OPEN ACCESS
Research Journal of Information Technology
ISSN 1815-7432
DOI: 10.3923/rjit.2017.38.45

Research Article
Joint Photographic Experts Group Image Compression using Revised Run Length Encoding

Fawaz Waselallah Alsaade and Mahaboob Basha Shaik

College of Computer Science and Information Technology, King Faisal University, Hofuf, P.O. Box 400, 31982 Al-Ahsa, Kingdom of Saudi Arabia

Abstract

Background and Objective: There is a continuous increase in the use of multimedia contents that requires improvement in image compression methods. The main objective of this study is to present an improved Joint Photographic Experts Group (JPEG) compression method based on Revised Run Length Encoding (RRLE). **Materials and Methods:** The study was mainly based on reversible encoding techniques, Run Length Encoding (RLE) in JPEG and Optimized Run Length Encoding (ORLE) techniques. In RRLE, encoded data consisted of two sets namely the data set and head set. The data set comprises of a series of run or level values while the head set consists of a series of bits to represent whether corresponding item in the data set was run or level. **Results:** The problem of redundancy in representation of run of zeros associated with RLE and Optimized Run Length Encoding (ORLE) techniques were overcome in this technique. The revised run length encoding algorithm represented the run and level as individual items instead of pair and used the status bits to differentiate them. **Conclusion:** This study showed the effectiveness of the proposed algorithm in improving the compression rate as compared with RLE and ORLE. It was found that RRLE offered considerable improvement both in the compression rate and efficiency for the encoding process. This was mainly due to the elimination of the redundancy in the representation of run of zeros.

Key words: Revised run length encoding, end of block, compression rate, efficiency

Citation: Fawaz Waselallah Alsaade and Mahaboob Basha Shaik, 2017. Joint photographic experts group image compression using revised run length encoding. Res. J. Inform. Technol., 9: 38-45.

Corresponding Author: Fawaz Waselallah Alsaade, College of Computer Science and Information Technology, King Faisal University, Hofuf, P.O. Box 380, 31982 Al-Ahsa, Kingdom of Saudi Arabia

Competing Interest: The authors have declared that no competing interest exists.

Data Availability: All relevant data are within the paper and its supporting information files.

INTRODUCTION

Multimedia data requires considerable storage capacity and transmission bandwidth. Despite rapid progress in mass storage density, processor speeds and digital communication system, the demand for data storage capacity and data transmission bandwidth continues to outstrip the capabilities of available technologies[1]. Image compression is used as an effective solution to overcome this problem. The main goal of the image compression is to reduce the correlation and redundancy in an image. However, the different encoding techniques to compress images introduced by researchers were categorized into reversible and irreversible data compression techniques[1,2]. The reversible technique is loss less technique whereas irreversible technique is lossy.

The well known reversible methods are Huffman encoding[3], Lempel-ziv-welch[4] (LZW), Run Length Encoding (RLE)[2,3,5] and arithmetic coding[6]. The Huffman technique developed a dictionary corresponding the original data to a group of code words where each code word length is reciprocally proportional to the frequency of occurrence of the various symbol of the input data. According to the Lempel-ziv-welch method, the dictionary was developed dynamically for evaluating the input image or data. The run length encoding method does not apply any dictionary. It comprises of producing the sequences of the length with consecutive repetitions of a data in the input, being commonly utilized to compact binary sequences[2].

The irreversible compression technique achieved higher compression rate at the cost of loss of some data and was evaluated by the two metrics i.e., the distortion value and the compression ratio. The irreversible compression methods were divided into direct and transform schemes. In the direct method, the data of the image was analyzed and redundancies were removed to compress the image[7]. The transform methods used spectral and energy distribution analysis for the compression of data. Among the transform methods i.e., Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) are very widely used techniques[1,8]. Because such methods will help for the removal of redundancies during image compression.

The main aim of this study was to present investigations for resolving the redundancy problem present in different existing techniques and enhancing the overall efficiency of JPEG image compression.

MATETIALS AND METHOD

Reversible encoding techniques: A detailed description is presented for RLE[9] and ORLE[10] in the following below;

Run Length Encoding (RLE) in JPEG: In the JPEG image compression process[11,12] the RLE was the technique used as one of the steps[10]. Basically the JPEG algorithm breaks an image into a series of 8×8 matrices and proceeds to run a DCT transformation on each matrix. The DCT transformation isolates the important image components in the upper left portion of the matrix. In quantization, each entry in the frequency space block was divided by an integer and then rounded. This results in matrix with lot of zeros. To maximize the length of zero runs, JPEG scans the matrix in a zigzag order and then uses RLE to encode multiple zeros in a row efficiently. The RLE is based on the fact that the repeated symbols can be substituted by a number indicating how many times the symbol was repeated by itself. The RLE algorithm was illustrated in Fig. 1. The run of zeros are combined with non-zero elements and represented as (run, level) pair.

Based on the algorithm as shown in Fig. 1, it is understood that for consecutive non-zero elements, the value zero was redundantly paired with non-zero element to simply indicate the non-occurrence of zero. The redundancy representing the run of zeros was an important problem

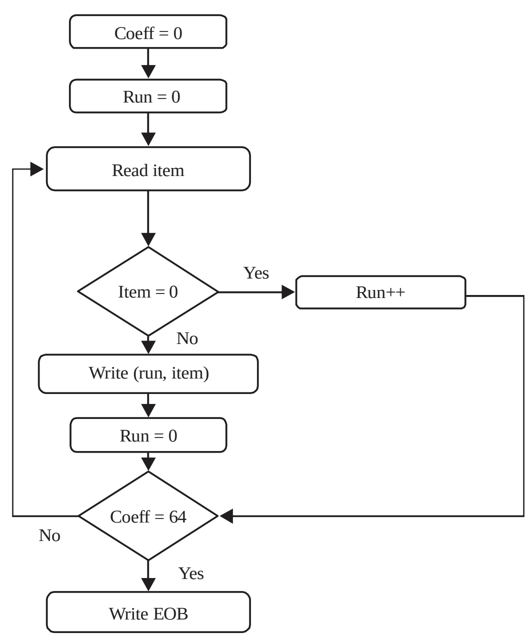

Fig. 1: Flow diagram for run length encoding
EOB: End of block, Coeff: Coefficient

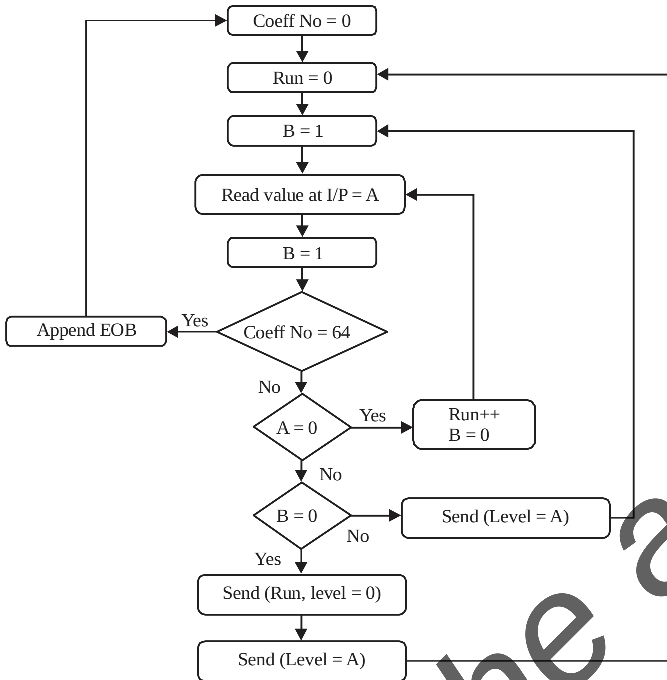

Fig. 2: Flow diagram for optimized run length encoding
EOB: End of block, Coeff: Coefficient

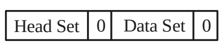

Fig. 3: RRLE encoded format

associated with RLE that can adversely influence the compression rate. In some scenarios where the runs of smaller length were in excess, RLE performance was poor and failed to compress[13]. The End of Block (EOB) is represented with the (0, 0) pair.

ORLE: Recently, the Optimized Run Length Encoding (ORLE)[9,10] was introduced to overcome the problem associated with RLE. However, the ORLE is similar to RLE but differs in representing the way run of symbols. The non-zero values were represented just with level excluding the run parameter. Only when a series of consecutive zeroes occur, it uses the (run, level) pair. The ORLE algorithm is illustrated in Fig. 2. The flag B is used to check whether a zero or successive zeros are present in the data to be encoded. The End of Block (EOB) is represented with the (0, 0) pair same as in the RLE technique.

The ORLE reduces the unintended redundancy to some extent, but could not eliminate it completely[9,10]. The reason is that whenever a run of zeros is present, the value zero was repeated along with that run. It represents the single zero between non-zero characters with a 2-digit sequence of (1, 0). When more number of single zeros are present, there will be more redundancy affecting the compression rate badly.

The main cause of redundancy in both these techniques seems to be the (run, level) pair. In order to overcome this problem, a Revised Run Length Encoding (RRLE) technique was presented in this study.

Proposed revised run length encoding technique: In RRLE, encoded data consisted of two sets namely the data set and head set. The data set comprised of a series of run or level values while the head set consisted of a series of bits to represent whether corresponding item in the data set is run or level. A run and level are represented by zero and one, respectively. The two sets (head set and data set) are merged into one set. A zero is used in between the two sets and also at the end as shown in Fig. 3. The first zero indicated the end of head set and the beginning of data set. The second zero indicated the End of Block (EOB). This differs from RLE and ORLE where the EOB is represented with two consecutive zeroes.

Algorithm 1 of Revised Run Length Encoding (RRLE) is illustrated below:

Algorithm 1: RRLE_ALGORITHM

```
Procedure PROC_RRLE
Input: QB-quantized image block in zigzag sequence
Output: EB-encoded block
runCount←0
headData←0
for all cbyte in QB do
        if cbyte!=0 then
           if runCount!=0 then
                append runCount to dataSet
                append 0 to headSet
                runCount←0
           append cbyte to dataSet
           append 1 to headset
        else
           rundCount←runCount+1
        add headset to EB
        append 0 to EB
        append dataset to EB
        append 0 to EB
```

In RRLE approach, instead of using run and level as pair, they were individually represented based on their occurrence and differentiated with one bit of information. Another difference between the proposed RRLE and the other two encoding techniques was that RRLE used only one bit to indicate the case of no run of zero. On the other hand, RLE and ORLE used a full byte for the same. Hence, the proposed RRLE was expected to resolve the redundancy problem and in turn can enhance the overall efficiency of image compression.

RESULTS AND DISCUSSION

Initially, the effectiveness of the proposed image compression method was analyzed by using a sample of 8×8 block. Then, a set of images of various sizes and complexities were used to check and confirm the performance of the proposed technique. Later on, this performance was compared with the performance of RLE and ORLE techniques. The proposed method was evaluated using the MATLAB. In this simulation set up, tif images were considered as the input images. The two parameters such as the Compression Rate (CR) and the percentage of efficiency (E%) were used as performance metrics to compare the proposed technique with the existing techniques.

The formula for compression rate is give in Eq. 1:

$$CR = \frac{Size_{original}}{Size_{compressed}} \quad (1)$$

The percentage of efficiency was calculated by the following formula in Eq. 2:

$$E(\%) = \frac{Size_{original} - Size_{compressed}}{Size_{original}} \times 100 \quad (2)$$

$$\begin{pmatrix} 102 & -33 & -3 & -4 & -2 & -1 & 00 \\ 21 & -2 & -3 & 0 & -1 & 0 & 00 \\ -3 & 0 & 1 & 0 & 0 & 0 & 00 \\ 2 & 0 & 0 & 0 & 0 & 0 & 00 \\ 1 & 0 & 0 & 0 & 0 & 0 & 00 \\ -2 & 0 & 0 & 0 & 0 & 0 & 00 \\ 0 & 0 & 0 & 0 & 0 & 0 & 00 \\ 0 & 0 & 0 & 0 & 0 & 0 & 00 \end{pmatrix}$$

Fig. 4a: Quantized block

$$\begin{pmatrix} (-33, 21, -3, -2, -3, -4, -3, 0, 2, 1, 0, 1, 0, -2, -1, \\ -1, 0, 0, 0, -2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0..0) \end{pmatrix}$$

Fig. 4b: Zig zag sequence

Fig. 4(a-b): Sample data

$$\begin{pmatrix} (0, -33), (0, 21), (0, -3), (0, -2), (0, -3), 0, -4), (0, -3), (1, 2), \\ (0, 1), (1, 1), (1, -2), (0, -1), (0, -1), (3, -2), (11, 1), (0, 0) \end{pmatrix}$$

Fig. 5a: RLE

$$\begin{pmatrix} (-33, 21, -3, -2, -3, -4, -3, (1, 0), 2, 1, (1, 0), 1, (1, 0), \\ -2, -1, -1, (3, 0), -2, (11, 0), 1, (31, 0), (0, 0)) \end{pmatrix}$$

Fig.5b: ORLE

Fig. 5(a-b): Encoded data

where, $Size_{Original}$ is the original size of the image and $Size_{Compressed}$ is the compressed size of the image in bytes.

Input image- 8×8 block: The 8x8 block shown in Fig. 4(a) is the sample data block with size of 64 bytes used for demonstrating the performance of the three techniques i.e., RLE, ORLE and RRLE. The zigzag sequence of that block is represented in Fig. 4(b).

The encoded data generated by RLE is shown in Fig. 5(a). It is consisted of 32 bytes and compression rate of 2. It is observed from Fig. 5(a) that the symbol zero is repeated 10 times, affecting the compression rate negatively. On the other hand, the encoded data generated by ORLE is shown in Fig. 5(b). This encoded data consisted of 29 bytes with a compression rate of 2.21. It was observed that in ORLE the (run, 0) pair is used whenever a run of zero occurs. Since, the value zero was repeated for each run of zeros, still some redundancy persisted and caused a fall in the compression rate. Further, the impact could be more when large numbers of small sequence of zeros are present in between non zero elements. In the current example, zero was repeated 6 times thus affecting the compression rate.

As explained above, the proposed RRLE took the data generated in the quantized phase and stored a series of runs and levels in the data set. It stored the bit information in head

$$(-33, 21, -3, -2, -3, -4, -3, 1, 2, 1, 1, 1, 1, -2, -1, -1, 3, -2, 11, 1))$$

Fig. 6a: Data set

$$((1,1,1,1,1,1,1,0,1,1,0,1,0,1,1,1,0,1,0,1))$$

Fig. 6b: Head set in bits

$$((-2, -4, 80))$$

Fig.6c: Head set in bytes

$$\begin{pmatrix} (-2, -41, 80, 0, -33, 21, -3, -2, -3, -4, -3 \\ 1, 2, 1, 1, 1, 1, -2, -1, -1, 3, -2, 11, 1, 0) \end{pmatrix}$$

Fig. 6d: FMerged

Fig. 6(a-d): RRLE encoded data

Table 1: Compression rates and efficiencies for RLE, ORLE and RRLE using a sample 8×8 block

Input size	Technique	Encoded size	CR	E (%)
64	RLE	32	2.00	50.0
64	ORLE	29	2.21	54.7
64	RRLE	25	2.56	60.9

RLE: Run length encoding, ORLE: Optimized run length encoding, RRLE: Revised run length encoding

set, representing whether a corresponding item in the data set was a run or level. The run was represented with a zero and the level with one. For the sample data considered, Fig. 6(a-d) represented data set, head set in bit form, head set in byte form and final merged encoded data, respectively.

In the final merged encoded data, the first zero separated the head set and data set while the second zero indicated the End of Block (EOB). For the sample input, the encoded data generated is with size of 25 bytes and compressed rate of 2.56.

A comparative analysis is carried out on the results from the proposed model with the existing methods for the sample 8×8 data block (Table 1). It clearly reveals that the compression ratio and the energy are higher than the exiting methods. Besides, it also provides direct comparison between the three encoding techniques with compression rate and efficiency.

It is found in Table 1 that the use of the proposed RRLE technique has increased the compression rate and the efficiency for the encoding process. These outcomes confirm the earlier suggestion that the use of RRLE can resolve the redundancy for representing the run of zeros. The effectiveness of RRLE is due to the way of representation of run and level. However, with respect to the efficiency, the results showed that the ORLE improved the RLE by 9.4% whereas the RRLE improved the RLE and ORLE by 21.9 and 13.8%, respectively. The study results are in line with the findings of Khan *et al.*[14] who adopted the least significant bits substitution method for data hiding in the image followed by the RLE scheme to reduce the size of stego image. They achieved a hiding capacity of 50% with a reasonable high PSNR (i.e., greater than 30 dB limit) and a compression ratio of greater than 1 has been achieved.

Fig. 7: Cameraman-original image

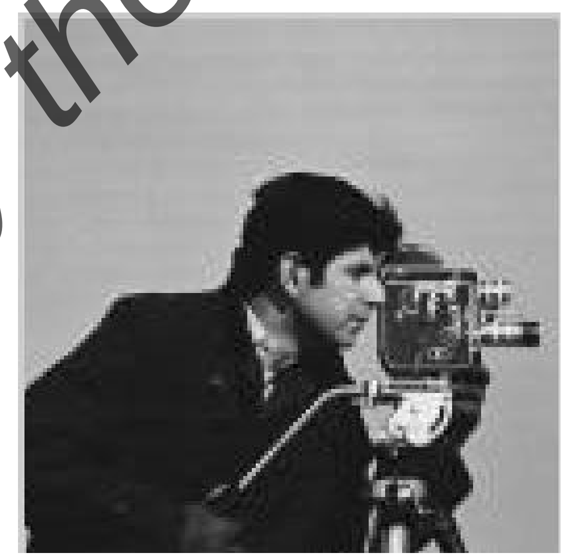

Fig. 8: Restored test image by using RRLE

Input images: Images of different types and sizes are considered for the simulation purpose. The size of the input images ranges from 51296-6120546 bytes. Two parameters such as the compression rate and efficiency are considered to compare the performance. Figure 7 showed the original input for performing the RRLE process. However, the test image restored by using the proposed method is presented in Fig. 8.

The experimental results of the compression ratio and efficiency of the proposed RRLE were compared with different techniques are shown in Table 2.

Results in Table 2 are in agreement with the earlier observation that better performance can be obtained using the proposed RRLE technique. It can be observed from the average values of CR and E in Table 2 that the best encoding technique in this experiment setup is RRLE with an average value of 8.52 and 82.5% for CR and E, respectively. Furthermore, it is worth mentioning that the RRLE improved the average efficiency of RLE and ORLE by 14.4 and 10.7%, respectively. While the ORLE improved the RLE by 3.4%. However, in some specific cases, the ORLE failed to improve RLE. This is clearly shown in cases for the two images: paper1.tif and cameraman.tif as shown in Table 2. This proved that the performance of RRLE is consistent irrespective of the size and contents of the image. Similar results were reported by Singh *et al.*[15] who stated that run length coding is the standard coding technique for compressing the images, especially when images are compressed by block transformation. Also, this method counts the number of repeated zeros which is represented as RUN and appends the non-zero coefficient represented as LEVEL following the sequence of zeros. While, Long *et al.*[16] reported that image classification can be approximately divided into two categories namely the scene recognition and object recognition applying two important steps such as dictionary learning and feature coding.

A direct comparison of the efficiencies and the compression rates of three techniques is shown in Fig. 9 and 10, respectively, where the x-axis represents images with their serial order as given in Table 2 and the lines having blue, red

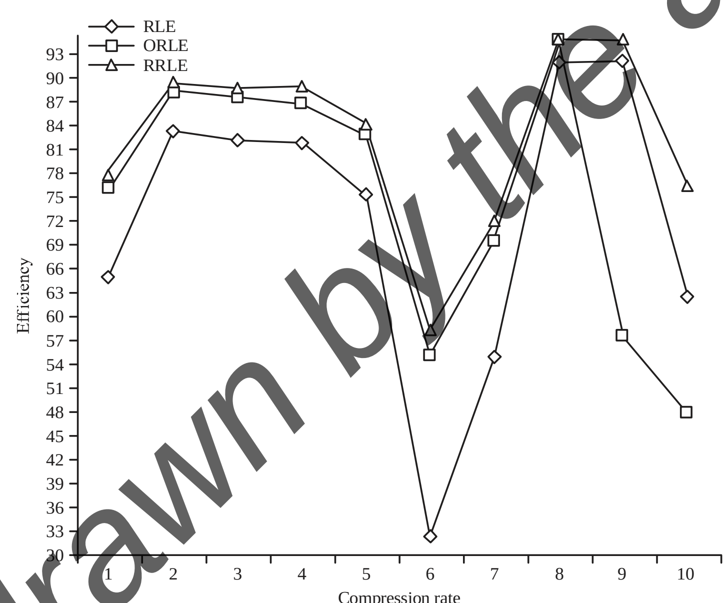

Fig. 9: Efficiency of RLE, ORLE and RRLE

RLE: Run length encoding, ORLE: Optimized run length encoding, RRLE: Revised run length encoding

Table 2: Compression rates and efficiencies of RLE, ORLE and RRLE using different sample images

		JPEG-RLE			JPEG-ORLE			JPEG-RRLE		
Image	Original size	Size	CR	E (%)	Size	CR	E (%)	Size	CR	E (%)
Trees.tif	90300	31618	2.85	64.9	21579	4.18	76.1	19626	4.60	78.2
Kids.tif	127200	21098	6.02	83.4	14764	8.61	88.4	13436	9.46	89.4
Moon.tif	192246	34318	5.60	82.1	24254	7.92	87.4	22203	8.65	88.5
Mandi.tif	6120546	1117374	5.47	81.7	811604	7.54	86.8	691494	8.85	88.7
Eight.tif	74536	18484	4.03	75.2	12854	5.79	82.8	11803	6.31	84.2
Westconcord ortho.png	133224	90112	1.47	32.4	60050	2.21	54.9	55551	2.39	58.3
Forest.tif	134547	60458	2.22	55.1	40803	3.29	69.7	37612	3.57	72.0
Peppers.png	589824	49106	12.01	91.7	33830	17.40	94.3	30634	19.30	94.8
Paper1.tif	51296	4050	12.67	92.1	21833	2.34	57.4	2875	17.80	94.4
Cameraman.tif	65536	24554	2.66	62.5	34161	1.91	47.8	15407	4.24	76.5
Average	757926	145117	5.50	72.1	107573	6.12	74.6	90064	8.52	82.5

CR: Compression rate, E: Efficiency

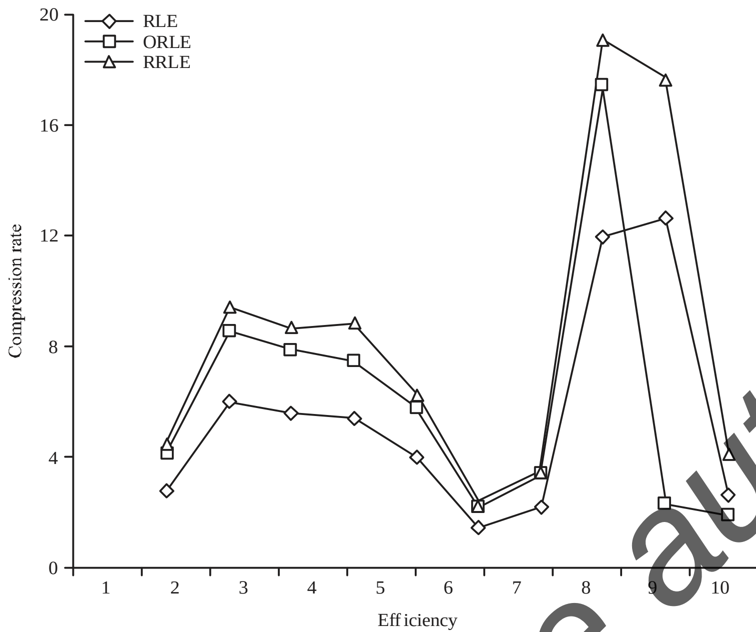

Fig. 10: Compression rate of RLE, ORLE and RRLE

RLE: Run length encoding, ORLE: Optimized run length encoding, RRLE: Revised run length encoding

and green color represent the 3 encoding techniques namely RLE, ORLE and RRLE, respectively. It can be seen from Fig. 9 and 10 that (in all cases) RRLE was more effective than RLE and ORLE. This confirmed the earlier notion that the use of run and level as individual items improved the encoding process with respect to compression rate and efficiency. These study findings are in agreement with the findings of Bandyopadhyay *et al.*[17], Wu *et al.*[18] and Irfan *et al.*[19] who suggested that two-dimensional run-length encoding (I2DRLE) scheme is more appropriate for representing grayscale images. Also, they stressed on the lossless compression of image using approximate matching technique and run length encoding.

CONCLUSION

This study presented a novel and promising encoding technique for image compression which based on the experimental investigations. It was found that RRLE offered considerable improvement both in the compression rate and efficiency for the encoding process. The main reason behind this was to resolve the redundancy problem in the representation of run of zeros. Additionally, RRLE was consistent in achieving better compression rate irrespective of the size and the contents of the image. Whereas the performance of RLE and ORLE was affected by the suitability of the contents of the image.

SIGNIFICANT STATEMENT

This study discovered that continuous increase in the use of multimedia contents requires improvement in image compression methods. This study was carried to present an improved JPEG compression method based on Revised Run Length Encoding (RRLE). The study was mainly based on reversible encoding techniques such as RLE and ORLE.

REFERENCES

1. Subramanya, A., 2001. Image compression technique. IEEE. Potentials, 20: 19-23.
2. Merkl, F.J., 1988. Binary image compression using run length encoding and multiple scanning techniques. M.Sc. Thesis, Rochester Institute of Technology, New York.
3. Yang, E. H. and L. Wang, 2009. Joint optimization of run-length coding, huffman coding and quantization table with complete baseline JPEG decoder compatibility. IEEE Trans. Image Process., 18: 63-74.
4. Welch, T.A., 1984. A technique for high performance data compression. Computer, 17: 8-19.
5. Tu, C., J. Liang and T.D. Tran, 2003. Adaptive runlength coding. IEEE Signal Process. Lett., 10: 61-64.
6. Witten, I.H., R.M. Neal and J.G. Cleary, 1987. Arithmetic coding for data compression. ACM Commun., 30: 520-540.
7. Taskin, C. and S.K. Sarikoz, 2008. An overview of image compression approaches. Proceedings of the 3rd International Conference on Digital Telecommunications, Jun. 29-Jul. 5, Bucharest, pp: 174-179.

8. Gonzalez, R.C. and R.E. Woods, 2008. Digital Image Processing. 3rd Edn., Prentice Hall, New Jersey, USA., ISBN-13: 9780131687288, Pages: 954.
9. Islam, Q., M.B. Akhtar and A.M. Qureshi, 2014. Storage area and temporal based optimized run length codec for classic JPEG image compression (SAT-BORL CODEC). United States Patent Application No. US20140205199 A1, Jul 24, 2014.
10. Akhtar, M.B., A.M. Qureshi and Qamar-ul-Islam, 2011. Optimized run length coding for jpeg Image compression used in space research program of IST. Proceedings of the International Conference on Computer Networks and Information Technology, July 11-13, 2011, IEEE., pp: 81-85.
11. Pennebaker, W.B. and J.L. Mitchell, 1993. JPEG: Still Image Data Compression Standard. Springer, New York, USA., ISBN-13: 9780442012724, Pages: 638.
12. Wallace, G.K., 1991. The JPEG still picture compression standard. Commun. ACM., 34: 30-44.
13. Amin, A., H.A. Qureshi, M. Junaid, M.Y. Habib and W. Anjum, 2011. Modified run length encoding scheme with introduction of bit stuffing for efficient data compression. Proceedings of the 2011 International Conference for Internet Technology and Secured Transactions, December 11-14, 2011, IEEE., pp: 668-672.
14. Khan, S., T. Khan, M. Naeem and N. Ahmad, 2015. Run-length encoding based lossless compressed image steganography. Sindh Univ. Res. J. (Sci. Ser.), 47: 541-544.
15. Singh, A. and V.P. Singh, 2013. An enhanced run length coding for JPEG image compression. Int. J. Comput. Applic., 72: 21-26.
16. Long, X.Z., H.T. Lu and W.B. Li, 2014. Image classification based on nearest neighbor basis vectors. Multimedia Tools Applic., 71: 1559-1576.
17. Bandyopadhyay, S.K., T.U. Paul and A. Raychoudhury, 2011. Image compression using approximate matching and run length. Int. J. Adv. Comput. Sci. Applic., 2: 117-121.
18. Wu, P., S. Zhou1, B. Wan1, F. Fang and S. Zhou, 2016. An improved two-dimensional run-length encoding scheme and its application. Int. J. Signal Process. Image Process. Pattern Recogn., 9: 339-346.
19. Irfan, M.A., N. Ahmad and S. Khan, 2014. Analysis of varying least significant bits DCT and spatial domain stegnography. Sindh Univ. Res. J. (Sci. Ser.), 46: 301-306.