IMAGE COMPRESSION USING HYBRID BTC AND DCT TECHNIQUE

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
VLSI DESIGN AND EMBEDDED SYSTEM

Submitted by:

ADITIKA THAKUR
2K16/VLS/03

Under the supervision of

PROF. S. INDU

(H.O.D, ECE department)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

MAY, 2018

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, Aditika Thakur, Roll No. 2K16/VLS/03 student of M.Tech. (VLSI DESIGN AND EMBEDDED SYSTEM), hereby declare that the project Dissertation titled **"IMAGE COMPRESSION USING HYBRID BTC AND DCT TECHNIQUE"** which is submitted by me to the Department of ELECTRONICS AND COMMUNICATION ENGINEERING, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology. This work has not previously formed the basis for the award of any Degree, Diploma Associate ship, Fellowship or other similar title or recognition.

Place: Delhi **ADITIKA THAKUR**

Date:

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled **"IMAGE COMPRESSION USING HYBRID BTC AND DCT TECHNIQUE"** which is submitted by ADITIKA THAKUR, Roll No 2K16/VLS/03 Department of Electronics And Communication Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date:

PROF. S. INDU

SUPERVISOR

ACKNOWLEDGEMENT

I would like to thank my supervisor, Prof. S. INDU, Department of Electronics and Communication Engineering, DTU for the guidance, encouragement and advice she has provided throughout my time as her student. I have been extremely lucky to have a supervisor who cared so much about my work and who responded to my questions and queries so promptly. I attribute the level of my Masters degree to her encouragement and effort and without her this thesis, too, would not have been completed or written. One simply could not wish for a better or friendlier supervisor.

Finally, I take this opportunity to extend my deep appreciation to my family and friends, for all that they meant to me during the crucial times of the completion of my project.

Date Aditika thakur

2K16/VLS/03

M.Tech (VLSI Design and Embedded System)

ABSTRACT

With the growth of modern communication techniques, demand for image data compression is increasing rapidly. The main goal of image compression techniques is to save a image using as much less bits as possible so that it occupies less space and it transmission is easy. Thus save lots of time while transmitting the image and also storage space. Image compression techniques are divided mainly into two domains: spatial domain and transform domain. Spatial domain techniques have the advantages of hardware complexity whereas transform domain techniques have the advantage of better compression ratio. This thesis is based on combining the two domain techniques to achieve better results. The two techniques used are BTC and DCT. Various test images are used to find the results and compare it with other compression techniques. Also compression efficiency, PSNR, MSE are found for all and compared.

TABLE OF CONTENTS

List of Figures

List of Tables

LIST OF ABBREVIATIONS

Abbreviation	Full Form
BTC	Block Truncation Coding
DCT	Discrete Cosine Transform
WHT	Walsh Hadamard Transform
IDCT	Inverse Discrete Transform
IWHT	Inverse Walsh Hadamard Transform
CR	Compression Ratio
MSE	Mean Square Error
PSNR	Peak Signal To Noise Ratio
BIC	Binary Image Compression
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
BP	Bit Plane
WM	Walsh Matrix
IMG	Image

CHAPTER 1

INTRODUCTION

Image compression is data compression technique applied to digital images which results in reduction of cost for storing and transmitting the images. It can also be regarded as reducing the image size in bits with resulting image quality of acceptable level. As a result a large number of images can be stored in the given space. It also reduces time of transmitting or downloading the image.

In a digital image each pixel has three color components that is R G B and each R G B requires 8 bits to store its value this implies that each pixel requires 24 bits to store its value. Also most the data stored in these pixels are redundant and need to be stored every time because doing this will use a lot of memory space and thus neglecting such redundancies can help a lot in saving space.

1.1 MOTIVATION

With the advancements in the digital era, the transmitting and storing a vast amount of data has increased to a great extant. Storing such large data requires a large amount of space and thus compression of data for better utilization of space and ease in transmission has become main concern of the researchers. A lot of researchers worked in the domain of digital image compression and various image compression methods have been introduced.

Image compression techniques have been broadly divided into two categories spatial domain compression and frequency domain compression with each technique having its advantages and disadvantages. Spatial domain image compression techniques have advantage of hardware complexity. They are less complex than frequency domain techniques. Whereas frequency domain techniques achieve better compression ratio than spatial domain methods.

1.2 RESEARCH OBJECTIVE

The main object of this thesis is to compare the advantages of using a hybrid compression technique combining both spatial as well as frequency domain compression technique with all other compression techniques such as BTC, DCT, WHT etc. by calculating the various parameters such as compression ratio (CR), mean square error (MSE), peak signal to noise ratio (PSNR).

1.3 ORGANIZATION OF THESIS

This thesis has been organized in several chapters.

Chapter 1 gives the brief introduction on image compression and its various techniques. It also includes motivation and objective of thesis.

In chapter 2, a literature survey on the various image compression techniques has been given.

In chapter 3, brief introduction on image compression and its various types and domains has been given.

In chapter 4, spatial domain image compression technique has been discussed. Block truncation coding is a spatial domain compression technique. Its algorithm along with the working example has been explained.

In chapter 5, frequency domain image compression technique has been discussed. Discrete cosine transform is a frequency domain compression technique. DCT equation and DCT matrix has been given along with the algorithm.

In chapter 6, walsh hadamard transform based image compression technique has been discussed. Its benefits over other transform are explained. Also walsh functions and wlash matrix and its various forms are discussed.

In chapter 7, hybrid image compression technique based on BTC and DCT is discussed along with its algorithm.

In chapter 8, various parameters that measure the quality of image compression technique are discussed such as compression ratio, mean square error, peak signal to noise ratio.

In chapter 9, all the results from various techniques are compared.

In chapter 10, conclusion and future scopes are given.

CHAPTER 2

LITERATURE SURVEY

In, literature various image compression techniques have been discussed. Each technique has its advantage and disadvantage. Image compression is achieved by removing the redundant data in the image [7]. Different types of redundant data are inter-pixel, spatial domain redundant data and coding redundancies and visual redundancy.

Inter-pixel redundant data is the similarities among the various pixel values of the image and thus can be removed by encoding or transform. Visual redundancy [1] arises from the fact that different frequency components have different sensitivity toward our eyes. Our eye is less sensitive toward the high frequency components of the image so using quantization they can be removed without much affecting the quality of image.

The image compression techniques are broadly classified in two categories: spatial domain compression techniques and transform domain compression techniques. Various spatial domain compression techniques are BTC, BIC etc.

In 1979, Edward J. delph and O. Robert Mitchell presented a novel method [2] of image compression called block truncation coding. In this technique each pixel value is compressed such that it requires only 1 bit to store its value rather than 8 bits [12]. The block truncation coding gives the bit rate of 2bits/pixel (originally 8bits/pixel) for gray scale images and thus compression ratio of 75%.

In 1994, Watson, Andrew B. presented a novel method [3] of image compression based on discrete cosine transform. In this technique image is divided into blocks and than transform is applied to convert it into frequency domain. This is very popular method and offers the transform gain same as that of KLT and is better than DFT because its computational complexity is less.

In 2010, Jau-Ji Shen et al presented a vector quantization based image compression technique [5]. This technique involves adjustment of encoded difference map between the input image and stored in VQ compressed version.

In 2011, Suresh Yerva, et al presented a novel method of image folding providing lossless image compression [8]. This method is based on the fact that digital images have pixel redundancies i.e. adjacent pixels in the image store almost same data and this property has been used for prediction. According to this method, first column folding is applied to the image followed by row folding. This process is done iteratively on the image which reduces the image size to a smaller required value. The proposed method showed good performance when compared with the existing lossless image compression algorithms.

In 2012, Firas A. Jassim, et al presented a novel technique for image compression called five module method (FMM) [6]. According to this technique the image is converted into 8x8 blocks and each pixel in 8x8 blocks is converted to a multiple of 5 for each of RGB array. Than each pixel values are divided by 5 and new values are obtained which require only one bit to encode and thud it requires less storage space than the original values in which each pixel is 8 bits. The main advantage of their method is that it provides high PSNR (peak signal to noise ratio) but the disadvantage is that it provides low CR (compression ratio).

In 2014, Khin Thida Win, Nang Aye Aye Htwe Another very popular compression technique is using walsh hadamard transform [4]. The advantage of this technique is that its computational complexity is very less even lesser than DCT because its matrix has only +1 or -1 value.

Different techniques give different values for compression ratio, mean square error, peak signal to noise ratio. The various frequency domain image compression techniques are DCT, WHT, DWT, KLT etc. KLT has the advantage of energy compaction but its disadvantage is that it depends on data value and requires sending the transform which in turn reduces the transform gain

CHAPTER 3

IMAGE COMPRESSION

3.1 INTRODUCTION

Image compression is data compression technique applied to digital images which results in reduction of cost for storing and transmitting the images. It can also be regarded as reducing the image size in bits with resulting image quality of acceptable level. As a result a large number of images can be stored in the given space. It also reduces time of transmitting or downloading the image. In a digital image each pixel has three color components that is R G B and each R G B requires 8 bits to store its value this implies that each pixel requires 24 bits to store its value.

Also most the data stored in these pixels are redundant [3] and need to be stored every time because doing this will use a lot of memory space and thus neglecting such redundancies can help a lot in saving space. Since we know that human eyes are less sensitive to high frequency components [3] so in various image compression algorithm specially transform domain algorithm uses this concept and perform the quantization on transformed image which results in strings of zeros in high frequency components and need not to be stored and can be neglected easily.

Various quantization matrices are available to perform the quantization process and one can choose according to its requirement but there will always exist a trade off among the amount of compression and quality of the reconstructed image. If the quantization matrix is chosen such that it provides higher compression ratio than quality of reconstructed will be bad and vice versa.

With the advancements in multimedia and internet it has become very crucial to incorporate various compression techniques in order to ensure a better communication. With the help of compression techniques it has become possible to share, store or download an image in less space in less time [9]. Thus advancements in the image compression techniques have always attracted the researchers.

3.2 TYPES OF IMAGE COMPRESSION

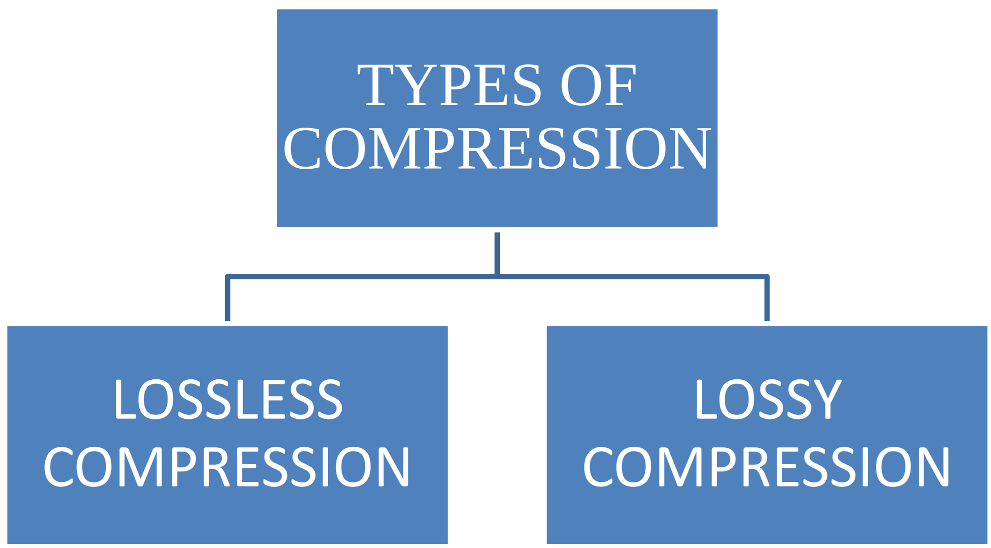

Figure 3.1: Types of image compression

3.2.1 LOSSLESS COMPRESSION

These are the compression techniques that results in the extraction of the original data from the compressed data i.e. no data is lost and hence lossless [3]. It is used where it is necessary that original and decompressed data should be same that is no data component is lost. Some example of this type are medical images, source code, text files etc.

The lossless compression techniques consist of two steps where first is to generate a statistical model of the input data and then in second step this data is used for mapping the input into stream of bits in such way that high frequency data will generate small output than less frequency data [9]. Encoding algorithms mostly used for generating bit stream are Huffman coding and arithmetic coding. Arithmetic coding results in better compression than Huffman coding.

Different lossless compression techniques are [3]:

i. Run-length coding: This technique provides better compression for the data in which data elements occur more than once and data contains many such data elements. In this lossless compression technique data elements that occur more than one sequence of data are stored only once along with the number of times it has occurred.
ii. Huffman coding: This method uses a variable length code table for encoding the input string. This table is obtained from the probability that how many times a data occurs in a string.
iii. Prediction by partial matching: this algorithm uses prediction phenomena. This algorithm predicts the next data set from the previous data set.

3.2.2 LOSSY COMPRESSION

These are the image compression techniques that uses approximations or removal of high frequency or comparatively less valuable data in order to compress the size of the image so that it can be stored in small space for take less time and space during transmission and reception [9]. This is different from lossless compression methods that do not remove the data. These techniques results in higher data compression than that from lossless methods.

A good quality lossy compression technique results in better image compression with slight data degradation such that it is not noticed by the user. These compression methods are mostly used in compressing images, videos, audios and internet telephony.

Different types of lossy image compression techniques are as follows [3]:

i. Transform coding: In these compression methods the raw data is transformed into another domain which more clearly specifies the information content. Data minimizing is the main goal of these techniques but along with this they can also represent data accurately for the same space.

ii. Chroma sub-sampling: This uses the fact that our eye is more sensitive to spatial changes rather than color. Thus ignoring some of the chrominance values of the image.

iii. Fractal compression.

1.3 VARIOUS DOMAINS OF IMAGE COMPRESSION

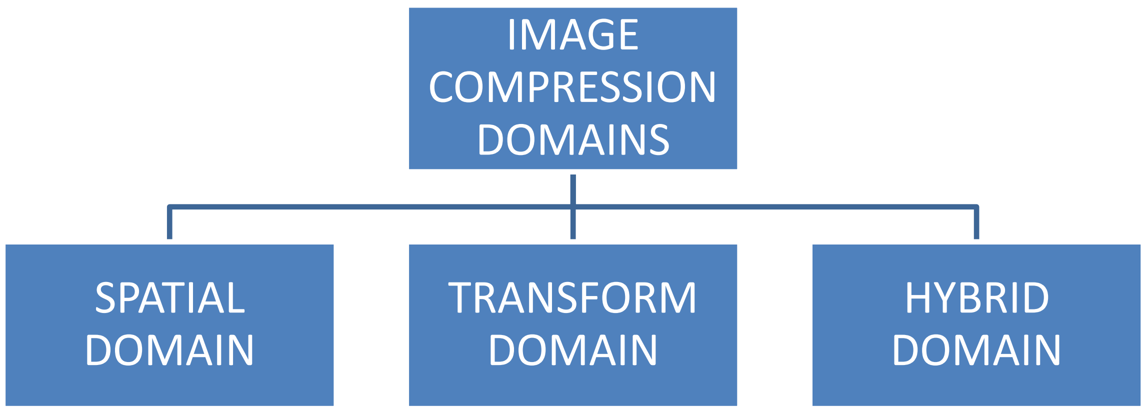

Figure 3.2: Image compression domains

3.3.1 Spatial domain

In spatial domain approach for digital image compression image data is directly tempered with to remove the less important component for the components that are repeated such that they are saved only once and thus remove the redundancy.

Spatial domain image compression techniques are easy to implement and require less memory and thus are widely used. Effect of noise is difficult to tackle in this domain or is more dominant. Some spatial domain digital image compression techniques are block truncation coding (BTC) and binary image compression (BIC) [3].

3.3.2 Transform domain

In this domain of image compression the input image data is transformed in the frequency domain. Since we know that our human eye are more sensitive to some data than other thus those irrelevant data if informed does not affect our image much. This particular concept is used in transform domain in which it becomes visibly plausible to differentiate the more sensitive data from less sensitive and thus only more sensitive data is stored ignoring the less sensitive data and thus saving much space.

Transform domain image compression techniques gives better compression ratio than spatial one [9]. Effect of noise is less in transform domain. Some transform domain image compression techniques are dct, dwt, haar wavelet etc.

3.3.3 Hybrid domain

This technique is the combination of spatial and transform domain image compression techniques. Both the domain techniques when used together gives the benefits of individual compression techniques and also increases the compression ratio and thus gives better performance. Nowadays more researchers are working more toward these hybrid compression techniques.

3.4 PROPERTIES OF IMAGE COMPRESSION SCHEMES

i. Scalability: It is defined as the reduction in the image quality due to changing the bit stream. Scalability is also known as progressive coding and embedded bit-streams. It is used to preview images when we are to download them. There are various forms of scalability:

ii. Region of interest coding: This includes coding some important parts of the image with high quality than others.

iii. Processing power: Different algorithms require different amount of processing powers for encoding and decoding.

CHAPTER 4

BLOCK TRUNCATION CODING

4.1 INTRODUCTION

Block truncation coding is a spatial domain compression technique. It is a very efficient and gives really good compression ratio with reliable PSNR [2]. It has also been used for video compression as well. It was discovered in 1789 by Prof. Edward J. delph and O. R. Mitchell.

According to this algorithm each pixel value is replaced by two mean value low and high depending upon the value of that pixel [4]. Image is divided in 4x4 blocks and than mean and standard deviation of each block is calculated and than two mean values are calculated that will be used further in decompression process.

4.2 BTC THEORY

At first image is divided into blocks of 4x4 with each block having pixel values such as $p_1, p_2 \dots\dots\dots., p_m$ [2]. Second step is to calculate the mean and standard deviation for each block. Mean and standard deviation is calculated as follows [2]:

$$\bar{Y} = \frac{1}{m}\sum_{i=1}^{m} p_i \qquad (1)$$

$$\overline{Y^2} = \frac{1}{m}\sum_{i=1}^{m} p^i \qquad (2)$$

Variance is given by:

$$\sigma^2 = \bar{Y}^2 - \overline{Y^2} \qquad (3)$$

Where Y is the mean value and Y^2 is the squared mean value and two safeguard these two values following two equations should be satisfied [4]:

$$m\bar{Y} = (m - q)a + qb \quad (4)$$

$$m\overline{Y^2} = (m - q)a^2 + qb^2 \quad (5)$$

Where m: total number of pixels in each block

q: number of pixels having values greater than or equal to mean in each block

a,b: two mean values

Next step is to calculate the two mean values that will be used in the reconstruction process which are given from above two equations as follows [4]:

$$a = \bar{X} - \sigma\sqrt{\frac{q}{m-q}} \quad (6)$$

$$b = \bar{X} + \sigma\sqrt{\frac{m-q}{q}} \quad (7)$$

For compressing image each pixel having value greater than mean is replaced by 1 and each pixel having value less than mean is replaced by 0 and transmitted or stored along with two mean levels a and b which are used while reconstructing the image. This collection of 1's and 0's forms bit plane BP [9].

Thus the compressed image is stored as set {BP,a,b}. Thus the compressed image have only values 1 and 0 for each pixel thus can be used using 1 bit only and since we have to save mean levels a and b values as well [4]. They require 8 bits each thus for one block of say 4x4 i.e. total 16 pixels 32 bits are required only instead of 128 bits as required for original image thus compressing the image to a quite extant.

4.3 BTC ALGORITHM

ENCODER:

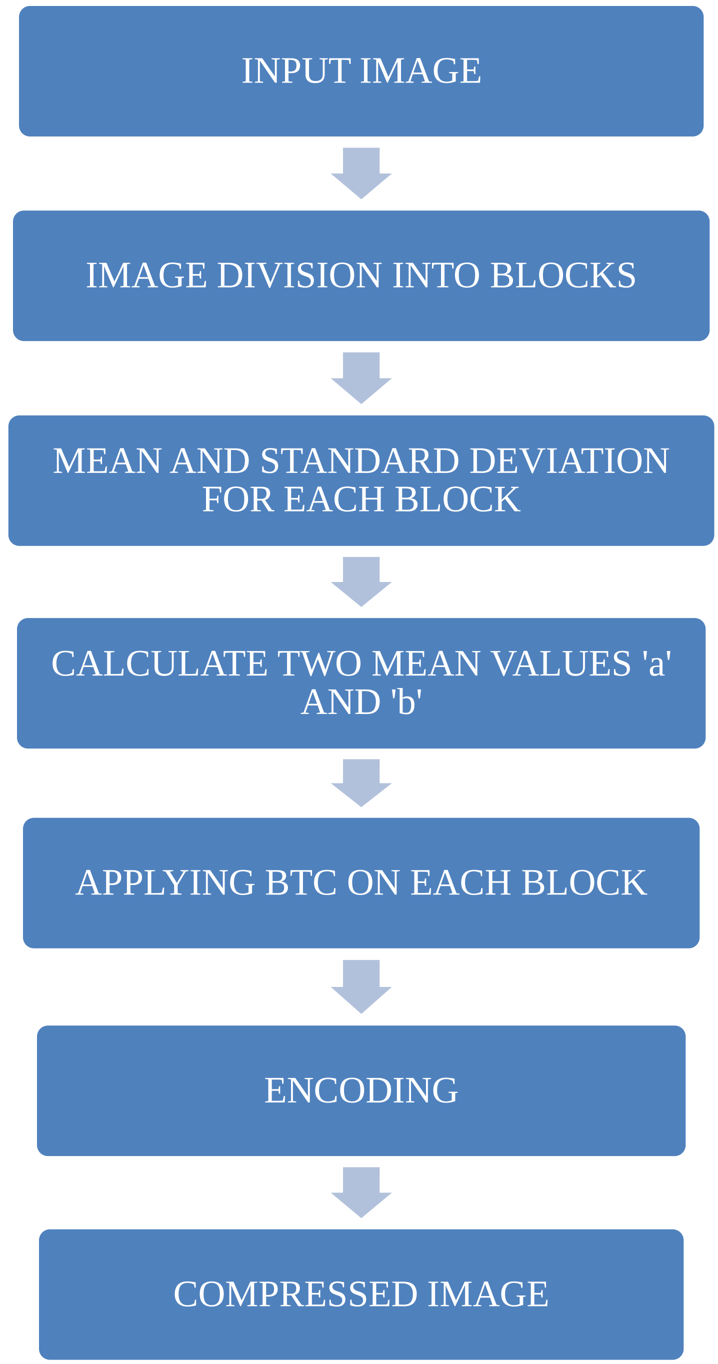

Figure 4.1: Image compression algorithm using BTC [2]

DECODER:

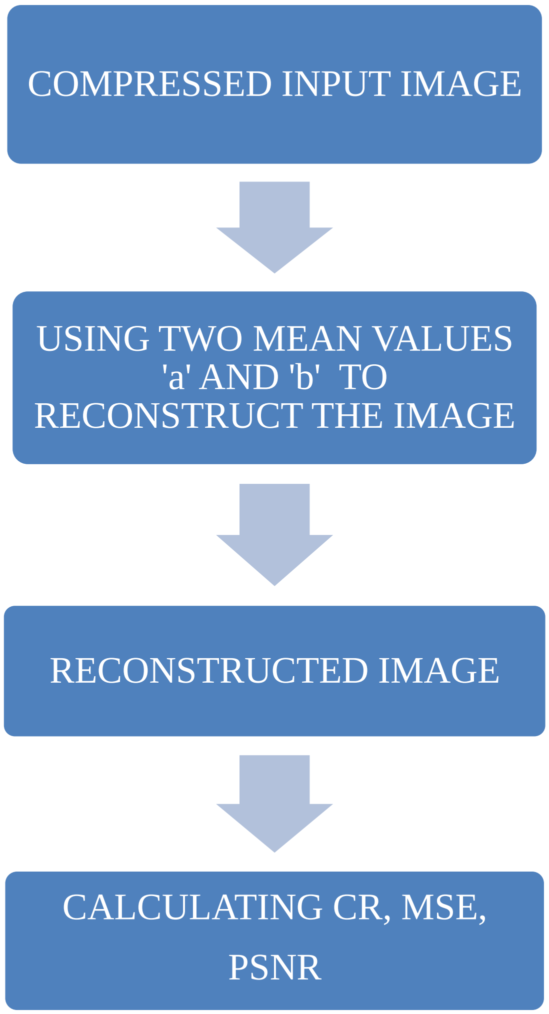

Figure 4.2: Decompression algorithm in BTC [2]

4.4 EXAMPLE

Let us consider a 4x4 block of an image be given below:

245	239	249	239
245	245	239	235
245	245	245	245
245	235	235	239

ENCODER:

- Calculate the mean and the two mean levels for the above matrix.
  Mean = 241.81
    a = 245.71
    b = 236.9
- Replacing the pixel value greater than mean with 1 and pixel value less than mean with 0, we get the compressed matrix as

1	0	1	0
1	1	0	0
1	1	1	1
1	0	0	0

- Now this matrix is saved using 1 bit along with ‘a’ and ‘b’.

DECODER:

- Reconstructed image is obtained by replacing 1 with ‘a’ and 0 with ’b’.

245	236	245	236
245	245	236	236
245	245	245	245
245	236	236	236

- Calculate the MSE, PSNR.

CHAPTER 5

DISCRETE COSINE TRANSFORM

5.1 INTRODUCTION

Discrete cosine transform is a fourier kind of transform and is somewhat similar to DFT but with an exception that it only uses real numbers [5]. It represents the input in terms of sum of the cosine variables at different frequencies. It can be seen as the real part of the fourier series since it only uses cosine function. DCT is used to convert the image from spatial domain into frequency domain [3].

Different frequency domain coefficients of the DCT have different importance level which is measured in the terms of variance level and these variance levels decide that whether that co-efficient can be neglected or not [5]. If a co-efficient have high variance level in an image than neglecting it can cause damage to our image. It is used in various applications such as image compression, video compression, image watermarking etc.

Cosine functions are used mostly in image compression techniques because only few variables of them can reconstruct an approximate original image [5]. DCT is an orthogonal transform and it has a defined number of functions. Advantages of using DCT are as follows [5]:

- It can transfer the energy to the lower frequencies in the image data.
- It can eliminate the blocking artifact phenomena.

5.2 DCT EQUATIONS

The below equations known as 2D-DCT equation calculates the pixel wise entry of dct in the image [5]:

$$d(i,j) = \frac{1}{\sqrt{2M}}\, C(i)C(j) \sum_{u=0}^{M-1} \sum_{v=0}^{M-1} P(u,v)\, cos\left[\frac{(2u+1)i\pi}{2M}\right] cos\left[\frac{(2v+1)j\pi}{2M}\right] \qquad (1)$$

Where

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}}, & if\ x = 0 \\ 1\ , & if\ \ x > 0 \end{cases}$$

P(u, v): is u,v^{th} the pixel of the image matrix

M: Size of the blocks in which the image is divided

For a block size of 8x8 the DCT equation is given by:

$$d(i,j) = \frac{1}{4}\ C(i)C(j)\sum_{u=0}^{7}\sum_{v=0}^{7} P(u,v)\ cos\left[\frac{(2u+1)i\pi}{16}\right] cos\left[\frac{(2v+1)j\pi}{16}\right] \qquad (2)$$

The 2D-IDCT equation is given by:

5.3 DCT MATRIX

Equation (1) is obtained in the matrix form by using the following equation [10]:

$$D_{x,y} \quad = \begin{cases} \frac{1}{\sqrt{M}},\ if\ x = 0 \\ \sqrt{\frac{2}{M}}\ cos\left[\frac{(2x+1)y\pi}{2M}\right], \quad if\ \ x > 0 \end{cases} \qquad (4)$$

For an 8x8 block the resulting matrix D is:

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

The first row of the matrix has all the values equal to $\frac{1}{\sqrt{8}}$ as given by the equation (4). The resulting matrix D is a orthogonal matrix which makes the inverse of D easy to calculate.

5.4 DCT ALGORITHM

Compression

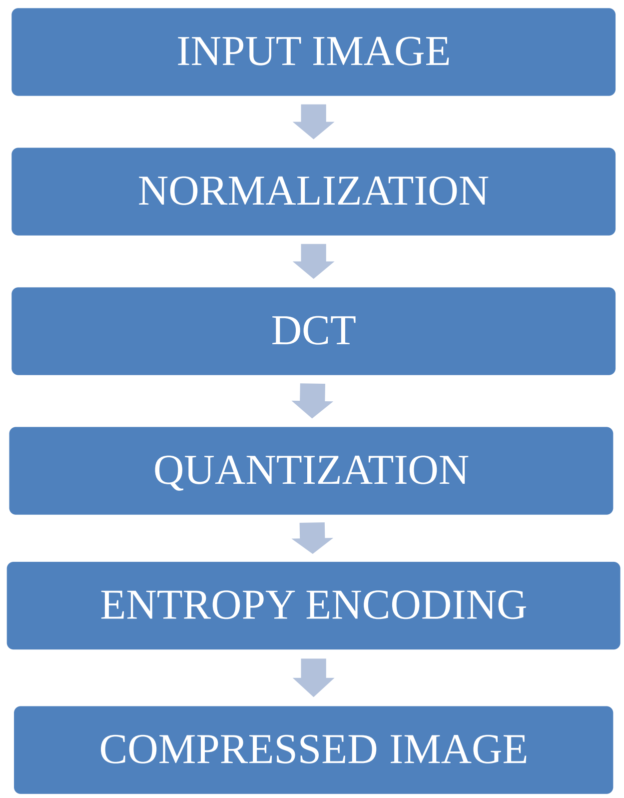

Figure 5.1: Compression algorithm for DCT [5]

NORMALIZATION: Each pixel in the input image has values between 0 and 255. Where'0' represents black and '255' represents white [5]. Now DCT is applied to the values between '-128' to 127, thus at first step we normalize the input image by subtracting '128' from each pixel value. Thus all the pixels of the input image will have value ranging between '-128' to '127'.

Normalized input = input image – 128

DCT: The steps performed to get the DCT of the image are as follows:

- Divide the image into 8x8 blocks.
- Starting from left to right and top to bottom apply DCT on each 8x8 block using DCT matrix.

Dct_Matrix = D*B*D'

where 'D' is the DCT matrix and 'B' is the 8x8 block on which DCT is applied.

QUANTIZATION: The next step is quantizing the obtain matrix after applying DCT. There are various quantization matrix available to achieve the required compression. The Q_{50} quantization matrix [10] provide both good compression as well as good quality of decompressed image, Q_{90} quantization matrix provide excellent compression but quality of decompressed image is not that good whereas Q_{10} provides less compression but good quality decompressed image.

The Q_{50} quantization matrix is given by:

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

Quantized matrix = rounded (Dct_Matrix / Q_{50})

Since we know that human eye is less sensitive towards high frequency components so when divided by quantization matrix and than rounded off will give zero for all the high frequency components [10]. Only low frequency components will be left which will than be saved and used in decompression process.

ENTROPY ENCODING: After quantization step most of the pixel in the quantized matrix will have zero values so we need to save only non zero values and neglecting zeroes. This saving is done by entropy encoding [12]. Mostly the zero values will be high frequency values that is the pixels away from (0,0) position(as the index increases the pixel values decreasing and finally most of them become zero). So the non zero values will be in the start index and the remaining will be tail of zeroes. So we will use zig-zag ordering to code the matrix [12].

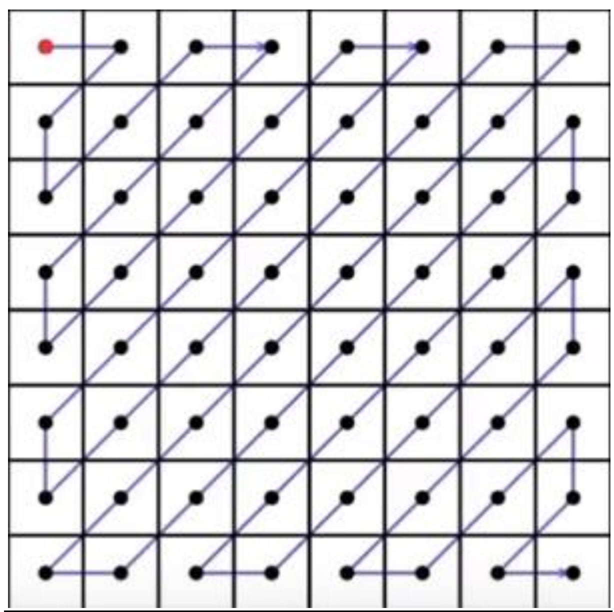

Figure 5.2: Zig-zag ordering used while encoding

The 8x8 block of the quantized matrix will be something like below matrix:

$$\begin{matrix} 100 & -60 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 13 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Applying the zig-zag ordering you will get the matrix in linear form as well:

[100,-60,0,0,0,0,6,0,0,0,13,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,,0,0,0,0,0,0,0]

Now the non zero values will be encoded using three terms as follows:

[(r,s),c]

Where **r:** Run-length which tells us the number of zeroes before the current in zero value

s: is the number of bits required to save the non zero value

c: is the actual value

(0,0): indicate that next are all zeros (end of block)

The above example will be encoded as follows:

[(0,7),100], [(0,6),-60], [(4,3),6], [(3,4),13], [(8,1),-1], [(0,0)]

Now since we know that the pixel value of quantized matrix can be 255 maximum. This implies that to encode this we require 11 bits at maximum i.e.

$$1 \leq s \leq 11$$

This implies that we only need '4' bits to encode **'s'**

Also maximum number of zeros between two non zero values will be 15 i.e.

$$0 \leq r \leq 15$$

So **'r'** will also need 4 bits to encode

The pair **(r,s)** will need 8 bits.

Decompression

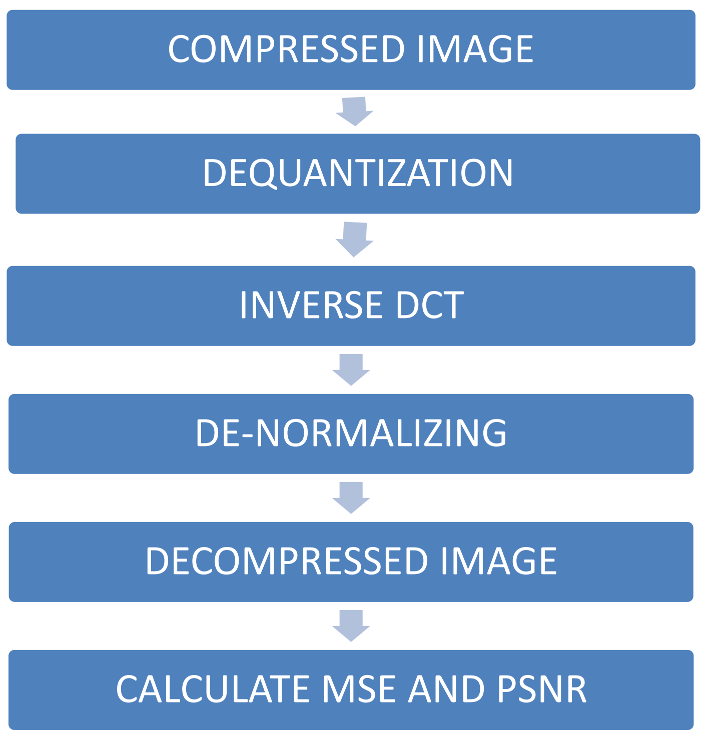

Figure 5.3: DCT decompression flow chart [5]

DEQUANTIZATION: The decompressed image is divided into 8x8 blocks and then starting from left to right and then top to bottom each block is multiplied by quantization matrix given below:

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

The resulting de-quantized matrix is given by:

De_quant = compressed image * Q_{50}

INVERSE DCT: The next step is finding the IDCT of the De_quant image by using the DCT matrix which is given below [5]:

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

IDCT_IMAGE = D’*De_quant*D

DE-NORMALIZATION: Since the DCT can be applied to pixel values between -128 and 127 only it was normalized by subtracting 128 during compression. Now during decompression it is de-normalized by adding 128 to its each pixel.

Decompressed image = IDCT_IMAGE + 128

CHAPTER 6

WALSH HADAMARD TRANSFORM

6.1 INTRODUCTION

The Hadamard transform belongs to the category of Fourier transforms. It is also known as the Walsh–Hadamard transform, Hadamard–Rademacher–Walsh transform, Walsh transform, or Walsh–Fourier transform [3]. It performs various operations which are orthogonal, symmetric, on 2^m real numbers or complex numbers, even though the Hadamard matrices are purely real on its own.

The Hadamard transform is said to be built from size-2 (DFTs), and is said to be similar to a multi-dimensional DFT of size 2 × 2 × ⋯ × 2 × 2 [3]. It disintegrates the input given into various walsh functions that when recombine form the input again. The transform is invented by French mathematician Jacques Hadamard, the German-American mathematician Hans Rademacher, and the American mathematician Joseph L. Walsh. The Hadamard transform is calculated in $O(p^2)$ steps ($p = 2^q$), using the walsh Hadamard transform algorithm [6].

6.2 WALSH HADAMARD EQUATION

The walsh-hadamard equation is given below [6]:

$$W_{2p+q}(t) = (-1)^{\lfloor p/2 \rfloor + q} \{ W_p(2t) + (-1)^{p+q} W_p(2t-1) \}$$

Where $\lfloor p/2 \rfloor$ is the largest integer which is less than and equal to p/2, q=0 or 1, p=0,1,2……..

$$W_0(t) = \begin{cases} 1 \ for \ \leq t < 1 \\ 0 \ elsewhere \end{cases}$$

The walsh functions obtained by using above equations are not orthonormal. Each wlash function needs to be normalised by either multiplying each walsh function with either 1/√T if it is the continuous case, or with 1/√N if it is the discrete case [7]. So we arrange them in lexicographic order and than obtain the hadamard Matrix.

The walsh function forms the rows and columns of the symmetric hadamard Matrix. These walsh function that form the rows and columns of the hadamard matrix are not in the increasing sequencies or number of zero crossing. They are arranged in hadamard order.

6.3 WALSH MATRIX

The walsh matrix is given below [6]:

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

WHT is the best known of the non sinusoidal orthogonal transforms. It has been widely used in digital image processing, because of its ease in applications. The basic walsh hadamard functions are square or rectangular waves having peak value of ±1 [6].

For 2-D images the obtained forward and backward walsh functions similar because the array obtained from these functions is a symmetric matrix that have orthogonal rows and columns, and since we know that orthogonal matrix have same inverse as original matrix so we can say that it has same inverse array as the array itself.

Walsh transform is calculated as [6]:

WHT = WM*IMG*WM'

6.4 ADVANTAGES OF WALSH TRANSFORM OVER HAAR TRANSFORM:

(i) Less computational complexity:- The matrices of the wash hadmard transform contains only ± 1 values [6] and the transforms are calculated by addition and subtraction only, since there is no multiplications required, which is considered to be the most time requiring step by a computer and decides computational time required so we can say that this method of obtaining the walsh functions require very less time compare to other transforms like dct requires multiplication which increases its computational time.

(ii) Less walsh functions can reconstruct the image in a better way than same number of haar function [6] because the major information about the image is stored in the first few walsh functions as shown below:

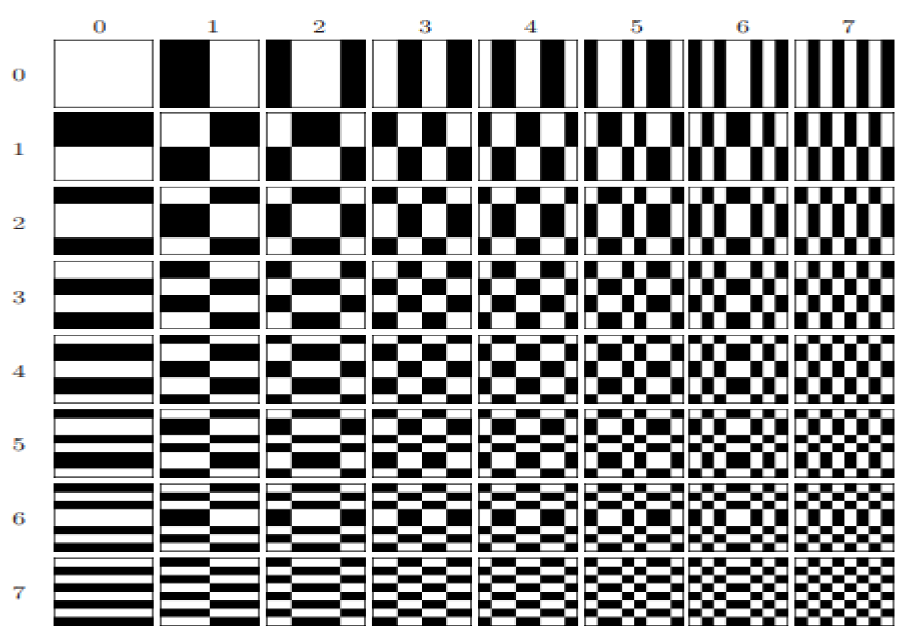

Figure 6.1: reconstruction of image using walsh functions

6.4 WHT ALGORITHM

Compression

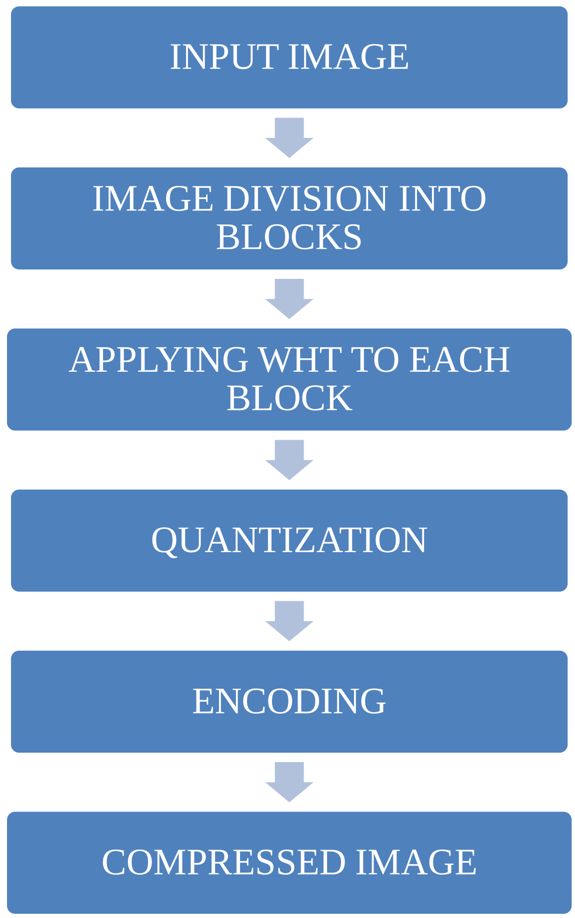

Figure 6.2: Flow chart of Walsh-hadamard transform [6]

IMAGE DIVISION: The first step is to divide the input image into blocks of 8x8 and than all the operations will be performed individually on each block.

WALSH HADAMARD TRANSFORM: After dividing the image now second step is to find the WHT of each block by multiplying each block with walsh matrix, this multiplication is done in three levels for each block for Y Cb Cr color coding to obtain equivalent transformed matrix corresponding to each color code. The transform is find using following equation:

WHT = WM*BLOCK*WM'

where WM is the walsh- hadmard matrix given as [6]:

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

QUANTIZATION: Quantization is done using the fact that our human eye is less sensitive toward the brightness levels of the high frequency components of the image. So using this fact we can round off the high frequency components to zero without affecting the image quality and thus need not to be saved in the compressed image. Different color components of the image have different quantization matrix which when divides the transformed matrix gives and rounded off gives the quantized matrix.

There are various quantization matrix available to achieve the required compression. The Q_{50}[6]quantization matrix provide both good compression as well as good quality of decompressed image, Q_{90} quantization matrix provide excellent compression but quality of decompressed image is not that good whereas Q_{10} provides less compression but good quality decompressed image.

Quantized matrix = rounded (Dct_Matrix /Q)

Where Q is the quantization matrix having different value for different color components.

The quantization matrix for luminance is given as [6]:

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

The quantization matrix for chrominance is given as [6]:

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

Encoding: After quantization step most of the pixel in the quantized matrix will have zero values so we need to save only non zero values and neglecting zeroes. This saving is done by entropy encoding [12].

Mostly the zero values will be high frequency values that is the pixels away from (0,0) position(as the index increases the pixel values decreasing and finally most of them become zero). So the non zero values will be in the start index and the remaining will be tail of zeroes. So we will use zig-zag ordering to code the matrix [12].

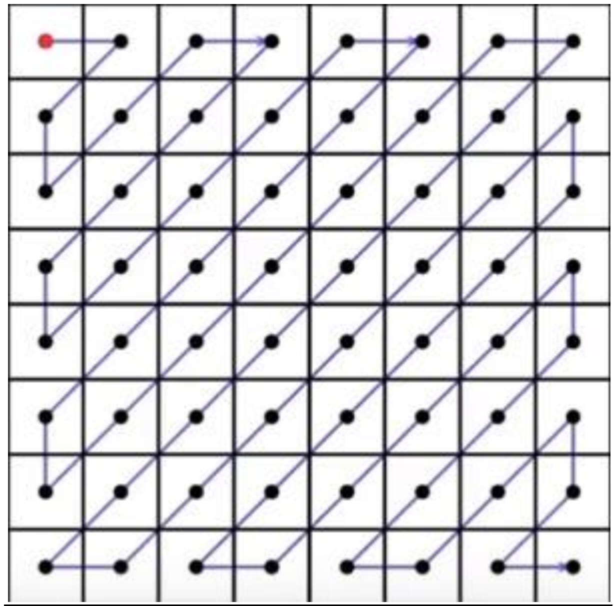

Figure 6.3: Zig-zag ordering used while encoding

The 8x8 block of the quantized matrix will be something like below matrix:

$$
\begin{matrix}
100 & -60 & 0 & 6 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
13 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
$$

Applying the zig-zag ordering you will get the matrix in linear form as well:

[100,-60,0,0,0,0,6,0,0,0,13,0,0,0,0,0,0,0,0,-1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,,0,0,0,0,0,0,0]

Now the non zero values will be encoded using three terms as follows:

[(r,s),c]

Where **r:** Run-length which tells us the number of zeroes before the current in zero value

s: is the number of bits required to save the non zero value

c: is the actual value

(0,0): indicate that next are all zeros (end of block)

The above example will be encoded as follows:

[(0,7),100], [(0,6),-60], [(4,3),6], [(3,4),13], [(8,1),-1], [(0,0)]

Now since we know that the pixel value of quantized matrix can be 255 maximum. This implies that to encode this we require 11 bits at maximum i.e.

$$\mathbf{1 \leq s \leq 11}$$

This implies that we only need '4' bits to encode **'s'**

Also maximum number of zeros between two non zero values will be 15 i.e.

$$\mathbf{0 \leq r \leq 15}$$

So **'r'** will also need 4 bits to encode

The pair **(r,s)** will need 8 bits

Decompression:

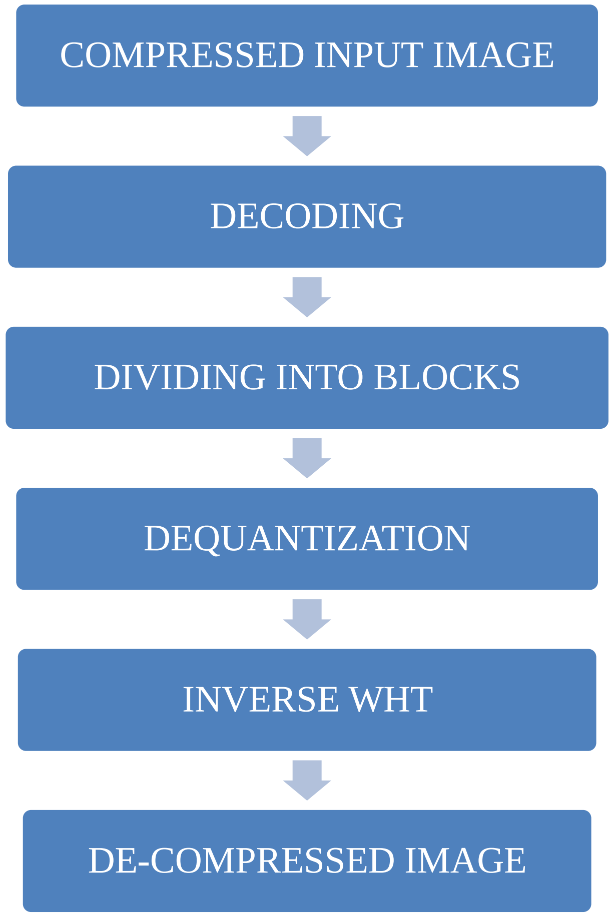

Figure 6.4 : Flow chart of decompression using WHT [6]

Decoding: The input compressed image is decoded and its matrix form is obtained.

Dividing into blocks: The matrix is divided into 8x8 non overlapping blocks. The matrix is than de-quantized by multiplying with different quantization matrix for different color components.

De-quantized matrix = compressed image * Q

Where Q for luminance levels is [6]

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

And Q for chrominance level is [6]:

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

Inverse WHT: The next step is to find the inverse walsh hadamard transform of the de-quantized image by multiplying it with walsh matrix and its inverse.

De-compressed image = WM'* de-quantized image*WM

Where walsh matrix WM is given as:

$$W = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

CHAPTER 7

IMAGE COMPRESSION USING HYBRID BTC AND DTC TECHNIQUE

7.1 INTRODUCTION

Hybrid technique refers to the combination of both spatial domain and frequency domain compression techniques. The spatial domain image compression techniques such BTC etc. provide hardware simplicity where as frequency domain image compression technique provides better compression. So combining both the techniques gives us both hardware simplicity and good compression ratio [1].

7.2 ALGORITHM

Compression

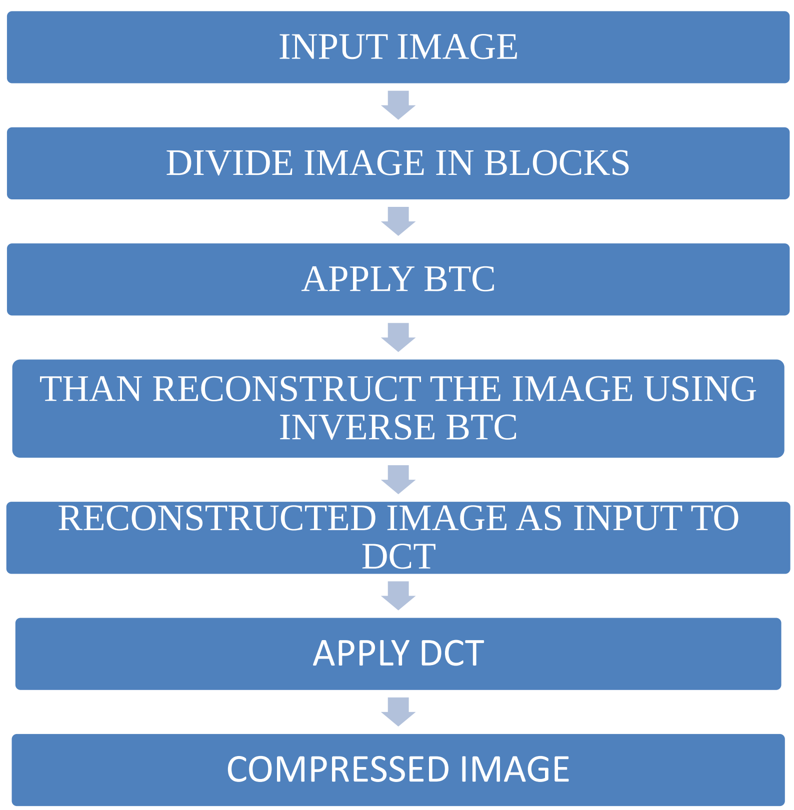

Figure 7.1: Compression Algorithm [1]

Decompression

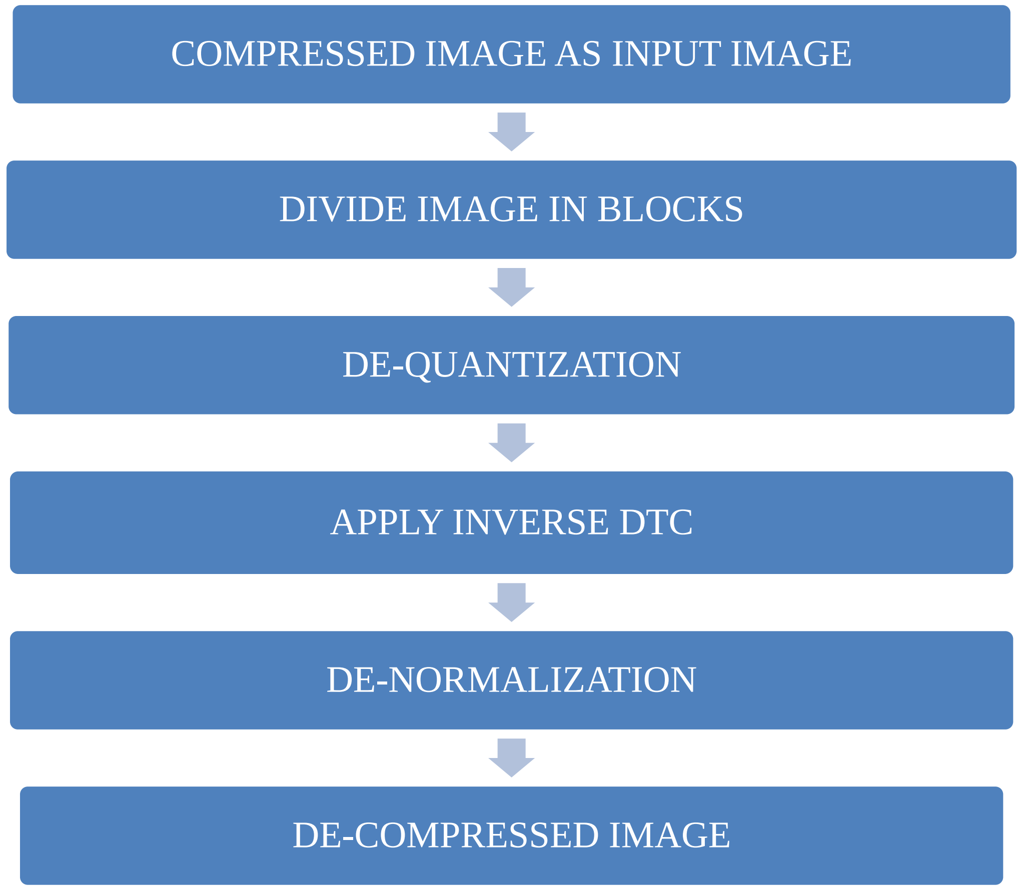

Figure 7.2: Decompression Algorithm [1]

CHAPTER 8

IMAGE QUALITY PARAMETERS

8.1 Compression ratio: It is the most important parameter to measure the image compression algorithm quality [18] [19]. It tells us about the compression efficiency of the algorithm. It is defined as the ratio of the compressed image to the uncompressed image. With increase of compression ratio image quality increases.

CR = uncompressed image size / compressed image size

8.2 Bit rate: It is defined as the number of bits required to store the each pixel value [18] [19]. It is ratio number of bits required to store compressed image to the number of pixels in the original image.

8.3 The mean squared error: It is evaluated as the difference in the pixel values of the compressed and uncompressed images [18] [19]. It is reciprocal of the PSNR value. With increase in the mean square value PSNR decreases. It indicates the quality of image. For lossless compression MSE is zero. It has some value for lossy compression. As the more compression is achieved MSE increases and thus PSNR decreases.

8.4 Peak signal to noise ratio: It is defined as the ratio of maximum signal power to the noise power [18][19]. It also measures the image quality. An image compression technique with high PSNR value for same compression ratio is considered better than the one with lower PSNR value. The formula for PSNR is given below:

$$\mathbf{PSNR = 10 \log \frac{255^2}{MSE}}$$

CHAPTER 9

RESULTS

Original image **decompressed image**

(a) (b)

Figure 10.1: (a) original image (b) decompressed images using btc

(a) (b)

Figure 10.2: (a) original image (b) decompressed image using DCT

(a) (b)

Figure 10.3: (a) original image (b) decompressed image using WHT

(a) (b)

Figure 10.4: (a) original image (b) decompressed image using hybrid BTC and DCT

Original image **Decompressed image**

(a) (b)

Figure 10.5: (a) original image (b) decompressed image using BTC

(a) (b)

Figure 10.6: (a) original image (b) decompressed image using DCT

Original image **Decompressed image**

(a) (b)

Figure 10.7: (a) original image (b) decompressed image using WHT

(a) (b)

Figure 10.8: (a) original image (b) decompressed image using hybrid BTC and DCT

Original image **Decompressed image**

(a)

(b)

Figure 10.9: (a) original image (b) decompressed image using BTC

(a)

(b)

Figure 10.10: (a) original image (b) decompressed image using DCT

Original image **Decompressed image**

(a)

(b)

Figure 10.11: (a) original image (b) decompressed image using WHT

(a)

(b)

Figure 10.12: (a) original image (b) decompressed image using hybrid BTC and DCT

Image	BTC	DCT	WHT	HYBRID BTC AND DCT
MAN	2.15	1.8	1.41	2.75
SANTA	1.84	1.68	1.66	2.4
FIELD	1.73	1.48	1.45	2.15

Table 10.1: Comparison of Compression ratio for various image compression techniques

Image	BTC	DCT	WHT	HYBRID BTC AND DCT
MAN	19.02	20.11	54.2	14.276
SANTA	28.73	32.55	34.8	32.7
FIELD	21.5	25.6	29.7	20.03

Table 10.2: Comparison of MSE for various image compression techniques

Image	**BTC**	**DCT**	**WHT**	**HYBRID BTC AND DCT**
MAN	**39.74**	**32.46**	**27.52**	**31.76**
SANTA	**28.85**	**29.61**	**26.89**	**29.707**
FIELD	**34.67**	**43.8**	**23.4**	**30.878**

Table 10.3: Comparison of PSNR for various image compression techniques

CHAPTER 10

CONCLUSION AND FUTURE SCOPE

Image compression based on hybrid technique of BTC and DCT provides good Compression ratio along with affordable mean square error and peak signal to noise ratio. This technique combines a spatial domain image compression technique (BTC) and a transform domain image compression technique (DCT). Comparison among various other image compression techniques show that this technique provides better results.

References

1. Markandeya, Nehal, and Sonali Patil. "Digital image compression hybrid technique based on block truncation coding and discrete cosine transform." Trends in Electronics and Informatics (ICEI), 2017 International Conference on. IEEE, 2017.
2. Ghrare, Seddeq E., and Ahmed R. Khobaiz. "Digital image compression using block truncation coding and Walsh Hadamard transform hybrid technique." Computer, Communications, and Control Technology (I4CT), 2014 International Conference on. IEEE, 2014.
3. Vijayvargiya, Gaurav, Sanjay Silakari, and Rajeev Pandey. "A survey: various techniques of image compression." *arXiv preprint arXiv:1311.6877* (2013).
4. Delp, Edward, and O. Mitchell. "Image compression using block truncation coding." *IEEE transactions on Communications* 27.9 (1979): 1335-1342.
5. Watson, Andrew B. "Image compression using the discrete cosine transform." *Mathematica journal* 4.1 (1994): 81.
6. Khin Thida Win, Nang Aye Aye Htwe. "Image compression method using Walsh-Hadamard transform." U.S. Patent No. 4,549,212. 22 Oct. 2014.
7. Shen, Jau-Ji, and Hsiu-Chuan Huang. "An adaptive image compression method based on vector quantization." Pervasive Computing Signal Processing and Applications (PCSPA), 2010 First International Conference on. IEEE, 2010.
8. Jassim, Firas A., and Hind E. Qassim. "Five modulus method for image compression." arXiv preprint arXiv:1211.4591(2012).
9. Chandrasekhar, VK Padmaja B. "Literature Review of Image Compression Algorithm." *IJSER,(3)* (2012): 1-6.
10. Pujar, Jagadish H., and Lohit M. Kadlaskar. "A new lossless method of image compression and decompression using Huffman coding techniques." Journal of Theoretical & Applied Information Technology 15 (2010).

11. Dwivedi, Ashutosh, et al. "A novel hybrid image compression technique: wavelet-MFOCPN." Proc. of 9th SID. Vol. 6. 2006.
12. AHMED, N., T. NATARAJAN, and KR RAO. "On image processing and a discrete cosine transform', ibid., 1974." *C-23*: 90-93.
13. Wallace, Gregory K. "The JPEG still picture compression standard." IEEE transactions on consumer electronics 38.1 (1992): xviii-xxxiv.
14. Delp, Edward, and O. Mitchell. "Image compression using block truncation coding." IEEE transactions on Communications 27.9 (1979): 1335-1342.
15. Kishimoto, T., et al. "Block coding of still pictures." Technical Group of Communication System of the Institute of Electronics and Communication Engineers of Japan (1978): 63-69.
16. Somasundaram, K., and I. Kaspar Raj. "Low Computational Image Compression Scheme based on Absolute Moment Block Truncation Coding." World Academy of Science, Engineering and Technology 19 (2006): 166-171.
17. Gonzalez, Rafael C. "Digital image processing." (2016).
18. Eskicioglu, Ahmet M., and Paul S. Fisher. "Image quality measures and their performance." IEEE Transactions on communications 43.12 (1995): 2959-2965.
19. Yamsang, Nuntapong, and Somkait Udomhunsakul. "Image quality scale (IQS) for compressed images quality measurement." Proceedings of the international multiconference of engineers and computer scientists. Vol. 1. 2009.