The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM)
ISSN: 2602-3199

The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 2022

Volume 21, Pages 173-181

IConTES 2022: International Conference on Technology, Engineering and Science

Three States QRLE (Quantized Run Length Encoding) Based JPEG Image Compression Method

Sid Ahmed CHOUAKRI
University of Sidi Bel Abbes

Fatiha MESKINE
University of Sidi Bel Abbes

Abstract: We present in this work, an improved JPEG-based image compression method. The proposed method appeals a so-called QRLE (Quantized Run Length Encoding) technique where the couple of values ('zero' followed by its run number) is replaced just by one value. In this work, we substitute the Huffman encoding process by the QRLE method. The key idea of the QRLE method is to guarantee that the MSB bit of the binary representation of the non-null values is '0' while that of the 'zero' run number is '1'; this is done by adding the value 2^{N-1} to that number where N is its binary representation length. However, given that the zero run numbers could exceed (2^7-1≡127) value, provided that the RGB colored images are 8 bits long, the MSB is, necessarily, '1'. Even worse, the zero run number can be greater than 255 which implies occupying more than one octet. To solve that; the basic idea is to treat the zero run numbers of the AC zig-zag coefficients, according to 3 pre-defined ranges as follows: [1..127], [128..255], and [256..maximum. For the first range, '128' value is added to the zero run numbers; while for the second one, the zero run numbers are unchanged but preceded by '0' value. Finally, for the third range, the zero run numbers are kept unchanged while preceded by the number of octets of their binary representation. In terms of obtained results, comparing the proposed method with the classical JPEG, the reconstitution values remain the same for both technics. On the other hands, the compression ratio is considerably improved rising from 1:16.35 to 1:22.62 being provided an RGB image with 3 times a typical quantization matrix (for a quality of 50% as specified in the original JPEG Standard) while the PSNR is around 26.215both algorithms.

Keywords: JPEG, Run length encoding, Quantized, Compression ratio.

Introduction

Digital image compression is an important operation to save memory space when storing digital images or videos. Compression is used to reduce redundant data while maintaining the image's quality. There are two basic types of compression techniques: lossless, which allows the original image to be recovered without any loss, and lossy, which only allows for an approximate reconstruction of the original image. JPEG (Joint Photographic Experts Group) is the most widely used image compression standard since its introduction in 1992 (Wallace, 1992). It is a lossy compression method, and is widely used in digital cameras and mobile phones. The algorithm is designed to specifically discard the information that human eye cannot see easily. This can be done because slight change in the color are not perceived well by the human eye, while slight changes in the intensity (light and dark) can be easily detected. JPEG compression standard was evolved for compressing the color or gray scale still images like photographs, graphics and video stills (Firat & Fatih, 2018).

Through the years, many efforts have been made to further improve its quality and performance. Jin et al., (2008) have used a detection algorithm for zero quantized DCT coefficients. The experimental results show that the proposed algorithm can significantly reduce the redundant computations and speed up of the image encoding

- Selection and peer-review under responsibility of the Organizing Committee of the Conference

without causing any performance degradation. Bheshaj et al., (2012) introduced the performance evaluation of JPEG image compression using symbol reduction technique. A new technique has been proposed by combining the JPEG algorithm and Symbol Reduction Huffman technique for achieving more compression ratio. The result shows that the performance of standard JPEG method can be improved by the proposed method by achieving about 20% more compression ratio than the Standard JPEG. Sandhya & Vijay (2017) proposed a hybrid technique that combines DCT and fractal quad tree decomposition with Huffman encoding for color images. Candra et al., (2017) adopted a Zigzag Scan with Mapping method implemented in electronic circuit. It is able to accelerate the sorting process of DCT quantized coefficients period because the input data can be immediately located in sequence position which has been determined without any value comparison and repetition process. The efforts were invested also in creation of a quantization matrix that is more suited to the human visual system (Lamia et al., 2017) optimizing for a subjective quality improvement. Attempts were also made to adapt the quantization matrix to the content of the image block; both in the spatial and transformation domain (Poth et al., 2020).

JPEG Algorithm Steps

The basic block diagram for the JPEG image compression is shown in figure 1. There are four main steps in the process of JPEG compression (Firat et al., 2018).

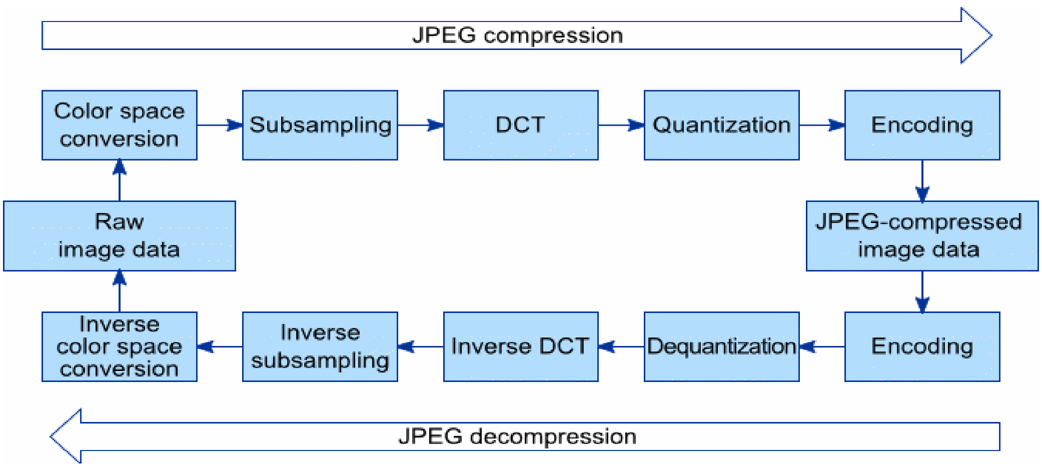

Figure 1. JPEG compression steps.

Color Transformation

RGB color format is used for easy for human to select colors form the color spaces. On the other side, YCrCb color format is useful for processing images. This operation converts the RGB image to YCbCr color space. Y is luminance commonly known as “luma” component and Cb and Cr are the blue-difference and red-difference respectively. These Y Cb Cr components can be calculated from RGB color components as given in following equations.

$$Y=0.299(R)+0.587(G)+0.114(B)$$
$$Cr=0.713(R-Y)=0.500(R)-0.419(G)-0.081(B)$$
$$Cb=0.564(B-Y)=-0.169(R)-0.331(G)+0.500(B)$$

Sub-Sampling

The human eye is less sensitive to brightness than to color. Therefore, reducing the Cr and Cb values by a certain amount will not change the image much for the human eye. Different formats used for sub-sampling are shown in the following figure.

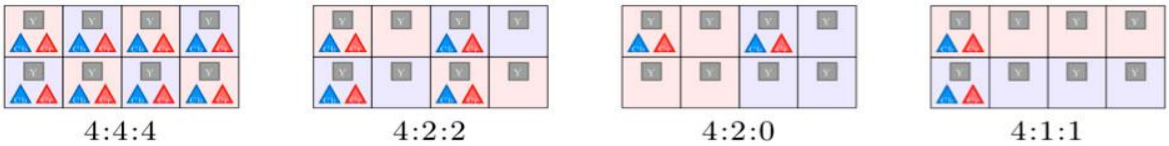

Figure 2. Formats of subsampling Y, Cb and Cr.

DCT Transform

The basic idea of data compression is to reduce the data correlation. By applying Discrete Cosine Transform (DCT), the data in time (spatial) domain can be transformed into frequency domain. Because of the less sensitivity of human vision in higher frequency, we can compress the image or video data by suppressing its high frequency components but do no change to our eye. The image data is divided up into 8x8 blocks of pixels and a DCT is applied to each block.

Forward DCT:

$$F(u,v)=\frac{1}{4}C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)\cos\left[\frac{\pi(2x+1)u}{16}\right]\cos\left[\frac{\pi(2y+1)v}{16}\right]$$

for $u=0,\ldots,7$ and $v=0,\ldots,7$

$$\text{where } C(k)=\begin{cases}1/\sqrt{2} & \text{for } k=0\\ 1 & \text{otherwise}\end{cases}$$

Where u and v reprsents the horizontal and vertical spatial frequency respectively. $f(x,y)$ is the pixel value at coordinates (x,y) and $f(u,v)$ is the DCT coefficient at coordinates (u,v).

Quantization

Quantization is the step where most of the compression takes place. DCT really does not compress the image because it is almost lossless. Quantization makes use of the fact that higher frequency components are less important than low frequency components. It allows varying levels of image compression and quality through selection of specific quantization matrices. Thus quality levels ranging from 1 to 100 can be selected, where 1 gives the poorest image quality and highest compression, while 100 gives the best quality and lowest compression. As a result quality to compression ratio can be selected to meet different needs. A quantization matrix is used in combination with a DCT coefficient matrix to carry out transformation. Quantization is achieved by dividing transformed image matrix by the quantization matrix used as depicted in this equation.

$$F_q(u,v)=round\left(\frac{f(u,v)}{Q(u,v)}\right)$$

Huffman and RLE Encoding

The quantization step performs the major and lossy part of the compression in the pEG algorithm. Here, the number of different coefficient values is reduced and the number of zero value coefficients is increased. After quantization, the quantized coefficient for the zero frequency in both dimensions (so-called DC coefficient) is encoded as the difference from the DC term of the previous block. Finally, all coefficients are ordered into a "zig-zag" sequence (see Fig.3). This ordering makes entropy coding easier by placing the low-frequency coefficients before the high-frequency coefffcients (Jussi & Tapani, 1995).

As final step entropy encoding is performed and the image is compressed. There are many entropy coding algorithms available where the most known are Huffman Coding, Arithmetic Coding, Run Length Coding (RLE).

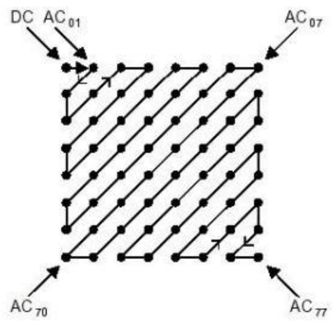

Figure 3. Zig-Zag reordering

Proposed Encoding Method

We present in this work, an improved JPEG-based image compression method. This is done by substituting, in the classical JPEG image compression algorithm, the Huffman encoding process by the QRLE method as illustrated by the following scheme:

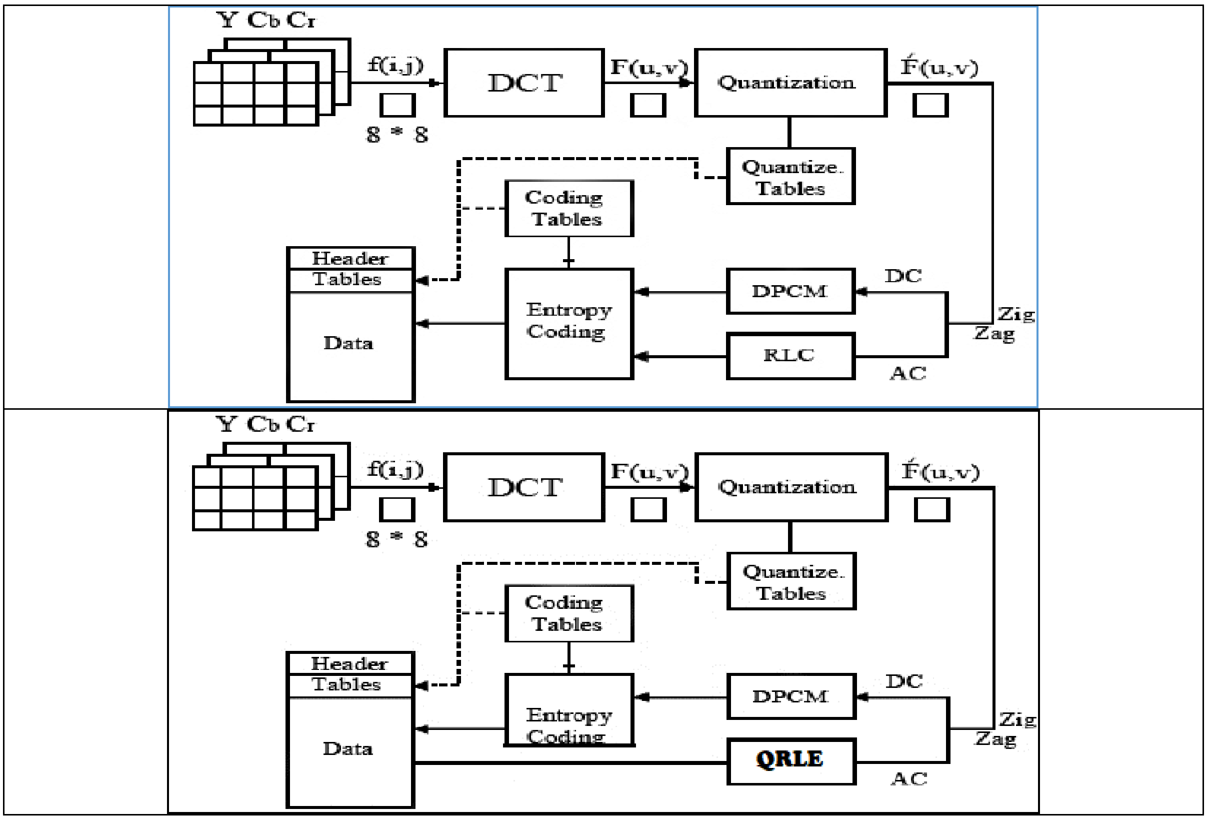

Figure 4. The classical jpeg-based huffman (shahbahrami et al., 2011) –at top- and the proposed modified jpeg-based 3 states qrle –at bottom- image compression synoptic scheme

It is necessary to mention that the original images are resized to [1536/R 1536/R] for computational reasons and will be recovered to their original sizes at the final step of the decompression process and where R is an integer value permitting better approaching to the original size of input images.
The proposed method appeals a so-called QRLE (Quantized Run Length Encoding) technique, that we have developed previously (Chouakri et al. 2018), where the couple of values (zero followed by its run number) is replaced just by one value.

The key idea of the QRLE method is to make sure that the MSB (the most significant bit) of the binary representation of the non-null values is '0' while that (the MSB) of the value indicating the 'zero' run number of zero is ‘1’. This is done by adding the value 2^{N-1} to that number where N is its binary representation length as illustrated by the following figure:

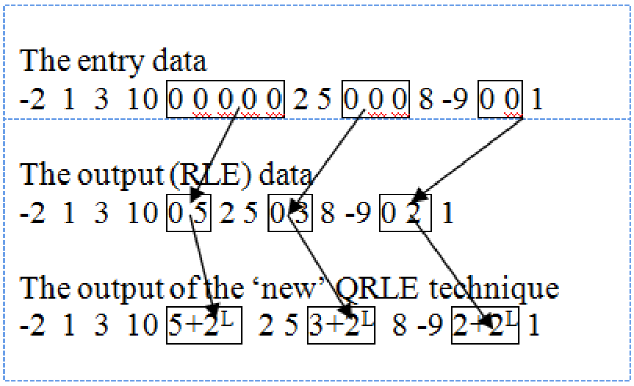

Figure 5. Quantized run-length encoding qrle example (Chouakri et al., 2018).

The first step in our QRLE method is to guarantee that the non-null values be less than 127, provided the RGB colored images are 8 bits long, that the MSB (the most significant bit) of the binary representation of these non-null values is ‘0’. This is done by accomplishing the following pseudocode:

Compute the minimum of non-null values ***min_non_null***;
Elevate the non-null values ***min_non_null*** to suppress the negative part ***Elev_non_null***;
Compute the maximum value of the elevated non-null values ***Max_Elev_non_null***;

$$\frac{Elev\_non\_null * ((2^{8}-1)-5)}{Max\_Elev\_non\_null}$$

Limit the elevated non-null values ***Limit_Elev_non_null*** given by:
Add a DC offset value of ***5*** to ***Limit_Elev_non_null*** to obtain ***Final_non_null***;

Adding a DC offset value of ***’(’***, to the limited elevated non-null values ***Limit_Elev_non_null*** is to avoid any interference with the byte-length of the huge zero run numbers; this value is sufficient to cover the highest possible amount of zero run number which is ‘16777216’ that is $2^{24} \equiv 2^{(8*3)}$. It is worthy to mention that to make sure that the final obtained non-null values ***Final_non_null*** will not exceed 127 ($\equiv 2^{7}$ -1), the DC offset value of ***’5’***, has been first subtracted from 127 in the 4th line of the pseudocode.

Coming back to the zero run number, the proposed algorithm is faced to the following problem situation: given that the zero run numbers exceeds, at least for once, (2^{7} -1≡127) value, the MSB is, necessarily, ‘1’. Even worse, the zero run number can be greater than 255 which implies occupying more than one octet. To solve that; the basic idea is to treat the zero run numbers of the AC zig-zag coefficients, according to 3 pre-defined ranges as follows:

[1..127], [128..255] , and [256..maximum value of zero run numbers]; where it comes the “3 states” expression. In the case of the first range, ‘128’ value is added to the zero run numbers, to force the MSB equals ‘1’, as given by the classical QRLE proposed previously (Chouakri et al., 2018) as well as figure 5. Fundamentally, as the size of the zero run numbers binary coded, in the second rage, i.e. [128..255], occupy strictly one byte with the MSB equals ‘1’, the zero run numbers are unchanged but preceded by ‘0’ value, to distinguish them from the former case.

Finally, for the third range, i. e. [256...maximum value of zero run numbers], and the zero run numbers are kept unchanged while preceded by the number of octets of their binary representation. Calculating the number of octets of the huge zero run numbers binary representation is carried out by applying the following Matlab macro command:

ceil(log10(zero_run_num+1)/log10(256))

The pseudocode of the Figure of the Three States QRLE (Quantized Run Length Encoding)-based JPEG image compression method is summarize as follows:

```
while not the end of RLE array

        while RLE coefficient ≠ 0
        Peek the RLE coefficient;
        Enqueue the RLE coefficient in QRLE array;
        Increment pointer values of RLE and QRLE arrays;
        end

        while RLE coefficient = 0
        Increment pointer value of RLE array;

        if RLE coefficient <128

        RLE coefficient ⟵ RLE coefficient +128;
        Increment pointer values of RLE and QRLE arrays;

        elseif    RLE coefficient ∊ [128,255]

        QRLE coefficient ⟵ 0;
        Increment pointer values of QRLE array;
        Peek the RLE coefficient;
        Enqueue the RLE coefficient in QRLE array;
        Increment pointer values of RLE and QRLE arrays;

        else

        QRLE coefficient ⟵ ceil(log10(zero_run_num+1)/log10(256));
        Increment pointer values of QRLE array;
        Peek the RLE coefficient;
        Enqueue the RLE coefficient in QRLE array;
        Increment pointer values of RLE and QRLE arrays;
        end

        end

end
```

Results and Discussion

This section presents and discusses the comparative analysis of our proposed algorithm performance versus the classical Huffman-based JPEG image compression. Two numerical criteria as well as visual inspection of the differential imge are employed for different algorithm's performance assessment. The numerical performance measurement criteria are the compression ratio (CR) and the Peak Signal to Noise Ratio (PSNR); they are given, respectively, by:

$$CR = \frac{numbre\ of\ samples\ of\ the\ original\ image}{number\ of\ samples\ of\ the\ compressed\ image}$$

and

$$PSNR = 10log_{10}(\frac{(255)^2}{MSE})$$

where

$$MSE = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\ \left[I_{org}(i,j) - I_{rec}(i,j)\right]^2}{\text{NxM}}$$

and I_{org} and I_{rec} are, respectively, the original and reconstructed images while N and M are the number of rows and columns of the image. Obviously, a large value of PSNR indicates the good quality of reconstruction of the compressed image. It is worthy to mention that the obtained results (in terms of CR and PSNR) are given being provided an RGB image with 3 times a typical quantization matrix (for a quality of 50% as specified in the original JPEG Standard).

The following table summarizes the comparative analysis of applying our QRLE vs. Huffman based JPEG image compression algorithms. The table lists, preliminary, the name as well as the type of the original images, their original size as well as resizing value. In terms of obtained results, the table depicts the compression ratio for our proposed algorithm vs; classical JPEG and the common value of PSNR.

Table 1. The assessment numerical values of our propose qrle vs. Huffman based-jpeg algorithm in terms of psnr and cr.

Name	Type	Original size	Resize value	PSNR	CR (our propose QRLE based-JPEG algorithm)	CR (classical JPEG algorithm)
ngc6543a	jpg	650 x 600	1536/2	38.7500	49.8780	14.0588
two	tif	3208x2672	1536	35.4824	82.2051	59.4265
peppers	png	384 512	1536/2	35.9946	17.2687	6.6498

The obtained results shows, obviously, the higher performance of our proposed JPEG-based QRLE algorithm compared to the classical one in terms of compression ratio (CR) that can reach in favorable cases till more than **82:1**. Conjointly, the restitution quality of the compressed image, which is the same for both algorithms, remains, in most of cases, good as demonstrated by the obtained numerical values. Moreover, the following figure illustrates the original images as well as their differential with their restituted versions. The visual inspection shows clearly the good restitution quality.

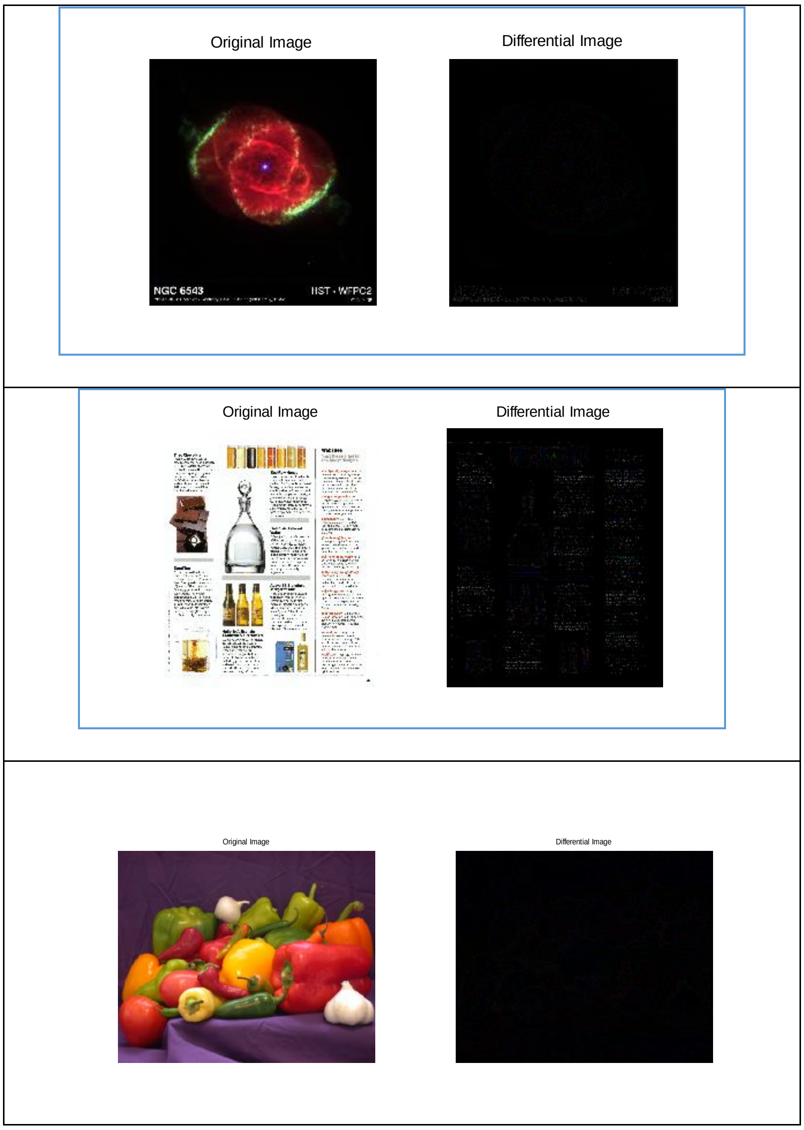

Figure 6. The visual inspection of the jpeg compression (psnr is similar for both algorithms: proposed jpeg-based qrle and huffman) in terms of original (at left) vs. differential image (at right) of images: ngc6543a.jpg, two.tif, and peppers.png (from top bottom).

Conclusion

We have presented in this work an new modified JPEG image compresssion algorithm based upon extending our anterior developed data compression method called QRLE (Quantized Run Length Encoding), firstly devloped uni-dimentional data, to the classical Huffman based-JPEG algorithm. The final work treated the processed pre-compressed image in 3 different ranges that comes the expression "Three states QRLE -based JPEG compression algorithm". This was done done by substituting the entropy Huffman coding by the QRLE technique. To assess the developed image compression algorithm performance, this latter has been applied to, mainly, 3 different types of coloured RGB images with distint sizes. Though the good and satisfactory compression results, in terms of compression ratio CR and Peak Signal to Noise Ratio PSNR as well as the visual inspection, provided by the classical JPEG-based Huffman coding, the developed JPEG-based QRLE image compression algorithm provides further performance as it was depicted along this paper. This considerably high image compression performance efficiency leads us to envisage to introduce the developed QRLE algorithm to video compression such as MPEG as future work.

Scientific Ethics Declaration

The authors declare that the scientific ethical and legal responsibility of this article published in EPSTEM journal belongs to the authors.

Acknowledgements or Notes

* This article was presented as an oral presentation at the International Conference on Technology, Engineering and Science (www.icontes.net) held in Antalya/Turkey on November 16-19, 2022.

* This work was supported by Directorate General for Scientific Research and Technological development (DGRSDT)

References

Alam, L., Dhar, P. K., Hasan, M. A. R., Bhuyan, M. G. S., & Daiyan, G. M. (2017). An improved JPEG image compression algorithm by modifying luminance quantization table. *International Journal of Computer Science and Network Security (IJCSNS), 17*(1), 200.

Artuger, F., & Ozkaynak, F. (2018). Performance comparison for lossy image compression algorithm. *International Journal of Advances in Electronics and Computer Science*, *5*(10), 57-61.

Candra, R., Madenda, S., Sudiro, S. A., & Subali, M. (2017). The Implementation of an efficient zigzag scan. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *9*(2), 95–98.

Chouakri S. A., & Taleb-Ahmed, (2018). Quantized run length encoding QRLE -new compression method-: Application to ECG transmission via IEEE802.11b WLAN Channel. *International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies* (3ICT), (pp. 1-8).

Kadam, S., & Rathod, V. J. (2017). DCT with Quad tree and Huffman coding for color ımages. *International Journal of Computer Applications*, *173*(9),33-37.

Kumar, B., Thakur, K., & Sinha, G. R. (2012). *Performance evaluation of JPEG image compression using symbol reduction technique.* Natarajan Meghanathan, et al. (Eds): ITCS, SIP, JSE-2012, CS & IT 04, 217–227.

Lammi, J., & Sarjakoski, T. (1995). lmage compression by the JPEG algorithm. *Photogrammetric Engineering & Remote Sensing*, (pp. 1261.-1266). October 1995

Li., Takala, J., Chen, M., Gabbouj, J., & Hexin. (2009). Detection algorithm for zero quantized DCT coefficients in jpeg. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 31 March-4th April 2008, USA.

Miklós, P., Trpovski, Z., & Lončar-Turukalo, T. (2020). Analysis and improvement of JPEG compression performance using custom quantization and block boundary classifications. *Acta Polytechnica Hungarica*, *17*(6),171-191.

Shahbahrami, A., Bahrampour, R., Rostami, M. S., & Mobarhan, M. A. (2011). Evaluation of Huffman and arithmetic algorithms for multimedia compression standards. arXiv preprint arXiv:1109.0216.

Wallace, G. K. (1992). The JPEG still picture compression standard. *In IEEE Transactions on Consumer Electronics, 38(1)*.

Author Information

Sid Ahmed CHOUAKRI
Telecommunications and Digital Signal Processing Laboratory (LTTNS), Electrical engineering Faculty
Djillali Liabes University of Sidi Bel Abbes,
22000, Algeria
Contact e-mail: *sa_chouakri@hotmail.com*

Fatiha MESKINE
Communication Networks, Architecture and Multimedia (RCAM) Laboratory, Electrical engineering Faculty
Djillali Liabes University of Sidi Bel Abbes,
22000, Algeria

To cite this article:

Chouakri, S.A. & Meskine, F. (2022). Three states qrle (quantized run length encoding) based jpeg image compression method. *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 21*, 173-181.