

# New Modified RLE Algorithms to Compress Grayscale Images with Lossy and Lossless Compression

Hassan K. Albahadily<sup>1,2</sup>

<sup>1</sup>Dept. of Telecommunication and Network Devices  
BSUIR

Minsk, Belarus

<sup>2</sup>University of Mustansiriyah  
Baghdad, Iraq

Viktar U. Tsviatkou

Dept. of Telecommunication and  
Network Devices

BSUIR

Minsk,

Belarus

Alaa A. Jabbar Altaay

Dept. of Computer Sciences

University of Mustansiriyah

Baghdad, Iraq

Valery K. Kanapelka

Dept. of Telecommunication and Network Devices

BSUIR

Minsk, Belarus

**Abstract**—New modified RLE algorithms to compress grayscale images with lossy and lossless compression, depending on the probability of repetition of pixels in the image and the pixel values to reduce the size of the encoded data by sending bit 1 instead of the original value of the pixel if the pixel's value is repeated. The proposed algorithms achieved good reduction of encoded size as compared with other compression method that used to compare with our method and decrease encoding time by good ratio.

**Keywords**—compression; Run Length Encoding; quantization

## I. INTRODUCTION

Data files frequently contain the same character repeated many times in a row. Digitized signals can also have runs of the same value, indicating that the signal is not changing, also images and music. Run-length encoding is a simple method of compressing these types of files [1]. The basic idea of RLE is to remap a sequence of numbers into a sequence of pairs (Value, Run), where value represents the data in the input sequence and run represents the number of times that data is contiguously repeated. An example illustrating RLE for a binary sequence is shown in Fig.1 below

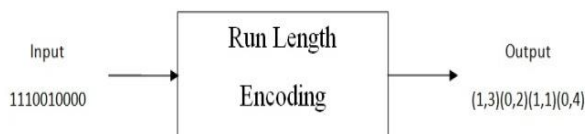


Fig. 1. Illustration of RLE for a binary input sequence

The goal of lossless image compression is to represent an image signal with the smallest possible number of bits without loss of any information, thereby speeding up transmission and minimizing storage requirements. While The goal of lossy compression is to achieve the best possible fidelity given an

available communication or storage bit rate capacity or to minimize the number of bits representing the image signal subject to some allowable loss of information. In this way, a much greater reduction in bit rate can be attained as compared to lossless compression, which is necessary for enabling many realtime applications involving the handling and transmission of audiovisual information. In general, there is significant redundancy present in image signals. This redundancy is proportional to the amount of correlation among the image data samples [2].

The lossy compression of images is currently widely used algorithms like JPEG (Joint Photographic Experts Group) [3] and JPEG 2000 [4], which are based on entropy coding (Huffman and arithmetic), coefficients of discrete cosine and wavelet transforms. These algorithms are used to compress the image in hundreds of times with acceptable quality, but have a high computational complexity.

The lossless image compression algorithms are effective for encoding pixel values like LZW (Lempel Ziv Welch), GIF (Graphics Interchange Format), TIFF (Tagged Image File Format) [5,6], Deflate and LZMA (Lempel Ziv Markov chain algorithm) used in archive Zip, 7-zip [7], PPM (portable pixmap) and LZSS (Lempel Ziv Storer Szymanski) used in the archive Rar [7]. These algorithms are used to lossless compress grayscale images to about 2 times depending on the brightness distribution of pixels, but these techniques have a high computational complexity, and because the time and computational resources are limited we should use simpler algorithms for efficient coding like Run Length Encoding algorithm RLE[8], which is based on the character repeats. It can be used for lossless image compression with a small number of sharp luminance differences (animation, medical, segmented and quantized). In combination with pre-quantization image, RLE algorithm may be used for lossy compression. The disadvantage of this algorithm is the lack of

consideration of the probability of repetition of pixel values images.

The aim of this paper is to develop a lossless and lossy new compression algorithm based on the RLE method by using probability of pixel value repetition.

Our technique is implemented using MATLAB2012 on WINDOWS7 Operating System.

This study is organized as follows: Section I presented an introduction about the compression and RLE algorithm, Section II describes the Related work and some of recent modifications of the RLE compression method, Section III explain the suggested RLE algorithm and its modifications, Section IV explains the quantization which is necessary to make good compression ratio when using lossy compression, section V presented the experiment of suggested new algorithm and discuss the results and compare it with all others results, And Section VI presents conclusions of our work and suggestions for future studies.

## II. RELATED WORK

There are some recent modifications of the RLE compression method, some of these modifications focusing on the way of scanning pixels (row, column, hilbert, ...) [9], or on the bit depth of the runs of the repeated pixels and try to decrease the length of the bit reserved to represent the runs [10], or using a modified Entropy Coding to enhance RLE [11], or quantize image to decrease the number of values of the pixels and make them similar then using DCT to achieve a high compression ratio [11,12], or quantizing adjacent pixels which have small difference in the value of pixel [13], or negligee the non-repeated value [14]. All of these modifications are good to enhance RLE and achieving better results.

## III. RLE ALGORITHM AND ITS MODIFICATIONS

### A. RLE Algorithm

RLE algorithm has been chosen because it is mathematically simple and not complex so that we can achieve a high compression speed for the compression process.

The RLE algorithm based on counting the number of repetitions of the values of successive symbols and it is can be represent by the following diagram of structure coded data in Fig. 2 and the block diagram in Fig. 3

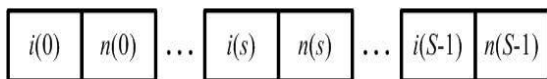


Fig. 2. Coded data structure

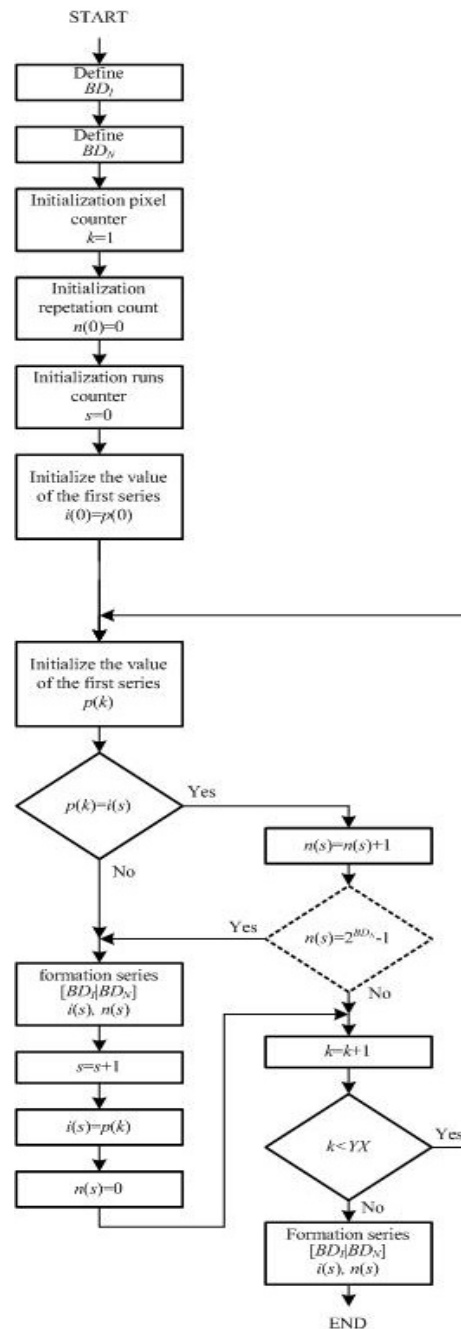


Fig. 3. RLE Block diagram

Modular implementation of RLE algorithm assumes that we have the accumulation of the encoded data for the analysis and selection of encoding parameters. The analysis is based on the table I/N length episodes in which one or more of the same value and consecutive symbols I put in correspondence number N as shown in table 1 below.

TABLE I. LENGTH SERIES

I	$i(0)$	$i(1)$	...	$i(s)$	...
N	$n(0)$	$n(1)$	...	$n(s)$	...

On the basis of run-length table determined bit depth  $BD_I$  and run length values  $BD_N$  using expressions:

$$BD_I = \left\lceil \log_2 \left( \max(i(s))_{(s=0, S-1)} \right) \right\rceil \quad (1)$$

$$BD_N = \left\lceil \log_2 \left( \max(n(s))_{(s=0, S-1)} \right) \right\rceil \quad (2)$$

Where  $i(s)$  is encoded value with the first character of the run length of the table;  $n(s)$  is the number of repetitions for  $i$ -th symbol (length of the series); S number of encoded symbols (the number of rows in Table 1).

Modular implementation of RLE algorithm provides a minimal amount of code. Its disadvantage is the delay in the coding required for the accumulation of data and the construction of the table of run lengths.

When the production implementation of the algorithm RLE table run length cannot be built, and count the number of characters it can be carried out as they become available. This can significantly improve the coding rate. The structure of the algorithm and the encoded data in the production implementation of RLE algorithm are the same as in the block. Values of the bit depth  $BD_I$  an image bit-depth and the lengths of series of values  $BD_N$  are selected independently from the incoming data. They can be selected with an excess or deficiency (the overflow  $n(s)$  formed a new series  $\{i(s), n(s)\}$ ). In some cases this can lead to an increase in the amount of code which is a disadvantage of RLE algorithm implementation. To improve the characteristics of the RLE algorithm, the production implementation can be through the formation of a dynamic table of run lengths and periodic updating of coding parameters.

When encoding image size  $R_{I/N}$  (bit) code, CR compression ratio and computational complexity  $C_{I/N}$  of RLE algorithm is determined using the following expressions:

$$R_{I/N} = S(BD_I + BD_N) \quad (3)$$

$$CR = 8YX/R_{I/N} \quad (4)$$

$$C_{I/N} = YX + 4S \quad (5)$$

where YX image size, determines the number of operations on the buffer and the formation of the table I/N size S records; 4S additional operations to find a maximum value for I (S operations), search for maximum values N (S operations), encoding and transmitting (2S operations).

### B. RLE Algorithm I2BN

Suggested modification of RLE is I2BN algorithm to compress grayscale images based on the probability of repetition pixel values in rows. In the process of the algorithm builds a table N/P probability of repeats (table. 2), wherein each value of n the run length number placed such series  $p_n(n)$ .

TABLE II. PROBABILITY OF REPEATS

N	1	2	...	n	...
P	$p_n(1)$	$p_n(2)$	...	$p_n(n)$	...

For images characterized by a gradual decrease of the function  $p_n(n)$  with increasing values n.

When encoding the length of Series the algorithm I2BN first formed character I. Then, if the character I is repeated, the formed bit  $b1(s) = 1$ , otherwise  $b1(s) = 0$  (Repeat the first character). If the symbol I is repeated again, the bit generated  $b2(s) = 1$ , otherwise  $b2(s) = 0$  (second repeat symbol). If the symbol I is repeated again, the character is formed  $n(s)$ , taking into account the number of repetitions (originally  $n(s) = 0$ , if symbol I It is repeated again,  $n(s) > 0$ ). As a result, the series may be formed as:

$$\{i(s), b1(s) = 0\}, \quad \{i(s), b1(s) = 1, b2(s) = 0\}, \quad \{i(s), b1(s) = 1, b2(s) = 1, n(s)\}.$$

The structure of RLE coded data can be represented by the following block diagram

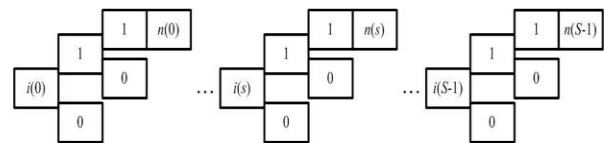


Fig. 4. Coded data structure of algorithm I2BN

The size  $R_{I/2B/N}$  (Bit) code for the algorithm I2BN defined by the expression:

$$R_{I/2B/N} = S(BD_I + 1) + \sum_{s=0}^{S-1} b1(s) + BD_N \sum_{s=0}^{S-1} b2(s) \quad (6)$$

To determine the computational complexity of the expression

$$C_{I/2B/N} = YX + 4S + \sum_{s=0}^{S-1} b1(s) + \sum_{s=0}^{S-1} b2(s) \quad (7)$$

Equation (7) takes into account YX buffering operations and formation table I/N- run length size S records, S

Operations for searching for a maximum value I, S Operations for searching for a maximum value N,  $2S + \sum_{s=0}^{S-1} b1(s) + \sum_{s=0}^{S-1} b2(s)$  operations for encoding and transmission.

C. RLE Algorithm I3BN

Another suggested modification of RLE is I3BN, which is differs from the algorithm I2BN by using an additional symbol  $b3(s)$ , which takes the value 1, if symbol I repeated for the third time in a row, and 0 if absent. The structure of coded data according to the algorithm I3BN is shown in the Fig.5 below

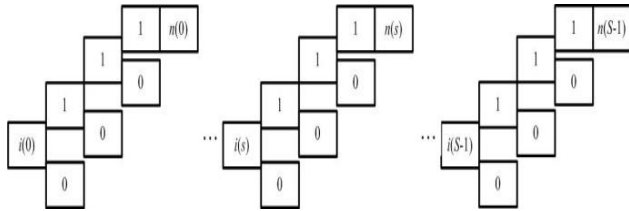


Fig. 5. Coded data structure of algorithm I3BN

Size of  $R_{I/3B/N}$  (bit) code algorithm I3BN defined by the expression

$$R_{I/3B/N} = S(BD_1 + 1) + \sum_{s=0}^{S-1} b1(s) + \sum_{s=0}^{S-1} b2(s) + BD_N \sum_{s=0}^{S-1} b3(s) \quad (8)$$

The computational complexity of the algorithm I3BN estimated using the expression

$$C_{I/3B/N} = YX + 4S + \sum_{s=0}^{S-1} b1(s) + \sum_{s=0}^{S-1} b2(s) + \sum_{s=0}^{S-1} b3(s) \quad (9)$$

Equation (9) accounts for buffering YX operations and forming a table I/N run length size records S, S maximum search operations on the values I, S maximum search operations on values N,  $2S + \sum_{s=0}^{S-1} b1(s) + \sum_{s=0}^{S-1} b2(s) + \sum_{s=0}^{S-1} b3(s)$  operations for encoding and transmission.

D. Other modifications of RLE algorithm

In addition to I2BN, I3BN we developed algorithms characterized by the use of different numbers of additional characters to encode pixels repeat, all of these algorithms shown in table3 below

TABLE III. THE RLE MODIFIED ALGORITHMS

No	Algorithm	The peculiarity of the encoded data structure
1	I/S/N	variable size field run length
2	I/B/N	additional repeat symbol
3	I/B/S/N	additional character repeat and variable length fields run length
4	I/2B/N	Repeat two additional symbol
5	I/2B/S/N	Repeat two additional characters and a variable size field run length
6	I/3B/N	three additional characters repeat
7	I/3B/S/N	Repeat three additional characters and a variable size field run length
8	2I/N	encoded characters repeat
9	2I/S/N	repeat the encoded symbols and variable length fields run length
10	2I/B/N	Repeat encoded character and an additional

		character repeat
11	2I/B/S/N	repeat the encoded symbols, additional character repeat and variable length fields run length
12	2I/2B/N	encoded symbol is repeated and two additional symbol repetition
13	2I/2B/S/N	repeat the encoded symbols, two additional character repeat and variable-length field size series
14	2I/B/2N(L/R)	repeat the encoded symbols, additional characters, and repeat two-segment length field series

IV. IMAGE QUANTIZATION FOR LOSSY COMPRESSION IN THE SPATIAL DOMAIN

Methods of image lossy compression based on an efficient coding of transform coefficients with their pre-quantization. Quantization makes many lossy techniques determines the mainly compression ratio. Coding of transform coefficients can achieve the greatest compression ratios due to the concentration of the primary energy in a relatively small number of significant transform coefficients. However, the transformation itself requires substantial computing resources and time. Therefore, an urgent task is to develop an algorithm for using the quantization of the pixel values of the image and their subsequent efficient coding.

Determined quantized pixel neighborhood (left or one of the top three), the closest in value to the central pixel. We calculate modulus of the difference of the pixel values of all the values of neighboring pixels in the neighborhood. If these differences less than the threshold  $\Delta_s$ , the central pixel in the neighborhood is set to the value of the quantized value of the pixel in the vicinity. If this condition is not met, then the central pixel keeps its value.

V. EXPERIMENTAL RESULTS

The test images shown in Fig. 6 below -which includes different grayscale images- has been used to test our algorithms



Fig. 6. Test images used in the experiment

A. Evaluating loseless Algorithms

We got the results shown in Table 4 below, which shows the code sizes obtained for RLE, Huffman, Zip, Rar and all modified algorithms.

TABLE IV. CODE SIZE (BYTE) FOR LOSSLESS COMPRESSION

Algorithm	Code size (byte) image		
	M1	M2	M3
RLE	18620	77896	242604
I/S/N	18739	72412	143162
I/B/N	21504	91397	303235
I/B/S/N	20957	81407	165901
I/2B/N	14694	57583	106871
I/2B/S/N	17689	68649	118177
I/3B/N	14577	56701	103512
I/3B/S/N	17687	68581	117997
2I/N	30714	120377	208857
2I/S/N	34484	135364	261921
2I/B/N	32599	127870	222123
2I/B/S/N	32826	128820	225640
2I/2B/N	32713	128345	223002
2I/2B/S/N	32825	128753	225460
2I/B/2N(L/R)	30046	118028	208692
Zip	13070	44603	76470
Rar	12505	43062	75887
Huffman	14361	58281	113091

The Fig 7 below shows these results

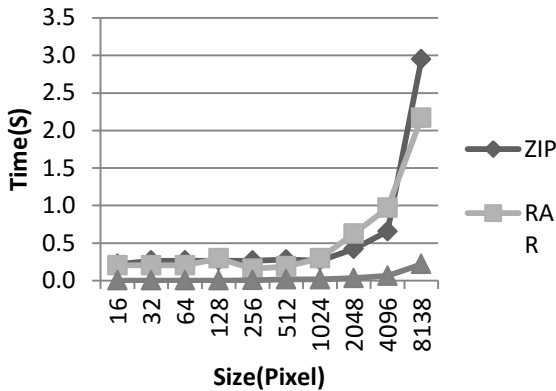


Fig. 7. Dependence of encoding time with image size for Zip, Rar and proposed RLE algorithms

From Table 4 we can see that the minimum amount of code provided by algorithms Rar, Zip and modified algorithm I/3B/N which shows about 26% of the worst result, but exceeds 57% and 8%, for RLE and Huffman respectively. Thus algorithms RLE, I/2B/N, I/3B/N provide comparison algorithms and Rar Zip decrease encoding time 31.2 and 53 times respectively, with an image size of 256×256 pixels, 16 and 18.5, 6 times, respectively, with image size 1024×1024 pixels, in 10 and 13.5 times respectively, when the image size 8192×8192 pixels.

B. Evaluating lossy Algorithms

Fig.6 shows the test images which used in the experiment, for which the coding efficiency analysis for lossy compression performed by RLE algorithm and its modifications with respect to the Zip, Rar, JPEG and JPEG2000

TABLE V. COMPRESSION TIME(SECOND) FOR LOSSY COMPRESSION

Algorithm	image compression time (Second)		
	M3	Lena	cameraman
RLE	0.00000085	0.00000241	0.00000085
I2/B/N	0.00000028	0.00000144	0.00000028
I3/B/N	0.00000142	0.000000962	0.00000028
RAR	0.249	0.219	0.300
ZIP	0.324	0.322	0.682
JPEG	0.073	0.200	0.116
JPEG 2000	0.102	0.182	0.190

For lossy compression based on RLE algorithm and its modifications I2BN and I3BN, Zip algorithms, Rar and Huffman algorithm used two-threshold progressive image quantization, we have performed a tests on the images above and we found that the probabilistic proposed RLE algorithms provides compression ratio up to 1.1-1.6 times, and 1.2-1.5 times as compared with Rar and Zip respectively. Also they are provides a ratio up to 1.4-2 times and 1.2-2.3 times according to RLE and Huffman respectively. The proposed algorithms provide a reduction in the mean square error MSE 3-10 times in comparison with JPEG 2000, and up to 1.2-18 times when the same proposed algorithm compared with JPEG.

Table 5 showing the compression time of lossy compression for the test images using different algorithms. From Table 5 the probabilistic proposed RLE algorithms provide a reduction in the compression time up to 1.7-6 times as compared with RLE, 170-1750 times as compared with the RAR, 230-2200 times as compared with ZIP, 50-400 times as compared with JPEG, 70-670 times as compared with the JPEG 2000.

VI. CONCLUSION AND FUTURE WORK

Results showed that these algorithms provide better performance in encoded image size compared with RLE and Huffman algorithms. The proposed lossless compression algorithms provide encoded image size reduction by 57% and 8% compared with RLE and Huffman algorithms, respectively; decrease encoding time 10-31 times and 13-53 times when changing the image size from 256×256 pixel to 8192×8192 pixel compared with Rar and Zip algorithms respectively. In lossy compression, the proposed algorithms provide improved image compression ratio up to 2 times in comparison with the algorithm RLE, and 2.3 times compared with the Huffman algorithm, reducing the mean square error (MSE) up to 10 times compared with the JPEG2000 compression algorithm when factor 2-5 time. also it is found that the proposed algorithms provide encoding time decreased up to 6 times compared with the RLE, 1750 times in comparison with RAR, 2200 times in comparison with ZIP, 400 times as compared with JPEG, and up to 670 times as compared with JPEG 2000.

For the future work it will be good to use the proposed algorithm with other modified algorithms especially to compress color images.

REFERENCES

- [1] S. Smith, The Scientist and Engineer's Guide to Digital Signal Processing, 2nd Edition, California Technical Publishing 1999.
- [2] A. Bovik, The Essential Guide to image Processing, Elsevier Inc. USA 2009.

- [3] W. Pennebaker, J. Mitchell, JPEG Still Image Compression Standard, New York, Van Nostrand Reinhold, 1993.
- [4] T. Ebrahimi, JPEG2000 still image coding versus other standards, Proc. of the SPIE, San Diego, CA, USA, Vol. 4115 July 2000.
- [5] J. Miano, The formats and image compression algorithms in action, Publishing. Triumph, 2003.
- [6] D. Salomon, Data compression: The Complete Reference, Springer-Verlag London Limited 2007.
- [7] D. Vatolin, Data compression methods. Archiver device, image compression and video, Moscow Dialog-MIFI, 2003.
- [8] S.W. Golomb, Run-Length Encoding, IEEE Transactions on Information Theory. July 1966.
- [9] B. Karthikeyan, A Performance Analysis of Different Scanning Paths on Lossless Image Compression for Radiographic Welding Images, India, Journal of Scientific & Industrial Research Vol.73, April 2014.
- [10] I. Made, Agus Dwi Suarjaya ,A New Algorithm for Data Compression Optimization, Udayana University Bali, Indonesia, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No.8, 2012
- [11] R. Mala, M.Phil, S. Sarathadevi, A Lossless Image Compression using Modified Entropy Coding, International Journal on Recent and Innovation Trends in Computing and Communication, India, Vol. 2, Issue 8.
- [12] S. Akhter and M. A. Haque, ECG COMPRESSSION USING RUN LENGTH ENCODING, Bangladesh University of Engineering and Technology, 18th European Signal Processing Conference Denmark, 2010.
- [13] S. Joseph, A Novel Approach of Modified Run Length Encoding Scheme for High Speed Data Communication Application, International Journal of Science and Research (IJSR) Vol. 2 Issue 12, Dec 2013.
- [14] M. VidyaSagar, J.S. Rose Victor, Modified Run Length Encoding Scheme for High Data Compression Rate, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Vol.2, Issue 12, Dec. 2013.