DEVELOPED RLE ALGORITHM AND BITPLANE SLICING TO COMPRESS GRAYSCALE IMAGE

H. K. ALBAHADILY
Aspirant in telecommunication and computer network, BSUIR

V. TSVIATKOU, PhD
Head the Department of the Telecommunication Systems, BSUIR

Belarusian State University of Informatics and Radio Electronics, Republic of Belarus
E-mail: hasan@bsuir.by

Abstract. New suggested RLE compression algorithm to compress grayscale images with bitplane slicing technique to reduce the size of the encoded data by separating image into 8 binary layers, then use our modified RLE algorithm to compress the bitplanes. Our modified algorithm designed perfectly to compress bitplane. The proposed method achieved very good compression ratio especially with the MSB layer.

Introduction. Data files frequently contain the same character repeated many times in a row or column. The digitized signals can also have runs of the same value, indicating that the signal is not changing, also images and music [1]. The Image can be considered as a two dimensional array of pixel intensities or can be considered as a discrete representation of data possessing both spatial (layout) and intensity (color) information [2].

There is significant redundancy present in image signals. This redundancy is proportional to the amount of correlation among the image data samples [3].

The goal of image compression is to represent an image signal with the smallest possible number of bits, thereby speeding up transmission, minimizing storage requirements,reduces the cost of data transmissionand reduces the errors of transmission.

Our method is implemented using MATLAB2012 on WINDOWS7 Operating System.

Bitplane slicing. The bitplane slicing is a fundamental technique of image processing in which the image is sliced into different planes(each layer contains sequences of only binary digits 0 or 1). It is ranges from plane1 which contains the least significant bit (LSB) to the last planeN which contains the most significant bit (MSB), where the number of layers depends on the bit depth of the image. The bit depth means how many bits need to represent the pixel's intensity. for example if the image is grayscale i.e. bitdepth is 8bit and it will be separated into 8 layers, or into 24 layers if the image is colored i.e. bitdepth is 24bit.

It is clear that the intensity value of each pixel can be represented by 8-bit binary vector (b8,b7, b6, b5, b4, b3, b2, b1) bk, where k is from 1 to 8 and each bk is either "0" or"1". In this case, an image may be considered as an overlay of eight bit-planes. Each bit-plane can be thought of as a two tone image and can be represented by a binary matrix [6][7].The formation of $bitplane_k$ is given by Equation below [4]:

$$BitPlane_K = Reminder\left\{\frac{1}{2} floor\left[\frac{1}{2^{K-1}}\ Image\right]\right\} \quad (1)$$

The bitplane decomposition is very useful for image compression. It allows some bi-level compression [5], i.e. it is used in some ways to compress images based on the idea of splitting the image into layers of binary values then either omit layers which are not highly effect on the image quality or by using the idea of similarity of elements in the bit plane which would be appears highly in the MSB layers, and by this way a long runs of similar values would result in very good compression rates [8][9]. Thus The RLE may be advantageously applied because the long runs in the bit planes which is the backbone of RLE [10]. This technique is very useful even if there are no repeated runs in the pixels, and by using bitplane slicing we will find some kind of repetition especially with last layer which contains MSB and achieving the highest compression ratio because it is contain frequently repeated runs.

The modified RLE algorithm (I3BN). The RLE method counts the values and their runs or repeated time as pairs of value and run (I,N), where I is the vector of values and N is the vector of repeats.

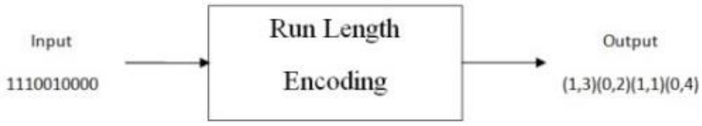


Fig. 1. Illustration of RLE for a binary input sequence

The modified RLE algorithm I3BN using the same idea of RLE which countsrepeated runs but instead of sending the values and runs as pairs we will send only the repeated runs by sending one bit or two or three followed by the number of repeated runs [11].

The algorithm I3BN using three symbol b1, b2 and b3, which takes 1 if the value I repeated and 0 if absent. The structure of coded data for algorithm I3BN can be represented by the following diagram:

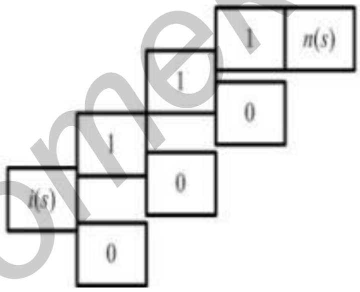


Fig. 2. Coded data structure of I3BN

So if the run repeated one time we will send the binary series 0, if the run repeated two times we will send the binary series 10, if the run repeated three times we will send 110 and if the run repeated more than three times we will send the sequence 111 followed by subtraction four from the run, for example if the value repeated 8 times, we will send 111(4), and we need to represent the run in binary and reserve number of bits to represent the new subtracted runby finding the binary logarithm which will be 3 bits so we will send 111 (100).

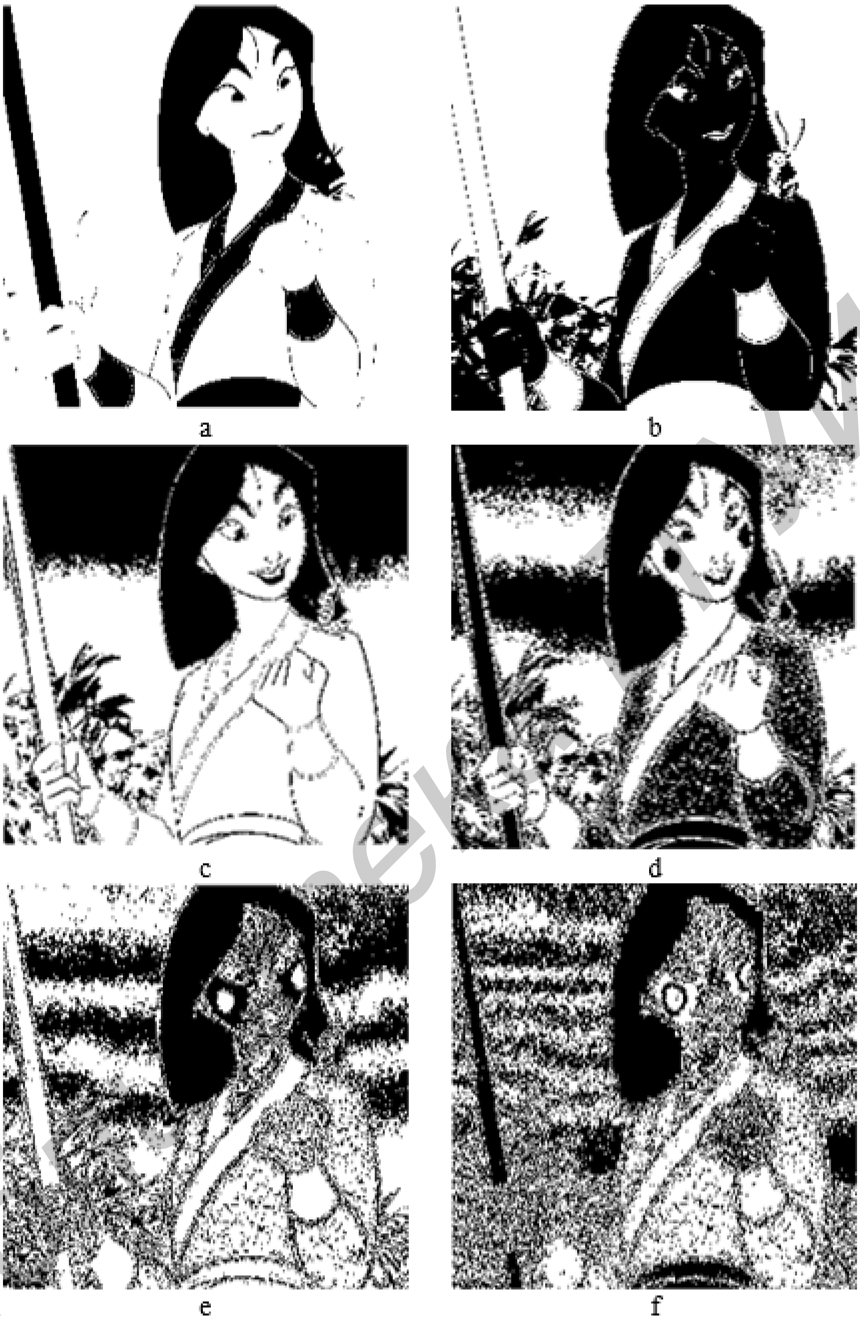
The code Size R_{I3BN} (bit) ofalgorithm I3BN defined by the expression:

$$R_{I3BN} = S(BD_I + 1) + \sum_{s=0}^{S-1} b1(s) + \sum_{s=0}^{S-1} b2(s) + BD_N \sum_{s=0}^{S-1} b3(s) \quad (2)$$

Where BD_I is the bit depth of the image and BD_N is the bitdepth of the maximum repeat.

The modified algorithm I3BN will work perfectly with bitplane by increasing the number of runs and decreasing the number of bits to represents runs.

Experiment implementation of bitplane slicing. First step is converting the images into bit-planes(each layer will be binary image contains 0,1 only)



a - bitplane8; b - bitplane7; c - bitplane6; d - bitplane5; e - bitplane4; f - bitplane3

Fig. 3. The bitplanes for test image

It is possible to remove some information from an image without any apparent change in its visual appearance because the first three bits does not contribute so much information in image formation.

The image can be stored with the information provided by bit4 to bit8 only. Thus number of bits per pixel can be reduced to 5 which save more storage space [2][4].

Fig. 4. The test image with their 5bit reduced image

We can see that there is no big change in the visual appearance of the images because the 3 LSB does not contribute big value.

Results and disscution. First part of experiment is compressing the original test images without bitplanes, which achieved not high compression ratio as we can see from the table below:

Table 1. Compression ratio without bitplane

Image	Img1	Img2	Img3
CR	1,082	1,053	0,928

From results above we can see that the proposed algorithm provides compression ratio up to 1,082-0,928 times for the original images without using bitplane technique.

The second part is splitting the images into 8 bitplanes and implementing the modified RLE algorithm I3BN on each plane, and we got the result in the table below:

Table 2. Compression ratio with bitplane

Image	Layer8	Layer7	Layer6	Layer5	Layer4	Layer3	Layer2	Layer1
Img1	3,134	2,294	1,669	1,039	0,863	0,765	0,681	0,638
Img2	3,551	3,221	2,354	1,853	1,213	0,744	0,638	0,608
Img3	2,206	1,385	1,116	0,913	0,752	0,638	0,616	0,605

From results above we can see that the proposed algorithm provides compression ratio up to 3,551-0,605 times for the MSB plane of the images.

The third part is implementing the modified algorithm on the images after reducing the three LSB and we got the results below:

Table 3. Compression ratio for reduced images

Image	Img1	Img2	Img3
CR	2,028	2,970	1,572

We can see that the modified algorithm has achieved very good ratio of compression up to 2,970-1,572 times without big change in the image visual details.

Conclusion and future work. The Results showing that the modified algorithm provides compression ratio for the original images without bitplane up to 1,082-0,928 times while the compression ratio was up to 3,551-0,605 times for bitplanes of images. The modified algorithms provide encoded image size reduction up to 2,970-1,572 times when removing the first three LSB bits.

For the future work it will be good to use the proposed algorithm with other compression methods like JPEG to get a high compression ratio in the lossy compression methods with keeping the image visual details as high as we can.

References

[1]. S. Smith The Scientist and Engineer's Guide to Digital Signal Processing 2nd Edition, California Technical Publishing 1999.

[2]. C. Solomon, T. Breckon Fundamentals of Digital Image Processing, A Practical Approach with Examples in Matlab , John Wiley Ltd. 2011.

[3]. A. Bovik The Essential Guide to image Processing, Elsevier Inc. USA 2009.

[4]. S. Halder [et al] A Low Space Bit-Plane Slicing Based Image Storage Method using Extended JPEG Format, International Journal of Emerging Technology and Advanced Engineering Vol 2, Issue 4, April 2012.

[5]. M. Luís O. Matos Lossy-to-Lossless Compression of Biomedical Images Based on Image Decomposition, Applications of Digital Signal Processing through Practical Approach, InTech.

[6]. R. Gonzalez, R. Woods Digital Image Processing 2nd ed., Prentice Hall, Inc., New Jercy 2008.

[7]. K. Ting, D. Bong, Y. Wang Performance Analysis of Single and Combined Bit-Planes Feature Extraction for Recognition in Face Expression Database, International Conference on Computer and Communication Engineering 2008, 2008 Kuala Lumpur, Malaysia.

[8]. M. Rangaraj Biomedical Image Analysis, University of Calgary, Canada, CRC Press LLC 2005.

[9]. M. Alasdair An Introduction to Digital Image Processing with Matlab, Victoria University of Technology, 2004.

[10]. A. BOVIK Hand Book Of Image And video Processing, Academic Press Austin USA 2000.

[11]. H. Albahadily [et al] New Modified RLE Algorithms to Compress Grayscale Images with Lossy and Lossless Compression, International Journal of Advanced Computer Science and Applications, Vol. 7, No. 7, 2016.