Kurdistan Journal of Applied Research (KJAR) | Print-ISSN: 2411-7684 – Electronic-ISSN: 2411-7706 | kjar.spu.edu.iq
Volume 2 | Issue 3 | August 2017 | DOI: 10.24017/science.2017.3.34

A Proposed Lossy Image Compression based on Multiplication Table

Mohammed Salih Mahdi
BIT Dept.
Business Information College
University of Information Technology and Communications
Baghdad, Iraq
mohammed.salih@uoitc.edu.iq

Nidaa Falih Hassan
Computer science Dept.
University of technology
Baghdad, Iraq
110020@uotechnology.edu.iq

Abstract: ***Lately, Internet improved in the various trends, especially, the use of the image increased due to the daily use in several scopes like social media (Facebook, Twitter, WhatsApp, etc.), connected devices (sensor, IP camera, Internet of Things (IoT) Internet of Everything (IoE), etc) and smart phone devices that users interchanged images estimated in the billions. So, images issues in internet can be summarized into two criteria, the first criteria is considered with transmit image size. The second criteria is considered with low bandwidth through transmission. This paper exhibits a methodology for image compression using an idea of multiplication Table. The suggested algorithm helpful in realizing a preferable achievement by presenting a high Compression Ratio, preserve image quality with a high PSNR, small losing in the original image and efficiently in running time.***

Keywords: lossy compression**,** multiplication Table, IoT

1. INTRODUCTION

Data compression is a major role in transmission and memory space, especially image compression which is minifying the images size that it requires to be sent, so that it is lead to increase the speed of transmission [1,2].

Data compression is one of interested mechanisms in computer science. It is utilized to minify the length of data. It is done by minifying the amount of bits that are needed to act for each an individual byte of data [3]. The objective of compression process is minifying the recurrence of input by producing the compressed data from the input [4]. On the other hand, the objective of decompression process is reestablishing the data value M from compressed data; M either is the same or is not the same to input but close to input [3].

Data compression is classified into two types. The 1st type refers to lossless compression method which it is potential to precisely reestablish the original input stream from the compressed data. Through the compression process in this method, there is no damage for each an individual element of data. Therefore, it is called reversible compression. This compression type method process data like exe files, text files and images kept for legal purpose. The 2nd type is referred to lossy compression method which it is not potential to reestablish the precisely original input stream from the compressed data. During the compression process in this method, there is a little loss of data, so it is called irreversible compression. This loss of data will not perceive by the human [5, 6].

Two factors can be used for testing the performance of suggested algorithm:

- Compression Ratio (CR): it refers to the size of the difference between compressed data (C) and original data (O) bit per bit as shown in Eq. (1) [5,7].

$$CR = (1 - \frac{C}{O}) * 100\% \quad (1)$$

- Distortion: it refers to the amount of lost in the compression. Its calculated after decompressing the data based on two standard, mean square error (MSE) and peak_signal to noise_ratio(PSNR) as shown in Eq.(2) and Eq.(3) respectively[5,7].

$$MSE = \frac{\sum_{i=0}^{M}\sum_{j=0}^{N}[S(i,j)-D(i,j)]^2}{M*N} \quad (2)$$

$$PSNR = 10 * \log_{10}\frac{(MAX(S))^2}{MSE} \quad (3)$$

Therefore, the character S refers to source data. The character D represents the decompressed data. The MAX (S) value is the largest possible value in the source file. Lower PSNR and higher MSE led to higher distortion. While, higher PSNR and lower MSE led to Lower distortion [8].

The suggested imag.e Lossy compression is depends on an idea of intensity quantization is presented using Multiplication Table (shortly called MT).

2. LITERATURE REVIEW

There are various papers applied to improve image compression. In [9] offered image compression using SVD transform to act any image with a set of useful features, it leads to reducing the size of the image, which is stored in the memory.

In [10] presented three composite wavelet (WT) methods according to compressed color image, those methods consist of stationary WT, composite WT and composite multi methods. These methods are combined to provide images with high compression. In[11] presented a technique for image compression according on a concept of data folding which is linked on shading with different volumes of color image, it starts by using column folding followed by row folding and this process repeated until the image size minify to potential esteem as indicated by the different levels of folding.

Regarding the previous works discussed above, the contribution of this paper is presents suggested algorithm for lossy image compression to achieve image with high compression, good image quality, solving images issues in the internet.

3. Suggested Algorithm

The main idea of proposed algorithm is to process images issues by present a lossy compression technique based on MT to accomplishment the intensity quantization, where the intensity of pixels are collected based on the number of MT. According to the concept of the many multimedia, the correlation of the nearby intensity is the rife characteristic. Figure (1) shown structure of MT algorithm compressed. Figure (2) shown structure of MT algorithm decompression.

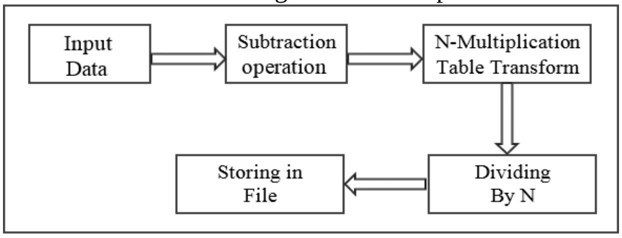

Figure 1: Structure of MT algorithm Compression

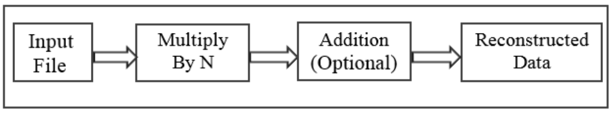

Figure 2: Structure of MT algorithm Decompression

The following steps described the suggested MT algorithm compression process:

- **Input data:** The first step in the suggested MT algorithm is to read data, which are gets from different area like PC, mobile, sensor, IoT, etc.,. Different sizes of images are used in suggested MT algorithm.
- **Subtraction operation:** Obviously, the range of values of images for each pixel between 0 to 255. So, it is represented by one byte for each pixel in grayscale images and three bytes for each pixel in color images. During the study on bytes of images, four least significant bits can be deleted from each byte, and this will not perceived by Human visual System (HSV). Thus, the maximum deleting is 4 bits. So, for each value between 0 to 15 represented by 1 bit instead of 4 bit and value 16 represented by 1 bit instead of 5 bit, etc. Table (1) shows values after removing 15 from each byte.
- **N-Multiplication Table Transform:** This step is the core of the suggested MT algorithm. The values quantization can be done by converting each value into closest double value by N. I.e. Convert each value into a nearest value divided by N. The range of N, which be, used greater than 2 and less than 13. The best of range N between 4 to 10. Table (2) shows values after this step.

Table1: Original and Update represented

Values	Original represented	Update Values	Update represented
1-15	4 bit	0	1 bit
16	5 bit	1	1bit
17-18	5 bit	2-3	2 bit
19-22	5bit	4-7	3 bit
23-30	5 bit	8-15	4 bit
31	5 bit	16	5 bit
32-46	6 bit	17-31	5 bit
47-63	6 bit	32-48	6 bit
64-78	7 bit	49-63	6 bit
79-127	7bit	64-112	7 bit
128-142	8 bit	113-127	7 bit
143-255	8 bit	128-240	8 bit

Table2: N-Multiplication

N	MT ranges of values
4	(0 4 8 12 16 20 24 28 32 36 40 …….. 240)
5	(0 5 10 15 20 25 30 35 40 45 …….. 240)
6	(0 6 12 18 24 30 36 42 48 54 …….. 240)
7	(0 7 14 21 28 35 42 49 56 63 …….. 238)
8	(0 8 16 24 32 40 48 56 64 72 …….. 240)
9	(0 9 18 27 36 45 54 63 72 81 …….. 234)
10	(0 10 20 30 40 50 60 70 80 90 …….. 240)

- **Dividing by N:** Dividing each value by N so values changed from values in Table (2) to values in Table (3) which can be shown how changed.

Table3: Dividing by N

N	MT ranges of values
4	(0 1 2 3 4 5 …….. 60)
5	(0 1 2 3 4 5 …….. 48)
6	(0 1 2 3 4 5 …….. 40)
7	(0 1 2 3 4 5 ……..,34)
8	(0 1 2 3 4 5 ……..,30)
9	(0 1 2 3 4 5 ……..,26)
10	(0 1 2 3 4 5 ………22)

- **Storing values file**: the final step, two methods are proposed to be accomplished. The first method (run length) by storing a value and concatenation with symbol then ˥ occurrences and so on. While, the second method requires kernel (L*L), for each kernel, finding the smallest value and subtracted this value from each value in the selected kernel. Storing small value and concatenation with symbol ˥ then next value in kernel and so on.

The ˥ character represent in binary 2 bit (10) instead of space which represented by 6 bit (100000) in and this symbol used to split the value from occurrences of it in first method. In addition, the same symbol used to split the minimum value and normalized in each kernel of second method.

The following steps described the suggested MT algorithm decompression Process**:**

- File: The first step in the decompression process is to read data from file then convert it to appropriate form based on the same principle which be used in compression process but by the reverse way.
- Multiply by N: Multiply each value by N, so values are changed from values in Table (3) to values in Table (2).
- Adding operation: this step is optional, adding K to each value in previous step, where K is (1, 3,7,15). If K does not added then there is no effects on the HSV. However, the Adding operation is used to preserve image quality with a high PSNR
- Reconstructed Data: The values from previous step can be represented the decompressed data based on either run length method or kernel(L*L). The extracted data are similar to original data but not the same due to the suggested MT algorithm is lossy compression. Algorithm (1), (2) shown the basic operation of suggested MT algorithm compression and decompression.

Algorithm (1): MT Algorithm Compression

Input: Data (image), N(number of multiplication table), **K=(1;3;7;15)**

Output: Storing File

Begin

Step$_1$: Read data as vector of bytes.

Step$_2$: Subtract **K** from each element in vector of bytes.

Step$_3$: Quantization each byte in vector by converted into double value by **N**.

Step$_4$: Dividing bytes in vector by **N.**

Step$_5$: Storing the vector of bytes in file.

End

Algorithm (2): MT Algorithm Decompression

Input: Storing File , N(number of multiplication table)), **K=(1;3;7;15)**

Output: Reconstructed Data (image)

Begin

Step$_1$: Read data from file and convert it to vector of bytes.

Step$_2$: Multiply bytes in vector by **N.**

Step$_3$: Adding **K** to each byte in vector (optional step).

Step$_4$: Pset vector of Reconstructed Date bytes as image.

End

4. RESULTS

A proposed algorithm is implemented using C# 2015. The performance factor of the proposed MT algorithm is extracted based on Compression Ratio (CR) and Peak to Signal Ratio (PSNR). A proposed algorithm is applied on standard images like: color image (Lena and Penguins) and grayscale image (Einstein and Elaine) with different sizes (256*256 and 512*512) based on MT with run length method and kernel (L*L), as shown in the following Tables.

Table 4: CR for test images using run length method

Image / 256 size	MT(5)	MT(7)	MT(9)
Elaine	74%	81%	94%
Penguins	73%	79%	83%

Table 5: PSNR for test images using run length method

Image / 256 size	MT(5)	MT(7)	MT(9)
Elaine	24.45	24.26	24.01
Penguins	38.91	36.008	33.72

Table 6: CR for test images using 4*4 kernel

Image / size	MT(5)	MT(7)	MT(9)
Lena 512	75%	78%	80%
Penguins256	73%	76%	78%
Einstein 256	64%	69%	72%
Elaine512	73%	77%	79%

Table 7: C.R for test images using 8*8 kernel

Image / size	MT(5)	MT(7)	MT(9)
Lena 512	72%	76%	79%
Penguins256	69%	72%	74%
Einstein 256	59%	63%	67%
Elaine512	69%	73%	76%

Table 8: C.R for test images using 16*16 kernel

Image / size	MT(5)	MT(7)	MT(9)
Lena 512	66%	70%	74%
Penguins256	60%	64%	67%
Einstein 256	50%	56%	65%
Elaine512	61%	66%	70%

Table 9: PSNR for test images using 4*4 kernel

Image / size	MT(5)	MT(7)	MT(9)
Lena 512	45.13	42.13	39.90
Penguins256	45.30	42.33	40.13
Einstein 256	45.20	42.09	39.93
Elaine 512	45.16	42.09	39.97

Table 10: PSNR for test images using 8*8 kernel

Image / size	MT(5)	MT(7)	MT(9)
Lena 512	45.13	42.13	39.90
Penguins256	45.30	42.33	40.13
Einstein 256	45.20	42.09	39.93
Elaine512	45.16	42.09	39.97

Table 11: PSNR for test images using 16*16 kernel

Image / size	MT(5)	MT(7)	MT(9)
Lena 512	45.13	42.13	39.90
Penguins256	45.30	42.33	40.13
Einstein 256	45.20	42.09	39.93
Elaine512	45.16	42.09	39.97

5. CONCLUSIONS

This paper presents a lossy compression technique based on MT to achieve the intensity quantization and solving images issues in the internet.

From the conclude results of applying the suggested algorithm, it could be used to compress images into various scopes like PC, mobile, sensor, IoT, etc. Different size of images has been used in suggested MT algorithm, with two types of images (grayscale and color images).

The suggested algorithm is helpful in realizing a preferable achievement presented by high CR; furthermore, it is preserving image quality with a high PSNR, small losing in the original image.

6. DISCUSSION

In this section some important issues have to be confirmed, and they are: Table (4) expressed Compression ratio (CR) for two images (Elaine and Penguins) with equal size (256*256), and Table (5) expressed PSNR for these images by using the first method (run length), which is suggested for storing data based on (MT (5), MT (7) and MT (9)).

The high CR is 94% in MT (9) as showing in Table (4) but it has higher distortion due to the value of PSNR is 24.01 as shown in Table (5). The low CR is 73% in MT (5) but it has lower distortion due to the value of PSNR is 38.91. The run length produced a high C.R with low PSNR either large size image or small size.

From Table (6) to Table (11) explained CR and PSNR for the second method (L*L kernel) which is suggested for storing data. 4*4, 8*8 and 16*16 kernels are tested for four images and provided lower distortion. So, it has the ability to preserve image quality with a high PSNR (45.13, 45.3, 45.2 and 45.16) in MT (5) for the three chosen kernel. The (L*L kernel) produced a high CR either large size image or small size.

Table (12), illustrates the Comparison of the suggested algorithm with literature review and finally figure (3) illustrated the simple images of standard images which is used in this research.

Table 12: Comparison with related work system

Paper	Methodology	PSNR	CR %
P. Dhumal, S. Deshmukh [9]	SVD	High	45%
Z. Abood [10]	Three composite wavelet (WT)	Good	60%
N. Asia , R. Heba[11]	Folding Technique	Not tested	20%
Suggested method (MT)	Multiplication Table and set of techniques	High in L*L kernel, Good in run length	73% As average in two method

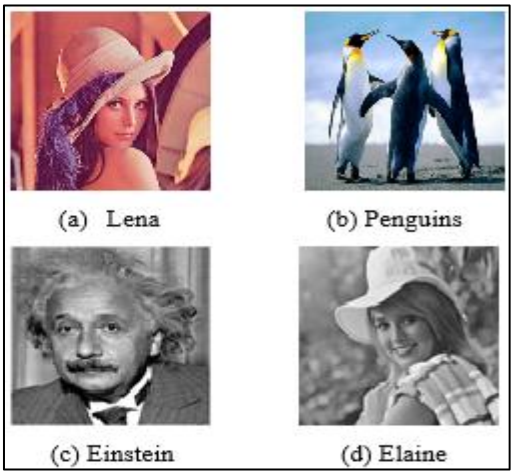

Figure 3: Standard Images

7. REFERENCE

[1] S. Kaur, V. Sulochana, "Design and Implementation of LZW Data Compression Algorithm", International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.4, July 2012.

[2] K. Sayood," Introduction to Data Compression", Third Edition, University of Nebraska, Lincoln, Morgan Kaufmann is an imprint of Elsevier,2000.

[3] M. Nelson, L. Gailly, "The data compression" 2nd Edition. M&T Books New York. 1996.

[4] M. Pu, "Fundamental Data Compression",1st Edition, Butterworth-Heinemann. 2005.

[5] M. Pu, "Fundamental Data Compression", Butterworth-Heinemann is an imprint of Elsevier Linacre House, Jordan Hill, Oxford OX2, Corporate Drive, 2006.

[6] D. Salomon, M. Giovanni," Handbook of Data Compression", Fifth Edition, Springer London Dordrecht Heidelberg New York, British Library Cataloguing in Publication Data Springer-Verlag London, 2010.

[7] J. Santoso, L. Nugroho, G. Suparta, R. Hidayat, " Compression Ratio and Peak Signal to Noise Ratio in Grayscale Image Compression using Wavelet", International Journal of Computer Science and Technology 2 (2): 7–11, 2011.

[8] O. Khalifa, "Wavelet Coding Design for Image Data Compression", the international Arab Journal of Information Technology 2 (2): 118–127, 2005.

[9] P. Dhumal, S. Deshmukh, "Survey on Comparative Analysis of Various Image Compression Algorithms with Singular Value Decomposition", International Journal of Computer Applications, Vol. 133, No. 6, PP. 18-21, 2016.

[10] Z. Abood, "Composite Techniques Based Color Image Compression", Journal of Engineering Number 3 Volume 23 March, Baghdad, 2017

[11] N. Asia , R. Heba, "Lossless Color Image Compression Based on Folding Technique", The Forth Scientific Conference of the College of Science University of Kerbala, Journal of University of Kerbala, 2016.

Mohammed Salih Mahdi is currently a PhD candidate under the supervision of Assist.Prof.DR. Nidaa Falih.Hassan in Computer Science Dept, at University of Technology, Iraq, Baghdad. His BSc degree in hiding data in 2010 from University of Technology and his MSc degree in a security of cloud computing in 2012 is from University of Technology. Currently. Lecturer in Business Information College, University of Information Technology and Communications. His research interests include data mining, Arterial intelligent, Computer Security, image processing, data compression, Healthcare, mobile application, cloud computing, internet of things, internet of everything.

Assist. Prof .Dr. Nidaa F. Hassan received theMSc. and PhD. in Computer Science from University of Technology, Iraq, 1996 and 2005 respectively. She has around 21 years of teaching experience. Her areas of interest's computer security and image processing.