International Journal of Hybrid Information Technology
Vol.9, No.1 (2016), pp. 319-328
http://dx.doi.org/10.14257/ijhit.2016.9.1.27

Optimized Image Compression Techniques for the Embedded Processors

Ali A. Al-hamid, Ahmed Yahya and Reda A. El-Khoribi

Electrical Eng. Dept. Faculty of Engineering, Al-Azhar University
alihamid@azhar.edu.eg
Electrical Eng. Dept. Faculty of Engineering, Al-Azhar University
ahmed_yahya_1@yahoo.com
Computer Eng. Dept. Faculty of Engineering, Cairo University
ralkhoribi@staff.cu.edu.eg

Abstract

Optimizing the utilization factor of the system resources such as efficiency, bandwidth, and the storage capacity for cost reduction is one important aim of enormous amount of studies. For the image compression one can use the embedded processors as the most suitable ones. This image compression schemes for images will be based on the Discrete Cosine Transform (DCT). This paper implements an efficient and effective algorithm for still image compression of relatively high signal to noise ratio. The implemented technique considers that only zeros of the zigzag scanning is the repeated runs. This results in possibility of zero byte of the Run Length Encoding (RLE) output elimination. Word-length reduction and higher compression ratio can be customized.

Keywords: Image Compression, DCT, RLE, Embedded Processors, and storage capacity.

1. Introduction

Data compression is the art or since of representing information in a compact form i.e. reduces the size of the data to reduce storage space and transmission bandwidth [1]. In this paper, we propose a new approach for image Compression (Proposed RLE 1, and Proposed RLE 2) and discuss the idea behind the two proposed techniques. This done using graphs, tables and analysis to evaluate our results contribution due to the proposed techniques. The rest of the paper is organized as follows. Section 2 introduces image compression techniques. Section 3 discusses the JPEG standard with some details, the main features of DCT and 2D Equations. Section 4 shows the main compression parameters. Section 5 discusses the traditional Run Length Encoding (RLE) features, and introduces the proposed algorithms. Section 6 presents the simulation results and experimental work; it is also compares the proposed results with the old ones. Finally, Section 7 concludes the paper.

2. Image Compression Techniques

Image compression reducing the amount of data required to represent a digital image by removing redundant or non-vital data. There are different types of redundancy present in an image, such as Spatial Redundancy, Statistical Redundancy and Human Vision Redundancy (HVR) [3].

2.1 Lossless Compression

ISSN: 1738-9968 IJHIT

Lossless compression technique involves no loss of information. If data have been lossless compressed the original data can be recovered exactly from the compressed data, Lossless compression is generally used for applications that cannot tolerate any difference between the original and reconstructed data [1]. Source coding (Huffman, LZW, and Arithmetic) and RLE are the most common Lossless compression techniques. Lossless methods are used for text and image compression such as a medical imaging, compression ratio (CR) is usually below 2 for this compression technique.

2.2 Near Lossless Compression

Near lossless compression technique is a lossy compression method where the reconstructed pixels differ from the original pixels by no more than a predetermined value, in near-lossless compression data is guaranteed to be within a specified range based on the near-lossless threshold [3].

2.3 Lossy Compression

Lossy compression reduces the original image size by removing non-vital information, *i.e.*, this technique involves some loss of information and data that have been compressed, using lossy techniques generally cannot be recovered or reconstructed exactly. In return for accepting this distortion in the reconstruction we can generally obtain much higher compression ratios than is possible with lossless compression [1]. This type of compression is suitable for pictures, video and audio. CR can reaches over 10 for this compression technique.

3. DCT based JPEG Standard

JEPG is the international standard for the effective compression of the still digital images, it is defines the standard steps to compress an image in to a stream of bytes and decompress it again to generate the original image back [6]. DCT based JPEG Encoder /Decoder block diagram are shown in figure (1) and figure (2). The JPEG is used for both color or grayscale still images [2]. A DCT based method is specified for lossy compression, and a predictive method for lossless compression [5]. JPEG features a simple lossy technique known as the Baseline method, a subset of the other DCT based modes of operation. The Baseline method has been by far the most widely implemented JPEG method to date, and is sufficient in its own right for a large number of applications.

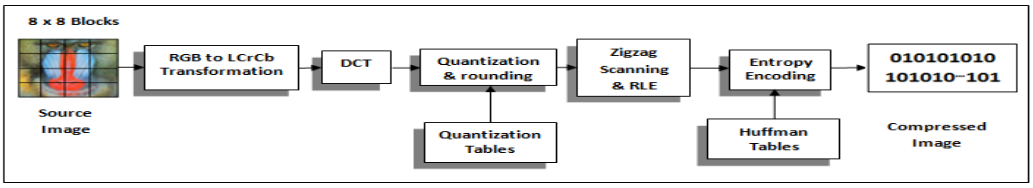

Figure 1. DCT based JPEG Encoder

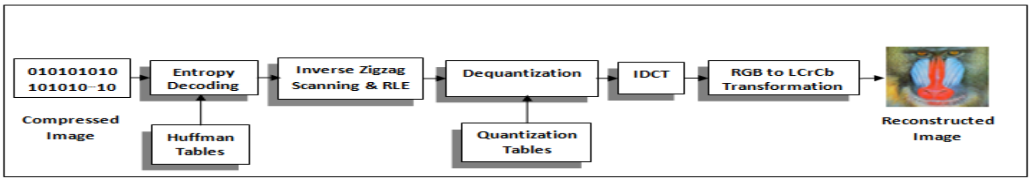

Figure 2. DCT based JPEG Decoder

The DCT is usually applied to reduce spatial redundancy in order to achieve good compression performance. The main feature of using DCT is that it takes correlated input data and concentrates its energy in just the first few transform coefficients. If the input data consists of correlated quantities, then most of the transform coefficients produced by the DCT are zeros or small numbers, and only a few are large (normally the first ones) [2]. It minimizes the block like appearance called blocking arti-facts. DCT is primarily a lossy method of compression. It was designed specifically to discard the information that the human eye cannot easily see. The two dimensions DCT equation as following bellow [2]:

$$C(u,v) = \propto(u) \propto(v) \sum_{x=0}^{N-1} \sum_{x=0}^{M-1} f(x,y) \mathrm{Cos}\left[\frac{(2x+1)u\pi}{2N}\right] \mathrm{Cos}\left[\frac{(2x+1)v\pi}{2M}\right] \quad (1)$$

For u=0, 1, 2... N – 1 , v=0, 1, 2... M – 1,

$$\propto(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u \neq 0 \end{cases},$$

$$\propto(v) = \begin{cases} \sqrt{\frac{1}{M}} & \text{for } v = 0 \\ \sqrt{\frac{2}{M}} & \text{for } v \neq 0 \end{cases} \quad (2)$$

The 2D IDCT equation as following:

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} \propto(u) \propto(v) C(u,v) \mathrm{Cos}\left[\frac{(2x+1)u\pi}{2N}\right] \mathrm{Cos}\left[\frac{(2x+1)v\pi}{2M}\right] \quad (3)$$

4. Performance Parameters of Image Compression

The two major parameter of image compression qualification are quality of image and compression factor.

4.1 Error Metrics

Mean square error (MSE) is a criterion for an estimator the choice is the one that minimizes the sum of squared errors due to bias and due to variance.

$$\mathrm{MSE} = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left(A(i,j) - B(i,j)\right)^2 \quad (4)$$

Signal to peak noise ratio (PSNR) is ratio between the maximum possible power of a signal and the power of corrupting noise. The PSNR is used to measure the quality of reconstructed image [4].

$$\mathrm{PSNR} = 10 \log_{10}\left(\frac{255}{\mathrm{MSE}}\right) \quad (5)$$

4.2 Compression Metrics

Compression ratio (CR) is defined as the ratio between the uncompressed image size and compressed image size.

$$CR = \left(\frac{\text{Uncompressed file size}}{\text{Compressed file size}}\right) \quad (6)$$

Bits per pixel ratio (BPP) give the number of bits store one pixel of the image.

$$BPP = \left(\frac{\text{Compressed file size in Bits}}{\text{Total number of pixels in Image}}\right) \quad (7)$$

5. Run Length Encoding (RLE)

RLE is a simple lossless compression algorithm. The idea behind using RLE approach in data compression is that if a data item d occurs n consecutive times in the input stream, replace the n occurrences with the single pair nd, the n consecutive occurrences of a data item are called a run length of n [2]. In this case we replace each data run with three byte (Pilot, Run, and Data).

Consider the following data pattern: **67 1 1 0 0 0 0 2 0 1 1 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

The generated RLE output will be**: 67 1 1 255 4 0 2 0 255 2 1 3 255 52 0**

5.1 Analysis of Traditional RLE

From the previous example we can see that: First using RLE is not optimum in all cases as example we replaced data **1 1** with **255 2 1** *i.e.*, replace two bytes with three bytes, Second in image compression after Quantization step almost all Runs is Zeros.

From the last two results we propose that when dealing with RLE results, we can neglect any run with length 2, so **1 1** will be leaved as it **1 1** without coding to save 1 byte. The complete analysis to Runs Properties for different types of standard images (natural, human, animal… *etc*.) are shown in Figures 3 through 5. The most effective Run value for the given results is Zero.

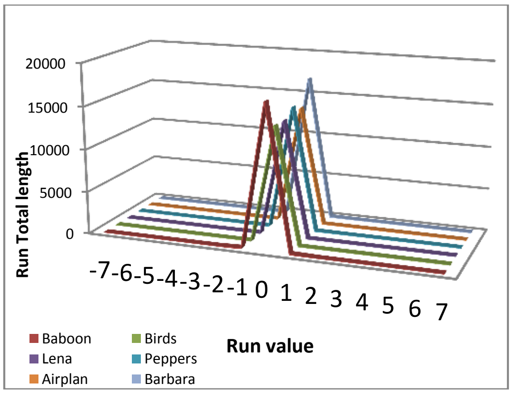

Figure 3. Runs Status for Standard Images

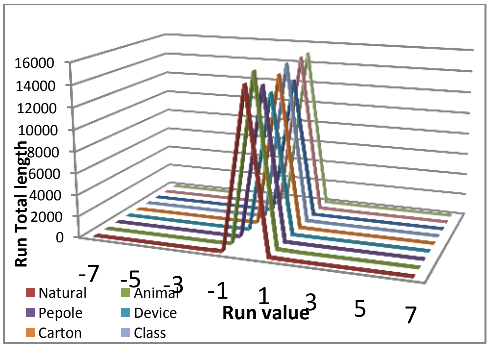

Figure 4. Runs Status for Image Types

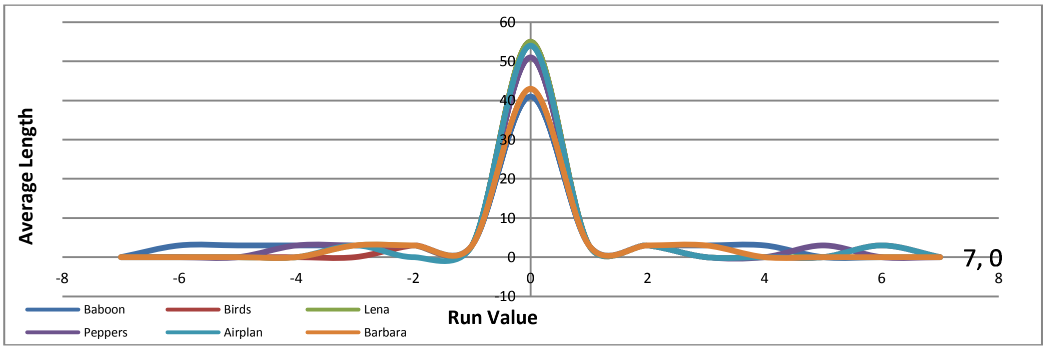

Figure 5. Avrage Runs Lengths for Zigzag Scanning

Based on the above analysis we can propose two modifications on RLE in image compression, proposed RLE 1 and proposed RLE 2.

5.2 Proposed RLE 1

The proposed algorithm considers that any run will be only zeros (even it is not). Thus the RLE output will be as follow, Runs (255, X), where 255 is the pilot and X is the Run length. If the output matrix from Quantization & Zigzag steps is: **67 1 1 0 0 0 0 2 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.**

Then the proposed RLE Encoding output in this case will be: **67 1 1 255 4 2 0 255 3 255 52**

In the last proposed RLE 1 encoding output the numbers of saved bytes are (3), the RLE Decoding output will be: **67 1 1 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**

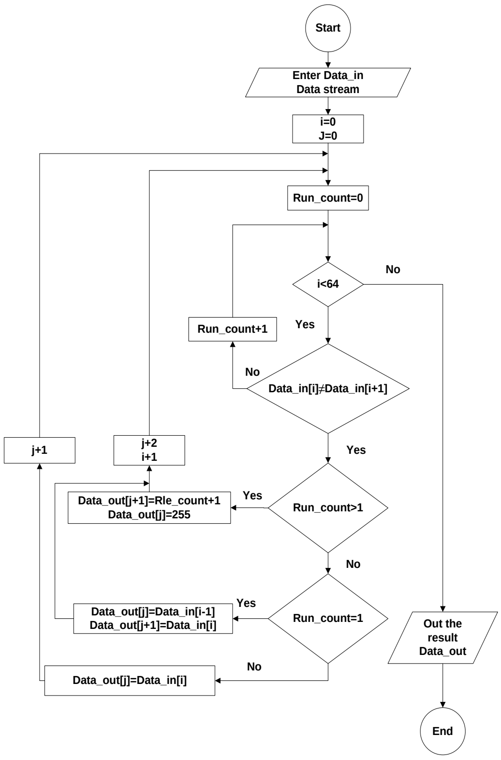

Figure 6. Proposed Flow Chart Encoder (Proposed RLE 1)

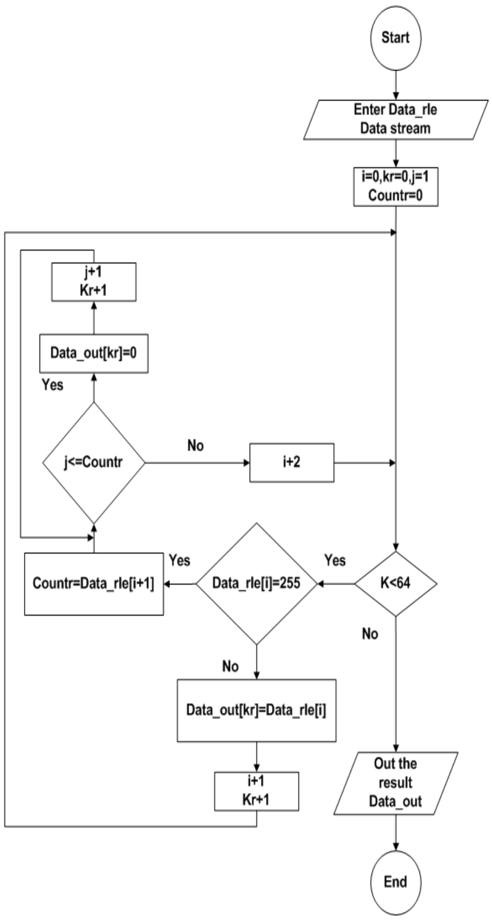

Figure 7. Proposed Flow Chart Decoder (Proposed RLE 1)

We can see from the RLE Decoding output that the Run (1 1 1) were decoded as (0 0 0) due to our proposed algorithm which decrease PSNR. Finally Proposed RLE 1 algorithm save a byte every time using RLE which leads to increase the compression ratio, But at the same time this reduce PSNR as shown in the last example. The proposed algorithms (Encoder / Decoder) are showing in figures (6, 7).

5.3 Proposed RLE 2

The proposed RLE 2 is dealing only with Zero generated due to runs, and ignores any other values. The RLE output (pilot, Runs) will save byte every time using RLE. To achieve that, consider the following example: If the output matrix from Quantization & Zigzag steps is: **67 1 1 0 0 0 0 2 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**.

The proposed RLE 2 Encoding output in this case will be: **67 1 1 255 4 2 0 1 1 1 255 52**

In this encoding output of RLE 2, the numbers of saved bytes are 2, and the Proposed RLE 2 Decoding output will be: **67 1 1 0 0 0 0 2 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**.

This RLE decoding output indicates that this algorithm has no effect on PSNR. This result verifies that the proposed algorithm Proposed RLE 2 increase Compression ratio while maintaining the same PSNR. The proposed Encoding algorithm as showing in Figure (8), while its decoding algorithm is the same as shown in Figure (7).

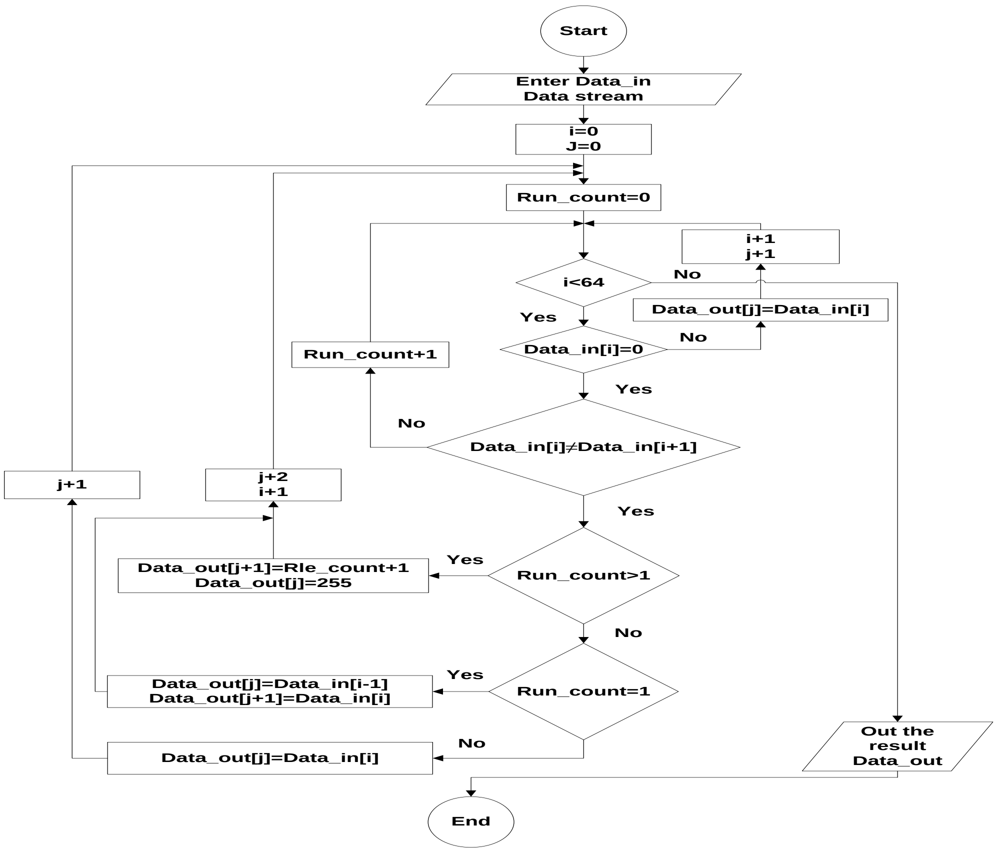

Figure 8. Proposed Flow Chart Decoder (Proposed RLE 2)

6. Analysis of Experimental Results

The two proposed algorithms are implemented using C language and C-Free compiler. We have applied our algorithms to 6 different standard color images {Mandrill (Baboon), Birds, Lena, Pepper, Airplane, and Barbara with dimensions (512x512)}.

Table 1. Experimental results for Standard images

Image	RLE			Proposed RLE 1			Proposed RLE 2		
	Size K byte	CR	PSNR	Size K byte	CR	PSNR	Size K byte	CR	PSNR
Mandrill	121	6.37	26.54	103.7	7.4	26.46	104	7.39	26.54
Birds	66.7	11.5	31.18	53.2	14.4	31.13	53.26	14.4	31.18
Lena	69.5	11.1	31.69	56.1	13.7	31.67	56.18	13.7	31.69
Pepper	81.6	9.41	31.30	67.3	11.4	31.24	67.36	11.4	31.30
Airplane	72.7	10.6	32.29	59.1	13	32.24	59.18	13	32.29
Barbara	96.3	7.97	27.60	79.7	9.63	27.58	79.85	9.62	27.60

As we can show from the experimental results listed in Table 1 the two proposed algorithms increase the overall compression ratio. The tested performance of the proposed algorithms can be described as follow:

1) The first proposed algorithm (Proposed RLE 1) increase the compression ratio over that both (RLE, Proposed RLE 2), but at the same case decrease PSNR.
2) The second proposed algorithm (Proposed RLE 2) increases the compression ratio over than RLE at the same PSNR not proposed RLE 1.
3) The increasing in compression ratio in Proposed RLE 1 with respect to RLE in range (14.30% to 20.24%) with average value 17.88%, and the decreasing in PSNR in range (0.02 to 0.08 dB) with average value 0.046667 dB.
4) The increasing in compression ratio in Proposed RLE 2 with respect to RLE in range (14.05% to 20.15%) with average value 17.75%.
5) The decreasing in compression ratio in Proposed RLE 2 with respects to Proposed RLE 1 is slightly small in range (0.09% to 0.29%) with average value 0.16%.

The tabulated results of the proposed algorithms are shown in Figure 9, and Figure 10.

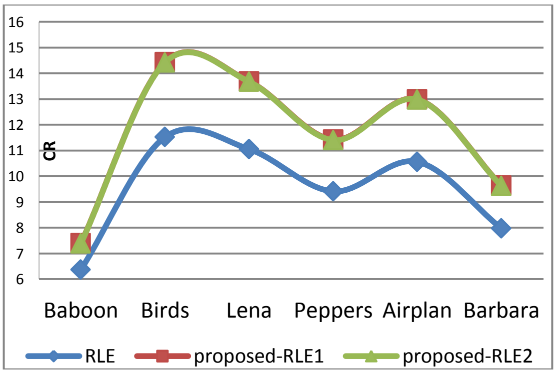

Figure 9. CR for Standard Image

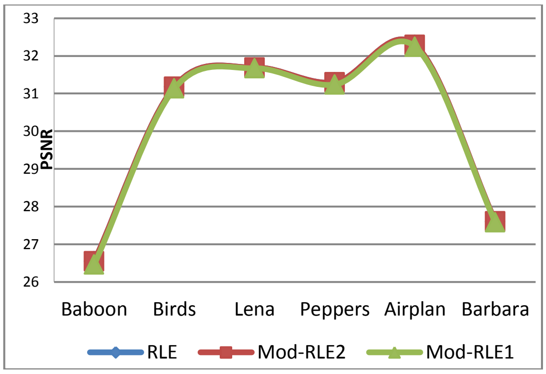

Figure 10. PSNR for Standard Image

The experimental results of standard testing images for the proposed and traditional RLE are shown in Figure11 through Figure 16, these Figures verified the proposed ideas in this paper.

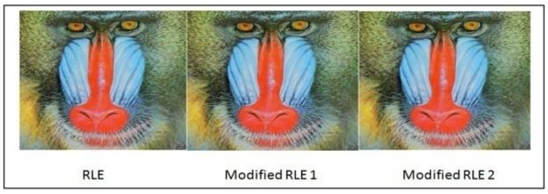

Figure 11. Mandrill

Figure 13. Birds

Figure 15. Lena

Figure 12. Pepper

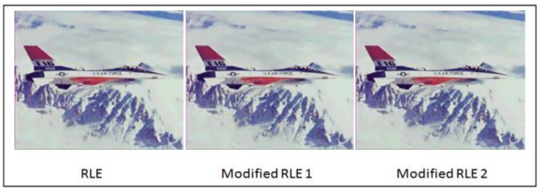

Figure 14. Airplane

Figure 16. Barbara

7. Conclusion

Modern applications require significant computing power for advanced media processing and complex data coding. In this paper we have introduce a new strategy for image compression that significantly provides greater performance without loss in image quality. The DCT technique has been used for image compression. This paper implements an efficient and effective algorithm for still image compression of relatively high signal to noise ratio. The proposed implemented technique considers that only zeros of the zigzag scanning is the repeated runs. This results in possibility of Zero byte of the RLE output elimination. Thus a higher compression ratio was achieved due to word-length reduction, the increasing in compression ratio in Proposed RLE 1 with respect to Traditional RLE with average value 17.88%, and the decreasing in PSNR with average value 0.046667 dB. The increasing in compression ratio in Proposed RLE 2 with respect to Traditional RLE with average value 17.75% at the same PSNR. We show experimentally that the proposed algorithms offer a competitive alternative to Traditional RLE without extra processing i.e. at the same system complexity which makes it compatible for SoC application**.**

References

[1] K. Sayood, "Introduction to data compression", Newnes, **(2012)**.
[2] D. Salomon and G. Motta, "Handbook of data compression", Springer Science & Business Media, **(2010)**.
[3] A. S. Mamatha and V. Singh, "Near lossless image compression system", Microelectronics and Electronics (PrimeAsia), Asia Pacific Conference on Postgraduate Research in. IEEE, **(2012)**.
[4] A. Rakshit and T. Pattnaik, "A Hybrid JPEG & JPEG 2000 Image Compression Scheme for Gray Images", International Journal of Scientific Engineering and Technology, vol. 1, **(2014)** June, pp. 771-774.
[5] M. Kasar and Y. D. Chincholkar, "JPEG Algorithm Implementation Using ARM Processor", International Journal of Advanced Research in Computer Science and Software Engineering, **(2013)** July.
[6] S. J. Pinto and J. P. Gawande, "Performance analysis of medical image compression techniques", Internet (AH-ICI), Third Asian Himalayas International Conference on. IEEE, **(2014)**.
[7] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete cosine transform", Computers, IEEE Transactions on, vol. 100, no. 1, **(1974)**, pp. 90-93.
[8] I. M. Pu, "Fundamental data compression", Butterworth-Heinemann, **(2005)**.

