Enhanced approach of Entropy Coding in Image Compression

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Software Engineering

Submitted By
Amritpal Singh
(Roll No. 801131005)

Under the supervision of:
Dr. V. P. Singh
Asst. Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2013

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "***Enhanced Approach in Entropy coding for Image Compression***", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. V.P. Singh* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Amritpal Singh

(Amritpal Singh)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. V.P. Singh)
Assistant Professor
Computer Science and
Engineering Department
Thapar University
Patiala

Countersigned by

(Dr. Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

First of all I am thankful to God for blessings and showing me the right decision. With his mercy, it has been possible for me to reach so far.

I would like to thank Thapar Institute of Engineering and Technology, Patiala for providing me an opportunity to carry out thesis as a part of ME (Software Engineering) degree programme. They provided me with all the necessary resources and research environment without which it would not have been possible to complete this work.

I consider it a great privilege to express my deep sense of gratitude to Dr. V.P. Singh, for his invaluable guidance, constant inspiration and critical review throughout the course of this work. He has always been more than generous in sparing his valuable time.

I am also thankful to my friends for their time to time useful comments and constructive suggestions.

Finally, I convey deep sense of gratitude towards my family members for their encouragement and moral support without which it would not have been possible to bring out this thesis.

Amritpal Singh

Amritpal Singh

(801131005)

ABSTRACT

In transmission applications, multimedia systems and computer communications, compression techniques are constrained by real time and on-line considerations that are due to limiting size and complexity of the hardware. The reasons for multimedia systems to compress the data, large storage is required to save the compressed data, the storage devices are relatively slow that do not allow playing multimedia data in real-time, and the network bandwidth, which does not allow real-time data transmission.

In the present research work an enhanced approach of entropy coding has been proposed. The compression of data at source encoding and decompression at its destination decoding, achieves larger compression by using this approach. JPEG image compression standard is used in this research work to show the effectiveness of the proposed run length coding. This standard is flexible and gives user the choice for selecting retrieved image quality. This scheme represents the occurrence of repeated zeros by RUN, and a non-zero coefficient by LEVEL. It removes the value of RUN, as for the sequence of non-zero coefficients it is zero for most of the time and unintended RUN, LEVEL (1, 0) pair used for the zero present between non-zero coefficients is replaced by ‘0’ which results in larger compression. This scheme has been tested on various images and results confirmed that it produces effective results.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 What is an image?

An Image may be defined as a two dimensional function $f(x, y)$, where x and y are spatial coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point [1]. In other words, an image is an array or a matrix of square pixels arranged in rows and columns as shown in figure 1.1.

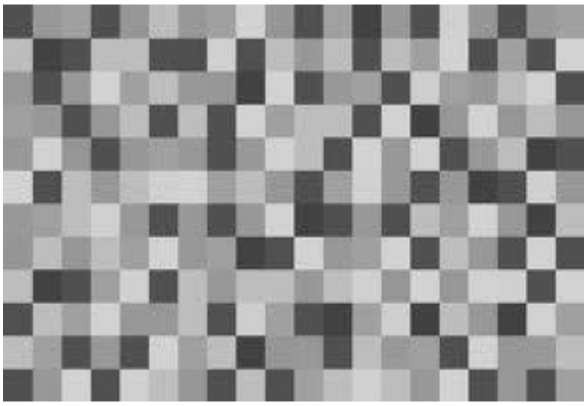

Figure 1.1 Matrix of pixels arranged in rows and columns

A digital image is a numeric representation of a two-dimensional image. Raster images have a finite set of digital values, called picture elements or pixels. The digital image contains a fixed number of rows and columns of pixels. Pixels are the smallest individual element in an image, holding quantized values that represent the brightness of a given colour at any specific point.

Each pixel is a sample of an original image; more samples typically provide more accurate representations of the original, it can be seen in figure 1.2. The intensity of each pixel is variable. In colour image systems, a colour is typically represented by three or four component intensities such as red, green, and blue, or cyan, magenta, yellow, and black [1].

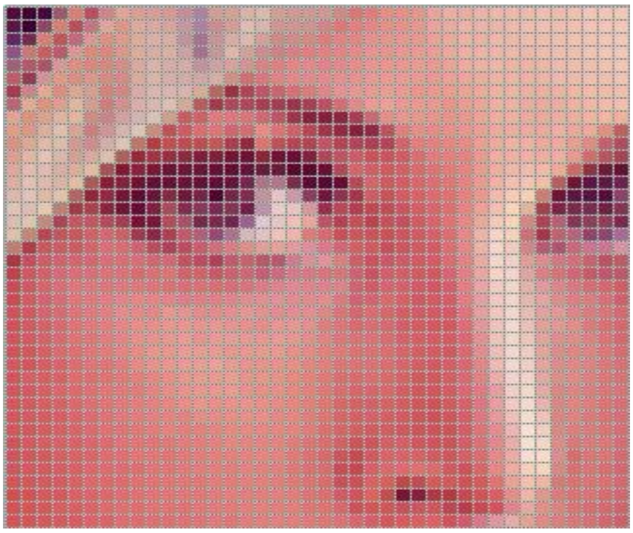

Figure 1.2 Image made up of pixels

1.1.1 Greyscale Image

In an 8 bit greyscale image each picture element has an assigned intensity that ranges from 0 to 255. A grey scale image is what people normally call a black and white image, but the name emphasizes that such an image will also include many shades of grey. Each pixel has a value from 0 (black) to 255 (white) as shown in figure 1.3. The possible range of the pixel values depend on the colour depth of the image, here 8 bit = 256 shades of greyscales [2].

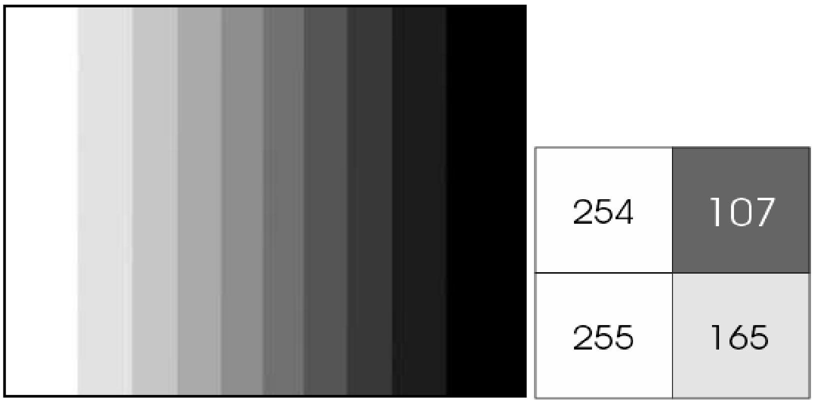

Figure 1.3 Different Shades of Grey with its intensities assigned [2]

1.1.2 Coloured Image

A normal greyscale image has 8 bit colour depth = 256 greyscales. A “true colour” image has 24 bit colour depth = 8 x 8 x 8 bits = 256 x 256 x 256 colours = ~16 million colours. Some greyscale images have more greyscales, for instance 16 bit = 65536 greyscales. Colours in images is motivated by the factor that humans can distinguish thousands of colour shades and intensities, compared to about only two dozen shades of gray.

Figure 1.4 True colour image assembled from three greyscale images

1.2 Colour models

Most colour models in use today are hardware oriented or application oriented. In terms of digital image processing, the hardware oriented models most commonly used in practice are the RGB (red, blue, green) model for the colour monitors and broad class of colour video cameras; the CMY (cyan, magenta, yellow) and CMYK (cyan, magenta, yellow, black) models are used for colour printing [1].

1.2.1 The RGB Colour Model

In the RGB model, each colour appears in its primary spectral components of red, green and blue as shown in Figure 1.5. Red, green, and blue are the primary stimuli for human colour perception and are the primary additive colours. The RGB colour model relates very closely to the way we perceive colour with the r, g and b receptors in our retinas. RGB uses additive colour mixing and is the basic colour model used in television or any other medium that projects colour with light. It is the basic colour model used in computers and for web graphics, but it cannot be used for print production.

The secondary colours of RGB – cyan, magenta, and yellow – are formed by mixing two of the primary colours (red, green or blue) and excluding the third colour. 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.

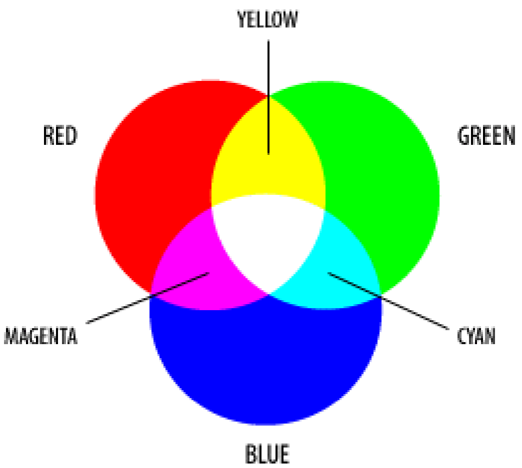

Figure 1.5 Additive model of RGB [2]

Images represented in RGB colour model consists of three component images, one for each primary colour. Consider an RGB image in which each of the red, green, and blue images is an 8-bit image. Then the RGB colour pixel is aid to have a depth of 24-bits. The total number of colours in a 24 bit RGB image is $(2^8)^3 = 16{,}777{,}216$.

1.2.2 The CMY and CMYK Colour Model

Cyan, Magenta and Yellow are the secondary colours of light. That is when a surface coated with cyan pigment is illuminated with white light, no red light is reflected from the surface i.e. cyan subtracts red light from reflected white light. This is the reason if we want a conversion from RGB to CYM, it can be done by a simple operation:

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

Here, assumption made is that all colour values have been normalized to the range [0, 1]. Because of this operation the CMY colour model as shown in figure 1.6 is known as the Subtractive model. The 4-colour 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 colour model.

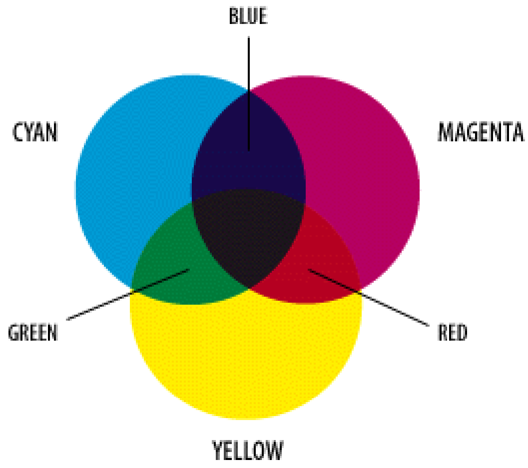

Figure 1.6 Subtractive model of CMYK[2]

The colours created by the subtractive model of CMYK don't look exactly like the colours created in the additive model of RGB. CMYK cannot reproduce the brightness of RGB colours.

1.3 Digital Image Processing

The term digital image processing generally refers to the processing of a two dimensional picture by a digital computer. A digital image can be considered as finite array of real or complex numbers represented by a finite number of bits.

Digital image processing focuses on two major tasks

- Improvement of pictorial information for human interpretation.
- Processing of image data for storage, transmission and representation for autonomous machine perception.

The processes of acquiring an image of the area containing the text, pre-processing that image, extracting the individual characters, describing the characters in a form suitable for computer processing, and recognizing those individual characters are in the scope of Digital image processing. Figure 1.7 shows the Fundamental steps of digital image processing.

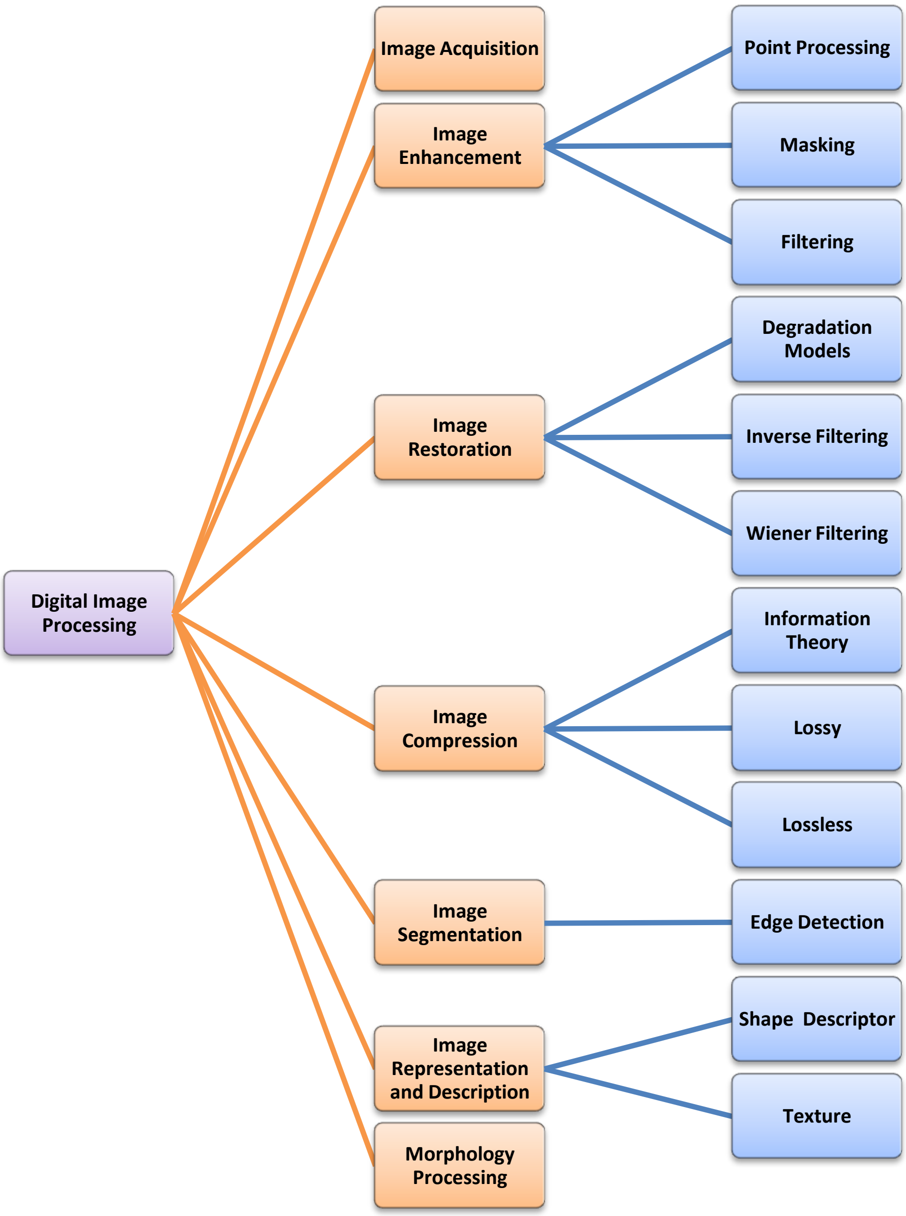

Figure 1.7 Fundamental Steps of Digital Image Processing

1.3.1 Image Acquisition

Image acquisition in image processing can be broadly defined as the action of retrieving an image from some source. Performing image acquisition in image processing is always the first step in the workflow sequence because, without an image, no processing is possible. Image acquisition could be as simple as being given an image that is already in digital form. One of the ultimate goals of image acquisition in image processing is to have a source of input that operates within such controlled and measured guidelines that the same image perfectly reproduced under the same conditions so anomalous factors are easier to locate and eliminate [3]. Figure 1.8 shows the acquisition process.

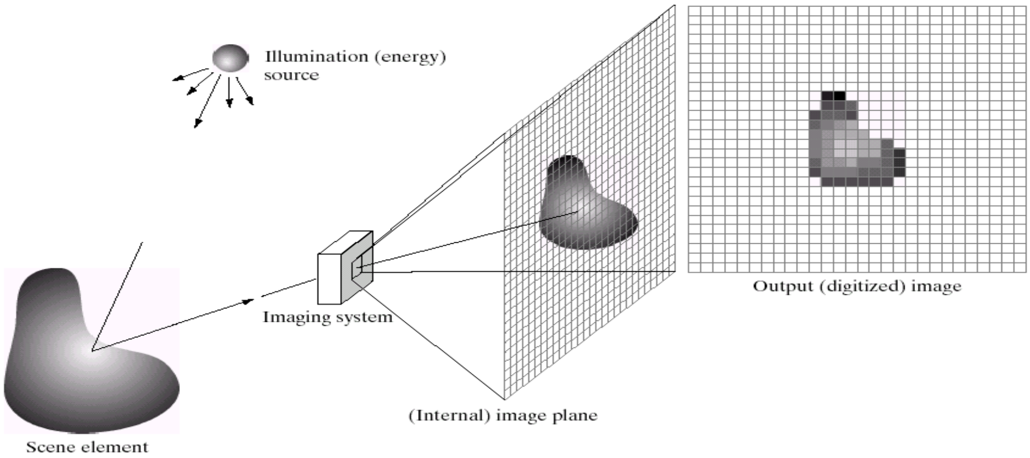

Figure 1.8 Digital Image Acquisition process [4]

1.3.2 Image Enhancement

Image enhancement is the improvement of digital image quality without knowledge about the source of degradation. The idea behind enhancement techniques is to bring out detail that is not clearly visible, to highlight certain areas of interest in an image. Such as, changing brightness, contrast etc [5].

The aim of image enhancement is to improve the interpretability or perception of information in images for human viewers, or to provide better input for other automated image processing techniques. Image enhancement techniques can be divided into two broad categories:

- Spatial domain methods, which operate directly on pixels, and
- Frequency domain methods, which operate on the Fourier transform of an image.

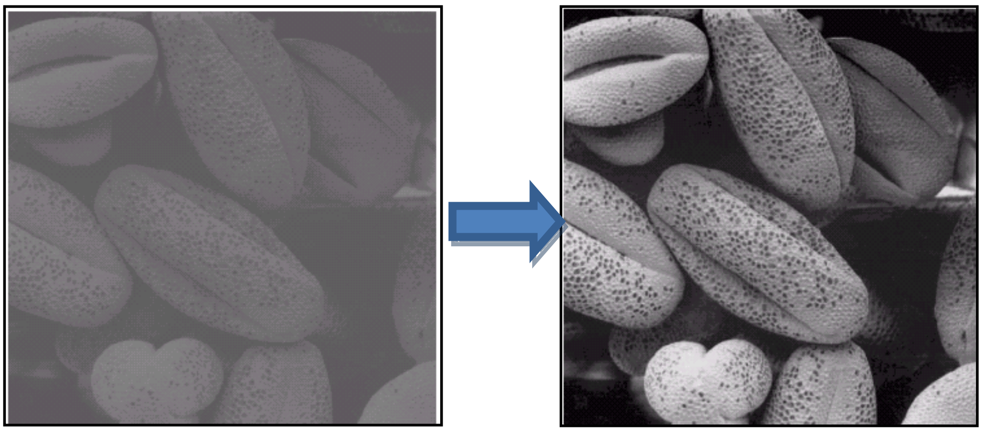

Figure 1.9 Digital Image Enhancement [4]

When it comes to human perception there is no general metric for determining which one is the better image, after enhancement or before enhancement.

1.3.3 Image Restoration

Like image enhancement Image restoration is also an area that also deals with improving the appearance of an image. Image restoration is the operation of taking a corrupted/noisy image and estimating the clean original image. Corruption may come in many forms such as motion blur, noise, and camera misfocus.

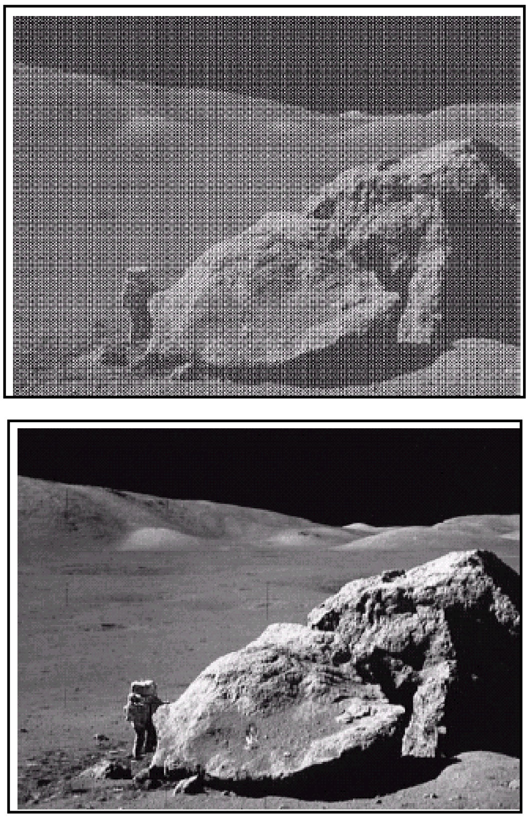

Figure 1.10 Digital Image Restoration [4]

Image restoration is different from image enhancement in that the latter is designed to emphasize features of the image that make the image more pleasing to the observer, but not necessarily to produce realistic data from a scientific point of view. Techniques to restore an image are:

- Degradation model
- Inverse filtering
- Wiener filtering

Deconvolution is an example of image restoration method. It is capable of increasing resolution, especially in the axial direction removing noise and increasing contrast [6].

1.3.4 Image Compression

The objective of image compression [7] is to remove irrelevant data and reduce redundancy of the image data in order to be able to store or transmit data in an efficient form. Compression deals with techniques for reducing the storage required to save an image or the bandwidth to transmit it. Particularly in the uses of internet it is very much necessary to compress data. Some of the well known compression methods are:

- Huffman coding
- Arithmetic coding [8, 9]
- LZW coding
- Run length coding
- Predictive coding [10]
- Block Transformation coding
- Wavelet coding

1.3.5 Image Segmentation

Image segmentation is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyse [11]. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually.

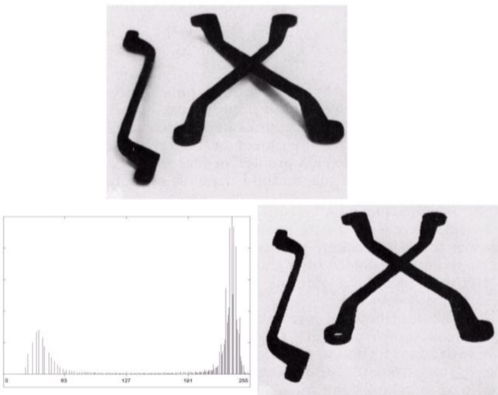

Figure 1.11 Digital Image Segmentation [4]

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region is similar with respect to some characteristic or computed property, such as colour, intensity, or texture

Some of the practical applications of image segmentation are Content-based image retrieval, Medical imaging - Locate tumours and other pathologies, Object detection-Face detection, Locate objects in satellite images (roads, forests, crops, etc.), Recognition Tasks-Face recognition, Fingerprint recognition, Traffic control systems, Video surveillance.

1.3.6 Image Representation and Description

Image Description almost always follows the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region or all the points in the region itself. Description deals with extracting attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another.

- Shape Descriptors
- Texture

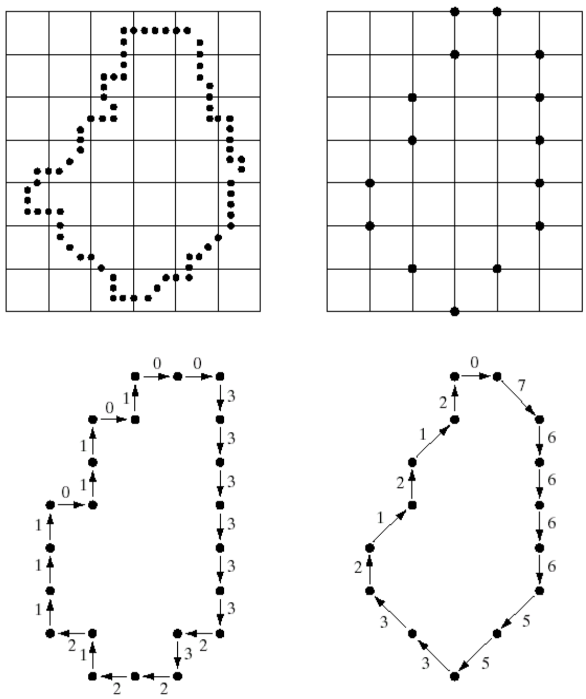

Figure 1.12 Digital Image Representation and Description [4]

1.3.7 Morphological Processing

Binary images may contain numerous imperfections. In particular, the binary regions produced by simple thresholding are distorted by noise and texture. Morphological image processing pursues the goals of removing these imperfections by accounting for the form and structure of the image. Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. Morphological operations rely only on the relative ordering of pixel values, not on their numerical values, and therefore are especially suited to the processing of binary images. Morphological operations can also be applied to greyscale images such that their light transfer functions are unknown and therefore their absolute pixel values are of no or minor interest [12].

1.4 Structure of the Thesis

The rest of thesis is organized in the following order:

Chapter-2: This chapter describes various Image Compression techniques. The analysis provides the advantages and disadvantages of each technique. It also discusses the general image compression model highlighting the JPEG image compression standard. The chapter discusses the major area of research in image compression that is entropy coding using run length coding. It also gives the detail description of previous entropy coding techniques which are being in use till now.

Chapter-3: This chapter provides the problem statement of the thesis and the methodology used to solve the problem. The clear insight about the problem statement is defined. The new approach is discussed for entropy encoding using Run length Coding. Based on the gaps identified in this chapter problem statement for the thesis has been formulated in first section and then Objectives of the thesis are enlisted in the next section of this chapter.

Chapter-4: This chapter provides the complete solution to the problem defined in chapter 3. The new algorithm is defined which is used in the entropy coding phase of JPEG image compression. The new approach to use Run Length Coding is proposed for the problem defined. Also the new algorithm is compared with the traditionally used algorithm.

Chapter-5: This chapter demonstrate the proposed algorithm by implementing it using Matlab tools and results of the experimental setup are presented.

Chapter-6: This chapter describes the conclusion and future scope of the thesis.

CHAPTER 2

LITERATURE REVIEW

In the previous Chapter, the concept of Digital Imaging was introduced by discussing its types and different colour models followed by the fundamental steps of Digital Image Processing in detail. One of crucial process in the field of processing a digital image is Image Compression. Focusing on the stated process, this chapter discusses state of art and research work in the field of Image Compression. This literature reviews different methods of compressing an image proposed by various authors across the globe, then a standard image compression process has been reviewed and finally some of the metrics for image compression are discussed.

2.1 Image Compression

Image Compression deals with techniques for reducing the storage required to save an image, or the bandwidth required to transmit it. Image compression is the process of reducing the amount of data required to represent a digital image. This is done by removing all redundant or unnecessary information. An uncompressed image requires an enormous amount of data to represent it. Image compression is thus essential for the efficient storage, retrieval and transmission of images. The purpose of data compression is to reduce the same part in a datum or represent in a different way and to reduce the time for transfer [7].

There are three main reasons why present multimedia systems require that data must be compressed:

1. Large storage requirements of multimedia data.
2. Relatively slow storage devices that do not allow playing multimedia data in real-time, and
3. The present network bandwidth, which does not allow real-time data transmission.

2.1.1 Image Compression History

Modern techniques for data compression have been started to be used with the development of data theory in the late of 1940s. In 1949, Claude Shannon and Robert Fano developed a systematic way for code words with respect to the block probability [13]. In 1951, the best way to do this was found by David Huffman [14, 15]. In the middle of 1970s, Huffman discovered words opinion, which is updated dynamically, relying on the data used [16] and in the late of 1970s, by the fact that compression of word files is common in any time, compression programs with regards to nearly all adaptive Huffman coding have been created. In 1977, Abraham Lempel and Jacob suggested the opinion of coding in terms of pointer. In the middle of 1980s, the algorithm named LZW [17] by the progressing study of Teri Welch has been a common way for nearly all compression techniques. In the end of 1980s, numerical image has been common and for the compression of these [18], some standards have occurred. In the beginning of 1990s, lossy image compression has been used [19].

2.2 Broad Category of Image Compression

In general, there are two main categories of compression.

2.2.1 Lossless compression

The feature of the lossless compression technique is that the original image can be perfectly recovered from the compressed image. It is also known as Entropy coding since it use decomposition techniques to eliminate or minimize redundancy. Lossless compression is mainly used for applications like medical imaging, where the quality of image is important. The following are the methods that fall under lossless compression: Run length coding, Huffman encoding, and LZW coding.

2.2.2 Lossy compression

Lossy compression technique provides higher compression ratio than lossless compression. In this method the compression ratio is high; the decompressed image is not exactly identical to the original image, but close to it. The quantization process applied on lossy compression technique result in loss of information. The methods that fall under lossy compression technique are Vector Quantization, Fractal coding, Block truncation coding. Sub band coding, etc [7].

2.3 Sources of Redundant Data

Various amounts of data may be used to represent the same amount of information. Some representations may be less efficient than others, depending on the amount of redundancy eliminated from the data [7]. When talking about images there are three main sources of redundant information:

2.3.1 Coding Redundancy

This refers to the binary code used to represent grey values. Assume that the level of the grey is showed by 1 byte. The average word used for an image can be calculated by this equation:

$$L_{avg} = \sum_{k=0}^{L-1} p_r(r_k) = m \qquad \text{eq.}\,(2.1)$$

But if we give smaller amount of bytes to the grey levels, repeating frequently rather than the ones repeating not so much or ever, we can compress the image. This is called as Variable Length Coding (VLC). The most common samples are Huffman and LZW techniques. The comparison between average L values of an image with 3 byte expression and one value of the image with varying coding.

2.3.2 Interpixel Redundancy

This refers to the correlation between adjacent pixels in an image. In the time and space domain, the pixels neighbours to each other have the similar values. This approach depends on the structural or geometrical relations of the objects in the images. Therefore, it's called as Spatial, Geometrical or Inter frames Repetition. In the image, there are two main ways for reducing the repetitions between the pixels. The first one is the one working well in the environment where there is no noise. Another one is the one not sensitive to the noise. Figure 2.1 shows a hierarchical diagram for interpixel redundancy.

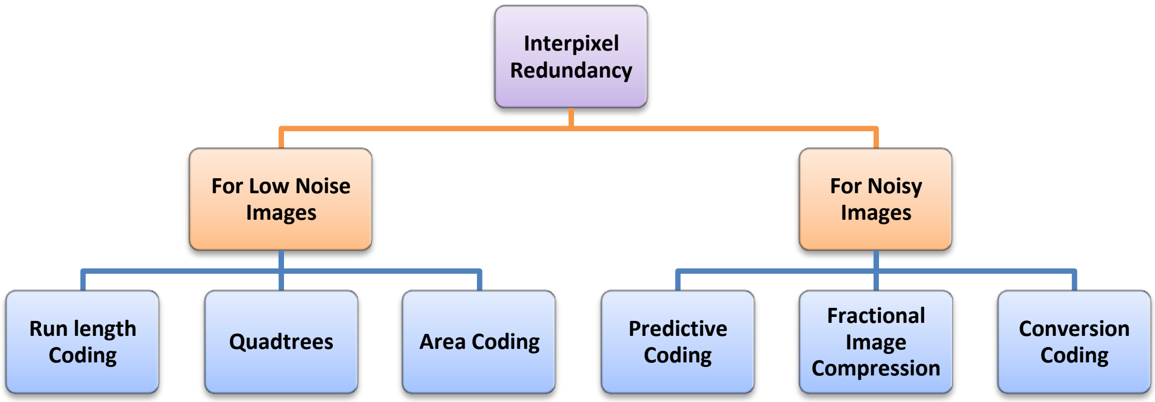

Figure 2.1 Hierarchical diagram for interpixel redundancy

2.3.3 Psychovisual Redundancy

This refers to the unequal sensitivity of the human eye to different visual information. Human eye is not in the level of perceiving some kind of colours. It cares less for the ordinary data compared to extraordinary ones. Some data can be annihilated by not causing any breakdown in the image. This method cannot prevent data loss. The method mainly relates to Quantizing.

Psychovisual Redundancy is fundamentally different from the coding redundancy and interpixel Redundancy. Unlike coding redundancy and interpixel redundancy, psychovisual redundancy is associated with real or quantifiable visual information. Its elimination is possible only because the information itself is not essential for normal visual processing. Since the elimination of psychovisual redundant data results in a loss of quality information. Thus it is an irreversible process.

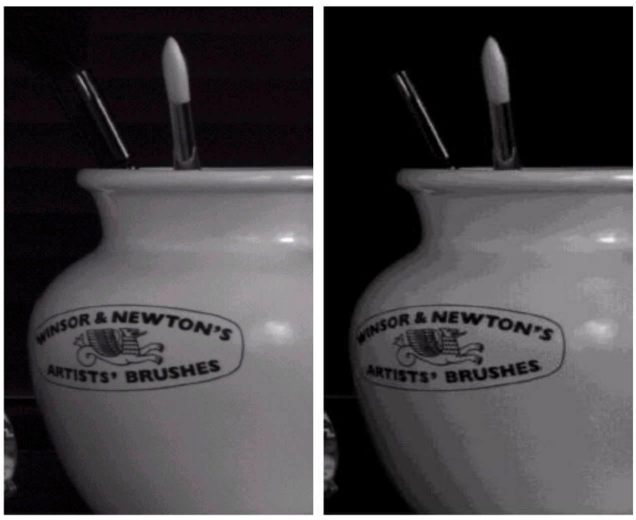

Figure 2.2 (A) Original image (B) Uniform quantization to 16 levels

The benefits of compression are immense. If an image is compressed at a ratio of 100:1, it may be transmitted in one hundredth of the time, or transmitted at the same speed through a channel of one-hundredth the bandwidth (ignoring the compression/decompression overhead). Since images have become so commonplace and so essential to the function of computers, it is hard to see how we would function without them.

2.4 A General Compression Model

Generally a compression system consists of two distinct structural blocks: an encoder and a decoder [20]. An input image f(x, y) is fed into the encoder, which create a set of symbols from the input data. After transmission over the channel, the encoded representation is fed to the decoder, where a reconstructed output image f'(x, y) is generated. In general f'(x, y) may or may not be an exact replica of f(x, y).

A general compression model in figure 2.3 shows that encoder and decoder consists of two relatively independent functions or sub blocks. The encoder is made up of source encoder, which removes input redundancies, and a channel encoder, which increases the noise immunity of the source encoder's output. Similarly, the decoder includes a channel decoder followed by a source decoder. If the channel between the encoder and decoder is noise free, the channel encoder and decoder are omitted, and the general encoder and decoder become the source encoder and decoder, respectively [1].

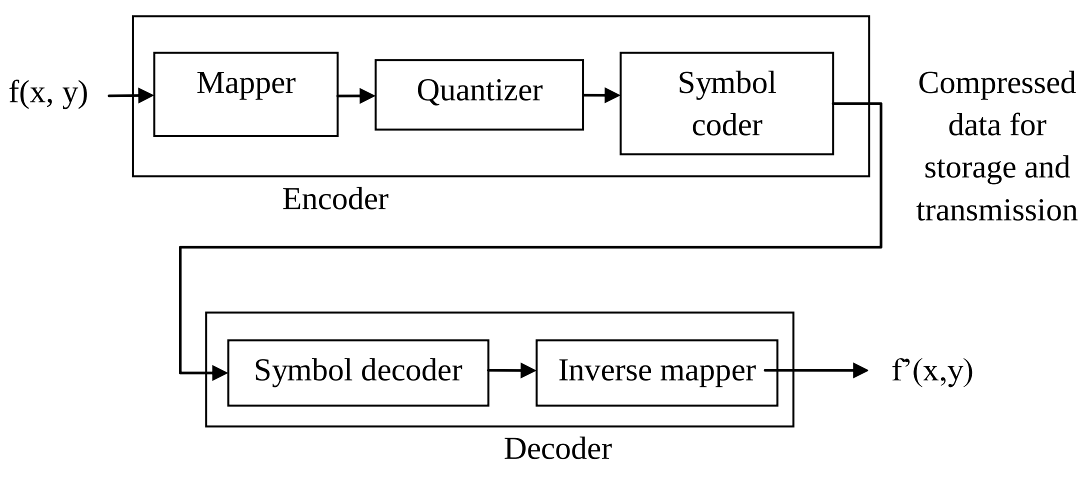

Figure 2.3 General Compression Model

2.4.1 The Source Encoder

The source encoder is responsible for reducing or eliminating any coding, interpixel, or psychovisual redundancies in the input image. The specific application dictates the best encoding approach. Normally, the approach can be modelled by a series of three independent operations. Operation is designed to reduce one of the three redundancies discussed earlier.

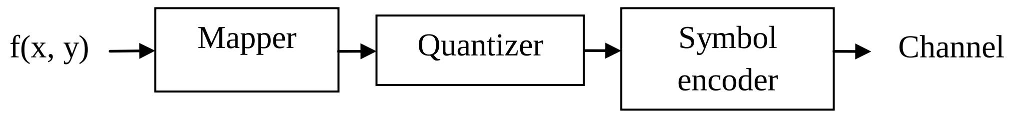

Figure 2.4 Source Encoder

2.4.2 Mapper

In the first stage of the source encoding processes, the mapper transforms the input data into a (usually nonvisual) format designed to reduce interpixel redundancies in the input image. This operation generally is reversible and may or may not reduce directly the amount of data required to represent the image.

2.4.3 Quantizer

The second stage or quantizer block reduces the accuracy of the mapper's output in accordance with some pre-established fidelity criterion. This stage reduces the psychovisual redundancies of the input image. This operation is irreversible. Thus, it must be omitted when error-free compression is desired.

2.4.4 Symbol Encoder

In the third and the final stage of source encoding processes, the symbol coder create a fixed or variable-length code to represent the quantizer output and maps the output in accordance with the code. The term symbol coder distinguishes this coding operation from the overall source encoding processes. In most cases, a variable length code is used to represent the mapped and quantized data set. It assigns the shortest code words to the most, frequently occurring output values and thus reduces coding redundancy. The operation is completely reversible. Upon completion of symbol coding step, the input image has been processed to remove each of the three redundancies discussed earlier in section 2.3. Here we have shown the source encoding process as three successive operations, but all three operations are not necessarily included in every compression. For example, the quantizer may be omitted when error free compression is desired. In addition, some compression techniques normally are modelled by merging blocks that are physically separate in figure 2.4.

2.4.5 Source Decoder

The source decoder shown in figure 2.5 contains only two components: a symbol decoder and an inverse mapper. These blocks perform, in reverse order, the inverse operations of the source encoder's symbol encoder and mapper blocks. Because quantization results in irreversible information loss, an inverse quantizer block is not included in the general source decoder model shown in the figure 2.5.

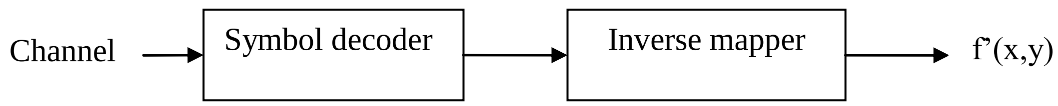

Figure 2.5 Source Decoder

2.5 The JPEG Compression Standard

JPEG is the image compression standard developed by the Joint Photographic Experts Group that is compatible with continuous tone still images both greyscale and colour [1]. JPEG is a flexible, general purpose compression standard which meets the need of all continuous-tone still image applications. It is a standard which enables the exchange of images across the application boundaries. The JPEG standard [21] specifies three elements in an encoder: discrete cosine transformer, quantizer (used in lossy compression) followed by entropy encoder as shown in Figure 2.6. This standard does the image compression without compromising with the quality of image.

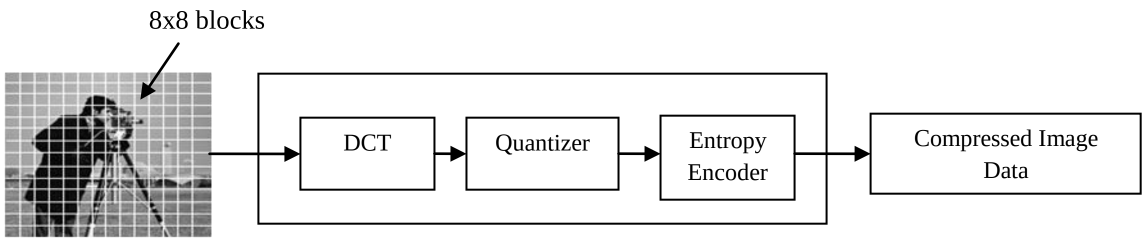

Figure 2.6 The JPEG 8×8 Block Encoding Process

The standard specifies three elements in an decoder: entropy decoder, Inverse quantizer followed by discrete cosine transformer as shown in Figure 2.7.

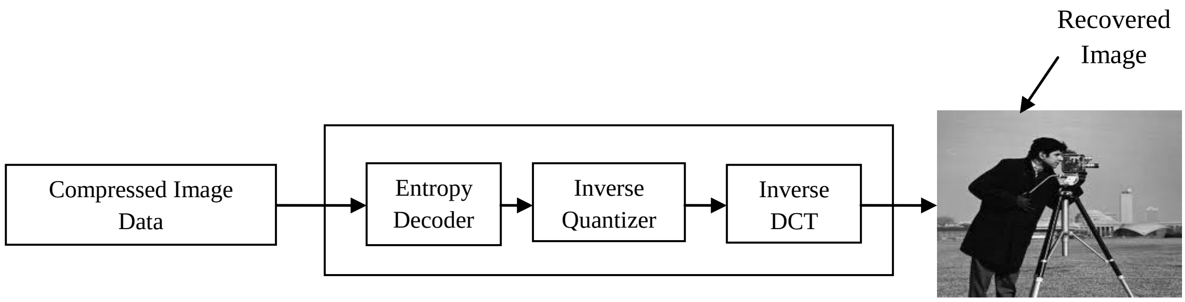

Figure 2.7 The JPEG 8×8 Block Decoding Process

2.5.1 Discrete Cosine Transform

JPEG is a popular DCT-based [22, 23] still image compression standard. DCT is a Fourier-related transform [24] which expresses a finite sequence of image data points in terms of a sum of cosine functions oscillating at different frequencies. The function of the transformation block in compression algorithm is to reduce the correlation between the pixels. The frame is first divided into blocks of 8x8 pixels each. The 2-dimensional 8x8 Forward Discrete Cosine Transform [25] converts the image in a digital format where the data representation is more compressible and can be computed by a machine. Among all DCT coefficients, the one at the top left is called DC coefficient, while the rest are AC coefficients shown in Fig 2.10. After transformation, most of the image energy will be concentrated at the upper left region of the block, and a lower right region of the block will have the frequencies with the amplitude of 0. The 8x8 2-Dimentional Forward DCT [26] is:

$$F(u,v) = \frac{1}{4} C(u)C(v) \sum_{x=0}^{7} \sum_{y=0}^{7} f(x,y) \cos\left[\frac{\pi(2x+1)u}{16}\right] \cos\left[\frac{\pi(2y+1)v}{16}\right] \quad \text{eq. (2.2)}$$

for u=0,....,7 and v=0,....7

where $C(k) = \begin{cases} \frac{1}{\sqrt{2}} & for\ \ k = 0 \\ 1 & otherwise \end{cases}$

The 8x8 2-Dimentional Inverse DCT [27] is:

$$f(x,y) = \frac{1}{4} \sum_{x=0}^{7} \sum_{y=0}^{7} C(u)C(v)F(u,v) \cos\left[\frac{\pi(2x+1)u}{16}\right] \cos\left[\frac{\pi(2y+1)v}{16}\right] \quad \text{eq. (2.3)}$$

for x=0,....,7 and
y=0,....7

2.5.2 Quantization

After the transformation from the DCT, each of the 64 DCT coefficients is uniformly quantized [27] with a 64-element Quantization Table, which must be specified by the application (or user) as an input. The JPEG recommended luminance quantization array [28] is given in Figure 2.8 and it can be scaled to provide variety of compression levels. The goal of this processing step is to discard Information which is not visually significant. This phase of JPEG compression method makes it a lossy compression method [29, 30], as the truncation of the frequencies is done during quantization phase which is not recoverable while reconstructing the original data of the image. The JPEG committee has recommended certain Q matrix that work well and the performance is close to the optimal condition, the Q matrix for luminance and chrominance components is defined in figure 2.5 and 2.6. So in this step, each DCT coefficient is divided by its corresponding quantizer step size in the Quantization Matrix Q(u,v), followed by rounding to the nearest integer using eq. 2.4

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

Figure 2.8 Quantization Matrix Q(u, v), step size matrix for luminance components

$$Q_C = \begin{pmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{pmatrix}$$

Fig 2.9 Quantization Matrix Q(u, v), step size matrix for Chrominance components

During Quantization every coefficients in the 8×8 DCT matrix is divided by a corresponding quantization value. The quantized coefficient and and the reverse the process can be achieved by the eq.2.5

$$F_q(u,v) = \mathrm{Round}\frac{F(u,v)}{Q(u,v)} \qquad \text{eq. (2.4)}$$

$$F_{deq}(u,v) = F_q(u,v) * Q(u,v) \qquad \text{eq. (2.5)}$$

2.5.3 Entropy Coding

This step achieves additional compression in a lossless manner by encoding the quantized DCT coefficients more compactly based on their statistical characteristics. It is useful to consider entropy coding as a 2-step process.

2.5.3.1 Zigzag

The first step converts the zigzag sequence of quantized coefficients into an intermediate sequence of symbols as shown in Figure 2.10. The second step converts the symbols to a data stream [15] in which the symbols no longer have externally identifiable boundaries.

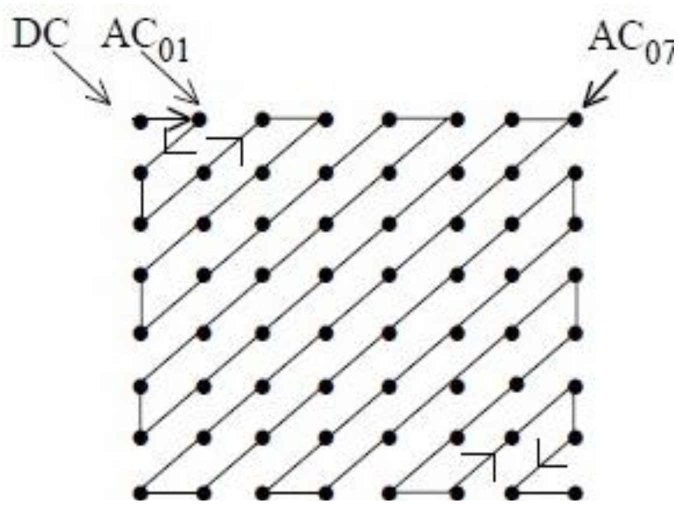

Figure 2.10 Zigzag Sequence

All of the quantized coefficients are ordered into the “zigzag” sequence to form a 1-D sequence of quantized coefficients. This ordering helps to facilitate entropy coding by placing low-frequency coefficients (which are more likely to be nonzero) before high-frequency coefficients. Then this sequence of AC coefficients for each block is encoded using run length encoding which is designed to take advantage of the long runs of zeros that normally results from the reordering.

2.5.3.2 Differential Coding

The mathematical representation of the differential coding is in eq. 2.6:

$$Diff_i = DC_i - DC_{i-1} \qquad \text{eq. (2.6)}$$

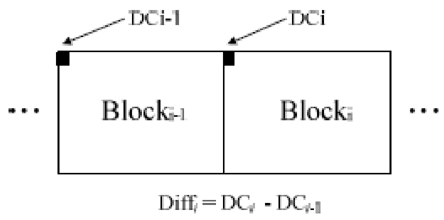

Figure 2.11 Differential Coding

We set $DC_0 = 0$. DC of the current block DCi will be equal to DC$_{i-1}$ + Diff$i$. Therefore, in the JPEG file, the first coefficient is actually the difference of DCs. Then the difference is encoded with Huffman coding algorithm together with the encoding of AC coefficients.

2.5.3.3 Run-Length Coding

After quantization and zigzag scanning, we obtain the one-dimensional vectors with a lot of consecutive zeroes [31, 32]. We can make use of this property and apply run-length coding, which is variable length coding. Let us consider the 63 AC coefficients in the original 64 quantized vectors first. For instance, we have:

57, 45, 0, 0, 0, 0, 23, 0, -30, -16, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, ..., 0

We encode the given stream of number with run length coding. Result comes out to be:

(0,57) ; (0,45) ; (4,23) ; (1,-30) ; (0,-16) ; (2,1) ; EOB

The notation (R, L) means that there are R zeros in front of L, and EOB (End of Block) is a special coded value means that the rest elements are all zeros. If the last element of the vector is not zero, then the EOB marker will not be added. On the other hand, EOB is equivalent to (0, 0), so we can express the above stream as:

(0,57) ; (0,45) ; (4,23) ; (1,-30) ; (0,-16) ; (2,1) ; (0,0)

2.6 Image Quality Factors:

There are various factors that determine the quality of image and few of them are listed below:

1. Sharpness: Determines the amount of detail an image can convey. System sharpness is affected by the lens and sensor. Over sharpening can degrade the image quality. Images from many compact digital cameras are over sharpened.
2. Noise: It is the random variation of image density, visible as grain in film and pixel level variations in digital images. Lower the noise in the image better is the quality of the image.
3. Dynamic range: It is the range of light levels a camera can capture. It is closely related to noise, high noise implies low dynamic range.
4. Colour Accuracy: it is an important but ambiguous image quality factor. Many viewers prefer enhanced colour saturation; the most accurate colour isn't necessarily the most pleasing.
5. Distortion: It is an aberration that causes straight lines to curve near the edges of images. It can be troublesome for the architectural photography and metrology. Distortion is worst in wide angle, telephoto and zoom lenses.
6. Tone Reproduction: It is the relationship between scene luminance and the reproduced image brightness.
7. Contrast: It is the slope of the tone reproduction curve in a log-log space. High contrast usually in involves loss of dynamic range -- loss of detail, or clipping, in highlights or shadows.
8. Vignette: Also called as light falloff, darkens images near the corners. It can be significant with wide angel lenses.
9. Colour moiré: It is an artificial colour banding that can appear in images with repetitive patterns of high spatial frequencies, like fabrics and picket fences. It is affected by lens sharpness, the anti-aliasing (low pass) filter (which softens the image), and demosaicing software. It tends to be worst with the sharpest lenses.
10. Artifacts: Software can cause significant visual artifacts, including data compression and transmission losses, over sharpening halos and loss of fine, low contrast detail.

2.7 Measuring image quality

In many image processing algorithms, there are certain parameters that need to be determined by users to yield the best results. This is often a difficult task for naive users as the best values may be image dependent [33]. It is important to measure the quality of the image for image processing application. How good the image compression algorithm is dependent upon the quality of compressed image produced on application of that algorithm. To measure the efficiency of image compression algorithm various image quality assessment algorithms/tools have been developed with the time which are used for optimization purpose, where one maximize quality at a given cost, comparative analysis between different alternatives and for quality monitoring in real-time applications etc.

There are basically two approaches for image quality measurement:-

1. Subjective Measurement
2. Objective Measurement

2.7.1 Subjective Measurements

Digital images are subject to a variety of distortions during compression, transmission, processing, and reproduction. Since most of the image data will eventually be consumed by humans, the most reliable means of assessing image quality is subjective evaluation [34]. In fact, in image compression systems, the truly definitive measure of image quality is perceptual quality. The distortion is specified by MOS (Mean Opinion Score), which is result of perception based subjective evaluation. The meaning of the 5-level grading scale of MOS is 5-imperceptible, 4-perceptible, but not annoying, 3-slightly annoying, 2-annoying and 1-very annoying [35]. Another possible absolute rating scale could be 1-excellent, 2-fine, 3-passable, 4-marginal, 5-inferior, and 6-unusable. This rating scale is shown in table 2.1.

Table 2.1 Grading Scale for MOS (Mean Opinion Score)

Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

2.7.2 Objective Measurements:

Objective image quality metrics can be classified according to the availability of the original image, with which the distorted image is to be compared. Most existing approaches are known as:-

- Full-reference: in this a complete reference image is assumed to be known.
- No-reference: In this the reference image is not available, and a no reference or blind quality approach is desirable.
- Reduced-reference: The reference image is partially available, in the form of a set extracted features made available as information to help evaluate the quality of the distorted image.

The three common objective measurements used are as follows :

2.7.2.1 Mean Square Error:

It is the average squared difference between a reference image and a distorted image [36]. It is computed pixel-by-pixel by adding up the squared differences of all the pixels and dividing it by the total pixel count. The squaring of the differences dampens the small differences between the two pixels but penalizes large ones. The larger value of MSE means that image is of poor quality. Let x(m,n) denotes the sample of original image, x'(m,n) denotes the sample of compressed image. M and N are number of pixels in row and column directions, respectively. MSE can be calculated using eq. 2.7 [4]:

$$\text{MSE} = \frac{1}{MN}\sum_{m=1}^{M}\ \sum_{n=1}^{N}\ (x(m,n) - x'(m,n))^2 \qquad \text{eq.}\,(2.7)$$

2.7.2.2 Peak signal Noise Ratio:

Often abbreviated as PSNR, defines a ratio between the maximum possible power of the signal and power of corrupting noise. PSNR is the parameter that shows the difference between the original image and compressed image. PSNR is usually expressed in terms of the logarithmic decibel scale. The PSNR is most commonly used as a measure of quality of reconstruction of the lossy compression codec's. The signal in this case the original data, and the noise is the error introduced by compression. It can be calculated using eq. 2.8. The small value of the Peak Signal to Noise Ratio means that image is of poor quality.

$$PSNR = 10log\frac{255^2}{MSE} \qquad \text{eq.}\,(2.8)$$

2.7.2.3 The Ratio of Compression

For instance, if there are two data that yield the same results and the first one consist of n1 data and the second one consists of n2 data, for these two data set, the ratio of the compression is CR=n1/n2. Here CR refers to Compression Ratio.

Rate of Data Redundancy calculates the percentage of redundant data in a file. It can be calculated using eq. 2.9.

$$RD = 100\left(1 - \frac{1}{CR}\right) \qquad \text{eq.}\,(2.9)$$

We can find out two solutions;

1. If n2<<n1: CR extends to infinity and RD goes to 1. In this situation, the data in n1 repeats highly.
2. If n2>>n1: CR goes to 0 and RD extends to minus infinity. In this situation, n2 consists of lots of data.(Increasing despite the compression)

For example, if CR=10, then RD=100(1-1/10) = 90. It means that 10 percent of the data used, the rest of it is unnecessarily repeated [7].

CHAPTER 3

PROBLEM STATEMENT

Based on literature survey in the previous chapter, gaps have been identified for entropy coding phase of Image Compression. Gaps identified in this chapter have been recognized by many researchers and they had altered it in many ways to get the desired result. Based on these gaps problem statement for this thesis has been formulated in first section and then Objectives of this thesis are enlisted in the next section of this chapter.

3.1 Problem Definition

Image Processing has a large area of research. There is profound research on various techniques for different applications. One of the applications for image processing is Image Compression which has undergone a large amount of research and many algorithms has been provided in last 2-3 decades that how to maximize the Compression ratio of an image without much quality loss. One of such algorithm is the Run Length Coding technique which encodes the data by presenting/storing the redundant information in such a way that the overall storage size of the image is reduced.

Besides its broad area of applications [37, 38], there are many issues related to the Run length coding as well which are stated as follows:

- If there is only a single byte to encode, add an extra byte for the 'Run' byte.
- The traditional algorithm removes the redundant data, but creates other redundant data while encoding in some other way.
- The size of the compressed file depends on the complexity of the image. The more the detail, the worse the compression. The traditional run length coding is quite simple to handle such complexities.

- It does not work well on continuous-tone images, although JPEG uses it quite effectively on the coefficients that remain after transforming and quantizing image blocks.

3.2 Objectives

The main objective of the thesis is to provide the solution for the above mention issues for entropy encoding using run length coding in image compression. The objectives of this thesis are:

1. Develop an algorithm which can effectively encode the compressed data in a lossless manner.

2. Remove as much redundant data from an image as it can without altering the basic phases of an image compression model.

3. Compare and analyze the proposed algorithm with traditional Run Length Coding algorithm.

4. Verify the results with existing algorithm.

3.3 Methodology used in proposed solution

Following are the steps performed to implement the objectives:

1. Literature Survey is carried out. Various research papers are explored to formulate the problem.
2. Greyscale images are used for compression.
3. Algorithm using run length coding for improved compression ratio has been formulated.
4. Traditional run length coding is improved by removing the unintended zeros for the sequence of non-zero coefficients.
5. Single byte (0) is encoded instead of using (1, 0) pair for the zero between non-zero characters to achieve larger compression.
6. The traditional as well as the proposed algorithm is implemented using MATLAB.

CHAPTER 4

PROPOSED SOLUTION

The gaps and the objectives of this thesis have been explained in previous chapter. In this chapter a solution to optimize the conventionally used Run length coding scheme for JPEG image Compression Standard and advantages and disadvantages of this scheme have been discussed. First of all, Conventional Run length scheme has been discussed with its Flow diagram and pseudo code. Then the proposed algorithm is discussed in the same manner which overcomes the drawbacks of the original Run length coding.

4.1 Original Run Length Coding

Run length coding is the standard coding technique for compressing the images, especially when images are compressed by block transformation. This method counts the number of repeated zeros which is represented as RUN and the non-zero coefficient represented as LEVEL is appended following the sequence of zeros. Run Length Coding is flexible in nature and therefore many authors have altered it for its suitable use in various applications [39, 40].

In JPEG compression algorithm the run length only encodes the consecutive number of zeros, as the probability of occurrence of consecutive zeros is very high. The method counts the number of repeated zeros between two non-zero numbers and appends the non-zero number following the count of zeros. For a series of consecutive non-zero numbers it adds redundancy in the encoded data, when it was meant to compress the original data. The encoded message for a fixed coding scheme consists of an 8 bit value of RUN and an 8 bit signed value of LEVEL. This kind of scheme is called Byte Level Run Length Coding or Zero Run Length Coding.

The flow diagram for Original Run Length Coding is shown in Figure 4.1.

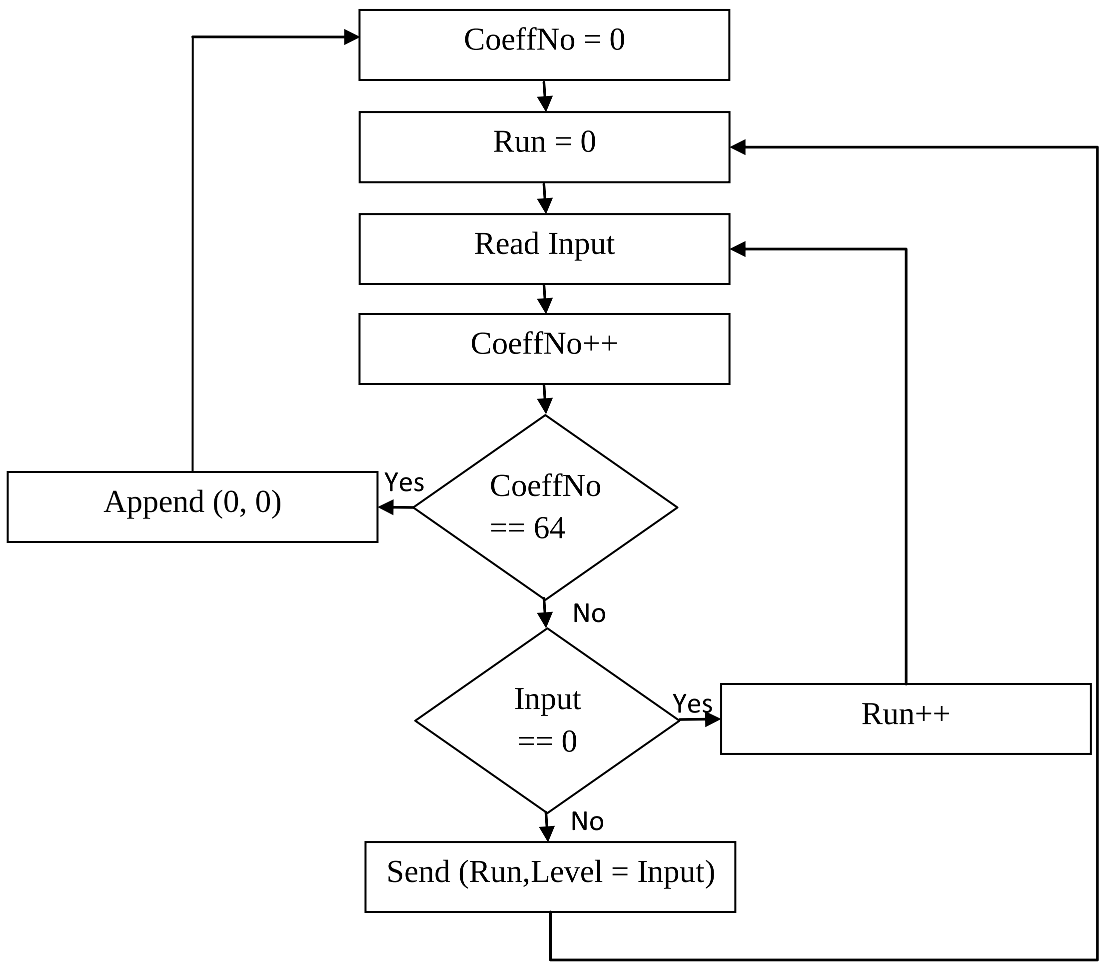

Figure 4.1 Flow Diagram for Original Run Length Coding [41]

4.2 Optimized Run Length Coding

To overcome the drawbacks of conventional Run Length Coding, a more optimized scheme is proposed. The Optimized Run Length Coding uses a pair of (RUN, LEVELS) only when a pattern of consecutive zeros occur at the input of the encoder. The non-zero digits are encoded as their respective values in LEVELS parameter. RUN parameter is eliminated from the final encoded message for the non-zero digits. The FLAG bit is used to identify the presence of a zero or consecutively occurring zeros in the input data stream [41]. This way it represents the single zero between non-zero characters with a 2-digit sequence of (1, 0) and adds an extra digit (for the zero between non-zero characters) to the final encoded data which again creates a scope of optimizing the algorithm further.

4.3 Proposed Run Length Coding

Further optimization can be done by removing the unintended RUN, LEVEL (1, 0) pair used to represent a single zero present between the non-zero coefficients. So instead of using (1, 0) pair for the zero between non-zero characters, single '0' is encoded. Figure 4.2 shows the flow diagram of this scheme.

The pseudo code for the Proposed Run Length Coding is as follows:

```
Run = 0;
FLAG = 1;
for (CoeffNo = 0; CoeffNo < 64; CoeffNo++)
    Read Input;
    If (Input == 0)
        Run++;
        FLAG = 0;
    Else if (FLAG == 0)
        If (Run ==1)
            Print (Level = 0);
            Print (Level = Input);
            Run = 0;
            FLAG = 1;
        Else
            Print (Run, Level = 0);
            Print (Level = Input);
            Run = 0;
            FLAG = 1;
            End if
    Else
        Print (Level = Input);
        FLAG = 1;
        End if
APPEND EOB;
```

This modification makes use of the following two observations,

- Consecutively occurring zeros in the input data stream is very common.

- Non-zero digits with same value occurring consecutively is a very rare case in classic JPEG image compression.

The optimized coding algorithm is illustrated in figure 4.2:

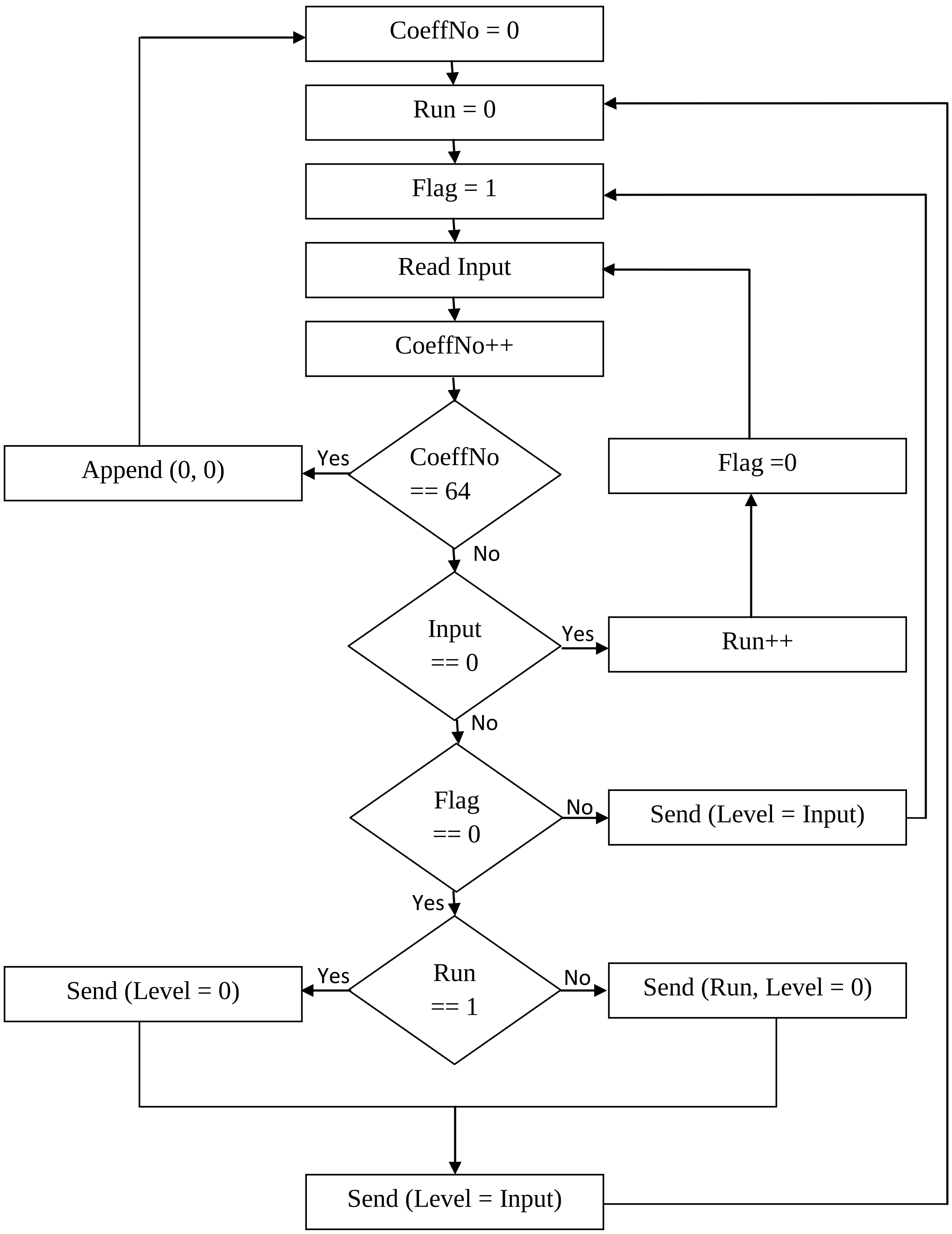

Figure 4.2 Flow Diagram of the Proposed Run Length Coding Scheme

Added FLAG bit to the run length encoding algorithm, is used to identify the presence of a zero or consecutively occurring zeros in the input data stream. Presence of continuous zero digits in the input data stream toggles the initialized value of FLAG bit once. The FLAG bit is reinitialized at the end of the continuous zero digits in the input data stream. For consecutively occurring zeros at the end of block the process to increment the value of RUN variable is bypassed to the process of appending (0, 0), "the End of Block indication", to the encoded message. Then to remove the unintended (1, 0) stream encoded for the Single zero present between two non zero coefficients the algorithm makes a check to the value of RUN count, if it is equal to '1' that means there is not a stream of consecutively occurring zeros instead we have a single zero between the non zero coefficients, so for that we will encode '0' instead of (1, 0) Run, Level pair.

4.4 Comparing the algorithms

Comparing the flow graphs for original and the proposed run length encoding algorithms, one can see an increase in the cyclomatic complexity of the proposed algorithm due to an extra FLAG bit and a check for the single zero (RUN == 1). These characteristics of the proposed algorithm can always be compromised with a high amount of compression, or the percentage efficiency. In context to prove the optimization of the algorithm following 8x8 block is considered as shown in Figure 4.3.

102	-33	-3	-4	-2	-1	0	0
21	-2	-3	0	-1	0	0	0
-3	0	1	0	0	0	0	0
2	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0
-2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 4.3 8x8 image block after Quantization Phase

4.4.1 Original Run Length Coding

According to the Original Run Length Coding algorithm the output (33 digit sequence) of the 8x8 image block in Fig.4.3 would be:

[(0,-33) (0,21) (0,-3) (0,-2) (0,3) (0,-4) (0,-3) (1,2) (0,1) (1,1) (1,-2) (0,-1) (0,-1) (3,-2) (11,1) (0,0)].

4.4.2 Optimized Run Length Coding

The output (29 digit sequence) of the block under consideration would be:

[-33 21 -3 -2 -3 -4 -3 (1, 0) 2 1 (1, 0) 1 (1, 0) -2 -1 -1 (3, 0) -2 (11, 0) 1 (31, 0) (0, 0)].

4.4.3 Proposed Run Length Coding

The output (26 digit sequence) of the block under consideration using this scheme would be:

[-33 21 -3 -2 -3 -4 -3 **0** 2 1 **0** 1 **0** -2 -1 -1 (3, 0) -2 (11, 0) 1 (31, 0) (0, 0)].

The results show an efficient compression using the new Run Length Coding scheme applied on all type of images. The results of various algorithms applied on the image block taken as an example in Figure 4.3, are stated in Table 4.1.

Table 4.1 Results of the Image block under consideration

S. No	Algorithm Used	Digits used in Encoding Sequence
1	Original Run Length Coding	33-digits
2	Optimized Run Length Coding	29-digits
3	Proposed Run Length Coding	26-digits

So, clearly the newly proposed algorithm is much more efficient. Moreover Table 4.1 shows the result of only one 8x8 image block, there are many 8x8 blocks in which the image is divided at the DCT phase of JPEG compression, and on an average 1 to 2 such single zeros present between two non-zero characters can be encountered.

CHAPTER 5

IMPLEMENTATION

This chapter focuses on implementation of proposed algorithm. Tools and hardware used for implementation are explained briefly. Then implementation and results of the various images from the data set are shown. Results show the effectiveness of the new algorithm.

5.1 Hardware and Tools used

Experiments are conducted to verify the proposed algorithm on the machine of 2 GB Ram, 2-core 2 GHz processor and 320 Gb of hard disk with Windows 7 installed.

5.1.1 MATLAB 7.10

MATLAB or Matrix laboratory is a numerical computing environment developed by MathWorks. It is a high-level language and interactive environment that is used to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and Fortran. MATLAB can be used in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modelling and analysis, and computational biology. Add-on toolboxes extend the MATLAB environment to solve particular classes of problems in these application areas. It provides a number of features for documenting and sharing the work [42].

5.2 Implementation and Results

A set of three images is used to implement the proposed algorithm. The details of these images are shown in table 5.1. The set of images are shown in figure 5.1-5.3.

Table 5.1 Details of Images used in Implementation

S.No.	Image name	Image Extension	Image Size
1.	Lena	.bmp	512x512
2.	Cameraman	.tiff	256x256
3.	Gold hill	.bmp	256x256

Figure 5.1 Lena.bmp

Figure 5.2 Cameraman.tiff

Figure 5.3 GoldHill.bmp

When the above shown images are compressed the following results are produced:

1. Reconstructed Image - Image constructed after compression.
2. Comparison Chart – Shows the difference between Compression Ratios when used with Proposed Run Length Coding and Original Run Length Coding.

For each of the image present in the data set, the metric data produced for Original Run length coding is in table 5.2.

Table 5.2 Results of Images at quality factor=50 for Original RLC

S.No.	Image Name	Compression Ratio	Rate of Data Redundancy (%)	Mean Square Error	Peak Signal to Noise Ratio
1.	Lena.bmp	4.5960	78.2417	6.6813	39.8822
2.	Cameraman.tiff	3.3893	70.4956	7.3115	39.4907
3.	Goodhill.bmp	2.6175	56.0150	4.8041	41.3146

For each of the image present in the data set, the metric data produced for Original Run length coding is in table 5.2.

Table 5.3 Results of Images at quality factor=50 for Proposed RLC

S.No.	Image Name	Compression Ratio	Rate of Data Redundancy (%)	Mean Square Error	Peak Signal to Noise Ratio
1.	Lena.bmp	5.4222	81.5575	6.6813	39.8822
2.	Cameraman.tiff	4.2110	76.2527	7.3115	39.4907
3.	Goodhill.bmp	3.2560	69.2871	4.8041	41.3146

5.2.1 Results for lena.bmp

Figure 5.5 shows the reconstructed image after compression at quality factor of 50 percent. Figure 5.6 shows a chart which summarizes the compression ratio of the image using both algorithms at different quality factors (20-90). The comparative results show that the proposed algorithm compresses the image more than that of the traditional one.

Figure 5.5 Reconstructed Image for Lena.bmp

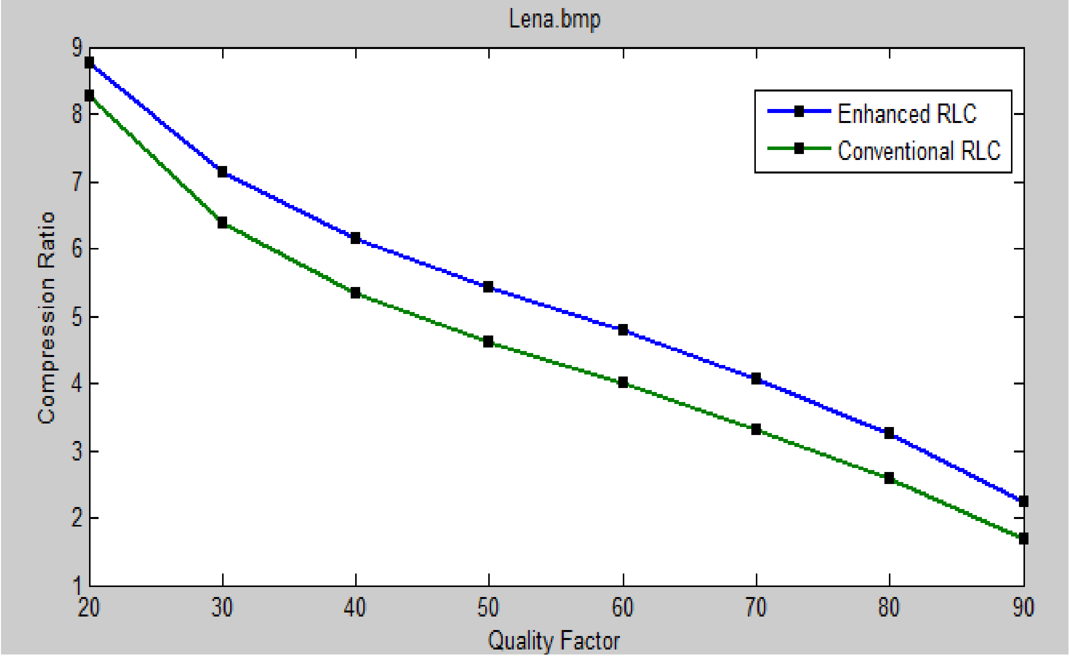

Figure 5.6 Comparison Chart for Lena.bmp

5.2.2 Results for cameraman.tiff

Figure 5.7 shows the reconstructed image after compression at quality factor of 50 percent. Figure 5.8 shows a chart which summarizes the compression ratio of the image using both algorithms at different quality factors (20-90). The comparative results show that the proposed algorithm compresses the image more than that of the traditional one.

Figure 5.7 Reconstructed Image for Cameraman

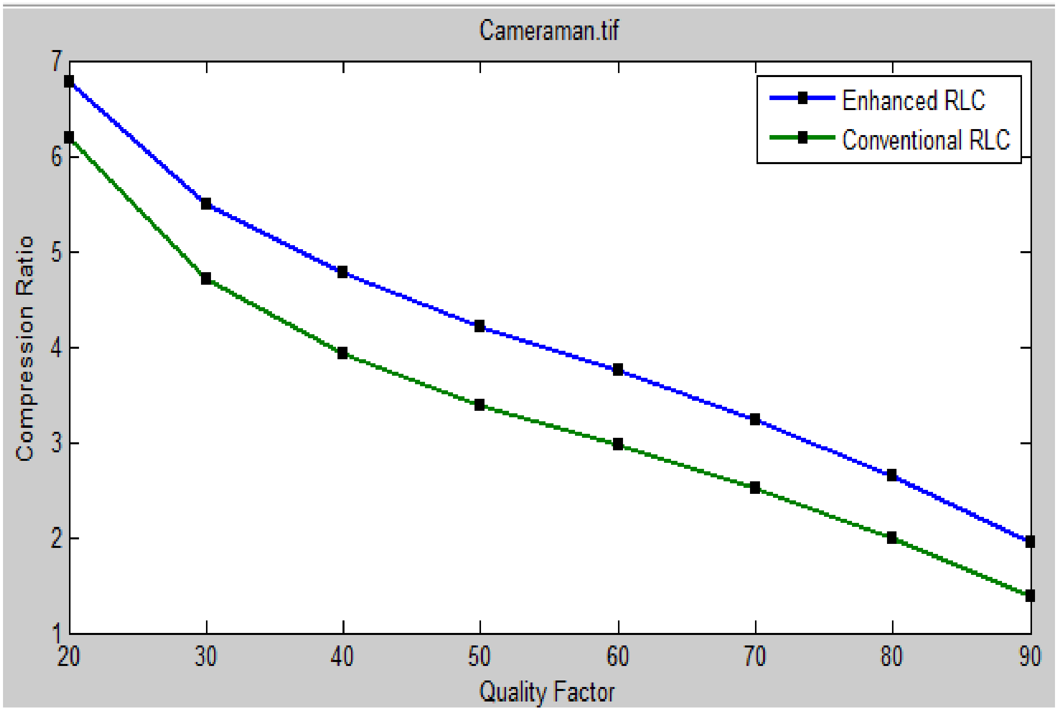

Figure 5.8 Comparison Chart for Cameraman.tiff

5.2.3 Results for Goodhill.bmp

Figure 5.9 shows the reconstructed image after compression at quality factor of 50 percent. Figure 5.10 shows a chart which summarizes the compression ratio of the image using both algorithms at different quality factors (20-90). The comparative results show that the proposed algorithm compresses the image more than that of the traditional one.

Figure 5.9 Reconstructed Image for Goodhill.bmp

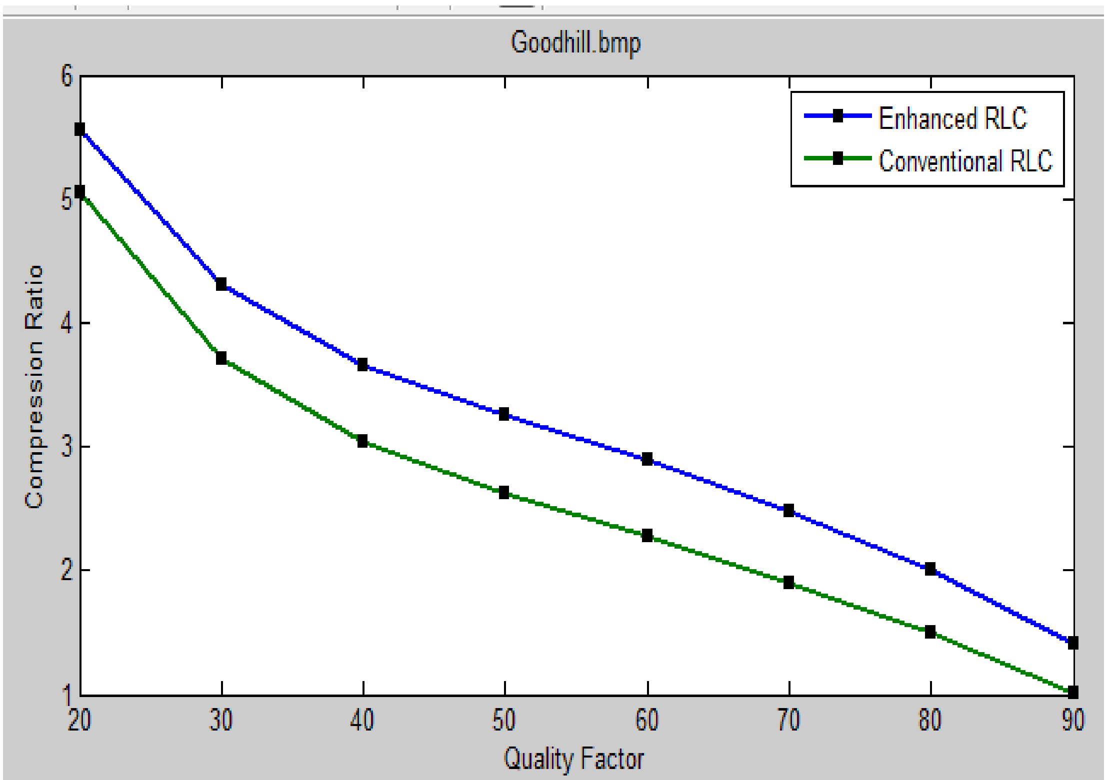

Figure 5.10 Comparison Chart for Goodhill.bmp

5.3 Snapshots of Implementation using Matlab

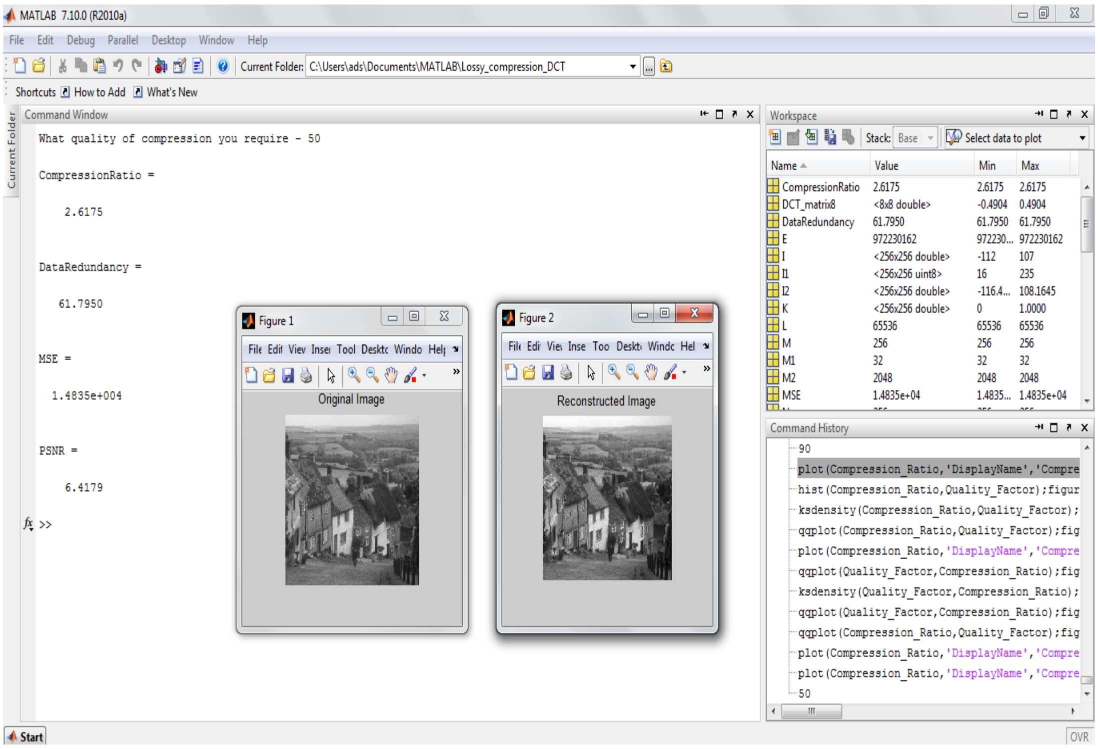

Figure 5.11 Executing orignlrunlength.m file for goodhill.bmp

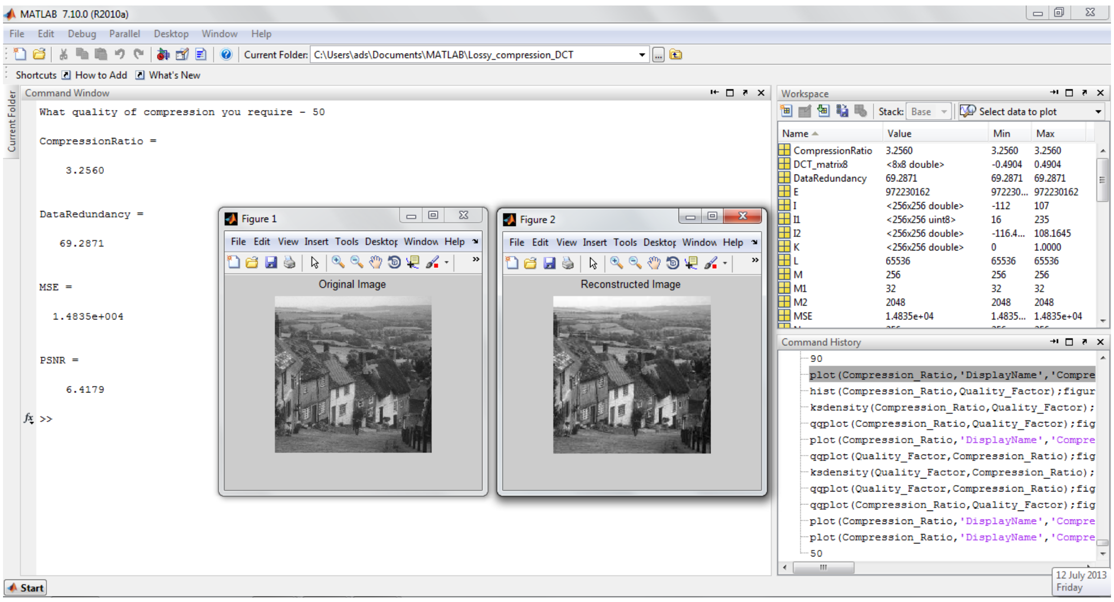

Figure 5.12 Executing propsdrunlength.m file for goodhill.bmp

CHAPTER 6

CONCLUSION

This Chapter concludes the work presented in this thesis. Some future directions have been proposed at the end of this chapter, which can be considered to implement using other compression standards.

6.1 Conclusion

To save the storage space and communication bandwidth, compression of large information files is a feasible approach. The compression technique developed in the research work is based on transformation coding. Discrete Cosine Transform is used as it is found to be superior to other transformation techniques. The improved Run Length Coding gives a good amount of compression without much degradation in picture quality. Present technique of compression has been tested and found successful to compress the grey scale and coloured images. For most of the images quality factor of 5 or less produces loss of picture quality. The images with quality factor of 20 or higher show graceful degradation in the image quality. In most cases images can be compressed up to 80% without much loss in picture quality.

6.2 Future Scope

Much of work is still to be done for this algorithm. Some of which is listed below:

1. In the present work, only still images are analyzed. This technique further must be used for M-JPEG motion pictures.
2. As run length coding is used in MPEG1, MPEG2 and MPEG4 standards, it must be implemented using the proposed algorithm to compress the videos in the above stated file formats.

REFERENCES

[1] R C. Gonzalez and R E. Woods, Digital Image Processing. Prentice-Hall, 2008.

[2] The Hubble European Space Agency Information Centre. [online] http://www.spacetelescope.org/projects/fits_liberator/improc/

[3] Wisegeek clear answer for common question, [online.] http://www.wisegeek.com/what-is-image-acquisition-in-image-processing.htm

[4] Brian M. Namee, Digital Image Processing: Introduction, [online], http://www.comp.dit.ie/bmacnamee/

[5] R Sanderson, "Introduction to remote sensing", New mexico space grant consortium, Aug 2010.

[6] Rice University, [online] http://www.owlnet.rice.edu/~elec539/Projects99/BACH/proj2/intro.html

[7] T. Cebrail, S.K. Sarikoz, "An overview of Image Compression Approaches", The 3rd International conference on Digital Telecommunications, IEEE 2008.

[8] I.H. Witten, R. M. Neal, J. G. Cleary, Arithmetic Coding For Data Compression, Communications of the ACM, Volume 30 Number 6, pp 520-540, 1987.

[9] Paul G. Howard, Jeffrey Scott Vitter, Arithmetic Coding for Data Compression, Proceedings of the IEEE. Vol. 82, No. 6, June 1994.

[10] V. E. 1. Neagoe, "Predictive ordering technique and feedback transform coding for data compression of still pictures," IEEE Trans. Commm., vol. 40, pp. 385-396, Feb. 1992.

[11] Hossein Mobahi, Shankar Rao, Allen Yang, Shankar Sastry and Yi Ma, Segmentation of Natural Images by Texture and Boundary Compression, International Journal of Computer Vision (IJCV), 95 (1), pg. 86-98, Oct. 2011.

[12] Nick Efford, Digital Image Processing: A Practical Introduction Using Java™, Pearson Education, 2000.

[13] C. E. Shannon, “A mathematical theory of communication,” Bell Syst. Tech. J., vol. 27, pp. 398-403, July 1948.

[14] David A. Huffman, the creator of Huffman Coding. [online] http://www.huffmancoding.com

[15] O. Johnson, “On the Redundancy of Binary Huffman Codes” IEEE Trans. Inform. Theory, Vol. IT-26, No-2, pp.220-223, Mar. 1980.

[16] David A. Huffman, A Method for the Construction of Minimum-Redundancy Codes, I.R.E 1952.

[17] Terry A. Welch, A Technique for High data Compression, IEEE, Volume 17 Issue 6, pp 8-19, June 1984.

[18] Beaudoin, Normand,” An Accurate Discrete Fourier Transform for image processing”, Pattern Recognition, Vol.3, pp. 935-939, 2002.

[19] W.B. Pennebaker, J.L. Mitchell, “JPEG still image data compression standard”, Van Nostrand Reinland, New York, 1993.

[20] Jain A. K. - "Fundamentals of Digital Image Processing", Prentice Hall, 1989.

[21] G. K. Wallace, 'The JPEG Still Picture Compression Standard', Communications of the ACM, Vol. 34, Issue 4, pp.30-44, 1991.

[22] N. Ahmed, T. Natarajan, and K. R. Rao. “Discrete cosine transform,” IEEE Trans. Comput., vol. C-23, pp. 90-93, Jan. 1974.

[23] W.H.Chen, C.H.Smith, and S.Fralick, “A Fast Computational Algorithm for Discrete Cosine Transform”, IEEE, Trans. on Commun., vol. COM-25, pp.1004-1009 Sept. 1977.

[24] Beaudoin, Normand,” An Accurate Discrete Fourier Transform for image processing”, Pattern Recognition, Vol.3, pp. 935-939, 2002.

[25] R.C. Reininger, J.D. Gibson, “Distribution of Two Dimensional DCT Coefficients for Images” IEEE Trans. Comm. Vol. 31, pp.835-839, June, 1983.

[26] Using Cosine Transform, IEEE Transactions On Communications, Vol. 43, No. 11, November 1995.

[27] H.A. Peterson, H. Peng. J.H. Morgan, W.B. Pennebaker, “Quantization of color image components in the DCT domain”, Human Vision, Visual Processing, and Digital Display. Proc. SPIE, pp. 210-222, 1991.

[28] A.J. Ahumada Jr., H.A. Peterson, “Luminance Model-Based DCT Quantization for color Image Compression” Human Vision, Visual Processing, and Digital Display III, B.E. Rogowitz, ed., SPIE, 1992.

[29] N. Ponomarenko, V. Lukin, K. Egiazarian, E. Delpc,” Comparison of Lossy Compression performance on Natural color Images”, IEEE 2009.

[30] Wai C. Chu, “On Lossless and Lossy Compression of Step Size Matrices in Jpeg Coding” IEEE 2013.

[31] S. W. Golomb, “Run-length encodings,” IEEE Trans. Informat. Theory, vol. IT-12, no. 4, pp. 399401, July 1966.

[32] S. Arazaki, M. Saigusa, S. Hashiguchi, M. Ohki, Image Data Compression by DCT with Adaptive Run-Length Coding, IEEE Transactions on Consumer Electronics, Vol. 37, No. 4, Nov 1991.

[33] Z. Wang, ”Applications of Objective Image Quality Assessment Methods” IEEE Signal Processing Magazine, pp.137-142, Nov 2011.

[34] Z. Wang, G.Wu, H.R.Sheikh, E.P.Simoncelli, E.-H. Yang, and A. C. Bovik, “Quality-aware images,” IEEE Trans. Image Processing, vol. 15, pp. 1680–1689, June 2006.

[35] N. Yamsang, S. Udomhunsakul,” Image Quality Scale (IQS) for Compressed Images Quality Measurement”, Proceedings of the International MultiConference of Engineers and Computer Scientists 2009, Vol I IMECS 2009, March 18 - 20, 2009.

[36] Zhou Wang and Alan C. Bovik, “Mean squared error, Love it or leave it?”, IEEE Signal Processing Magazine Jan 2009.

[37] M. Fatemi, M. Kelarestaghi, “Application of run-length coding in reprojection”, IEEE pp. 1166-1168, 1993.

[38] Y. Yasuda, K. Arai, and K. Tsuno, “Digital data compression of facsimile signal by rearranging picture elements,” Int. Elect. Commun. Eng. Jpn., vol. C-575, no. 85, Sept. 1975.

[39] C. Tu, J. Liang, and T. D. Tran, "Adaptive run-length coding," IEEE Signal Processing Letters, vol. 10, pp. 61-64, Mar. 2003.

[40] E. H Yang and L. Wang, "Joint optimization of run-length coding, Huffman coding and quantization table with complete baseline JPEG decoder compatibility," in Proc. ICIP 2007, vol. 3, pp. 181- 184, Sept. 2007.

[41] Akhtar M.B., Qureshi A.M., Qamar-ul-Islam, “Optimized Run length coding for jpeg Image Compression used in Space Research Program of IST”, IEEE 2011.

[42] Mathworks-Accelerating the pace of engineering and science. [online], http://www.mathworks.in

LIST OF PUBLICATIONS

[1] Amritpal Singh & V.P. Singh, "An Enhanced Run Length Coding for JPEG Image Compression", *International Journal of Computer Application, Vol. 72-No.20, June 2013*, pp. 21-26.