ENHANCING QUALITY OF LOSSY COMPRESSED IMAGES USING MINIMUM DECREASING TECHNIQUES

باستخدام تقنيات إنقاص الحد الأدنى تحسين جودة الصور المضغوطة عن طريق الضغط المنقوص

By

Ahmed Lebanon Al Shami

Supervisor

Prof. Mohammad Otair

A Thesis Submitted for the Partial Fulfillment of the Requirement of the Master Degree in Computer Science

College of Computer Sciences and Informatics

Amman Arab University

May /2017

Authorization

AMMAN ARAB UNIVERSITY

Form (9) College of Scientific Research and Graduate Studies

Authorization

We, the undersigned, pledge to grant Amman Arab University for discretion in the publication of the academic content of the dissertation, so that the intellectual property rights of Master's thesis be back to the university in accordance with the laws, regulations and instructions relating to intellectual property and patent.

Advisor Name	Student Name
Prof. Mohammad Otair	Ahmed Lebanon Al Shami
Signature: Mohammad Otair	Signature:
Date: 5/6/2017	Date: 5/6/2017

Committee Members Decision

Committee Members' Decision

The thesis entitled "ENHANCING QUALITY OF LOSSY COMPRESSED IMAGES USING MINIMUM DECREASING TECHNIQUES" was submitted by the student Ahmed Lebanon Al Shami, was examined and approved on 28/5/2017.

Committee Members

Name		Signature
Prof. Mohammad Otair	Chair/Advisor	Mohammad Otair
Dr. Husam Ahmad Al Hamad	Member	
Prof. Asem Al-Shekh	External/ Member	

DEDICATION

To my father and mother, the reason behind all my successes, inspiration and motivation to be a better man

To my children, thanks for having you presence in my life

To my brothers, for their support and assistance without any condition or hesitation

To my professors, who gave me useful knowledge, and for being good example in their morals and guiding me

To all my family and loyal friends

To all who ever felt happy for me and supported me

To the only dream, I wish it will come true

Acknowledgments

I would like to express my gratitude to my supervisor Prof. Dr. Mohammad Otair for his great supports and patience, also for his smile every time I visit his office. I present my respect to Prof. Dr. Akram Al Mashaykhi the professor and the mentor who did not hesitate to advise me in every time and for any reason. I would like to thank Dr. Hussam Al-Hamad for his precious time and kind efforts helping me in this thesis.

List of Contents

contents

List of Tables

List of Figures

	PSNR improvement after using MDT on Gray 16-bit images
	MSE improvement after using MDT on color images
	MAE improvement after using MDT on color images
	PSNR improvement after using MDT on color images
	MSE improvement after using MDT on Gray 8bit images
	MAE improvement after using MDT on Gray 8bit images
	PSNR improvement after using MDT on Gray 8bit images
	MSE improvement after using MDT on Gray 16-bit images
	MAE improvement after using MDT on Gray 16-bit images
	PSNR improvement after using MDT on Gray 16-bit images
	MSE improvement after using MDT on color images
	MAE improvement after using MDT on color images
	PSNR improvement after using MDT on color images

List of Appendix

List of Equations

No	Subject
	Compression ratio (CR)
	Mean square error (MSE)
	Peak-to-Peak Signal to Noise Ratio(PSNR)
	Mean absolute error

List of Abbreviation

Acronym	Description
BMP	Bitmap
CR	Compression Ratio
DCT	Discrete Cosine Transform
GIF	Graphics Interchange Format
JPEG	Joint Photographic Experts Group
LZW	Encoding (Dictionary) developed by Lempel, Ziv and Welch
MAE	Mean Absolute Error
MDT	Minimized Decreasing Technique
MSE	Mean Square Error
PNG	Portable Network Graphics
PSNR	Peak-To-Peak Signal To Noise Ratio
RIFD	Rounding the Intensity Followed by Dividing
RLC	Run Length Coding
RLE	Run-Length Encoding

ENHANCING QUALITY OF LOSSY COMPRESSED IMAGES USING MINIMUM DECREASING TECHNIQUES

Prepared by

Ahmed Lebanon Al Shami

Supervised by

Prof. Mohammed Otair

Abstract

The acceleration in the development of technology has come with the urgent need to use a large amount of data, and the method of storing or transferring this information through various media in an important area, particularly in terms of quality preservation.

Digital image usage has become widespread in many fields, both scientific and commercial applications or social media, and it become improtant to process these images for optimal use within each field to maintain the quality of data used, especially during the compression process with the lossy technique.

This research aimed to improve the quality of compression in lossy techniques while preserving the compression ratio levels. This was accomplished by adding the proposed technique that has certain steps that depend on decreaseing the minimum pixels values from the pixels values inside the image, first by decreasing each row by its minimum values then dcreasing each coloumn in the resulted image by its minimum pixels value, finally dividing the image into (2×2) blocks then finding each block minimum value then decreasing it from the entire block.

Two methods of lossy compression techniques were selected RIFD and JPEG to be applied with the proposed technqiue that should precede these lossy techniques, by applying these techniuqes on diverse images with different extensions and sizes, the results were obtained and quality standards metrics MSE, MAE and PSNR were calculated, which are known mterics in the image compression field to evalute the compression and decompressed image quality by comparing with the original image before compression.

The proposed techqniue had reduced the error results from the compression that varied depening on the used lossy technique and the image charactersitcs which are its bitdepth, deimention and pixels values.

تحسين جودة الصور المضغوطة عن طريق الضغط المنقوص باستخدام تقنيات إنقاص الحد الأدنى

إعداد

أحمد لبنان الشامي

إشراف

الأستاذ الدكتور محمد عطير

الملخص

وقد جاء مع التسارع في تطوير التكنولوجيا الحاجة الملحة لاستخدام كمية كبيرة من البيانات، وان طريقة تخزين أو نقل هذه المعلومات من خلال الوسائط المختلفة هي مجال مهم، وخاصة من ناحية الحفاظ على الجودة. ومن هذه البيانات الصور الرقمية والذي أصبحت واسعة الانتشار في العديد من المجالات، سواء كانت في التطبيقات العلمية أو التجارية أو وسائل التواصل الاجتماعي، وأصبح من المهم معالجة هذه الصور للاستخدام الأمثل داخل كل مجال للحفاظ على جودة البيانات المستخدمة، وخصوصا أثناء عملية الضغط باستخدام تقنية الضغط المنقوص.

يهدف هذا البحث إلى تحسين نوعية الضغط في تقنيات الضغط المنقوصة مع الحفاظ على مستويات معدل الضغط. وتم إنجاز ذلك بإضافة التقنية المقترحة التي تحتوي على خطوات معينة تعتمد على تقليل قيم الصورة بقدر القيم الدنيا، ويتم ذلك أولا بتخفيض كل صف في الصورة بمقدار القيمة الدنيا لنقاط الصف ثم تخفيض قيم كل عمود في الصورة الناتجة بمقدار الحد الأدنى لقيم كل عمود. ومن ثم تقسيم الصورة إلى مصفوفات أصغر مربعة ذات أبعاد (2×2) والعثور على كل قيمة الحد الأدنى لكل مربع ثم خفضه من القيم بأكملها.

تم اختيار طريقتين لتقنيات الضغط المنقوصة RIFD و JPEG ليتم تطبيقها مع التقنية المقترحة التي يجب أن تسبق تقنيات الضغط المنقوصة، ومن خلال تطبيق هذه التقنيات على صور متنوعة ذات امتدادات وأحجام مختلفة، تم الحصول على النتائج مع التركيز على مقاييس ومعايير الجودة (MSE, MAE, PSNR) والتي هي معايير معروفة في مجال معالجة الصور وخاصة في تخصص ضغط الصور والتي تستخدم لتقييم طريقة ضغط الصورة من ناحية الجودة مقارنة مع الصورة الأصلية قبل الضغط.

وقد خفضت التقنية المقترحة مدى الخطأ الناتج من عملية الضغط بتبيان التخفيض اعتمادا على تقنية الضغط المنقوصة المستخدمة وخصائص الصورة التي هي عدد المراتب الثنائية المستخدمة لقيم النقاط bitdepth، أبعاد الصورة وقيم نقاط الصورة ككل.

CHATER ONE

INTRODUCTION

Introduction

A Digital Image is a digital representation of a real scene that is obtained by using scanners, digital photography devices or radiography devices. This image can be easily used or processed inside computational devices, which is called digital image processing.

As the digital image became widely used, image processing became a necessary field in many areas and for numerous reasons such as medical imaging, social media, communications and security cameras. In the area of digital image processsing the input for each process is an image, where the output can be an image or a certain attribute or information associated with the original or processed image that results after particular one or several processes applied on the original image (Hlavac, 2011, Young, et al., 1998).

Many digital image processing techniques support image distinctive features of interest. After that they extract valuable information about the scenery from the enhanced image. These processes could be achieved by digital computer methods over suitable algorithms for particular purpose which is implemented on digital images (Philip, et al., 2011).

Image compression means minimizing size of the images file without effect on the quality of the image. Therefore; the aim of compression is reduce the image size that allowing the amount of disk or memory space to store more images. In addition, it reduces the time required for images to be sent or downloaded over the Internet (Chowdhury, andKhatun, 2012).

This thesis focuses on the compression technique; it utilizes a preprocessing lossless procedure prior to a lossy technique which produces a hybrid technique that merges two techniques from the lossless and lossy categories. The intent of combining these techniques is to get benefit from high compression ratio in the lossy techniques and low distortion ratio in the lossless techniques.

Problem statement

The importance of the digital images and transmitting them along all networks either wired or wireless is increasing day-by-day. These images have very huge sizes in general and need to be compressed in order to accelerate transmitting process.

The compression techniques are categorized into Lossless and Lossy. The first one has a good performance on the quality of the compressed images with no loss of any part of the images. However when we evaluate these techniques in terms of the compression ratio, we will find that they have very low performance comparing with the lossless techniques. On the other hand, lossy techniques have high distortion rates. So, most of the current researchers in this field try to find a good level of hashing both techniques, preserving or improving the compression ratio and reducing the distortion.

Research questions

The current study raises the following two questions

What is the impact of the merging between both lossy and lossless techniques?

Is the technique improving distortion rate by using only lossy techniques?

Significant of the study

The proposed technique can be applied on the digital images in general whether it was color or gray scale with various Bit-Depth values, and it is a mix between lossless and lossy techniques.

The proposed technique provides a good compression ratio and lower distortion values, to get the best features from each technique that has been used before merging them. Moreover, this thesis aimed to reduce the Bit-Depth (maximum number of bits to represent the pixel) along with reducing the error and improving the quality of the comparison process. All of the above were achieved by adding the proposed lossless technique.

Research model

Proceeding with a technique that keeps every bit value after been processed and then followed by a lossy technique will reduce distortion and could enhance improve the compression ratio. Figure (1) shows the detailed steps of the proposed compression phase.

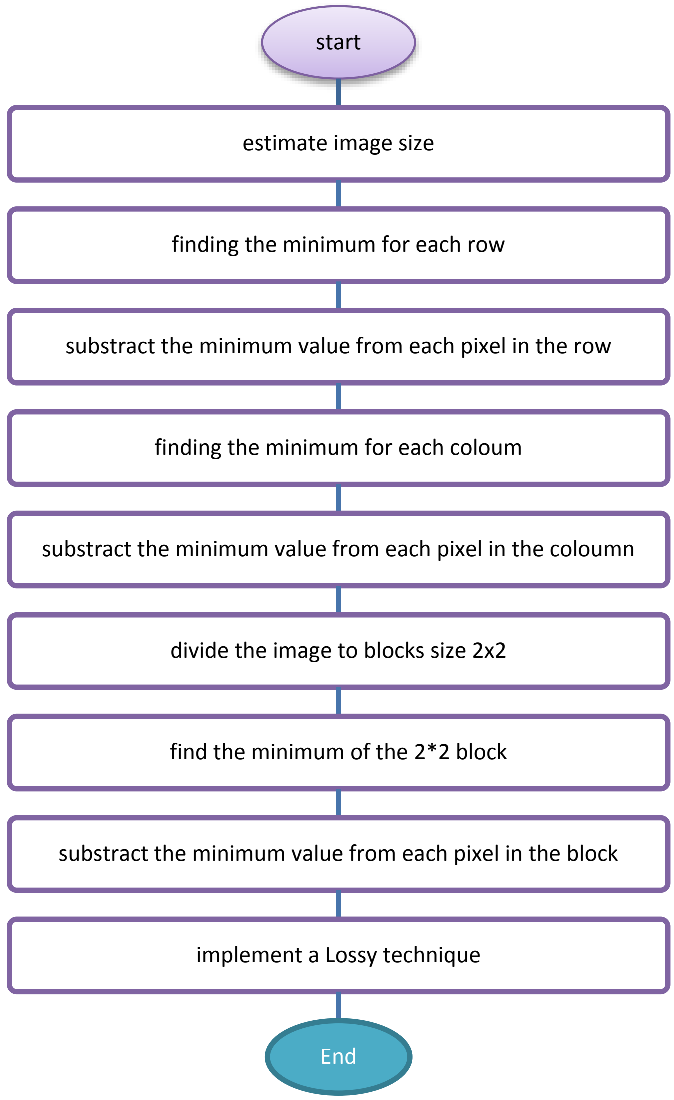


Figure (1): Proposed Technique Compression Algorithm

Reversing the compression process in the right sequence called de-compression which is shown in the Figure (2).

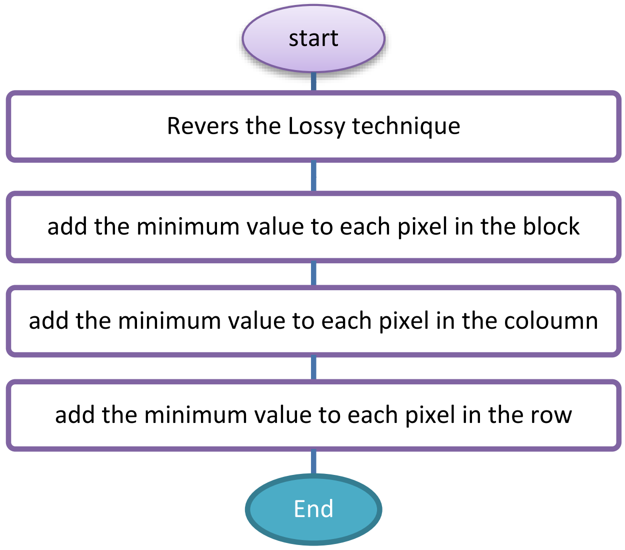


Figure (2): Proposed Technique De-Compression Algorithm

Limitation

One of the most important difficulties, the researcher faced during the preparation of this thesis, is the lack of clear studies in the subject of thesis. The exiting studies indicated and distinguished the topics in general, many information were not sufficient to enrich this study. Therefore, a big amount of time was required to collect the useful information from the literature.

This thesis depends on specific mathematical equations and algorithms that should be applied to experimental results, and checking the validity of these results to be within the acceptable range of values. The writing of the programming code went through several stages to ensure applying the proposed technique correctly on the test images, by checking the programming errors and the code sequence which sometime had to be changed for experimental purposes

Examining more than one image to calculate the results and assess the technique considering the type of image and the appropriate way to apply the technique.

Collecting the test images specially that some images with specific bit-depth cannot be easily found

Thesis organization

This thesis includes five chapters:-

Chapter one declares the main problem in image processing from the researcher perspective which is represented in compression techniques.

Chapter two explains the main concepts of the image compression and presents some of the literature reviews that are related to the core subject of the thesis.

Chapter three presents the proposed methodology that has been followed in the practical part of the thesis and represents it through algorithms and flowcharts.

Chapter four presents a number of experiments on images and evaluating each experiment and display its results.

Chapter five summarizes the main conclusions of the thesis and suggesting recommendations for future.

CHAPTER TWO

BACKGROUND

Introduction

This chapter will discuss the main concepts of the image process especially the image compression and the quality metrics that assess the compression, as well as presenting some of the literature reviews that are related to the core subject of the thesis.

Survey

In recent years, there has been an explosive growth in digital imaging technology and applications .The digital images and video spread widely in almost all parts of our daily life details and widely distributed either over the Internet or via memory drivers by in using PCs, mobile applications, websites, newspapers, academic lectures, advertising and personal photos (Lyon, 2006; Wolfgang et al, 1999).

The digital image can be defined as a two-dimensional function, that its variables represent the coordinates. Function’s values are called intensities or gray levels of the image. When the amplitude values of this function are all limited and discrete quantities, it is called a digital image. (Shinde and Dani, 2011).

The two-dimensional function values can be represented by a matrix. Each matrix elements are considered as picture element or image element or the most common used term is a pixel. Each image might have thousands, but always finite numbers of pixels, and the more pixels the image has more accurate it will be (Shinde and Dani, 2011; Abdalla and Osman, 2016).

As mentioned above, the value of each pixel is the intensity, and the amount of bits required in each pixel to represent this intensity is the Bit-Depth of the image as shown in Figure (3), which varies according to the image type and characteristics for example if the image has 8-bitdepth then the values of its pixels will vary between 0-255, where for 16-bitdepth the maximum value will be (65535), and the main types of digital image are (Kuppusamy et al, 2013):

• Binary Image: that appears only in black and white and its pixel value is either zero or one.

• Grayscale Image: this image contains only black and white color and all the color gradations between them. The pixel value varies between 0 and 255, each pixel represented by eight bits to express the colors scale.

• Color Image: this image type could support colors, where each pixel has three parts. Each part contains eight bits used to measure the intensity of the particular basic color Red, Green and Blue.

Digital Images can be represented through different image files such as GIF, JPEG, BMP, PNG…etc.

178	180	179	183	180	181	184	180	179	182	181	182
179	183	181	182	182	184	185	184	182	184	182	184
181	181	181	184	183	184	182	186	184	184	181	181
179	181	179	181	181	181	182	181	182	183	180	181
181	178	178	181	178	181	180	183	181	182	180	181
181	178	180	174	179	180	174	179	184	178	178	178
179	183	181	183	[illegible]	[illegible]	[illegible]	[illegible]	181	182	181	181
180	182	178	182	[illegible]	[illegible]	[illegible]	[illegible]	181	182	180	182
182	185	184	184	180	183	183	184	184	185	183	182
181	180	180	182	181	183	181	183	184	184	184	184
182	180	177	182	180	183	179	181	181	183	180	181
178	179	179	181	183	178	183	181	182	180	183	181

$f(4,5) = 181$

$181_{10} = (10110101)_2$

Figure (3): Sample Data from a Digital Image

The processing of a digital image is a leading subject, and exists in several domains like photography, satellite imaging, medical imaging, and image compression.All processes on digital images focus on two main tasks (Kuppusamy, et al, 2013):

Improvement of pictorial information for human interpretation.

Processing of image data for storage, transmission and representation for autonomous machine perception some argument about where image processing ends and fields such as image analysis and computer vision start.

Here are some of the standard Image Process operations, which show the different problems or required actions, which can be solved by performing one of the digital image processes:

Geometry transformations such as amplification, reduction, and rotation

Color repairs such as brightness and contrast adjustments, color mapping, and color stabilize.

Image Compression.

Digital blending or optical compositing.

Image distinguish and morphing.

Image perception, like obtain the text image.

Image segmentation.

Image Compression

The use of digital images is grows continually, along with this increasing use of digital images becomes the serious issue of storing and transferring the huge volume of data representing the images because of the uncompressed multimedia (graphics, audio and video) data requiring considerable storage capacity and transmission bandwidth. (Talukder and Harada, 2010)

Thus, compressing images becomes significant task to minimize storage space and transport time and use of bandwidth or even to qualify rapid browsing and recovery of imagery from databases, in addition to, using digital image data can help display the additional feature for choosing computer proper format which can be stored and analyzed (Poobal and Ravindran, 2011; Vidal and Amigo, 2012).

Image compression is the process of detraction the image size without affecting the image type or it will be influenced by reasonable level according to the purpose the compression is used for. Compression methods are divided into two major categories, Lossy and Lossless compression. (Otair and Shehade, 2016; Sharma, 2010).

Lossy compression indicates that some amount of image data is lost when it is decompressed. This compression technique is based on the proposition that the stream data files save more information than the human beings can "realize”. Therefore, the irrelevant data could be removed, such as in videos or images when transferred through mobile social applications because they are intended for human interpretation.

Lossless compression indicates that while the data is decompressed, the outcome is a bit-for-bit ideal match with the original one, the name lossless indicates that “no data is lost". The data is just saved more efficiently in its compressed form but nothing of it is removed. (Sharma, 2010)

The compression ratio can be measured by dividing the compressed image size by the original image size as in the following formula. In general, this ratio is low in Lossy compression techniques, while it can be high in Lossless compression techniques. (Wang and Li, 2011; Hore and Ziou, 2010).

$$Compression\ ratio = \frac{compressed\ image\ size}{original\ image\ size} \quad \text{................Equation (1)}$$

Obviously the best compression ratio value is whenever it’s away from one and close to zero, since the best compression technique produces a compressed image that is much smaller than the original image size.

The most popular compression techniques according to their types include:

Some of Lossy compression techniques:

Transforms

Fourier transforms.

Discrete cosine transforms

Haar, Walsh, hadmard

Some of Lossless compression techniques:

Run Length Encoding

Huffman Coding

LZW (Lempel, Ziv and Welch)

Quality Evaluation Metrics

Important image waste of information or property that may take place during the different image processing; Therefore, Quality Assessment (IQA) is deeply essential characteristic of evaluating image quality after been processed compared with the original image to ensure that any particular process is performing the required results, as in image compression it required to check the variation between the original and the processed Image (Samajdar and Quraishi, 2015), the widely popular IQA techniques are as followed, which is also shown in figure (4):

Objective: include Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Mean Sequare Error (MAE).

Subjective: this type gets along with human visual system (HVS).

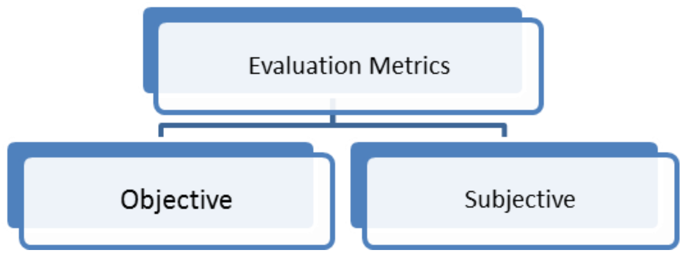


Figure (4): Main Evaluations Metrics types

The broad diversity of distortions that images are revealed to during acquisition, processing, storage, and reproduction can downgrade their perceived quality.

Subjective evaluation is time-consuming, costly and resource-intensive. Therefore, objective ways of evaluation have been proposed. (Pedersen and Hardeberg, 2012).

Objective image feature metrics can be grouped according to the availability of an original (distortion-free) image, in which the distorted image is to be compared. Most of the current approaches; used in this thesis; and known as full-reference, indicating that an entire reference image is assumed to be associated (Kumar and Rattan, 2012).

Mean Square Error (MSE)

Is the cumulative difference between the compressed image and original image [11].

$MSE = \frac{\sum_{M,N}[I_1(m,n)-I_2(m,n)]^2}{M\times N}$ ……………Equation (2)

Where: - M,N are pixel co-ordinate

I1: compressed image pixel

I2: original image pixel

Peak-to-Peak Signal to Noise Ratio (PSNR)

The rate within largest possible power and effective distorted noise on image impersonation (Kaushik and Sharma, 2012):

$PSNR = \frac{10\times\log_{10}(Intensity_{max})^2}{MSE}$…… Equation (3)

For 8-bit pixel gray scale, Intensitymax =255 ,

$$PSNR = \frac{10 \times \log_{10} 255^2}{MSE}$$

The PSNR rate reaches infinity, where MSE decreases to zero; this reveals that a higher PSNR value presents a better image quality (Wang and Li, 2011).

Experimental results made by many researchers indicate that MSE and PSNR are easy to execute, and have low computational intricacy. In image comparison both mentioned above metrics are good, when slight differences in distortion of a particular type (ECE and Mullana, 2011).

Mean Absolute Error (MAE)

This metric represent the cumulative absolute value for the variance between the initial image and the refined one.

$MAE = \frac{\sum_{M,N}|I_1(m,n)-I_2(m,n)|}{M\times N}$…………..Equation (4)

Where: - M,N are pixel co-ordinate

I1: compressed image pixel

I2: original image pixel

Joint Photographic Experts Group (JPEG)

This is a well-known and efficient lossy compression method standardized by the Joint Photographic Experts Group (JPEG). In the late Eighties, JPEG has chosen three from twelve proposed methods aiming to select an exquisite process depending on the blind estimation of subjective image quality. Next year they declared that the “DCT” proposal was the most accurate method; by using the 8x8 DCT it produced excellent compression with best picture quality among the proposed methods.

The JPEG has been suggested as a standard compression scheme for continuous-tone motionless images. It utilizes a 64 (8 by 8) pixel-block discrete cosine transform (DCT) for gathering the information into several transform coefficients, this block design takes the advantage of the local spatial correlation property of images and also reduce the processing time. Yet, it is well known that this individual processing of each block will create visually disturbed blocking effects, especially when a high quantization parameter is used for high compression (Huang, et al., 2012).

Several types of blocking effects on JPEG-decompressed images one is the escalator or staircase noise on image edges appear when an eight by eight block that includes an image edge, this edge degrades so that the block frame looks like the edge. The other turbulence is the grid noise in the monotone area, can be described as a slight change of image intensity along the DCT block boundary, it is easily noticeable in the monotone area. Finally, the corner outlined in the corner point of the DCT block, which is clearly visible. Since the corner point is either much larger or much smaller than neighboring pixels (Fidler, et al., 2014).

Huffman

It is a method of formation a minimum redundancy code. Dr. David A. Huffman proposed this technique in 1952 from which its name derived. Huffman coding is a commonly used scheme for data compression because due to its simplicity and effectiveness. This technique uses the same coding table in both the encoder and the decoder, depending on statistical information about the frequency of the data to be encoded.

Huffman's greedy algorithm looks at the appearance of each character and saves it as a binary string in the best way. Huffman coding is a method of statistical coding which tries as much as possible to reduce a number of bits that is required to represent a string of symbols in the encoded data (Srikanth, and Meher, 2013).

The algorithm shorter codes are specified to the most frequently used symbols, while longer codes to the symbols which appear less frequently in the string (that's where the statistical part comes in). Code word lengths are no longer fixed like ASCII .Code word lengths vary and will be shorter for the more frequently used characters.

Huffman Coding Algorithm is presented in three main steps: the first step is Building a Huffman tree by sorting the histogram and successively combines the two bins of the lowest value until only one bin remains. The second step is encoding the Huffman tree and save the Huffman tree with the coded value. The third step is encoding the residual image (Dubey, and Singh, 2012).

Rounding the Intensity Followed by Dividing (RIFD)

This new compression method was chosen along with a well known one to evaluate the proposed technique concerning improving its quality metrics, and that the proposed technique is suitable to process before different kind of lossy techniques.

A Novel lossy image compression technique called RIFD for compressing images. (Otair and Shehade, 2016) . This scheme leans on increasing the redundancy and resemblance among the close pixels of images by rounding the pixels' intensities followed by the dividing process, which makes compression attainable, the main idea of the RIFD algorithm is based on two facts:

Adjacent pixels are correlated or so identical.

The human sight can perceive a very limited number of intensities. So; if the intensity values of the adjacent pixels are rounding to the same value, then the redundancy will increase, and the updated intensity values will not be detectable by human sights. Raising the information redundancy supports the image to be more compressed. Therefore, finding a less correlated representation of the image is a significant thing.

This technique can be implemented either individually or beside any lossless compression algorithm. The RIFD technique can be implemented as shown in the following figure (5)

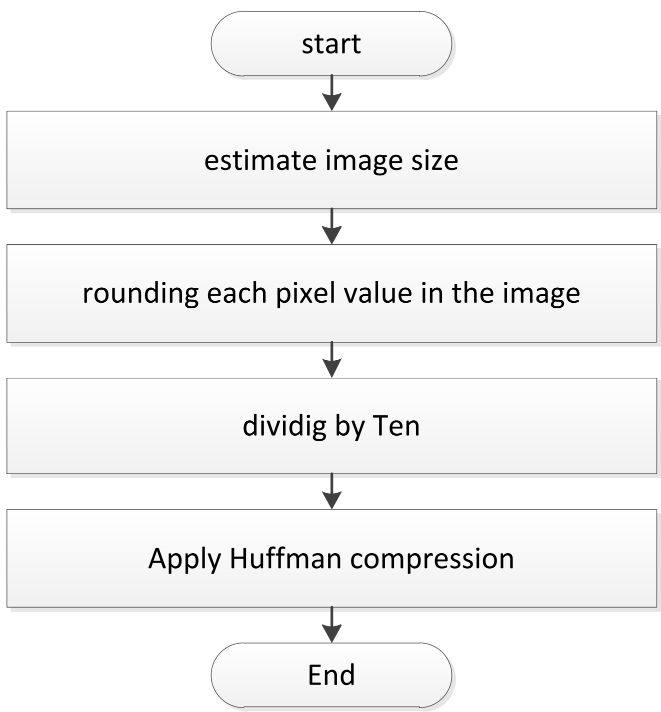


Figure (5): compression in RIFD technique

While the decompression can be applied through simple step by reversing the Huffman compression followed by multiplying each pixel by ten as shown in the Figure (6).

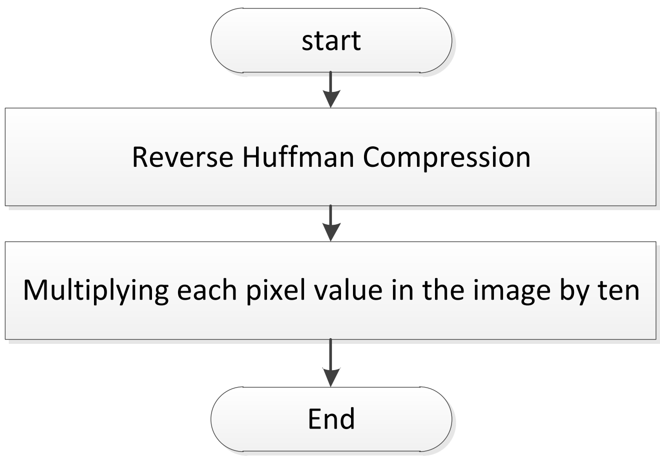


Figure (6): de-compression for RIFD technique

RIFD works on image pixels by rounding first and later dividing the intensity. This sequence aims to reduce the range of the intensities, as well as increasing intensities redundancy, which achieves better compression performance. A significant performance of RIFD technique remarked while followed by Huffman algorithm.

The results are satisfying particularly for all colored images and natural images with high bit depths. Human perception cannot observe the losses that took place after decompressing an image been compressed using RIFD.

Using a good variety of image test set, RIFD showed ten time better performance than Huffman when it is applied on colored images, also higher compression ratio, where the average of compression ratio on these set of images for Huffman was (1.11), where the RIFD showed (6.42) compression ratios (Otair and Shehade, 2016).

Related Work

The researcher reviewed many past related studies to have clear vision about the research area and to make some type of analysis for the previous studies results with the results that will be obtained from this proposed technique.

(Chawla, et al., 2014) explain lossy Compression Technique that minimizes a file by constantly terminate conformed information, specially excessive information, if the file is uncompressed, only a part of the main information is remains there, lossy compression is mostly used for video and sound, where a particular amount of data loss will not be detected by several users, the JPEG image file, generally used for photographs and other complicated yet images on the Web, it is an image which has lossy compression by utilize JPEG compression, the innovator can be determined how much loss to introduce and activated a trade-off between the size of file and the image type.

They indicate that the most lossy compressors are three-step algorithms, every one of this is in accordance with three type of redundancy , the first phase is a convert to eliminate the inter pixel redundancy to bundle information efficiently, then a quantize is applied to tear out psycho-visual redundancy to appear the packed information with as few bits as potential , the quantized bits are then effectively encoded to obtain more compression from the coding redundancy, also they show in this study to the most lossy image compression coding that have been used which are : Croma Coding, Fractal Coding, Transform Coding and Vector quantization.

(Vijayvargiya, et al., 2013) Explain the main goal of image compression is to exemplify an image in the smallest number of bits without losing the major information content within an original image Compression methods are being speedily improved for compress large data files like images, , there are several algorithms which perform this compression in various ways; Some of these compression methods are designed for the specific type of images, thus they will not be perfect for other kinds of images, this study addresses about different image compression method.

In this study they look over various kinds of current procedure of image compression such as Inter Pixel Redundancy where about in image adjacent pixels are not statistically separate, it is according to the connection between the neighboring pixels of an image, this kind of redundancy is known as Inter-pixel redundancy, this kind of redundancy is may also known as spatial redundancy, this redundancy may be examine in many ways, one of which is through expecting a pixel value depend on the values of its adjacent pixels. In order to do so, the original 2-D array of pixels is generally mapped into a various shape.

They also talked about Coding Redundancy depending on in using variable length code words chosen as to identify the statistics of the main source, in this situation , the image itself or a handle fiction of its pixel values, this kind of coding is constantly reversible and generally performed using lookup tables (LUTs), in Psycho Visual Redundancy much experience on the psycho physical side of human sight have confirmed that the human eye does not react with equivalent sensibility to all coming visible information; some part of information are more significant than others. Most of the image coding mechanism in use today employ this type of redundancy, like the Discrete Cosine Transform (DCT) depend algorithm at the core of the JPEG encoding standard.

(Jassim and Qassim, 2012) indicate that Data can be compressed by decreasing the redundancy in the main data, but this makes the data have further errors. In this study a novel method of an image compression depend on a different technique which has been formed for image compression which has been called Five Modulus Method (FMM). The new procedure consists of converting every pixel value in an 8×8 block into a several of 5 for every of the R, G and B arrays. Then the new values can be divided by 5 to have new 6-bit bit length pixel values, and it has lower storage area than the original image. This study offered a new compression order of the new values as a stream of bits, which improved the chance for storing the new compressed image easily.

This study explains the potential of the FMM depend on image compression technique, the priority of this method is the high PSNR even though it has low compression ratio.

This technique is appropriate for bi-level images, where the pixel is symbolize by one byte (8-bit). Because of the low compression ratio, this method cannot be used as a standalone technique, but it could add as scheme within other compression techniques.

(Venkataramanan, et al., 2014), this study presents computationally adequate encoders and decoders approach for loss compression by using a Sparse Regression Code. The codebook is defined by a designed two dimensional array and also a code words that structured linear integration of columns of this matrix. The suggest encoding algorithm constantly select columns of the design matrix to respectively approximate the origin sequence.

This research paper, investigates a category of codes called Sparse Superposition (SPAhRC) for lossy compression with squared-errors distortion criterion, the size of the design matrix is a low-order polynomial in the block length, as effect of which the storage complication is much low than that of the random codebook, it suggest a sequential approximation encoder with computational complexity raising polynomials in the block-length. .

The encoding algorithm could be interpreted as successively purification the source through an asymptotically large number of phases with asymptotically small rate in every stage, they confirmed that the successive refinement interpretation is unique to this specific algorithm, and is not an inherent ownership of the sparse regression codebook the section coefficients were pick to optimize the encoding algorithm, the coefficients allocate ‘power’ through sections of the design matrix and they are chosen based on the encoder.

(Gupta and Garg, 2012) This study points out that Image compression is the implementation of Data compression on digital images; they show that discrete cosine transform (DCT) is a method for transforming a signal into primary frequency components, it is successfully used in image compression, this study develop some easy functions to calculate the DCT and to compress images.

IMAQ block IMAP block of MATLAB was used to resolve and study the effect of Image Compression using DCT and modifying co-efficient for compression were improved to show the resulting image and error image from the Maine images image compression algorithm was understood using Mat lab code, and altered to perform best when performed in hardware description language.

Image Compression was calculated by using 2-D discrete Cosine convert where the main image is converted in 8-by-8 blocks and after that transformed 8-by-8 blocks to produce the reconstructed image, the inverse DCT should be performed by using subset of DCT coefficients, the fault image would be presented , Error value for each image would be calculated through different values of DCT co-efficient as chosen by the user and should be displayed at the end to discover the precision and compression in the final image and out coming execution parameter would be specified in phrase of MSE (Mean Square Errorki).

(Ha, et al., 2016) describes a preprocessing technique for lossless compression of 3D geometry data, this type of data contained of vertices that are the three-dimensional location data in 3D space. usually, a 3D scanner create a enormous data contained 3D geometry data from an object, therefore , a compression technique is required to convey 3D data through networks and state-of-the-art techniques have been studied.

In this study they propose a method to develop the existing methods, to do so, a simple and functional classification method is insert, classification in the suggest method can minimize the number of vertices, the overhead data from the classification are compressed by a lossless compression like an arithmetic coder. experiential results showed an excellent improvement to the existing methods, also this method can be joint with any existing method, a lossless compression technique for 3D geometry data has been proposed depend on preprocessing, their method can be easily joint with the existing methods and therefore the performance of compression develop in terms of compression ratio.

(Teke, 2012) shows that the research stratifies diverse compression schemes depending on the Haar Wavelet methods and examines the impact of them on the path length the required destination.

The amount of data treated by computers are continues growing, while data size grows, processing time raises, as well as storage devices capacity will be increased to deal with the usage demand. Many applications exist where the storage devices have very fixed capacity as they were designed for applications that do not require large storage devices. This study review the use of different forms of the well-known Haar wavelet, and use an experimental approach to evaluate the efficiency of the storage schemes in facilitate navigation in the network.

(Kipli, et al., 2012) Proposes new method for constructing image quality metrics, this paper is an attempt to examine the potential of combining existing metrics with ANN to predict the quality of images. It was confirmed that Levenberg-Marquardt back propagation algorithm has good ability to predict the MOS with high correlation for training and testing with training error and testing error of 0.244 and 0.399 respectively. The regression, R showed that it is highly correlated with mean opinion score (MOS) compared to individual metrics (PSNR, MSE or SSIM).

They explain that Image quality analysis is to investigate the quality of images and improve the methods to efficiently and swiftly prepare the quality of images, compression and conversion are compulsory. In this case, this paper introduced a hybrid method to determine the image quality by utilizing Levenberg-Marquardt Back-Propagation Neural Network (LMBNN). Three known quality metrics were combined as the input element to the network.

(Yeganeh and Wang, 2013) This study explains Tone-mapping operators (TMOs) that convert high dynamic range (HDR) to lower dynamic range (LDR) images provide practically useful tools for the visualization of HDR images on standard LDR displays. Different TMOs create different tone mapped images, and a natural question is which one has the best quality. Without an appropriate quality measure, different TMOs cannot be compared, and further improvement is directionless. Subjective rating may be a reliable evaluation method, but it is expensive and time-consuming, and more importantly, is difficult to be embedded into optimization frameworks.

This paper develops an objective model to assess the quality of tone mapped images by combining a multi-scale structural fidelity measure and a statistical naturalness measure. The proposed measure not only provides an overall quality score of an image, but also creates multi-scale quality maps that reflect the structural fidelity variations across scale and space. Their experiments show that TMQI is reasonably correlated with subjective evaluations of image quality.

(Zhou, et al., 2014) proved a new image compression–encryption hybrid algorithm based on compressive sensing and random pixel exchanging, where the compression and the encryption are completed simultaneously, where the key is easily distributed, stored or memorized. The image is divided into 4 blocks to compress and encrypt. Then random pixel exchanging is introduced to scramble the compressed and encrypted blocks.

Compared with the methods adopting the whole measurement matrix as key, the proposed algorithm shortens the key greatly, which is of importance for a practical encryption algorithm. By utilizing the circulate matrix to construct the measurement matrix, in CS and controlling the original row vector of circulate matrix with chaos system, the proposed algorithm is secure. By introducing the random pixel exchanging and binding the random matrices with the measurement matrices, the security is enhanced further. The simulation results show that this image compression– encryption hybrid algorithm can provide good security and nice compression performance.

(Kekre, et al., 2013) explain that performance of column transform, row transform, and full transform is compared using Root Mean Square Error (RMSE) as a performance measure on compression ratio. RMSE values are calculated for compression ratio 1 to 5. Experimental results prove that RMSE values obtained for various compression ratios in column transform are closer to those obtained in full Transform of an image. Hence instead of the full transform of an image, column transform can be used for image compression, saving half number of computations. RMSE obtained in row transform is quite higher than the column and full transform at higher values of compression ratio. Hence it is not recommended. Good PSNR is obtained using column transform. Among all the seven transforms used, DFT, DCT and DST gave better results regarding RMSE and reconstructed image quality than other transforms. Walsh and Haar transform also give acceptable results with an advantage of less computation whereas Slant and Kekre transform do not give good results. Hence they are not recommended.

This research intended to improve the affection of compression in lossy techniques when applying them on images but with preserving the compression ratio levels as much as possible, proposed lossless steps were produced to be applied before various lossy techniques to reduce the pixels alteration with minimizing the intensity in the compressed image.

CHAPTER THREE

THEORETICAL DESIGN

Introduction

The compression techniques can be mainly divided into two types: Lossy and Lossless, each one of them has its own positive characteristics. The priority of the lossy technique is the high compression ratio but with a percentage of distortion, while the lossless techniques compression produce a low rate of compression ratio without any distortion.

This thesis has been implemented by adding a proposed lossless pre-processing technique in order to enhance the lossy techniques by producing a minimized distortion rates in the compressed images using these lossy techniques. For this purpose two lossy techniques were selected the classic JPEG compression technique (Strang, 1999) and new novel lossy technique called RIFD (Otair, and Shehadeh, 2016) to observe the efficiency of the proposed pre-processing technique to achieve the desired goals.

Set of images with various kind and bit-depth were applied by the proposed pre-process steps which is called MDT (Minimized Decreasing Technique) and then separately on the chosen lossy technique, then compare the compression ratios and quality results.

The RIFD is simple and depends on rounding numbers, in other hand the JPEG perform complicated mathematical operation to compress the images. These two techniques could be good samples to experiment the efficiency of implementing the MDT before them.

The proposed model has been implemented in two parts (compression and de-compression) and each part includes two procedures (lossy and lossless) as shown in the Figure (7).

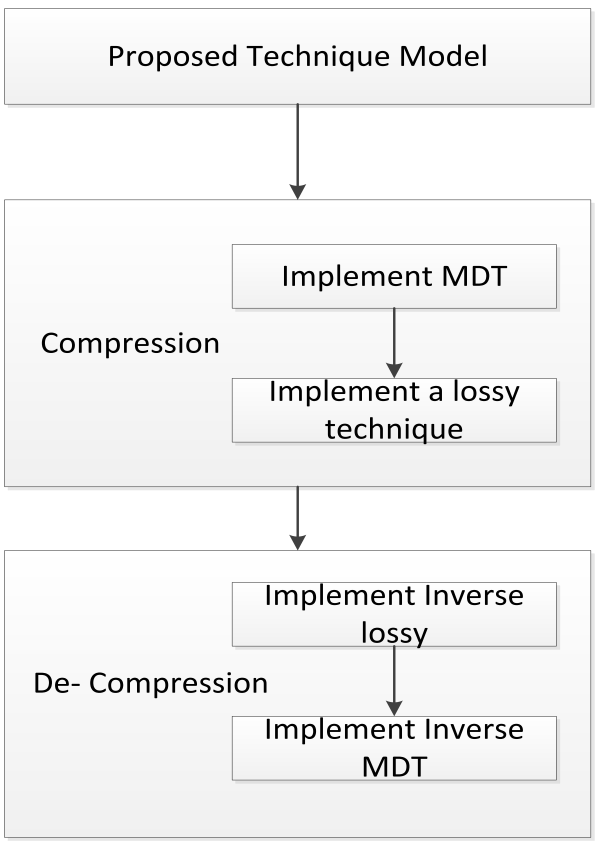


Figure (7): MDT proposed compression technique

Tools

To verify the validity of the proposed technique, a suitable programming application and set of images are required to observe the results and compare them:

Matlab 2015a (version):

MATLAB is a fourth-generation programming language and numerical analysis environment. This program includes matrix calculations, developing and running algorithms, creating user interfaces and data visualization. That is usually used with the experiments in image process filed.

There are different releases of MATLAB programming. This thesis using the (MATLAB 8.5) which released on March 5, 2015 and also called (R2015a); this version includes new releases of MATLAB and Simulink, with new Simulink graphical controls and displays, which are not available in the previous versions.

Set of images:

There is no standard set of test images only few well know images in image process, so the researcher choose multiple images of a different type and extensions that represent the most used images, as shown in appendix (A), the set of images contains the following file extensions (.tif, .jpg, .gif and .jpg), with different dimensions that varied between (120 X 120) till (1200 x 1200).

There are different types of digital image which is a numeric representation of a two-dimensional image. The researcher chooses two types of digital image:

The gray scale image: an image which the values of each pixel are between the maximum white color value and the black zero value, so it is composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest and we can find many classes for those images depending on its bit-depth values, the researcher chose the 8 and 16 bit-depth values to experiment them

The color image: an image that includes color information for each pixel usually consists of three layers of colors by combining those layers the required color is obtained, also known as RGB (Red, Green and Blue) color space which is commonly used in computer displays, other spaces for color images such as YCbCr, HSV are often used in other contexts.

Algorithms used

The proposed technique has been achieved through implementing three procedures: lossless, lossy and the MDT technique, this model was utilized to improve the effect of the MDT when processed before the chosen lossy techniques. For this purpose, the results of the quality metrics obtained from applying the lossy techniques are compared with the ones that appeared after employing MDT technique before each lossy technique

Lossy part

Two different lossy compression techniques were chosen to improve the images quality resulting after applying these techniques on a set of images with various types, seeking to preserve the high compression ratio, and to prove that the proposed pre-processing is suitable for various types of Lossy techniques since they are different in techniques and complicity.

(A) Rounding the intensity followed by dividing (RIFD):

A simple lossy technique that depends on plain mathematical process was considered to be improved and apply the experiments; it was rounding the intensity followed by dividing (RIFD).

The images that were used had either eight or sixteen bit-depth, and this technique is intended to reduce the intensity of the image to obtain high compression ratio by rounding the numbers to the nearest tenth in case of eight bit images and nearest thousandth in case of sixteenth bit-depth images, the researcher of this thesis found that if it was rounded to the nearest eighth it will sustain the compression ratio and the intensity to five digits bit-depth with reducing the error after compression, since rounding to nearest tenth produce maximum pixel value of 25 or less, but when rounding to the nearest eighth maximum pixel value can reach 31 also using five digits intensity but with less error.

The RIFD technique suggested rounding the sixteenth bit-depth images to the nearest thousand. After many experiments the researcher found that rounding the pixel value to the number 999 will maintain the compression ratio since both rounding numbers results in seven digit compressed image but new rounding value will reduce the error.

The RIFD technique is implemented according to the flow chart shown in the Figure (8).

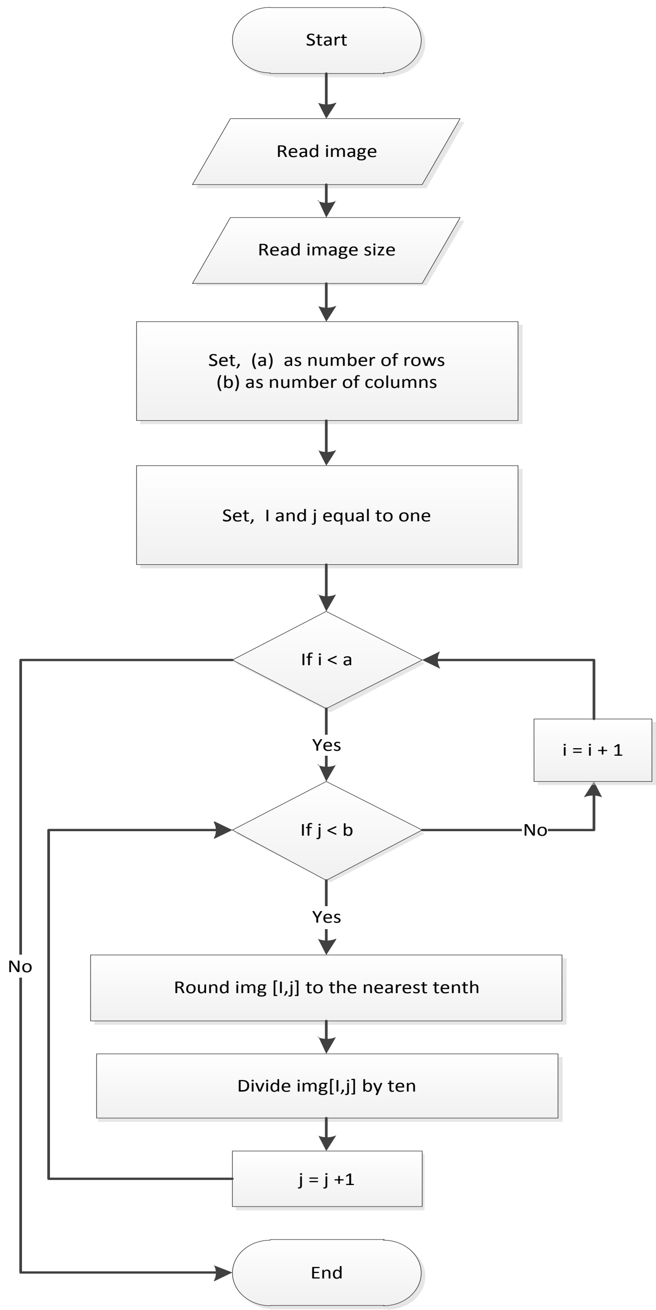


Figure (8): RIFD technique

The procedure has been implemented as the following algorithm, as shown in appendix (C):

Algorithm: RIFD algorithm (M, N)

Input: image

Output: image and table

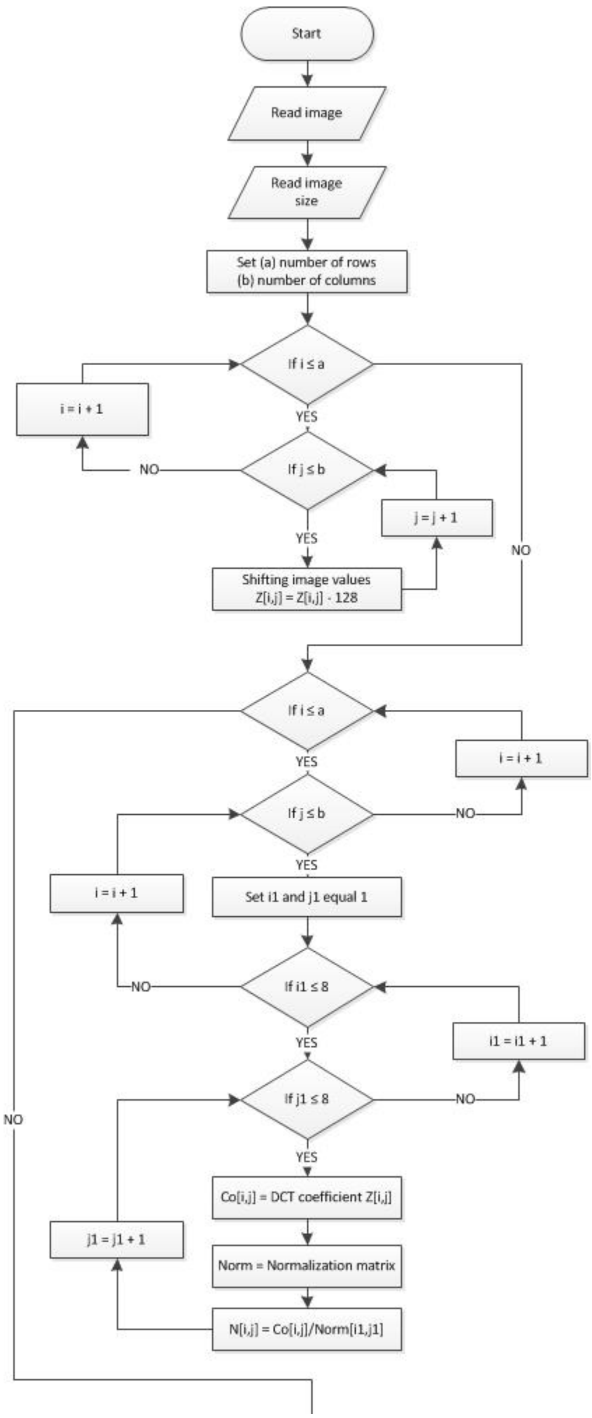
Rounding the numbers according to its bit-depth value

Dividing the pixels by the number they were rounded by.

Save the results to apply de-compression

(B) Joint Photographic Experts Group (JPEG)

The well known compression technique that depends on dividing the image into blocks then calculate the coefficient for each pixel using discrete cosine transformation or DCT, after dividing the block by a standard quantization matrix a set of vectors are produced that will represent each block in the image. Then to reduce the amount of data that represent the image an encoding technique will be applied on each vector, this will be either Huffman or Run length encoding RLE. In this research we will use the RLE, using this lossless technique will reduce the redundancy of the bits representing the image without any further data loss that DCT transformation results. As shown in Figure (9).


Start
Read image
Read image size
Set (a) number of rows (b) number of columns
If i ≤ a
YES
i = i + 1
If j ≤ b
NO
YES
j = j + 1
Shifting image values Z[i,j] = Z[i,j] - 128
NO
If i ≤ a
YES
i = i + 1
If j ≤ b
NO
YES
i = i + 1
Set i1 and j1 equal 1
If i1 ≤ 8
NO
YES
i1 = i1 + 1
If j1 ≤ 8
NO
YES
NO
Co[i,j] = DCT coefficient Z[i,j]
j1 = j1 + 1
Norm = Normalization matrix
N[i,j] = Co[i,j]/Norm[i1,j1]

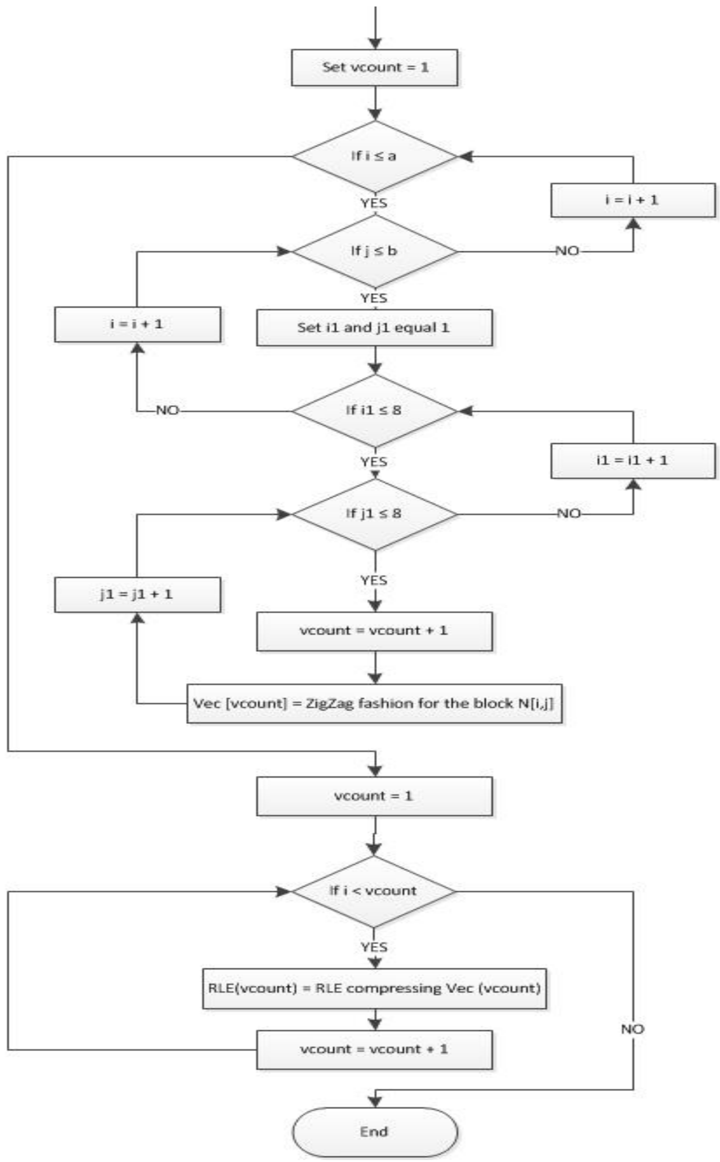


Figure (9): JPEG technique

The following algorithm describe the JPEG technique, as shown in appendix (D)

Algorithm: JPEG algorithm (M, N)

Input: image

Output: image and table

Divide the image into 8x8 blocks

Shifting all pixels by subtracting (2^bitdepth/2)

Apply DCT and finding each pixel coefficient

Normalize by dividing each block by Normalization matrix

Apply ZigZag fashion to change each block into vectors

Compressing using RLE

MDT procedure

The main idea of this thesis is to implement a MDT steps before implementing a lossy technique to reduce the distortion and attempting to preserve the high compression ratio, the MDT can be implemented as the following flowcharts as shown in the following Figure (10):

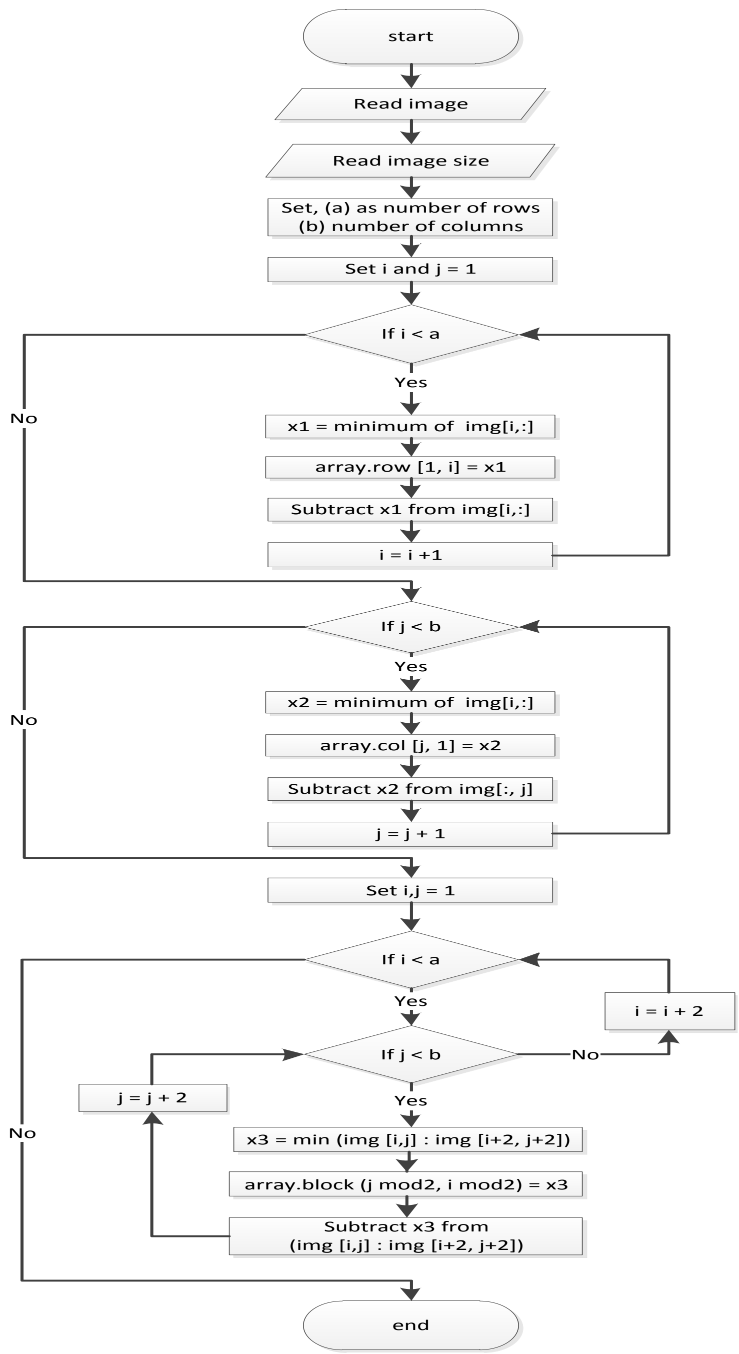


Figure (10): MDT Model

Below mentioned algorithm is for the pre-processing lossless MDT steps, as shown in appendix (B):

Algorithm: MDT Steps (M, N)

Input: image

Output: image and table

Read the image file

Find the dimensions of each image

Calculate each row minimum value and save to one dimension array

Reduce each row by its minimum value

Save the result into a new array

Calculate each column minimum value and save to one dimension array

Reduce each column by its minimum value

Save the result into a new array

Divide the array into 2 by 2 block

Find the minimum value of each block and save to two dimension array

Reduce the entire block by its minimum value

Save the result into a new array

The compressed image will consist of four arrays; two of them are one domination array that contains the minimum values of original image rows and columns. In addition, two dimension array consist of the minimum values for each 2 by 2 block, the fourth array contain the remaining pixel after subtracting all minimum values. As shown in the following example in figures (11) to figure (14):

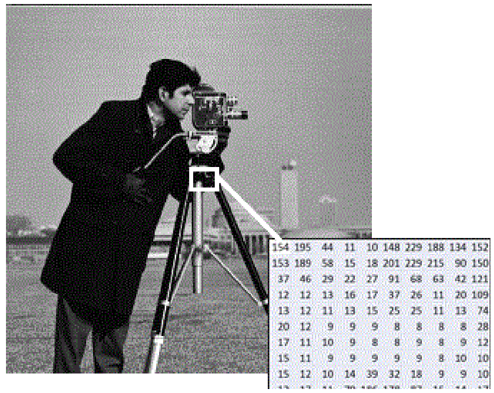


Figure (11): Taking a pixel portion from Cameraman image

A portion of pixel was taken from Cameraman image as shown in Figure (11), which has good variety of pixels to experiment the MDT steps.

154	195	44	11	10	148	229	188	134	152		10
153	189	58	15	18	201	229	215	90	150		15
37	46	29	22	27	91	68	63	42	121		22
12	12	13	16	17	37	26	11	20	109		11
13	12	11	13	15	25	25	11	13	74		11
20	12	9	9	9	8	8	8	8	28		8
17	11	10	9	8	8	9	8	9	12		8
15	11	9	9	9	9	9	8	10	10		8
15	12	10	14	39	32	18	9	9	10		9

Figure (12): finding the minimum values for each row

From each row finding, the minimum value and save it in additional one dimensional array to be used later in the de-compression phase.

144	185	34	1	0	138	219	178	124	142
138	174	43	0	3	186	214	200	75	135
15	24	7	0	5	69	46	41	20	99
1	1	2	5	6	26	15	0	9	98
2	1	0	2	4	14	14	0	2	63
12	4	1	1	1	0	0	0	0	20
9	3	2	1	0	0	1	0	1	4
7	3	1	1	1	1	1	0	2	2
6	3	1	5	30	23	9	0	0	1
1	6	0	68	175	167	76	4	3	6

1	1	0	0	0	0	0	0	0	1

Figure (13): finding the minimum values for each column

After subtracting the minimum value of each row from the entire row, Figure (13) shows finding the minimum value of each column and save it in another additional.

143	184	34	1	0	138	219	178	124	141
137	173	43	0	3	186	214	200	75	134
14	23	7	0	5	69	46	41	20	98
0	0	2	5	6	26	15	0	9	97
1	0	0	2	4	14	14	0	2	62
11	3	1	1	1	0	0	0	0	19
8	2	2	1	0	0	1	0	1	3
6	2	1	1	1	1	1	0	2	1
5	2	1	5	30	23	9	0	0	0
0	5	0	68	175	167	76	4	3	5

137	0	0	178	75
0	0	5	0	9
0	0	0	0	0
2	1	0	0	1
0	0	23	0	0

6	47	34	1	0	138	41	0	49	66
0	36	43	0	3	186	36	22	0	59
14	23	7	0	0	64	46	41	11	89
0	0	2	5	1	21	15	0	0	88
1	0	0	2	4	14	14	0	2	62
11	3	1	1	1	0	0	0	0	19
6	0	1	0	0	0	1	0	0	2
4	0	0	0	1	1	1	0	1	0
5	2	1	5	7	0	9	0	0	0
0	5	0	68	152	144	76	4	3	5

Figure (14): finding the minimum values for each 2×2 block

The next step is to divide the matrix into 2x2 blocks, finding the minimum value of each block and reduce the entire block value by its founded minimum value to result the final matrix before using the lossy technique , as shown in Figure (14).

MDT-Lossy Techniques:

This technique has been implemented by merging MDT and lossy. As shown in Figure (15).

Algorithm: MDT-loss algorithm (M, N)

Input: image

Output: image and table

Read the image file

Applying the pre-processing steps (MDT)

Applying the lossy technique

Save the results

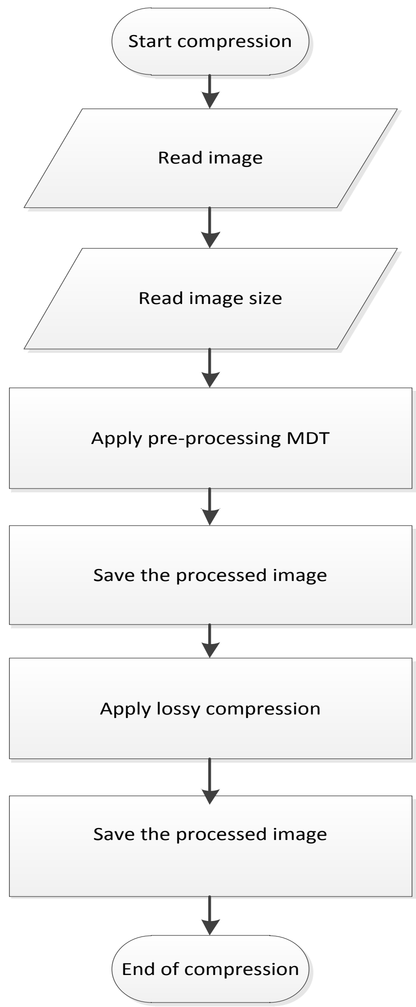


Figure (15): MDT-Lossy Model

CHAPTER FOUR

THE EXPERIMENTAL WORK

4.1. Introduction

This chapter will present the results of applying the different techniques used in this thesis on each image in the test images set, with calculating the quality metrics and the compression ratio for each image compression.

Each lossy technique was implemented and the results calculate then used again after adding the proposed technique to compare the results and show the enhancement in the compression quality.

Various charts and tables show the improvement on the used lossy compression technique after applying the MDT technique before each lossy technique, with presenting the images before and after each compression process.

4.2. Implementation

All experiments were accomplished using MATLAB application with the necessary programming codes following the flowcharts steps mentioned in the previous chapter, the images were divided into three main groups:

Gray scale 8 bit-depth images

Gray scale 16 bit-depth images

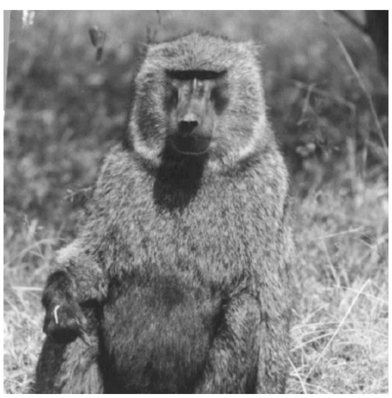
Color images

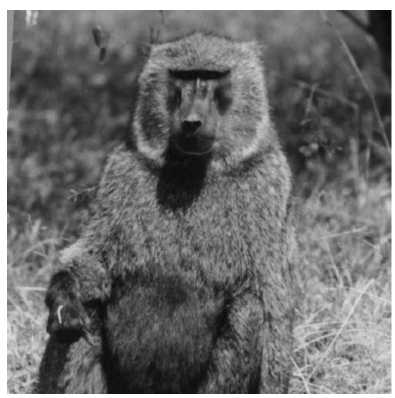
Each group has different types of image formatting or extensions which are (gif, tif, png, jpg) the images also had diverse dimensions that varied between (120 X 120) up to (1200 X 1200).

Each image in this thesis test set has been compressed using the chosen lossy techniques, after calculating the MSE, MAE and PSNR same image will be compressed by the lossy techniques but after adding the proposed technique also with calculating the same quality metrics, then compare both results.

4.2.1. RIFD

The proposed model utilized RIFD technique on a set of Gray scale 8-bit images, to verify the difference between the original and de-compressed images, as shown in the figure (16) as an example:





(b) de-compressed image (a) Original image

Figure (16): RIFD compression on Gray 8-bit

When implementing the compression process on set of Gray scale 8-bit images using only RIFD technique, several quality metrics had been computed (MSE, MAE, PSNR) with the compression ratio (Cr), to be compared later with the proposed model.

Table (1): The results of RIFD compression on Gray 8-bit

Gray 8 bit	MSE	MAE	PSNR	CR1
g1.gif	9.93	3.10	38.19	0.48
g2.gif	8.53	2.63	38.86	0.54
g3.png	8.34	2.60	38.95	0.49
g4.GIF	8.53	2.62	38.86	0.48
g5.png	11.87	3.21	37.42	0.49
g6.jpg	8.77	3.08	38.74	0.44
g7.tif	8.49	2.67	38.88	0.52
g8.png	7.76	2.63	39.27	0.43
g9.png	8.57	2.69	38.84	0.55
g10.tif	8.71	2.28	38.76	0.51
g11.jpg	8.45	3.24	38.89	0.55
g12.gif	8.43	3.24	38.90	0.53
g13.png	8.32	2.51	38.96	0.55
g14.jpg	8.41	2.53	38.92	0.58

To prove the validity and efficiency of the proposed technique, the same sets of Gray scale 8-bit images were compressed using the MDT_RIFD technique, as shown in the following figure (17) as example:


(b) de-compressed image (a) Original image

Figure (17): MDT_RIFD compression on Gray 8-bit

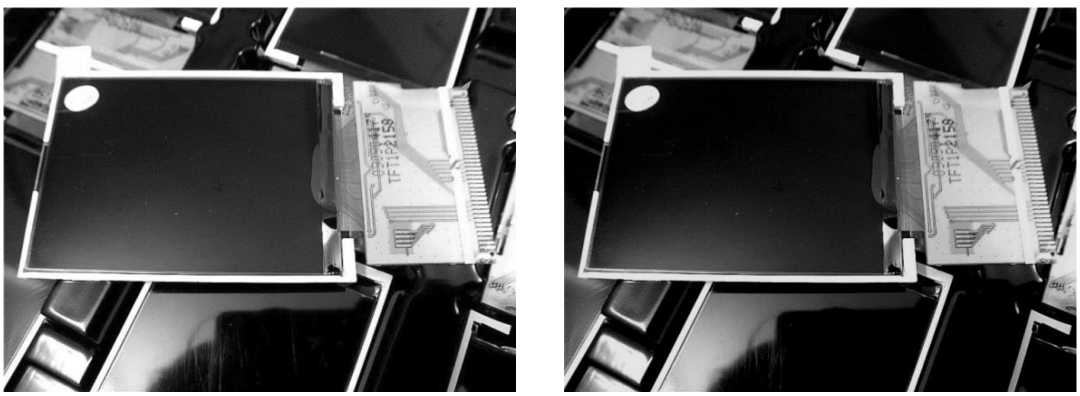
When implementing the compression process on set of Gray scale 8-bit images using only MDT_RIFD technique, several quality metrics had been computed (MSE, MAE, and PSNR) with the compression ratio (Cr).

Table (2): Gray 8-bit MDT_RIFD

Gray 8 bit	MSE	MAE	PSNR	CR1
g1.gif	4.35	1.33	41.78	0.46
g2.gif	6.52	1.90	40.02	0.51
g3.png	6.39	1.88	40.11	0.49
g4.GIF	5.36	1.66	40.88	0.47
g5.png	3.61	1.18	42.59	0.41
g6.jpg	5.71	1.74	40.60	0.45
g7.tif	6.04	1.81	40.35	0.44
g8.png	5.23	1.57	40.98	0.46
g9.png	4.60	1.44	41.54	0.49
g10.tif	5.96	1.79	40.41	0.44
g11.jpg	6.30	1.86	40.17	0.53
g12.gif	6.02	1.81	40.37	0.51
g13.png	5.87	1.76	40.48	0.54
g14.jpg	6.50	1.90	40.04	0.53

The image (g5) showed the highest PSNR value (42.5859636) and the lowest MSE and MAE values (3.613129861, 1.181732639)

The proposed model utilized RIFD technique on a set of Gray scale 16-bit images, to verify the difference between the original and de-compressed images, as shown in the following figure (18) as example:



(b) de-compressed image (a) Original image

Figure (18): RIFD compression on Gray 16-bit

When implementing the compression process on set of Gray scale 16-bit images using only RIFD technique, several quality metrics had been computed (MSE, MAE, PSNR) with the compression ratio (Cr), to be compared later with the proposed model.

Table (3): Gray 16-bit RIFD

Gray 16 bit	MSE	MAE	PSNR	CR1
1.gif	21,868.98	125.96	47.64	0.35
2.gif	21,671.42	125.80	47.67	0.37
3.png	21,602.72	125.18	45.04	0.35
4.png	21,486.74	124.53	47.71	0.24
5.png	21,470.23	124.44	46.65	0.33
6.png	21,277.04	123.26	48.81	0.35
7.gif	21,052.71	122.15	45.13	0.35
8.png	21,858.97	126.43	37.05	0.31
9.png	21,489.96	124.85	45.06	0.31
10.gif	15,509.64	90.12	46.26	0.31
11.png	21,536.45	124.81	45.05	0.35
12.png	18,150.00	110.19	46.75	0.17
13.tif	20,677.91	117.69	45.20	0.35
14.tif	21,041.67	119.22	48.32	0.31
15.png	21,388.40	123.90	45.07	0.36
16.png	21,616.52	125.27	47.15	0.28

To prove the validity and efficiency of the proposed technique, the same sets of Gray scale 8-bit images were compressed using the MDT_RIFD technique, as shown in the following figure (19) as an example:

(b) de-compressed image (a) Original image

Figure (19): MDT_RIFD compression on Gray 16-bit

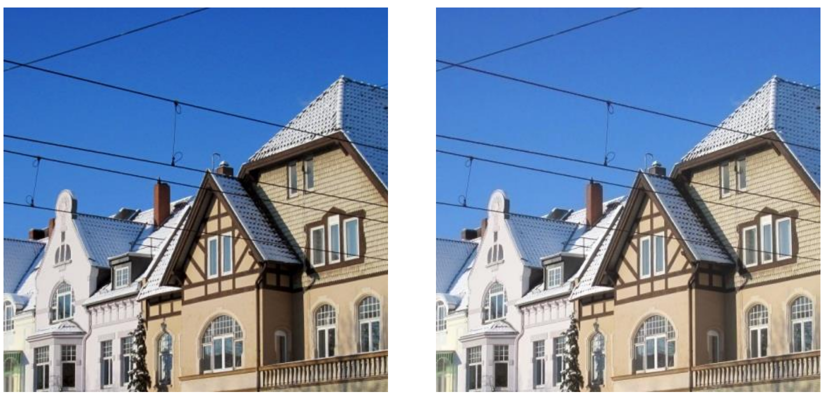
When implementing the compression process on set of Gray scale 16-bit images using only MDT_RIFD technique, several quality metrics had been computed (MSE, MAE, and PSNR) with the compression ratio (Cr).

Table (4): Gray 16-bit MDT_RIFD

Gray 16 bit	MSE	MAE	PSNR	CR1
1.gif	18,065.07	103.17	53.76	0.41
2.gif	17,234.09	98.84	53.97	0.45
3.png	18,154.40	104.81	53.74	0.41
4.png	18,668.90	107.55	53.62	0.40
5.png	17,192.11	99.37	53.98	0.46
6.png	16,995.62	98.40	54.03	0.44
7.gif	16,457.22	95.29	54.17	0.42
8.png	17,876.02	103.90	53.81	0.39
9.png	18,996.70	109.35	53.54	0.38
10.gif	10,839.70	65.04	55.98	0.40
11.png	15,109.09	80.41	54.54	0.47
12.png	15,089.26	74.22	54.54	0.43
13.tif	14,447.44	72.78	54.73	0.44
14.tif	14,732.38	70.62	54.65	0.41

15.png	19,372.39	111.45	53.46	0.37
16.png	17,613.40	101.90	53.87	0.40

The proposed model utilized RIFD technique on a set of Color 8-bit images, to verify the difference between the original and de-compressed images, as shown in the following figure (20) as example:



(b) de-compressed image (a) Original image

Figure (20): RIFD compression on Color images

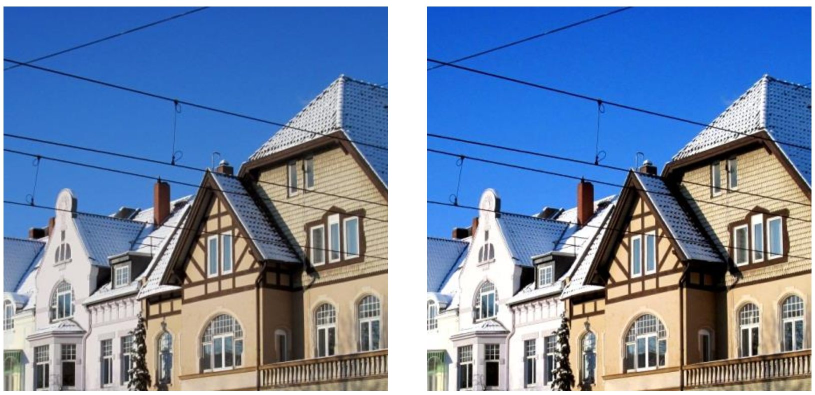
When implementing the compression process on set of Color 8-bit images using only RIFD technique, several quality metrics had been computed (MSE, MAE, PSNR) with the compression ratio (Cr), to be compared later with the proposed model.

Table (5): Color 8-bit RIFD

Color	MSE	MAE	PSNR	CR1
c1.jpg	8.54	2.50	38.85	0.56
c2.png	8.95	2.62	38.65	0.45
c3.png	8.56	2.51	38.84	0.55
c4.jpg	8.32	2.46	38.96	0.43
c5.png	8.40	2.48	38.92	0.53
Color	MSE	MAE	PSNR	CR1
c6.jpg	8.56	2.51	38.84	0.53
c7.png	8.55	2.50	38.85	0.56
c8.jpg	8.39	2.48	38.93	0.52
c9.jpg	8.48	2.50	38.88	0.44
c10.tif	8.81	2.47	38.72	0.52
c11.png	8.46	2.48	38.89	0.58
c12.jpg	8.59	2.52	38.83	0.54
c13.tif	8.41	2.47	38.92	0.55
c14.png	8.40	2.45	38.92	0.38

c15.tif	9.14	2.59	38.56	0.48
c16.jpg	8.71	2.53	38.76	0.55

To prove the validity and efficiency of the proposed technique, the same sets of Color 8-bit images were compressed using the MDT_RIFD technique, as shown in the following figure (21):



(b) de-compressed image (a) Original image

Figure (21): MDT_RIFD compression on Color images

When implementing the compression process on set of Color 8-bit images using only MDT_RIFD technique, several quality metrics had been computed (MSE, MAE, and PSNR) with the compression ratio (Cr).

Table (6): Color 8-bit MDT_RIFD

Color	MSE	MAE	PSNR	CR1
c1.jpg	6.31	1.86	40.16	0.35
c2.png	3.95	1.33	42.20	0.26
c3.png	5.92	1.78	40.44	0.36
c4.jpg	2.79	1.10	43.70	0.23
c5.png	5.30	1.64	40.93	0.34
c6.jpg	5.51	1.68	40.75	0.33
c7.png	6.21	1.84	40.24	0.33
c8.jpg	5.65	1.73	40.65	0.29
c9.jpg	4.42	1.36	41.71	0.27
c10.tif	4.92	1.52	41.25	0.28
c11.png	6.34	1.86	40.15	0.44
c12.jpg	5.96	1.79	40.41	0.32
c13.tif	5.96	1.79	40.41	0.33
c14.png	6.22	1.85	40.23	0.25
c15.tif	3.73	1.31	42.45	0.24
c16.jpg	6.42	1.89	40.09	0.28

All experiments show significant improvement when using MDT steps before RIFD lossy technique, this obvious in the results of the three quality metrics MSE, MAE and PSNR. There is also an improvement in the compression ratio.

4.2.2. JPEG

The proposed model utilized JPEG technique on a set of Gray scale 8-bit images, to verify the difference between the original and de-compressed images, as shown in the following figure (22):

(b) de-compressed image (a) Original image

Figure (22): JPEG compression on Gray 8-bit

When implementing the compression process on set of Gray scale 8-bit images using only JPEG technique, several quality metrics had been computed (MSE, MAE, PSNR) with the compression ratio (Cr), to be compared later with the proposed model.

Table (7): The results of JPEG compression on Gray 8-bit

Gray 8 bit	MSE	MAE	PSNR	CR1
g1.gif	82.24	12.12	29.01	0.33
g2.gif	255.00	143.73	24.10	0.36
g3.png	68.87	6.98	29.78	0.31
g4.GIF	150.13	64.75	26.40	0.33
g5.png	186.57	33.11	25.46	0.25
g6.jpg	119.79	33.55	27.38	0.30
g7.tif	254.98	136.00	24.10	0.31
g8.png	254.43	164.69	24.11	0.37
g9.png	147.75	58.97	26.47	0.33
g10.tif	145.99	181.47	26.52	0.39
g11.jpg	103.53	29.24	28.01	0.39
g12.gif	117.24	13.14	27.47	0.40
g13.png	144.33	59.09	26.57	0.38
g14.jpg	116.31	37.28	27.51	0.34

To prove the validity and efficiency of the proposed technique, the same sets of Gray scale 8-bit images were compressed using the MDT_JPEG technique, as shown in the following figure (23) as example:

(b) de-compressed image (a) Original image

Figure (23): MDT_JPEG compression on Gray 8-bit

When implementing the compression process on set of Gray scale 8-bit images using only MDT_JPEG technique, several quality metrics had been computed (MSE, MAE, and PSNR) with the compression ratio (Cr).

Table (8): Gray 8-bit MDT_JPEG

Gray 8 bit	MSE	MAE	PSNR	CR1
g1.gif	60.48	9.21	30.35	0.38
g2.gif	86.06	6.04	28.82	0.42
g3.png	51.32	5.19	31.06	0.43
g4.GIF	60.96	4.44	30.31	0.40
g5.png	30.76	2.89	33.28	0.29
g6.jpg	112.04	6.45	27.67	0.37
g7.tif	81.22	5.45	29.07	0.33
g8.png	114.38	6.92	27.58	0.39
g9.png	95.37	5.54	28.37	0.45
g10.tif	58.77	4.54	30.47	0.37
g11.jpg	97.19	6.68	28.29	0.43
g12.gif	93.46	6.32	28.46	0.48
g13.png	129.08	7.62	27.06	0.54
g14.jpg	101.33	8.88	28.11	0.51

The proposed model utilized JPEG technique on a set of Gray scale 16-bit images, to verify the difference between the original and de-compressed images, as shown in the following figure (24) as example:

(b) de-compressed image (a) Original image

Figure (24): JPEG compression on Gray 16-bit

When implementing the compression process on set of Gray scale 16-bit images using only JPEG technique, several quality metrics had been computed (MSE, MAE, PSNR) with the compression ratio (Cr), to be compared later with the proposed model.

Table (9): Gray 16-bit JPEG

Gray 16 bit	MSE	MAE	PSNR	CR1
1.gif	32,193.34	18,818.80	51.25	0.25
2.gif	39,248.96	23,831.55	50.39	0.32
3.png	24,193.14	11,572.47	52.49	0.25
4.png	23,636.97	1,260.80	52.59	0.29
5.png	32,695.80	3,313.40	51.18	0.35
6.png	28,447.43	6,049.55	51.79	0.32
7.gif	39,180.89	7,058.73	50.40	0.41
8.png	21,480.09	3,070.13	53.01	0.27
9.png	19,476.07	2,875.16	53.43	0.27
10.gif	19,176.61	5,577.25	53.50	0.28
11.png	27,717.29	11,603.64	51.90	0.34
12.png	32,813.70	22,801.79	51.17	0.24
13.tif	30,881.97	6,030.34	51.43	0.32
14.tif	25,024.44	3,150.38	52.35	0.30
15.png	31,658.85	5,609.98	51.32	0.26
16.png	27,698.48	1,679.78	51.91	0.29

To prove the validity and efficiency of the proposed technique, the same sets of Gray scale 8-bit images were compressed using the MDT_JPEG technique, as shown in the following figure (25) as example:

(b) de-compressed image (a) Original image

Figure (25): MDT_JPEG compression on Gray 16-bit

When implementing the compression process on set of Gray scale 16-bit images using only MDT_JPEG technique, several quality metrics had been computed (MSE, MAE, and PSNR) with the compression ratio (Cr).

Table (10): Gray 16-bit MDT_JPEG

Gray 16 bit	MSE	MAE	PSNR	CR1
1.gif	22,404.35	680.38	52.83	0.40
2.gif	22,803.64	912.39	52.75	0.43
3.png	21,506.24	624.31	53.00	0.40
4.png	20,636.97	592.30	53.18	0.43
5.png	26,695.80	1,391.48	52.07	0.46
6.png	23,447.43	934.15	52.63	0.44
7.gif	32,180.89	2,306.40	51.25	0.34
8.png	19,480.09	447.59	53.43	0.40
9.png	18,476.07	364.23	53.66	0.40
10.gif	15,176.61	716.13	54.52	0.40

Gray 16 bit	MSE	MAE	PSNR	CR1
11.png	23,717.29	1,052.30	52.58	0.45
12.png	23,513.33	859.92	52.62	0.42
13.tif	23,881.97	942.50	52.55	0.44
14.tif	21,024.44	544.91	53.10	0.41
15.png	19,271.49	378.76	53.48	0.40
16.png	21,698.48	582.72	52.97	0.42

The proposed model utilized JPEG technique on a set of Color 8-bit images, to verify the difference between the original and de-compressed images, as shown in the following figure (26) as example:

(b) de-compressed image (a) Original image

Figure (26): JPEG compression on Color images

When implementing the compression process on set of Color 8-bit images using only JPEG technique, several quality metrics had been computed (MSE, MAE, PSNR) with the compression ratio (Cr), to be compared later with the proposed model.

Table (11): Color 8-bit JPEG

Color	MSE	MAE	PSNR	CR1
c1.jpg	119.53	8.64	27.39	0.30
c2.png	185.84	14.12	25.47	0.32
c3.png	120.98	7.21	27.34	0.22
c4.jpg	98.96	8.61	28.21	0.14
c5.png	172.56	14.98	25.80	0.49
c6.jpg	221.79	11.17	24.71	0.46
c7.png	144.24	12.46	26.57	0.57
c8.jpg	160.20	13.73	26.12	0.37
c9.jpg	290.66	15.41	23.53	0.27
c10.tif	91.88	7.29	28.53	0.38
c11.png	120.14	11.39	27.37	0.24
c12.jpg	120.50	8.00	27.35	0.44

c13.tif	95.38	8.99	28.37	0.55
c14.png	173.28	18.23	25.78	0.18
c15.tif	94.56	8.48	28.41	0.36
c16.jpg	274.04	13.55	23.79	0.39

To prove the validity and efficiency of the proposed technique, the same sets of Color 8-bit images were compressed using the MDT_JPEG technique, as shown

in the following figure (27) as an example:

(b) de- (a) Original image

Figure (27): MDT_JPEG compression on Color images

When implementing the compression process on set of Color 8-bit images using only MDT_JPEG technique, several quality metrics had been computed (MSE, MAE, and PSNR) with the compression ratio (Cr).

Table (12): Color 8-bit MDT_JPEG

Color	MSE	MAE	PSNR	CR1
c1.jpg	9.66	1.26	38.32	0.26
c2.png	5.71	0.99	40.60	0.54
c3.png	9.98	1.21	38.17	0.14
c4.jpg	1.96	0.61	45.24	0.46
c5.png	9.56	1.22	38.36	0.46
c6.jpg	11.32	1.48	37.63	0.54
c7.png	14.24	1.81	36.63	0.33
c8.jpg	6.36	0.95	40.13	0.28
c9.jpg	4.25	0.75	41.88	0.35
c10.tif	8.88	1.29	38.68	0.84
c11.png	26.25	2.69	33.97	0.43

Color	MSE	MAE	PSNR	CR1
c12.jpg	10.12	1.42	38.11	0.49
c13.tif	9.55	1.21	38.36	0.15
c14.png	4.66	1.04	41.48	0.18
c15.tif	1.71	0.48	45.84	0.31
c16.jpg	8.85	1.34	38.70	0.31

4.3. Discussion and Result

The following tables and figures show the improvement percentage for each quality metric (MSE, MAE and PSNR) for each image type related to the compression process:

Improvement using MDT with RIFD

The following table show the enhancement after using MDT technique on Gray 8bit images, Table (13):

Table (13): Improvement on compressing Gray 8-bit images after using MDT

	Improvement		
Gray 8 bit	MSE	MAE	PSNR
g1.gif	56%	57%	9%
g2.gif	24%	28%	3%
g3.png	23%	28%	3%
g4.GIF	37%	37%	5%
g5.png	70%	63%	14%
g6.jpg	35%	44%	5%
g7.tif	29%	32%	4%
g8.png	33%	40%	4%
g9.png	46%	46%	7%
g10.tif	32%	21%	4%

Gray 8 bit	MSE	MAE	PSNR
g11.jpg	25%	43%	3%
g12.gif	29%	44%	4%
g13.png	29%	30%	4%
g14.jpg	23%	25%	3%

Each of the following figures shows the improvement for each quality parameter

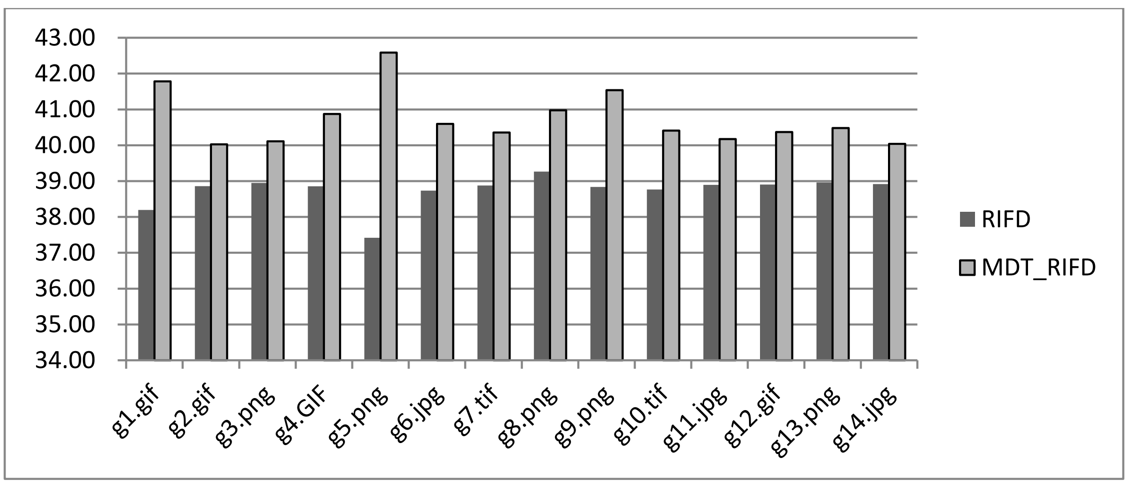


Figure (28): MSE improvement after using MDT on Gray 8bit images

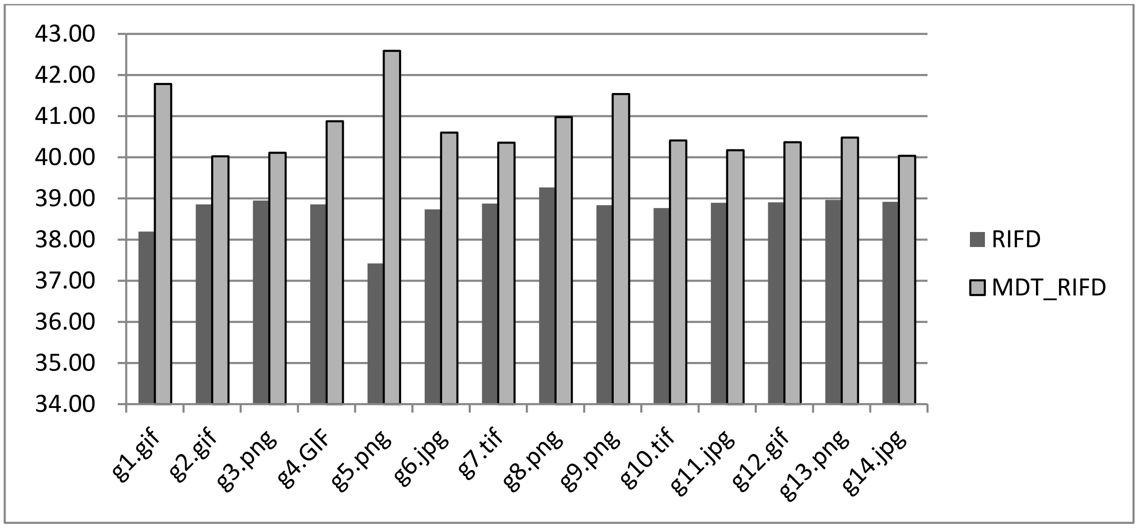


Figure (29): MAE improvement after using MDT on Gray 8bit images

The variation in MSE and MAE is clear for all images types and dimensions as shown in the figures (4.13) and figure (4.14)

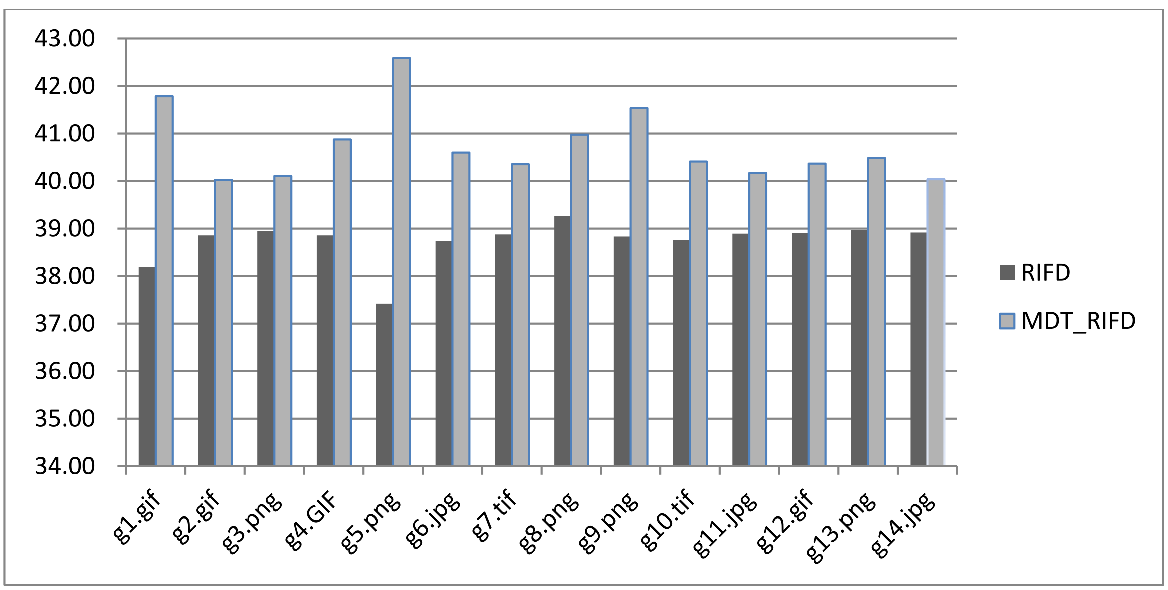


Figure (30): PSNR improvement after using MDT on Gray 8bit images

In figure (30) shows the PSNR simple variation but in few image a noticeable one.

The following table show the enhancement after using MDT technique on Gray 16bit images, see Table (14):

Table (14): Improvement on compressing Gray 16bit images after using MDT

	Improvement		
Gray 16 bit	MSE	MAE	PSNR
1.gif	21%	22%	21%
2.gif	26%	27%	26%
3.png	19%	19%	19%
4.png	15%	16%	15%
5.png	25%	25%	25%
6.png	25%	25%	25%
7.gif	28%	28%	28%
8.png	22%	22%	22%
9.png	13%	14%	13%
10.gif	43%	39%	43%
11.png	43%	55%	43%
12.png	20%	48%	20%

13.tif	43%	62%	43%
14.tif	43%	69%	43%
15.png	10%	11%	10%
16.png	23%	23%	23%

Each of the following figures shows the improvement for each quality parameter

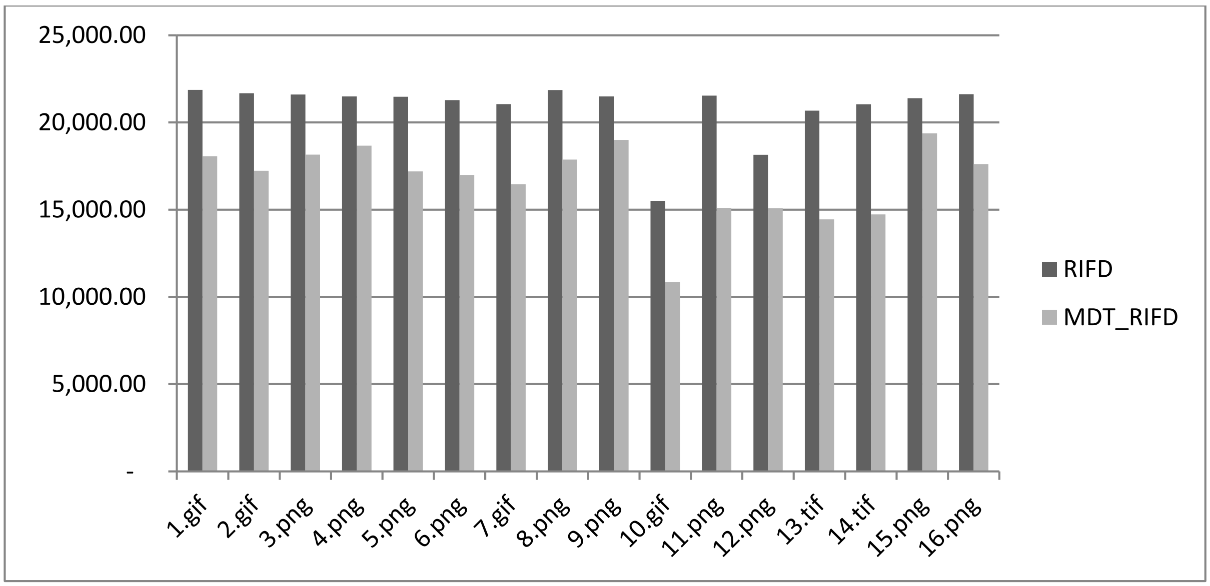


Figure (31): MSE improvement after using MDT on Gray 16-bit images

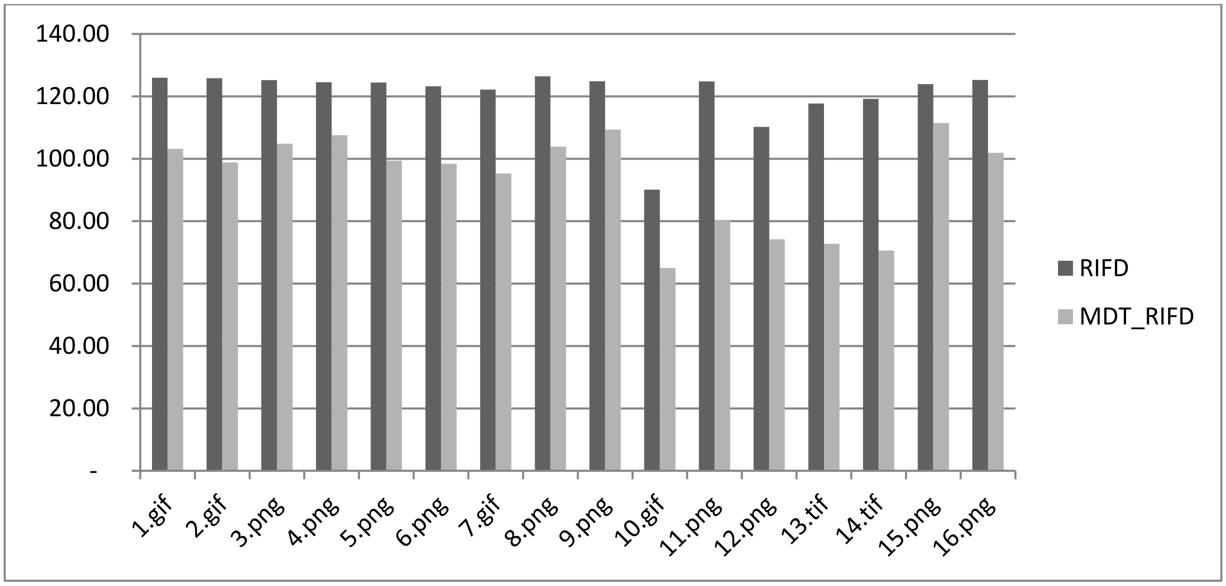


Figure (32): MAE improvement after using MDT on Gray 16-bit images

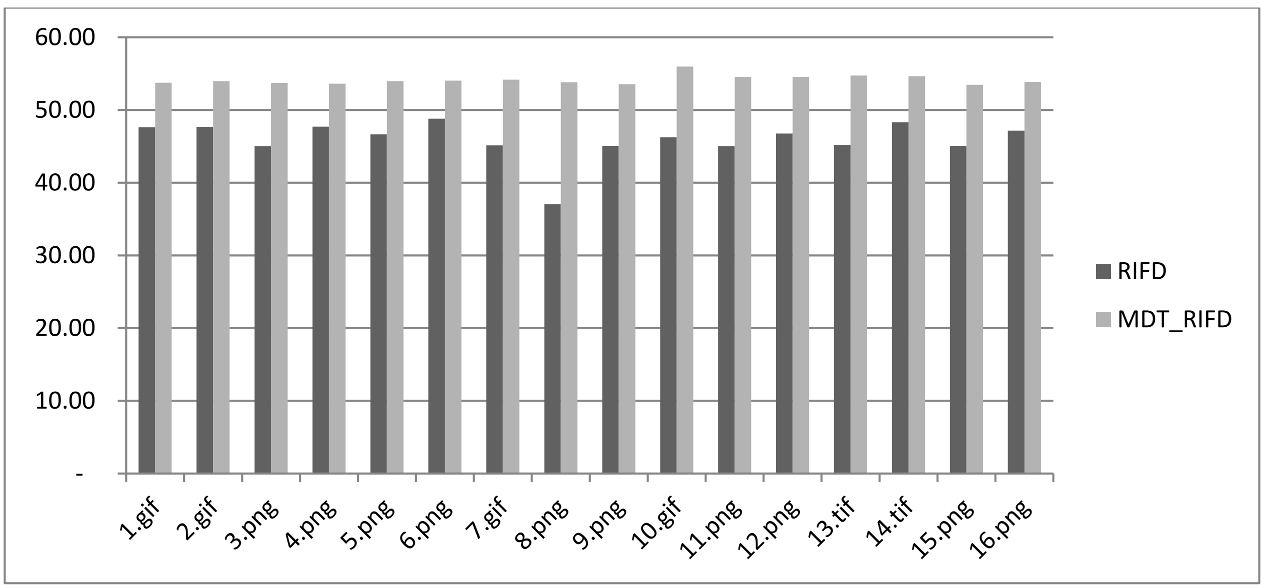


Figure (33): PSNR improvement after using MDT on Gray 16-bit images

The above figure (4.16), (4.17) and (4.18) show a slice difference between RIFD and MDT-RIFD compression.

The following table show the enhancement after using MDT technique on color images, see Table (15):

Table (15): Improvement on compressing color images after using MDT

	Improvement		
color	MSE	MAE	PSNR
c1.jpg	26%	26%	3%
c2.png	56%	49%	9%
c3.png	31%	29%	4%
c4.jpg	66%	55%	12%
c5.png	37%	34%	5%
c6.jpg	36%	33%	5%
c7.png	27%	27%	4%
c8.jpg	33%	30%	4%
c9.jpg	48%	45%	7%
c10.tif	44%	39%	7%
c11.png	25%	25%	3%
c12.jpg	31%	29%	4%
c13.tif	29%	28%	4%

c14.png	26%	25%	3%
c15.tif	59%	49%	10%
c16.jpg	26%	26%	3%

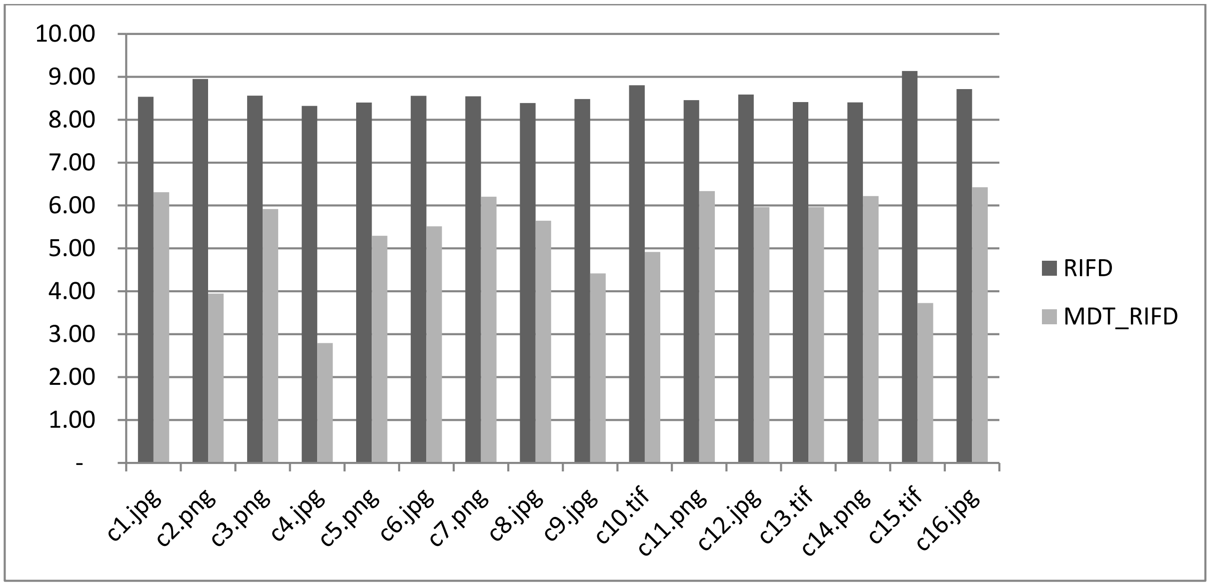


Figure (34): MSE improvement after using MDT on color images

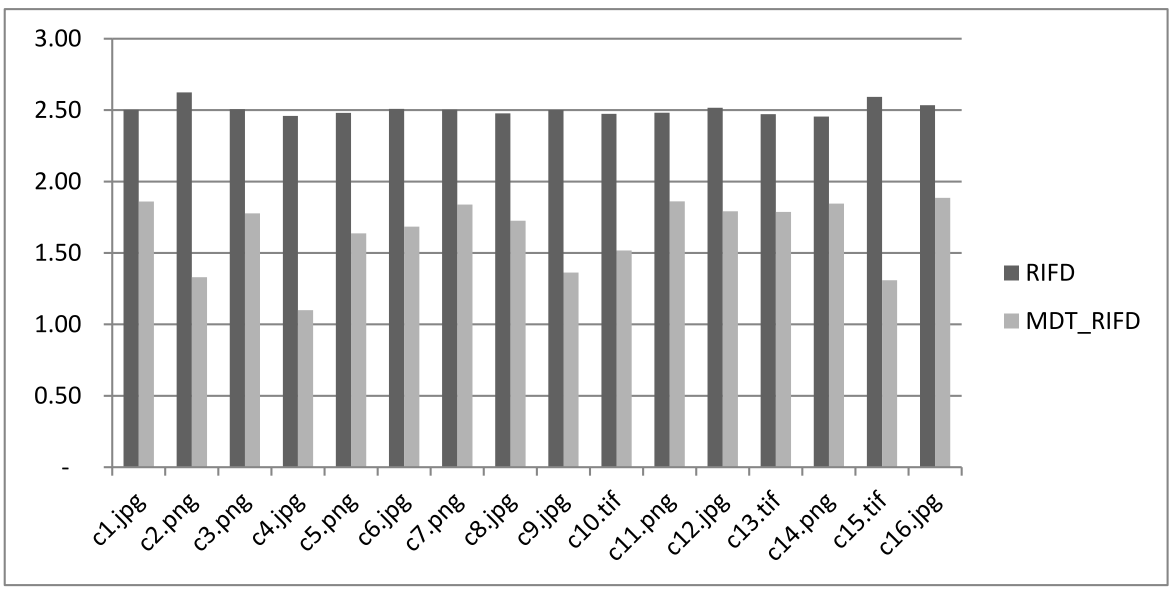


Figure (35): MAE improvement after using MDT on color images

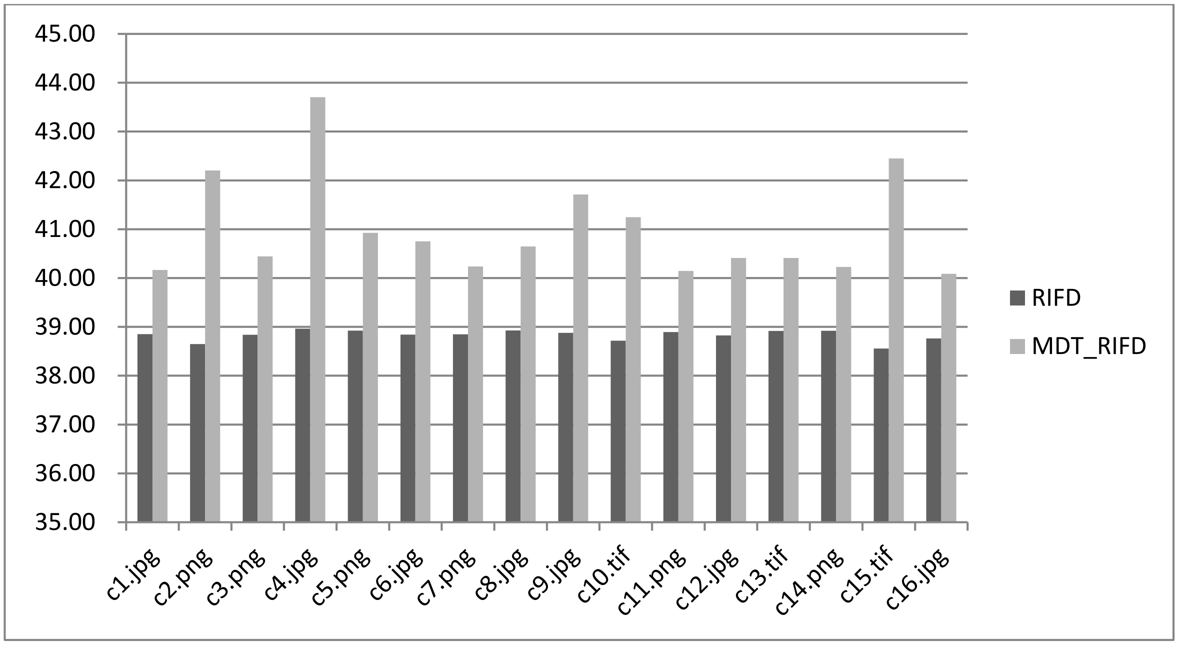


Figure (36): PSNR improvement after using MDT on color images

Improvement using MDT with JPEG

The following table show the enhancement after using MDT technique on Gray 8bit images, see Table (16):

Table (16): Improvement on compressing Gray 8-bit images after using MDT

	Improvement		
Gray 8 bit	MSE	MAE	PSNR
g1.gif	26%	24%	4%
g2.gif	66%	96%	16%
g3.png	25%	26%	4%
g4.GIF	59%	93%	13%
g5.png	84%	91%	24%
g6.jpg	6%	81%	1%
g7.tif	68%	96%	17%
g8.png	55%	96%	13%
g9.png	35%	91%	7%
g10.tif	60%	97%	13%
g11.jpg	6%	77%	1%

g12.gif	20%	52%	3%
g13.png	11%	87%	2%
g14.jpg	13%	76%	2%

Each of the following figures shows the improvement for each quality parameter

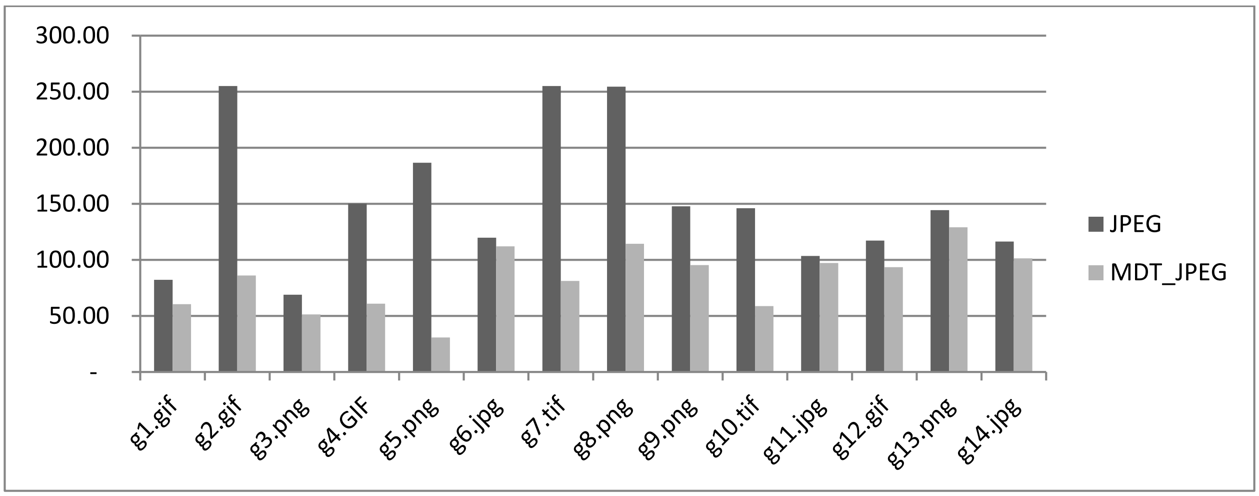


Figure (37): MSE improvement after using MDT on Gray 8bit images

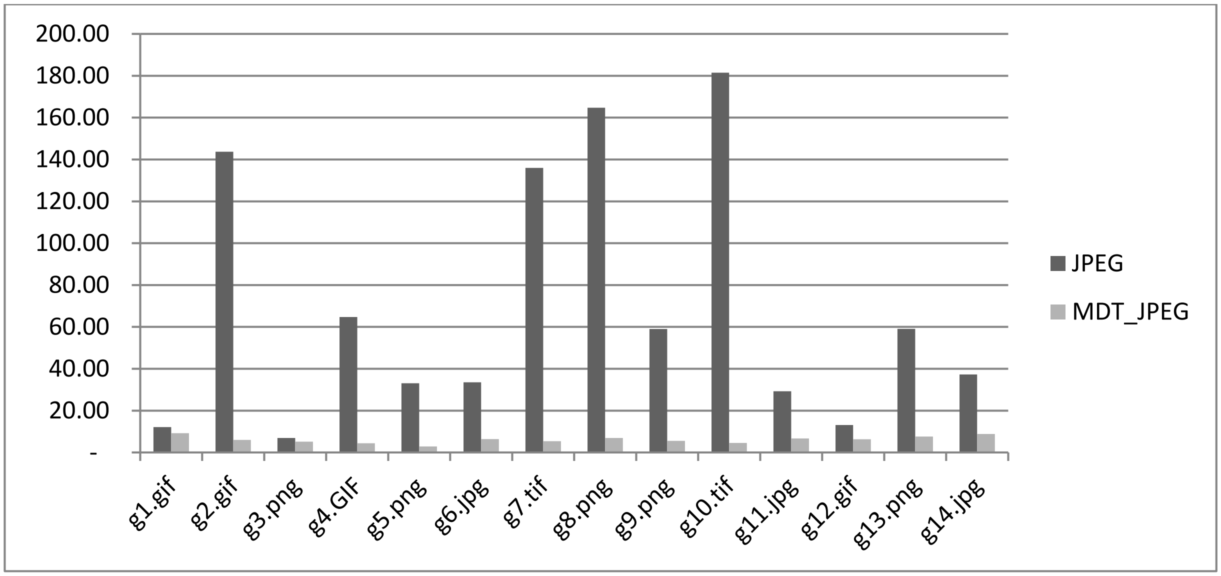


Figure (38): MAE improvement after using MDT on Gray 8bit images

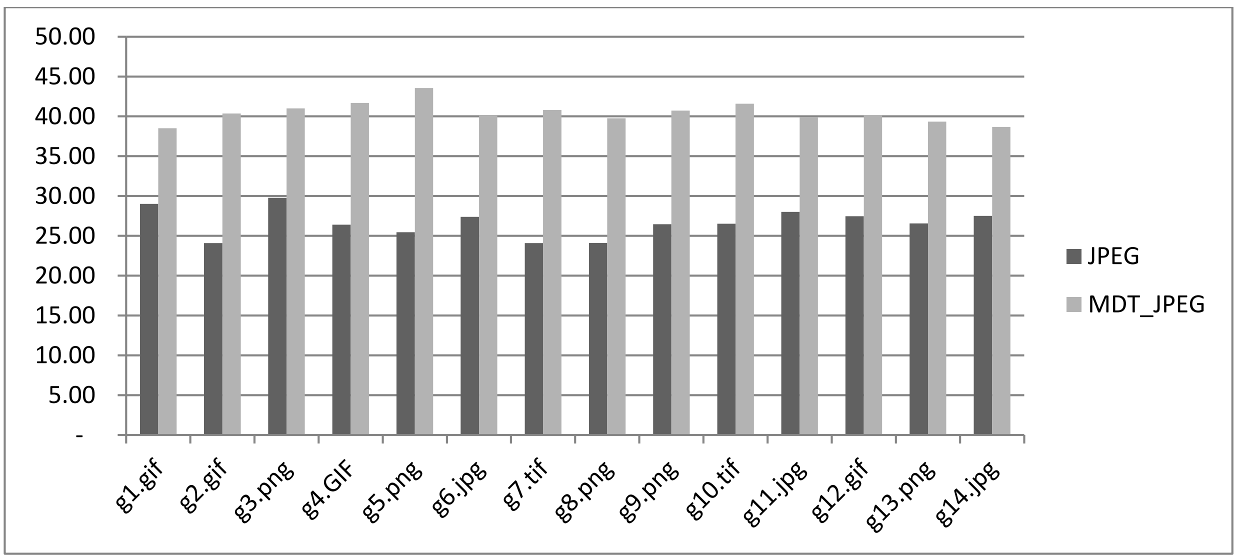


Figure (39): PSNR improvement after using MDT on Gray 8bit images

The following table show the enhancement after using MDT technique on Gray 16bit images, see Table (17):

Table (17): Improvement on compressing Gray 16bit images after using MDT

	Improvement		
Gray 16 bit	MSE	MAE	PSNR
1.gif	30%	96%	3%
2.gif	42%	96%	5%
3.png	11%	95%	1%
4.png	13%	53%	1%
5.png	18%	58%	2%
6.png	18%	85%	2%
7.gif	18%	67%	2%
8.png	9%	85%	1%
9.png	26%	87%	2%
10.gif	21%	87%	2%
11.png	14%	91%	1%
12.png	28%	96%	3%
13.tif	23%	84%	2%
14.tif	16%	83%	1%

15.png	39%	93%	4%
16.png	22%	65%	2%

Each of the following figures shows the improvement for each quality parameter

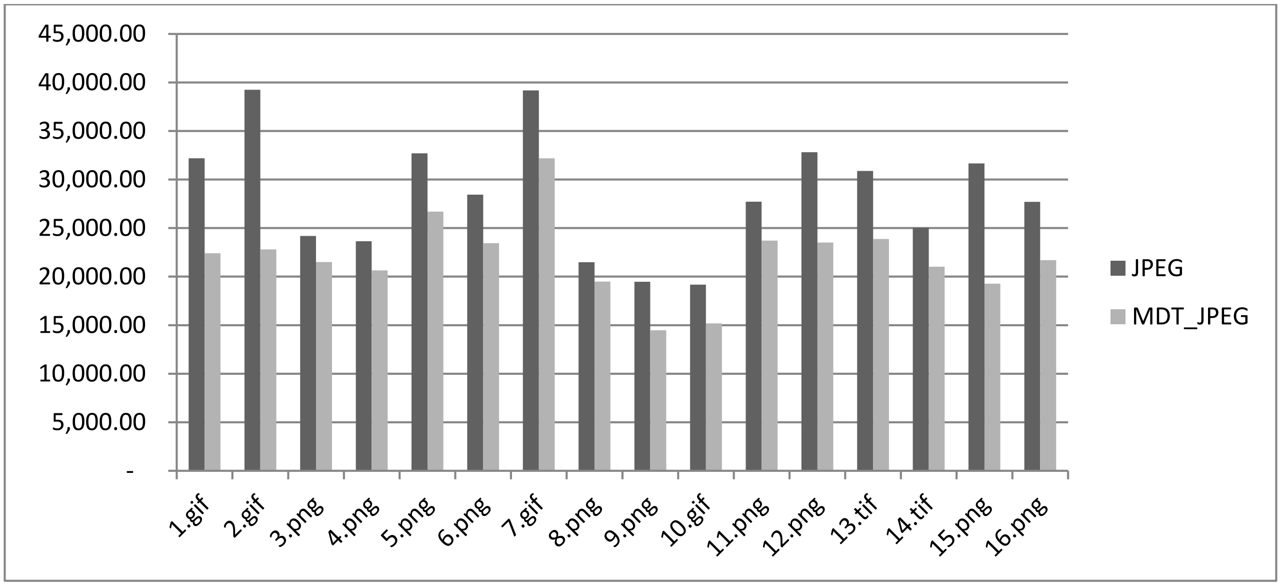


Figure (40): MSE improvement after using MDT on Gray 16-bit images

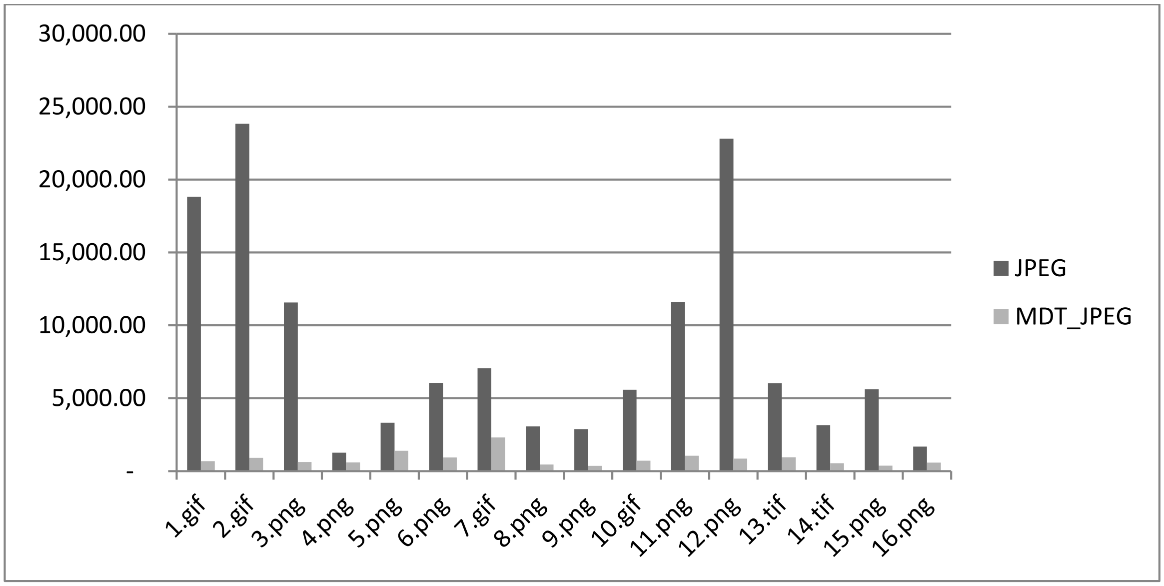


Figure (41): MAE improvement after using MDT on Gray 16-bit images

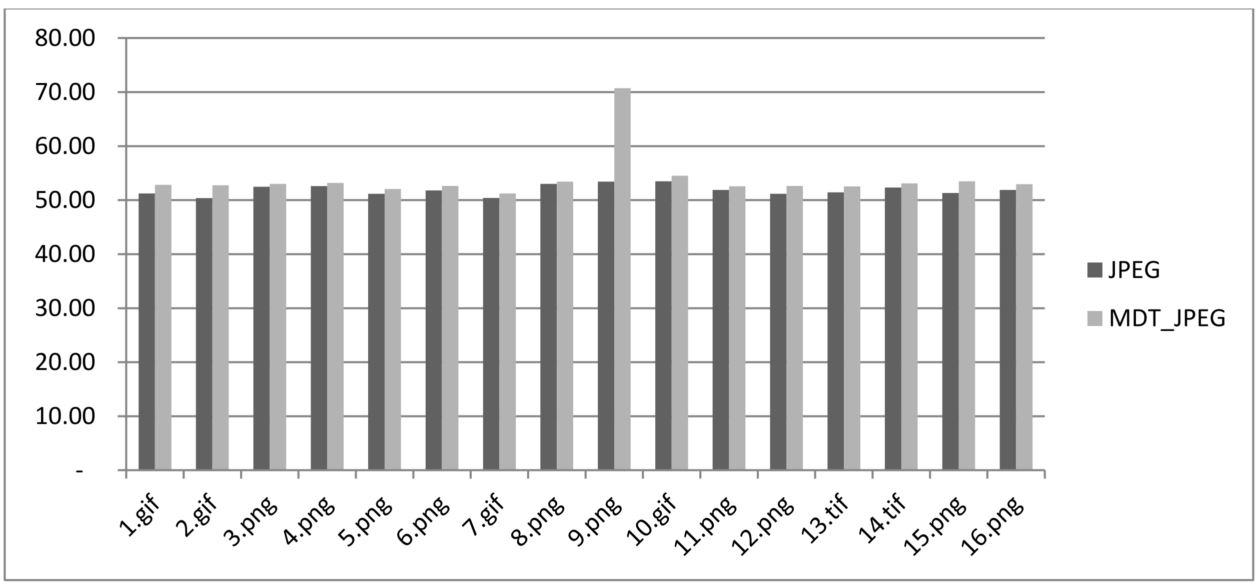


Figure (42): PSNR improvement after using MDT on Gray 16-bit images

The following table show the enhancement after using MDT technique on colors images, see Table (18):

Table (18): Improvement on compressing Gray 16bit images after using MDT

	Improvement		
Gray 8 bit	MSE	MAE	PSNR
c1.jpg	92%	85%	40%
c2.png	97%	93%	59%
c3.png	92%	83%	40%
c4.jpg	98%	93%	60%
c5.png	94%	92%	49%
c6.jpg	95%	87%	52%
c7.png	90%	86%	38%
c8.jpg	96%	93%	54%
c9.jpg	99%	95%	78%
c10.tif	90%	82%	36%
c11.png	78%	76%	24%
c12.jpg	92%	82%	39%

c13.tif	90%	86%	35%
c14.png	97%	94%	61%
c15.tif	98%	94%	61%
c16.jpg	97%	90%	63%

Each of the following figures shows the improvement for each quality parameter

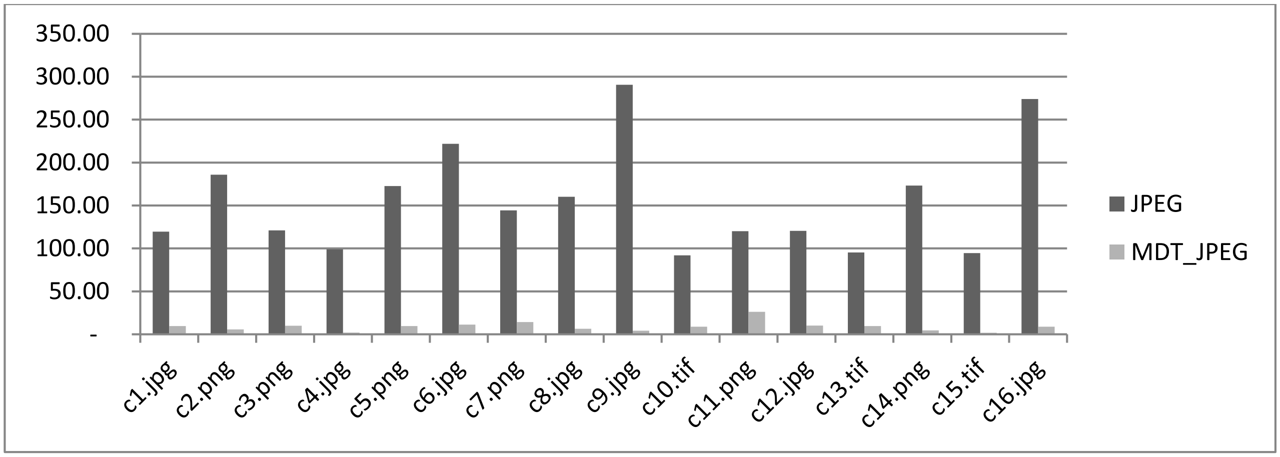


Figure (43): MSE improvement after using MDT on color images

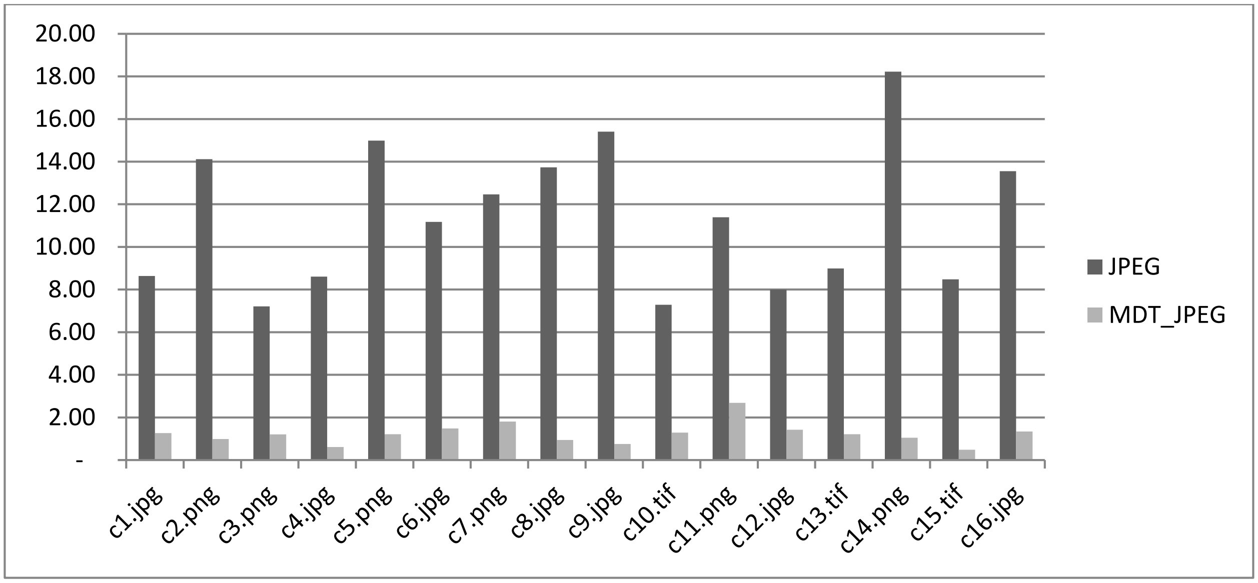


Figure (44): MAE improvement after using MDT on color images

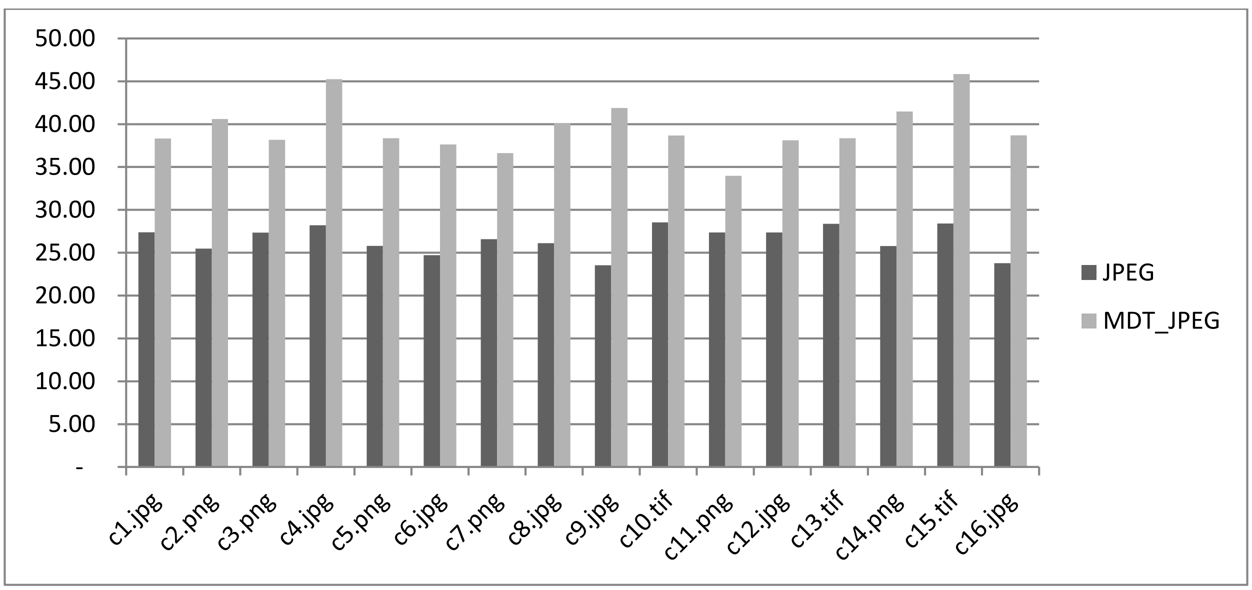


Figure (45): PSNR improvement after using MDT on color images

The MDT works better with JPEG compression since the last depends on the redundancy of zeros in raising the compression ratio using RLE encoding, and the MDT through its steps increase the number of zeros in the compressed image before using JPEG technique.

CHAPTER FIVE

CONCLUSION & FUTURE WORK

5.1. Conclusion:

This study presented lossless pre-processing steps that could be generalized to be implemented before any lossy technique and was performed on various images that varied in types, dimension, and bit-depth. Two distinct lossy techniques were chosen RIFD and JPEG to be used with of the proposed technique.

These lossy techniques vary from each other since RIFD which is novel compression technique that uses simple steps of rounding and dividing the pixels while JPEG a well known and widely used lossy compression method uses a complicated linear equations with discrete cosine transformation, this indicates that the proposed technique can be applied along with most of the available and future lossy techniques.

The MDT proposed technique are based on decreasing the minimum values of pixels in order steps to reduce the intensity used in the image, that leads to fewer bits used to represent the compressed image but with preserving the values that was reduced from the pixels. The process begins by reducing each image row by the row's minimum value then applying the same step of each column, then to increase the redundancy of zeros and more pixel intensity reducing, the image will be divided into two by two blocks and deduct the block's minimum value.

Many experiments were made by compressing many colors and gray scale images using the selected lossy techniques once, followed by compressing the same images after merging the proposed pre-process steps before the lossy technique and then compare the results.

The results showed improvement in reducing the disparity that occurs from using only the lossy techniques; this was proved by calculating the quality metrics MSE, MAE, and PSNR. Those enhancements were varied from one image to another regarding the image characteristics, and the used lossy technique was shown in the previous chapter. For example when comparing compression on Gray 8-bit the elevation between RIFD and MDT-RIFD the MSE improved (23-70%), MAE (21-63%) and PSNR (3-14%), and when comparing the enhancement of including MDT before JPEG the values improved MSE (6-8%), MAE (24-97%) and PSNR (1-3%).

The disadvantage of the proposed technique was the compression ratio that vary between decreasing increasing because of the lossless pre-processing tried to preserve the minimum values saved inside the additional arrays, only the increasing is not very high, and when evaluating the whole numbers the enhancement of the error with the limited increasing of the compression ratio is considered valuable. Also adding additional steps to the selected lossy techniques will raise the compilation time.

5.2. Future work

The suggestions and recommendations for the future work are to experiment this proposed technique on different kind of data like videos, audio and higher intensity images. it might be useful to study the use at the additional matrixes that were subtracted from the image during the prepossessing steps to hide data as in security field. Finally could be worked on a solution to improve the compression ratio.

References

Abdalla, S. and & Osman, S. (2016). Digital Image Processing Technology based on MATLAB. International Journal of Advanced Research in Computer Science, Vol. 7, No.3.

Chawla, S., Beri, M., & Mudgil, R. (2014). Image Compression Techniques: A Review. International Journal of Computer Science and Mobile Computing, Vol. 3, No. 8.

Chowdhury, H., & Khatun, A. (2012). Image compression using discrete wavelet transform. IJCSI International Journal of Computer Science Issues, Vol. 9, No. 4, PP. 327-330.

Dubey, G., & Singh, J. (2012). 3D Medical Image Compression Using Huffman Encoding Technique. International Journal of Scientific and Research Publications, Vol. 2, No. 9.

ECE, C. & Mullana, M. (2011). Image quality assessment techniques PN spatial domain. Int. J. Comput. Sci. Technol, Vol. 2, No. 3.

Fidler, A., Likar, B., & Skalerič, U. (2014). Lossy JPEG compression: easy to compress, hard to compare. Dentomaxillofacial Radiology , Vol. 35, PP 67–73.

Gupta, M. and Garg, K. (2012).Analysis of image compression algorithm using DCT. International Journal of Engineering Research and Applications (IJERA), Vol. 2, No. 1, PP. 515-521

Ha, M., Kim, K. & Yoo, H. (2016). Lossless preprocessing of floating-point data for 3D geometry data compression. In Workshop on Image Processing and Image Understanding (IPIU2016).

Hlavac, V. (2011). Fundamentals of Image Processing. Optical and Digital Image Processing: Fundamentals and Applications, PP. 71-96.

Hore, A. & Ziou, D. (2010). Image quality metrics: PSNR vs. SSIM. In Pattern recognition (icpr), IEEE 2010 International Conference on Pattern Recognition. PP. 2366-2369).

Huang, F., Huang, J., & Shi, Q. (2012). New channel selection rule for JPEG steganography. IEEE Transactions on Information Forensics and Security, Vol. 7, No. 4, PP.1181-1191.

Jassim, A. & Qassim, E. (2012).Five modulus method for image compression. Signal & Image Processing : An International Journal (SIPIJ) Vol.3, No.5, PP. 1211.4591.

Kaushik, P. & Sharma, Y. (2012). Comparison of different image enhancement techniques based upon PSNR and MSE. International Journal of Applied Engineering Research, Vol. 7, No.11, PP. 2010-2014

Kekre, B., Sarode, T., & Natu, P. (2013). Efficient image compression technique using full, column and row transforms on color image. International Journal of Advances in Engineering & Technology, Vol. 6, No.1.

Kipli, K., Muhammad, S., Masra, S., Zamhari, N., Lias, K., & Azra, D. (2012). Performance of Levenberg-Marquardt backpropagation for full reference hybrid image quality metrics. In Proceedings of International Conference of Muti-Conference of Engineers and Computer Scientists (IMECS'12).

Kumar, R. & Rattan, M. (2012).Analysis of various quality metrics for medical image processing. International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, No. 11.

Kuppusamy, K., MCA, B. & Mehala, R. (2013).Sparse Transform Matrix at Low Complexity for Color Image, Compression. International Journal of Computer Trends and Technology (IJCTT) – Vol. 4 , No. 6.

Lyon, R. (2006). 'A brief history of "pixel"', in Digital Photography II, IS&T/SPIE Symposium on Electronic Imaging, San Jose, California, PP. 15–19 January.

Otair, M. & Shehadeh, F. (2016). Research Article Lossy Image Compression by Rounding the Intensity Followed by Dividing (RIFD).Research Journal of Applied Sciences, Engineering and Technology Vol. 12, No. 6, PP. 680-685.

Pedersen, M. & Hardeberg, J. (2012). Full-reference image quality metrics: Classification and evaluation. Foundations and Trends® in Computer Graphics and Vision, Vol. 7, No. 1, PP. 1-80.

Philip, A., Oluwaranti, A., Afolabi, B., & Oluwatolani, O. (2011). Development of an image retrieval model for biomedical image databases. INTECH Open Access Publisher.

Poobal, S. & Ravindran, G. (2011). The performance of fractal image compression on different imaging modalities using objective quality measures. International Journal of Engineering Science and Technology (IJEST), Vol. 2, No. 1, PP. 239-246.

Samajdar, T. & Quraishi, M. (2015). Analysis and evaluation of image quality metrics. Information Systems Design and Intelligent Applications, PP. 369-378. Springer India

Sharma, M. (2010). Compression using Huffman coding. IJCSNS International Journal of Computer Science and Network Security, Vol. 10, No. 5, PP. 133-141.

Shinde, B. & Dani, A. (2011). The Origins of Digital Image Processing & Application areas in Digital Image Processing Medical Images. IOSR Journal of Engineering, Vol. 1, No. 1, PP. 066-071.

Sonka, M., Hlavac, V., & Boyle, R. (2014). Image processing, analysis, and machine vision. Cengage Learning.

Srikanth, S., & Meher, S. (2013). Compression efficiency for combining different embedded image compression techniques with Huffman encoding. In Communications and Signal Processing (ICCSP), 2013 International Conference on , PP. 816-820. IEEE.

Strang, G. (1999). The discrete cosine transform. SIAM review, Vol.41, No.1, PP.135-147.

Talukder, K. & Harada, K. (2010). Haar wavelet based approach for image compression and quality assessment of compressed image. IAENG International Journal of Applied Mathematics, Vol.36, No.1.

Venkataramanan, R., Sarkar, T. & Tatikonda, S. (2014). Lossy compression via sparse linear regression: Computationally efficient encoding and decoding. IEEE Transactions on Information Theory, Vol. 60, No. 6, PP. 3265-3278.

Vidal, M., & Amigo, J. (2012). Pre-processing of hyperspectral images. Essential steps before image analysis. Chemometrics and Intelligent Laboratory Systems, Vol. 117, PP: 138-148.

Vijayvargiya, G., Silakari, S. & Pandey, R. (2013). A Survey: Various techniques of image compression. arXiv preprint arXiv:1311.6877.

Wang, Z., & Li, Q. (2011). Information content weighting for perceptual image quality assessment. IEEE Transactions on Image Processing, Vol. 20, No. 5, PP: 1185-1198.

Wolfgang, R., Podilchuk, C., & Delp, E. (1999). Perceptual watermarks for digital images and video. Proceedings of the IEEE, Vol. 87, No. 7, PP: 1108-1126.

Yeganeh, H., & Wang, Z. (2013). Objective quality assessment of tone-mapped images. IEEE Transactions on Image Processing, Vol. 22, No. 2, PP: 657-667.

Young, T., Gerbrands, J., & Van Vliet, J. (1998). Fundamentals of image processing. The Netherlands: Delft University of Technology.

Zhou, N., Zhang, A., Zheng, F., & Gong, L. (2014). Novel image compression–encryption hybrid algorithm based on key-controlled measurement matrix in compressive sensing. Optics & Laser Technology, Vol. 62, PP. 152-160.

APPENDIX

Appendix (A): Set of the images used in the experiments

Gray 8-bit

g1

g2

g3

g4

g5

g6

g7

g8

g9

g10

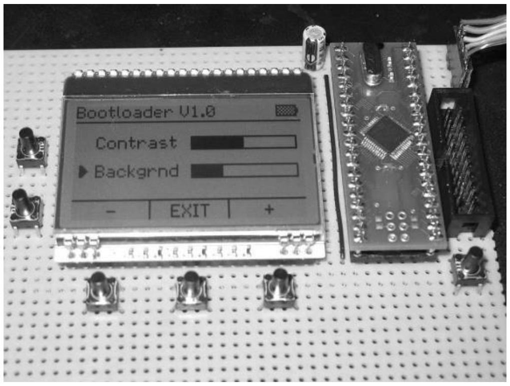
g11

g12

g13

g14

Gray 16-bit



1

2

3

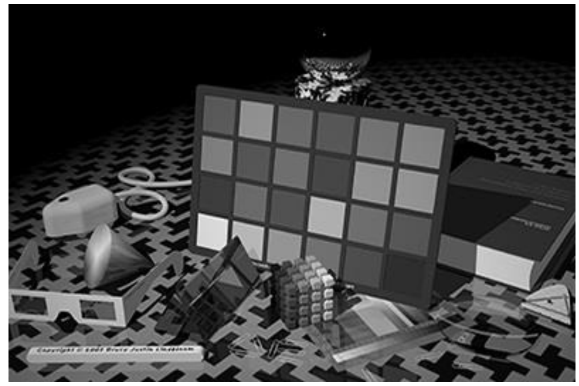
4

5

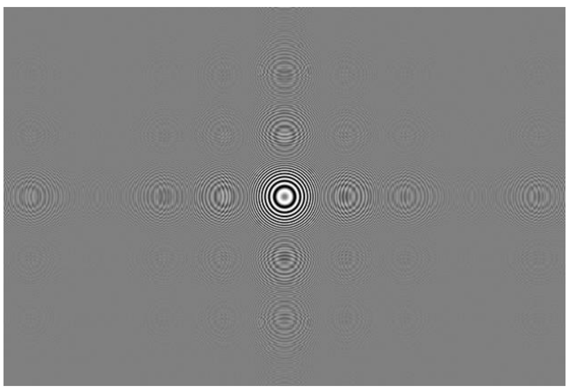
6

7

8



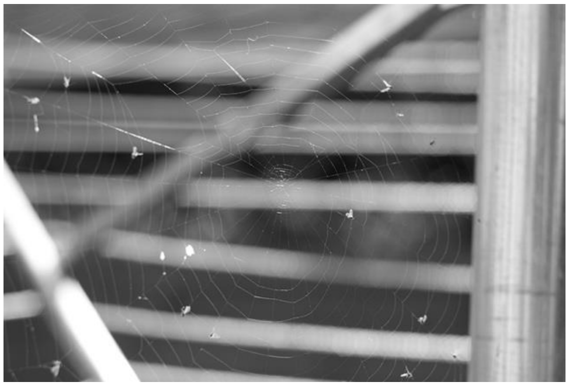
9



10

11

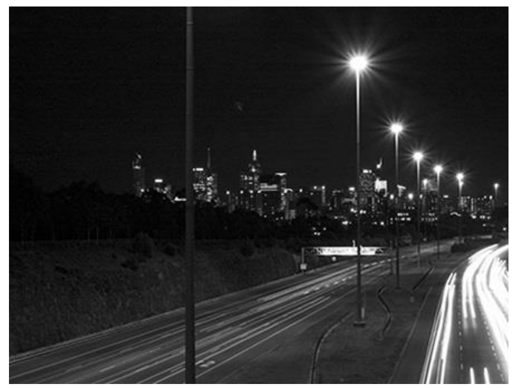
12



13

14

15



16

Color 8-bit

C1

C2

C3

C4

C5

C6

C7

C8

C9

C10

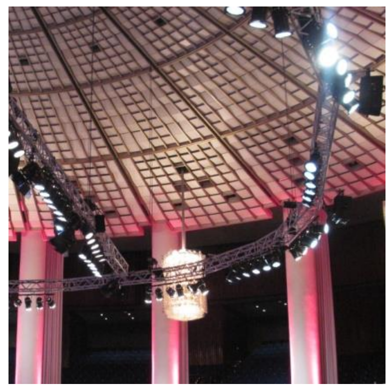
C11

C12

C13

C14

C15



C16

Appendix (B): MDT Matlab code using Matlab ver. R2015a:

```
% (img) read image of any type and in any type (.tif,.jpg,.gif,.png)

% (new_img) image after decreasing min of row

% (new_img1) image after decreasing min of column

% (new_img2) image after decreasing min of 2x2 block


%% read image

img = imread(file_name1);


% read image size

[a,b]=size(img);


%% work on min row

% find minimum value in each row

for z = 1:a

min_per_row(1,z) = min(img(z,:));

end
```

```
% subscribt each row from the minimum
for w = 1:a
    for k = 1:b
    new_img(w,k) = img(w,k)- min_per_row(1,w);
    end
end

%% work on min column
% find minimum value in each col
for z = 1:b
min_per_col(1,z) = min(new_img(:,z));
end
% subscribt each col from the minimum
for k = 1:b
    for w = 1:a
    new_img2(w,k) = new_img(w,k)-min_per_col(1,k);
    end
end
```

```
%% find 2x2 min value

% find where to put the 2x2 mask iin x axis

% find if matrix x axis is even or odd

if mod(a,2) == 0

   % if even

   max_even_a = a-1;

else % if odd

   max_even_a = a-2;

end

% find where to put the 2x2 mask iin y axis

if mod(b,2) == 0

max_even_b = b-1;

else

max_even_b = b-2;

end

% find minimum value in each 2x2 mask

count1 = 0;

for h = 1:2:max_even_a

   count1 =count1+1;

   count2 = 0;
```

```
    for hh = 1:2:max_even_b
        count2 =count2+1;
% subscribt each 2x2 mask from the minimum value
% if image is 8bit min1=256, if image is 16bit min1=65536

        min1 = bitdepth ^2;
        for j = h:h+1
            for m = hh:hh+1
             if new_img2(j,m)< min1
                 min1=new_img2(j,m);
             end
            end
        end
    min_per_seq(count1,count2)= min1;
    end
end
```

```
% subscribt each 2x2 mask from the minimum value
count1 = 0;
for h = 1:2:max_even_a
   count1 =count1+1;
   count2 = 0;
   for hh = 1:2:max_even_b
      count2 =count2+1;
      min1 = 256;
      for j = h:h+1
         for m = hh:hh+1
            new_img2(j,m)=new_img2(j,m)- min_per_seq(count1,count2);
         end
      end
   end
end
```

Appendix (C): RIFD Matlab code using Matlab ver. R2015a:

```
% (new_img2) read image after decreasing min of 2x2 block

% (new_img3) image with rounding process implement

% (new_img4) image after devide by 10


%% read image

new_img2 = imread(file_name1);


%% read image size

[a,b]=size(new_img2);


%% rounding the image

If bidepth = 8

     Rounding = 10

  Else rounding = 1000;

end;

new_img3= rounding*(round((new_img2-1)/rounding));
```

```
%% apply Huffman
%% read image after rounding
h = new_img3;
[m,n] = size ( r);
totalcount = m*n;

%% variable to count the prop
cnt = 1;  sigma = 0;

%% computing the comulative prop
%% computing the maximum bits value
bitt = (2^bitdepth)-1;

for i=0:bitt
k = r == i;
count(cnt) = sum (k(:));
```

```
% pro array is having the prop
pro (cnt) = count (cnt) / totalcount;
sigma = sigma + pro (cnt);
cumpro (cnt) = sigma;
cnt = cnt +1;
end;


%% symbols for an image
symbols = [0:bitt];


%% Huffman code Dictionary
dict = huffmandict (symbols, pro);
```

```
%% function which convert array to vector
vec_size = 1;
for p = 1:m
for q = 1:n

newvec (vec_size) = r (p,q);
vec_size = vec_size + 1;

end
end

%% huffman Encoding
hcode = huffmanenco (newvec, dict);
leng = length (hcode);
```

Appendix (D): JPEG Matlab code using Matlab ver. R2015a:

%% LOSSY COMPRESSION USNIG DISCRETE COSINE TRANSFORM TECHNIQUE.

% "filename" is image name and its extension

N=8; % Block size for which DCT is Computed.

M=8;

I=imread(filename); % Reading the input image file and storing

intensity values in 2-D matrix I.

I_dim=size(I); % Finding the dimensions of the image file.

I_Trsfrm.block=zeros(N,M); % Initialising the DCT Coefficients Structure

Matrix "I_Trsfrm" with the required dimensions.

Norm_Mat=[16 11 10 16 24 40 51 61 % Normalization matrix (8 X 8) used

to Normalize the DCT Matrix.

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];
%% Computing Quantized & Normalized DCT.
for a=1:I_dim(1)/N
    for b=1:I_dim(2)/M
        for k=1:N
            for l=1:M
                prod=0;
                for i=1:N
                    for j=1:M
                        prod=prod+double(I(N*(a-1)+i,M*(b-1)+j))*cos(pi*(k-1)*(2*i-1)/(2*N))*cos(pi*(l-1)*(2*j-1)/(2*M));
                    end
                end
                if k==1
                    prod=prod*sqrt(1/N);
                else
```

```
            prod=prod*sqrt(2/N);
          end
          if l==1
            prod=prod*sqrt(1/M);
          else
            prod=prod*sqrt(2/M);
          end
          I_Trsfrm(a,b).block(k,l)=prod;
        end
      end
      % Normalizing the DCT Matrix and Quantizing the resulting values.
      I_Trsfrm(a,b).block=round(I_Trsfrm(a,b).block./Norm_Mat);
    end
  end
```

```
%% zig-zag fashion coding

for a=1:I_dim(1)/N

    for b=1:I_dim(2)/M

        I_zigzag(a,b).block=zeros(1,0);

        freq_sum=2:(N+M);

        counter=1;

        for i=1:length(freq_sum)

            if i<=((length(freq_sum)+1)/2)

                if rem(i,2)~=0

                    x_indices=counter:freq_sum(i)-counter;

                else

                    x_indices=freq_sum(i)-counter:-1:counter;

                end

                    index_len=length(x_indices);

                    y_indices=x_indices(index_len:-1:1); % Creating reverse of the array
as "y_indices".
```

```
                for p=1:index_len
                   if I_Trsfrm(a,b).block(x_indices(p),y_indices(p))<0
bin_eq=dec2bin(bitxor(2^n1,abs(I_Trsfrm(a,b).block(x_indices(p),y_indices(p)))),
n);
   else

bin_eq=dec2bin(I_Trsfrm(a,b).block(x_indices(p),y_indices(p)),n);
                   end
                   I_zigzag(a,b).block=[I_zigzag(a,b).block,bin_eq(1:m)];
                end
         else
            counter=counter+1;
            if rem(i,2)~=0
               x_indices=counter:freq_sum(i)-counter;
            else
               x_indices=freq_sum(i)-counter:-1:counter;
            end
               index_len=length(x_indices);
% Creating reverse of the array as "y_indices".
```

```
y_indices=x_indices(index_len:-1:1);
for p=1:index_len
            if I_Trsfrm(a,b).block(x_indices(p),y_indices(p))<0
              bin_eq=dec2bin(bitxor(2^n-1,abs(I_Trsfrm(a,b).block(x_indices(p),y_indices(p)))),n);
            else
bin_eq=dec2bin(I_Trsfrm(a,b).block(x_indices(p),y_indices(p)),n);
            end
            I_zigzag(a,b).block=[I_zigzag(a,b).block,bin_eq(1:m)];
          end
      end
    end
  end
end
```

```
% Run-Length Encoding the resulting code.
enc_len = 0;
for a=1:I_dim(1)/N
    for b=1:I_dim(2)/M

        % Computing the Count values for the corresponding symbols and
        % savin them in "I_run" structure.
        count=0;
        run=zeros(1,0);
        sym=I_zigzag(a,b).block(1);
        j=1;
        block_len=length(I_zigzag(a,b).block);
        block_len;
        for i=1:block_len
            if I_zigzag(a,b).block(i)==sym
                count=count+1;
            else
                run.count(j)=count;
                run.sym(j)=sym;
```

```
        j=j+1;
        sym=I_zigzag(a,b).block(i);
        count=1;
    end
    if i==block_len
        run.count(j)=count;
        run.sym(j)=sym;
    end
end
```

```
% Computing the codelength needed for the count values.
dim=length(run.count);  % calculates number of symbols being encoded.
maxvalue=max(run.count);  % finds the maximum count value in the
count array of run structure.
codelength=log2(maxvalue)+1;
codelength=floor(codelength);
```

```
% Encoding the count values along with their symbols.

I_runcode(a,b).code=zeros(1,0);

dim;

for i=1:dim

I_runcode(a,b).code=[I_runcode(a,b).code,dec2bin(run.count(i),codelength),run.sym(i)];

end

enc_str=I_runcode(a,b).code;

enc_len1 = length (enc_str);

%% summation of all compressed vectors that represent the image size

enc_len = enc_len + enc_len1;

end

end
```