

SQUASHED JPEG IMAGE COMPRESSION VIA SPARSE MATRIX

Shaista Qadir

Department of Computer Science, King Khalid University, Abha, Saudi Arabia

ABSTRACT

To store and transmit digital images in least memory space and bandwidth image compression is needed. Image compression refers to the process of minimizing the image size by removing redundant data bits in a manner that quality of an image should not be degrade. Hence image compression reduces quantity of the image size without reducing its quality. In this paper it is being attempted to enhance the basic JPEG compression by reducing image size. The proposed technique is about amendment of the conventional run length coding for JPEG (Joint Photographic Experts Group) image compression by using the concept of sparse matrix. In this algorithm, the redundant data has been completely eliminated and hence leaving the quality of an image unaltered. The JPEG standard document specifies three steps: Discrete cosine transform, Quantization followed by Entropy coding. The proposed work aims at the enhancement of the third step which is Entropy coding.

Keywords

Entropy coding, JPEG image compression, Compaction using sparse, optimized/reduced run length coding.

1. INTRODUCTION

Image compression is the removal of redundant data bits of digital images to reduce the actual image size. It is the process of encoding information in fewer bits than their original representation. It is simply a technique of applying compression on digital images to reduce the size of actual image in order to transfer it easily in least possible memory size. Digital images commonly contain lots of redundant data, these images need to be compressed to remove redundancy and minimize the storage space and transport bandwidth. Instead of keeping track of runs of redundant values as in conventional RLC, the proposed technique keeps track of the exact location of the non-zero element in the zig-zag matrix and the value itself (concept of sparse matrix). Further the technique keeps track of only first element in the sub-string if the non-zero elements are stored at consecutive location and rest of the elements are assumed to be in continues locations. As the non-zero elements are left in the quantized matrix so we keep track of the VALUE and its LOCATION using the zig-zag sequence used to read DTC coefficients.

The proposed modification in the JPEG image compression algorithm has been tested on various JPEG images under consideration in matlab. Results have proved the efficiency of the proposed algorithm for all the images used for testing.

2. RELATED WORK

Numbers of researches have been carried out to work upon the image compression and most of these are using the concept of conventional run length coding scheme and some have Optimized the other blocks of the compression technique [1]. Work on international standards for image compression started in the late 1970s with the CCITT (currently ITU-T) need to standardize

binary image compression algorithms for Group 3 facsimile communications [2],[3],[4]. Some authors have modified the entropy encoding part by modifying the run length coding for space research program of IST [5] , Other technique [6] had modified version of image compression/decompression algorithm using block optimization and byte compression method (BOBC). After BOBC, it is followed by run-length encoding and its block is optimized by varying block size. The Experimental results show that compression ratio of this algorithm is better than the previous BOBC algorithm and JPEG compression techniques. Image quality (PSNR) is almost the same or better as compared to that of the above mentioned compression techniques.

Although the proposed algorithm is no exception but aims towards the better or even equal compression rates (in worst case) as ever offered by the other algorithms by further modifying the conventional [7] and optimized run length coding [8].

3. JPEG IMAGE COMPRESSION

JPEG Compression is an image compression algorithm, developed by Joint Photographic Expert group which is used to curtail the file size of photographic images. Fig 1 shows the main procedures for all encoding processes based on the DCT.

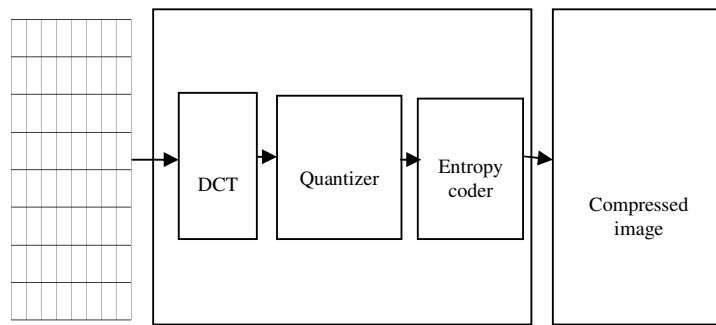


Figure 1. DCT-based encoder simplified diagram

The standard JPEG specifies the following three steps:

- Discrete cosine transform
- Quantization
- Entropy coding

3.1. Discrete cosine transform

Discrete Cosine Transform (DCT) exploits cosine functions, it transform a signal from spatial representation into frequency domain [9]. In the encoding process the input component's samples are grouped into 8x8 blocks, and each block is transformed by the DCT into a set of 64 values referred to as DCT coefficients. One of these values is referred to as the DC coefficient and the other 63 as the AC coefficients. DCT actually transforms image data from temporal to spatial domain. With JPEG image compression fourier-based DCT transformation is employed which AIMS at reduction of correlation between the pixels [10], [11]. The 8x8 2-Dimensional DCT is

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

For $u = 0, 1, \dots, 7$ & $v = 0, 1, \dots, 7$

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

3.2. Quantization

Each of the 64 coefficients is then quantized by simply dividing each Component in the frequency domain by a constant for that component as shown in quantization Matrix $Q(x,y)$ (Fig 2), and then rounding to the nearest integer. The image reformed later at the receivers end can't bring back and hence given the name loosy.

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

Figure 2. Quantization Matrix $Q(X,Y)$

3.3. Entropy coding

The DC coefficient and the 63 AC Coefficients are prepared for entropy encoding. The previous quantized DC coefficient is used to predict the current quantized DC coefficient, and the difference is encoded. The 63 quantized AC coefficients undergo no such differential encoding, but are converted into a one-dimensional zig-zag sequence Fig 3. The quantized coefficients are then passed to any of the entropy encoding procedures for image compression such as run length coding, arithmetic coding or Huffman coding.

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Figure 3. Zig-Zag sequence.

4. RUN LENGTH CODING

Run length coding is a lossless data compression technique in which runs of data are stored as a single data value and count rather than as original run. This coding technique is very useful with the data representation which contains numerous redundant runs. For such redundant data by employing run length coding we can represent an image in very fewer bits. It is not useful with files that don't have many runs as it could greatly increase the file size.

Run length is the number of consecutive zero-valued AC coefficients in the zig-zag sequence present before the non-zero AC coefficient.[8] 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 countered 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.

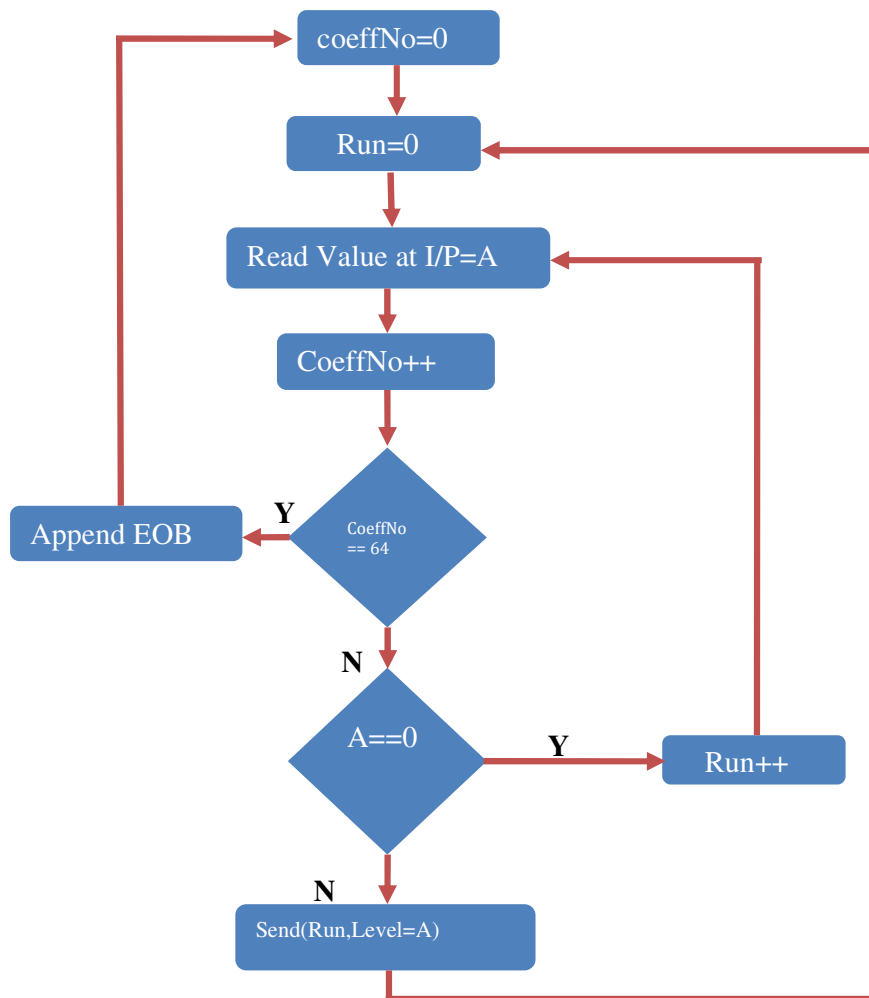


Figure 4. Flow diagram for original run length encoding[3]

According to the original run length coding algorithm the output (32 digit sequence) of the 8x8 image block in Fig.5 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

Figure 5. 8x8 Image block after Quantization Phase

Output Fig 5 using conventional RLC:

[(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)].
 {32 digits after compression}

According to the original run length coding algorithm the output (36 digit sequence) of the 8x8 image block in Fig.6 using Conventional Run Length Coding would be:

72	15	-5	8	12	-2	0	0
9	6	-1	0	4	0	0	0
-1	5	3	0	0	0	0	0
7	0	0	8	6	7	0	0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 6. 8x8 image block after Quantization Phase

Output Fig 6 using conventional RLC:

[(0,15)(0,9)(0,-1)(0,6)(0,-5)(0,8)(0,-1)(0,5)(0,7)(2,3)
 (1,12)(0,-2)(0,4)(7,8)(6,6)(6,1)(1,7)(0,0)].
 {36 digits after compression}

According to the original run length coding algorithm the output (36 digit sequence) of the 8x8 image block in Fig.7 using Conventional Run Length Coding would be:

98	-13	0	10	0	0	0	0
1	7	-6	0	0	0	0	0
8	15	0	0	0	0	0	0
2	0	0	0	0	0	0	18
17	0	0	0	0	0	7	0
-2	0	0	0	0	-8	0	0
1	0	0	0	-2	0	0	0
0	0	-5	-1	0	0	0	0

Figure 7. 8x8 image block after Quantization Phase

Output Fig 7 using conventional RLC:

[(0,-13)(0,1)(0,8)(0,7)(1,10)(0,-6)(0,15)(0,2)(0,17)(9,-2)
 (0,1)(15,-5)(0,1)(0,-2)(0,-8)(0,7)(0,18)(0,0)].
 {36 digits after compression}

5. PROPOSED COMPACT CODING

The Flow Diagram for the proposed compact coding is shown in Fig 8. Instead of keeping track of runs and Levels as in conventional run length coding the proposed coding represents the strings of LOCATIONS (Loc as in Fig 8) and the VALUES (AcCoff As in Fig 8) of non-zero Ac coefficients present in zig-zag matrix. This technique will help in further compression of image size or it may yield equal sized result in worst case. The proposed technique is a slightly modified run length coding. As conventional run length coding keeps track of both runs of zeros and ac coefficients, but the proposed technique will track ac coefficient and its location only. If ac coefficients are placed at consecutive locations in Quantized matrix, there is no need to store locations for all the consecutive coefficients in that particular sub-string, only the location of first coefficient need to be stored rest can be determined. In that case the proposed technique can help in more compaction.

This modification allows removing the extra parameter from the run length coded messages that utilized an extra memory space. This modification has been made after studying many image samples and making the following observations,

- AC coefficients are placed at consecutive locations i.e., one after another.
- The coefficients are stored in sparse matrix where non-zero entries are very fewer.
- There is no need to keep track of non-zero entries.

Applying proposed compact coding to the same matrices (Figures 5,6,7) we can get far much better results as compared to conventional run length coding, as shown below:

Output of Fig 5 using proposed RLC :

[(1, -33 -> 21 -> -3 -> -2 -> -3 -> -4 -> -3) (9, 2 -> 1) (12, 1) (14, -2 -> -1 -> -1) (20, -2) (32, 1)(0)]
 {22 digits used after compression}

Output of Fig 6 using proposed RLC:

$[(\vec{1}, 15 \rightarrow 9 \rightarrow -1 \rightarrow 6 \rightarrow -5 \rightarrow 8 \rightarrow -1 \rightarrow 5 \rightarrow 7) (\vec{12}, 3) (\vec{14}, 12 \rightarrow -2 \rightarrow 4) (\vec{24}, 8) (\vec{3}, 6) (\vec{38}, 1) (\vec{40}, 7)(0)]$
 {25 digits used after compression}

Output of Fig 7 using proposed RLC:

$[(\vec{1}, -13 \rightarrow 1 \rightarrow 8 \rightarrow 7) (\vec{6}, 10 \rightarrow -6 \rightarrow 15 \rightarrow 2 \rightarrow 17) (\vec{20}, -2 \rightarrow 1) (\vec{47}, -5 \rightarrow 1 \rightarrow -2 \rightarrow -8 \rightarrow 7 \rightarrow 18) (0)]$
 {22 digits used after compression}

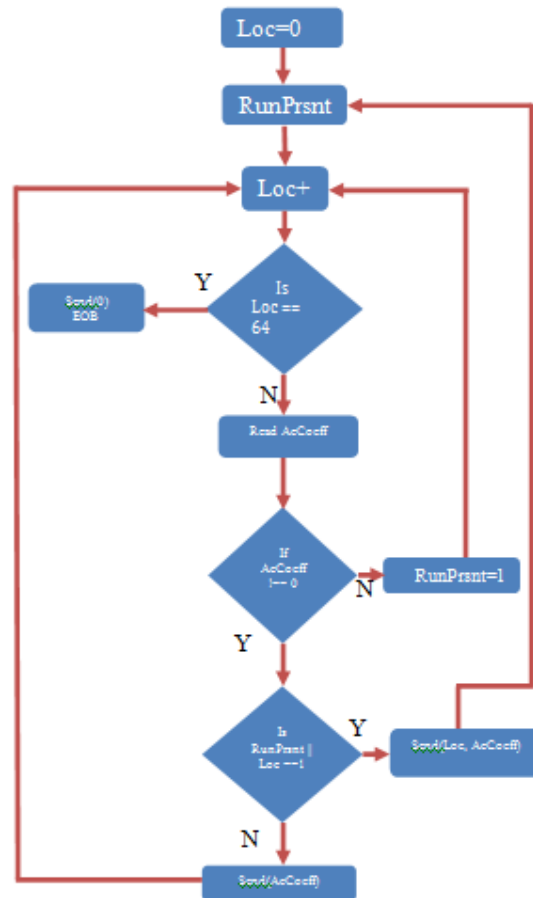


Figure 8. Flow diagram for proposed compact coding.

6. METHOD

In output (proposed compact coding) for Fig 5 LOCATION in the sub-strings is represented as an element with arrow head and Ac coefficient is represented as an element without any arrow head. Starting from left to right as shown in the output for the sub-string

$(\vec{1}, -33 \rightarrow 21 \rightarrow -3 \rightarrow -2 \rightarrow -3 \rightarrow -4 \rightarrow -3)$

At location $\vec{1}$ in the zig-zag sequence matrix Ac coefficient -33 is stored. As you can see Next element 21 in the sub-string is not any location but Ac coefficient again which simply means the

Ac coefficient 21 is stored in the next consecutive location (there is no 0 element in between -33 and 21) and likewise -3 is at the next consecutive location followed by -2 then -3 , -4 and -3 . As it is clear sub string ends at Ac coefficient -3 which can be determined to be at LOCATION $\vec{7}$.

In the next sub-string from the output of fig 5 i.e,

$$(\vec{9}, 2 \rightarrow 1)$$

At LOCATION $\vec{9}$ Ac coefficient 2 is stored and at the very next LOCATION ($\vec{10}$ th LOCATION) there is Ac coefficient 1 stored . As we don't need to mention consecutive locations in output so if the next element in the sub string is an element without arrowhead it's understood the element is an Ac Coefficient and not LOCATION.

Likewise in the next sub-string from the same output Fig 5

$$(\vec{12}, 1)$$

At LOCATION $\vec{12}$ Ac coefficient 1 is stored.

The next sub-string in the output is

$$(\vec{14}, -2 \rightarrow -1 \rightarrow -1)$$

At LOCATIONS $\vec{14}$, $\vec{15}$ and $\vec{16}$ Ac coefficients -2,-1 and -1 are stored respectively but the sub string must show the first LOCATION only as the other 2 LOCATIONS ($\vec{15}$ and $\vec{16}$) are consecutive so no need to mention in the string, hence reducing the number of digits in the output.

We will continue in the same manner in the next sub-strings

$$(\vec{20}, -2) \ \& \ (\vec{32}, 1)$$

At LOCATION $\vec{20}$ Ac coefficient -2 and at $\vec{32}$ Ac coefficient 1 is stored.

The sub-string (0) represents the end of the string.

In between the sub-strings there are Runs of 0's present

($\vec{1}, -33 \rightarrow 21 \rightarrow -3 \rightarrow -2 \rightarrow -3 \rightarrow -4 \rightarrow -3$) Runs of one 0 present at LOCATION 8 ($\vec{9}, 2 \rightarrow 1$) Runs of one 0 present at LOCATION 11 ($\vec{12}, 1$) Runs of one 0 present at LOCATION 13 ($\vec{14}, -2 \rightarrow -1 \rightarrow -1$) Runs of three 0's present at LOCATIONS 17, 18 & 19 ($\vec{20} \rightarrow -2$) Runs of eleven 0's present from LOCATIONS 21 to 31 ($\vec{32} \rightarrow 1$) Runs of thirty-one 0's present from LOCATION 33 to 64 (0)].

7. RESULTS

Proposed algorithm when implemented turned out to generate better compressed image size as compared to conventional RLC. The proposed algorithm has been tested and used upon various images using matlab. The result of different image blocks under consideration after compression is shown in Table1. And Fig 9 Shows the Chart of results of different image blocks under consideration.

$$Efficiency = \frac{Size_{original} - Size_{optimal}}{Size_{original}} \dots\dots\dots (2)$$

S. No	Algorithm Used	Digits used in Encoding Sequence		
		Fig 5	Fig 6	Fig 7
1	Conventional RLC	32	36	36
2	Proposed RLC	22	25	22
3	Efficiency %	31.25	30.55	38.89

Table 1. Results of different image blocks under consideration

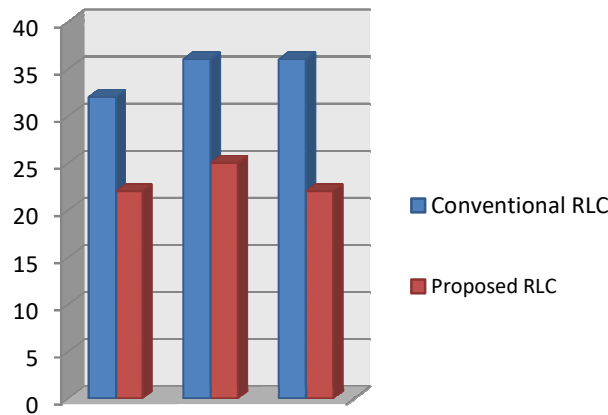


Figure 9. Chart of results of different image blocks under consideration

8. CONCLUSION

In this work, we have proposed an amendment to the conventional run length coding. The proposed technique is based on encoding runs of non-zero values only as rest of the runs are 0s. Based on the experimental results, it has been demonstrated that the proposed technique outperformed the conventional run length coding. Further the proposed technique has resulted in significant increase in the test data compression ratio for all considered test cases and improving the compression ratio from 31% to 39% for sample images. Thus the proposed technique has the advantage of enormously reducing image size. This is evident by the reduction in the total number of encoded runs.

AUTHORS

Shaista Qadir is Lecturer at King Khalid University , Abha, Kingdom Of Saudi Arabia and MCA from Jamia Humdard University, New Delhi. She received her PGDCA from National Institute of Electronics & Information Technology (NIELIT) .Her research interests include image processing, big data, cloud computing and Data Warehouse.



REFERENCES

- [1] C.Taskin and S.K. Sarikoz. : An Overview of Image Compression Approaches. [Remote Sensing and GIS Data Processing and Other Applications](#). IEEE Conference Publications , Page(s): 174 – 179, 2008
- [2] A.M.Raid ,W.M.Khedr , M. A. El-dosuky and WesamAhmed. : Jpeg Image Compression

- Using Discrete Cosine Transform. - A Survey International Journal of Computer Science & Engineering Survey (IJCSES) Vol.5, No.2, April 2014
- [3] Walaa M. Abd-Elhafiez. : New Approach for Color Image Compression. International Journal of Computer Science and Telecommunications. Volume 3, Issue 4, April 2012
- [4] Walaa M. Abd-Elhafiez, WajebGharibi. : Color Image Compression Algorithm Based on the DCT Blocks. IJCSE Volume-4, Issue-4 , Page no. 34-38, Apr-2016
- [5] M.B Akhtar, , A.M Qureshi, , Qamar-ul-Islam . :Optimized run length coding for jpeg image compression used in space research program of IST. International Conference on Computer Networks and Information Technology, IEEE Conference Publications, Page(s): 81 – 85, 2011
- [6] A. Banerjee and A. Halder. : An Efficient Dynamic Image Compression Algorithm based on Block Optimization,Byte Compression and Run-Length Encoding along Y-axis. , published in (ICCSIT), 3rd IEEE International Conference on (Volume:8), 2010.
- [7] A. Gupta, M.C. Srivastava, S.D. Pandey and V. Bhandari, . :Modified Runlength Coding for Improved JPEG Performance. IEEE Conference Publications, Page(s): 235 – 237, 2007.
- [8] A. Singh and V P Singh. . : An Enhanced Run Length Coding for JPEG Image Compression. Volume 72 - Number 20, IJCA Journal, 2013
- [9] S.P. Bagal and V.B. Raskar . : JPEG Image Compression by Using DCT. Research Paper, pages(34-38), Volume-4 , Issue-4 , 2016.
- [10] Nasir Ahmad. The DCT – an algorithm that impacts the world of digital audio and video. QUANTOM – Research and scholarship at the university of Mexico.
- [11] A.K Jain –. : Fundamentals of digital image processing. , prentice hall , 1989, ISBN 0-13-336165-9.