms (DWT). It treats the image as a signal or wave. It involves the following steps:

- Transformation
- Quantization
- Encoding

The transform organizes the image information into a continuous wave, typically with many peaks and dips, and centre it on zero. The image is treated as a series of waves, one for each color channel (i.e., Red, Green, and Blue), and it may break up big images into large tiles for ease of processing. Having centered the wave, the transform records the distances from the zero line to points along the wave i.e. these distances are known as coefficients, and then takes the average between adjacent coefficients to produce a simplified version of the wave; in effect, it reduces the image's resolution or detail by one-half.

4.1.2 Historical perspective

In the history of mathematics, wavelet analysis shows many different origins. Much of the work was performed. As regarding the literature resource about wavelets, there are some excellent bibliographies [31].

In 1807, Joseph Fourier developed a method for representing a signal with a series of coefficients based on an analysis function, now it is known as Fourier

Transformation. He laid the mathematical basis from which the wavelet theory is developed.

After 1807, by exploring the meaning of functions, Fourier series convergence, and orthogonal systems, mathematicians gradually were led from their previous notion of frequency analysis to the notion of scale analysis. This sort of scale analysis is less sensitive to noise because it measures the average fluctuations of the signal at different scales.

In 1909, Alfred Haar mentioned wavelets in his PhD thesis. One property of the Haar wavelet is that it has compact support, which means that it vanishes outside of a finite interval. Unfortunately, Haar wavelets are not continuously differentiable which somewhat limits their applications.

In 1930, several groups working independently researched the representation of functions using scale -varying basis functions. Paul Levy found the scale-varying basic function called Haar basis function i.e. superior to Fourier basis functions.

In 1930, another research effort by Little wood, Paley, and Stein involved computing the energy of a function f(x) : [31]

$$\boldsymbol{energy} = \int |\boldsymbol{f}(\boldsymbol{x})|^2 \, \boldsymbol{dx}$$

Between 1960 and 1980, the mathematicians Guido Weiss and Ronald R. Coifman studied the simplest elements of a function space, called atoms, with the goal of finding the atoms for a common function and finding the “assembly rules" that allow the reconstruction of all the elements of the function space using these atoms [31].

In 1980, the work provided by David Marr imported an effective algorithm for numerical image processing using wavelets. The computation produced different results if the energy was concentrated around a few points or distributed over a larger interval. The researchers discovered a function that can vary in scale and can conserve energy when computing the functional energy.

In 1980, Grossman and Morlet, a physicist and an engineer, broadly defined wavelets in the context of quantum physics. These two researchers provided a way of thinking for wavelets based on physical intuition.

In 1981, the transformation method of decomposing a signal into wavelet coefficients and reconstructing the original signal again was derived by Jean Morlet and Alex Grossman.

In 1985, Stephane Mallat gave wavelets an additional jump-start through his work in digital signal processing. He discovered some relationships between quadrature mirror filters, pyramid algorithms, and orthonormal wavelet bases.

Inspired in part by these results, Y. Meyer constructed the first non-trivial wavelets. Unlike the Haar wavelets, the Meyer wavelets are continuously differentiable; however they do not have compact support.

A couple of years later, Ingrid Daubechies used Mallat's work to construct a set of wavelet orthonormal basis functions that are perhaps the most elegant, and have become the corner stone of wavelet applications today.

In 1986, Stephane Mallat and Yves Meyer developed a multi resolution analysis using wavelets. They mentioned the scaling function of wavelets for the first time; it allowed researchers and mathematicians to construct their own family of wavelets using the derived criteria.

Around 1998, Ingrid Daubechies used the theory of multi resolution wavelet analysis to construct her own family of wavelets. The set of wavelet orthonormal basis functions have become the cornerstone of wavelet applications today. With her work the theoretical treatment of wavelet analysis is as much as covered.

In 2006, “Hybrid Image Compression Using Fractal-Wavelet Prediction” was presented in International Conference on Information Security and Privacy by Liangbin Zhang, Lifeng Xi.

In 2007, “Image Compression using Modified Haar Wavelet-Base Vector Quantization” was published in ecti transactions on computer and information technology by Anusorn Jitkam and Satra Wongthanavasu.

In 2008, “Fractal Image Coding Based on Oriented Wavelet Sub-tree” was published by Jiang Shan, Shaung Kai Sun Li Wei.

In 2009, “The Design and Implementation of Fractal Image Coding Based on Wavelet” was presented in International Conference on Intelligent Human-Machine Systems and

Cybernetics by Haiming Gu, Liang Li.

In 2010, “Performance Comparison of Image Compression Using Wavelets” was published in International Journal of Computer Science & Communication by Preeti Aggarwal1 & Babita Rani.

In 2012, “Comparative Performance Analysis of Haar, Symlets and Bior wavelets on Image compression using Discrete Wavelet Transform” was published in International Journal of Computers & Distributed Systems by Jashanbir Singh Kaleka and Reecha Sharma.

4.1.3 Wavelet overview

Wavelets are analyzed with respect to the scale. In wavelet analysis, the scale is playing a special role. Wavelet algorithms can process data at different scales or resolutions. Many researchers of this area related to wavelet feel that, one is adopting a whole new perspective in data processing by using wavelet.

Wavelet is function. It is used to satisfy certain mathematical requirement. It is used for representing a data or other functions. It’s not a new idea. From early 1800’s the approximation using superposition of functions has been existing. Fourier has discovered that he could superpose sines and cosines to represent other functions.

To comprise the bases of Fourier analysis, the researchers and scientists have tried to fine more appropriate functions than the sines and cosines. It does a very poor job in approximating sharp spikes. With the help of wavelet analysis, we can use approximating functions that are contained neatly in finite domains. Wavelets are well-suited for approximating data with sharp discontinuities.

The wavelet analysis procedure is to adopt a wavelet prototype function, which is called an analyzing wavelet or mother wavelet. There are two types of analysis which are performed with the wavelet prototype such as temporal analysis and frequency analysis. Temporal analysis is performed with a contracted, high frequency version of the prototype wavelet. Frequency analysis is performed with a dilated, low frequency version of the same wavelet. Because the original signal or function can be represented in terms of a wavelet expansion, data operations can be performed using just the corresponding wavelet coefficients. If you want to select the best wavelets adapted to the data or try to truncate the coefficients below a threshold, your data is sparsely represented. This sparse coding is used to make wavelets an excellent tool in the field of data compression.

The use of wavelets is evident in other applied fields such as astronomy, signal and image processing, sub band coding, music, speech discrimination, nuclear engineering, magnetic resonance imaging, radar, earth quake prediction, optics, turbulence, human vision and pure mathematics applications such as solving partial differential equations.

The image wavelet compression process consists of the following steps: Wavelet Transformation, Quantization and Entropy Coding. The figure given below represents the block diagram of general compression and decompression process and wavelet based image compression process.

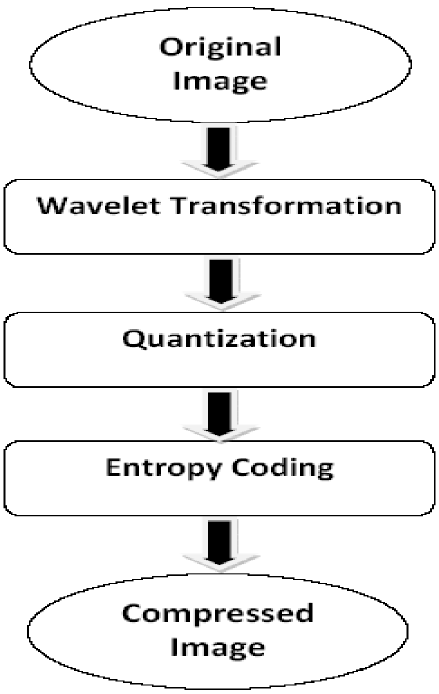


Figure – 4.1 Wavelet image compression routines

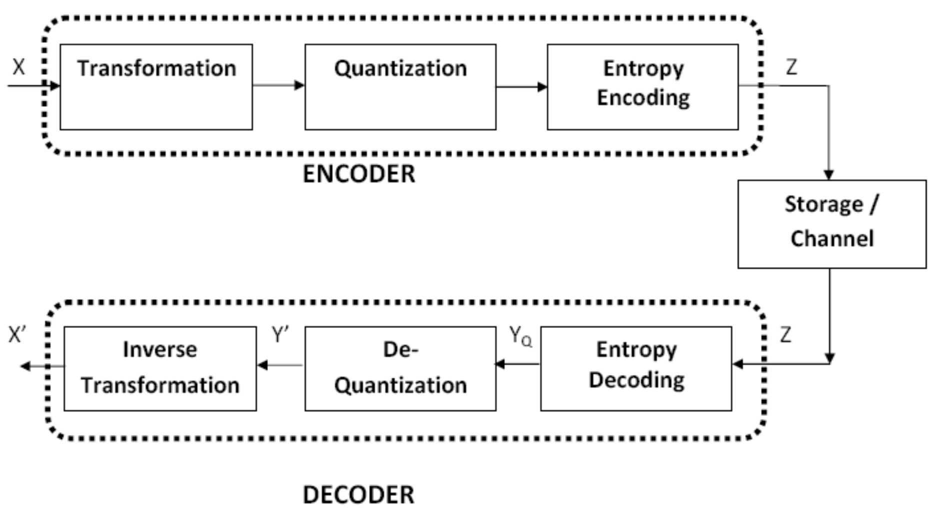


Figure – 4.2 Block diagram of general compression and decompression process [32]

In this thesis, we have written most of the equations by refereeing to the book “Digital Image Processing" by R. C. Gonzalez and R. E. Woods and S. L. Eddins.

4.1.4 Wavelet Property

The following are most important properties of wavelets:

- Admissibility Conditions
- Regularity Conditions

These are the two different properties which give the name wavelets. It can be seen that square integral function Ψ(t) satisfies the admissibility condition [33].

$$\int \frac{|\Psi(w)|^2}{|w|} \, dw < +\infty$$

This can be used to analyze the signal and after that reconstruction of the signal without any loss of information. In above equation Ψ(w) stands for the Fourier Transform of Ψ(t). The admissibility condition shows that the Fourier Transform Ψ(t) vanishes at the zero frequency, i.e.[33].

$$\left| |\Psi(w)|^2 \right|_{w=0} = 0$$

It means that wavelets must have a band pass like spectrum. It is used to construct the efficient wavelet transform.

In above equation, zero at the zero frequency means that the average value of the wavelet with respect to time domain must be zero, i.e.

$$\int \Psi(t)dt = 0$$

In other words, Ψ(t) must be a wave.

The product of the wavelet transformation with respect to time bandwidth is the square of the input signal. The wavelet transformation of the one dimensional function is two dimensional and the wavelet transformation of the two dimensional function is four dimensional.

The wavelet property, admissibility condition, is not a desirable property in most practical applications. It imposes some additional conditions in the wavelet functions. It decreases the wavelet transformation quickly while scale s is decreasing. This is the regularity condition. It states that the wavelet function must have some smoothness and concentration in both frequency and time domains. It is a very complex concept. It can be explained with the concept of vanishing moments.

$$\gamma(s,\tau) = \int f(t)\, \Psi^{*}_{s,\tau}(t)dt$$

In above equation, if we expand the wavelet transformation into the Taylor Series at level t=0 until the order n, we get [33]

$$\gamma(s,0) = \frac{1}{\sqrt{s}}\left[\sum_{p=0}^{n} f^{(p)}(0)\int \frac{t^p}{p!}\Psi\left(\frac{t}{s}\right)dt + O(n+1)\right]$$

In above equation $f^{(p)}$ is stands for the P^{th} derivative of function *f* and *O(n+1)* represent the rest of the expansion.

If we define the moments of the wavelet by M_p, where

$$M_p = \int t^p \; \Psi(t)dt$$

Then we can rewrite above equation of Taylor Series into the finite development, i.e.

$$\gamma(s,0) = \frac{1}{\sqrt{s}}\left[f(0)M_0 s + \frac{f^{(1)}(0)}{1!}M_1 s^2 + \frac{f^{(2)}(0)}{2!}M_2 s^3 + \cdots + \frac{f^{(n)}(0)}{n!}M_n s^{n+1} + O(s^{n+2}\ )\right]$$

In admissibility condition, we have already 0^{th} moment, i.e. $M_0 = 0$. So that the fist term of the right hand site in above equations must be zero. If we can manage to make the other moments as zero up to the M_n, then the wavelet transform coefficients $\Psi(s,t)$ will decay as fast as s^{n+2} for a smooth signal *f(t)*. This is known as the vanishing moments or approximation order in literature.

If any wavelet has *N* vanishing moments, then the approximation order of the wavelet transform is also *N*. The moments do not have to number of vanishing moments required depends heavily on the application [34].

As a summary of these both conditions, regularity condition gave us the wave and the vanishing moments gave us the fast decay of let. If we put both together then it gave the wavelet.

4.1.5 Image Quality Evaluation

The image quality can be evaluated following two different methods:

- Objective
- Subjective

Objective Method: - These methods are based on computable distortion measures. A standard objective measure of image quality is reconstruction error. Suppose that one has a system in which an input image element block { *x(n)* }, *n=0,1….,N-1* is reproduces as { *y(n)* }, *n=0,1…,N-1*. The reconstruction error *r(n)* is defined as the difference between *x(n)* and *y(n)*.

$$r(n) = x(n) - y(n)$$

The variances of *x(n), y(n)* and *r(n)* are $\sigma^2_{x,}$ σ^2_y and σ^2_r. In the special case of zero-means signals, variances are simply equal to respective means-square values over appropriate sequence length *M*.

$$\sigma_z^2 = \frac{1}{M}\sum_{n=1}^{M} z^2\,(n)\,, \qquad z = x, y, \text{or } r$$

A standard objective measure of coded image quality is Signal-to-Noise Ratio (SNR) which is defined as the ratio between signal variance and reconstruction error variance (MSE) usually expressed in decibels (dB).

$$\mathbf{SNR(dB)} = \mathbf{10log_{10}}\left(\frac{\sigma_x^2}{\sigma_r^2}\right) = \mathbf{10log_{10}}\left(\frac{\sigma_x^2}{\mathbf{MSE}}\right)$$

When the input signal is an *R*-bit discrete variable, the variance or energy can be replaced by the maximum input symbol energy $(2^R-1)^2$. For the common case of 8 bits per picture element of input image, the Peak Signal-to-Noise Ratio (PSNR) can be defined as

$$\mathbf{PSNR(dB)} = \mathbf{10log_{10}}\left(\frac{\mathbf{255^2}}{\mathbf{MSE}}\right)$$

SNR is not adequate as a perceptually meaningful measure of picture quality, because the reconstruction errors in general do not have the character of signal independent additive noise, and the seriousness of the impairments cannot be measured by a simple power measurement [37].

Small impairment of an image can lead to a very large value of σ^2_r and, consequently, a very small value of PSNR, in spite of the fact that the perceived image quality can be very acceptable. In fact, in image quality is perceptual quality. The distortion is specified by Mean Opinion Score (MOS) or by Picture Quality Scale (PQS) [38] [39].

In addition to the commonly used PSNR, we chose to use a perception based subjective evaluation, quantified by MOS, and a perception based objective evaluation, quantified by PQS, for the set of distorted images [40].

The double stimulus impairment scale method uses reference and test conditions which are arranged in pairs such that the first in the pair is the unimpaired reference and the second is the same sequence impaired. The method uses the five grade impairment scale with proper description for each grade:

Grade	Meaning
5	imperceptible
4	perceptible, but not annoying
3	slightly annoying
2	annoying
1	very annoying

Table – 4.1 Grade of image quality evaluation

MOS for each test condition and test image are calculated

$$\mathbf{MOS} = \sum_{i=1}^{5} i \cdot p(i)$$

Where *i* is grade and

p(i) is grade probability.

Subjective Method: - Subjective assessments methods of image quality are experimentally difficult and lengthy, and the results may vary depending on the test conditions. Instead of MOS, it used PQS methodology which was developed in the last few years for evaluating the quality of compressed images. It combines various perceived distortions into a single quantitative measure. The PQS methodology uses some of the properties of HVS relevant to global image impairments, such as random errors, and emphasizes the perceptual importance of structured and localized errors. PQS is constructed by regressions with MOS, which is a five level grading scale. PQS closely approximates the MOS in the middle of the quality range. For very-high-quality images, it is possible to obtain values of PQS larger than 5. At the low end of the image quality scale, PQS can obtain negative values also. It was the reason that it uses subjective evaluation.

4.1.6 Fourier Analysis

The Fourier representation of the functions as a sure position of sines and cosines has become more popular everywhere for analytical and numerical solutions of differential equations. It also used for the analysis and treatment of different communications signals.

Fourier represents the functions in the following transforms:

- Fourier Transforms (FT)
- Discrete Fourier Transforms (DFT)
- Windowed Fourier Transforms (WFT)
- Fast Fourier Transforms (FFT)

4.1.6.1 Fourier Transforms (FT)

The main utility of Fourier transforms is, it can analyze the signal in the time domain for its frequency content. It is the main ability of Fourier transforms. It works as, first it translates the function into the time domain, and then it translates the function into the frequency domain. The signal is also analyzed for its frequency content because Fourier coefficient of the transformed function represents the contribution of each sine and cosine function at each frequency. The inverse Fourier transform of signal is also possible. It can transform the signal or data from the frequency domain into the time domain.

4.1.6.2 Discrete Fourier Transforms (DFT)

The Discrete Fourier Transform (DFT) estimates the Fourier transform of a function from a finite number of its sampled points. This sample points may be typical of what the signal looks like at all other times.

It has the symmetric property. These properties are almost same as the Continuous Fourier Transform (CFT). Inverse Discrete Fourier Transformation formula is easily derived and calculated by using the one for the Discrete Fourier Transform because both the formulas are identical and almost the same formula.

4.1.6.3 Windowed Fourier Transforms (WFT)

Here we take the example to understand the Windowed Fourier Transforms (WFT). If *f(t)* is a non-periodic signal, the summation of the periodic functions, sine and cosine, does not represent the signal accurately. We can extend the signal to become a periodic but it requires the additional continuity at the end points of the signal.

The WFT is the solution of the problem for representing the non-periodic signal in a better way. It can give the information about the signals simultaneously in both time and frequency domain.

In WFT, the input signal *f(t)* can be divided into different sections. Each section can be analyzed for its frequency content separately. If the signal has sharp transitions, it windows the input data, so that the sections coverage to zero at the end points. This windowing is accomplished through the weight function that places less emphasis near the interval's end points than at the middle. The effect of the window is to localize the signal in time.

4.1.6.4 Fast Fourier Transform (FFT)

To approximate the function by using samples and to approximate the Fourier integral by using DFT, it requires to apply a matrix with the order of the number sample points n. To increase the number of sample points quickly, it multiplies an $n \times n$ matrix by a vector cost on the order of n^2 arithmetic operations.

If the samples are uniformly spaced, then the Fourier matrix can be factored into a product of just a few sparse matrices. The resulting factors can be applied to the vector in a total of order *nlogn* arithmetic operations. This is known as Fast Fourier Transform (FFT).

4.1.7 Types of Wavelets

Several families of wavelets that have proven to be especially useful are included in the wavelet toolbox. Here five different types of wavelets are used for image compression. The details of these wavelet families have been shown below [41]:

- Haar
- Daubechies
- Biorthogonal
- Symlets
- Coiflet

4.1.7.1 Haar Wavelet

The first DWT was invented by the Hungarian mathematician Alfred Haar in 1909. Haar wavelet is the first and simplest method. It is the simplest of all wavelets and its operation is easy to understand. It has its own limitations also. It is piecewise constant and hence produces irregular, blocky approximations [42].

There are several other wavelets available like the Daubechies wavelet, Donoho's wavelet, Meyer wavelet, etc. However these wavelets are not easy to comprehend and are also computationally intensive [42].

4.1.7.1.1 1-D Haar Wavelet Transform

Scaling Function:

The figure given below shows the 1-D Haar Scaling Function:

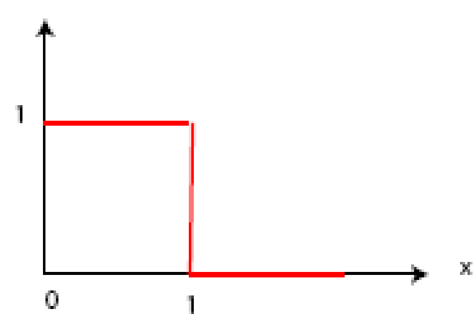


Figure – 4.3 1D Haar Scaling Function [42]

The 1-D Haar wavelet of above figure is expressed as shown below:

For ϕ: R → R :

$$\emptyset(t) = \begin{cases} 1 & t \in [0,1) \\ 0 & t \notin [0,1) \end{cases}$$

The Haar scaling function is given by

$$\emptyset_{j,k}(t) = 2^{j/2}\,\emptyset(2^j t - k)$$

Here j refers to the dilation and k refers to the translation.

The above scaling function can be expressed as[42]:

$$\emptyset_{j,k}(t) = \sum_n h_{\emptyset}(n) 2^{(j+1)/2} \emptyset(2^{j+1}t - n)$$

Wavelet Function:

The figure given below shows the 1-D Haar Wavelet Function:

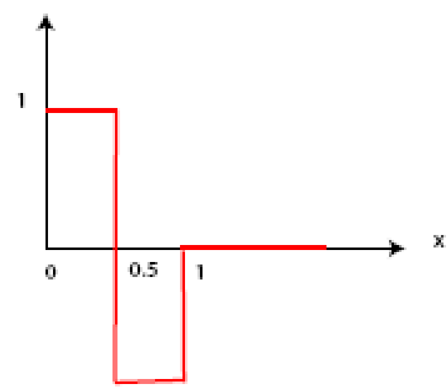


Figure – 4.4 1D Haar Wavelet Function [42]

For Ψ: R → R :

$$\Psi(t) = \begin{cases} 1 & t \in [0, \frac{1}{2}) \\ -1 & t \in [\frac{1}{2}, 1) \\ 0 & t \notin [0, 1) \end{cases}$$

The wavelet function in general can be expressed as:

$$\Psi_{j,k}(t) = 2^{j/2}\Psi(2^j t - k)$$

Where $\Psi_{j,k}$ belongs to the W_j subspace and Ψ belongs to the W_o subspace[42].

4.1.7.1.2 2-D Haar Wavelet Transform

Scaling and Wavelet Functions:

The 2D scaling and wavelet functions are expressed as follows:

$$\emptyset_{j,m,n}(x, y) = 2^{j/2}\emptyset(2^j x - m, 2^j y - n)$$

$$\Psi^i_{j,m,n}(x, y) = 2^{j/2}\Psi^i(2^j x - m, 2^j y - n), i = \{H, V, D\}$$

Where H represents the Horizontal Components, V represents the Vertical Components and D represents the Diagonal Components [42].

4.1.7.2 Daubechies (db4) Wavelet

It is a family of orthogonal wavelets. It defines the discrete wavelet transform. It can be characterized by a maximum number of vanishing moments for some given support. Each wavelet type of this class has a scaling function which is known as father wavelet. It generates an orthogonal multi resolution analysis.

$$Ci = g_0 s2i + g_1 s2i + 1 + g_2 s2i + 1 + g_3 s2i + 1$$

$$C[i] = g_0 s[2i] + g_1 s[2i + 1] + g_2 s[2i + 2] + g_3 s[2i + 3]$$

The wavelet function value is calculated from each iteration in the wavelet transform step. The index i is incremented by two with each iteration, and wavelet function values are calculated.

4.1.7.3 Biorthogonal Wavelet

The biorthogonal family of wavelets exhibits the property of linear phase, which is needed for signal and image reconstruction. By using two wavelets, one for decomposition and the other for reconstruction instead of the same single one, interesting properties are derived. The figure given below on left side represent decomposition and on right side represent reconstruction.

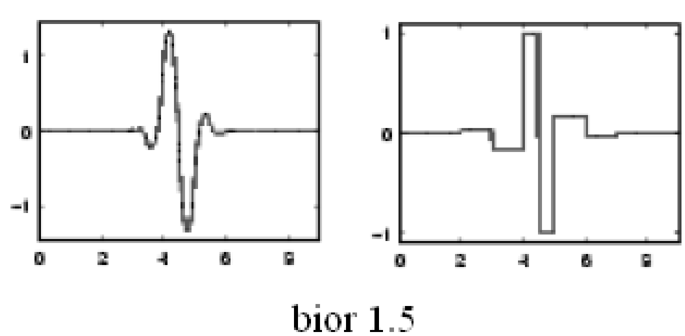


Figure – 4.5 Bior 1.5 Wavelet function waveform

4.1.7.4 Symlets Wavelet

The Symlets are nearly symmetrical wavelets proposed by Daubechies as modifications to the db family. The properties of the two wavelet families are similar. There are 7 different Symlets functions from sym2 to sym8. The figure given below shows the sym2 function.

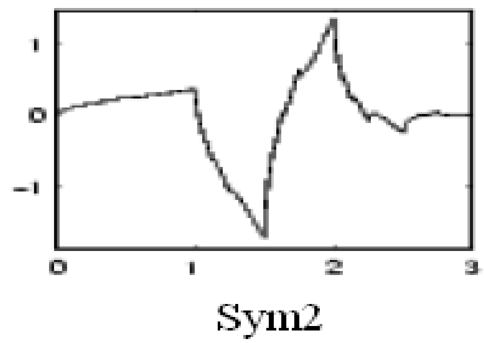


Figure – 4.6 Sym2 wavelet function waveform

4.1.7.5 Coiflet Wavelet

Coiflet are discrete wavelets designed by Ingrid Daubechies, at the request of Ronald Coifman, to have scaling functions with vanishing moments. This looks like

$$B_k=(-1)^k C_{N-1-k}$$

Where k is the coefficient index

B is a wavelet coefficient

C is a scaling function coefficient and

N is the wavelet index

4.1.8 Transformation

4.1.8.1 One-Dimensional Wavelet Transform

The one-dimensional discrete wavelet transform can be described in terms of a filter band as shown in the figure given below [43]:

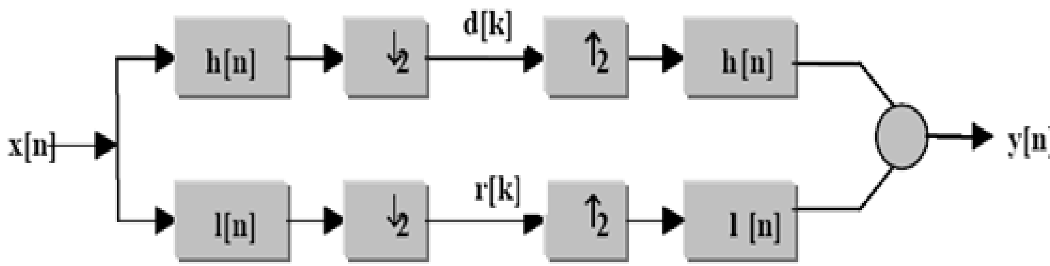


Figure – 4.7 One dimensional wavelet transform [43]

An input signal x[n] is applied to the low pass filter l[n] and to the analysis high pass filter h[n]. The odd samples of the outputs of these filters are then discarded, corresponding to a decimation factor of two. The decimated outputs of these filters constitute the reference signal r[k] and the detail signal d[k] for a new-level of decomposition. During reconstruction, interpolation by a factor of two is performed, followed by filtering using the low pass and high pass synthesis filters l[n] and h[n]. Finally, the outputs of the two synthesis filters are added together [43].

The above procedure can be expressed mathematically as the following equations.

$$d[k] = \sum_{n} x[n] \bullet h[2k - n]$$

$$r[k] = \sum_{n} x[n] \bullet l[2k - n]$$

$$x[n] = \sum_{n} (d[k] \bullet g[-n + 2k]) + (r[k] \bullet h[-n + 2k])$$

4.1.8.2 Multilevel Decomposition Wavelet Transform

In a multilevel decomposition, the above process is repeated. The previous level's lower resolution reference signal ri[n] becomes the next level sub-sampling input, and its associated detail signal di[n] is obtained after each level filtering. The figure given below shows this procedure [43].

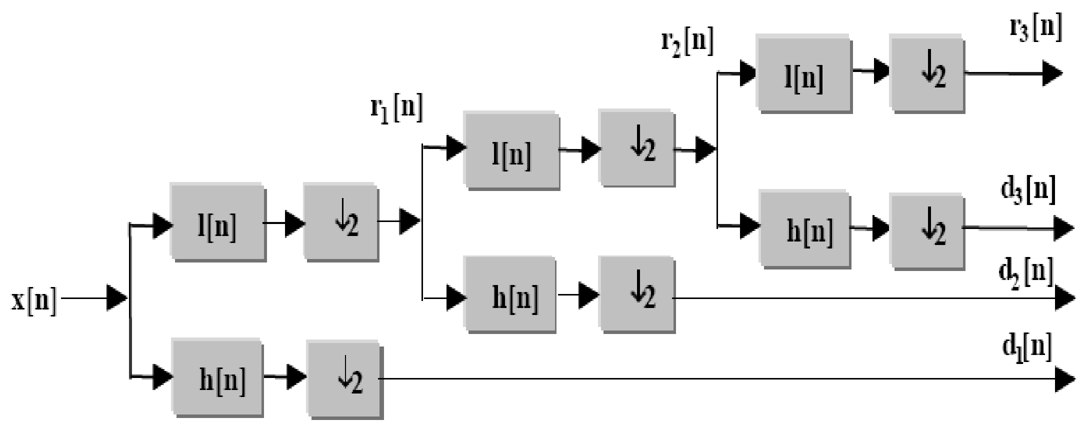


Figure – 4.8 Three level decomposition for wavelet transform [43]

The original signal x[n] is input into the low-pass filter l[n] and the high pass filter h[n]. After three levels of decomposition, a reference signal r3[n] with the resolution reduced by a factor of 23 and detail signals d3[n], d2[n], d1[n] are obtained. These signals can be used for signal reconstruction.

The three levels of wavelet transform implementation are shown in the figure given below [43]:

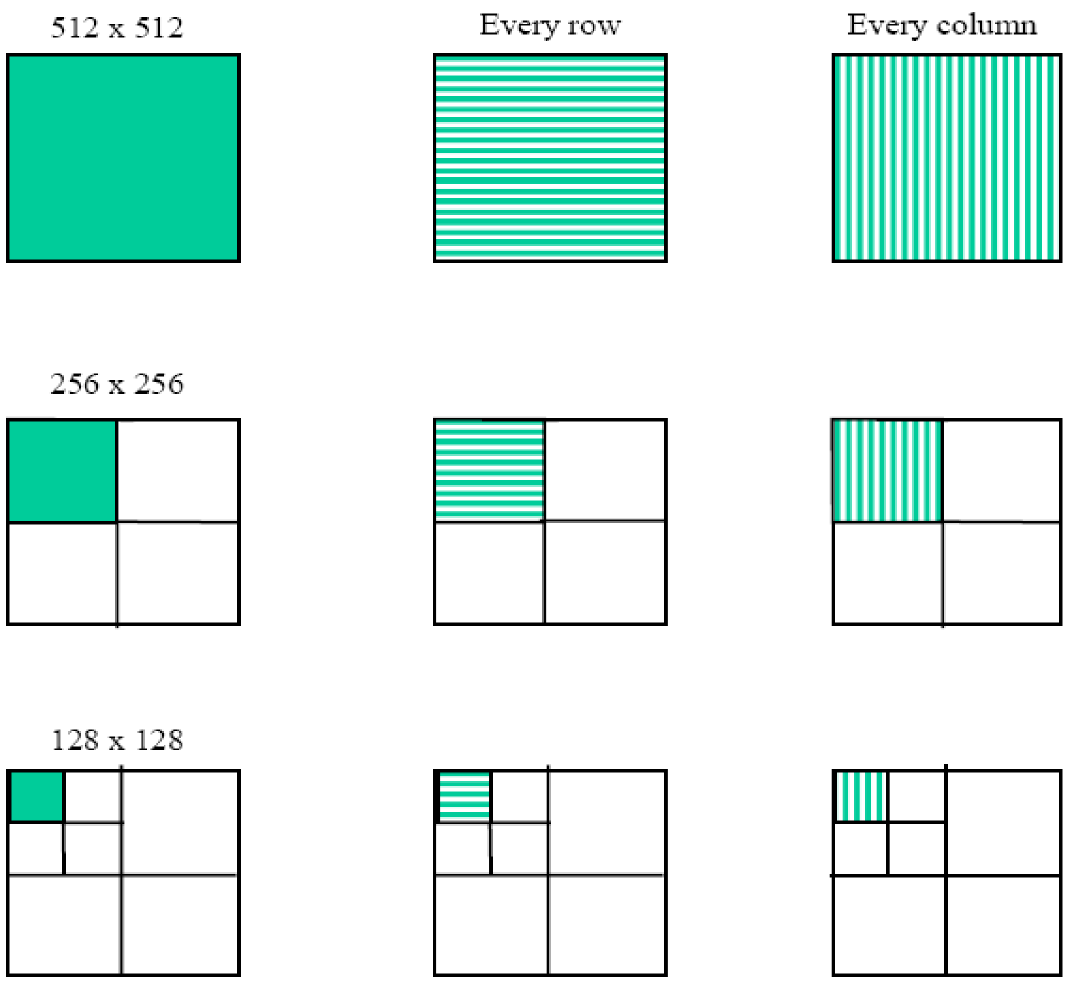


Figure – 4.9 Wavelet transform implementation [43]

The wavelet transformation can be divided into following three categories:

- Continues Wavelet Transformation
- Semi discrete Wavelet Transformation
- Discrete Wavelet Transformation

The above distinction among the various types of wavelet transformation depends on the way in which the scale and shift parameters are discredited.

4.1.8.3 Continues Wavelet Transform

The Continuous Wavelet Transform (CWT) is used to divide a continuous-time function into wavelets. Unlike the Fourier transform, it possesses the ability to construct a time frequency represented of a signal. It offers very good time and frequency localization [43].

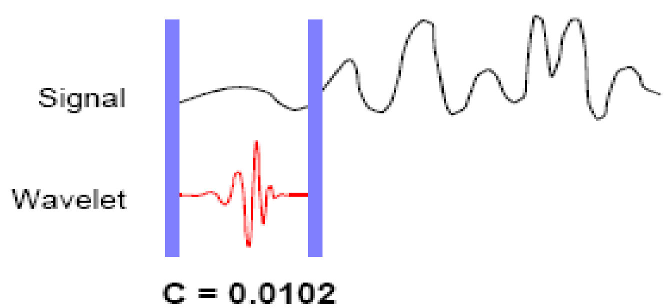


Figure – 4.10 Example of how a coefficient is found [43]

The wavelet analysis described in the introduction is known as the continuous wavelet transform. Formally it can be written as:

$$\gamma(s,\tau) = \int f(t)\Psi^{*}_{s,\tau}(t)dt$$

Where * denotes the complex conjugation.

The above equation shows how a function *f(t)* is decomposed into the set of basic functions $\Psi_{s,\tau}(t)$, which is called the wavelets. The variable scale s and translation τ, are the new dimensions after the wavelet transformed. The equation given below represents the inverse wavelet transforms [43].

$$f(t) = \iint \gamma(s,\tau)\,\Psi_{s,\tau}(t)\,d\tau\,ds$$

The wavelets are generated from a single basic wavelet $\Psi(t)$, the so-called mother wavelet, by scaling and translating:

$$\Psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\Psi\left(\frac{t-\tau}{s}\right)$$

Where s is the scale factor, τ is the translation factor and the factor $s^{-1/2}$ is for energy normalization across the different scales.

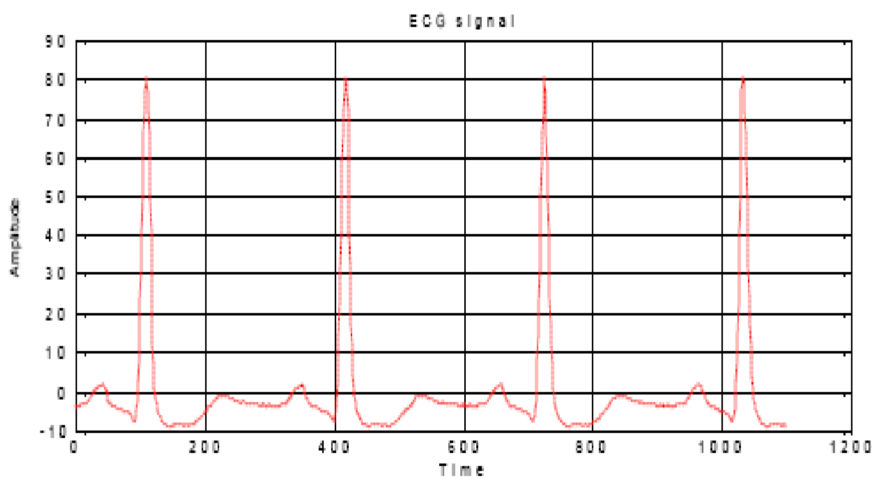


Figure – 4.11 Continuous wavelet transform [31]

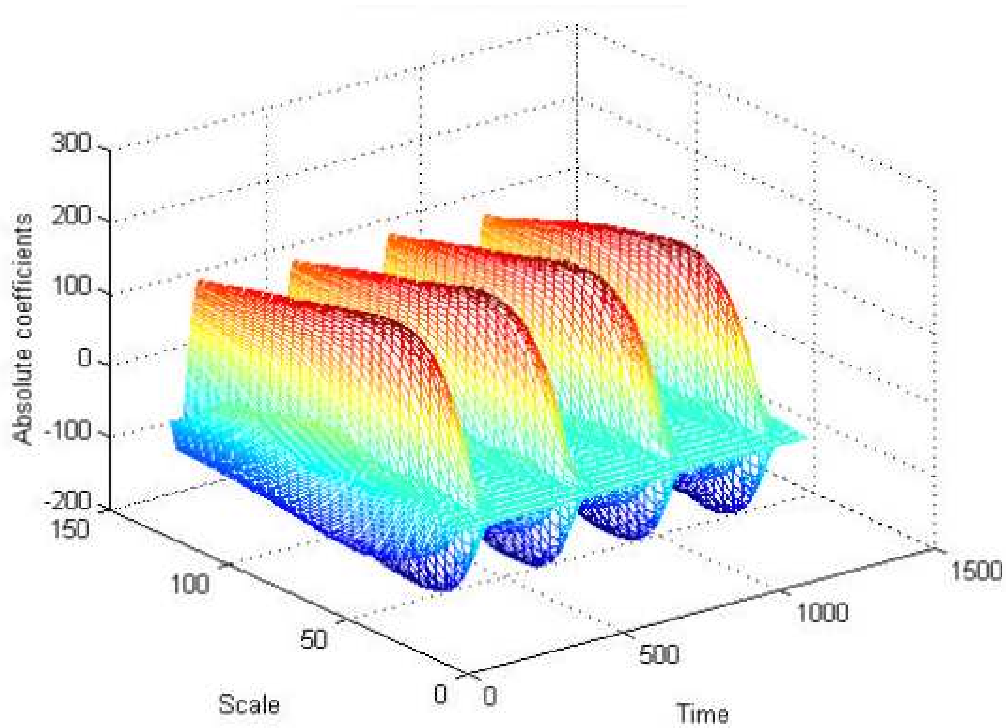


Figure – 4.12 Continuous wavelet analyses [31]

The above given figure represents the continuous wavelet analysis. In the top signal (gold standard) to be decomposed is shown. In the bottom corresponding wavelet coefficients are depicted [31].

4.1.8.4 Semi Discrete Wavelet Transform

It is more convenient to consider wavelet transformation for semi discredited values *a* and *b*. For example the dyadic scales $a=2^j$ and inter shifts $b=2^jk$ with $(j,k)\in Z^2$, It's called the scheme Semi discrete Wavelet Transform (SWT) [31].

The transform will be reversible if the corresponding set of templates defines a wavelet frame. In other words, the wavelet must be designed such as:

$$\boldsymbol{A}\|\boldsymbol{f}\|^2 \leq \sum_{\boldsymbol{a},\boldsymbol{b}} |\langle \boldsymbol{f}, \boldsymbol{\Psi}(\boldsymbol{a},\boldsymbol{b})\rangle|^2 \leq \boldsymbol{B}\|\boldsymbol{f}\|^2$$

Where *A* and *B* are two positive constants called frame bounds. Here we must integrate to get wavelet coefficients, the *f(t)* is still a continuous function [31].

4.1.8.5 Discrete Wavelet Transform

In a functional and numerical analysis, the DWT is any wavelet transform for which the wavelets are discretely sampled. The main advantage with respect to other wavelet transform is, it has over Fourier transforms is temporal resolution. It can capture both frequency as well as local information.

The one dimensional discrete wavelet transform of an input signal consists of discrete sets of coefficients. It is most often represented as an iterated structure of cascading low-pass and high pass filters. The cascading of the filters gives way to the multi-resolution analysis of the DWT and ultimately capability of denoising. The figure given below represents the general wavelet coefficient of analysis and synthesis filter bank [44].

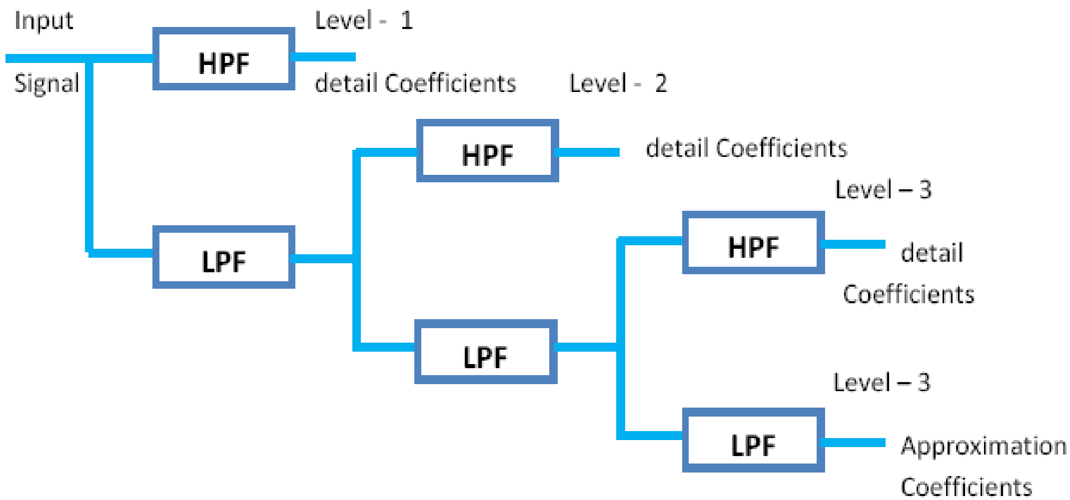


Figure – 4.13 Analysis filter bank [44]

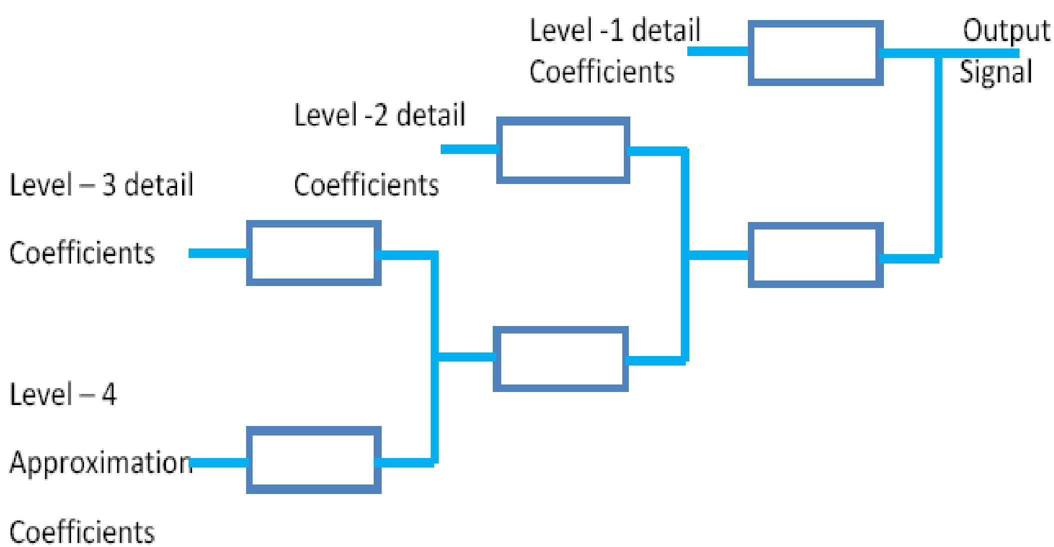


Figure – 4.14 Synthesis filter bank [44]

The figure given below shows the DWT analysis and synthesis system and 2-D DWT analysis filter bank.

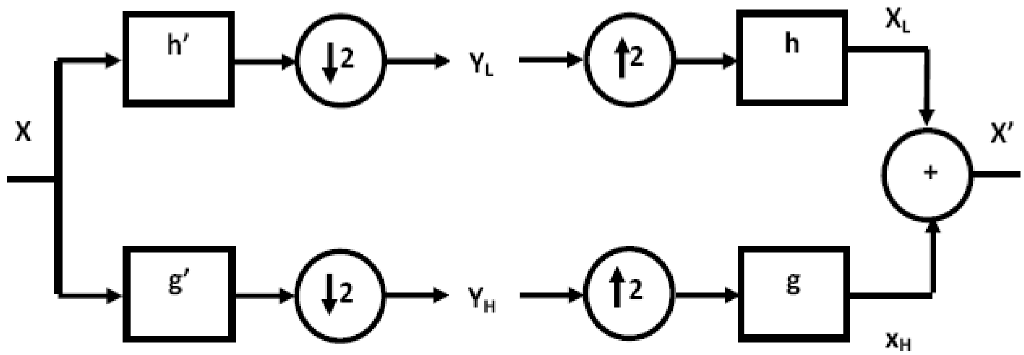


Figure – 4.15 DWT analysis and synthesis system [32]

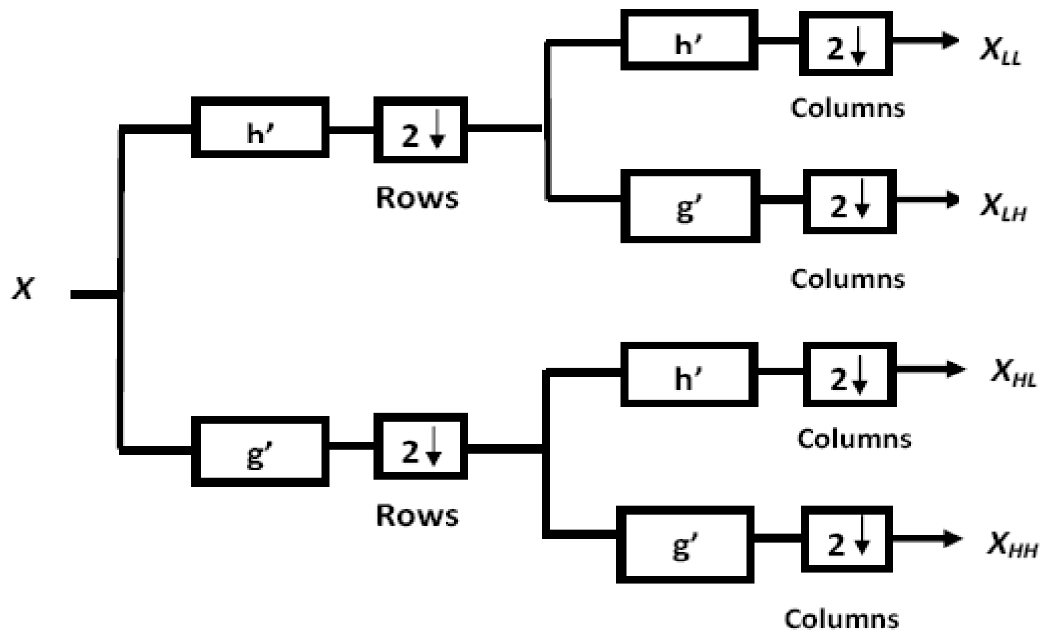


Figure – 4.16 2D DWT analysis filter bank [32]

The discrete wavelet transform of an image produces a multi-resolution representation where each wavelet coefficient represents the information content of the image at a certain resolution in a certain position. Similar to the 1-D case, cascades of high and low filters are applied in order to decompose the image. The figure given below shows the 2-D wavelet transforms [44]:

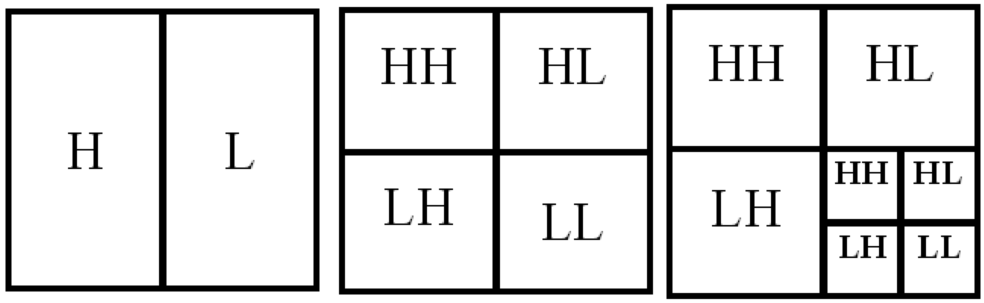


Figure – 4.17 2D Wavelet transform [44]

In the above given figure, left side is applying 1-D wavelet transform to all rows. Middle side is applying wavelet transform to all columns. Right side is applying wavelet transform in both directions.

Here, we have discrete function *f(n)* and the definition of Discrete Wavelet Transform (DWT) is represented by :

$$C(a,b) = C(j,k) = \sum_{neZ} f(n)\Psi j,k(n)$$

Where $\Psi_{j,k}$ is discrete wavelet defined as:

$$\Psi j,k(n) = 2^{-j/2}\Psi(2^{-j}n - k)$$

The parameters *a, b* are defined in such a way that $a=2^j$ and $b=2^jk$. Sometimes the analysis is called dyadic as well. The inverse transform is defined in a similar way as:

$$f(n) = \sum_{jeZ}\sum_{keZ} C(j,k)\Psi j,k(n)$$

For example, the image to be compressed has a dimension of M rows by N columns. The approach of the 2-D implementation of the DWT is to perform the one dimensional DWT in row direction and it is followed by a one dimensional DWT in column direction. A two dimensional row and column computation of DWT is depicted in the figure given below [45]:

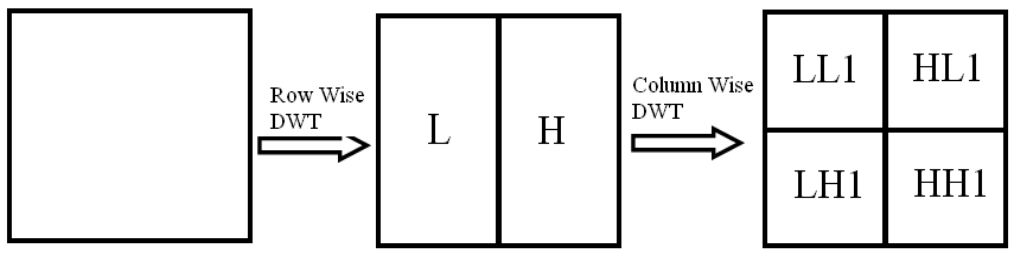


Figure – 4.18 2D row and column DWT [45]

In the above given figure, LL is a coarser version of the original image and it contains the approximation information which is in low frequency. LH, HL, and HH are the high-frequency sub band containing the detail information. Further computations of DWT can be performed as the level of decomposition increases. This concept is also shown in the figure given below [45]:

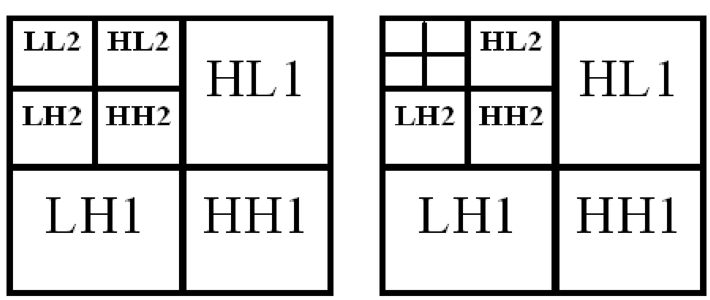


Figure – 4.19 Row and column decomposition [45]

In the above given figure, the second level (left) and third level (right) decompositions based on the principle of multi resolution analysis show that the LL1 sub band shown in above 2-D row and column DWT figure is decomposed into four smaller sub bands: LL2, HL2, LH2, and HH2.

4.1.9 Quantization

Image quantization is the conversion of elements of the transformed image into a data representation that requires less data storage than the original image. Quantization is achieved by the following two step process:

1. Normalization
2. Integer Conversion

All transform methods concentrate most of the image energy in the low-frequency regions. Most of the data in the high-frequency regions have values that are negligible with respect to the values of the low-frequency regions. These negligible values represent details of the original image. They are of such fine resolution that the human visual system cannot distinguish when one compares the perfectly reconstructed image and the reconstructed image in which only transformed values of significant magnitude were retained.

Normalization: - It converts all negligible values to 0 and simultaneously reduces the magnitude of the next level of lower high-frequencies while preserving the magnitude of the lowest frequency components. Thus, normalization is achieved by dividing each element in the transformed image by a value known as a “quantizer”. It is important to select a quantizer that will reduce all negligible values to 0.

The next step in quantization is the conversion of these normalized values to integer values. This process is achieved by the following operation:

$$f_q(x,y) = \textbf{int}\left[\frac{f(x,y)}{q(x,y)}\right]$$

Where *f(x, y)* is the transformed coefficient, *q(x, y)* is the quantizer, and $f_q(x, y)$ is the normalized value converted to an integer.

Integer Conversion: - It changes the normalized data into a set of integer values that can be encoded in such a way as to minimize their storage requirements. The quantizer must be adjusted to different frequency regions of the transformed image. In the low-frequency region, the quantizer must have relatively low value and, in the high-frequency region, the quantizer must have a relatively high value. This quantizer adaptation is accomplished by the use of “non-uniform” quantization. Non-uniform quantization utilizes two parameters:

- Shifting Size
- Step Size

Shifting Size: - This parameter is used to adjust the value of the quantizer for a given step size to different regions of the transformed image.

Step Size: - This parameter is used to determine the range of quantization. The minimum shifting size of 1 corresponding to the step size of 1 allows the quantizer to be adjusted for each element in the transformed image.

During the quantization, the image is divided into 10 blocks; the first four will be 64 x 64 pixels, then three will be 128 x 128, and the remaining three of 256 x 256 pixels. Every block executes the same quantization process. The figure given below represents this as a block diagram.

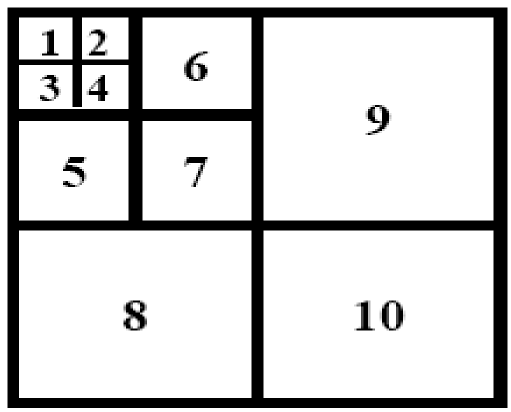


Figure – 4.20 Quantization and RLE block diagram

Before processing each block, some parameters should be prepared. Among this one parameter is the blockthresh. An array is used to hold these 10 block blockthreshes:

Blockthresh [10] = {0, 39, 27, 104, 79, 51, 191, 999, 9999, 99999}

For example: Block 1's blockthresh is 0, Block 10's blockthresh is 99999.

The next values that need to be calculated are the sixteen thresholds for each block, thresh1~thresh16; the formula to calculate this value is as follows:

$$thresh_n = \min + \frac{(\max - \min) \bullet n + 8}{2^4}$$

$$n: 1 \sim 16$$

The $thresh_n$ is the n^{th} thresh value in a block, and n value is from 1 to 16. Values min and max are the minimum and maximum pixel values within this block. After these numbers are computed, each block can run the quantization process. First, each input pixel's absolute value is compared with its corresponding blockthresh[n]. If it is smaller than the blockthresh value, the original pixel value is assigned to a constant value ZERO_MARK, which should be defined by the user.

The pixel values of the total image have been changed into integer values between 0 and 16, without changed the total image size. Also, because the values blockthresh[8], blockthresh[9] and blockthresh[10] are very large value, the absolute pixel value calculated from the wavelet transform routine will not exceed 99999. Therefore, these three block pixel values after the quantization will always be 16. Because of this, the last three blocks do not need to be processed in the image wavelet compression implementation design.

4.1.10 Encoding

Image Encoding is the final step in the image compression process using wavelet where integer values of the quantized image are converted to binary symbols that require, on the average, less storage than the integer representation. Typically, zero and positive integers less than 255 require only 1 byte of storage, while all other integer values require at least 2 bytes of storage. Binary symbols are composed of sequences of 0 and 1 each requiring only a single bit of storage. Thus, the bit requirements for the storage of a binary symbol equal the number of 0’s and 1’s that compose it.

The object of encoding is to assign short binary symbols to integer values with a relatively high probability and longer binary symbols to integer values with a relatively low probability.

There are a number of different encoding schemes available such as Variable Length Coding (VLC), Run Length Coding (RLC), etc.

The VLC method assigns binary symbols to the integer values the length of which is a function of its probability value. The RLC methods condenses streams of 0 values to special symbols called “tokens” whose binary representation requires less average storage than the assignment of a single bit to each of the consecutive 0’s.

The VLC method is applied to the nonzero values, while the RLC method is applied to the 0 values. The combination of these both methods give advantages of the quantization effect of concentrating nonzero integer values in a small region in the upper left-hand corner of the image.

In the case of the VLC method, a histogram analysis was first performed on the quantized image in order to determine the probability of all integer values. For the elements with the highest probability, a smaller binary symbol was used. The elements with the third, fourth, fifth and sixth highest probabilities were assigned the binary symbols: 00, 01, 10, and 11, etc. For the values of the lowest probabilities, longer binary symbols were assigned. However, the average number of bits per binary symbol was always equal to the total entropy of the encoded image.

Thus, the long binary symbols assigned to the integers with a low probability did not significantly increase the average storage requirements. Therefore, the RLC method reduces the number of 0's. If, for example, the following pattern of integers occurs: 13240000000000345, then the token (0:10) is assigned to 10 consecutive 0's.

The resulting representation is 1324(0:10)345. The token (0:10) will be converted to a binary symbol using the VLC method. When quantized integers and tokens for the sequences of 0's have been encoded, the data can be stored in a binary stream in which each binary symbol is distinguished from its adjacent symbols by a special prefix code. The quantized image can be recovered without loss by reversing this coding operation.

4.1.11 Experimental Study

Wavelet based image compression algorithm experiment is performed on Cameraman image. This experiment was performed in different algorithms such as Haar, Daubechies, Biorthogonal, Symlets and Coiflet at different level. In our experiment we perform this study up to five levels. The following images represent the different algorithm at different decomposition level and compressed image.

Figure – 4.21 Original cameraman image

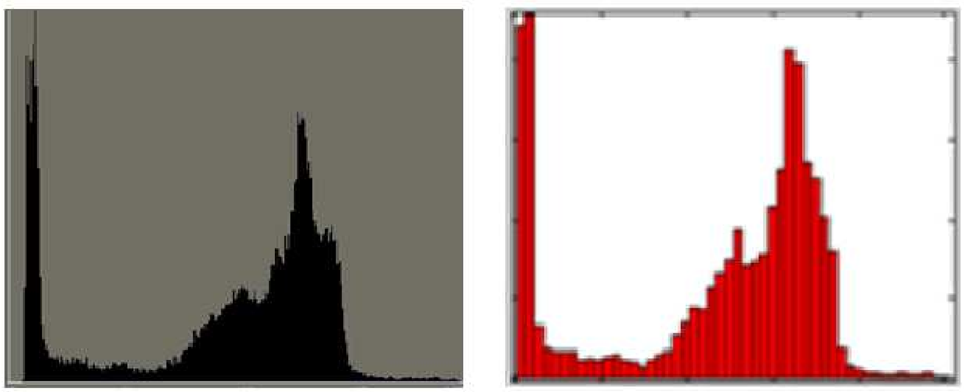
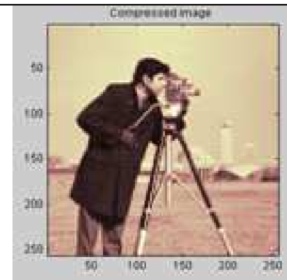
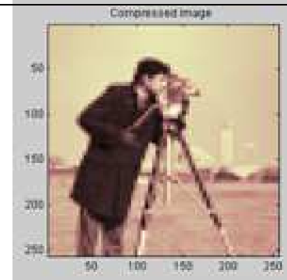
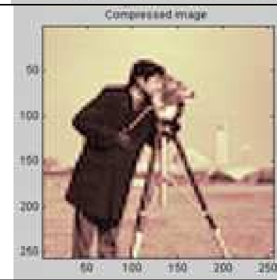
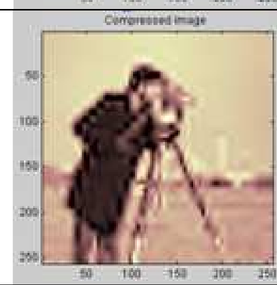
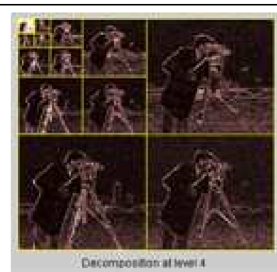
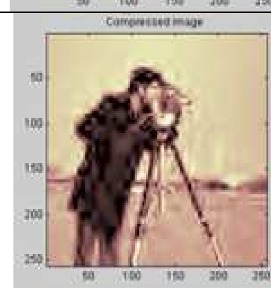


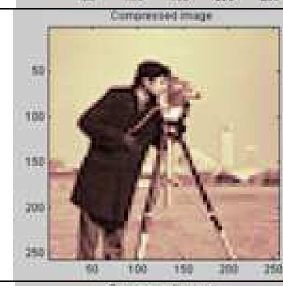
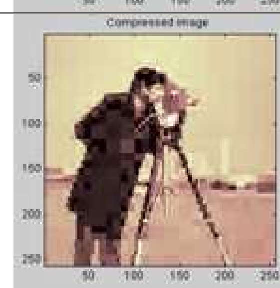
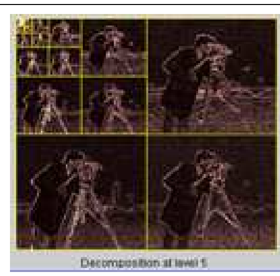
Figure – 4.22 Histogram of cameraman image

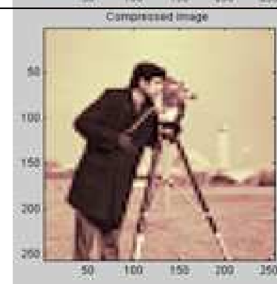
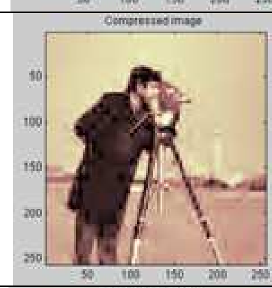
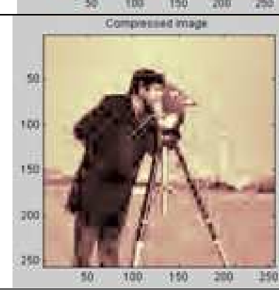
Mean	115.5
Median	141
Mode	8.56
Standard Deviation	64.01
Median Absolute Deviation	30
Mean Absolute Deviation	54.26

Table – 4.2 Statistics of cameraman image

Algorithm	Decomposition	Compressed Image
Haar Level – 1		
Haar Level – 2		
Haar Level – 3		
Haar Level – 4		
Haar Level – 5		

Daubechies Level – 1		
Daubechies Level – 2		
Daubechies Level – 3		
Daubechies Level – 4		
Daubechies Level – 5		

Biorthogonal Level – 1		
Biorthogonal Level – 2		
Biorthogonal Level – 3		
Biorthogonal Level – 4		
Biorthogonal Level – 5		

<table>
<tr><td>Symlets
Level – 1</td><td></td><td></td></tr>
<tr><td>Symlets
Level – 2</td><td></td><td></td></tr>
<tr><td>Symlets
Level – 3</td><td></td><td></td></tr>
<tr><td>Symlets
Level – 4</td><td></td><td></td></tr>
<tr><td>Symlets
Level – 5</td><td></td><td></td></tr>
</table>

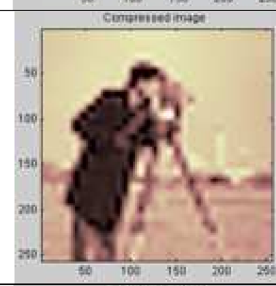
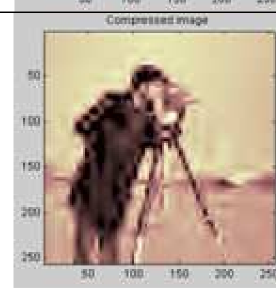
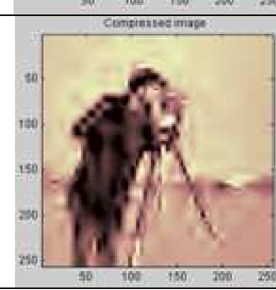
Coiflet Level – 1		
Coiflet Level – 2		
Coiflet Level – 3		
Coiflet Level – 4		
Coiflet Level – 5		

Figure – 4.23 Algorithms, decomposition level and compressed image

The following table shows the different wavelet based image compression algorithms at different decomposition level on cameraman image.

Compression Algorithm (size in KB)	Decomposition Level				
	1	**2**	**3**	**4**	**5**
Haar	25	11	9	10	10
Daubechies	26	12	10	10	9
Biorthogonal	61	54	50	49	50
Symlets	62	55	51	52	52
Coiflet	62	55	50	49	49

Table – 4.3 Cameraman image compression result

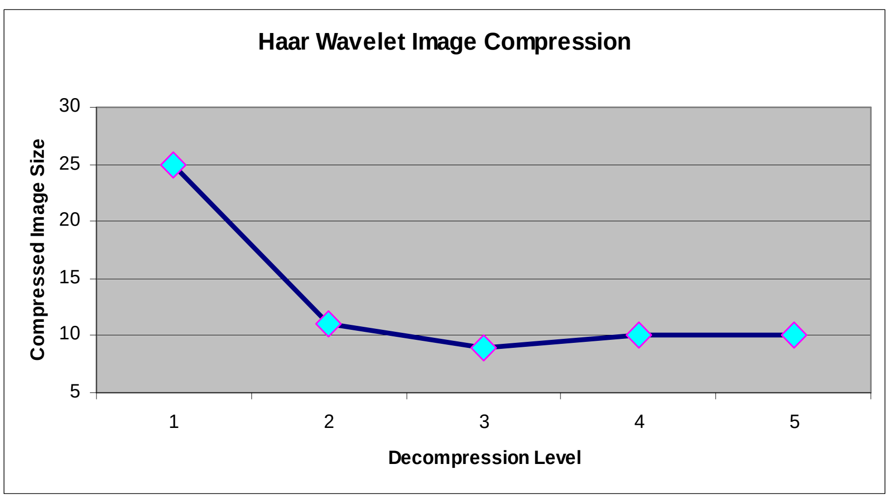


Figure – 4.24 Haar wavelet image compression chart

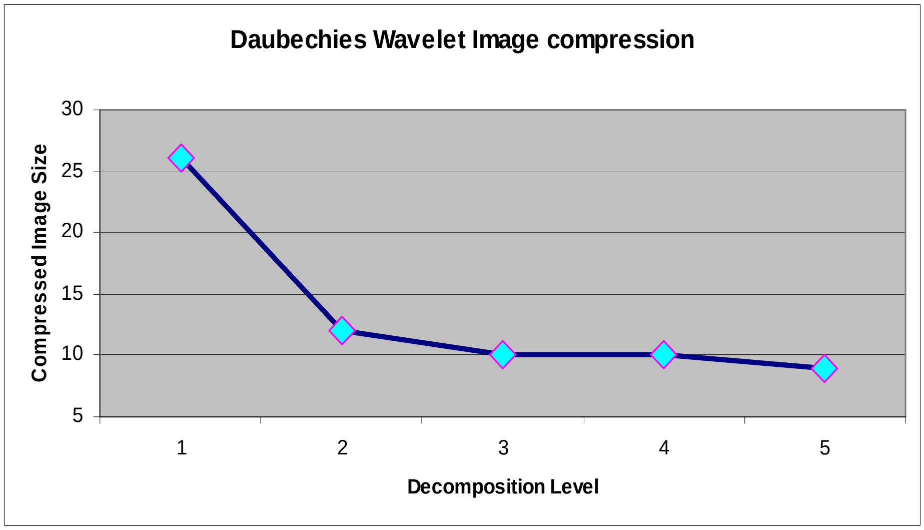


Figure – 4.25 Daubechies wavelet image compression chart

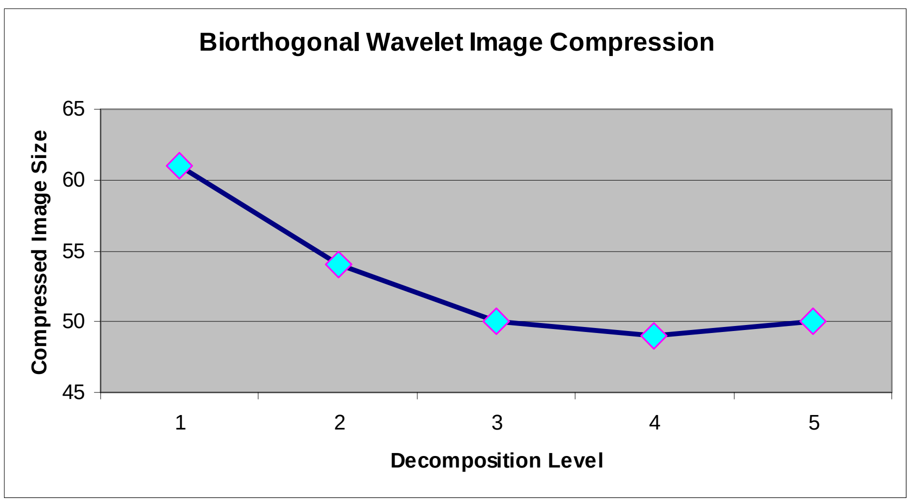


Figure – 4.26 Biorthogonal wavelet image compression chart

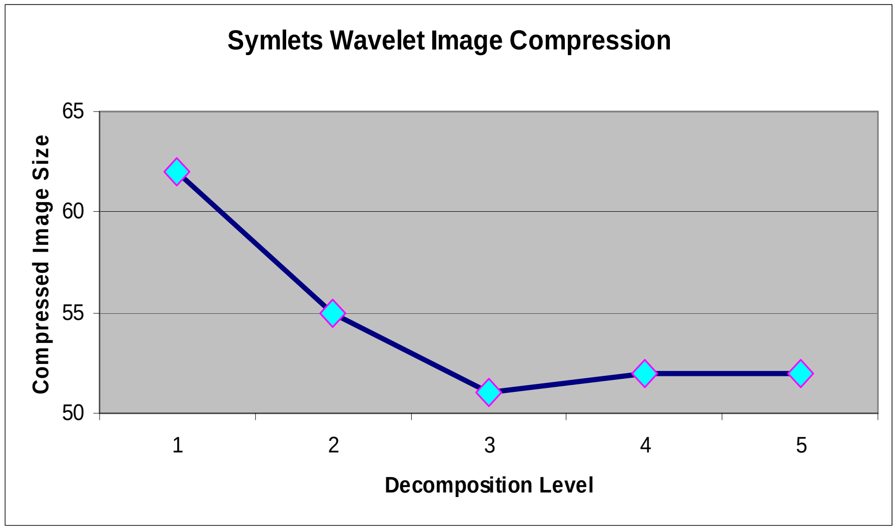


Figure – 4.27 Symlet wavelet image compression chart

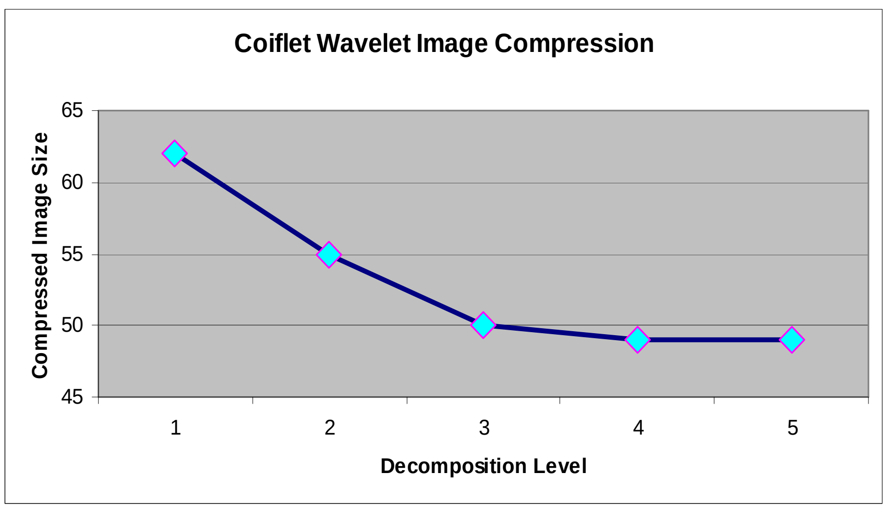


Figure – 4.28 Coiflet wavelet image compression chart

4.1.12 Conclusion

From the experimental study of cameraman image of different algorithms at different level, it can be concluded that when the decomposition level is increased, the compressed image size decreases which is shown in the above table. This experimental study concludes that the best compression is performed by Haar and Daubechies wavelet image compression algorithm. Symlets, Biorthogonal and Coiflet wavelet image compression algorithms are also compressed the image at good level.

4.2 JPEG COMPRESSION ALGORITHM

4.2.1 Introduction

The Joint Photographic Experts Group developed the JPEG algorithm in the late 1980's and early 1990's.

JPEG standard has been established by International Standards Organization (ISO) and International Electro-Technical Commission (IEC). The performance of these coders generally degrades at low bit rates mainly because of the underlying block-based Discrete Cosine Transform (DCT) scheme. The JPEG standard specifies the following three modes for lossy encoding and one mode for lossless encoding:

Lossy Encoding Modes:

1. Sequential
2. Progressive
3. Hierarchical

While discussing the nature of the compression algorithm, it is the best or excellent algorithm for image compression. It can compress full 24 bits color photographs or any gray scale photos which include many different shades of gray. This algorithm can not work properly with the web graphics, scanned text, line art, or any other images with sharp transition at the different edges of objects.

This algorithm can adjust the image compression ratio. That is the main feature of the image compression algorithm. As per user requirement the size and quality of the final image or compressed image can be determined. The image can be highly compressed with less quality or less compressed with high quality. Here the quality and size are the opposite of each other.

4.2.2 JPEG Compression

A different set of standards has to be created for compressing images of any size. Joint Photographic Expert Group is the first standard, which is known as JPEG and it is most widely used one. It is a very simple and easy to use standard that is based on the Discrete Cosine Transform (DCT). In JPEG compression there are main two processes [46]:

1. Encoder
2. Decoder

The figure given below shows the block diagram of JPEG Encoder and JPEG Decoder.

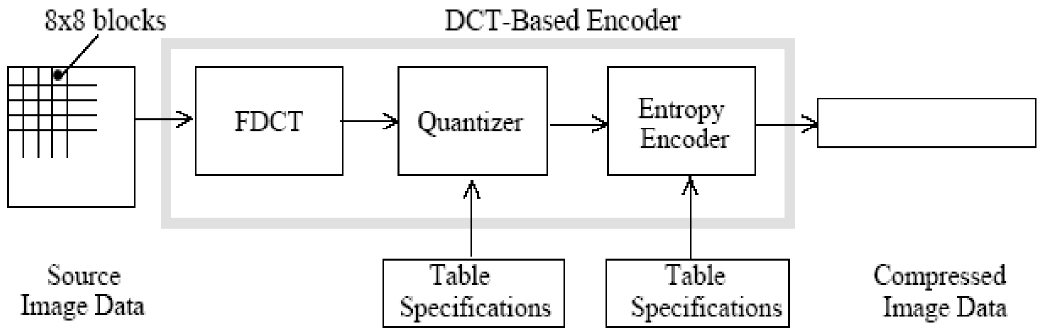


Figure – 4.29 DCT based Encoder process [46]

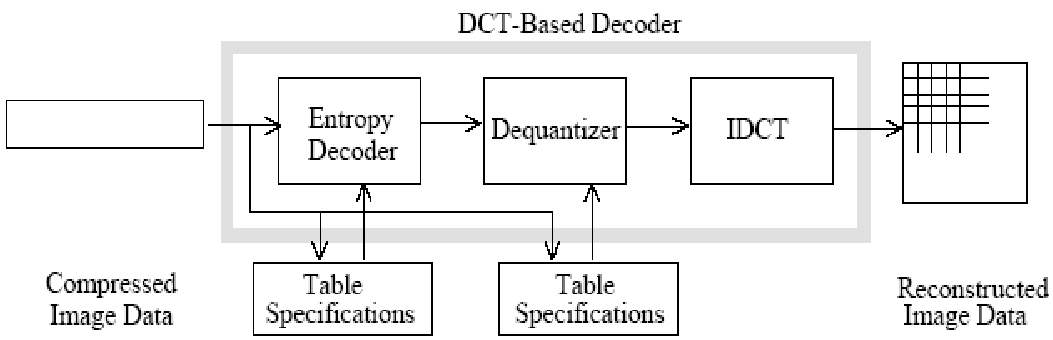


Figure – 4.30 DCT based Decoder process [46]

There are main three components, which as follows:

1. FDCT
2. Quantizer
3. Entropy Encoder

Forward Discrete Cosine Transform (FDCT): The images are first partitioned into non-overlapping blocks of size 8x8. The image samples are shifted from unsigned integers with range [0, 2p-1] to signed integers with range [-2p-1, 2p-1], where p is the number of bits, here, we use p=8. JPEG specifies neither any unique FDCT algorithm, nor any unique IDCT algorithms. The implementations may therefore differ in precision and JPEG has specified an accuracy test as a part of the compliance test [46].

Quantization: The 64 coefficients from the FDCT outputs are quantized according to the quantization table. The main objective to compress the image is that each step should be chosen as the perceptual threshold or just noticeable distortion without visible artifacts. To set the quantization table psycho visual experiments are performed. This appears in ISO-JPEG standard as a matter of information, but not as a requirement.

The quantized coefficients are zig-zag scanned. The DC coefficient is encoded as a difference from the DC coefficient of the previous block and 63 AC coefficients are encoded into pair [46].

Entropy Coder: This is the last and final processing step of the JPEG encoder. The JPEG standard specifies two entropy coding methods:

1. Huffman Coding
2. Arithmetic Coding

The baseline sequential JPEG uses Huffman coding only, but codecs with both methods are specified for the other modes of operation. Huffman coding requires that one or more sets of coding tables are specified by the application. The same table used for compression is used to decompress it. The baseline JPEG uses only two sets of Huffman tables – one for DC and the other for AC [46].

4.2.3 JPEG Standard

JPEG compression scheme has introduced and created an international standard for image compression in the Consultative Committee on International Telephone and Telegraph (CCITT) and International Standard Organization (ISO). It is a lossy compression scheme based on DCT algorithm.

JPEG lossless compression scheme are also proposed for some applications with the requirement for losslessness in the quality of the decompressed image. It is normally used for medical images. The JPEG lossless scheme is based on a completely different approach to the lossy scheme. A general block diagram of JPEG compression is shown in the figure given below :

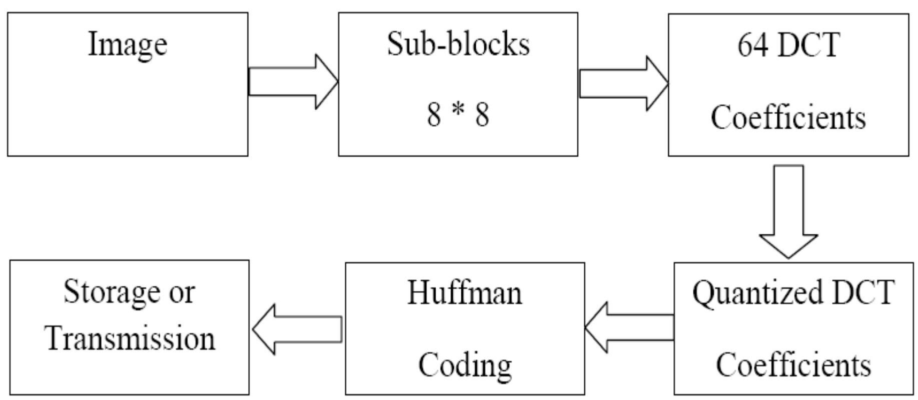


Figure – 4.31 JPEG standard block diagram [47]

The quantization procedure quantizes the DCT coefficients to the quantization steps at corresponding positions in the quantization table. For different block sizes, the quantization tables are different. This is the lossy compression method, so some information of the original image is lost and cannot be restored possibly affecting image quality [47].

4.2.4 JPEG Compression Process

JPEG is a lossy compression scheme for color and gray scale images. It works on full 24-bit color. It is designed to be used with photographic material and naturalistic artwork. It is not the ideal format for line-drawings, textual images, or other images with large areas of solid color or a very limited number of distinct colors. The figure given below shows the step in JPEG image compression process.

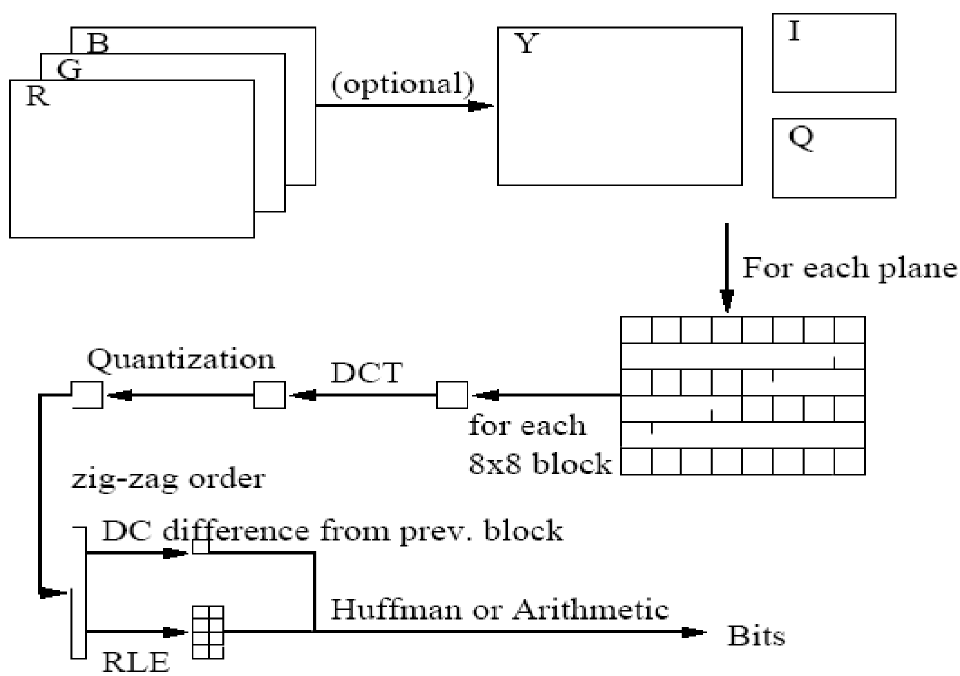


Figure – 4.32 JPEG compression steps

The input to JPEG is three color planes of 8-bits per-pixel each representing Red, Blue and Green (RGB). These colors are used by hardware to generate images. The following steps are performed in JPEG image compression:

1. Divide the image into 8x8 sub images.
2. Shift the gray-levels in the range (-128, 127).
3. Apply DCT on the partitioned image (64 coefficients will be obtained: 1 DC coefficient and 63 AC coefficients).
4. Quantize the coefficients and the less significant coefficients are set to zero.
5. Order the coefficients using zig-zag ordering and the coefficients obtained are in order of increasing frequency.

The first step of JPEG compression is optional. This step converts the RGB into YIQ color planes. The YIQ color planes are designed for better human perception. The Y plane is designed to represent the brightness of the image. It is a weighted average of red, blue and green (0.59 Green + 0.30 Red + 0.11 Blue). The weights are not balanced since the human eye is more responsive to green than to red, and more to red than to blue. The I (interphase) and Q (quadrature) components represent the color hue (chrominance).

The next step of the JPEG algorithm is to partition each of the color planes into 8x8 blocks. Each block is coded separately. The first step in coding a block is to apply a cosine transform across both dimensions. It returns an 8x8 block of 8-bit frequency terms. It does not introduce any loss, or compression. The block-size is motivated by wanting it to be large enough to capture some frequency components but not so large that it causes "frequency spilling".

After the cosine transform, the next step applied to the blocks is to use uniform scalar quantization on each of the frequency terms. This quantization is controllable based on user parameters. It is the main source of information loss in JPEG compression. The human eye is more perceptive to certain frequency components than to others. JPEG allows the quantization scaling factor to be different for each frequency component. The scaling factors are specified using an 8x8 table that is used simply to element-wise divide the 8x8 table of frequency components. JPEG defines standard quantization tables for both the Y and I-Q components. The table for Y is shown as:

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

Figure – 4.33 Default quantization table

In this table the largest components are in the lower-right corner. This is because these are the highest frequency components which humans are less sensitive to than the lower-frequency components in the upper-left corner. The selection of the particular numbers in the table seems magic, for example the table is not even symmetric, but it is based on studies of human perception.

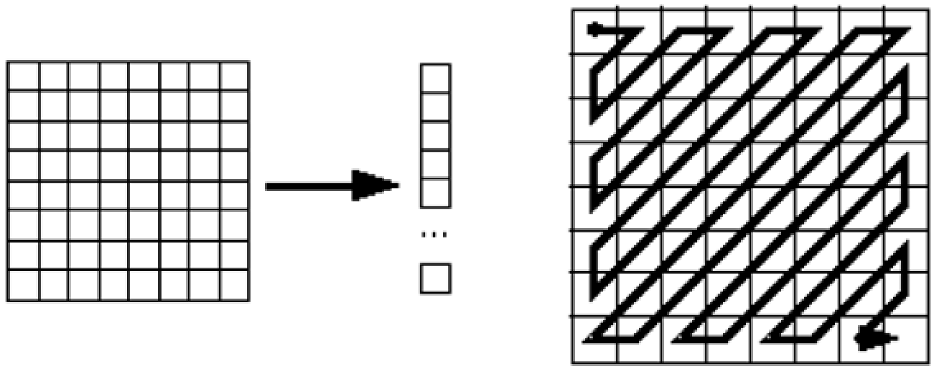


Figure – 4.34 Zig-Zag scanning of JPEG block

JPEG compression then compresses the DC component separately from the other components. Here DC component is on upper left most corners. It uses a difference coding by subtracting the value given by the DC component of the previous block from the DC component of this block. The DC component is often similar to block-to-block so that difference coding it will give better compression.

4.2.5 The JPEG Algorithm

JPEG compression and decompression process consist of four distinct and independent steps.

Step – 1: Divide the Image

In this step image is divided into 8 x 8 pixel blocks.

Step – 2: Conversion to the Frequency Domain

In this step Discrete Cosine Transform is applied to each block to convert the information from the spatial domain to the frequency domain.

Step – 3: Quantization

In this step the frequency information is quantized to remove unnecessary information.

Step – 4: Entropy Coding

In this step final bit stream is compressed by using standard compression techniques. The report will analyze the compression of a gray scale image, and will then extend the analysis to decompression and to color images.

Step – 1: Divide the Image

JPEG Image compression algorithms are used to compress the image. It will compress the entire image and it gives the optimal results. The first step of this algorithm is to divide the image; it divides the image into matrices of 8 X 8 pixel blocks. The division of image into blocks begins at the upper left corner of the image, and it moves towards the bottom right corner. If the image dimension is not 8 X 8, it means that is not the multiple of 8, then it adds the extra pixels at bottom right corner of the image and it is converted in form of multiple of 8. This dummy blocks are easily removed from the image during the decompression of the image.

It may bring some changes in color space. Mostly, one pixel is represented by 8 bit or 1 byte. Gray scale image have one per pixel and it must have the value in the range of 0 to 255. Here 0 represents the full black color and 255 represents the full white color. Color images have 3 bytes per pixel. One byte for red color, one byte for green color, and one byte for blue color i.e. RGB color.

The conversion of color space is possible. As per user requirement the color space conversions are performed. In JPEG algorithm, it converts the RGB color space into YCbCr color space because it compresses more than RGB. This color space conversion can be done in linear time. The color space conversion can be performed before the breaking into the blocks also.

In this algorithm, 128 are subtracted from each byte in the 64 byte block. It is used to change the scale of the byte values from 0…255 to -128…127. So the average value of the pixels will tend towards zero.

The following image shows the examples for division of image into blocks. The image is divided into 8 X 8 metrics of pixel. Here the size of image is double than the original size, so the blocks are difficult to see. This image is 200 X 220 pixels that means it will be separated into 700 blocks. Here some padding is added at the bottom of the image.

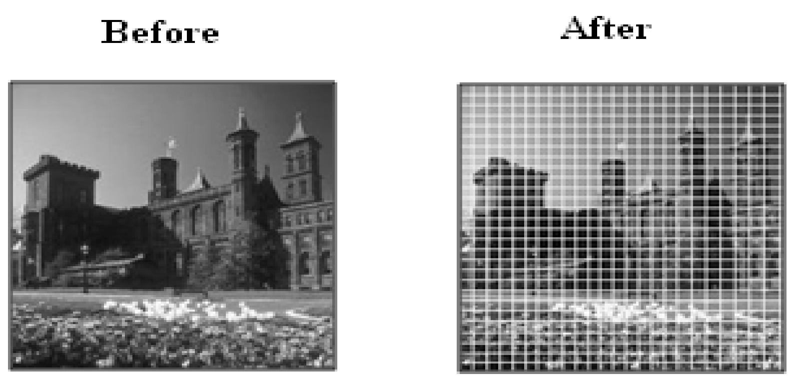


Figure – 4.35 Example of Image Division

Step – 2: Conversion to the Frequency Domain

The second step of the algorithm is performing the conversion. It converts the pixel information from the spatial domain to the frequency domain. Quantization is performed in the compression. The conversion is useful for making to the quantization process easier. By using the quantization, it can be known which parts of the image are more important and which parts of the image are less important. These less important parts play more important role in compression process.

Each value in the block represents the intensity of one pixel. After converting the block into the frequency domain, each value will be the amplitude of a unique cosine function. Each cosine functions have different frequencies. The block will represent by multiplying the functions with their corresponding amplitudes and then will add the result together. In JPEG algorithm, it puts the functions separately during the compression process so that it can remove the information, which has smallest contribution to represent the image.

The quantization process is used to remove the high frequency information, which results in a smaller representation of the image. Human are not able to see the higher frequencies, it has some limits. So image will give the different image that is represented in a computer. It looks nearest to the original as human, but not exactly same as original.

There are many algorithms available for conversing the spatial information into frequency domain. The popular algorithm is Fast Fourier Transform (FFT). The Discrete Cosine Transform (DCT) is faster than the FFT but the condition is that, there are no imaginary components available in image information. DCT is derived from the FFT. DCT requires some multiplications than the FFT, so it can perform only with real numbers. It gives some significant coefficients in its result, which is mostly related to greater compression. DCT is mainly designed to work with one dimensional data, but the image data are given in the blocks of two dimensions. Another summing term is added in the DCT to convert the equation into two dimensional. Here one dimensional DCT is applied in one direction say x and once in another direction say y, it will give effectively two dimensional discrete cosine transform.

A DCT, related to the Fourier Transform, can be used to discard higher frequency information that has little visual effect on the image. It is important to note that frequencies here refer to spatial frequencies rather than time frequencies. The DCT is central to JPEG compression. It performs no actual compression but it decides what data can be discarded, and what data needs to be retained. Starting with an 8 x 8 block of values, f(x, y), they can be transformed into a new set of values, F(u, v), by the Forward Discrete Cosine Transform. They can be converted back again to the original 64 values by the Inverse Discrete Cosine Transform. The equations used are [48]:

$$F(u,v) = {}^{1}/_{4}\, C(u)C(v)[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y) \mathrm{Cos} \frac{(2x+1)u\pi}{16} Cos \frac{(2y+1)v\pi}{16}$$

$$f(x,y) = {}^{1}/_{4}\, [\sum_{u=0}^{7}\sum_{v=0}^{7} C(u)C(v)F(u,v) \mathrm{Cos} \frac{(2x+1)u\pi}{16} Cos \frac{(2y+1)v\pi}{16}$$

Where

$$C(u), C(v) = {}^{1}/_{\sqrt{2}} \; for\, u, v = 0,$$

$$C(u), C(v) = 1 \; otherwise$$

The 2D DCT equation is given in above figure. Here it is also given that, where *C(u)* = 1/√2 if x is 0, and *C(u)* = 1 for all other cases. In the above equation, *f(x, y)* is the 8 bit image value at *(x, y)* coordinates and *F(u, v)* is the new entry in the frequency matrix [48].

The above equation shows that only constants terms are represented before the brackets. Only 16 different terms will be required for each different pair of (u, v) values. We may compute this ahead of time. After that it multiplies the correct pair of cosine terms to the spatial domain value for that pixel. There will be 64 additions in the two summations, one per pixel.

At last, it multiplies the sum by the three constants to get the final value in the frequency matrix. This will continue for all (u, v) pairs in the frequency matrix. u and v may be any value from 0...7. The frequency domain matrix is just as large as the spatial domain matrix.

The frequency domain matrix contains the values from 1024 to 1023. The upper left entry of the matrix is known as the DC value. It is the average of the entire block and it is the lowest frequency cosine coefficient. To move right the coefficients represent cosine functions in the vertical direction that increase in frequency. To move down, the coefficients belong to increasing frequency cosine functions in the horizontal direction. The highest frequency values occur at the lower right part of the matrix. The higher frequency values also have a natural tendency to be significantly smaller than the low frequency coefficients since they contribute much less to the image.

The lower right half portion of the matrix is factored out after the quantization process. This will remove half of the data per block. That is the reason that JPEG is so efficient at compression.

The most time consuming part of the JPEG image compression algorithm is to calculate the DCT. There are many different implementations of DCT which are possible. This implementation can replace all the multiplications with the shift instructions and additions. 1D and 2D Discrete Cosine Transformation (DCT) equations are as follows:

1-D DISCRETE COSINE TRANSFORMS [48]

DCT Transform:

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

Inverse Transform:

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

Scaling Factor:

$$\alpha(u) = \begin{cases} \sqrt{1/_N} \; for\, u = 0 \\ \sqrt{2/_N} \; for\, u = 1, 2, \dots, N-1 \end{cases}$$

2-D DISCRETE COSINE TRANSFORMS [48]

DCT Transform:

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

Inverse Transform:

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

Step – 3: Quantization

Quantization is achieved by compressing a range of values to a single quantum value. When the number of discrete symbols in a given stream is reduced, the stream becomes more compressible. A quantization matrix is used in combination with a DCT coefficient matrix to carry out transformation. Quantization is the process where most of the compression takes place. DCT does not compress the image because it is almost lossless. Quantization makes use of the fact that higher frequency components are less important than low frequency components. It allows varying levels of image compression and quality through selection of specific quantization matrices. The range for quality levels is 1 to 100. Here 1 gives the poorest image quality and highest compression, while 100 gives the best quality and lowest compression.

The compressed image quality and compression ratio can be selected to different needs. JPEG committee suggests matrix with quality level 50 as standard matrix. For obtaining quantization matrices with other quality levels, scalar multiplications of standard quantization matrix are used.

Quantization is achieved by dividing transformed image matrix by the quantization matrix used. Values of the resultant matrix are then rounded off. In the resultant matrix coefficients situated near the upper left corner have lower frequencies. Human eye is more sensitive to lower frequencies. Higher frequencies are discarded. Lower frequencies are used to reconstruct the image.

	0	1	2	3	4	5	6	7
0:	0	1	5	6	14	15	27	28
1:	2	4	7	13	16	26	29	42
2:	3	8	12	17	25	30	41	43
3:	9	11	18	24	31	40	44	53
4:	10	19	23	32	39	45	52	5
5:	20	22	33	38	46	51	55	60

This places the elements of the coefficient block in a reasonable order of increasing frequency. Since the higher frequencies are more likely to be zero after quantization, this tends to group zero values in the high end of the vector [26].

The coefficients of a block transformed by DCT are already quantized to a small degree, as they all have been rounded off to an integer of a set bit length. Study of the physiology and sensitivity of the human eye has revealed two important results for compression with DCT.

When a block of pixels from a continuous tone image is transformed, many of the DCT coefficients have values close to zero. The human visual system is quite insensitive to errors in these coefficients; values close to zero may be set to zero without much effect on visual quality. The figure given below shows the quantization equation that is used for each block in the image [48].

$$\boldsymbol{Q(u,v) - \mathbf{floor}\left[\frac{F(u,v)}{q(u,v)} + 0.5\right]}$$

Where *Q(u,v)* is the quantized value

F(u,v) is the DCT coefficient

q(u,v) is the corresponding quantization value [48].

Step – 4 : Entropy Coding

This is the last and final step of image compression. After completion of the quantization, the algorithm is left with blocks of 64 values and many of which are zero. To compress the data it would collect the entire zero and put value together, which is exactly done by the JPEG image compression algorithm. This algorithm uses a zig-zag order for encoding. It collects the high frequency quantized values into long strings of zeros.

To perform a zig-zag encoding on a block, the algorithm starts at the DC value and begins winding its way down the matrix, which is represent in the figure given below. It converts the 8 X 8 table into a 1 X 64 vector. The figure given below shows the zig-zag order encoding and sequence obtained by zig-zag.

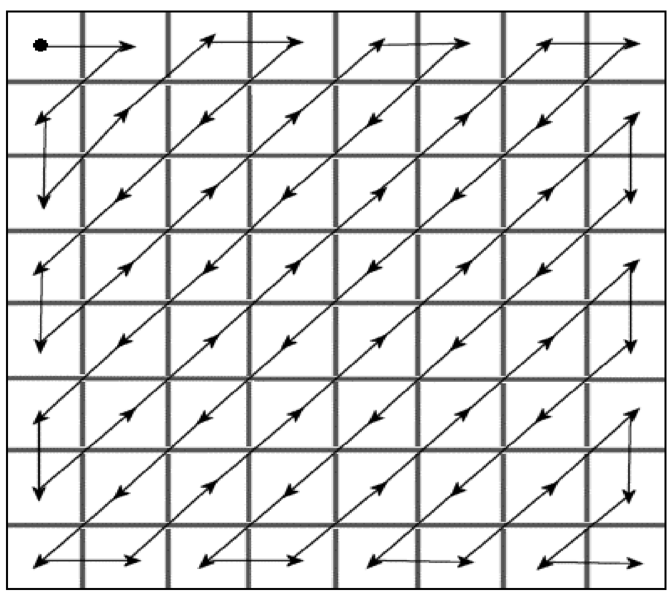


Figure – 4.36 Zig-Zag ordered encoding

The zig-zag order encode all the values from each blocks except one value i.e. DC value. All these values have two tokens which are used to represent the values in final file except DC value. Token is the combination of size and skip value i.e. {size, skip}. Here size value represents the number of bits required to representation of second token and skip value represents the number of zero that precede the token. The second token represents simply the quantized frequency value. The end-of-block sentinel is required to add the end of each block. It is used to inform the decoder where one block ends and the next block begins.

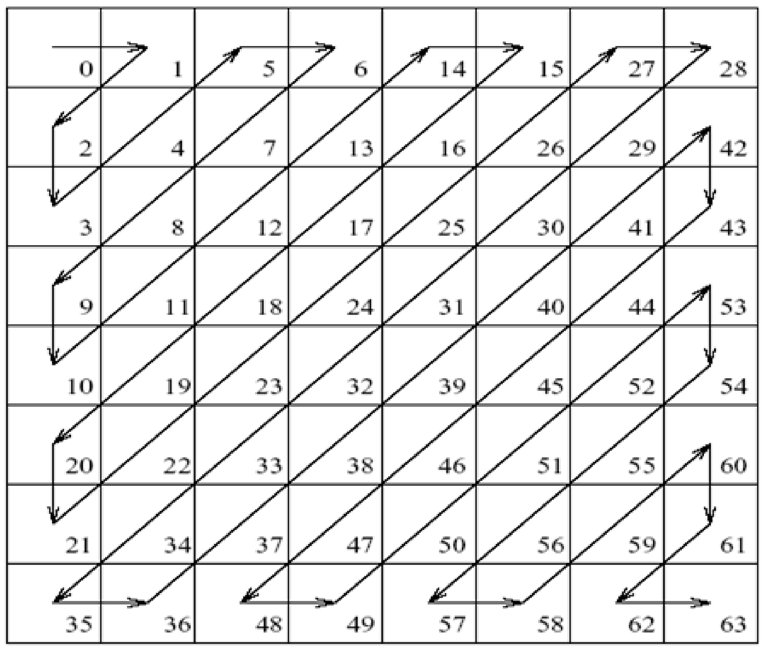


Figure – 4.37 Sequence obtained by zig-zag

Huffman Coding is used to encode the token information. It scans the data written in token. It assigns some bits to frequently occurring data and more bits for infrequently occurring data. JPEG algorithm is allowed to use the standard Huffman table. It also allows the custom tables by providing a field in the file that will hold the Huffman table.

Delta encoding is used for DC value. In delta encoding, each DC value is compare with previous value in zig-zag order. The comparison is done on the basis of block by block. That is the only process that blocks are treated independently from each other.

Zig – Zagging

It has been shown that DCT concentrates energy into the top-left coefficients. The quantization matrices emphasize this trend. The quantized block is then rearranged in a zig-zag order. These usually increase the run length of zeroes. The figure given below show the zig-zagging [48].

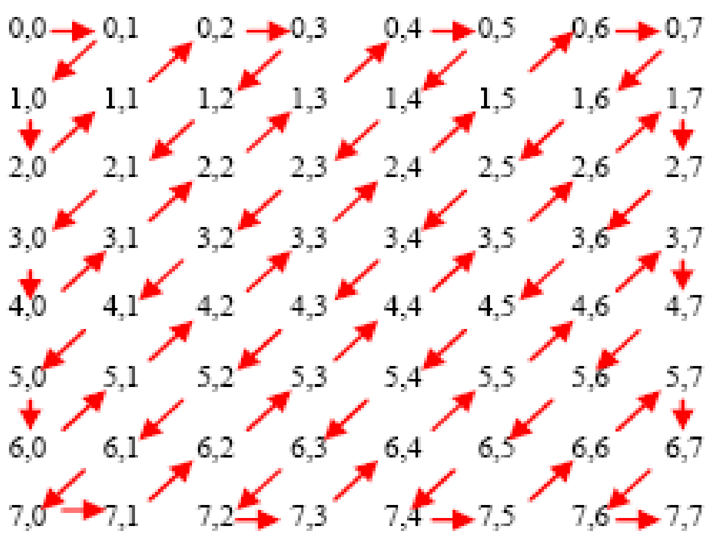


Figure – 4.38 Zig-Zagging [48]

Zero Run Length

The output from the zig-zag scanning process is then zero run length. It counts the number of zeroes in the zero run length blocks prior to any non-zero value. It stores the number of zeroes as an integer with fixed bit length, followed by the non-zero value. For example [48]:

23 1 2 1 0 0 0 -3 0 0 0 0 0 0 0 0 11 0 0 0 0 …… (rest zeroes).

This is compressed to: (23) (0 1) (0 2) (0 1) (3 -3) (8 11) (0 0) the (23) is the DC value, and is stored separately from the AC coefficients. The (0 0) indicates that the remainder of the block consists of zeroes. This is known as an End Of Block (EOB) indicator. If there are sixteen zeros compressed using zero run length, they are indicated by placing 15 in the zero run length value, and 0 in the non-zero value (15 0). This is known as the Zero Run Length (ZRL) indicator [48].

After zig-zagging, a 64-element vector is obtained. If the last element in this vector is not reduced to zero by the quantization, the zero-run-length vector does not take an EOB indicator. The decompressing algorithm should be able to recognize that it has reached the 64-element limit of the block it is decompressing. It can be seen that this reduces the number of bytes required to store the block of values. Multiplexing the zero run length value and the non-zero value into a single byte can reduce this further [48].

Huffman Encoding

The JPEG standard supports several forms of Huffman coding to further reduce the number of bits required. The basic idea is to use variable length codes to represent image data, using shorter codes to represent data with a higher probability of occurrence, and shorter codes for data with a lower probability. The result is that a large percentage of the coded data consists of very short codes, thereby reducing the overall amount of data for storage or transmission [48].

There are a few distinctive characteristics of codes generated by Huffman coding [48]:

1. No two code words may consist of identical sequences of code bits.
2. No information other than the code itself is required to indicate the beginning and end of any code value.
3. For a given number of symbols arranged in descending order of probability, the length of the associated code values will be such that the code associated with any given symbol will be less than or equal the length of the code associated with the next less probable symbol.
4. No more than two code words of the same length are alike in all but their final bits.

4.2.6 Experimental Study

JPEG image compression algorithm experiment is performed on gray scale image Boat and Barbara and color image Mandrill and Peppers image. In our experiment, we performed this study on different image quality level such as 20%(Low Quality), 40%(Good Quality), 60%(Better Quality) and 80%(Great Quality). The following images represent the original images and different image quality level with original image and compressed image.

Figure – 4.39 Original Boat and Barbara images

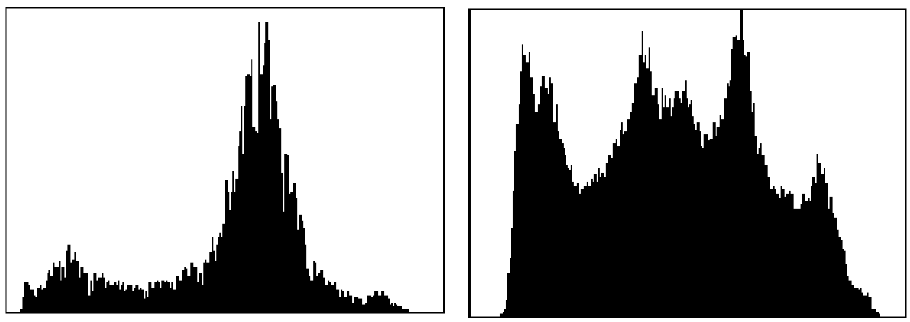


Figure – 4.40 Histogram of Boat and Barbara images

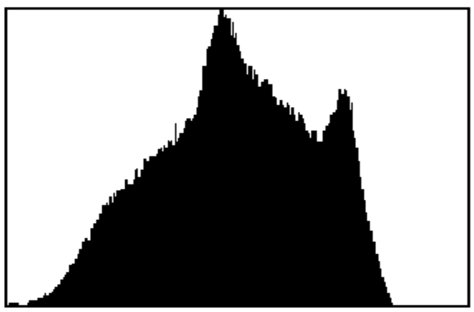
Boat		Barbara	
Mean	130.7	Mean	111.3
Median	144	Median	111
Mode	151.4	Mode	155.3
Standard Deviation	46.69	Standard Deviation	57.43
Median Absolute Deviation	18	Median Absolute Deviation	45
Mean Absolute Deviation	35.83	Mean Absolute Deviation	48.61

Table – 4.4 Statistics of Boat and Barbara images

Quality	Boat	Barbara
20% **Low Quality**		
40% **Good Quality**		
60% **Better Quality**		
80% **Great Quality**		

Figure – 4.41 Image quality level and compressed images of Boat and Barbara

Figure – 4.42 Original Mandrill and Peppers images



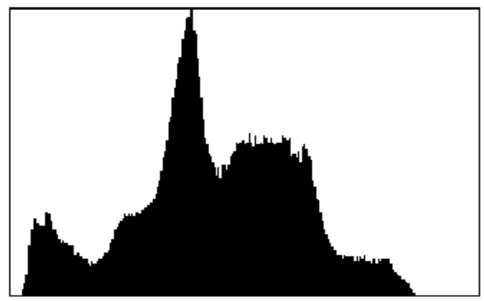


Figure – 4.43 Histogram of Mandrill and Peppers images

Mandrill		Peppers	
Mean	129.6	Mean	119.4
Median	130	Median	120
Mode	122.9	Mode	94.07
Standard Deviation	42.33	Standard Deviation	54.46
Median Absolute Deviation	33	Median Absolute Deviation	42
Mean Absolute Deviation	34.94	Mean Absolute Deviation	46.65

Table – 4.5 Statistics of Mandrill and Peppers images

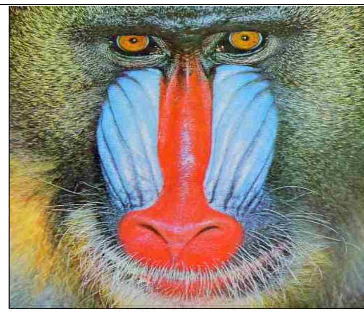
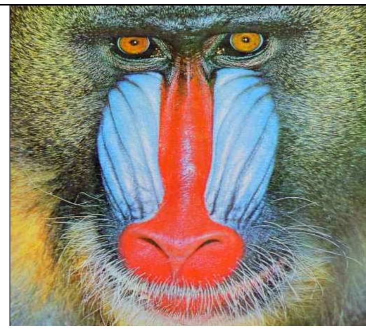
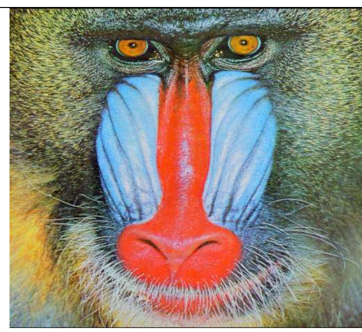
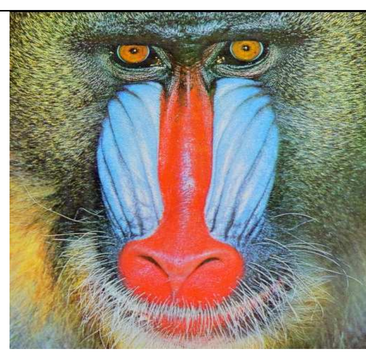
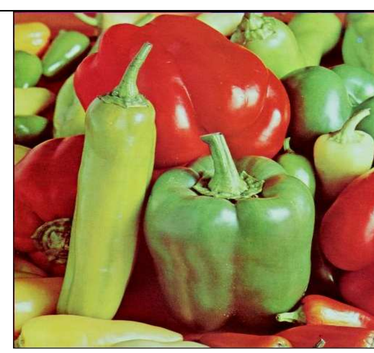
Quality	Mandrill	Peppers
20% **Low Quality**		
40% **Good Quality**		
60% **Better Quality**		
80% **Great Quality**		

Figure – 4.44 Image quality level and compressed images Mandrill and Peppers

The table given below represents the different images at different image quality level with original image size and compressed image size.

Image	Original Size (in KB)	Image Quality Level			
		Low	Good	Better	Best
		20%	40%	60%	80%
Boat	173.6	16.1	24.3	31.8	49.5
Barbara	181.4	18.5	27.6	35.3	51
Mandrill	622.3	26.7	42.6	57	86.7
Peppers	526.1	14.3	21.9	29.8	47.1

Table – 4.6 Image quality level

Image	Original Size (in KB)	Image Compression Time(in Sec)			
		Low	Good	Better	Best
		20%	40%	60%	80%
Boat	173.6	5.2	7.9	10.3	16
Barbara	181.4	6	9	11.5	16.6
Mandrill	622.3	8.7	13.8	18.5	28.1
Peppers	526.1	4.7	7.1	9.7	15.3

Table – 4.7 Image compression time

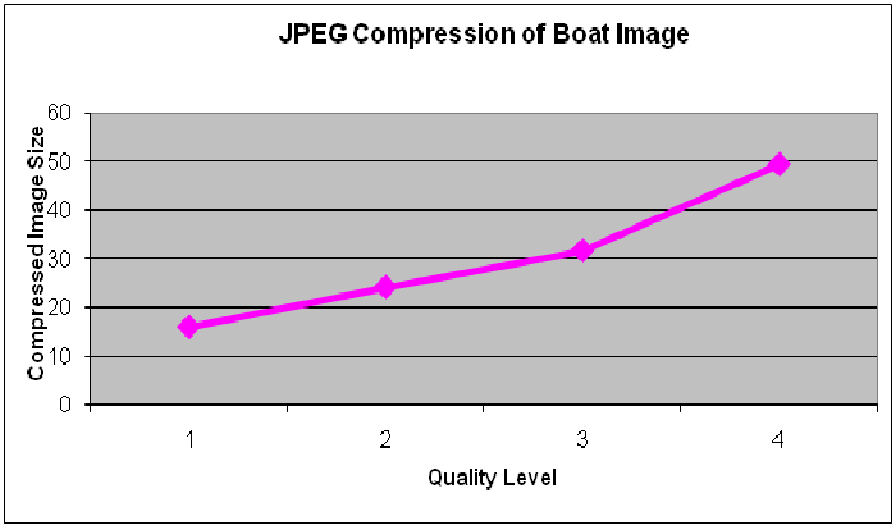


Figure – 4.45 Quality Level v/s Compressed Image Size of Boat image chart

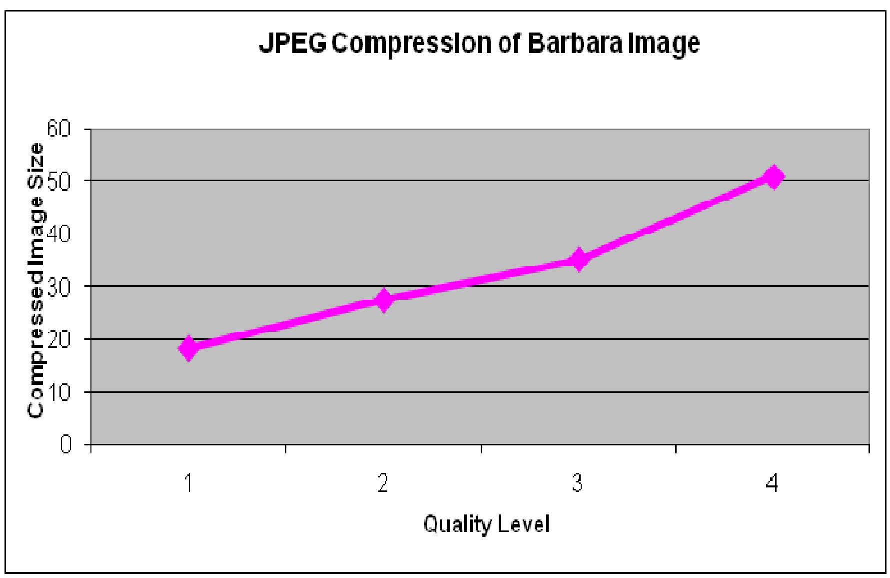


Figure – 4.46 Quality Level v/s Compressed Image Size of Barbara image chart

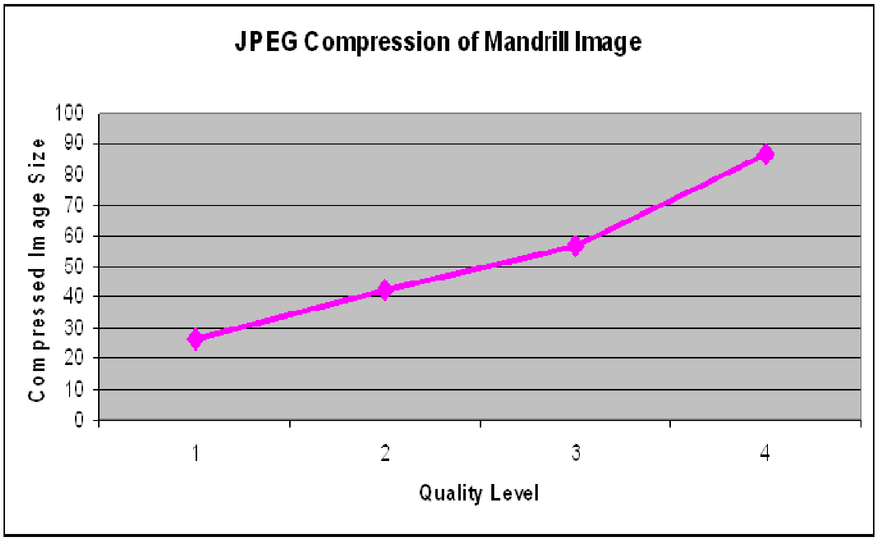


Figure – 4.47 Quality Level v/s Compressed Image Size of Mandrill image chart

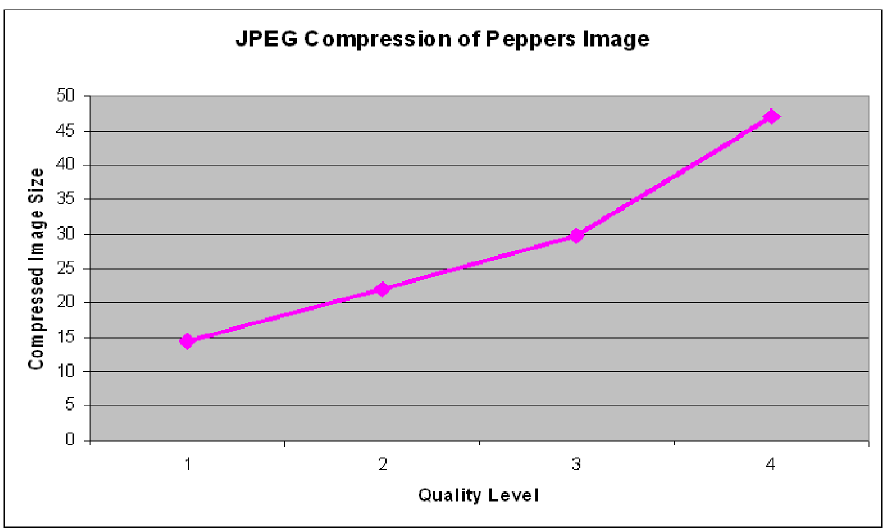


Figure – 4.48 Quality Level v/s Compressed Image Size of Peppers image chart

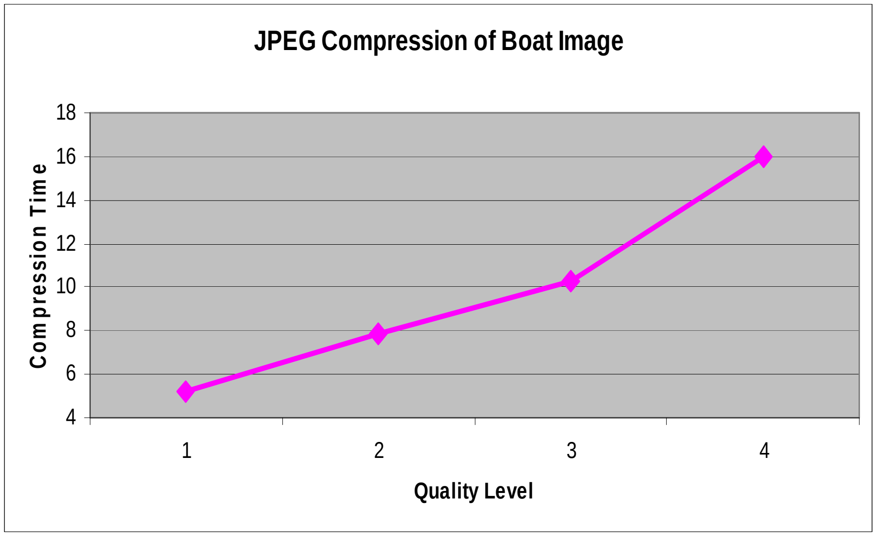


Figure – 4.49 Quality Level v/s Compression Time of Boat image chart

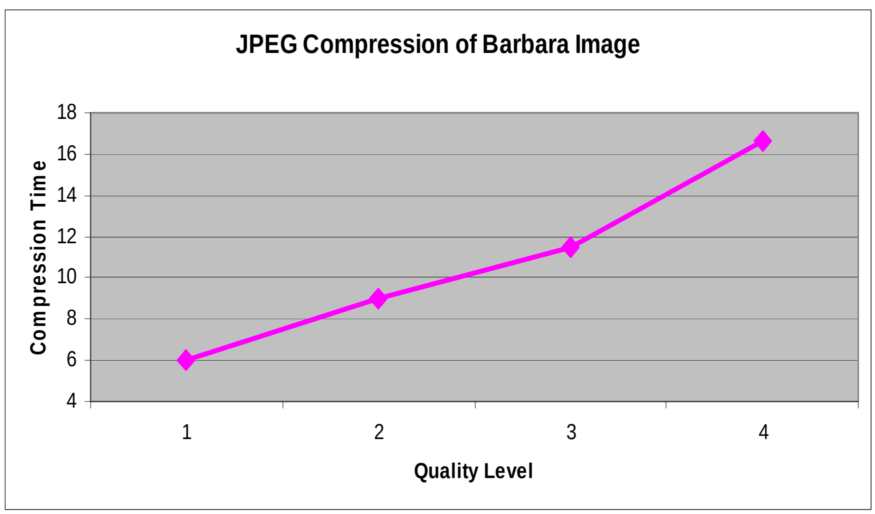


Figure – 4.50 Quality Level v/s Compression Time of Barbara image chart

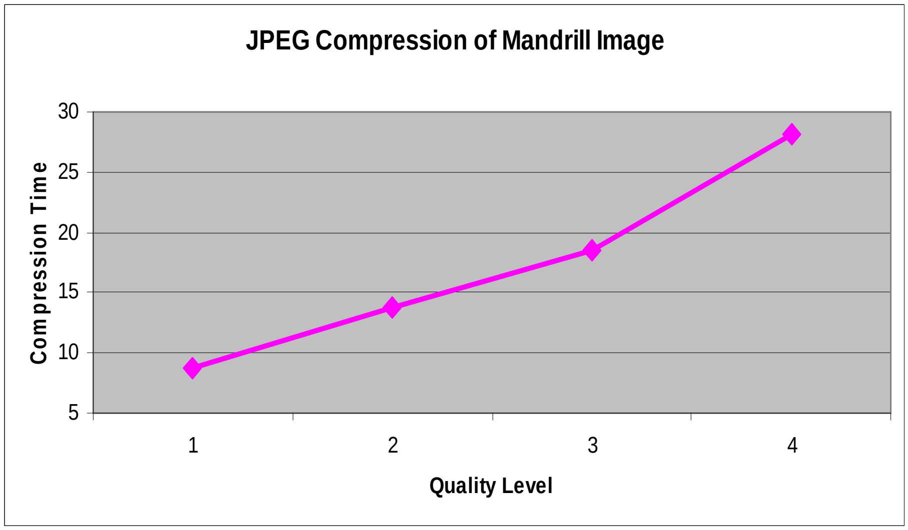


Figure – 4.51 Quality Level v/s Compression Time of Mandrill image chart

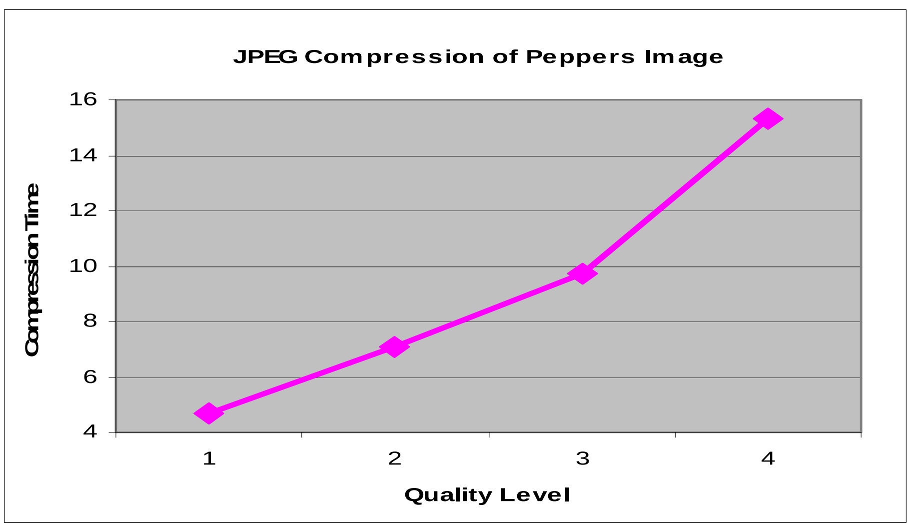


Figure – 4.52 Quality Level v/s Compression Time of Peppers image chart

In the above given chart quality level-1 represent 20% quality; level-2 represent 40% quality; level-3 represent 60% quality and level-4 represent 80% quality level.

4.2.7 Conclusion

From the experimental study of different images at different image quality level, it can be concluded that when you increase the image quality level, the compressed image size is also increase which is shown in above table. When you increased the image quality level, the image compression time required also increases which is shown in the above given table.

4.3 VECTOR QUANTIZATION COMPRESSION ALGORITHM

4.3.1 Introduction to Vector Quantization

The technique of obtaining the compact representation of an image while maintaining all the necessary information without much data loss is referred to as Image Compression.

Image Compression can be classified into two types: Lossless and Lossy compression.

Lossy compression can be broadly classified into two types:

- Scalar Quantization (SQ)
- Vector Quantization (VQ)

A popular technique for source coding of image and speech data since 1980 is VQ. Day by day the use of multimedia, images and the other picture formats are rapidly increasing in a variety of application. It is very straight forward image compression approach.

VQ has the particular advantage of being able to exploit prior knowledge on the images to be compressed. In this method, codebook has to be generated before compression. It is not a universal approach since it cannot work well for different types of images.

Vector Quantization has been observed as an efficient technique for image compression. There are mainly two components of VQ compression system:

- VQ Encoder
- VQ Decoder

The figure given below figure shows the VQ Encoder and Decoder:

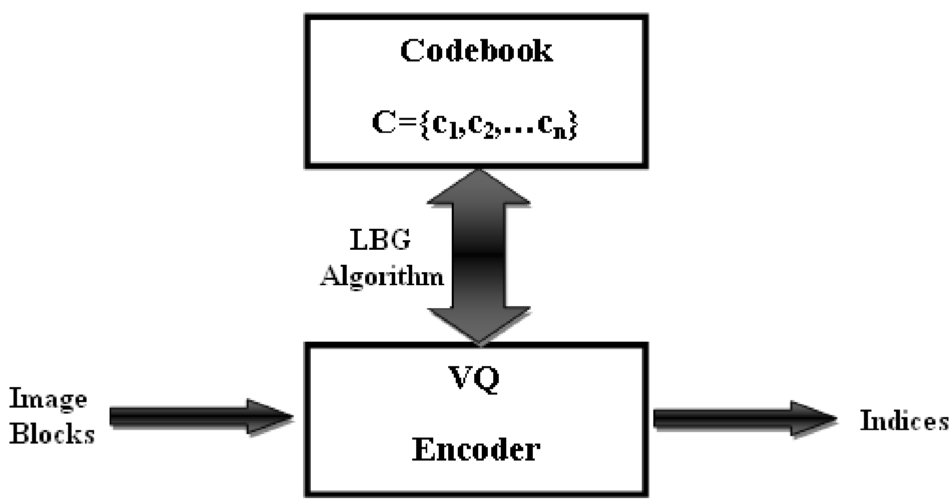


Figure – 4.53 VQ Encoder [49]

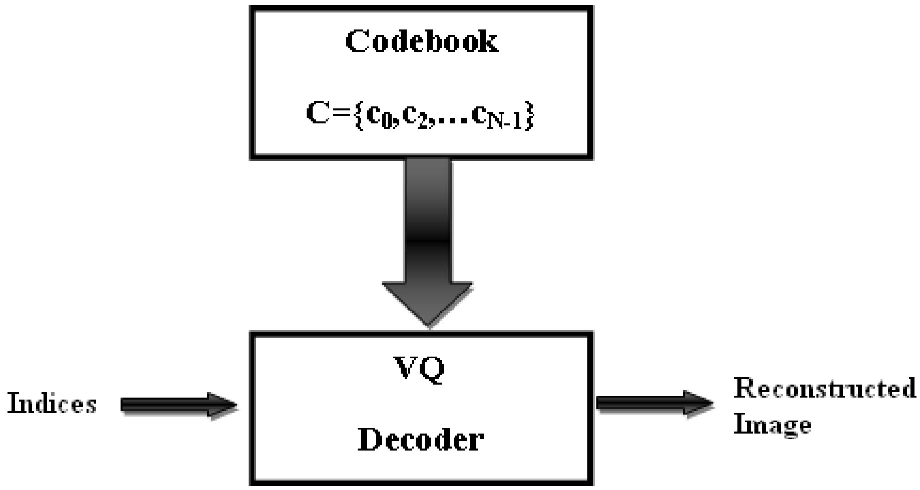


Figure – 4.54 VQ Decoder [49]

4.3.1.1 What is VQ?

A vector quantizer maps *k*-dimensional vectors in the vector space R^k into a finite set of vectors $Y=\{y_i : i=1,2, \ldots,N\}$. Each vector y_i is known as a Code Vector or a Code word. The set of all the code words is known as a Codebook. Associated with each code word, y_i, is the nearest neighbor region called Voronoi Region. It is defined as follows:

$$V_i = \{x \in R^k : \|x - y_i\| \leq \|x - y_j\|, for\ all\ j \neq i\}$$

The set of Voronoi regions partition the entire space R^k such that:

$$\bigcup_{i=1}^{N} V_i = R^k \ and \ \bigcap_{i=1}^{N} V_i = \emptyset \ for\ all\ i \neq j$$

As an example we take vectors in the two dimensional case without loss of generality. The figure given below shows some vectors in space.

In the figure given below input vectors are marked with an *X*, code words are marked with red circles, and the Voronoi regions are marked with yellow color and separated with boundary lines.

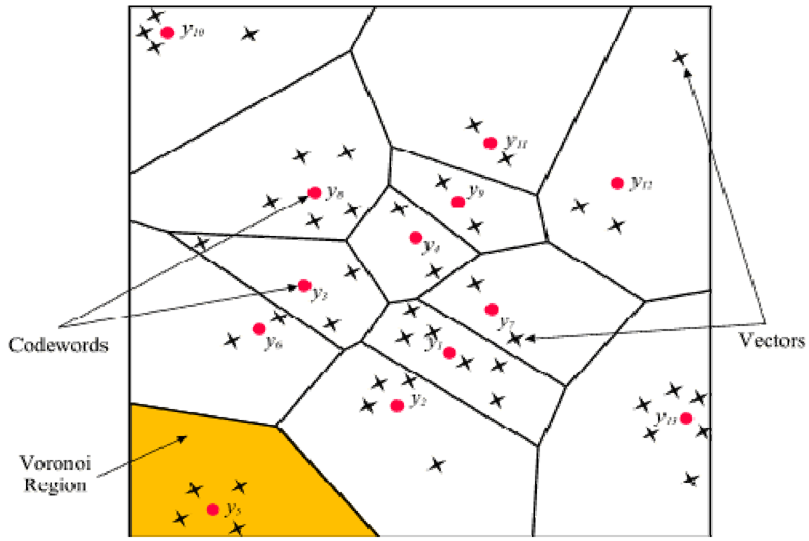


Figure – 4.55 Code words in 2D space

Associated with each cluster of vectors is a representative code word. Each code word resides in its own Voronoi region. These regions are separated with imaginary lines which are represented in above figure. Given an input vector, the code word that is chosen to represent is the one in the same Voronoi region.

The representative code word is determined to be the closest in Euclidean distance from the input vector. The Euclidean distance is defined as follows:

$$d(x, y_i) = \sqrt{\sum_{j=1}^{k} (x_j - y_{ij})^2}$$

Where x_j is the j^{th} component of the input vector, and y_{ij} is the j^{th} is component of the code word y_i.

4.3.1.2 Applications of VQ

Vector Quantization is used in many applications. Following are the name of some applications where VQ is used:

- Image Compression
- Voice Compression
- Voice Recognition
- Volume Rendering etc…

4.3.1.3 Work process of VQ

A vector quantizer is composed of two operations: Encoder and Decoder. The encoder takes an input vector and outputs the index of the code word that offers the lowest distortion. In this case the lowest distortion is found by evaluating the Euclidean distance between the input vector and each code word in the codebook. Once the closest code word is found, the index of that code word is sent through a channel. The channel may be computer storage, communications channel, and so on. When the encoder receives the index of the code word, it replaces the index with the associated code word. The figure given below shows a block diagram of the operation of the encoder and decoder.

Given an input vector, the closest code word is found and the index of the code word is sent through the channel. The decoder receives the index of the code word, and outputs the code word.

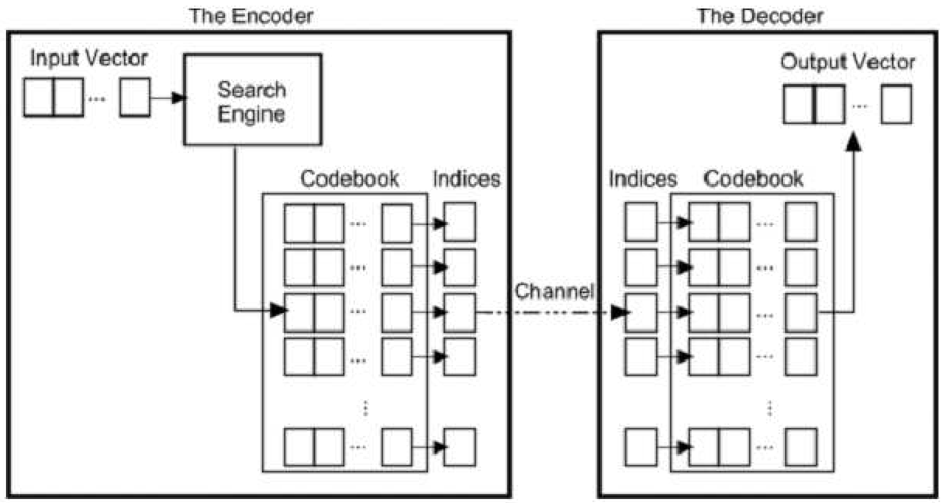


Figure – 4.56 The Encoder and Decoder in VQ

4.3.1.4 Performance measure in VQ

There is no proper way to measure the performance of VQ. This is because the distortion. VQ is evaluated by us humans and that is a subjective measure. It uses Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR). MSE is defined as follows:

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(\hat{x}_i - x_i)^2$$

Where *M* is the number of elements in the signal or image.

For example, if we want to find the MSE between the reconstructed and the original image, then we would take the difference between the two images pixel by pixel, square the results, and average the result, which is represented in above formula.

The PSNR is defined as follows:

$$PSNR = 10log_{10}\left(\frac{(2^n - 1)^2}{MSE}\right)$$

Where *n* is the number of bits per symbol.

For example, if we want to find the PSNR between two 256 gray level images, then we set *n* to 8 bits.

4.3.2 Vector Quantization

Image compression using Vector Quantization (VQ) is a lossy compression technique. It is defined as mapping Q of K-dimensional Euclidean space R^k into a finite subset Y of R^k Thus.

$$Q :\ R^k \rightarrow Y$$

Where y = (xi; i=1, 2..., N) is the set of reproduction vectors and N is the number of vectors in Y. The figure given below shows the conceptual diagram of Vector Quantization [50].

A vector quantizer is composed of two parts, encoder and decoder. An encoder will compare each input vector with every code vector in the codebook and generate index which represent the minimum distortion code vector from the input vector. A decoder takes the indexes to locate the code vector in codebook and generate the output vectors [50].

A codebook is the set of finite code vector for representing the input vector. The popular technique in codebook design is the Linde-Buzo-Gray (LBG) algorithm. The whole image partitioned into sub blocks and all sub blocks are used training this codebook [50].

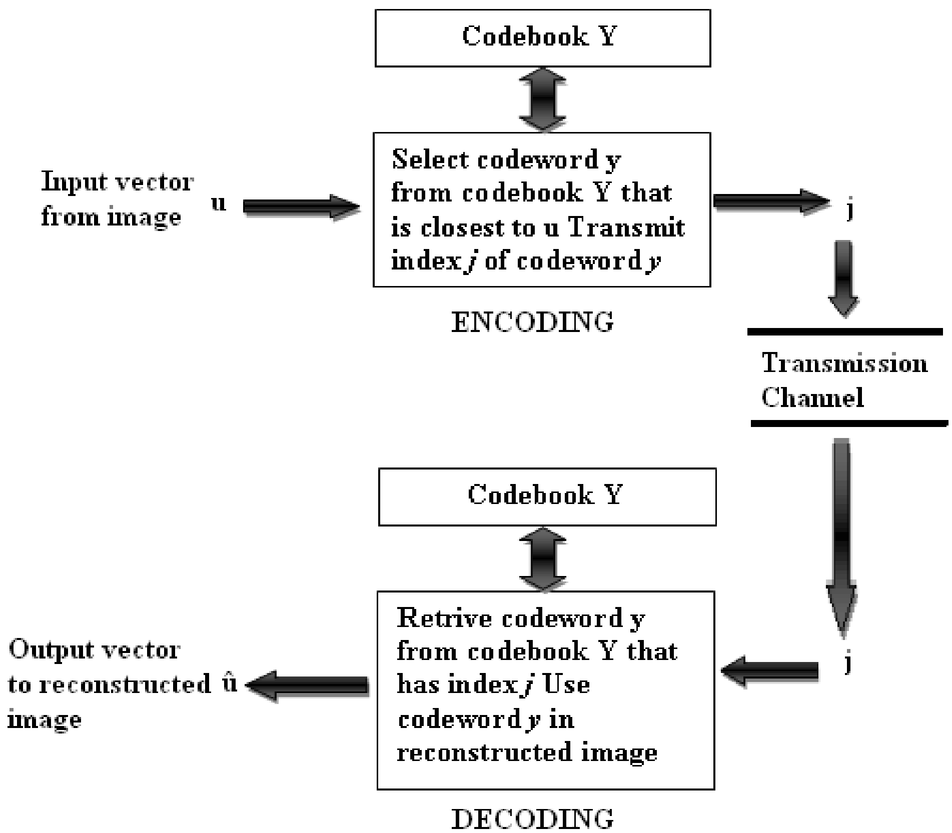


Figure – 4.57 Conceptual diagram of Vector Quantization [50]

The VQ system consists of an encoder, a decoder and a transmission channel which is represented in the above given figure. The encoder and the decoder each have access to a codebook, Y. The codebook Y is a set of Y code words or code vectors, y, where each y is dimension n^2 and has a unique index, j, and 0≤j≤Y-1[50].

In this method, the given image is partitioned into a set of non-overlapping image blocks $X = \{x_0, x_1 \ldots., x_{m-1}\}$ of size 4 X 4 pixels each. For example Linde-Buzo-Gray (LBG), is used to generate a codebook $C = \{Y_0, Y_1, \ldots.., Y_{N-1}\}$ the given set of image blocks. The codebook C consists of a set of representative image blocks called code words. The VQ encoder finds a closest match code word in the codebook for each image block and the index of the code word is transmitted to VQ decoder [50].

The image is broken into blocks of pixels called tiles. Each image tile of *n* X *n* pixels can be considered a vector, u, of dimension n^2. For each image tile, the encoder selects the code word y that yields the lowest distortion by some distortion measure *d(u,y)*. The index *j* of that code word is sent through the transmission channel. If the channel is errorless, the decoder retrieves the code word y associated with index *j* and outputs y as the reconstructed image tile u[50].

Mathematically, VQ encoding is a mapping from a *k*-dimensional vector space to finite set of symbols, J,

$$VQ : u = (u_1, u_2, \ldots\ldots , u_k) \rightarrow j$$

Where $k=n^2$, $j \in J$, and J has size J=Y. The rate, R, of the quantization is:

$$R = log_2 Y$$

Where *R* is bit per input vector. The compression rate is R/n^2 bits per pixel. Typically, *Y* is chosen to be a power of *2*, so *R* is integer. Consequently, VQ encoding generates codes of R bits in length with every *R*-bit code corresponding to some $y \in Y$. Two common distortion measures are squared error [50].

$$d_{sq}(u, y) = \sum_{i=1}^{k} (u_i - y_i)^2$$

and absolute error

$$d_{abs}(u, y) = \sum_{i=1}^{k} |u_i - y_i|$$

The performance of a vector quantization system depends on the composition of the codebook. Several criterions may be used to design an optimal codebook. It may be desired that the average distortion (Mean Squared Error) due to VQ is minimized. Another criterion is to maximize the entropy of the codebook; that is, to ensure that each of the code words is used equally frequently on the average. Once the criterions are decided upon, the optimal codebook must be determined [50].

In the decoding phase, VQ decoder replaces the index values with the respective code words from the codebook and produces the quantized image, called as reconstructed image [50].

The other image blocks are encoded utilizing the correlation with the neighboring encoded image blocks. Let *x* be the input image block, and *u* and *l* be the upper and left neighboring code words respectively. Let the size of the given image block size be *k* = *m x n*. The side match distortion of a code word *Y* can be defined as:

$$smd(Y) = \sum_{i=0}^{n-1} (u_{(m-1,i)} - Y_{(0,i)})^2 + \sum_{i=0}^{m-1} \left(l_{(i,n-1)} - Y_{(i,0)}\right)^2$$

Side Match Vector Quantization (SMVQ) sorts the code words according to their side match distortions of all code words and then selects N_S code words with smallest side match distortions from the master book C of size N to form the state codebook SC, where $N_S < N$ [50].

A best match code word Y_i is selected to encode an image block x from N_S code words and the corresponding index is coded in $log_2 N_S$ bits. Thus, the SMVQ reduces the bit rate of VQ. Since mean square error caused by state codebook is higher than that of master codebook, SMVQ degrades the image quality and also it requires long encoding time [50].

Classified Side Match Vector Quantization (CSMVQ) is an efficient low bit rate image compression technique which produces relatively high quality image. It is a variable rate SMVQ and makes use of variable sized state codebooks to encode the current image block. The size of the state codebook is decided based on the variances of left code words and upper code words that predict the block activity of the input blocks [50].

4.3.2.1 Algorithm

In image compression using vector quantization, an input image is divided into small blocks called training vectors ($x_j(k)$). This training vectors can be closely reconstructed from applying a transfer function (Q) to a specific region of an input image itself, which is called codebook ($x_i(k)$). Thus, only the set of transfer functions, which have fewer data than an image, were required for reconstruct the input image back. A transfer function (Q) is defined as follows:

$$Q : R^k \rightarrow Y$$

That Y={x_i; $i=1,2,3,....,N$} is the set of vectors $x_j(k)$ that recreate and N is the number of vectors in Y. VQ consists of 2 parts. First encoder will compare the training vector and codebook vectors. The result is an index (j) that represents the position of Codebook with minimum distortion. Second, decoder will bring the index to direct the same codebook as encoded [51].

The block diagram of vector quantization is shown in below figure:

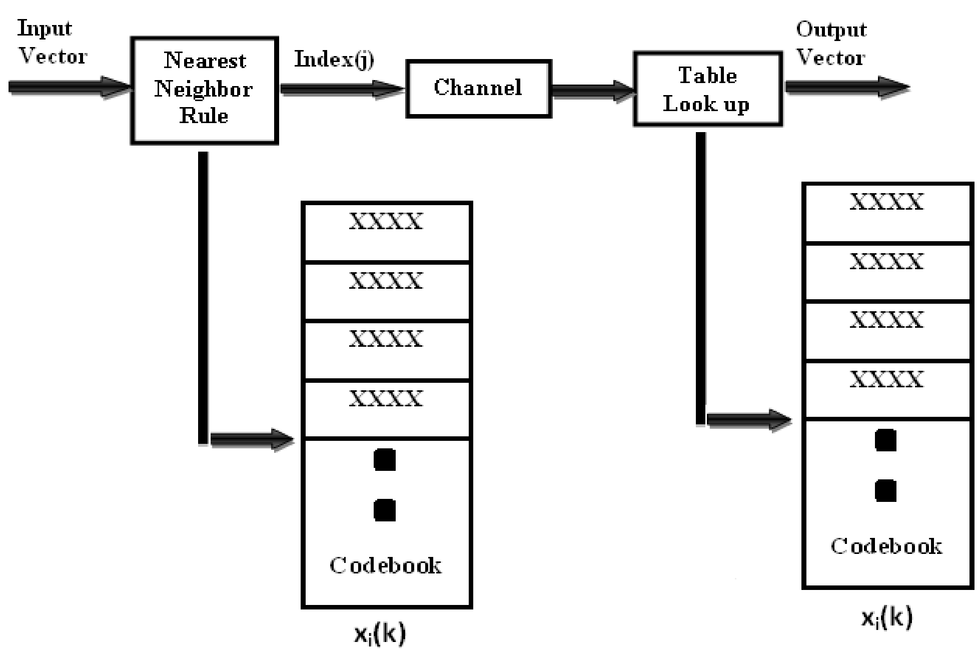


Figure – 4.58 Block diagram of Vector Quantization [51]

4.3.2.2 Codebook Design

Searching for the best codebook from studying the training set, Linde-Buzo-Gray algorithm (LBG) is the famous algorithm for codebook. It starts with define in the value of Y_m, and then update Codebook by replacing the value that makes less distortion. This update step will iterate until the distortion is under the LBG limit. LBG instructions will split the image into the same size parts and forms to training vectors $x_j(k)$ as[51]:

$$\mathbf{X_j(k) = [X_j(1)\ \ X_j(2)\ \ X_j(3) \ldots X_j(k)]}$$

Where j = 1, 2, 3..., n;

n = number of the image parts and

k = vector dimension; (k=1, 2, 3..., 16)

4.3.2.3 Steps of LBG

Step – 1 : Initiate the value of Codebook

$$\boldsymbol{x_j(k) = [x_i(1)\ \ x_i(2)\ \ x_i(3) \ldots x_i(k)]}$$

where i=1,2,3,....,N_c and N_c is number of vector in codebook [51].

Index (j)	Training Vector ($x_j(k)$)
1	$x_1(1)\ldots\ x_1(16)$
2	$x_2(1)\ldots\ x_2(16)$
3	$x_3(1)\ldots\ x_3(16)$
⋮	⋮
16,384	$x_{16384}(1)\ldots\ x_{16384}(16)$

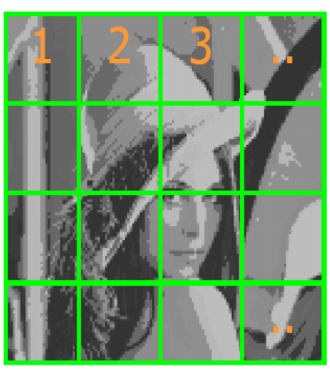
Table – 4.8 Training Vector [51]

Index (i)	Codebook ($\hat{x}_i$ (k))
1	$\hat{x}_1\ (1)\ldots\ \hat{x}_1\ (16)$
2	$\hat{x}_2\ (1)\ldots\ \hat{x}_2\ (16)$
3	$\hat{x}_3\ (1)\ldots\ \hat{x}_3\ (16)$
⋮	⋮
256	$\hat{x}_{256}\ (1)\ldots\ \hat{x}_{256}\ (16)$

Table – 4.9 Codebook [51]

Source Image

Training Vector



Index (j)	Training Vector ($x_j(k)$)
1	$x_1(1)\ldots x_1(16)$
2	$x_2(1)\ldots x_2(16)$
3	$x_3(1)\ldots x_3(16)$
⋮	⋮
16,384	$x_{16384}(1)\ldots x_{16384}(16)$

Figure – 4.59 Source image and training vector [51]

Step – 2 : Vector group (X_j) is the training vectors by the number of codebook vector $P(x_i(k))=(S_i;\ i=1,2,3,\ldots,N_c)$; S_i is the set of minimum distortion vectors. The grouping will use the minimum distortion values, which can find from: $X_j \in S_i$ if

$$d(x_j, \hat{x}_i) \leq d(x_j, \hat{x}_i)$$

For every *i* we can obtain the minimum distortion value from [51]:

$$d(x_j, \hat{x}_i) = \frac{1}{k} \sum_{m=1}^{k} \left[x_j(m) - \hat{x}_i(m)\right]^2$$

Index (i)	Group 1 ($X_{j=1}(k)$)
1	$X_{1,1}(1) \ldots X_{1,1}(16)$
2	$X_{1,2}(1) \ldots X_{1,2}(16)$

$$\vdots$$

$$l_1 \in X_{j=1}$$

Index (i)	Group 2 ($X_2(k)$)
1	$X_{2,1}(1) \ldots X_{2,1}(16)$
2	$X_{2,2}(1) \ldots X_{2,2}(16)$

$$\vdots$$

$$l_2 \in X_{j=2}$$

Index (i)	Group 3 ($X_3(k)$)
1	$X_{3,1}(1) \ldots X_{3,1}(16)$
2	$X_{3,2}(1) \ldots X_{3,2}(16)$

$$\vdots$$

$$l_3 \in X_{j=3}$$

Index (i)	Group 256 ($X_{256}(k)$)
1	$X_{256,1}(1) \ldots X_{256,1}(16)$
2	$X_{256,2}(1) \ldots X_{256,2}(16)$

$$\vdots$$

$$l_{256} \in X_{j=256}$$

Table – 4.10 Vector Group [51]

Training Vector

Index (j)	Training Vector ($x_j(k)$)
1	$x_1(1)\ldots x_1(16)$
2	$x_2(1)\ldots x_2(16)$
3	$x_3(1)\ldots x_3(16)$
⋮	⋮
16,384	$x_{16384}(1)\ldots x_{16384}(16)$

Codebook Design

Index (i)	Codebook ($\hat{x}_i$ (k))
1	$\hat{x}_1(1)\ldots \hat{x}_1(16)$
2	$\hat{x}_2(1)\ldots \hat{x}_2(16)$
3	$\hat{x}_3(1)\ldots \hat{x}_3(16)$
⋮	⋮
256	$\hat{x}_{256}(1)\ldots \hat{x}_{256}(16)$

Codebook Table

Table – 4.11 Codebook design [51]

New Image (Index Table)

i_1	i_5	i_{39}	i_{10}
i_{45}	i_{45}	i_{200}	i_{12}
i_{54}	i_{256}	i_{96}	i_{78}
i_{256}	i_{54}	i_5	i_9

Codebook Table

Index (i)	Codebook ($\hat{x}_i$ (k))
1	$\hat{x}_1(1)\ldots \hat{x}_1(16)$
2	$\hat{x}_2(1)\ldots \hat{x}_2(16)$
3	$\hat{x}_3(1)\ldots \hat{x}_3(16)$
⋮	⋮
256	$\hat{x}_{256}(1)\ldots \hat{x}_{256}(16)$

Table – 4.12 Codebook table [51]

Step – 3: Average distortion (D_{m+1}). The total number of group, which are computed, between $X_j(k)$ and $X_i(k)$ is 256 groups. Then the average distortion can be calculated from each group, is given by[51]:

$$D_{m+1} = D[(Ym, P(Ym))] = \frac{1}{n}\sum_{j=1}^{n} \min_{\hat{x}\subset Ym} d(x_j, \hat{x}_i)$$

Step – 4: If

$$\frac{D_m - D_{m+1}}{D_{m+1}} \leq \varepsilon$$

stop update the codebook [51].

We can obtain the new codebook (x_i(k)').

Index (i)	New Codebook ($\hat{x}_i$ (k)')
1	$\hat{x}_1$ (1)'… $\hat{x}_1$ (16)'
2	$\hat{x}_2$ (1)'… $\hat{x}_2$ (16)'
3	$\hat{x}_3$ (1)'… $\hat{x}_3$ (16)'
⋮	⋮
256	$\hat{x}_{256}$ (1)'… $\hat{x}_{256}$ (16)'

Table – 4.13 New Codebook [51]

Otherwise, the process will continue update in the codebook by average the group in 2nd step vectors as:

$$\hat{x}\big(P(Ym)\big) = (\hat{x}(S_i); i = 1, 2, 3, \ldots, N_c)$$

For $\mathbf{P}(\hat{\mathbf{x}}_\mathbf{i}\,(\mathbf{k}))$

Where

$$\hat{x}(S_i) = \frac{1}{\|S_i\|}\sum_{j:xj\in Si}^{m} x_j$$

The result $Y_{m+1}=x(S_i)$ increase i by 1, and then back to the 1st step also the codebook design presentation in the above given table [51].

The decoding of new codebook and the suitable index yields the output vector and codebook vectors region according to the image compression.

4.3.3 Experimental Study

Vector quantization image compression algorithm experiment is performed on Lena, Girl and Mandrill images. This experiment was performed on algorithm based on both random and splitting type codebook. The following table represents the experimental statistical values on different images.

Original Image

Compressed Image

Figure – 4.60 Original and compressed Lena image

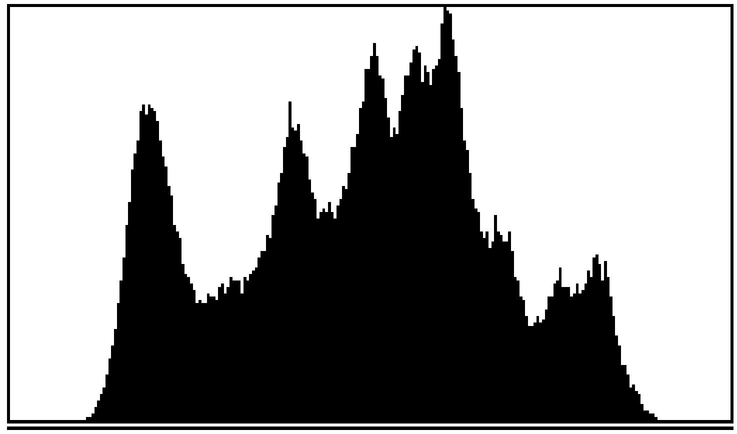


Figure – 4.61 Histogram of Lena image

Mean	124.1
Median	129
Mode	146
Standard Deviation	47.85
Median Absolute Deviation	33
Mean Absolute Deviation	39.72

Table – 4.14 Statistics of Lena image

The following table shows the PSNR (db) value of both random and splitting type codebook and bit rate (bpp) of Lena image.

PSNR - VQS	PSNR - VQR	Bit Rate
15.2	12.1	0.008
19.6	16	0.031
21.8	20.5	0.047
22.8	23.6	0.078
19.6	17.8	0.125
23.8	20.8	0.188
25.2	23.4	0.25
19.9	16.9	0.5
24.7	23.2	0.75
28.5	25.3	0.95

Table – 4.15 Lena image compression result

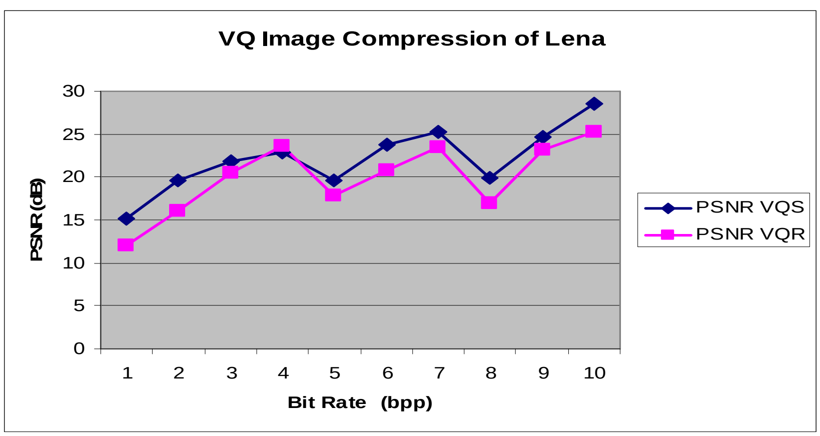


Figure – 4.62 Bit Rate v/s PSNR in VQ of Lena image chart

Original Image

Compressed Image

Figure – 4.63 Original and compressed Girl image

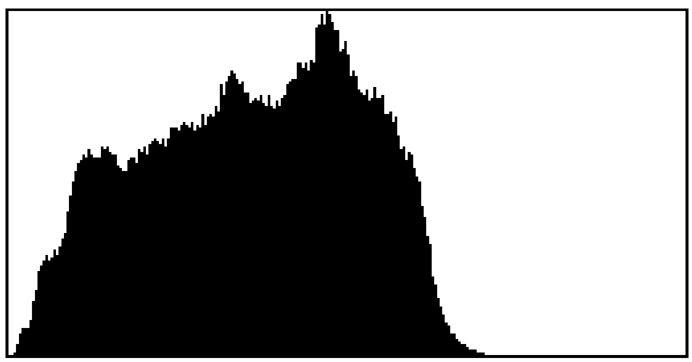


Figure – 4.64 Histogram of Girl image

Mean	92.17
Median	95
Mode	122.6
Standard Deviation	40.53
Median Absolute Deviation	32
Mean Absolute Deviation	34.57

Table – 4.16 Statistics of Girl image

The following table shows the PSNR (db) value of both random and splitting type codebook and bit rate (bpp) of Girl image.

PSNR - VQS	PSNR - VQR	Bit Rate
18.8	13.2	0.008
20.4	16.8	0.031
23.7	23.6	0.047
24.6	23.8	0.078
20.1	19.2	0.125
24.2	24.7	0.188
27.5	26.2	0.25
20.4	18.4	0.5
24.2	26.1	0.75
28.2	27.8	0.95

Table – 4.17 Girl image compression result

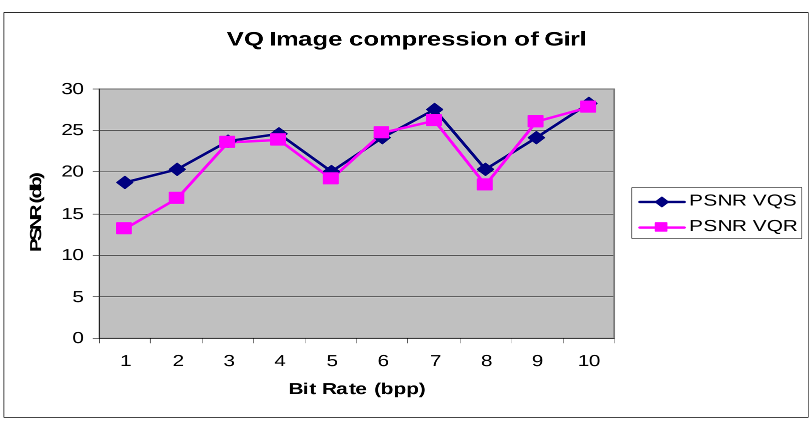


Figure – 4.65 Bit Rate v/s PSNR in VQ of Girl image chart

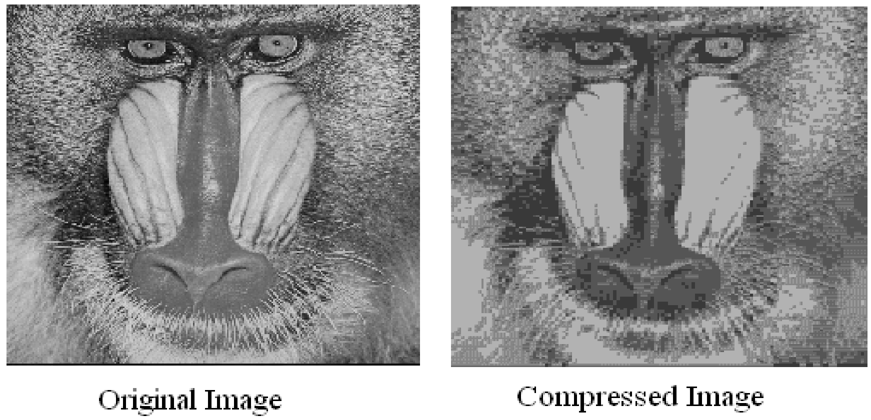


Figure – 4.66 Original and compressed Mandrill image

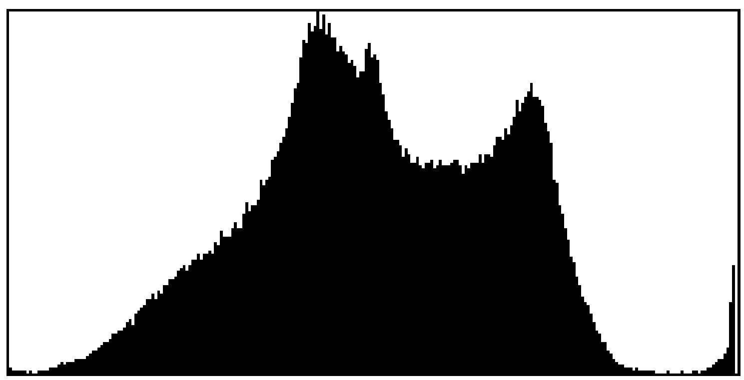


Figure – 4.67 Histogram of Mandrill image

Mean	130.4
Median	128
Mode	105.6
Standard Deviation	43.88
Median Absolute Deviation	32
Mean Absolute Deviation	36

Table – 4.18 Statistics of Mandrill image

The following table shows the PSNR (db) value of both random and splitting type codebook and bit rate (bpp) of Mandrill image.

PSNR - VQS	PSNR - VQR	Bit Rate
15.2	9.2	0.008
16.8	16.8	0.031
17.2	17.9	0.047
17.4	18.7	0.078
17.4	16.1	0.125
18.7	19.1	0.188
19.2	19.8	0.25
18.9	16.7	0.5
21.2	21.2	0.75
21.9	23.4	0.95

Table – 4.19 Mandrill image compression result

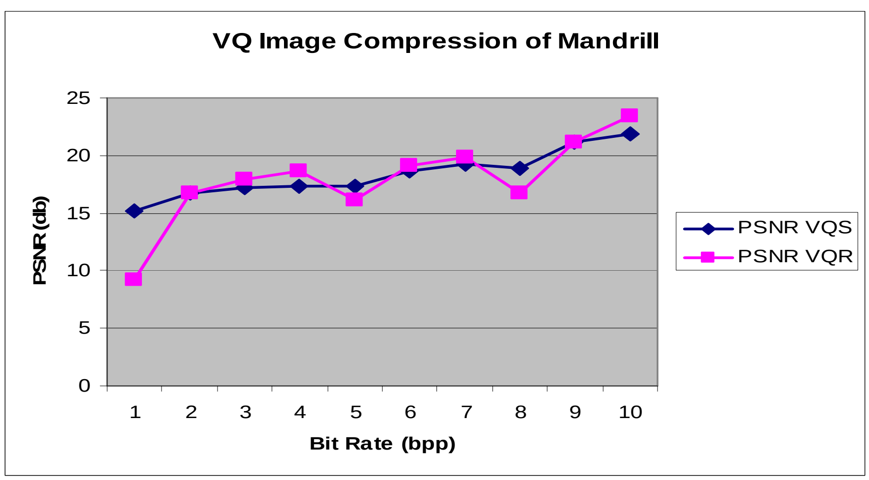


Figure – 4.68 Bit Rate v/s PSNR in VQ of Mandrill image chart

4.3.4 Conclusion

From the experimental study of different images such as Lena, Girl and Mandrill, it can be concluded that when the bit rate increases, the PSNR also increases in both VQS and VQR.

4.4 FRACTAL COMPRESSION ALGORITHM

4.4.1 Introduction

Fractal based image compression (FIC) techniques exploit redundancy due to self similarity properties in images to achieve compression. A fractal may be defined as a geometrical shape that is self-similar i.e. it has parts that is similar to the whole. An example of a fractal is the figure given below [52]:

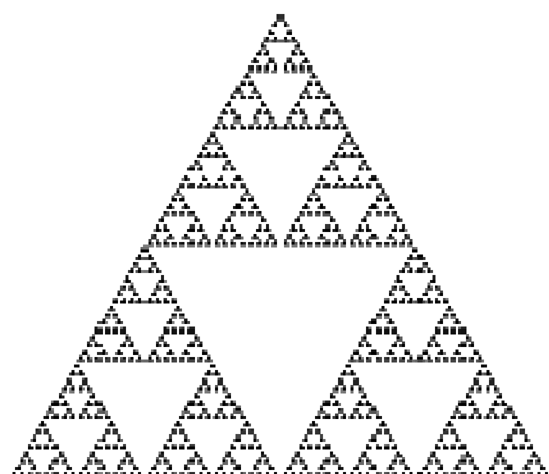


Figure – 4.69 The Sierpinski triangle [52]

In the above given figure, the three triangles at each corner of a triangle are similar to the whole triangle. The amount of information therefore needed to describe such images is a lot lesser than it appears and hence image compression using fractal based methods is possible. Fractal based image compression exploits redundancy due to ordinary symmetry/similarity also in addition to self similarity [52].

4.4.2 Fractals

The term fractal was first used by French mathematician Benoit B. Mandelbrot in 1975. This word was derived from the Latin word fractus, which means "broken", or "irregular and fragmented". The birth of Fractal Geometry is traced by Benot B. Mandelbrot. He has published his seminar book “The Fractal Geometry of Nature” in 1977. He claimed that classical Euclidean Geometry was not enough to describing many natural objects, such as trees, mountains, clouds, and coastlines. So he developed the fractal geometry.

There is no specific definition that exists. It includes all sets considered to be fractals. There are certain properties that can be used to classify a set as a fractal. We consider a set F to be a fractal if it has detail at every scale, is (exactly, approximately, or statistically) self similar, and can be described by a simple algorithm. An example of a fractal is shown below.

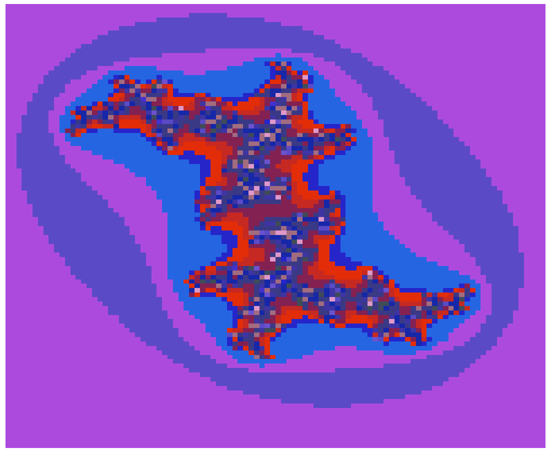


Figure – 4.70 Fractal example

Notice that the image has detail at every scale, and actually exhibits all of the properties described above.

There are two main groups of fractals:

- Linear
- Nonlinear.

The Mandelbrot set and Julia sets are fractals of the complex plane. The fractals used in image compression are linear, and of the real plane. So, they are not chaotic fractals. In other words, they are not susceptible to initial conditions. They are the fractals from Iterated Function Theory. An Iterated Function System (IFS) is simply a set of contractive affine transformations. It efficiently produces shapes such as ferns, leaves and trees.

4.4.3 Fractal Image Compression

The fractal image compression method uses the mathematical principles of fractal geometry. It is used to identify the redundant repeating patterns from the images.

These types of matching patterns may be identified by performing some geometrical transformation, such as rotating, scaling, etc… on the image. These indentified repeating patterns are stored once together with the information on its locations within the image and the required transformations in each case. It is a lossy compression technique, which can achieve large compression rates but the information or data may be lost during the compression. It works best with complex images and high color depths. Imagine a special type of photocopying machine that reduces the image to be copied by half and reproduces it three times on the copy. The figure given below shows the copy machine [53].

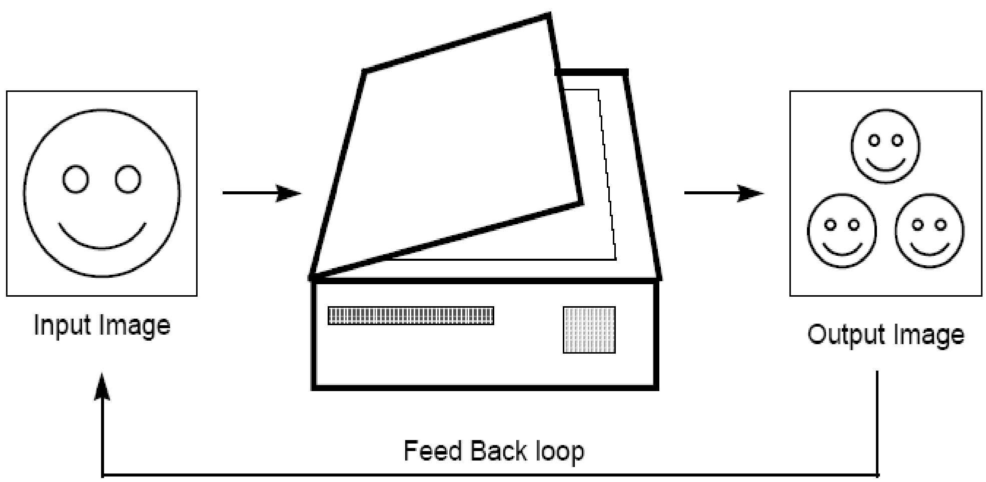


Figure – 4.71 Copy machine that makes three reduced copies of the input image [53] [54]

The figure given below shows several iterations of this process on several input images.

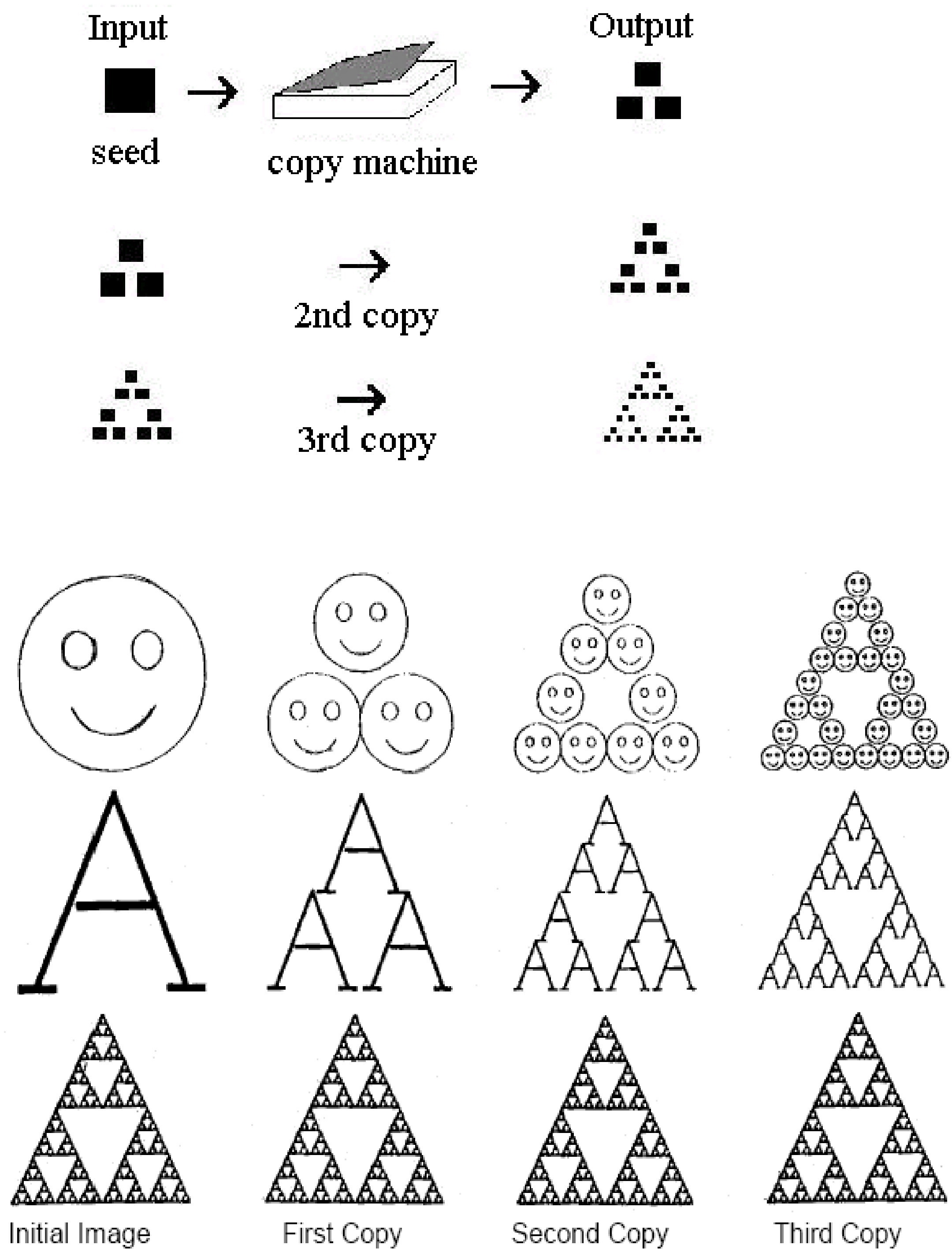


Figure – 4.72 First three copies generated on the copying machine [53] [54]

If we observe that all the copies appear to meet to the same as the final image. The copying machine reduces the input image. Any image can be placed on the copying machine, it will be reduced to a point as we repeatedly run the machine; It is only the position and the orientation of the copies that determines what the final image looks like.

It is the way in which the input image is transformed, and it determines the final result of copy of the image when running the copy machine in a feedback loop. We must constrain this transformation with the limitations. The limitation is that the transformation must be contractive. It is a given transformation applied to any two points in the input image must bring them closer to the copy. This technical condition is quite logical, since if points in the copy were spread out the final image would have to be of infinite size. Except for this condition the transformation can have any form.

In practice, choosing transformations of the form

$$w_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i \, b_i \\ c_i \, d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}$$

Where x, y – coordinates; a, b, c, d, e, f – coefficients, is sufficient to generate different interesting transformations. These types of transformation are called affine transformations of the plane. This transformation can skew, stretch, rotate, scale and translate the input image.

A common feature of these transformations is that, it run in a loop back mode. It means that for a given initial image each image is formed from transformed and reduced copies of itself, and hence it must have detail at every scale. That is, the images are fractals. This method of generating fractals is due to John Hutchinson, and more information about the various ways of generating such fractals can be found by Barnsley and Peitgen, Saupe, and Jurgens.

Each transformation w_i is defined by 6 numbers, a_i, b_i, c_i, d_i, e_i, and f_i , is shown in above equation, which do not require much memory to store on a computer (4 transformations x 6 numbers / transformations x 32 bits /number = 768 bits). Storing the image as a collection of pixels however requires much more memory (at least 65,536 bits for the resolution).

Three basic problems have been the focus of study in fractal image compression:

- Determining a family of contraction maps that can be used to effectively code images.
- Finding fast and effective algorithms for associating an image with IFS of which the image is the attractor.
- Analyzing the convergence properties of various families of contraction maps and to establish error bounds for decoded images.

The main competitor of fractal image compression is JPEG standard. This is a compression procedure that makes use of a modified form of Fourier analysis. It also divides up the image into small blocks 8 X 8 pixels in this case. JPEG does produce good compression results but it is claimed that it is inferior to fractal compression method.

One of the reasons is that fractal methods are not scale dependent. You can decompress a fractal image to any resolution you like by choosing the size of the image that you iterate. Of course beyond a certain resolution the extra detail you will see will be artificial but the argument is that because of its fractal nature it will not be too "obvious" and the image will still look natural.

The second advantage is that fractal compression doesn't share a defect that is inherent in the JPEG and all Fourier methods.

JPEG compression is almost universally used and many of the early claims for the powers of fractal compression made in the 1990s never really came to be. Fractal compression takes longer to implement and in most cases the gain in compression ratio over JPEG isn't significant. Also the claimed "better quality at higher resolution" also isn't really an issue.

Contractive Transformations

A transformation w is said to be contractive if for any two points $P1$ and $P2$, the distance

$$d(w(P1),w(P2)) < sd(P1,P2)$$

For some $s < 1$, where d is distance.

This formula says the application of a contractive map always brings points closer together (by some factor less than 1)[53].

The Contractive Mapping Fixed Point Theorem

The theorem says something that is intuitively obvious: if a transformation is contractive then when applied repeatedly starting with any initial point, we coverage to a unique fixed point.

If X is a complete metric space and W : X→ X is contractive, then W has a unique fixed point | W |[53].

4.4.4 Why the name "Fractal"

The image compression scheme describes the fractals in different senses [55]. This scheme will encode the image as a collection of transforms. It is very similar to the copy machine. There is no natural size in decoded image. It can be decoded at any size. Some extra details are required for decoding at larger sizes is generated automatically by the encoding transforms [53] [55] [56].

It can decode the image of a person to increase the size by performing the iteration and it can see the skin cells or perhaps atoms also, but it is not possible. Here the details are not related to the actual detail present when the image was digitized. It is only product of encoding transforms which originally only encoded the large scale features. In some cases the detail is realistic at low magnifications. It can be useful in security and medical applications related to image [53].

4.4.5 History of Fractal Image Compression

In 1977 Benoit Mandelbrot finishes the first edition of "The Fractal Geometry of Nature".

In 1981 John E. Hutchinson publishes "Fractals and Self Similarity".

In 1983 revised edition of "The Fractal Geometry of Nature" is published.

In 1985 Michael F. Barnsley and Stephen Demko introduce Iterated Function Theory in their paper "Iterated Function Systems and the Global Construction of Fractals".

In 1987 Iterated Systems Incorporated is founded in Georgia, US.

In 1988 Michael Barnsley and Alan D. Sloan wrote the article "A Better Way to Compress Images" published in Byte, claiming fractal compression ratios of 10000 to 1.

Barnsley publishes the book "Fractals Everywhere", which presents the mathematics of Iterated Function Systems, and proves a result known as the Collage Theorem.

In 1990 Barnsley's first patent is granted (US Patent # 4 941 193).

In 1991 M. Barnsley and A. Sloan are granted US Patent # 5 065 447 "Method and Apparatus for Processing Digital Data".

J.M. Beaumont publishes a paper "Image Data Compression Using Fractal Techniques".

In 1992 Arnaud E. Jacquin publishes an article that describes the first practical fractal image compression method.

Iterated Systems Inc finally breaks even and begins to make money.

Microsoft Corporation publishes the multimedia encyclopedia "Microsoft Encarta" which uses fractal image compression to great effect.

In 1993 the book "Fractal Image Compression" by Michael Barnsley and Lyman P. Hurd is published.

The Iterated Systems' product line matures.

In 2000 “The Mathematical Foundation of Image Compression” by Lisa A. Soberano.

In 2001 “Fractal-Based Technology to Compress Digital Image Files” Research and data for Status Report.

In 2009 “Genetic Algorithm Applied to Fractal Image Compression” was published in ARPN Journal of Engineering and Applied Sciences by Y. Chakrapani and K. Soundara Rajan.

In 2010 Project Final Report on “A Study on Fractal Image Compression” was prepared by Darshan S Alagud [57].

4.4.6 Mathematical Representation

Mathematically, in fractal image compression, an iterated function system (IFS) describes an image. An IFS is a set of contractive affine transforms. An affine transform is a linear transform of the form [58]:

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_{i1} & a_{i2} & 0 \\ a_{i3} & a_{i4} & 0 \\ 0 & 0 & c_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} d_{i1} \\ d_{i2} \\ b_i \end{bmatrix}$$

In the above equation, w_i is the transform applied on a point (x, y, z) in the i^{th} block of input image. Gray scale images can be represented by such points (x, y, z), in which case z would be the intensity or gray level at pixel co-ordinates (x, y) in an image [58].

The a_{i1}, a_{i2}, a_{i3}, a_{i4}, d_{i1} and d_{i2} are constants for the i^{th} image block to which the transform is applied. An affine transform is said to be contractive if the distance between transformed points is less than that between the original points in the metric space [58].

For example, consider the below IFS consisting of 2 affine transforms represented by their effect on a square. Both functions are applied to the input image and a union (U) of the resulting images is formed in iteration each. First three iterations are shown and then the final image after several iterations is shown [58].

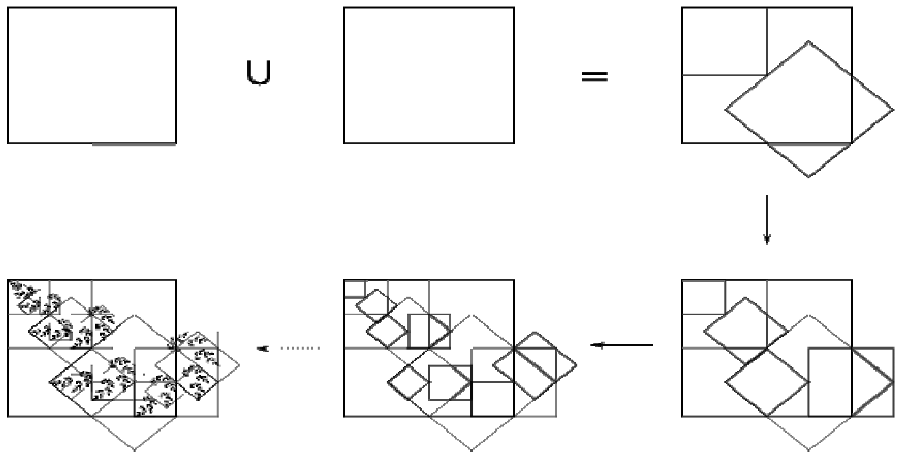


Figure – 4.73 Construction of an image using IFS of 2 affine functions [58]

4.4.7 The Collage Theorem

Theorem:

Let (X, d) be a complete metric space. Let T ∈ H(X) be given, and let $\varepsilon \geq 0$ be given. Choose an IFS (or IFS with condensation) {X; (w_0), w_1, w_2,..., w_n} with contractivity factor $0 \leq s < 1$ so that

$$h\left(T, \bigcup_{n=1 \; (n=0)}^{N} w_n(T) \right) \leq \varepsilon,$$

Where h(d) is the Hausdorff metric. Then

$$h(T, A) \leq \frac{\varepsilon}{1 - s},$$

Where A is the attractor of the IFS. Equivalently [59],

$$h(T, A) \leq (1 - s)^{-1}\, h\left(T, \bigcup_{n=1\,(n=0)}^{N} w_n(T) \right) \text{ for all } T \in \mathrm{H}(\mathbf{X}).$$

Proof:

The Collage Theorem is used to find the integrated function system, whose attractor is “close to” or “looks like” a given set. It must find the set of transformation - contraction mappings on suitable space within the given set of the lines, such that the union or collage of the images of the given set under the transformations is near to the given set. Housdroff metric is used to measure nearness. In other words, if an IFS can be found, which is applied once to the image, it gives the image which looks like the original image. After that iterating the IFS on any starting image, gives a result which is more like than the original image. This result is the key for finding IFS, which is used to compress the image. This theorem cannot solve the inverse problem, but it can provide the way of approaching it [59].

4.4.8 Fractal Image Coding

The fractal is a structure, which is made by the similar forms. The patterns of the fractal occur in many different sizes. The term fractal was first used by Mandelbrot for describing the repeat pattern, which is observed occurring in many different structures. These patterns appeared nearest to each other and identical in form at any size. He also discovered that the fractal is also described using mathematical terms. It can be created using very small and finite data and algorithms.

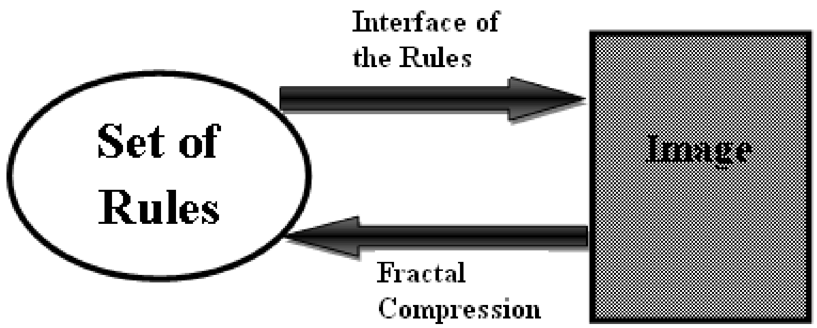


Figure – 4.74 Fractal Compressions

Fractals can be considered as a set of mathematical equations that generate fractal images; images that have similar structures repeating themselves inside the image. The image is the inference of the rules and it has no fixed resolution. The idea of fractal compression is to find a set of rules that represents the image to be compressed, which is shown in the above given figure. The decompression is the inference of the rules. Fractal compression tries to decompose the image into smaller regions which are described as linear combinations of the other parts of the image. These linear equations are the set of rules.

The theorems tell us that transformation W will have a unique fixed point in the space of all images. That is, whatever image we start with, we can repeatedly apply W to it and we will converge to a fixed image. The figure given below shows the encoding process.

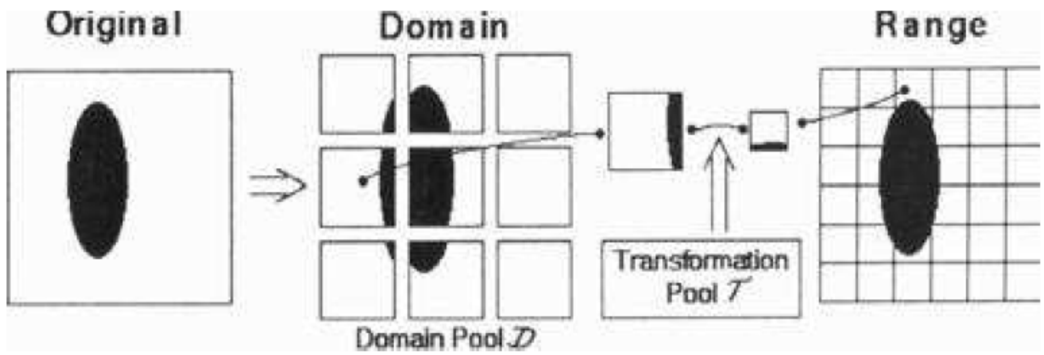


Figure – 4.75 The encoding process [60]

For example, we have given an image f, which is encoded. That means it is required to find the different collection of transformations w_1, w_2,.....,w_N. The image f can be fixed point of the map W. That means it divides the image f into different pieces to which it applies the transformation W_i and it gets back the original image f[60][61].

The main theory of fractal image coding is based on iterated function system, attractor theorem and collage theorem. Fractal coding process is quite complicated but decoding process is very simple, which makes use of potentials in high compression ratio.

Fractal image coding attempts to find a set of contractive transformations that map domain cells onto a set of range cells that tile the image. The basic algorithm for fractal encoding is as follows:

- The image is partitioned into non overlapping range cells { Ri } which may be rectangular or any other shape such as triangles.

- The image is covered with a sequence of possibly overlapping domain cells. The domain cells occur in variety of sizes and they may be in large number.
- For each range cell the domain cell and corresponding transformation that best covers the range cell is identified. The transformations are generally the affined transformations. For the best match the transformation parameters such as contrast and brightness are adjusted.
- The code for fractal encoded image is a list consisting of information for each range cell which includes the location of range cell, the domain that map onto that range cell and parameters that describe the transformation mapping the domain onto the range.

The attractive feature of fractal image compression is that it is resolution independent in the sense that when decompressing, it is not necessary that the dimensions of the decompressed image be the same as that of original image.

4.4.9 Transformations

4.4.9.1 Affine Transformation

Affine Transformation is applied to any image. It is the combination of the translation, rotation, skew and scaling. For example, if we take any image, then we rotate the image at 90 degree and after that it stretches, at that time you have subjected to an affine transformation. The transformation which is not affine, that bends the line that was straight, tears the image apart or introduces holes.

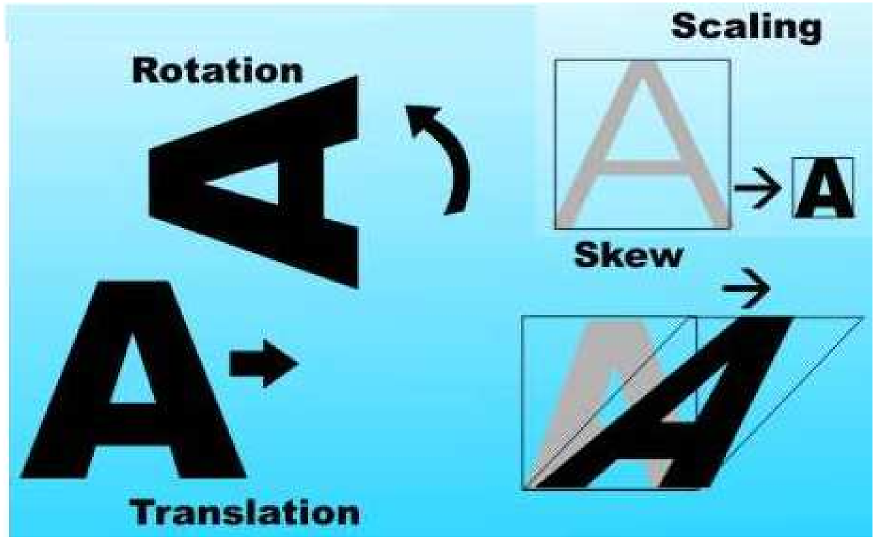


Figure – 4.76 Translation, Rotation, Skew and Scaling

In matrix notation a general affine transformation can be written as follows:

x' = Ax + d

Where x is a co-ordinate vector i.e. x,y.

A is a 2 X 2 matrix and

d is a displacement vector.

Writing this out in full gives:

[x'] = [a b] [x] + [e]
[y'] = [c d] [y] + [f]

OR

x' = ax + by + e
y' = cx + dy + f

Where a, b, c, d, e and f are constants, which is used to determine the exact nature of the transformation. The e and f are used to control the translation and the values of a, b, c and d control the rotation, scaling and skew.

4.4.9.2 Contractive Transformation

A special type of affine transformation is a contractive transformation. It is very broad concept; there is not specific definition of affine transformation. If you really want an accurate definition, then an affine transformation f is contractive if for all x and y:

$$d\,(f(x),f(y)) <= s\; d(x,y)$$

Where s is a constant between 0 and 1 and d(x, y) is the distance between the points x and y.

You can see from the definition that a contractive transformation moves points closer together.

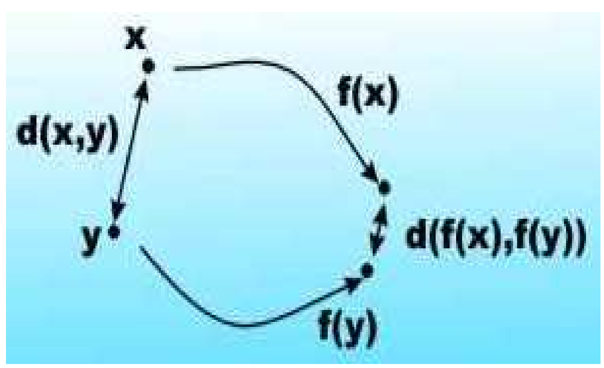


Figure – 4.77 Contractive Map

The constant S is called a contraction factor of the transformation and the largest such value gives you a measure of how much the contractive transformation moves points together.

Contractive transformations is very important because the reason is that if you take a point, any point, and apply the transformation and then apply the transformation to the result and so on, the sequence of points so generated moves closer and closer to a fixed point or limit. That is, if you iterate a contractive affine transformation it eventually settles down to a stable result.

In the figure given below, the effects of the contractive transformation f rotate 25 degrees and shrink by 50% on a square of points. All of the points move towards the single point at the center.

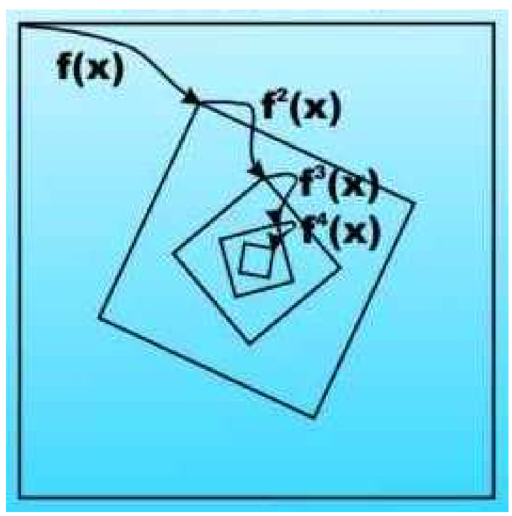


Figure – 4.78 Effect of contractive transformation

In mathematical terms iterating an affine transformation on a point x is written

x, f(x), f(f(x)) f(f(f(x))) and so on.

It is usual to write fn(x) to mean the result of applying fn times to x. This allows us to write the important property of a contractive transformation as:

$$\mathbf{fn(x) \rightarrow z \quad as\ n \rightarrow infinity}$$

Where → is read as "tends to". That is as n gets bigger fn(x) tends to a fixed value z.

4.4.9.3 Iterated Function Systems (IFS)

The contractive affine transformation can generate a fractal. By repeating the transformation an infinite number of times any starting pattern is transformed into a repeating pattern with the same structure at any level of detail and there are one of the basic characteristics of a fractal set.

To perform this experimentally, we need to define an iterated function system. This is just a collection of contractive affine maps c_1, c_2 and so on to c_n. The contraction factor of the set of transformations is simply the largest of each of their individual contraction factors.

The usual way of combining transformations is to apply them one after another. For example, $c_2\ c_3\ c_1(x)$ is just the result of applying c_1 to x and then c_3 to the result and then c_2 to the result of that. This is just the familiar composition of functions. IFS, on the other hand, combines its transformations by allowing them to each act on the original and forming the union of all their results.

In the figure given below, the usual method of making transformations work together is to apply them one after another. That calculates $C_3(C_2(C_1(x)))$

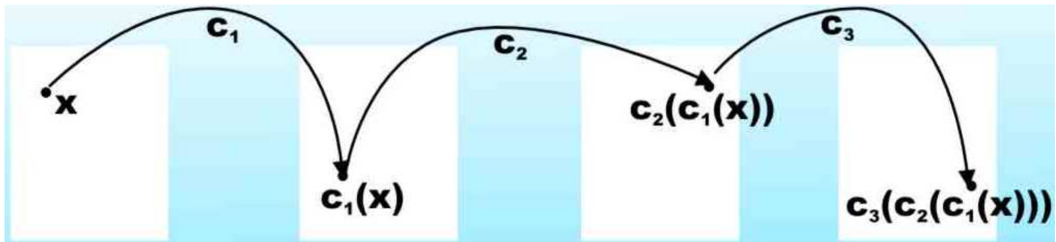


Figure – 4.79 Methods of Making Transformation

In the figure given below, an iterated function system works together by transforming each point by each transformation. That calculates $C_1(x)$ U $C_2(x)$ U $c_3(x)$

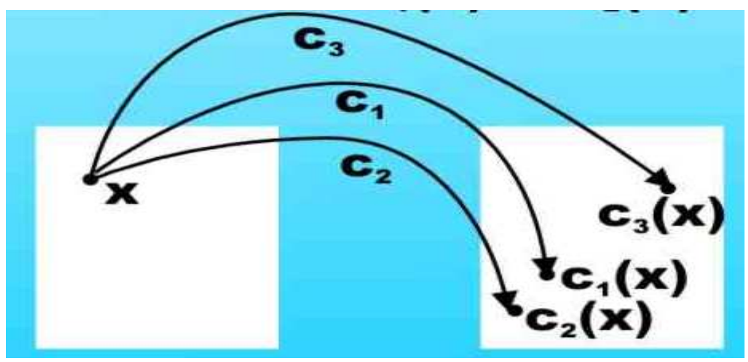


Figure – 4.80 IFS transformation

This doesn't make any sense unless you think of the transformations acting on sets of points. That is, if A is a set of points then the result of applying the IFS to A is:

c_1(A) U c_2(A) U c_3(A) and so on

Where U means the "union of" and c_1 (A) is the set of points that A is transformed into.

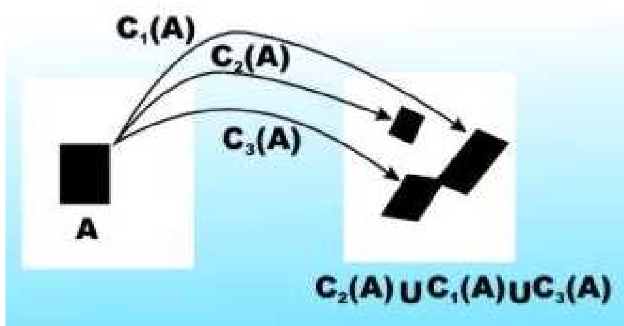


Figure – 4.81 IFS applying to affine transformations

In the above given figure, when applying an IFS to a set A each of the affine transformations creates a separate image and the result is the union of the images.

4.4.9.4 Iterating the IFS

The simple contractive transformations, applying IFS, i.e. iterating it, the result slowly settles down to a final result. In other words, the output of IFS converges to a particular set of points. This set of points is called the attractor of the IFS. In mathematical terms:

$$C_n(B) \rightarrow A \text{ as } n \rightarrow \text{infinity}$$

Where C_n is the IFS applied n times, B is any set of points and A is the attractor.

It also isn't difficult to see that if A is the attractor of the IFS then:

$$C(A)=A$$

That is the attractor is also a "fixed point" of the transformation.

4.4.9.5 An Attractor

The IFS converges to a particular result if you iterate it enough times on any starting set. A theoretical concept is accepted for the fact that the sets that an IFS can be made to coverage to can be very complex fractals even.

For example, it is worth introducing a shorthand way of writing IFS. Instead of writing out the values in matrix form it is simpler to list them in a table. If the contractive affine transformation is:

$$x' = ax + by + e$$
$$y' = cx + dy + f$$

Then the IFS can be listed in a table of values with columns a, b, c, d, e and f. For example, an IFS consisting of four affine transformations can be written as follow:

a	b	c	d	e	f
0.5	0	0	0.5	1	1
0.5	0	0	0.5	1	50
0.5	0	0	0.5	50	50

Table – 4.20 IFS affine transformation

Which you have to admit is shorter than writing out the four sets of equations.

The attractor of this simple IFS, start from any old set of points and apply the IFS. Next you apply the IFS to the result and carry on until there is no change in the output. For example, starting from a random set of dots you can see that the image slowly converges to something very surprising, a **Sierpinski triangle**.

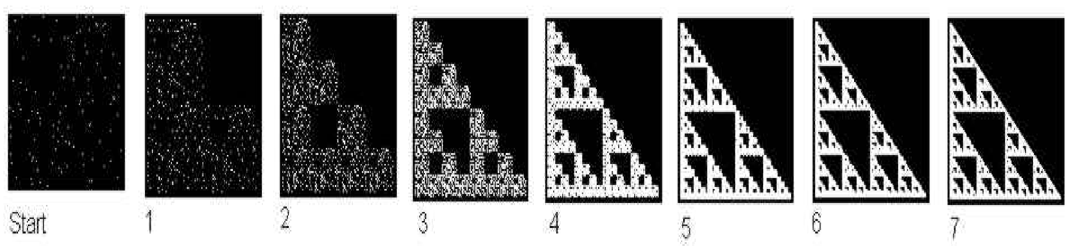


Figure – 4.82 The random set of points converges to the Sierpinski triangle

It doesn't matter what the starting set of points is, the end product is a Sierpinski triangle which is the attractor for this IFS.

At last we are very close to seeing how all of this might be used as a method of compressing images. Notice that the Sierpinski triangle is a very complex image. Also notice that the IFS method of generating it is resolution independent. If you start off with an array of 100x 100 points or 1000x1000 or whatever the transformation can be applied, you will see more detail the more points you have. Also notice that 18 coefficients generate this image. In the case of 100 X 100 points a 1.22KByte binary image can be compressed to 18Bytes - a compression ratio of just less than 70:1. Of course by increasing the size of the image generated the ratio can be made even higher because 18 bytes serve to generate the Sierpinski triangle at any resolution.

The basic idea is that we try to find IFS that will generate a given image and store the IFS coefficients in place of the original. When the original is required all we have to do is iterate the IFS on an arbitrary starting image of the desired resolution.

The collage theorem is used to find IFS that will generate a given image.

The collage theorem says that if

$$d(C(T),T) < d$$

then

$$d(T,A) < d\,/(1-s)$$

Where C is an IFS with contractivity s and attractor A and T is any image. The first line says that if the distance between C (T) and T is small then the second line tells you that the distance between T and A is even smaller. In other words, if you can find an IFS which when applied once to T gives an image that looks like T then iterating the IFS on any starting image gives a result that is even more like T. This is an amazing result and it is the key to finding IFS that can be used to compress an image.

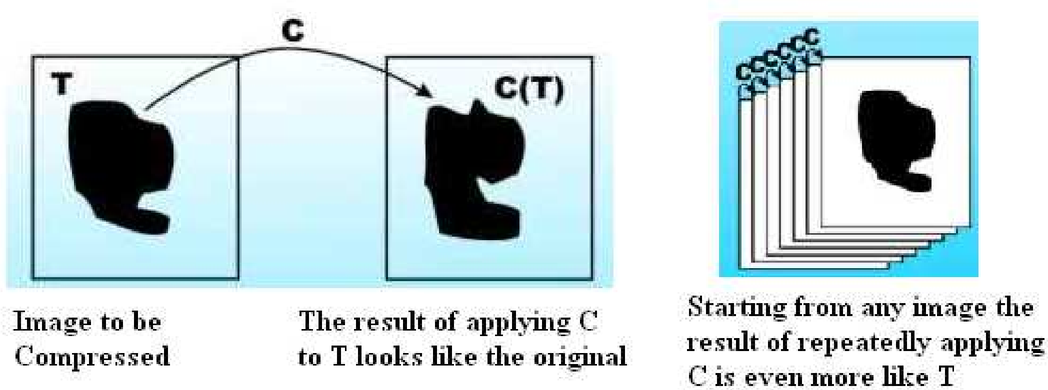


Figure – 4.83 Finding IFS for image compression

A secondary result that helps is the continuity condition. This says that as you alter the coefficients of the IFS the attractor changes smoothly and without any sudden jumps. This means that one method of performing the compression is to write a program which allows a user to manually fiddle with a set of controls in an effort to transform the original image into a passable likeness of itself under a set of contractive affine transformations. The resulting set of coefficients could then be used as the compressed version of the image.

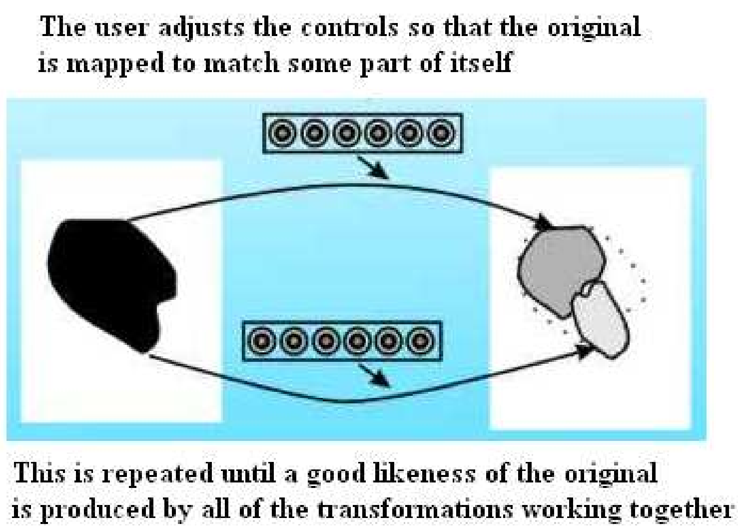


Figure – 4.84 Manual method of fractal compression

This is a workable method of compressing black and white images. As you add more transformations to the IFS then the accuracy of the representation increases. So you can trade storage for accuracy as in a traditional compression scheme.

4.4.9.6 A practical scheme

Instead of trying to find an affine transformation that maps the whole image onto part of itself you can try to find an affine transformation that maps one part of the image onto another part of the image. As most images have a high degree of affine redundancy i.e. one part of the picture is near as makes no matter an affine copy of another. This should work well and perhaps even better than the manual method described earlier.

A fully automatic approach is only possible using a fixed division of the image into so called "domain" and "range" blocks. The range blocks contain parts of the image that will be mapped to the domain blocks in an attempt to match the original image. The figure given below shows the domain-range block transformation.

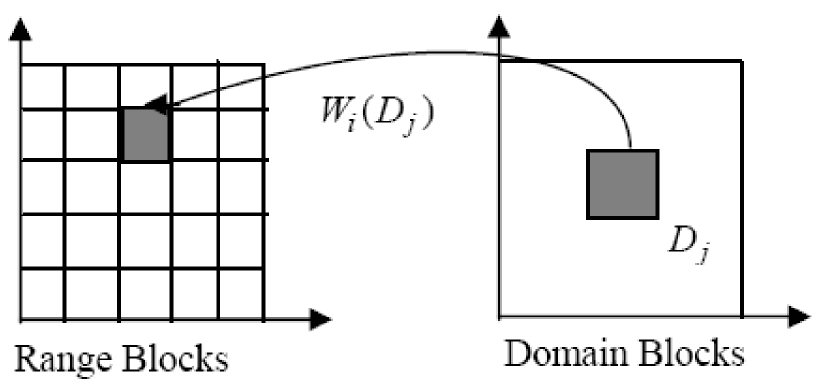


Figure – 4.85 Domains – Range Block Transformations

It is obvious that the domain blocks have to completely divide up the original image without overlap because we are interested in reconstructing the entire image. Also the range blocks have to overlap because they have to be bigger than the domain blocks.

For example, you might use the set of transformations which form the group of symmetry operations of the square:

Transform	Type
0	Identity
1	Reflection in the y axis
2	Reflection in the x axis
3	180 degree rotation
4	Reflection in 45 degree line
5	90 degree Rotation
6	270 degree Rotation
7	Reflection in -45 degree line

Table – 4.21 Types of Transform

4.4.9.7 Gray Level Compression

The contractive affine transformations and the compression methods that they give rise to have all been in terms of binary images. A grey level image can be dealt with in a number of different ways and all that really matters is that we use a suitable generalization of a contractive affine mapping.

The problem in trying to generalize the contractive affine transformations that we have been using is that they are put together, so to speak, using the idea of set union. This is possible because the black points in the image form a clearly defined set that can be modified by the affine transformations. In the case of a grey level image there is no set of points that we can concentrate on because every point in the image has a grey value.

The solution to the problem is to consider the grey level image as a function - g(x,y) that associates a grey level with each point in the image and allows the IFS to transform this function replacing set union by addition. The way that this has to be done is a little confusing at first but it makes good sense.

The brightness of the input image at x, y is given by g(x, y). The transformed image is h(x, y). To find the brightness of the transformed image at h(x', y') we have to add up the contributions of each point in the original image that has been transformed by the IFS to the point x',y'. If the IFS consist of transformations c_1, c_2 and so on, these points are:

c_1-1(x',y'), c_2-1(x',y') and so on.

Adding these brightness values together would give you a brightness value that on average will be n times brighter than the original where n is the number of transformations. To keep the grey levels in the same range all we have to do is to form a weighted sum of the brightness values. That is:

h(x',y')=p_1 g(c_1-1(x',y'))+p_2 g(c_2-1(x',y')) + and so on

Where the sum of all the p factors is one.

The transformation M(g(x,y)) generated in this way is a contractive affine transformation. If you iterate M then the result converges to an attractor and the collage theorem still operates - if M(g(x,y)) is close to g(x,y) then the attractor of M is even closer.

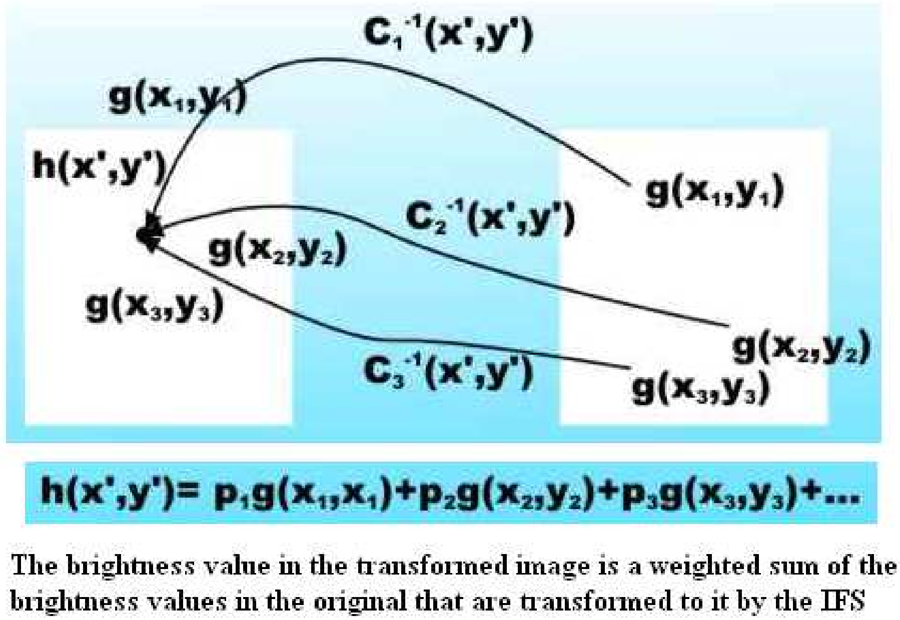


Figure – 4.86 Gray level transform

You can even use the same method of dividing up the image into range and domain blocks and looking for the best range block plus transformation to match each of the domain blocks. If this method works well with grey levels, then you can apply for color image also, just treat the color image as three grey level images one for red, one for blue and one for green.

4.4.10 Experimental Study

Fractal image compression algorithm experiment is performed on different images such as Lena and Peppers. In our experiment we performed this study up to on different panel size from 5 to 15. The following images represent the original image and compressed image.

Figure – 4.87 Original Lena and Peppers Images

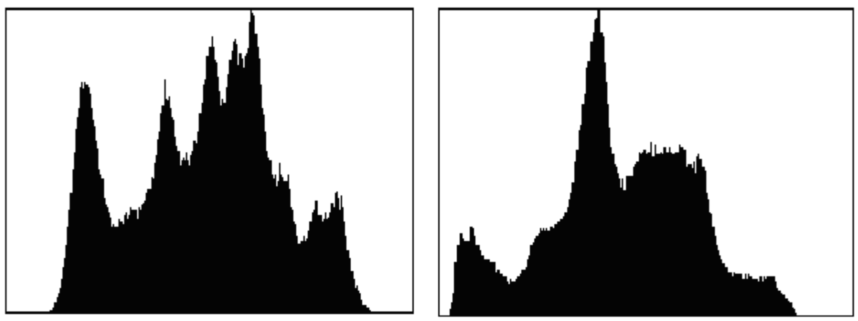


Figure – 4.88 Histogram Lena and Peppers Images

Lena		Peppers	
Mean	110.8	Mean	119.4
Median	116	Median	120
Mode	135.2	Mode	94.07
Standard Deviation	53.3	Standard Deviation	54.46
Median Absolute Deviation	36	Median Absolute Deviation	42
Mean Absolute Deviation	44.23	Mean Absolute Deviation	46.65

Table – 4.22 Statistics of Mandrill and Peppers images

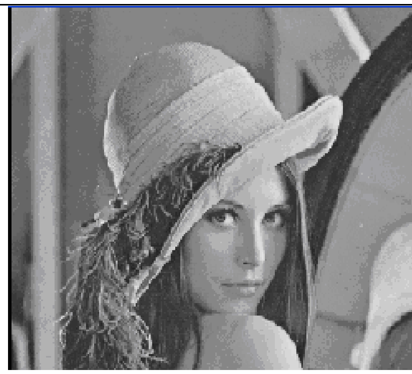
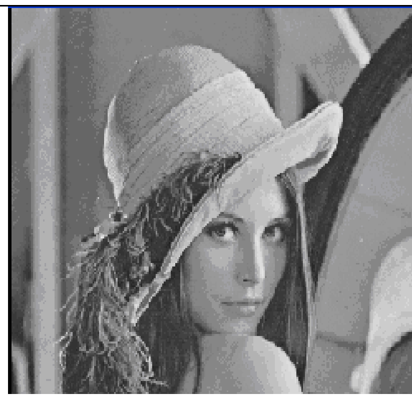
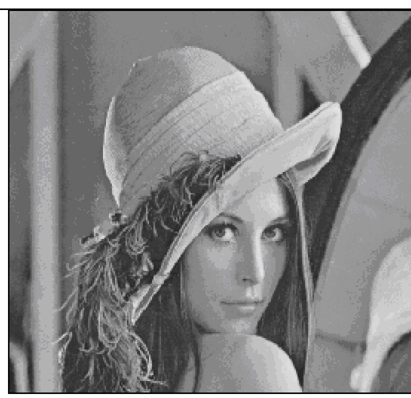
Panel Size	Lena	Peppers
5		
10		
15		

Figure – 4.89 Compressed images of Lena and Peppers

The following table represents the different images at different image quality level with original image size and compressed image size.

Panel Size	Original File (in bytes)	Compressed File (in bytes)	Compression Ratio	Processing Time (in milliseconds)
5	20401	13112	35.73	164891
6	20401	9365	54.10	161188
7	20401	6930	66.03	153922
8	20401	5512	72.98	163360
9	20401	4289	78.97	146734
10	20401	3442	83.13	134221
11	20401	2684	86.84	116609
12	20401	2511	87.69	133672
13	20401	1853	90.92	93797
14	20401	1887	90.75	128875
15	20401	1523	92.53	100313

Table – 4.23 Fractal Image Compression on Lena

Panel Size	Original File (in bytes)	Compressed File (in bytes)	Compression Ratio	Processing Time (in milliseconds)
5	248725	57228	76.99	3048016
6	248725	38570	84.49	3049834
7	248725	28833	88.41	3051750
8	248725	22859	90.81	2942594
9	248725	21763	91.25	2965445
10	248725	18455	92.58	3002922
11	248725	15869	93.62	2888341
12	248725	12511	94.97	2826984
13	248725	11243	95.48	2892432
14	248725	10372	95.83	2732492
15	248725	9651	96.12	2707827

Table – 4.24 Fractal Image Compression on Peppers

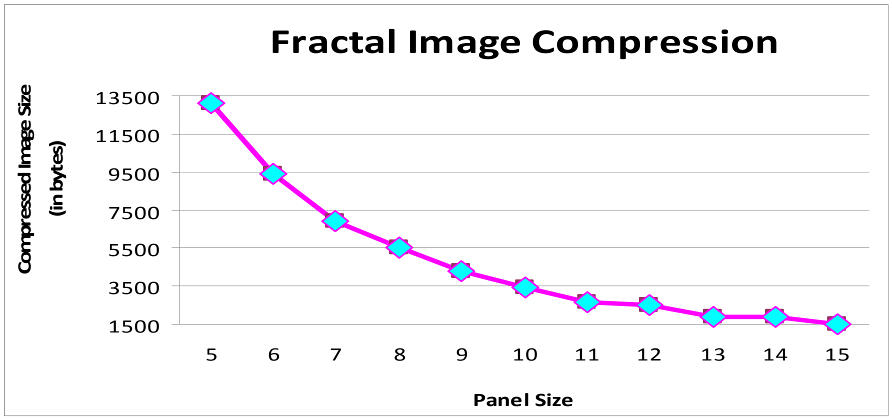


Figure – 4.90 Panel Size v/s Compressed Imag Size of Lena chart

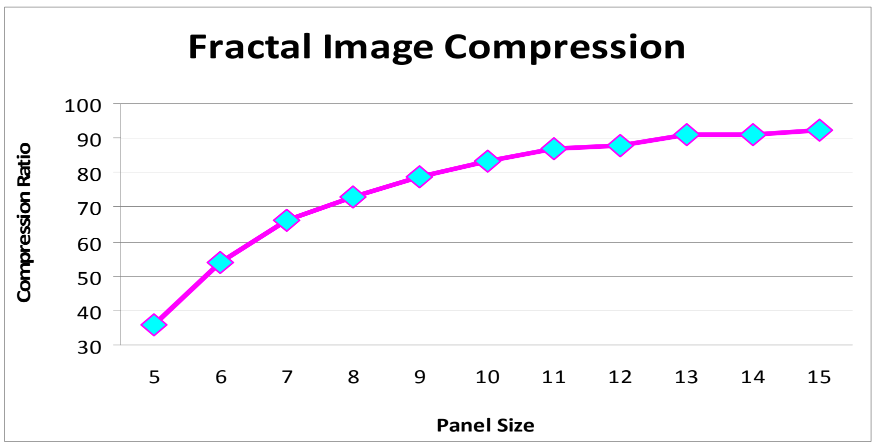


Figure – 4.91 Panel Size v/s Compression Ratio of Lena chart

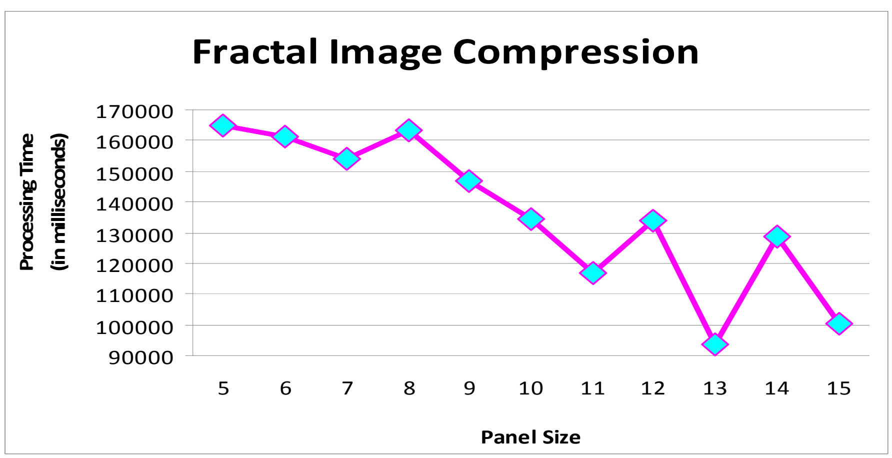


Figure – 4.92 Panel Size v/s Processing Time of Lena chart

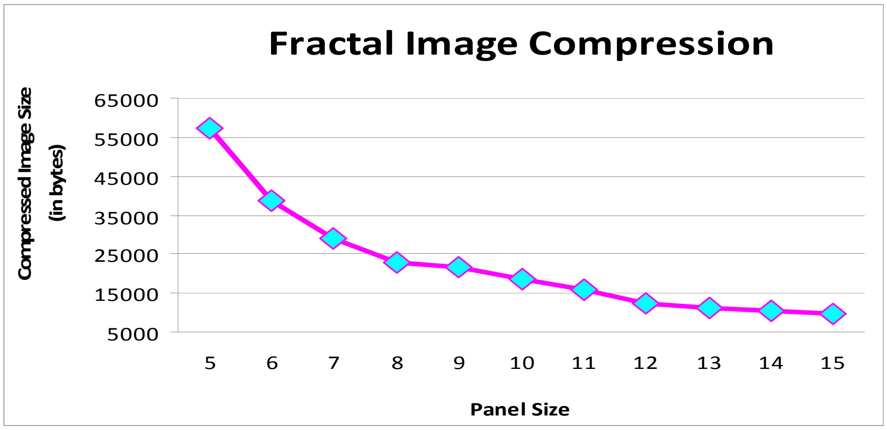


Figure – 4.93 Panel Size v/s Compressed Imag Size of Peppers chart

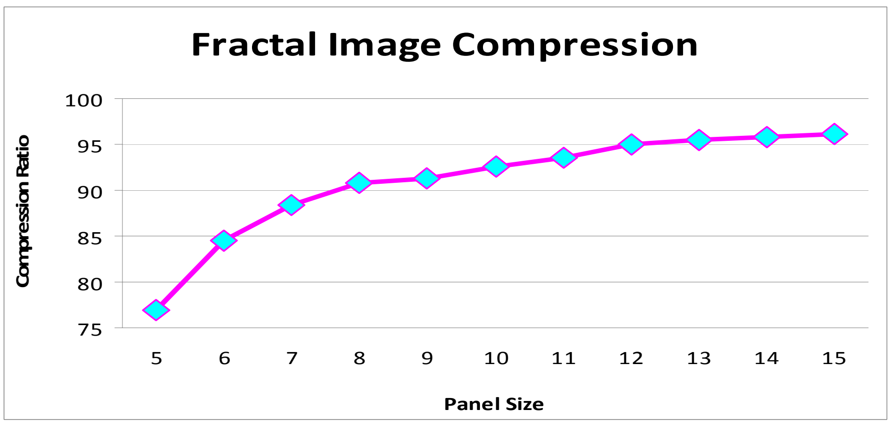


Figure – 4.94 Panel Size v/s Compression Ratio of Peppers chart

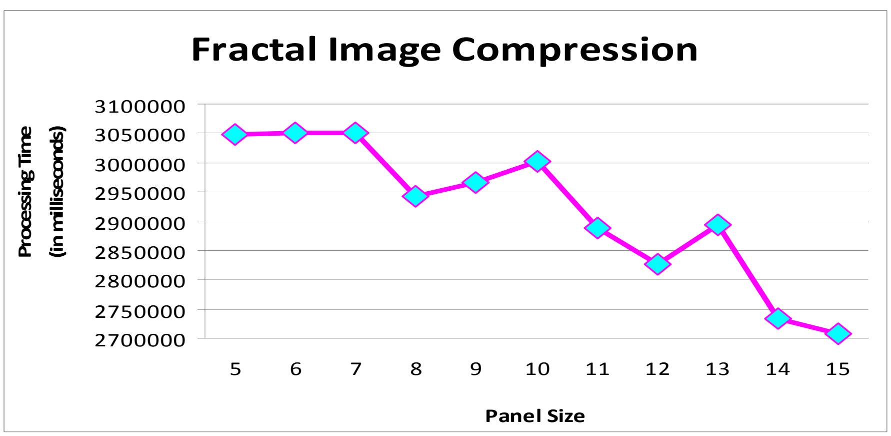


Figure – 4.95 Panel Size v/s Processing Time of Peppers chart

4.4.11 Conclusion

From the experimental study of different images at different panel size, it can be concluded that when you increase the image panel size, the compressed image size and processing time decreases, which is shown in the above given table. When you increase the image panel size, the image compression ratio increases, which is also shown in the above table.

4.5 COMPARATIVE STUDY OF IMAGE COMPRESSION ALGORITHMS

The main objective of this research work is to study the various image processing algorithms. From this various image processing, we select any one image processing for our study. So, we select image compression for the research. Earlier, we have studied different types of image compression methods.

In this work four different image compression algorithms are studied. These are Wavelet, JPEG, Vector Quantization and Fractal image compression algorithms. These algorithms are performed on different images with different properties set with the algorithms.

The table given below represents the compression ratio obtained by different image compression algorithms on different images.

Image Compression Algorithm / Method	Compression Ratio (CR)
Wavelet	>32
JPEG	<=50
VQ	<32
Fractal	>=16

Table – 4.25 Image Compression Algorithms and Compression Ratio

We have reviewed and summarized the characteristics of image compression, need of compression, principles behind compression, different classes of compression techniques and various image compression algorithms based on Wavelet, JPEG/DCT, VQ, and Fractal approaches. Experiments are performed on different images such as Lena, Peppers, Mandrill, etc…

From this experimental study, we hereby present some conclusions as follow:

- These all four algorithms give satisfactory result for 0.5 bits per pixel (bpp).
- Embedded Zero tree Wavelet (EZW) algorithms is superior to all other algorithm for very low bit rate i.e. 0.25 bpp or lower.

There are some advantages and disadvantages of each image compression algorithms. The table given below shows the advantages and disadvantages of different image compression algorithm.

Algorithm / Method	Advantages	Disadvantages
Wavelet	➢ High Compression Ratio ➢ Stage of the Art	➢ Coefficient Quantization ➢ Bit Allocation
JPEG	➢ Current Standard	➢ Coefficient Quantization ➢ Bit Allocation
VQ	➢ Simple Decoder ➢ No Coefficient Quantization	➢ Slow in Codebook Generation ➢ Small bpp
Fractal	➢ Mathematical Encoding Frame is Good ➢ Resolution Free Decoding	➢ Slow Encoding ➢ Bit Allocation

Table – 4.26 Characteristics of Image Compression Algorithms

For practical approaches to all image compression algorithms, we conclude as follow:

- Wavelet based image compression algorithms are strongly recommended.
- JPEG/DCT based image compression algorithms might use an adaptive quantization table.
- VQ based image compression algorithms are not appropriate for a low bit rate compression although they are simple.
- Fractal based image compression algorithms should utilize their resolution free decoding property for a low bit rate compression.

CHAPTER – 5

IMAGE COMPRESSION MODEL

5.1 OVERVIEW

In this chapter of the thesis, the aim is to represent a model for image compression algorithm. In previous chapters of the thesis, we performed a study of image processing fundamentals. It also discusses some process to be performed on images such as image compression, image registration, image segmentation, etc. From this processes, we selected the image compression to perform our detailed study. We performed our experimental study on four different image compression algorithms in different criteria, at different value. These algorithms are:

1. Wavelet Compression
2. JPEG Compression
3. Vector Quantization Compression
4. Fractal Compression

5.2 SCREEN LAYOUT

The figure given below shows the screen layout of model application. It contains following different menus, which is shown in the figure given below:

- File
- Edit
- View
- Tool
- ImgC&ImgQ

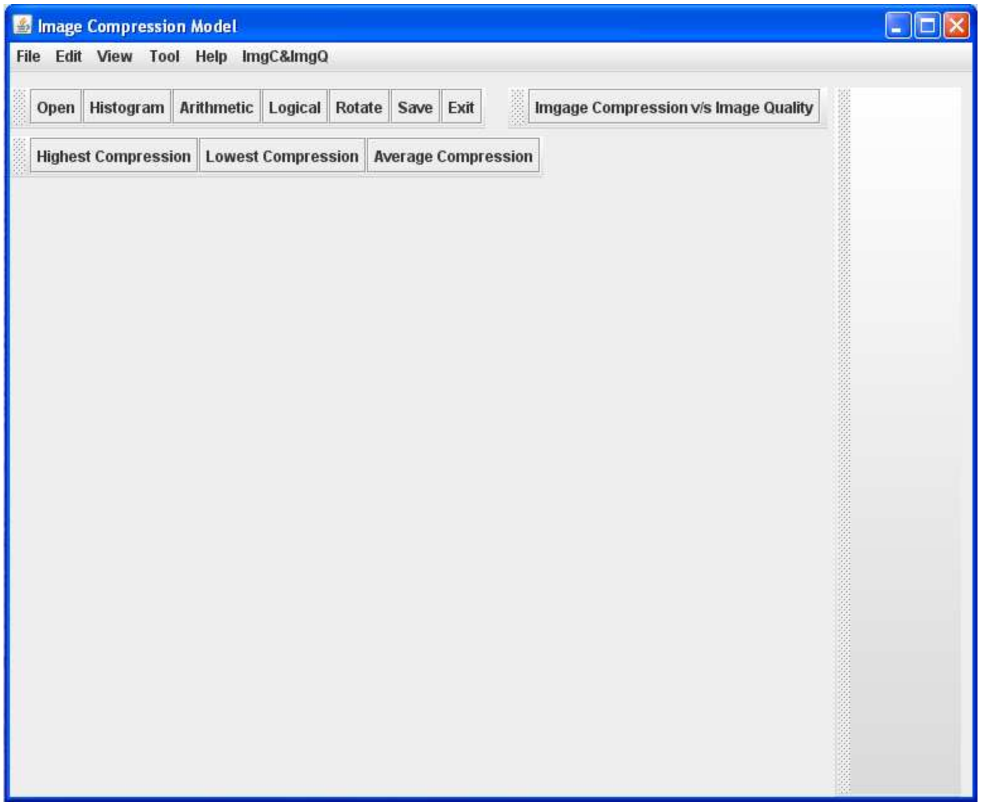


Figure – 5.1 Main Screen of Application

File menu contains the following options:

- **New: -** It is used to open new screen for image processing.
- **Open: -** It is used to open an image in screen. It opens the open dialog box for opening an image.
- **Open from URL: -** It is used to open an image from the URL in screen. It opens the dialog box for stating URL.
- **Save: -** It is used to save the present image file on the screen.
- **Save As: -** It is used to save a copy of open image file from the screen.
- **Exit: -** It is used to close the application.

Edit menu contains the following options:

- **Undo: -** It is used to undo the process.
- **Redo: -** It is used to redo the process.
- **Copy: -** It is used to copy the contents.
- **Cut: -** It is used to cut the contents.
- **Paste: -** It is used to paste the copied/cut contents.

View menu contains the following options:

- **Main Toolbar: -** It is used to display the main toolbar.
- **Compression Toolbar: -** It is used to display the compression toolbar.

Tool menu contains the following options:

- **Rotation: -** It is used to rotate the image. It rotates image at different angle such as clock wise 90^o and 180^o. It also rotates anticlock wise 90^o and 180^o. It also rotates a custom angle. User is allowed to enter the angle that he/she wishes.
- **Scatch: -** It is used to draw the scatch of open image in the screen.
- **Gray Scale: -** It is used to convert the image into gray scale image.
- **Image Negative: -** It is used to create a negative of open image in the screen.
- **Arithmetic Operation: -** It is used to perform the arithmetic operation on images. The arithmetic operations are addition, subtraction, multiplication and division of two or more images.
- **Logical Operation: -** It is used to perform the logical operation on images. The logical operations are AND,

OR and NOT images.

- **IC v/s IQ: -** It is used to show the image compression v/s image quality. It is used to allow the user to set the percentage of image compression and with respect to it shows the quality of the image.
- **Histogram Equalization: -** It is used to draw the histogram of the image.

ImgC&ImgQ menu contains the following options:

- **Highest Compression & Lowest Quality: -** It is used to perform the highest level compression of image. Here we set the 100% compression to the highest level compression.
- **Lowest Compression & Highest Quality: -** It is used to perform the lowest level compression of image. Here we set the 10% compression to the lowest level compression.
- **Average Compression & Maintain Quality: -** It is used to perform the average compression of image. Here we set the 50% compression to the average compression.

5.3 EXPERIMENT

In this testing part of the image compression, different types of image are used for testing. Here both gray scale and color images are tested. Many operations are to be performed such as copy, cut, paste, gray scale, negative, arithmetic, logical, rotation, scatch, histogram, resize image, if required.

The main aim of this work is image compression. So here we focus on image compression. To compress the image, here we set the three levels: lowest compression, average compression and highest compression. For lowest level compression, we set 10% compression of image. For average compression, we set 50% compression of image. For highest level compression, we set 100% compression of image.

5.3.1 Gray Scale Image Experiment

In this part of the thesis, the image compression operation is performed on gray scale image. Here we perform experimental study on Barbara, Boat, Goldhill and Girl images. The figure given below shows the compression of Barbara image at lowest, average and highest level compression.

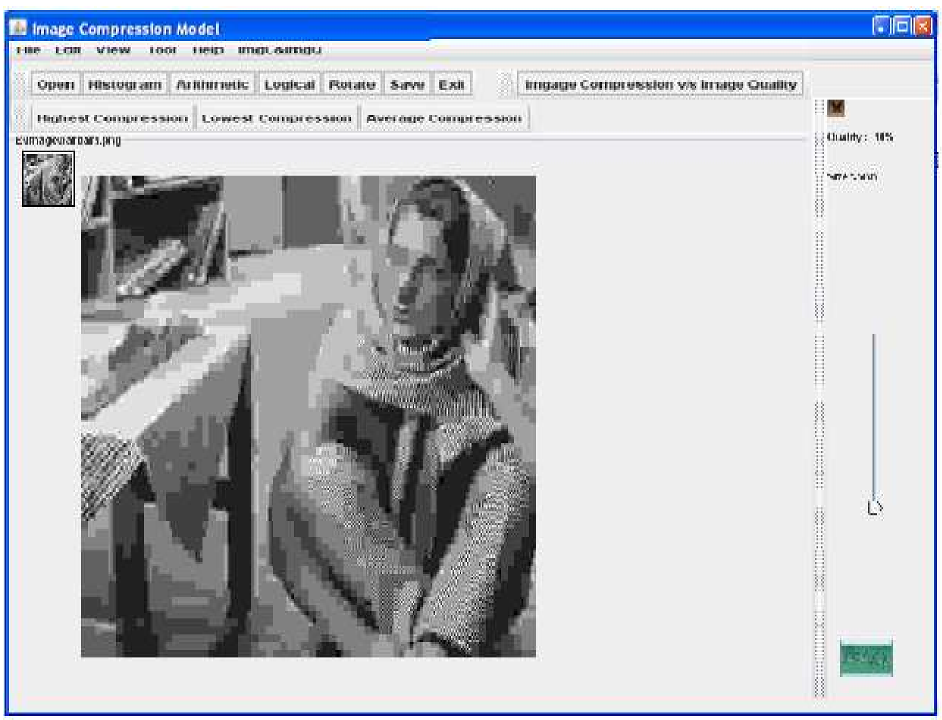


Figure – 5.2 Lowest Level Compression of Barbara Image

Figure – 5.3 Average Compression of Barbara Image

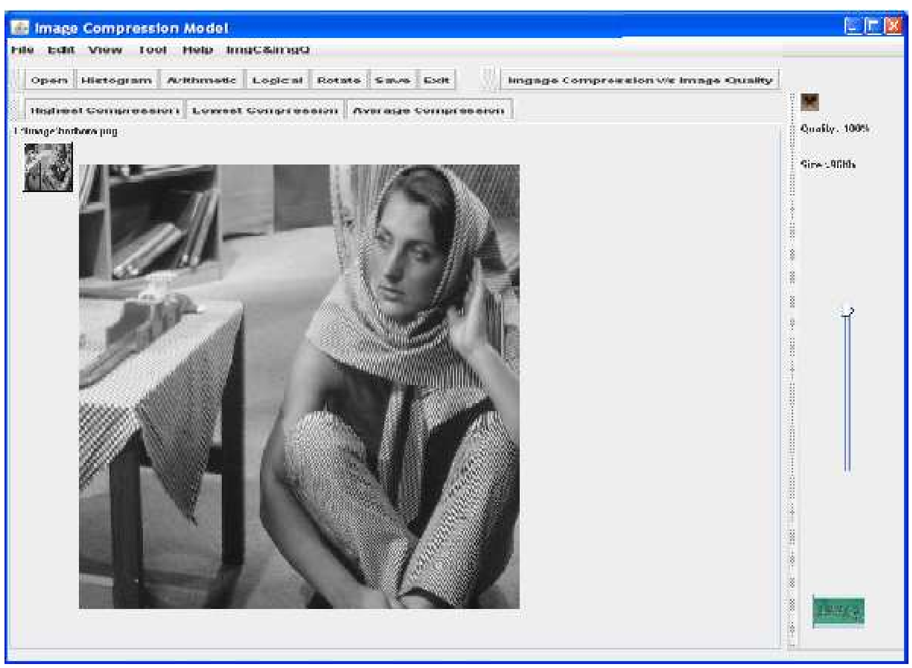


Figure – 5.4 Highest Level Compression of Barbara Image

5.3.2 Color Image Experiment

In this part of the thesis, the image compression operation is performed on color image. Here we perform experimental study on Peppers, Lena, Mandrill and Fruits images. The figure given below shows the compression of Peppers image at lowest, average and highest level compression.

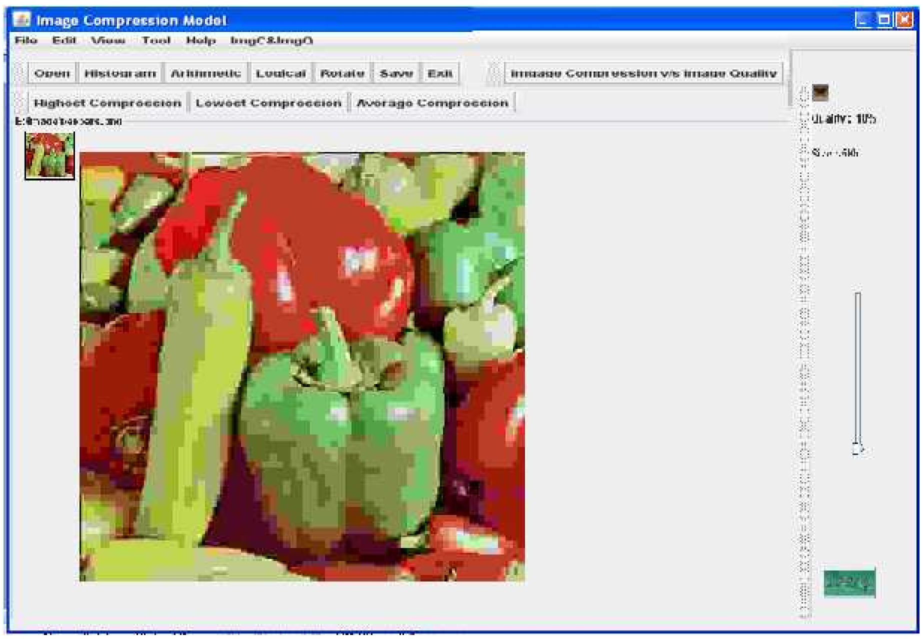


Figure – 5.5 Lowest Level Compression of Peppers Image

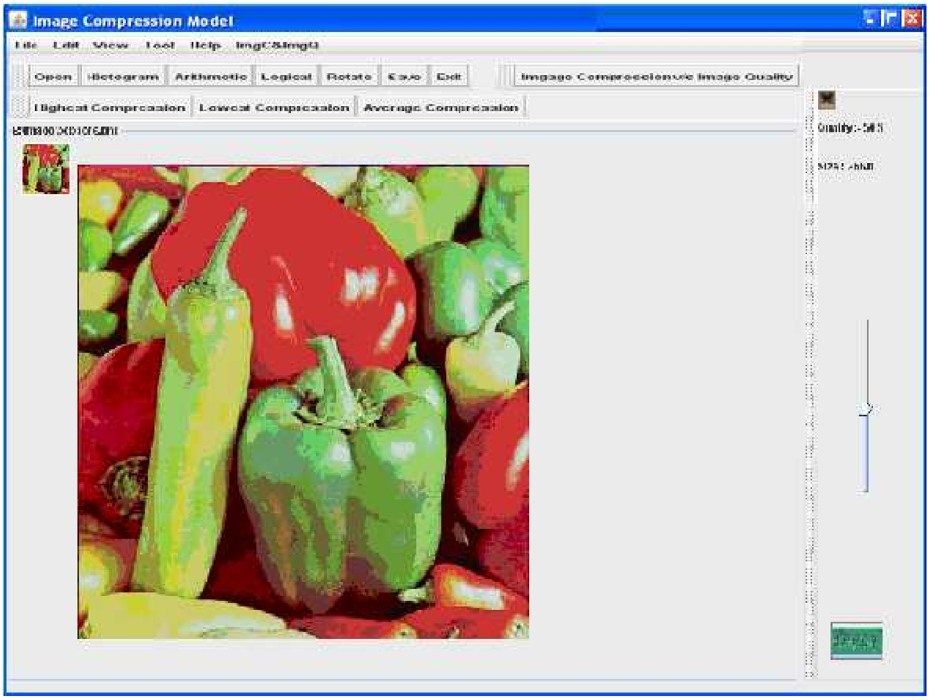


Figure – 5.6 Average Compression of Peppers Image

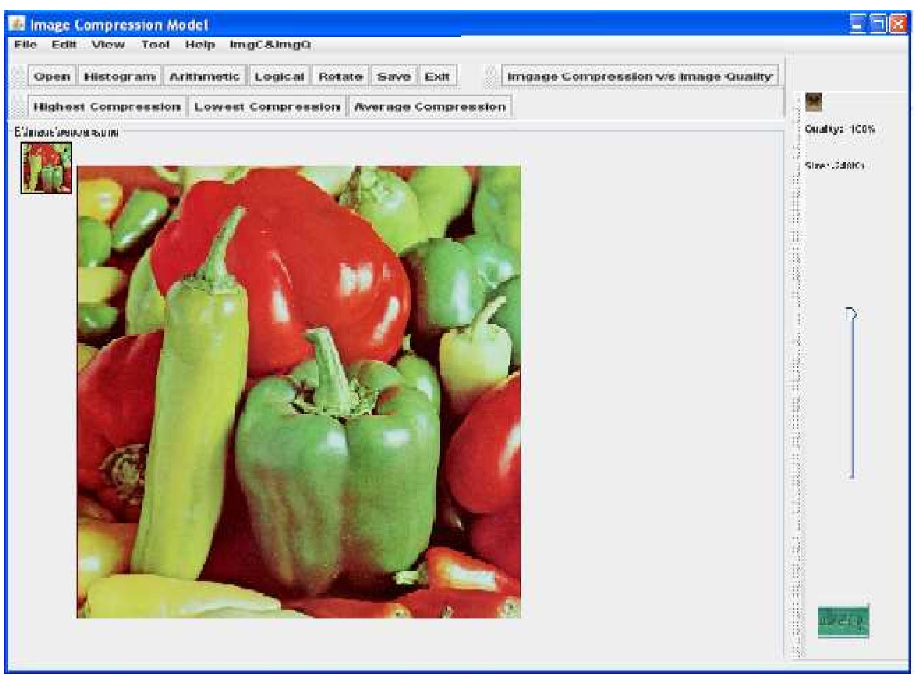


Figure – 5.7 Highest Level Compression of Peppers Image

5.4 RESULT

In this part of thesis, an experiment is performed in both gray scale images and color images.

5.4.1 Gray Scale Image

The table given below shows the result of experimental study on gray scale images. In this study four images are used such as Barbara, Boar, Goldhill and Girl.

Image	Original Image Size(KB)	Low Compression 10% (KB)	Average Compression 50% (KB)	High Compression 100% (KB)
Barbara	182	5	30	96
Boat	174	4	27	101
Goldhill	169	4	27	99
Girl	136	5	6	10

Table – 5.1 Original and Compressed gray scale image result

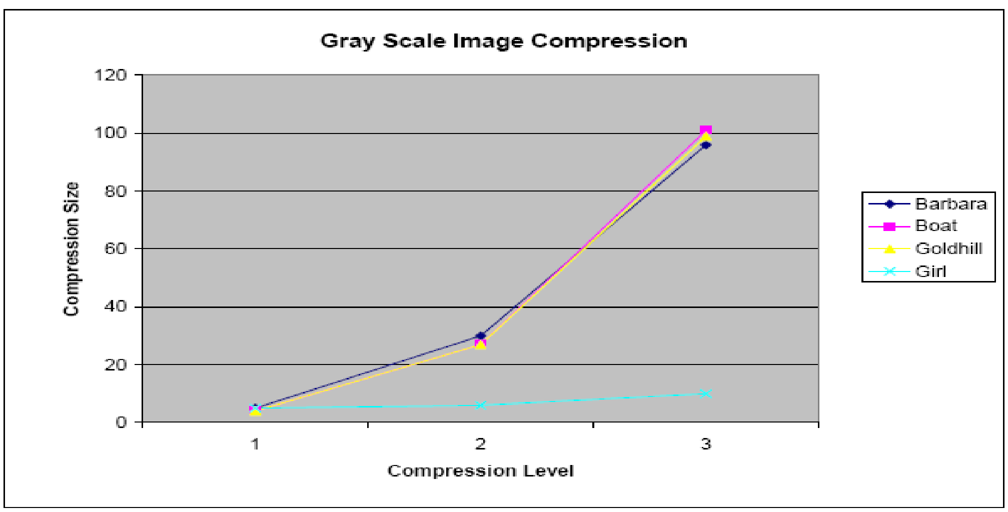


Figure – 5.8 Gray Scale Image Compressions

5.4.2 Color Image

The table given below shows the result of experimental study on color images. In this study four images are used such as Peppers, Lena, Mandrill and Fruits.

Image	Original Image Size(KB)	Low Compression 10% (KB)	Average Compression 50% (KB)	High Compression 100% (KB)
Peppers	527	5	9	44
Lena	501	5	24	96
Mandrill	335	6	50	164
Fruits	462	5	25	94

Table – 5.2 Original and Compressed color image result

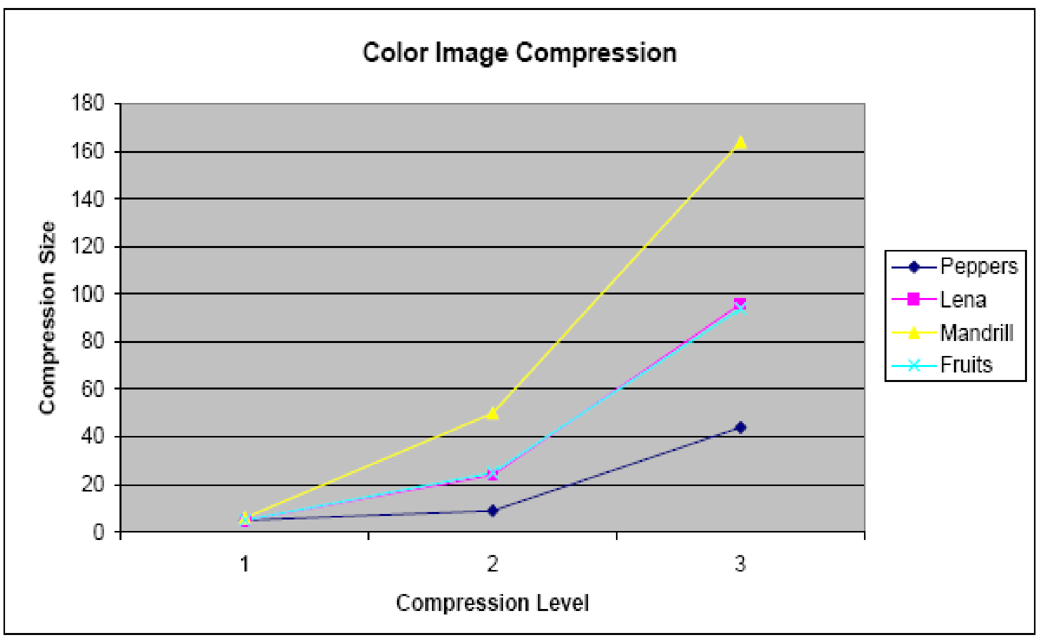


Figure – 5.9 Color Image Compressions

5.5 DISCUSSION

In this part of the thesis, we discuss the result of various compressed images. Both gray scale and color image are used to perform the experiment. Gray scale images Barbara, Boat, Goldhill and Girl as well as color images Peppers, Lena, Mandrill and Fruits are selected for experiment.

In image compression, here we set the three levels: lowest compression, average compression and highest compression. We also set any level of percentage. For lowest level compression, we set 10% compression of image and it gives low quality of image. For average compression, we set 50% compression of image and it gives average quality of image. For highest level compression, we set 100% compression of image and it gives high quality of image.

CHAPTER – 6

CONTRIBUTIONS OF THE RESEARCH WORK AND SUGGESTED FUTURE WORK

6.1 SUMMARY OF WORK

This research work is performed on various image processing techniques. It has many general goals and objectives. To achieve the goals and fulfill the objectives researcher has performed much work on this area.

In this research, first of all, we studied various types of images and image represents formats. Different types of image represent formats are used in different application. This work studies various image characteristics of different images.

In this work various image processing techniques are studied such as image compression, image registration, image segmentation, etc.. From these different techniques, the researcher has selected one technique i.e. image compression. So, the researcher has carried out his work on only image compression process.

For image compression, the researcher studied different types of image compression methods. Generally there are two different types of image compression method. These are Lossless and Lossy image compression. In lossless image compression method, the original image can be recovered from the compressed image. That means one can recover the original image from compressed image. There is no data lost from the image. So it is called lossless image compression. Lossless image compression has lower compression ratio. In lossy image compression method, it has less ability to recover original image from the compressed image. That means one cannot exactly recover the original image from compressed image. There may be data lost from the image. So, it is called lossy image compression.

In this research work four different image compression algorithms are studied. These are Wavelet, JPEG, Vector Quantization and Fractal image compression algorithms. These algorithms are performed on different images with different properties set with the algorithms.

The wavelet based image compression algorithm was performed on different five different algorithms such as: Haar, Daubechies, Biorthogonal, Symlets and Coiflet. It was performed on different decomposition level. Finally, we compared these image compression algorithms.

6.2 CONCLUSION

The research work started with the objectives of study, and modelling for image processing algorithms. In this research, the study of the various image processing algorithms was carried out. In our study, we selected image compression process because there are many difficulties and problems for image storage and transmission over network or internet.

The outcome or result of this research work provides the help to select the image compression algorithm. Image compression is most useful because there are many problems related to storage image data and transmission of image data over network. There are various image compression algorithms available. To select appropriate algorithm for image compression, this study is very useful. This work is also useful to develop a new image compression algorithm. These different image compression algorithms are suitable for different situations.

In this work, conclusion is written after every image compression algorithm. In this thesis, a comparison of these all image compression algorithms has been provided.

In this work, four different image compression algorithms are studied. These are Wavelet, JPEG, Vector Quantization and Fractal image compression algorithms. These algorithms are performed on different images with different properties set with the algorithms.

Each of image compression algorithms has some advantages and disadvantages.

The advantage of wavelet based image compression algorithm is that it gives high compression ratio. The disadvantages of wavelet based image compression algorithms are bit allocation and coefficient quantization.

The advantage of JPEG based image compression algorithm is that it can be used for current standard. The disadvantages of JPEG based image compression algorithms are bit allocation and coefficient quantization.

The advantage of VQ based image compression algorithm is that it provides simple decoder and it has no coefficient quantization. The disadvantage of VQ based image compression algorithm is that it gives good result only for small bpp and it has slow process of codebook generation.

The advantage of fractal based image compression algorithm is that it good for mathematical encoding frame and it provides resolution free decoding. The disadvantages of fractal based image compression algorithms are bit allocation and its slow pace for encoding process.

In this work, it can be concluded that, these all four algorithms give satisfactory result for 0.5 bits per pixel (bpp). Embedded Zero tree Wavelet (EZW) algorithms is superior to all other algorithms for very low bit rate i.e. 0.25 bpp or lower.

In practical approaches to all image compression algorithms, it can be concluded that:

(1) Wavelet based image compression algorithms are strongly recommended.

(2) JPEG/DCT based image compression algorithms might use an adaptive quantization table.

(3) VQ based image compression algorithms are not appropriate for a low bit rate compression although they are simple.

(4) Fractal based image compression algorithms should utilize their resolution free decoding property for a low bit rate compression.

6.3 FUTURE WORK

In this study, we selected some limited numbers of images. In future, we can extend this work for many different images for different characteristics.

In this study, we selected only one operation i.e. image compression. In future, we can extend it for many different operations such as image registration, image restoration, image fusion, image enhancement, image segmentation in image processing area.

In this research work, we selected four different image compression algorithms such as Wavelet, JPEG, Vector Quantization and Fractal. In future work, we can extend many different image compression algorithms.

To sum up, it can be stated that this work, provides a comprehensive and detailed analysis and modelling of various image processing algorithms. The resultant output is efficient and positively useful for future researchers in this area.

6.4 BIBLIOGRAPHY

[1] "Research Design: Qualitative, Quantitative, and Mixed Method Approaches" by J. W. Creswell, 2nd edition, Sage Publications Inc, 2003.

[2] "On the Selection of Image Compression Algorithms" by Chaur-Chin Chen (2009), Department of Computer Science National Tsing Hua University Hsinchu 300, Taiwan.

[3] "A DCT-domain System for Hiding Fractal Compressed Images" by Chin-Chen Chang, Chi-Lung Chiang and Ju-Yuan Hsiao(2005), Proceedings of the 19th International Conference on Advanced Information Networking and Applications.

[4] "A Review of Image Compression and Comparison of its Algorithms" by Sachin Dhawan Kurukshetra,

Haryana, India, May (2011), Dept. of ECE, UIET, Kurukshetra University.

[5] "A Secure Image Steganography using LSB, DCT and Compression Techniques on Raw Images" by K.B.Raja, C.R.Chowdary, Venugopal K R, L.M.Patnaik (2005), International conference on Image and Signal Processing, 2005.N.F.,

[6] "Blind consistency based steganography for information hiding in digital media" by Hairong Qi (2002), International Conference on Images Steganography.

[7] "A Secure Stegonographic Algorithm using LSB, DCT and Image Compression on Raw Images" by Niels Provos, (2004), Technical Report, Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University.

[8] "A survey of steganographic techniques, Information Hiding" by N. F. Johnson and S. Katzenbeisser. (2008), Artech House, Norwood.

[9] "Comparison of different image compression formats", ECE 533 Project Report, Paula Aguilera.

[10] "Digital Image Processing using Local Segmentation" by Torsten Seemann, School of Computer Science and Software Engineering, Faculty of Information Technology, Monash University, Australia, April 2002.

[11] "MIR: An approach to robust clustering, application to range segmentation" by Klaus Koster and Michael

Spann, IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(5):430.444, May 2000.

[12] "Digital Image Processing" by Rafael C. Gonzalez and Richard E. Woods. Addison-Wesley, 3rd edition, 1992.

[13] "Newgen Image Processing Techniques", chapter-3 image processing techniques.

[14] "An Improved Multi-Stage Method for Medical Image Registration Based on Mutual Information" by Maryam Zibaeifard, Mohammad Rahmati, Department of Computer Engineering and IT, Amirkabir University of Technology, Tehran, Iran.

[15] "Image Registration Techniques:An overview" by Medha V. Wyawahare, Dr. Predeep M. Patil and Hemant K. Abhyankar, International Journal of Signal Processing and Pattern Recognization, Vol-2, No.-3, 2009.

[16] "Investigating Polynomial Fitting Schemes for Image Compression" by S. Ameer, 2009.

[17] "Digital Image Processing Methods" by E. R. Dougherty, Marcel Dekker: CRC Press, 1994.

[18] "Fast Texture Synthesis Using Tree-Structured Vector Quantization", in Proceedings of SIGGRAPH 2000, by Li-YiWei, and M. Levoy, pp. 479–488.

[19] "Image Compression through Optimized Linear Mapping and Parametrically Generated Features" by S. Ameer and O. Basir, Malta, 2008, pp. 12-14.

[20] "Image compression using plane fitting with inter-

block prediction" in Image Vision Comput" by S. Ameer and O. Basir, 2009, pp. 385-390.

[21] "Digital Signal Processing: Principles, Algorithms and Applications" by P. J. G. and D. G. Manolakis, 3rd edition, Upper Saddle River, Prentice-Hall, 1996.

[22] "Simplified segmentation for compound image compression" by A. Said and A. Drukarev, in Proceeding of ICIP, 1999, pp. 229-233.

[23] "Simplified segmentation for compound image compression" by A. Said and A. I. Drukarev, in Proc. IEEE Int. Conf. Image Processing, Oct. 1999, pp. 24-28.

[24] "Multi-dimensional least-square polynomial curve fitting" by F. H. Lesh, vol. 2, Communications of ACM, 1959, pp. 29-30.

[25] "On Image Compression Using Curve Fitting", by Amar Majeed Butt, Rana Asif Sattar, School of Computing, Blekinge Institute of Technology, SE – 371 79 Karlskrona, Sweden.

[26] "Image Compression Using Discrete Cosine Transform & Discrete Wavelet Transform" by Bhawna Gautam, Department of Computer Science and Engineering, National Institute of Technology, Rourkela, May, 2010.

[27] "Image compression" by Andrian Brown, Digital Preservation Guidance Note 5: Reference-DPGN-05, Issue-1, July 2003.

[28] "Image Compression" DPGN-05, by Adrian Brown,

Digital Archives Analyst, Issue-1, July 2003.

[29] “The JPEG Image Compression Algorithm” by Damon Finell, Dina Yacoub, Mark Harmon.

[30] “Digital Image and Video Compression Techniques”, Research Project, EDC385G: Multimedia Authoring, Dr. Min Liu, October 29, 2003, Shin-Hyun Ka.

[31] “Wavelet theory, ECG processing using wavelets”, International Journal of Computers & Distributed Systems Volume 1, Issue 2, August, 2012.

[32] "Wavelet-based medical image compression" by E. Kofidisi, N.Kolokotronis, A. Vassilarakou, S. Theodoridis and D. Cavouras, Future Generation Computer Systems, vol. 15, no. 2, pp. 223-243, Mar. 1999.

[33] “Wavelet Transform” by Y. Sheng, In: The transforms and applications handbook. Ed. by A. D. Poularikas. P. 747-827. Boca Raton, Fl (USA): CRC Press, 1996.

[34] “Wavelet Transforms That Map Integers To Integers” by Calderbank, A. R. and I. Daubechies, W. Sweldens, B.-L. Yeo, Proceedings of the IEEE Conference on Image Processing. Preprint, 1996. IEEE Press, 1997.

[35] “Introduction To Wavelets And Wavelet Transforms - A Primer” by Burrus, C. S. and R. A. Gopinath, H. Guo, Upper Saddle River, NJ (USA): Prentice Hall, 1998.

[36] “Ten Lectures on Wavelets” by Daubechies, Ingrid, 2nd edition, CBMS 61, SIAM Press, Philadelphia PA, 1992.

[37] "Correlation between objective and subjective measurements for video compressed systems" by M. Ardito and M. Visca, SMPTE J., pp. 768–773, Dec. 1996.

[38] "Transmitted-Picture Assessment" by J. Allnatt, New York: Wiley, 1983.

[39] "Objective picture quality scale (PQS) for image coding" by M. Miyahara, K. Kotani, and V. R. Algazi. (1996)[Online]Available:http://info.cipic.ucdavis.edu/scripts/reportPage?96-12.

[40] "Methods for the subjective assessment of the quality of television pictures," ITU, Geneva, Switzerland, ITU-R Rec. BT. 500-7, Aug. 1998.

[41] "An Algorithm for Image Compression Using 2D Wavelet Transform" by Rohit Arora, Department of Electronics and Communication Engineering, Punjab Technical University, Punjab, India. International Journal of Engineering Science and Technology (IJEST).

[42] "Image Denoising In The Wavelet Domain Using Wiener Filtering" by Nevine Jacob and Aline Martin, December 2004.

[43] "Image Wavelet Compression Implementation Using A Run-Time Reconfigurable Custom Computing Machine" by Zhimei Ding, Master Of Science In Electrical Engineering, Virginia Polytechnic Institute and State University, June 2000.

[44] "The Use of Wavelet Filters to De-noise µPET Data",

by Joe Grudzinski, Department of Biomedical Engineering, December 2006.

[45] “Image Processing: Principles and Applications” by T. Acharya and A. K. Ray, Hoboken, NJ: John Wiley & Sons, 2005.

[46] “The JPEG Still Picture Compression Standard” by Gregory K. Wallace, Maynard, December 1991.

[47] “A Robust Image Compression Algorithm using JPEG 2000 Standard with Golomb Rice Coding” by Shaik. Mahaboob Basha and Dr. B.C. Jinaga, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.12, December 2010.

[48] “Digital Photography and Image Compression on the Arm Microcontroller”, Department Of Electronic Engineering, National University of Ireland, Galway. April 2003.

[49] “Modified Vector Quantization Method for Image Compression” by K.Somasundaram, and S.Domnic.

[50] “An Algorithm for Image Compression using Differential Vector Quantization” by James E. Fowler, Jr. B. S., The Ohio State University, 1995.

[51] “Image Compression using Vector Quantization” by Nopparat Pantsaena, Manas Sangworasil, Chinnapat Nantajiwakornchai, Tanasak Phanprasit, Research Center for Communication and Information Technology (ReCCIT), King Mongkut’s Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand.

[52] "A Study on Fractal Image Compression" by Darshan S. Alagud.

[53] "An Introduction to Fractal Image Compression", Literature Number : BPRA065, Texas Instruments Europe, October 1997.

[54] "Fractal Image Compression", Siggraph 92 course notes, Y. Fisher.

[55] "Fractals for Class Room", Springer Verlag, by Peitgten, Daupe, Jurgens, New York, 1991.

[56] "A Fractal based Approach to Image compression" by Walach, E. E. Karnin, Proceedings of ICASSP, Tokio, 1986.

[57] "Fractal Image Compression" by Martin V Sewell, Department of Computer Science, Birkbeck College, University of London, 1994.

[58] "The Data compression" by M. Nelson and J. L. Gaily, second edition, New York, M&T Books, 1996.

[59] "Fractals Everywhere", by Barnsley M F, Academic Press Professional, 1993.

[60] "Fractal Image Compression Using Iterated Transforms" by Fisher,Y.,E.W.Jacobs,R.D.Boss, Kluwer Academic Publishers, Norwall, 1989.

[61] "Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding" by Jacquin, A., Doctoral Thesis, Georgia Institute of Technology,1989.

[62] "A new, fast, and efficient image codec based on set partitioning in hierarchical trees" by A. Said, W.A. Pearlman, IEEE Trans. on Circuits and Systems for Video Technology, vol. 6, 243-250, 1996.

[63] "Vector Quantization by principal component analysis" by Y.W. Chen, M.S. Thesis, National Tsing Hua University, June, 1998.

[64] "Fractal Image Compression" by M.F. Barnsley, L.P. Hurd, AK Peters, Ltd. Wellesley, Massachusetts, 1993.

[65] "Fractal Image Compression" by Miroslav Galabov, PhD, Department of Computer Systems and Technologies, University of Veliko Turnovo.

[66] "Guide to image compression", Retrieved October 2003.

[67] "Comparison of JPEG image coders" by Grgic, S. Mrak, M., & Grgic, M. Retrieved October 2003.

[68] "Image compression techniques" by Lerner, G. Retrieved October 2003.

[69] "image compression techniques" by Philips, W. Second generation, Retrieved October 2003.

[70] "Data compression techniques in image compression for multimedia systems" by Rajaravivarma, V., Lord, E., & Barker, J., Retrieved October 2003.

[71] "More on the image compression techniques" by Tracey, L., Retrieved October 2003.

[72] “Image compression techniques” by Wilson, N. A. Retrieved October 2003.

[73] “Wavelets in Medicine and Biology” by Aldroubi, Akram and Unser, Michael (editors), CRC Press, Boca Raton FL, 1996.

[74] “Wavelets : Mathematics and Applications” by Benedetto, John J. and Frazier, Michael (editors), CRC Press, Boca Raton FL, 1996.

[75] “An Introduction to Wavelets” by Chui, Charles, Academic Press, San Diego CA, 1992.

[76] “Principles of Digital Image Synthesis” by Glassner, Andrew S., Morgan Kaufmann, San Francisco CA, 1995.

[77] “Compression Technique Using DCT & Fractal Compression – A Survey” by R M Goudar & Priya Pise, Advances in Physics Theories and Applications Journal, Vol 3, 2012.

[78] “Fractals and Self Similarity” by John E. Hutchinson, Indiana University Mathematics Journal.

[79] "Fractal Image Compression" by Anson L F.

[80] “Implementation of Vector Quantization for Image Compression – A Survey”, By G. Boopathy and S. Arockiasamy, Global Journal of Computer Science and Technology, Vol-10, Issue-3, April 2010.

[81] “Introduction to Medical Image Compression using Wavelet Transform” Time Frequency Analysis and Wavelet Transform Term Paper, National Taiwan

University, December 2007.

[82] "Wavelet based Image Compression" by James S. Walker, Sub-chapter of CRC Press Book: Transforms and Data Compression, University of Wisconsin-Eau Claire.

[83] "Still Image Compression Standards", Module-6, Ver-2, ECE IIT, Kharagpur.

[84] "The JPEG Still Picture Compression Standard" by Gregory K. Wallace, Maynard, December 1991.

[85] "JPEG Image Compression" by Rahul Garg and Varun Gulshan, December 2005.

[86] "JPEG Image Compression using an FPGA" by James Rosenthal, University of California, December 2006.

[87] "JPEG Compression" by Ben O'Hanen and Matthew Wisan, December 2005.

[88] "The Evolution of Evaluation for Image Segmentation" by Yu-Jin ZHANG, Tsinghua University, Beijing, China.

[89] "Genetic Algorithm Applied to Fractal Image Compression" by Y. Chakrapani and K. Soundara Rajan, ARPN Journal of Engineering and Applied Sciences, Vol- 4, No-1, February 2009.

[90] "Fundamentals of Image Processing" by Ian T. Young, Jan J. Gerbrands, Lucas J. van Vliet, Delft University of Technology.

[91] "Two and Three Dimensional Coding Schemes for

Wavelet and Fractal-Wavelet Image Compression" by Simon K. Alexander, University of Waterloo, Canada, 2001.

[92] "A Comparative Study of Image Compression Method" by Ashwin Swaminathan, Gaurav Agarwal.

[93] "Comparative Study and Implementation of Image Processing Techniques Using MATLAB" by Sukhjinder Singh, R. K. Bansal and Savina Bansal, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 3, March 2012.

[94] "Digital Image Processing using MATLAB" by Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins.

[95] "The mathematical Foundation of Image Compression" by Lisa A. Soberano, May 2000.

[96] "A study of Fractal Image Compression" by Darshan S., Project Final Report, Alagud, 2010.

[97] "Analysis of Image Compression Algorithms" by Abhishek Kaushik, Maneesha Gupta, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2, Mar-Apr 2012.

[98] "A Comparative Study: Block Truncating Coding, Wavelet and Fractal Image Compression", by Dinesh Gupta, Pardeep Singh, Nivedita, Sugandha Sharma.

[99] "Comparative Study of Different Approaches of Image

Compression" by Mrs. Harsha D.Zope, Prof. Mrs. Pallavi Patil, International Journal of Emerging Technology and Advanced Engineering, Website: www.ijetae.com ISSN 2250-2459, Volume 2, Issue 1, January 2012.

[100] "A Comparative Study of Image Compression Algorithms", by Kiran Bindu, Anita Ganpati, Aman Kumar Sharma, International Journal of Research in Computer Science eISSN 2249-8265 Volume 2 Issue 5, 2012.

[101] "Comparative Performance Analysis of Haar, Symlets and Bior wavelets on Image compression using Discrete Wavelet Transform", by Jashanbir Singh Kaleka and Reecha Sharma, Council for Innovative Research International Journal of Computers & Distributed Systems www.cirworld.com Volume 1, Issue 2, August, 2012.

[102] "Introduction to Data Compression" by Guy E. Blelloch, Carnegie Mellon University, September 2010.

[103] http://ftp.uu.net:/graphics/jpeg/jpegsrc.v6a.tar.gz

[104] The JPEG web page: http://www.jpeg.org

[105] Wikipedia: http://wikipedia.org

[106] www.google.com

[107] www.wfu.edu

[108] www.scribd.com

[109] www.logodesignworks.com

[110] www.gifted.uconn.edu

[111] www.yanziyang.wordpress.com

[112] www.debugmode.com

[113] www.web.cs.wpi.edu

[114] www.urmoon.wordpress.com

[115] www.amara.com

[116] www.gsl.net

[117] www.physicsarchives.com

[118] www.polyvalens.com

[119] www.math.hawaii.edu

[120] www.mgasem.net

[121] www.guptalab.org

[122] http://fractal-image-compression.martinsewell.com

List of Paper Publication

[1] "Interface for Image Registration Process", International Journal of Emerging Technologies and Applications in Engineering, Technology and Sciences, Issue – 1, Volume – 4, ISSN : 0974 – 3588, Page No. : 93 – 95, January – June 2011.

[2] "Experimental Study of Fractal Image Compression Algorithm", International Journal of Computer Applications & Information Technology, Issue – 2, Volume – 1, ISSN : 2278 – 7720, Page No. : 48 – 50, August 2012.

[3] "Analytical Study of Wavelet Based Color Image Compression Method", International Index Journal, Paripex Indian Journal of Research, Volume – 1, Issue – 8, Issue, ISSN : 2250 – 1991, Page No. : 32 – 34, August – 2012. Impact Factor (2012) : 0.3208

[4] "Image Compression using Vector Quantization", International Journal of IT, Engineering and Applied Sciences Research (IJIEASR), Volume – 2, No. – 2, ISSN: 2319 – 4413, Page No. : 72 – 74, February 2013,

[5] "A Study of Wavelet Based Image Compression Algorithms", International Journal of Research in Commerce, IT & Management, ISSN : 2231 – 5756. To be Published……….

[6] "JPEG Image Compression Algorithm", International Journal of Research in Computer Application & Management, ISSN : 2231 – 1009. To be Published………

INTERNATIONAL JOURNAL OF IT, ENGINEERING AND APPLIED SCIENCES RESEARCH

ISSN: 2319-4413

Volume 2, No. 2, February 2013

i-Xplore International Research Journal Consortium
www.irjcjournals.org

Image Compression using Vector Quantization

Chetan Dudhagara, Assistant Professor, Computer Science Department, N. V. Patel College of Pure and Applied Sciences, Vallabh Vidyanagar
Dr. Kishor Atkotiya, Head, Computer Science Department, J. H. Bhalodia Woman's College, Rajkot,

ABSTRACT

A popular technique for source coding of image and speech data since 1980 is Vector Quantization. Day by day the use of multimedia, images and the other picture formats are rapidly increasing in a variety of application. It is very straight forward image compression approach. The technique of obtaining the compact representation of an image while maintaining all the necessary information without much data loss is referred to as Image Compression. VQ has the particular advantage of being able to exploit prior knowledge on the images to be compressed. In this method, codebook has to be generated before compression. It is not a universal approach since it cannot work well for different types of images. Vector Quantization has been observed as an efficient technique for image compression. There are main two components of VQ compression system: VQ Encoder and VQ Decoder.

Keywords

Multimedia, Compression, Encoder, Decoder, Vector, Quantization.

1. INTRODUCTION

Image compression using Vector Quantization (VQ) is a lossy compression technique. It is defined as mapping Q of K-dimensional Euclidean space R^k into a finite subset Y of R^k. Thus. $\boldsymbol{Q} : \boldsymbol{R^k} \rightarrow \boldsymbol{Y},$ Where y = (xi; i=1, 2..., N) is the set of reproduction vectors and N is the number of vectors in Y. Below figure shows the conceptual diagram of Vector Quantization.
A VQ is composed of two parts, encoder and decoder. An encoder will compare each input vector with every code vector in the codebook and generate index which represent the minimum distortion code vector from the input vector. A decoder takes the indexes to locate the code vector in codebook and generate the output vectors.
A codebook is the set of finite code vector for representing the input vector. The popular technique in codebook design is the Linde-Buzo-Gray (LBG) algorithm. The whole image partitioned into sub blocks and all sub blocks are used training this codebook.

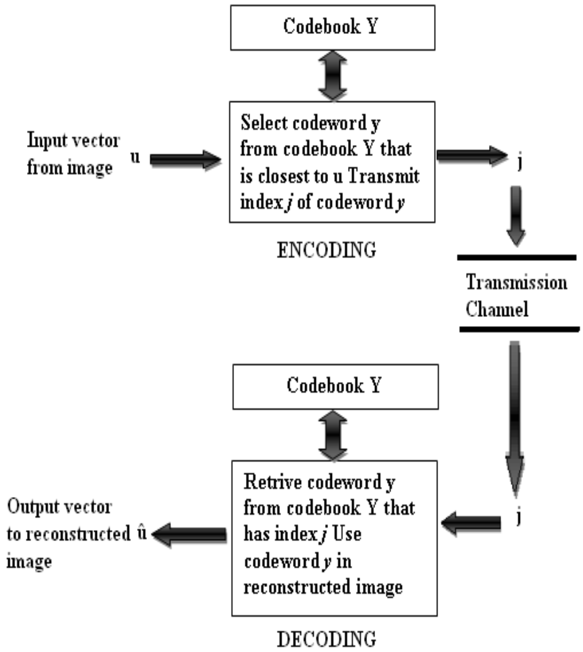


Fig-1: Conceptual Diagram of Vector Quantization

2. WORK PROCESS OF VQ

A VQ is composed of two operations: Encoder and Decoder. The encoder takes an input vector and outputs the index of the codeword that offers the lowest distortion. In this case the lowest distortion is found by evaluating the Euclidean distance between the input vector and each codeword in the codebook. Once the closest codeword is found, the index of that codeword is sent through a channel. The channel may be computer storage, communications channel, and so on. When the encoder receives the index of the codeword, it replaces the index with the associated codeword. Below figure shows a block diagram of the operation of the encoder and decoder.
Given an input vector, the closest codeword is found and the index of the codeword is sent through the channel. The decoder receives the index of the codeword, and outputs the codeword.

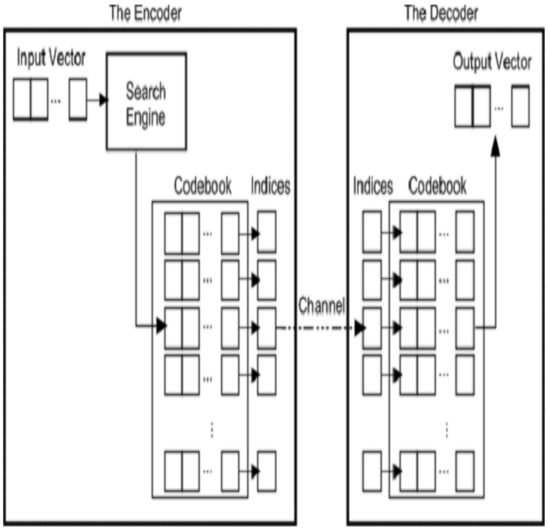


Fig-2:Encoder & Decoder in VQ

3. PERFORMANCE MEASURED IN VQ

There is no proper way to measure the performance of VQ. This is because the distortion. VQ is evaluated by us humans and that is a subjective measure. It uses Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR). MSE is defined as follows:

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(\hat{x}_i - x_i)^2$$

Where M is the number of elements in the signal or image. For example, if we want to find the MSE between the reconstructed and the original image, then we would take the difference between the two images pixel by pixel, square the results, and average the results, which is represent in above formula.

The PSNR is defined as follows:

$$PSNR = 10log_{10}\left(\frac{(2^n-1)^2}{MSE}\right)$$

Where n is the number of bits per symbol. For example, if we want to find the PSNR between two 256 gray level images, then we set n to 8 bits.

4. EXPERIMENTAL STUDY

Vector quantization image compression algorithm experiment is performed on Lena, Girl and Mandrill images. This experiment was performing on algorithm based on both random and splitting type codebook. Below table represent the experimental statistical values on different images.

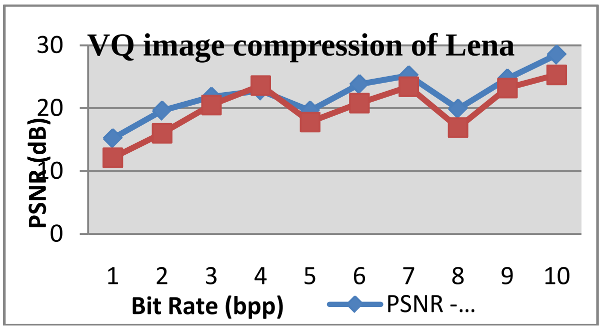
Original Image Compressed Image

Fig-3: Original and compressed Lena image

Below table shows the PSNR (db) value of both random and splitting type codebook and bit rate (bpp) of Lena image.

Table – 1: Lena image compression

PSNR -VQS	PSNR -VQR	Bit Rate
15.2	12.1	0.008
19.6	16	0.031
21.8	20.5	0.047
22.8	23.6	0.078
19.6	17.8	0.125
23.8	20.8	0.188
25.2	23.4	0.25
19.9	16.9	0.5
24.7	23.2	0.75
28.5	25.3	0.95



Fig–4: Bit Rate v/s PSNR in VQ of Lena image chart

Original Image Compressed Image

Fig–5: Original and compressed Girl image

Below table shows the PSNR (db) value of both random and splitting type codebook and bit rate (bpp) of Lena image.

Table – 2: Girl image compression

PSNR -VQS	PSNR -VQR	Bit Rate
18.8	13.2	0.008
20.4	16.8	0.031
23.7	23.6	0.047
24.6	23.8	0.078
20.1	19.2	0.125
24.2	24.7	0.188
27.5	26.2	0.25
20.4	18.4	0.5
24.2	26.1	0.75
28.2	27.8	0.95

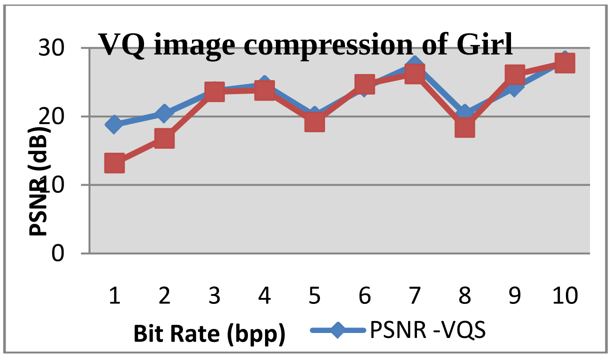


Fig-6: Bit Rate v/s PSNR in VQ of Girl image chart

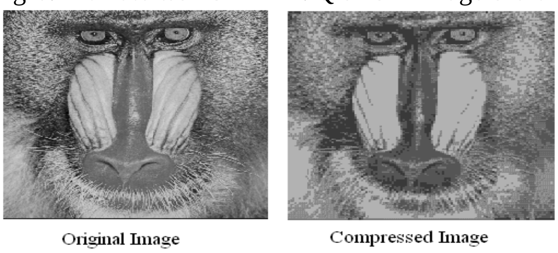


Fig-7: Original and compressed Mandrill image

Below table shows the PSNR (db) value of both random and splitting type codebook and bit rate (bpp) of Lena image.

Table – 3: Mandrill image compression result

PSNR -VQS	PSNR -VQR	Bit Rate
15.2	9.2	0.008
16.8	16.8	0.031
17.2	17.9	0.047
17.4	18.7	0.078
17.4	16.1	0.125
18.7	19.1	0.188
19.2	19.8	0.25
18.9	16.7	0.5
21.2	21.2	0.75
21.9	23.4	0.95

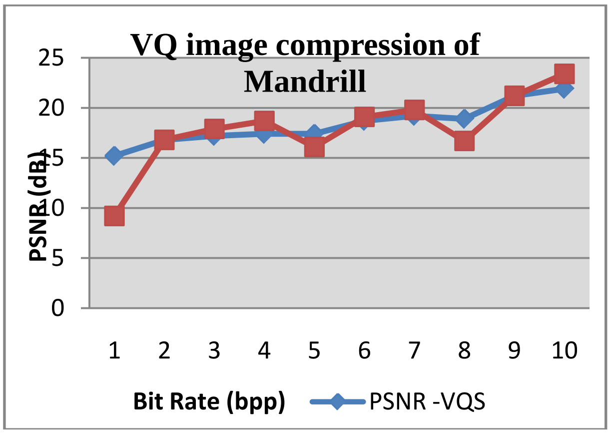


Figure-8: Bit Rate v/s PSNR in VQ of Mandrill image chart

5. CONCLUSION

From the experimental study of different images such as Lena, Girl and Mandrill, it concludes that when the bit rate is increase, the PSNR is also increase in both VQS and VQR.

6. REFERENCES

[1] On the Selection of Image Compression Algorithms by Chaur- Chin Chen, National Tsing Hua University, Taiwan.

[2] "An Algorithm for Image Compression using Differential Vector Quantization", M.Sc. Thesis, by James E. Fowler, Jr. B. S., The Ohio State University, 1995.

[3] "Fundamentals of Image Processing", by Ian T. Young, Jan J. Gerbrands, Lucas J. van Vliet, Delft University of Technology.

[4] "A Comparative Study of Image Compression Method" by Ashwin Swaminathan, Gaurav Agarwal.

[5] "Image Compression Using Vector Quantization", by Nopparat Pantsaena, M. Sangworasil, C. Nantajiwakornchaiand T. Phanprasit,
The Research Center for Communication and Information Technology (ReCCIT), Thailand.

[6] "Implementation of Vector Quantization for Image Compression – A Survey", By G. Boopathy and S. Arockiasamy, Global Journal of Computer Science and Technology, Vol-10, Issue-3, April 2010