Implementation of Digital Image Compression Using The Comparison Between The Adjacent Pixel On RGB Image With Variant Brightness And Contrast

Adem A. bensaid[1], Jalal M. Mehalhal[2*], *Muhmed F. Agbisha*[1]

[1] A.bensaid@uot.edu.ly, [2] Enjjalal@yahoo.com, [3] abuabdualaziz2240@gmil.com

[1,]Department of computer, faculty of Education Qasser Bin Ghesheer, Tripoli University, Libya

[2]High Institute-Alkhoms, Dept. of Computer Engineering, Alkhoms–Libya

Abstract

Information technology has witnessed a great development in the terms of the huge amount of information exchanged, the ways of exchanging this information, and the speed of its exchange.

One of the most essential media used to transfer data is the digital image. Image compression plays a very important role in the transfer and storing of image data due to storage limitations. The main goal of image compression is to represent the image with the fewest number of bits without losing the basic information content within the original image.

Digital images contain a large amount of digital information that needs efficient technologies to be stored and to transmit a large volume of data.

In this study, a method was used to compress digital images, in which the bits of each pixel are compared with the pixels adjacent to it, and the result of the comparison is a new code to represent the bits of the second pixel, and its size is different, either increasing or decreasing, and calculating the image size after performing the compression process.

Each pixel was divided into two parts, the right part of it was compared with the right part of the adjacent pixel, while the left part was compared to the left part of the adjacent pixel, therefore, a new code was obtained.

This method was applied on a group of images with different content to obtain good results.

Keywords—Data Compression, Lossless Compression, Lossy Compression

1. INTRODUCTION

With the current technology development and the increase of using computers, the excessive production, transmission and sharing of digital images, long-term storage and efficient data transmission, data compression is critical. Over the past decade, several compression techniques have been proposed. Some researchers have focused on general data compression, while others have designed more specific file compression techniques for movie, audio, and image files. When the file type is known, better compression results can be achieved with less effort because better prediction models can be designed.[1]

It should be noted that the issue of image compression is one of the important topics that occupies the forefront in the field of image processing, as recent studies have taken a broad interest in this field. for the image) [2]

The need for such a topic has increased as a result of the rapid and tremendous growth in the multimedia markets and the emergence of the Internet, as well as the development in motion picture technology, that most modern operating systems depend on graphics or images, and because the image data is very large only led to great interest in finding techniques for data compression Imagery in order to reduce the memory space needed to store these images, and there are several techniques for data compression and the choice of technology depends on the type of data to be compressed.

digital image production, transmission, and sharing have been excessively increased. The more small electronic devices that incorporate cameras and provide the user with technologies to share captured images directly on the Internet, the more storage devices will understand the necessity of efficiently storing a huge amount of image data.

The main goal of image compression methods is to represent the original images with fewer bits.

Compression techniques are essentially lossless. In the first, two-dimensional reversible transformations are used. Loss modes are particularly suitable for natural images where slight loss is acceptable. On the other hand, lossless compression is preferred for medical imaging, technical graphics, and clipart where every pixel information is important. [3]

Image lossless compression can recover the original image after the rebuilding, Because it is impossible to compress an image with a high compression ratio without errors, an image with loss of data compression was used to obtain high compression ratios. Thus, image size is reduced, and lossy image compression gives a more favorable ratio of image compression without data loss.

In this study, we have used an algorithm that compresses data based on a comparison between bits in the same cell, and better results were obtained for images with less contrast. Pressure, pressure factor and savings ratio.

2. BACKGROUND OF THE STUDY

2.1 Image Compression

Image compression methods are used as a try to reduce the amount of data required to represent a digital image. It is a process that aims to produce a compressed representation of the image, thus reducing the requirements for storing and transmitting images. Compression is achieved by removing one or more of the three primary redundancies:

1. Coding Redundancy 2. Interpixel Redundancy 3. Psychovisual Redundancy Coding redundancy is present when less than optimal code words are used. Interpixel redundancy results from correlations between the pixels of an image. Psychovisual redundancy is due to data that is ignored by the human visual system (i.e. visually non essential information). Image compression techniques reduce the number of bits required to represent an image by taking advantage of these redundancies.[4]

2.2 Image Compression Techniques

The image compression techniques are broadly classified into two categories depending whether or not an exact replica of the original image could be reconstructed using the compressed image. These are:

1. Lossless technique
2. Lossy technique. [5]

2.3 Inter Pixel Redundancy

"In image neighboring pixels are not statistically independent. It is due to the correlation between the neighboring pixels of an image. This type of redundancy is called Inter-pixel redundancy. This type of redundancy is sometime also called spatial redundancy. This redundancy can be explored in several ways, one of which is by predicting a pixel value based on the values of its neighboring pixels. In order to do so, the original 2-D array of pixels is usually mapped into a different format, e.g., an array of differences between adjacent pixels. If the original image pixels can be reconstructed from the transformed data set the mapping is said to be reversible".[2]

2.4 Basic flow of Image Compression Coding

Image compression coding involves storing the image in the form of bit-stream and then trying to display thedecoded image as exact as possible. It has two functional units: an encoder and a decoder as shown in Figure 1. The original image is fed into encoder then encoder converts it into series of binary data i.e. bit-stream and sends to the decoder unit of image compression coding where decoder decodes the encoded data into form of decoded image. To compress the image successfully, data quantity of original image must be more than that of bit stream.[6

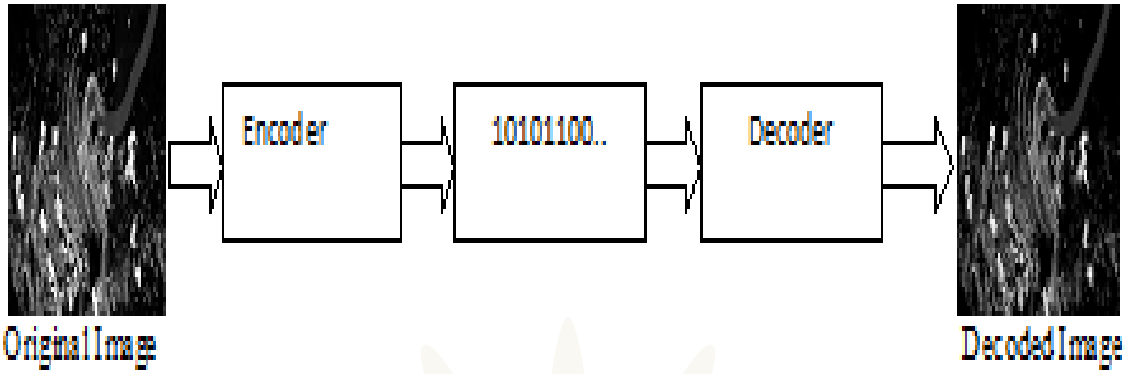


Figure 1: Flow of Image Compression Coding

3. RELATED WORK:

In [3] study points out Lossy as well as Lossless compression techniques as they are used in fields of image processing.

Researchers consider it necessary to improve lossless compression techniques.

In [1] it is proposed to develop special compression techniques to reduce the size of the image data. It used pseudo-distance technology (PDT) for oscillating images, which played a role in the best compression results for GIF and PNG files. Expected results in the PDT matrix.

PROPOSED METHOD

3-1 Introduction

The method was used in which the comparison between the bits of each pixel and the adjacent pixel was used so that the result of the comparison would be a new code to represent the second pixel bits and its size would be different either by increase or decrease, and by calculating the size of the image after the compression process.Each pixel is divided into two parts and the right section of it is compared to the right section of the adjacent pixel and the left section to the left section of the pixel next to it, and get a new blade.[7]

3.2 The Compression Method Using Bit-Per-Pixel Comparison

In this way, the values of each color of the pixel have been converted into a correct number representing the color and the value of this number has been converted to the corresponding in the binary system, and then the comparison process begins [1] [4], and the steps are **as follows:**

1- Converting the values of each color of the pixel into a correct number representing the color and converting the value of this number to the corresponding in the binary system,

2- The binary code of the first pixel is divided into two parts and the binary code of the second pixel into two so that the left part of the first pixel is compared with the left part of the second pixel and the right part of the first pixel is compared with the right part of the second pixel. One of the following cases will be inferred:

A. If the right end of the first pixel equals the right end of the second pixel, (1) is added. The left end of the first pixel is equal to the left end of the second pixel and is added.1

example:

Pixel	99	99	Size
The bicode of the pixel	11000110	11000110	16
The new code	11000110	11	10

The new size is therefore 10 bits less than the original size of 16 bits by 6 bits.

B- If the right end of the first pixel is not equal to the right end of the second pixel. The left end of the first pixel is equal to the left end of the second pixel, a value of 10 is placed with the value of the right end of the second pixel.

example:

Pixel	110	30	Size
The bicode of the pixel	01110110	01111000	16
The new code	01110110	101000	14

So the new size is 14 bits, which is 16 bits less than the original size by 2 bits.

C- If the right end of the first pixel is equal to the right end in the second pixel and the left end of the first pixel to equal the left end of the second pixel. The value of 01 is placed with the right end value of the second pixel.

example:

Pixel	89	94	Size
The bicode of the pixel	10011010	01111010	16
The new code	10011010	010111	14

So the new 14-bit is 16 bits less than the original size by 2 bits

D. In the case of the right end in the first pixel, the right end is equal to the second pixel and the left end in the first pixel is not equal to the left end in the second pixel. The value of 00 is placed with the right end value of the second pixel.

example:

Pixel	185	194	Size
The bicode of the pixel	10011101	01000011	16
The new code	1001101	0001000011	18

The new size is 18 bits more than the original size of 16 bits by 2 bits.

PRACTICAL PART

4.1 Introduction

The algorithm was executed on a set of images of different types and different divisions per byte and MATLAB was used, and the algorithm's compression ratio for these images was assessed.

4.2 Application Environment

When the MATLAB is turned on, one or more windows are opened on the computer screen, and one of these windows is called MATLAB desktop, this window is the primary graphic communication interface with users and within this window there is another window, command window, which is the primary place where the MATLAB is interacted with, the trigger signal is shown >> in the command window as it appears to the right of this inductor – when the command window is active – indicating flash (appear and disappear), and this The flash induced teaches you that the MATLAB is waiting for you to enter the MATLAB processes in this project, the COMMAND window has been used while the result is in MATLAB desktop window.

4.2.1 Image Processing Using MATLAB

There are a set of commands that have been used to process the image from reading it and converting each color value to the corresponding one in the decimal system, ending with an calculation of the size of the compressed image.

4-3 Implement The Program On The Images In Case Each Byte Is Divided Into Two Parts, Each Section Consists Of 4 Bits.

The program was implemented on a range of types of images taking into account the brightness and contrast ratio.

The images ware read in **Figure 2**: where the brightness and contrast different ratio:

a) RGB image with brightness and contrast = 0

b) RGB image with brightness =-30% contrast = - 20%

c) RGB image with brightness = + 40% contrast = - 40%

d) RGB image with brightness and contrast = 0

Figure (2)

5- RESULTS

The results obtained after the algorithm is implemented on RGB image will be clarified in a lateral manner.

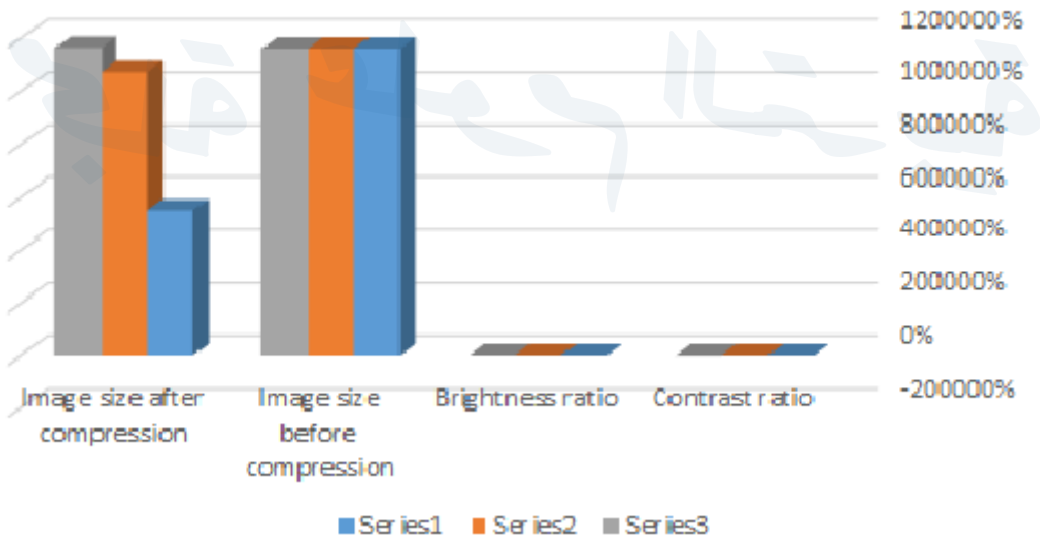
After studying this method, he found the **following**:

The size of the image decreases when the bits are divided into two groups, a group consisting of 6 bits and a group of 2 bits, and the size of the image has increased when the contrast ratio is =-40% and the brightness ratio = +40%. As the tablenext:

Table (5-1) image size if the group is divided into 6-2 bits

Image type	RGB		
Contrast ratio	%0	%20 -	%40 -
Brightness ratio	%0	%30 –	%40 +
Image size before pressing	11616 Bits	11616 Bits	11616 Bits
Image size after pressure	5500 Bits	10772 Bits	11642 Bits

Figure (3) image size if the group is divided into 6-2 bits

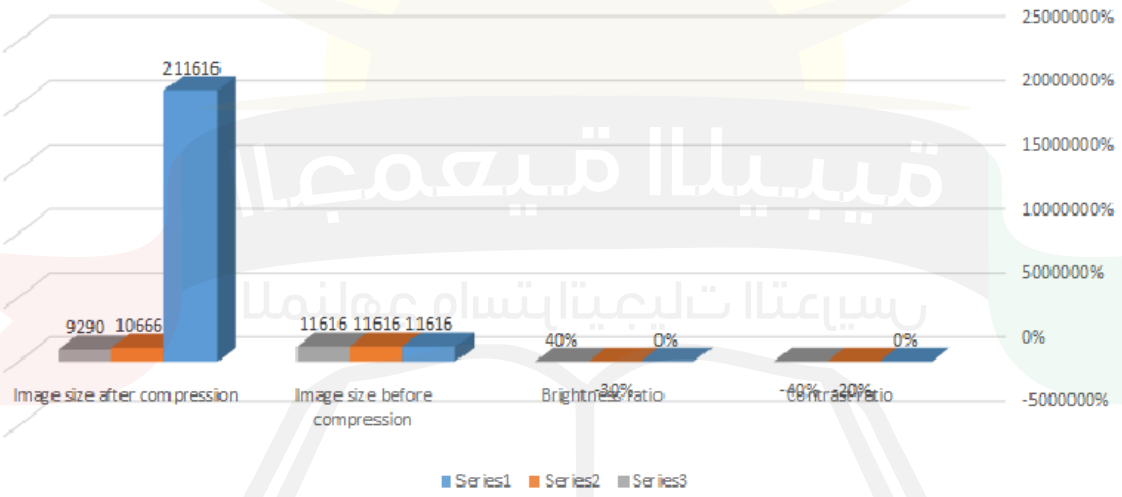


• The size of the image decreases when the bits are divided into two groups, a group consisting of 2 bits and a group of 6 bits, and the size of the image has increased when the contrast ratio is -0% and the brightness ratio is 0%. As the following table:

Table (5-2) image size if the group is divided into 2-6 bits

Image type	RGB		
Contrast ratio	%0	%20 -	%40 -
Brightness ratio	%0	%30 –	%40 +
Image size before pressing	11616 Bits	11616 Bits	11616 Bits
Image size after pressure	211616 Bits	10666 Bits	9290 Bits

Figure (4) image size if the group is divided into 2-6 bits

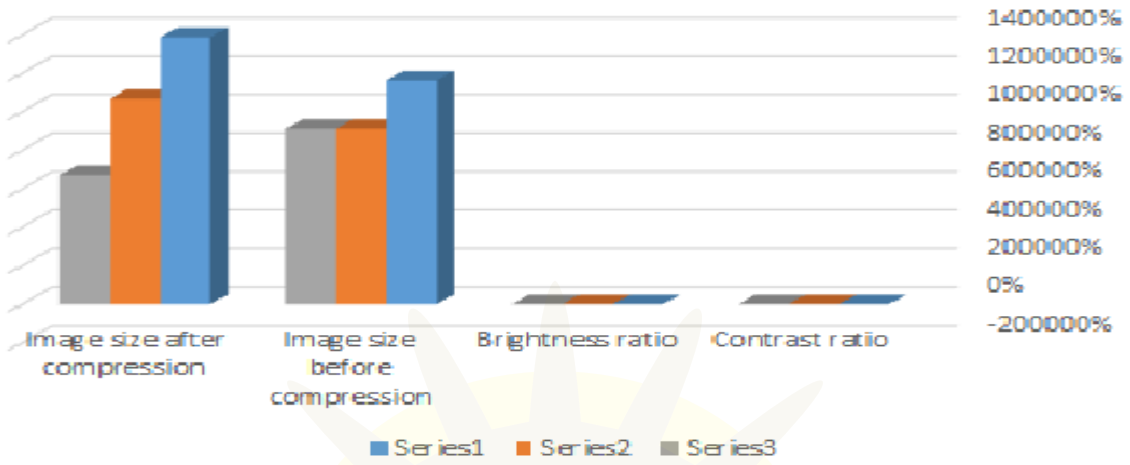


• Increase the size of the image when the bits are divided into two groups, each group consisting of 4 bits. As in the following table:

Table (5-3) image size if the group is divided into 4-4 bits

Image type	RGB		
Contrast ratio	%0	%20 -	%40 -
Brightness ratio	%0	%30 –	%40 +
Image size before pressing	11616 Bits	9120 Bits	9120 Bits
Image size after pressure	13878 Bits	10666 Bits	6718 Bits

Figure (5) image size if the group is divided into 4-4 bits

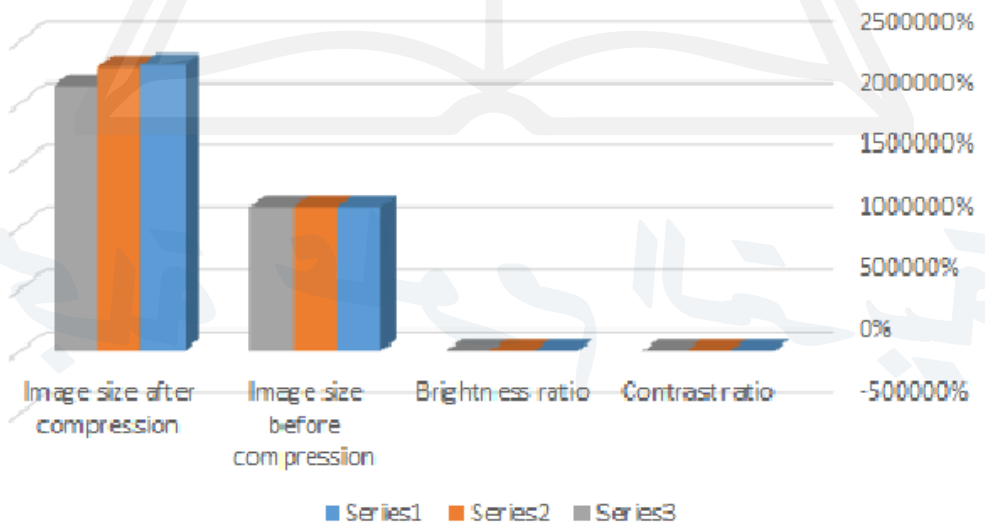


• Increase the size of the image when dividing the bits into two groups, a group consisting of 3 bits and a group consisting of 5 bits. As in the following table:

Table (5-4) image size if the group is divided into 3-5 bits

Image type	RGB		
Contrast ratio	%0	%20 -	%40 -
Brightness ratio	%0	– %30	%40 +
Image size before pressing	11616 Bits	1616 Bits	1616 Bits
Image size after pressure	23022 Bits	22840 Bits	21362 Bits

Figure (6) image size if the group is divided into 3-5 bits



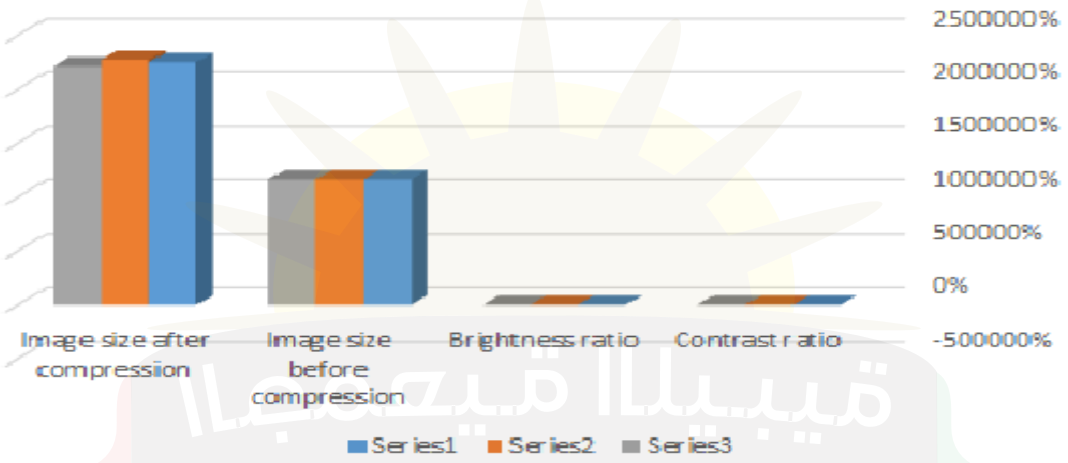
• Increase the size of the image when dividing the bits into two groups, a group consisting of 3 bits and a group consisting of 5 bits. As in the following table:

C- Increase the size of the image when dividing the bits into two groups, a group consisting of 5

Table (5-5) image size if the group is divided into 5-3 bits

Image type	RGB		
Contrast ratio	%0	%20 -	%40 -
Brightness ratio	%0	%30 –	%40 +
Image size before pressing	11616 Bits	11616 Bits	11616 Bits
Image size after pressure	22516 Bits	22754 Bits	2112 Bits

Figure (7) image size if the group is divided into 5-3 bits



6- CONCLUSIONS AND RECOMMENDATIONS

After studying some of the methods of data compression and other methods that were not addressed, we found differences in the analysis of these methods and dealing with them, and we came up with some conclusions and recommendations, including the following :

CONCLUSIONS

1. The algorithm is easy to analyze.

2. There are not many calculations, in terms of bytes division and comparison.

3. Don't take long.

4. Give better results if they have less contrast.

5. It is considered a lossless algorithm.

RECOMMENDATIONS

1. use this method to compress another images types.

2. Use this method to compress audio files and videos

3. Use other compression methods and compare them with the way they were used in the search.

4- Use this method to compress files containing still and moving images, texts written in languages other than English and files containing information where previous data meet.

7- REFERENCES

1. BasarKoc, ZiyaArnavut and HuseyinKocak, " Lossless Compression of Dithered Images", IEEE Photonics Journal , Vol. 5, No. 3, June 2013 .
2. Walaa Z. Wahba , Ashraf Y. A. Maghari, " Lossless Image Compression Techniques Comparative Study " International Research Journal of Engineering and Technology (IRJET) , Volume: 03 Issue: 02 | Feb-2016.
3. Dinesh V. , Nitesh D. , Buddhabhushanand Ravindra, "Image Compression Techniques: Lossy and Lossless" International Journal of Engineering Research and General Science Volume 3, Issue 2, March-April, 2015ISSN 2091-2730.
4. R.Navaneethakrishnan, " Study of Image Compression Techniques"International Journal of Scientific & Engineering Research, Volume 3, Issue 7, July-2012 1 ISSN 2229-5518.
5. Khobragade P. B., Thakare S. S., " Image Compression Techniques- A Review"International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 272-275.
6. Amandeep Kaur, Sonali Gupta, Lofty Sahi and Sheilly Padda, " COMPREHENSIVE STUDY OF IMAGE COMPRESSION TECHNIQUES ", JOURNAL OF CRITICAL REVIEWS , VOL 7, ISSUE 17, 2020.
7. Firas A. Jassim , Hind E. Qassim ," FIVE MODULUS METHOD FOR IMAGE COMPRESSION" Signal & Image Processing : An International Journal (SIPIJ) Vol.3, No.5, October 2012.