An Enhanced Run Length Coding for JPEG Image Compression

Amritpal Singh
Computer Science & Engineering Department
Thapar University, Patiala, Punjab

V.P. Singh
Computer Science & Engineering Department
Thapar University, Patiala, Punjab

ABSTRACT
Run length coding is the standard coding technique for compressing the images, especially when images are compressed by block transformation. 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. Then it was observed that for the occurrence of consecutive non-zero sequence the value of RUN is zero for most of the time, so this redundancy was removed by encoding the non zero coefficient (LEVEL) only, instead of an ordered pair of RUN (= 0)/ LEVEL. According to this scheme the single zero present between two non zero coefficients would be encoded as (1,0) .The proposed work aims at removing the unintended RUN, LEVEL (1,0) pair used for a single zero present between the two non-zero characters. So instead of using (1,0) pair for the zero between non-zero characters, a single '0' will be encoded. The proposed scheme has been tested on various images and results confirmed that the proposed run length scheme produces effective results.

Keywords
Image Compression; JPEG; entropy coding; run length coding;

1. INTRODUCTION
Image compression [1] is the application of data compression on digital images. JPEG is the image compression standard developed by the Joint Photographic Experts Group that is compatible with continuous tone still images both greyscale and colour [2,3]. JPEG is a flexible, general purpose compression standard which meets the need of all continuous-tone still image applications. It is a standard which enables the exchange of images across the application boundaries. The JPEG standard [4,5] specifies three elements in an encoder: discrete cosine transformer, quantizer (used in lossy compression) followed by entropy encoder as shown in Fig 1. This standard does the image compression without compromising with the quality of image [6].

2. THE JPEG COMPRESSION STANDARD

2.1 Discrete Cosine Transform
JPEG is a popular DCT-based [7] still image compression standard. DCT is a Fourier-related transform [8] which expresses a finite sequence of image data points in terms of a sum of cosine functions oscillating at different frequencies. The function of the transformation block in compression algorithm is to reduce the correlation between the pixels. The frame is first divided into blocks of 8x8 pixels each. The 2-dimensional 8x8 Forward Discrete Cosine Transform [9] converts the image in a digital format where the data representation is more compressible and can be computed by a machine. Among all DCT coefficients, the one at the top left is called DC coefficient, while the rest are AC coefficients shown in Fig 3. After transformation, most of the image energy will be concentrated at the upper left region of the block, and a lower right region of the block will have the frequencies with the amplitude of 0. The 8x8 2-Dimentional DCT is

$$F(u,v) =$$

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

for u=0,....,7 and v=0,….7

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

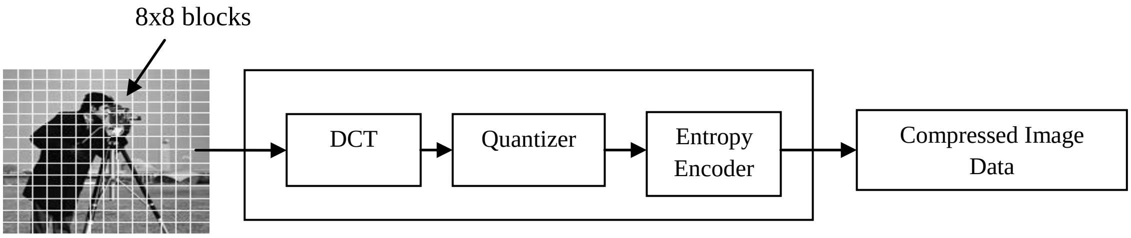

Fig 1: The JPEG 8×8 Block Encoding Process

2.2 Quantization

After the transformation from the DCT, each of the 64 DCT coefficients is uniformly quantized [10] with a 64-element Quantization Table, which must be specified by the application (or user) as an input. The JPEG recommended luminance quantization array [11] is given in Fig.2 and it can be scaled to provide variety of compression levels. The goal of this processing step is to discard Information which is not visually significant. This phase of JPEG compression method makes it a lossy compression method [12,13], as the truncation of the frequencies is done during quantization phase which is not recoverable while reconstructing the original data of the image. So in this step, each DCT coefficient is divided by its corresponding quantizer step size in the Quantization Matrix Q(u,v) shown in Fig.2 , followed by rounding to the nearest integer.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Fig 2: ***Quantization Matrix Q(u, v)*, step size matrix applied to luminance**

$$F_q(u, v) = \text{Round}\frac{F(u, v)}{Q(u, v)}$$

2.3 Entropy Coding

This step achieves additional compression in a lossless manner by encoding the quantized DCT coefficients more compactly based on their statistical characteristics. It is useful to consider entropy coding as a 2-step process. The first step converts the zigzag sequence of quantized coefficients into an intermediate sequence of symbols as shown in Fig 3. The second step converts the symbols to a data stream [14] in which the symbols no longer have externally identifiable boundaries.

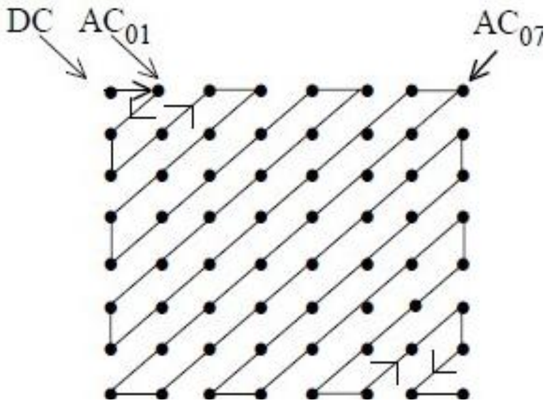

Fig 3: ***Zigzag Sequence***

All of the quantized coefficients are ordered into the “zigzag” sequence to form a 1-D sequence of quantized coefficients. This ordering helps to facilitate entropy coding by placing low-frequency coefficients (which are more likely to be nonzero) before high-frequency coefficients. Then this sequence of AC coefficients for each block is encoded using run length encoding which is designed to take advantage of the long runs of zeros that normally results from the reordering.

3. RUN LENGTH CODING

The Run length coding is the standard coding technique for block transforms based image/video compression. Run Length Coding is flexible in nature and therefore many authors have altered it for its suitable use in various applications [15, 16].

3.1 Conventional Run Length Coding

Run length is the number of consecutive zero-valued AC coefficients in the zigzag sequence present before the non-zero AC coefficient. 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. When the last (63rd) AC coefficient is encountered, a special sequence of (0,0) means End of Block is appended. When a sequence of non-zero coefficients is encountered it adds redundancy in the encoded data, as for the occurrence of consecutive non-zero sequence the value of RUN is zero for most of the time. So the Conventional Run Length Coding scheme encodes the redundant data, when it was meant to compress the original one. According to this algorithm the output (33 digit sequence) of the 8x8 image block in Fig.4 using Conventional Run Length Coding would be:

102	-33	-3	-4	-2	-1	0	0
21	-2	-3	0	-1	0	0	0
-3	0	1	0	0	0	0	0
2	0	0	0	0	0	0	0
1	0	0	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

Fig 4: ***8x8 image block after Quantization Phase***

Output: [(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)].

3.2 Optimized Run Length Coding

The Optimized Run Length Coding [17] uses a pair of (RUN, LEVELS) only when a pattern of consecutive zeros occur at the input of the encoder. The non-zero digits are encoded as their respective values in LEVELS parameter. The FLAG bit is used to identify the presence of a zero or consecutively occurring zeros in the input data stream. This way it represents the single zero between non-zero characters with a 2-digit sequence of (1, 0) and adds an extra digit (for the zero between non-zero characters) to the final encoded data. So the output (29 digit sequence) of the block under consideration would be:
Output: [-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)].

3.3 Proposed Run Length Coding Scheme

Further optimization can be done by removing the unintended RUN, LEVEL (1, 0) pair used to represent a single zero present between the non-zero characters. So instead of using (1, 0) pair for the zero between non-zero characters, single '0' is encoded. Fig. 5 shows the flow diagram of this scheme. The output (26 digit sequence) of the block under consideration using this scheme would be:

Output: [-33 21 -3 -2 -3 -4 -3 **0** 2 1 **0** 1 **0** -2 -1 -1 (3, 0) -2 (11, 0) 1 (31, 0) (0, 0)].

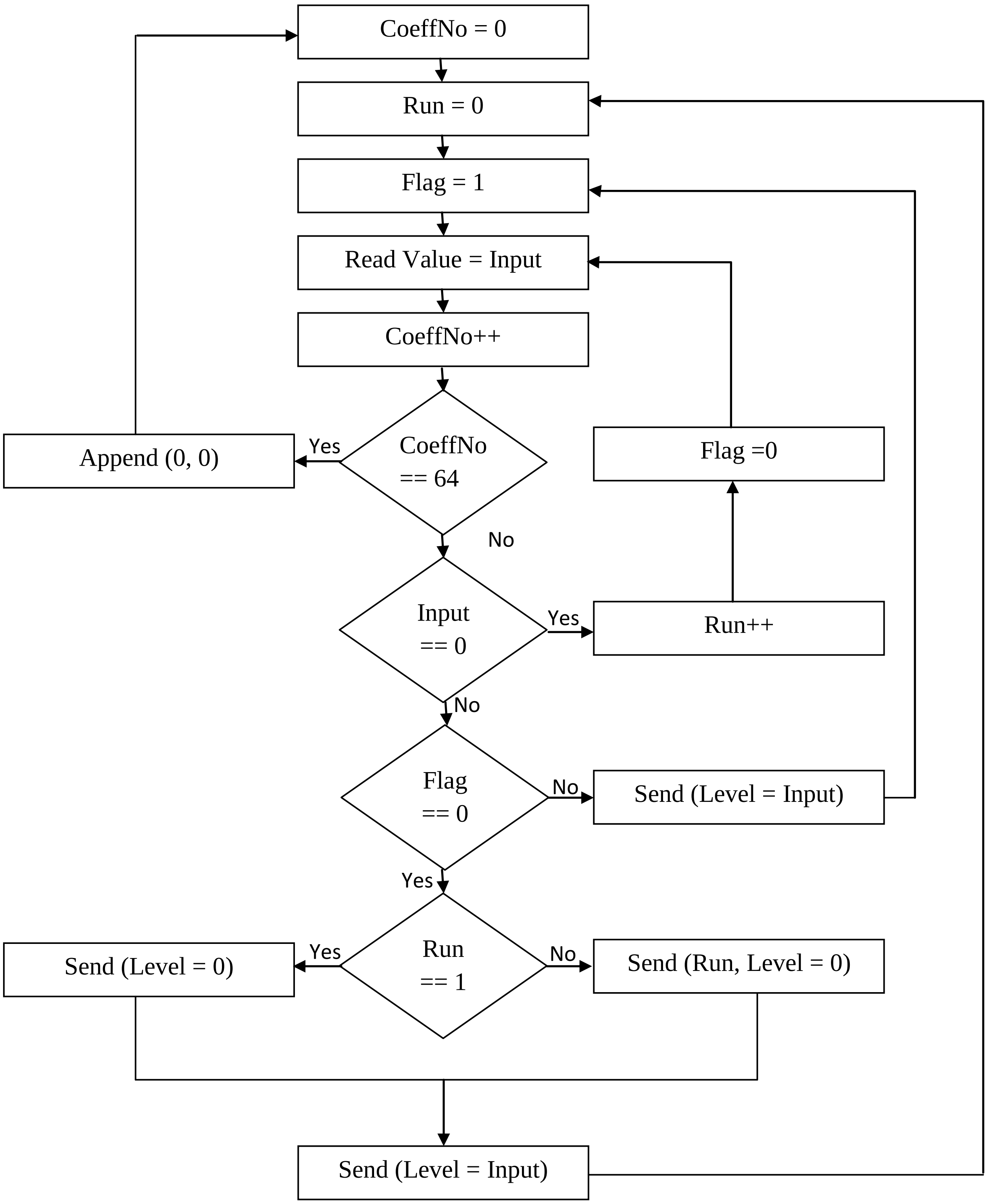

Fig 5: Flow Diagram of the Proposed Run Length Coding Scheme

4. RESULTS

The results show an efficient compression using the new Run Length Coding scheme applied on all type of images. The results of various algorithms applied on the image block taken as an example in Fig.4, are stated in Table 1.

Table 1. Results of the Image block under consideration

S. No	Algorithm Used	Digits used in Encoding Sequence
1	Conventional RLC	33
2	Optimized RLC	29
3	Proposed RLC	26

So, clearly the newly proposed algorithm is much more efficient. And moreover Table 1 shows the result of only one 8x8 image block, there are many 8x8 blocks in which the image is divided at the DCT phase of JPEG compression, and on an average 1 to 2 such single zeros present between two non-zero characters can be encountered. For evaluating the efficiency of the JPEG compression algorithm various metrics have been devised. Most commonly used are the Mean Square Error [18] and the Compression Ratio that is the ratio between the size of compressed data and the size of uncompressed data.

$$\text{Compression Ratio} = \frac{\text{Size of Compressed data}}{\text{Size of Uncompressed data}}$$

When the compressed data size is less than that of the original data then the algorithm is considered to be efficiently effective. The proposed scheme effectively follows these metrics as required.

Fig 6: **Original Test Image**

Fig 7: **Test Image restored from Proposed Run Length Coding Scheme**

5. REFERENCES

[1] T. Cebrail, S.K. Sarikoz, "An overview of Image Compression Approaches", The 3rd International conference on Digital Telecommunications, IEEE 2008.

[2] R C. Gonzalez and R E. Woods, Digital Image Processing. Prentice-Hall, 2008.

[3] Jain A. K. - "Fundamentals of Digital Image Processing", Prentice Hall, 1989.

[4] G. K. Wallace, 'The JPEG Still Picture Compression Standard', Communications of the ACM, Vol. 34, Issue 4, pp.30-44.

[5] W.B. Pennebaker, J.L. Mitchell, "JPEG still image data compression standard", Van Nostrand Reinland, New York, 1993.

[6] Z. Wang, G. Wu, H.R.Sheikh, E.P. Simoncelli, E.H. Yang, A.C. Bovik," Quality aware images" IEEE Trans. Image Processing, vol. 15, pp. 1680-1689, June 2006.

[7] N. Ahmed, T. Natarajan, and K. R. Rao. "Discrete cosine transform," IEEE Trans. Comput., vol. C-23, pp. 90-93, Jan. 1974.

[8] Beaudoin, Normand," An Accurate Discrete Fourier Transform for image processing", Pattern Recognition, Vol.3, pp. 935-939, 2002.

[9] R.C. Reininger, J.D. Gibson, "Distribution of Two Dimensional DCT Coefficients for Images" IEEE Trans. Comm. Vol. 31, pp.835-839, June, 1983.

[10] H.A. Peterson, H. Peng. J.H. Morgan, W.B. Pennebaker, "Quantization of color image components in the DCT domain", Human Vision, Visual Processing, and Digital Display. Proc. SPIE, pp. 210-222, 1991.

[11] A.J. Ahumada Jr., H.A. Peterson, "Luminance Model-Based DCT Quantization for color Image Compression" Human Vision, Visual Processing, and Digital Display III, B.E. Rogowitz, ed., SPIE, 1992.

[12] N. Ponomarenko, V. Lukin, K. Egiazarian, E. Delpc," Comparison of Lossy Compression performance on Natural color Images", IEEE 2009.

[13] Wai C. Chu, "On Lossless and Lossy Compression of Step Size Matrices in Jpeg Coding" IEEE 2013.

[14] O. Johnson, "On the Redundancy of Binary Huffman Codes" IEEE Trans. Inform. Theory, Vol. IT-26, No-2, pp.220-223, Mar. 1980.

[15] C. Tu, J. Liang, and T. D. Tran, "Adaptive run-length coding," IEEE Signal Processing Letters, vol. 10, pp. 61-64, Mar. 2003.

[16] E. H Yang and L. Wang, "Joint optimization of run-length coding, Huffman coding and quantization table with complete baseline JPEG decoder compatibility," in Proc. ICIP 2007, vol. 3, pp. 181-184, Sept. 2007.

[17] Akhtar M.B., Qureshi A.M., Qamar-ul-Islam, "Optimized Run length coding for jpeg Image Compression used in Space Research Program of IST", IEEE 2011.

[18] Zhou Wang and Alan C. Bovik, "Mean squared error, Love it or leave it?", IEEE Signal Processing Magazine Jan 2009.