International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 3 Issue 12, December 2014

The Implementation of Run Length Encoding for RGB Image Compression

Varsha Bansal[1] Pratishtha Gupta[2], Suhail Tomar[3]
Computer Science, Banasthali University, Jaipur, Rajasthan, India[1]
Computer Science, Banasthali University, Jaipur, Rajasthan, India[2]
Computer Science & Engg. Department, SRM University, Kattankulathur, Kancheepuram, Tamil Nadu, India[3].

Abstract: -

This research paper presents the implementation of Run Length Encoding that is one of the lossless image compression technique. This paper gives the implementation of Run length encoding compression algorithm which is proficiently suited for RGB images. Here considered colored and natural images for the inspection of implemented technique. By which image can be compressed and occupy lesser space in memory, and improve the performance of the system. Such that run length encoding split larger sequences of runs that affects compression ratio into small sequences of runs without degrading the quality of image.

Keyword; RLE (run length encoding), image compression, R (Red), G(Green), B(blue).

Introduction (Compression)

Image processing and compression is currently a prominent context for computer science field.
Basically, image compression is the processes of images that encode the images into small code without any loss of information. The image compression process provides the optimum utilization for storage, Gupta G. et al, [1].
The small size of images allows more images to be stored in a memory space or disk drive. It also required less time for images to be sent over the Internet or downloaded from web pages. Nagarajan A. et al [2].In other words, the basic motivation of image compression is using short quantity of information to represents the original image without loss of information. And reduce the size of image for decrease the transmission time.

Run length encoding

RLE (Run-length encoding) is a very easy and simple technique of data compression, in which the count of occurrence of same data is stored as a single data value and single count.
This is most useful for the data that contains many such runs: for example, a simple colored image such as same color occur many time. It is less useful with images that don't have many runs as it could greatly increase the files size The Run length encoding technique performs a lossless compression of input images that is based on sequences of identical values (runs), Amin A.,et al. [3]
In other word, the run length encoding scheme only encodes the consecutive number of same color pixels the probability of occurrence of consecutive same color is very high.
Run Length Encoding provides efficient compression of data, whereas the data with large number of runs or large number pixel contains same intensity value.
But this encoding also has the drawback; RLE scheme does not always provide data compression. In some case where runs are smaller length or each pixel value has different intensity value from its adjacent pixel, this method performs very poorly and instead of compressing data, Akhtarl M.B., et al. [4]

Working of Run Length Encoding

Here, let's take the example of image and perform the run length encoding. There will be too much long runs of white pixels, and short runs of black pixels. Let us take a single scan line or row of image with B representing a black pixel and W representing white pixel.
WWWWWWWWWWWW**B**WWWWWWWWWWWW
BBBWWWWWWWWWWWWWWWWWWWWWWWW
W**B**WWWWWWWWWWWWWW
Here apply the run-length encoding for image compression algorithm to the above scan line, we get the following:

(12W) (1B) (12W) (3B) (24W) (1B) (14W).
12 W, means 12 count of white color pixel, and so on.

Modified Run Length Encoding Scheme

The basic scheme of run length encoding is to improve the system working and performance. Decrease the memory used by Run Length Encoding (RLE) technique and

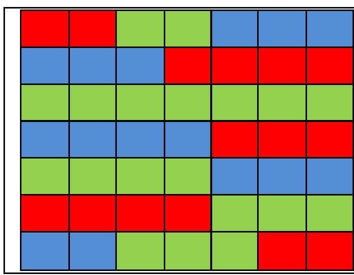
ISSN: 2278 – 1323

increase the compression rate. This paper proposed some modification in RLE scheme; this modification provides prime improvement in compression rate of image data.

Firstly, analyzing the inputting color image at the starting step of algorithm. If there are any big sequences of same intensity or pixel value, that may need the large number of bit for represent the length of each run. In proposed method if pixel of input image contains same or nearest value with its adjacent pixel then both pixel values consider as a same data or intensity value in RGB image, Joseph S., at al.[5]

Proposed Work

The basic philosophy behindhand the selecting Run Length Encoding technique, that is loss less technique and based on intrinsic property of images and they have same patterns in nearest pixel area of image. Specifically the intensity of two pixels is very much same in nearest area. This property of image is exploited to design a very effective image compression technique. The technique used in this compression methodology and run length coding are described in this section. Here consider run length compression for given image. The bellow image has RGB color combination. Image read from first pixel of image and starts compression.



2R, 2G, 3B
3B, 4R
7G
4B,3R
4G,3B
4R, 3G
2B,3G,2R

Fig 1. RGB image with pixel value.

- Here each cell of matrix represents the pixel.
- This algorithm scans image one row at a time.
- This technique shows result as above manner.
- The result of first row,"2R 2G 3B" that is represents
  2R means 2 pixels of red color.
  2G means 2 pixels of green color.
  3B means 3pixels of blue color. And so on.

The basic steps of proposed algorithm of Run Length Encoding are as following.

Compression

Step 1: Input the RGB source image file.
Step 2: Find out the size of source image
Step 3: Read pixel values from first pixel of source image.
Step 4: Read next pixel value, if current pixel is last then exit otherwise
a. If next pixel value that is j and j+1 is same as the previous pixel value then Count = count+1;
b. Else Any mismatch in RGB value of next pixel as the previous than save as the new value of pixel in array.
Step 5: Read and count all the value of pixel.
Step 6: Go to step 4 until all pixel read.
Step 7: Display the result with count of same intensity value of pixel.

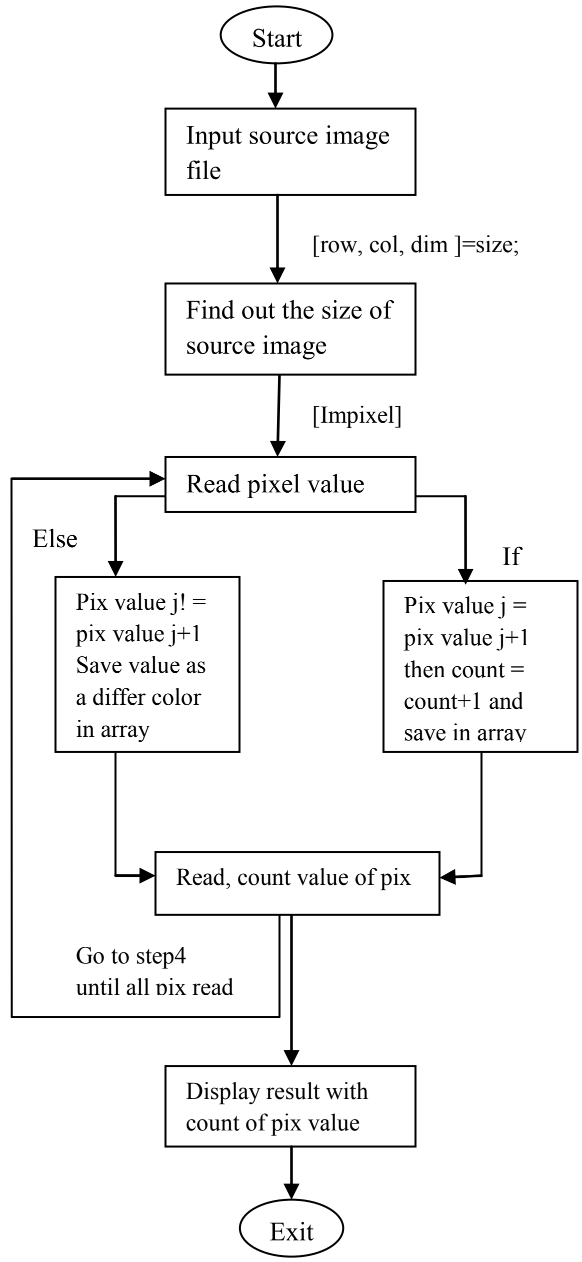


Fig 2.flow chart of compression

ISSN: 2278 – 1323

Decompression

Step 1: Read the compressed array and obtain the image size.

Step 2: Create the blank array for reconstruction of compressed image.

Step 3: For reconstructing compressed image**,**

a. Construct the i^{th} row of compressed image with putting run length value in reconstruct array from compressed array.

b. Then construct $i+1^{th}$ row then next row and so on.

Step 4: Step 3 is repeated until reconstruct array fill by value of compressed array.

Step 5: Reconstruct array, store as a decompressed image file.

Step 6: Display the decompressed image file.

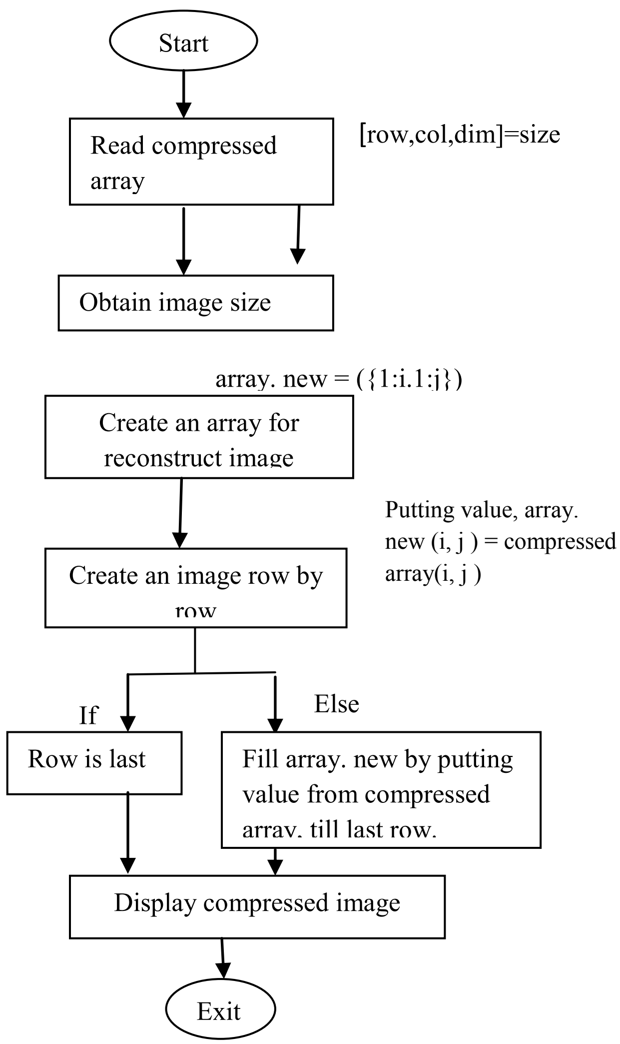


Fig 3.Flow chart of decompression

Test Result

Input: This function takes source colored images.
Output: This function provides the compressed image file.

Before compression for Cartoon.jpg image
Size 487*426*24 = 4979088 bits
4979088/8 = 622386 bytes.

Figure 4. Cartoon

After compression for above image
Compression percentage is 42.10%

Before compression for Tree.jpg image
Size 194*259*24 = 1205904 bits
1205904/8 = 150738bytes

Figure 5. Tree

After compression for above image
Compression percentage is 49.53%

ISSN: 2278 – 1323

Before compression of beg.jpf image

Before compression of Bag.jpg image
Size 235*300*24 = 1692000 bits
1692000/8 = 360990 bytes

Figure 6. Bag.

After compression of above image
Compression percentage is 82.5%

Before compression of Paint.jpg image
Size 514*346*24 = 4268256 bits
4268256/8 = 533532 bytes

Figure 7. Paint

After compression of above image
Compression percentage is 33.23%

Before compression for Thumbnail.jpg image
Size 699* 697*24 = 11692872 bits
11692872/8 = 1461609 bytes

Figure 8. Thumbnail

After compression of above image
Compression percentage is 20.29%

Before compression of Tom and jerry.jpg image
Size300*280*24 = 2016000 bits
2016000/8 = 252000 bytes

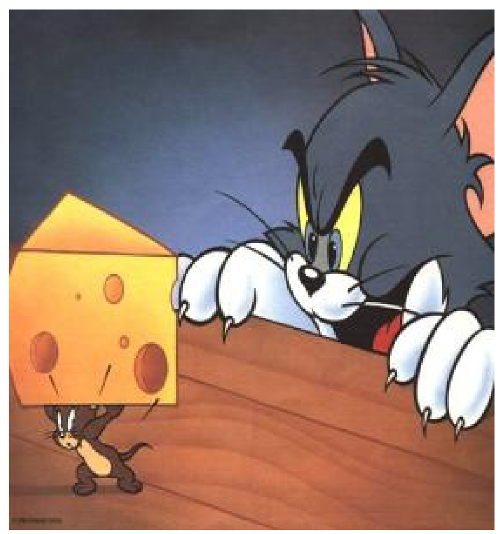


Figure 9. Tom and jerry

After compression of above image
Compression percentage is 57.73%

Analysis of Run Length Encoding scheme

According to above result of run length encoding scheme, this compression technique works efficiently where large areas of similar pixel value takes place in image data.

ISSN: 2278 – 1323

In the image data 'Bag.jpg', RLE compression technique shows much more compression percentage that is 82.5% because here is large number of pixels have same value in it.
Now, consider the 'Thumbnail.jpg' image, this image slightly compressed than 'Bag .jpg' image data because there are multi colors present in it.

Conclusion

This research paper provide a working of lossless image compression technique (RLE) of image data. It is the explicit from of algorithm that extract the pixel value from image data. Compression is very much important and useful part of image processing. Basically these techniques will discover comprehensive use in GIS images (geographical information system), Medical image, because these type of image has large area of identical pixel pattern.

Images Name	Row	col	RGB bits	Total size before compression	Size in bytes	Array size in row	Array size in col	Compression calculation	Compression Percentages.
Cartoon.jpg	487	426	3*8=24	487*426*24 = 4979088 bits	622386 bytes	2096640	96	2096640/ 49790888 *100	42.10%
Bag.jpg	235	300	3*8=24	235*300*24 = 1692000	360990 bytes	1397184	96	1397184/ 1692000 *100	82.5%
Paint.jpg	514	346	3*8=24	514*346*24 = 4268256 bits	533532 bytes	1418688	96	1418688/ 4268256 *100	33%
Thumbnail. jpg	699	697	3*8=24	699*697*24 = 11692872 bits	1461609 bytes	2373408	96	2373408/ 11692872 *100	20.29%

Table 1. This table contains the result of run length encoding scheme.

References

[1] Gupta G., Gupta K.L. Jyoti A. , **"**AN ADVANCED COMPRESSION APPROACH WITH RLE FOR IMAGE COMPRESSION" International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 2, February 2014

[2] Nagarajan A., Alagarsamy K. "AN ENHANCED APPROCH IN RUN LENGTH ENCODING SCHEME" International Journal of Engineering Trends and Technology- July to Aug Issue 2011

[3] Amin A., Aheman Q. ,Junaid M. Habib M.Y, Anjum W**. ,** "MODIFIED RUN LENGTH ENCODING SCHEME WITH INTRODUCTION OF BIT STUFFING FOR EFFICIENT DATA COMPRESSION**"** **6th** International conference on internet technology and secured transaction 11-14 December 2011 Abu Dhabi (978-1-908320-00-1-/11/$26.00 @ 2011 IEEE)

[4] Akhtarl M.B., Qureshi A.M. , and Islam Q., "OPTIMIZED RUN LENGTH CODING FOR JPGE IMAGE COMPRESSION USED IN SPEC RESEARCH PROGRAM OF IST.(978-1-61284-941-6/11/$26.00 ©2011 IEEE)

[5] Joseph S., Srikanth N. **"**A NOVEL APPROCH OF MODIFIED RUN LENGTH ENCODING SCHEME FOR HIGH SPEED DATA COMMUNICATION APPLICATION**"** International Journal of Science and Research (IJSR**)** Volume 2 Issue 12, December 2013

.

ISSN: 2278 – 1323