www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 3 March 2017, Page No. 20716-20720
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i3.57

An Improved Run Length Encoding Scheme for Image Compression

Ch.Samson[1], VUK Sastry[2]
[1]Dept. of Information Technology, MVSR Engineering College,
Hyderabad, India,
samchepuri@gmail.com
[2]Dept. of Computer Science & Engineering., SNIST,
Hyderabad, India,
vuksastry@rediffmail.com

Abstract: ***In this paper, we propose an Improved Run-Length Encoding (IRLE) scheme for image compression. Conventional Run-Length Encoding is a widely used scheme as it is simple and effective but not suitable to represent non-block images. The IRLE which we have developed, combines runs and run values and it reduces the space required to store the image significantly.* The compression ratio depends on the size and type of the input message.** ***From the results obtained in this analysis, we conclude that the proposed approach is an efficient method for image compression and it provides high compression ratio and better compression percentage.***

. **Keywords:** Image compression, Run-Length Encoding, IRLE, Compression ratio**,** Compression percentage

1. Introduction

With the advent of the Internet, the world has become a global village so that anybody can share any kind of data with anyone within few seconds. Unless compressed, data cannot be transmitted efficiently. Data Compression [1] has attracted the attention of many researchers in the recent years as large volumes of data are being transmitted through the Internet daily. Compression has wide range of applications ranging from seismic exploration, remote sensing, multimedia communication to Geographical Information System. So, many methods [2-6] have been proposed for data compression. The main purpose of compression is to reduce the storage space required to represent a given quantity of information and to increase the speed of transmission.

Compressing data or information before sending it over the Internet can reduce the amount of time it takes by a considerable margin. Compression could be either lossy or lossless. Lossless compression reduces bits by identifying and eliminating statistical redundancy. No information is lost in lossless compression. Lossy compression reduces bits by identifying unnecessary information and removing it.

Run Length Encoding (RLE) is a lossless compression technique in which a given quantity of information is represented by successive runs of the same value in the information as the value followed by the run count, rather than the original run values(runs). This is useful on data that contains many such runs, for example, relatively simple graphic images such as icons, line drawings, and animations. It is not useful with files that don't have many runs as it could potentially increase the size of files.

In our example, let us take a screen containing plain black text on a solid white background. There will be too much long runs of white pixels in the blank space, and many short runs of black pixels within the text. Let us take a hypothetical single scan line which is given below.

ZZZZZZZZZZZZEZZZZZZZZZZZZEEEZZZZZZZZZZZZ
ZZZZZZZZZZZZEZZZZZZZZZZZZZZ

If we apply the run-length encoding data compression algorithm to the above hypothetical scan line, we get the following:

12Z1E12Z3E24Z1E14Z

Interpret this as twelve Z's, one E, twelve Z's, three E's, twenty four Z's, one E, fourteen Z's. The run-length code depicts the original 67 characters in only 18.

In the present paper we have proposed a scheme called Improved Run Length Encoding (IRLE) for compression, which concatenates run values and run count. The rest of the paper is organized as follows. In Section 2, the related work is mentioned. Section 3 describes the proposed scheme. Section 4 deals with the implementation results obtained in our analysis. Finally in Section 5, we draw conclusions from the results obtained in our approach and mention future scope for the possible enhancement.

2. Related Work

Among various compression methods, run-length encoding (RLE) is a simple and effective compression scheme [7-9] in which runs of data are stored as a single data value and count. This characteristic makes it become widely used [11,13,14,15,17] and even become a component of other schemes, such as the JPEG standard [10], proposed in 1980's and in use for transmitting and storing images, uses discrete cosine transformation, quantization, run-length coding, and Huffman entropy coding [2]. The idea of run-length encoding has also been extensively exploited by researchers for other kinds of data compression. These approaches attempt to compress data as either 1D streams, 2D images or 3D volumes. For example, Žalik and Lukač [16] introduce a new efficient chain code compression algorithm based on move-to-front transform and adaptive run-length encoding for compressing the repetitions of symbols' combinations. Qian and Chen [8] present an adaptive 2D run-length encoding(2DRLE) method to minimize the total number of run-lengths automatically, which has been proved to be effective through experiment. Shen and Spann [11] discuss a volumetric representation of 3D

regions.
Many researchers tried to reduce the amount of space required to store the data, for faster communication and better security of secret data. Andrew B. Watson introduced Image Compression Using the Discrete Cosine Transform in 1994[8]. A. Alfalou C. Brosseau et al. [9] performed compression based on the discrete cosine transform (DCT). Two levels of encryption are used. The first one is due to the grouping of the DCTs in the spectral domain and after a second transformation, i.e. to hide the target images; one of the input images is used as encryption key. The compression is better than JPEG in terms of PSNR. Maher Jridi, Ayman Alfalou [10] presented a method that utilizes the DCT properties to achieve the compression and the encryption simultaneously. First for compression, 8-point DCT applied to several images. Second, only some special points of DCT outputs are multiplexed. For the encryption process, a random number is generated and added to some specific DCT coefficients.

3. Proposed Scheme

When network bandwidth and storage space are limited, the image to be transmitted has to be compressed. The input image is compressed by Improved Run Length Encoding (IRLE). IRLE works the best when applied to image data where there are successive runs of the same values. Algorithm for the proposed scheme is given below.

Algorithm for IRLE Compression:

1. Read the input image and convert the image matrix into an array M.
2. Find the difference between adjacent elements of M and store it in P .
3. Convert P to logical format. The elements without repetition are denoted with one and the repeated elements with zero.
4. Find the position of the elements of P that has the value one in the step 3.
5 Find the unique element values using the positions obtained from step 4 and store it in an array.
6. Obtain the occurrence of the first element alone from the matrix in the step 4. For the remaining elements, find the difference of the matrix in the step 4.
7. Obtain the elements without repetition and their occurrences in step 6.
8. Concatenate Run value with Run count and store the result in an array C.
9. Find the remainder using, R = C mod 256.

The R is the sent to the destination. It may be noted here that by reversing the above steps, we get back the original image. The IRLE compression and decompression can be understood easily as shown below in Figure 1.

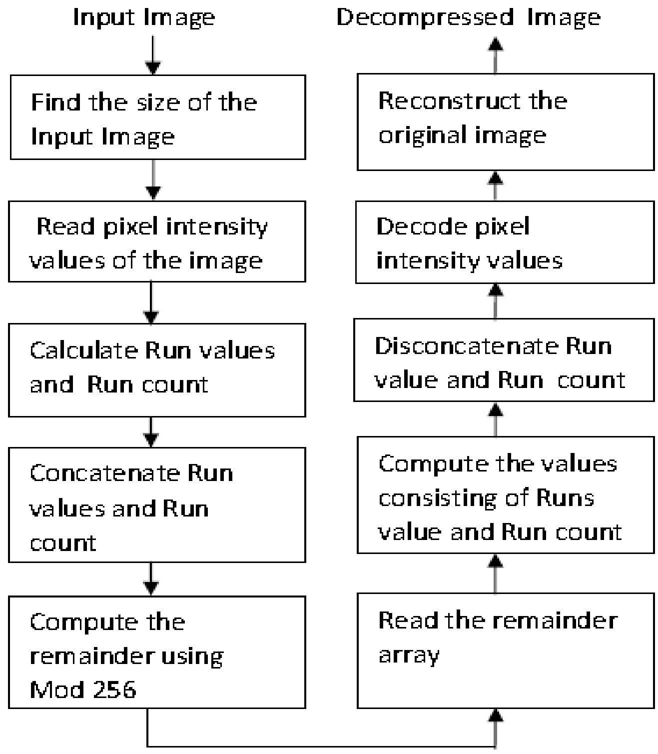

Figure 1. Schematic diagram of IRLE Compression and decompression

4. Experimental Results

Let us take an example of portion of an image, consisting of 15 picture elements (pixels) with intensity values are given by an array A,

A= [5 2 2 2 3 3 3 3 4 4 1 1 1 1 1 1 1 6 6 4 4].

In the given example, 5 has occurred 1 time, 2 has occurred 3 times, 3 has occurred 4 times, 4 has occurred 2 times, 1 has occurred 7 times, 6 has occurred 2 times, and 4 has occurred 2 times.
Let us see how to code this reduction method. Consider the above matrix A,
1. The difference between adjacent elements is calculated and it is given below.

[-3 0 0 1 0 0 0 1 0 -3 0 0 0 0 0 0 5 0 -2 0].

2. Convert it to logical format. The elements without repetition are denoted with one and the repeated elements with zero.
[1 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 1 0 1 0 1]

3. Find the position of the elements that has the value one in the above step.

DOI: 10.18535/ijecs/v6i3.57

[1 4 8 10 17 19 21].

4. Find the unique element values using the positions obtained from the above step. In the matrix A, the element at the position 1 is 5,the element at the position 4 is 2, the element at the position 8 is 3, similarly, the element at the position 21 is 4.

5. The first element in the matrix is 5, it has occurred 1 time and the second element occurred 3 times and so on. We obtained the occurrence of the first element alone from the matrix in the step 3. For the remaining elements, find the difference of the matrix in the step 3.
i.e. diff([1 4 8 10 17 19 21]);
The result after concatenating the first element of the matrix obtained in step 3 with difference for the matrix in the step 3 is [1 3 4 2 7 2 2].
6. Thus in the step 4 we obtain the elements without repetition, [5 2 3 4 1 6 4] and the occurrences in step 5, [1 3 4 2 7 2 2].
Therefore after Improved Run length encoding, we obtain the array without any repetition in the adjacent elements called Run values, [5 2 3 4 1 6 4],
and the occurrences of each element is called Run count given by
[1 3 4 2 7 2 2].
The vectors of Run values(RV) and their corresponding Run counts(RC) are shown below.

RV	RC
5	1
2	3
3	4
4	2
1	7
6	2
4	2

By concatenating Run values with Run count, we get

15
32
43
24
71
26
2 4

To limit the above values to lie within the range [0,255], Modulo division by 256 (Mod 256) is performed to get

15
32
43
24
71
26
2 4

This remainder array is sent to the destination.
Thus the matrix is reduced to 14 elements from 21 elements. It is observed that 7 Bytes are sufficient to represent 21 picture elements which require 21 Bytes actually. Compression ratio (CR) is defined as the ratio of No. of bytes of uncompressed data to No. of bytes of compressed data.

Compression Ratio = Size of Uncompressed data / Size of Compressed data.
Compression percentage = [1- (Size of compressed data / Size of uncompressed data)]*100.

Here compression ratio given by IRLE is 21/7 = 3 and compression percentage is 33.33 compression percentage.
By reversing the above steps, we get

5	1
2	3
3	4
4	2
1	7
6	2
4	2

It may be noted here that 5 has occurred 1 times, 2 has occurred 3 times, 3 has occurred 4 times, 4 has occurred 2 times, 1 has occurred 7 times, 6 has occurred 2 times and 4 has occurred 2 times. The original input can be reconstructed with help of the above result.
Now consider a full image of a cute baby of size 256 rows and 256 columns as shown in Figure 2.

Figure 2. Cute baby

On applying the IRLE Compression algorithm, the compression ratio obtained is 5.12 which is 80.47% as shown in Table 1. Some of other input images and their compression ratio and compression percentages are shown in the table.

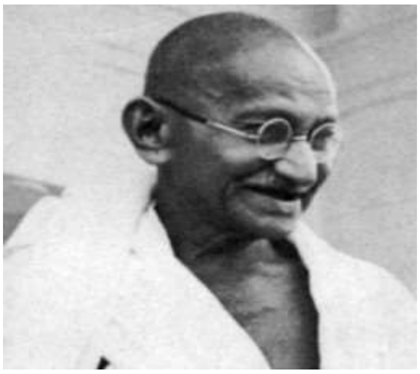

Figure 3. Gandhiji

DOI: 10.18535/ijecs/v6i3.57

Figure 4. Laden

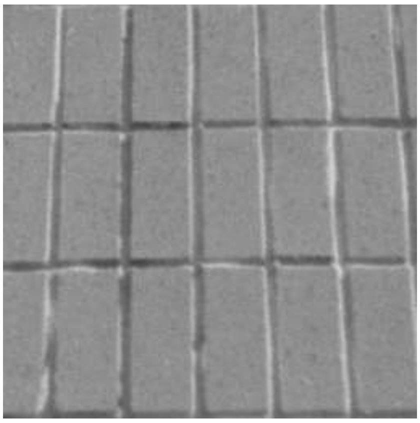

Figure 5. Texture

Figure 6. Clock

Name of the image	Size of the image	Compression Ratio	Compression Percentage(%)
Cutebaby.jpg	256x256	5.12	80.47
Gandhiji.jpg	256x256	2.23	55.08
Laden.jpg	256x256	3.16	68.36
Texture	256x256	2.0	50
Clock.tiff	256x256	2.15	53.52

Table 1. Compression ratio and compression % of images

From the above Table, it is observed that the highest compression ratio (5.12) and highest compression percentage (80.47) is obtained for Cute baby image which are better mhen compared to many existing methods. This The compression ratio may depend on the size and type of the input message. Security can be added to the compressed either by using secret sharing technique. In this analysis, all computations are carried out by using MATLAB [18].

5. Conclusion and Scope for Enhancement

In this investigation, we have developed an improved approach to image compression. This approach is very simple and efficient as it increases compression ratio. The IRLE reduces the bandwidth required to transmit image on the network and also the space required to store it. The compression ratio depends on the size and type of the input message. The proposed approach can be extended to any type of image. Security aspect can be incorporated to the compressed image so that the input image is sent to the destination in a secured manner. It is concluded that the proposed scheme is an efficient compression method for images and it provides high compression ratio and better compression percentage when compared to many existing methods.

References

[1] C. E. Rafael C. Gonzalez & Richard E. Woods, — Digital Image processing, 2nd Edition Pearson Education 2004.

[2] D. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," Proceedings of the IRE, vol. 40, **(1952)**, pp. 1098 - 1101.

[3] S. Golomb, "Run-length encodings," IEEE Transactions on Information Theory, vol. 12, **(1966),** pp. 399-401.

[4] D. M. Mark and D. J. Abel, "Linear quadtrees from vector representations of polygons," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 7, **(1985)**, pp. 344-9.

DOI: 10.18535/ijecs/v6i3.57

[5] J. P. Lauzon, D. M. Mark, L. Kikuchi and J. A. Guevara, "Two-dimensional run-encoding for quadtree representation," Computer Vision, Graphics, and Image Processing, vol. 30, **(1985),** pp. 56-69,.

[6] D. M. Mark, "The Use of Quadtrees in Geographic Information systems and spatial data handling," Proc. AutoCarto, London, vol. 1, **(1986)**, pp. 517-526.

[7] H. Samet, "Hierarchical Spatial Data Structures," in Proceedings of the first symposium on Design and implementation of large spatial databases (SSD '90), New York, NY, USA, **(1990)**, pp. 193-212.

[8] Z. Qian and R. Chen, "Adaptive 2d run-length coding and its applications," Journal of PLA institute of surveying and mapping, vol. 2, **(1993)**, pp. 18-23.

[9] H. Samet, "Spatial Data Structure," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 8, **(1995)**, pp. 532-538.

[10] G. K. Wallace, "The JPEG still picture compression standard", Commun. ACM, vol. 34, no. 4, **(1991)**, pp. 30-44.

[11] X. Shen and M. Spann, "3D Region representation based on run-lengths: operations and efficiency," Pattern Recognition, vol. 31, **(1998)**, pp. 575-585.

[12] Y. Yang, K. Chung and Y. Tsai, "A compact improved quadtree representation with image manipulations," Image and Vision Computing, vol. 18, **(2000)**, pp. 223 - 231.

[13] F. K. H. Quek, "An algorithm for the rapid computation of boundaries of run-length encoded regions," Pattern Recognition, vol. 33, **(2000)**, pp. 1637-1649.

[14] J. Shin, H. Hwang and S. Chien, "Detecting fingerprint minutiae by run length encoding scheme," Pattern Recognition, vol. 39, **(2006)**, pp. 1140-1154.

[15] C. M. Agulhari, I. S. Bonatti and P. L. D. Peres, "An Adaptive Run Length Encoding method for the compression of electrocardiograms," Medical Engineering & Physics, vol. 35, **(2013)**, pp. 145-153.

[16] B. Žalik and N. Lukač, "Chain code lossless compression using move-to-front transform and adaptive run-length encoding," Signal Processing: Image Communication, vol. 29, **(2014),** pp. 96-106.

[17] A. Alfalou C. Brosseau, N. Abdallah, and M. Jridi,"Simultaneous fusion, compression, and encryption of multiple images", OPTICS EXPRESS 24024Vol. 19, No. 24 OSA, 2011

[18] Alasdair Mcandrew, ―Digital Image processing with MatLab, Cengage learning 2004..

Author Profile

Dr. V. U. K. Sastry is presently working as Professor in the Dept. of Computer Science and Engineering (CSE), Director (SCSI), Dean (R & D), SreeNidhi Institute of Science and Technology (SNIST), Hyderabad, India. He was Formerly Professor in IIT, Kharagpur, India and worked in IIT, Kharagpur during 1963 – 1998. He guided 15 PhDs, and published more than 100 research papers in various international journals. His research interests are Network Security & Cryptography, Image Processing, Data Mining and Genetic Algorithms.

Dr. Ch. Samson obtained his Diploma from Govt Polytechnic, Hyderabad in 1994, B. E. from Osmania University in 1998 and M. E from SRTM University in 2000. He obtained Ph.D. from JNTUH, Hyderabad in 2015. He published 25 research papers in international journals and two papers in national conferences. He is currently working as Head in the Dept. of Information Technology, MVSR Engineering College, Hyderabad. His research interests are Image Processing, Image Cryptography and Network Security.