Lossless Image Compression using Shift coding

Ahmeed S. Farhan
Computer Center
University of Anbar
Anbar, Iraq

Fouad H. Awad
College of Business Informatics
University of Information Technology and Communication
Bagdad, Iraq

Meaad A. Khalaf
Collage of Computer Science
University of Anbar
Anbar, Iraq

ABSTRACT
The development of multimedia and digital imaging has led to high quantity of data required to represent modern imagery. This requires large disk space for storage, and long time for transmission over computer networks, and these two are relatively expensive. These factors prove the need for images compression. Image compression addresses the problem of reducing the amount of space required to represent a digital image yielding a compact representation of an image, and thereby reducing the image storage/transmission time requirements. The key idea here is to remove redundancy of data presented within an image to reduce its size without affecting the essential information of it. We are concerned with lossless image compression. In this paper our proposed approach is a mix of a number of already existing techniques. Our approach works as follows: first, we deal with colors Red, Green, and Blue separately, what comes out of the first step is forward to the second step where the zigzag operation is to rearrange values to be more suitable for preprocessing. At the final, the output from the zigzag enters to the shift coding. The experimental results show that the proposed algorithm could achieve an excellent result in lossless type of compression losing data.

Keywords
Lossless Compression, Shift Coding, Zigzag Operation.

1. INTRODUCTION
Image compression can be defined as minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. This reduction allows more images to be stored in a given amount of memory space but the major benefit is the reduction of the time required for images to be sent over the Internet or downloaded from Web pages [1].

There are several different ways in which image files can be compressed like JPEG, GIF, PNG, fractals and wavelets. For Internet use, the two most common compressed graphic image formats are the JPEG format and the GIF format. The JPEG method is more often used for photographs, while the GIF method is commonly used for line art and other images in which geometric shapes are relatively simple[2]. Other method like fractals and wavelets offer higher compression ratios than the JPEG or GIF methods for some types of images. In near future PNG format will replace GIF format.

Compression is achieved by removing one or more of the three basic data redundancies:

1) Coding redundancy, which is presented when less than optimal code words are used.
2) Interpixel redundancy, which results from correlations Between the pixels of an image.
3) Psychovisual redundancy, which is due to data that are ignored by the human visual system [3].

One major difficulty that faces lossless image compression is how to protect the quality of the image in a way that the decompressed image appears identical to the original one. In this paper we are concerned with lossless image compression based on Run-length and shift coding algorithms, which compresses different types of image formats [4].

2. IMAGE COMPRESSION
Decreasing the irrelevance or redundancy of an image is the fundamental aim of the image compression techniques to provide the facility for storing and transmitting the data in an effective manner .The initial step in this technique is to convert the image from the representation of their spatial domain into a separate type of the representation by the use of few already known conversions and then encodes the converted values i.e., coefficients. This technique allows the huge compression of data as compared to the predictive techniques, though at the cost of the huge computational needs. It is a process intended to yield a compact representation of an image, thereby reducing the image storage/transmission requirements. Image compression techniques reduce the number of bits required to represent an image by taking advantage of these redundancies. An inverse process called decompression (decoding) is applied to the compressed data to get the reconstructed image. The objective of compression is to reduce the number of bits as much as possible, while keeping the resolution and the visual quality of the reconstructed image as close to the original image as possible. Image compression systems are composed of two distinct structural blocks: an encoder and a decoder [5] [6].

3. RELATED WORK
A large number of data compression algorithms have been developed and used throughout the years. Some of which are of general use, i.e., can be used to compress files of different types (e.g., text files, image files, video files, etc.). Others are developed to compress efficiently a particular type of files. It has been realized that, according to the representation form of the data at which the compression process is performed, below is reviewing some of the literature review in this field.

A. Alarabeyyat, S. Al-Hashemi and T. Khdour. Introduces approach works as follows: first, we apply the well-known Lempel-Ziv-Welch (LZW) algorithm on the image in hand. What comes out of the first step is forward to the second step where the Bose, Chaudhuri and Hocquenghem(BCH) error correction and detected algorithm is used. To improve the compression ratio, the proposed approach applies the BCH algorithms repeatedly until “inflation” is detected. The experimental results show that The proposed algorithm could achieve an excellent compression ratio Without losing data when compared to the standard compression algorithms[7].

Mrs.Bhumika Gupta in this paper addresses the area of image compression as it is applicable to various fields of image processing. On the basis of evaluating and analyzing the current image compression techniques this paper presents the SIMPLE COMPRESSION TECHNIQUE (SCZ) approach applied to image compression. It also includes various benefits of using image compression techniques[8].

In this paper Rime Raj Singh Tomar and Kapil Jain presents differential pulse code modulation for image compression lossless and near-lossless compression method is introduced which is efficient due to its high compression ratio and simplicity. This method is consists of a new transformation method called Enhanced DPCM Transformation (EDT) which has a good energy compaction and a suitable Huffman encoding. After introducing this compression method it is applied on different images from Corel dataset for experimental results and analysis. Also we compare it with other existing methods with respect to parameter compression ratio, peak signal noise ratio and mean square error[9].

In [10] the authors present a strategy to increase the compression ratio with simple computational burden and excellent decoded quality. Higher compression ratio is achieved by applying different compression thresholds for the wavelet coefficients of each DWT band (LL and HH) while DCT transform is applied on (HL and LH) bands with preserving the quality of reconstructed medical image. The retained coefficients are quantized by using adaptive quantization according to the type of transformation. Finally the entropy coding (variable shift coding) is used to encode the quantization indices. The Discrete Wavelet Transform (DWT) analyzes the signal at different frequency bands with different resolutions by decomposing the signal into an approximation and detail information. Image coded by DWT do not have the problem of blocking artifacts which the DCT approach may suffer.

4. THE GENERAL LOSSLESS IMAGE COMPRESSION MODEL

The first step of image compression is deal with the colors Red, Green and blue separately. The next applied Zigzag operation is then applied to rearrange values to be more suitable for preprocessing, so, linearizes the elements scanning the image from left to right and then top to bottom, a zigzag scanning Pattern is used as shown in Figure 2.

After the Zigzag process, the image will become one dimensional. As we know the convergence of colors in adjacent pixels, it is possible to save the first value, then subtract the second from the first, produce a value, after that subtract the third value from the second, produce value and etc.

Then before shift coding begins, all values of data must be converted to positive. The negative values are multiplied by -2 to become even and positive. While the positive values are multiplied by 2 and then 1 is subtracted from the multiplied value to become odd in order to distinguish them from negative values in the decompression process. The algorithm (1) illustrates the conversion of all values to positive.

The shift coding step reduces the number of bits required to represent the data of Zigzag operation. After Zigzag, most of the values are between 0 and 15. These values need 4 bits instead of 8 bits to represent them. Therefore, applying shift coding reduces that waste by a measurable amount. According to input values, we used two methods to apply shift coding.

The final step in the image compression involves the proposed two shift coding based techniques. The first method works with Leading Short Word and the second one works with Lead Bit. This final step reduces the number of bits required to represent the values. The system model is shown in Fig. 1.

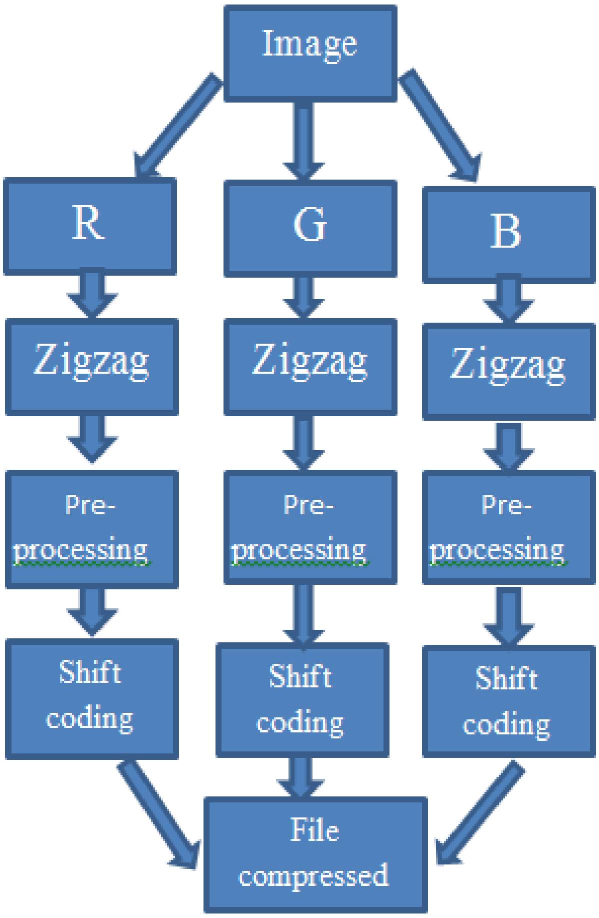
Algorithm(1) : The conversion of all values to positive

Goal **:** convert all values to positive

Input **:** ***REL_array*** // array
Output **:** ***positive_array*** // array

Step 1 **:**

For all i ***do*** {where $0 \le i \le$ ***REL_array*** length}
  If REL_array (i) < 0 then
  positive_array (i) ← ***REL_array*** (i)*-2
  Else
  positive_array (i) ← ***REL_array*** (i)*2-1
  End If
  End For i***Step 2*** **:**
  Return (***positive_array***)



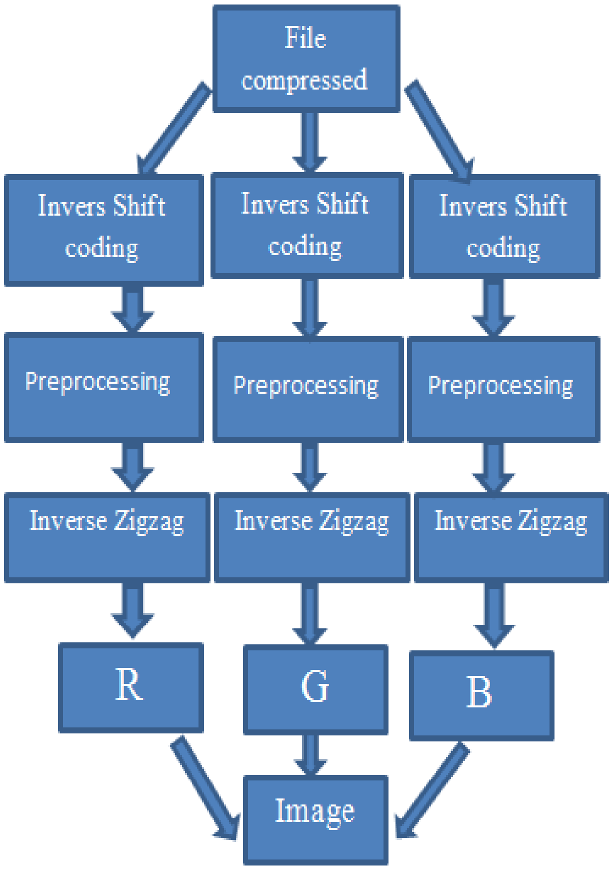


Figure.1 The steps of lossless image compression model

5. THE PROPOSED TWO METHODS FOR SHIFT CODING

The first method (with leading short word) is applied when most of the values are less than 15. These values will be represented using 4 bits while other values that are larger than 15 will take 10 bits (the first 4 bits is the integer 15 and the other 6 bits will take the value minus 15). The algorithm (2) illustrates first method.

The second method (with lead bit) is applied when the number of values larger than 15 is more than the number of values less than 15. In this method, the values that are larger than 15 take 8 bits and the values that are less than 15 take 5 bits. It sets the most significant bit to 1 if the value is less than 15 and gives this value 4 bits and if the value is larger than 15 it sets the most significant bit to 0 and gives 7 bits to the value. The algorithm (2) illustrates the second method. The tables below show an example of the shift coding process.

Table 1. Example 1 of shift coding methods

With Leading Short Word		With Lead Bit	
Number	***Coding***	***Number***	***Coding***
1	0001	1	10001
7	0111	7	10111
0	0000	0	10000
4	0100	4	10100
6	0110	6	10110
70	1111110111	70	01000110
3	0011	3	10011
11	1011	11	11011
2	0100	2	10100
1	0001	1	10001
Total no. of bits	46		53

The results of applying first method of shift coding as shown in Table (1) are better than those of the second method because most of the values are less than 15. In table (2), the results of applying the second method was better than those obtained from applying the first method, because the values that are less than 15 are not the most.

Table 1. Example 2 of shift coding methods

With Leading Short Word		With Lead Bit	
Number	***Coding***	***Number***	***Coding***
32	1111010001	32	00100000
7	0111	7	10111
18	1111000011	18	00010010
20	1111000101	20	00010100
6	0110	6	10110
70	1111110111	70	01000110
17	1111000010	17	00010001
11	1011	11	11011
40	1111100011	40	00101000
1	0001	1	10001
Total no. of bits	76		68

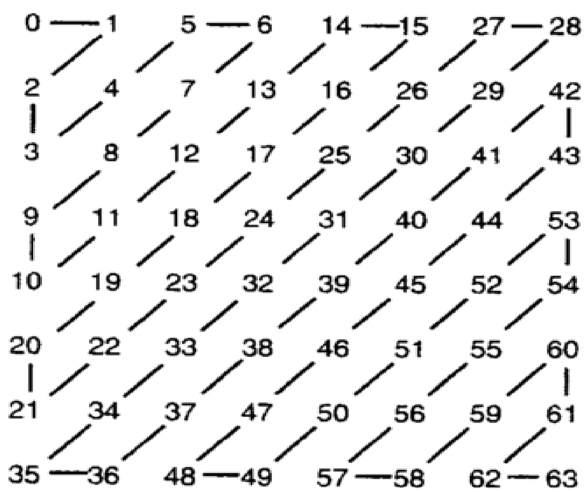


Figure.2 Sequence of zigzag

Algorithm(2) :

With Leading Short Word Method of shift coding

***Goal* :** Reduce the number of bits required to storage

Input* : *positive_array // array

Output* : *bit_array // array

***Step 1* :**

bit_array (0) ←1\\ this value is an indicator to use this method

Z←1

***Step 2* :**

For all I ***do*** {where 0 ≤ I ≤ ***positive_array*** length}

If positive_array (I) < 15 then

B()←convert_to_binary (***positive_array*** (I))

For all J ***do*** {where 0 ≤ J ≤ 3}

bit_array (Z) ← ***B***(J)

Z←Z+1

End for J

Else

For all J ***do*** {where 0 ≤ J ≤ 3}

bit_array (Z) ← 1

Z←Z+1

End for J

B()←convert_to_binary (***positive_array*** (I)-15)

For all J ***do*** {where 0 ≤ J ≤ 5}

bit_array (Z) ← ***B***(J)

Z←Z+1

End for J

End if

End for I

***Step 3* :**

Return (***bit_array***)

6. IMPLEMENTATION AND RESULT

During the implementation phase, the used images are of different types with different sizes. For the first image, the height* width was 512 x 512 while size of the Image size before compression was 786432 KB and image size after compression was 629245 KB.so, the 3rd image dimensional was 128*128 and image size before compression was 49152 while the size after compression was 41842 KB. The last image height * width was 176*144 and the size of image before compression was 76032 KB while the size after compression was 59065 KB, as shown in table 3.

No.	Height* Width	Image size Before Compression	Image size After Compression
1	512 x 512	786432 KB	629245 KB
2	156 x 256	196608 KB	156917 KB
3	128 x 128	49152 KB	41842 KB
4	704 x 576	1216512 KB	833891 KB
5	176 x 144	76032 KB	59065 KB

7. CONCLUSION

In this paper, we introduced a technique for image compression using shift coding we provided an overview of various existing coding standards lossless image compression techniques. We have proposed a high efficient algorithm which is implemented using the shift coding approach. The proposed method takes the advantages of the zigzag with the advantages of the shift coding which is known for its simplicity and speed. The ultimate goal is to give a keep the time and space complexity minimum. The way of image compression is lossless so that's means in compression and decompression will save the quality without any lost. The evaluation of the proposed method shows good performance image before compression cannot be distinguished from the image after compression. Also the proposed system operates efficiently and quickly in terms of memory and CPU.

8. REFERENCES

[1] Digital Image Processing by R. C. Gonzales and R. E. Woods, Addison-Wesley Publishing Company, 1992.

[2] Two-Dimensional Signal and Image Processing by J. S. Lim, Prentice Hall, 1990.

[3] H. Zha, "Progressive Lossless Image Compression Using Image Decomposition and Context Quantization," Master Thesis, University of Waterloo, Waterloo.

[4] R. Rajeswari and R. Rajesh, "WBMP Compression," *International Journal of Wisdom Based Computing*, Vol. 1, No. 2, 2011. doi:10.1109/ICIIP.2011.6108930.

[5]]. Pao-Yen Lin ," Basic Image Compression Algorithm and Introduction to JPEG Standard" , National Taiwan University, Taipei, Taiwan, ROC 2009.

[6] G.M.Padmaja and P.Nirupama ," Analysis of Various Image Compression Techniques", ARPN Journal of Science and Technology 2011- 2012. All rights reserved.

[7] A. Alarabeyyat1, S. Al-Hashemi," Lossless Image Compression Technique Using Combination Methods" Journal of Software Engineering and Applications, 2012, 5, 752-763.

[8] Pauri Garhwal,Uttrakhand," Study Of Various Lossless Image Compression Technique" International Journal of Emerging Trends & Technology in Computer Science

(IJETTCS) , Volume 2, Issue 4, ISSN 2278-6856 , July – August 2013.

[9] Rime Raj Singh Tomar and Kapil Jain ," Lossless Image Compression Using Differential Pulse Code Modulation and its Application" Published in: Computational Intelligence and Communication Networks (CICN), 2015 International Conference, DOI: 10.1109/CICN.2015.84, Date Added to IEEE Xplore: 18 August 2016.

[10] Vikash Yadav, Monika Verma and Vandana Dixit Kaushik, "A Hybrid Image Compression Technique for Medical Images", Published in: Computational Intelligence and Communication Networks (CICN), 2015 International Conference, DOI: 10.1109/CICN.2015.51, Date Added to IEEE Xplore: 18 August 2016.