International Journal of Advances in Embedded Systems
Volume 1, Issue 1, 2011, pp-01-03
Available online at: http://www.bioinfo.in/contents.php?id=99

Compression of Noisy and Noiseless Images through Run Length Encoding Approach

J.A. Sangvikar and G.R. Joshi
Vidya Pratishthan's Arts, Science and Commerce College, Baramati, MS, India
e-mail: sangvikarjagdish@yahoo.co.in, joshigajananr@gmail.com

***Abstract*—In this experiment we have concentrated on the performance of the Run Length Encoding for noisy and noiseless images. We have considered four black and white images and four types of noises. Here our focus is on the compression ratios calculated before and after adding the noise in the image. The encoding is done with the help of traditional RLE scheme converted into a MATLAB code. There are two codes noiserle.m and noiselessrle.m. In noiserle.m, first of all the image is converted into binary, noise is added and then RLE is applied. While in noiselessrle.m, the same thing as with previous is done but without a noise. For every image, temp and runtemp files are generated where temp is a binary text file and the runtemp is a text file obtained after applying RLE.**

The images considered for this experiment are tire, testpat1, moon and circle in MATLAB. After applying the RLE code, we found CR>1 for all the noiseless images. The work has shown apparent results such that adding noise in the image yields a good compression ration. To get the clear cut idea, tabular data as well as a graph of CR vs 'Image name & noise' is also drawn. This is a suitable phenomenon to save the bandwidth while transmission. Such images, which are tending extremely towards either the upper or lower value of the gray scale, yield the better compression ratios. Also images containing the regions filled with patterns give a better compression ratio.

***Keywords:* Compression Ratio, Image Compression, Noise, RLE.**

I. INTRODUCTION

It deals with techniques for reducing the storage required to save image or bandwidth required to transmit. Every day, an enormous amount of information is stored, processed, and transmitted digitally. Companies provide business associates, investors, and potential customers with financial data, annual reports, inventory, and product information over the Internet. Order entry and tracking, two of the most basic on-line transactions, are routinely performed from the match of one's own home. Because much of this on-line information is graphical or pictorial in nature, the storage and communications requirements are immense. In order to use the bandwidth efficiently, methods of compressing the data prior to storage and transmission are of significant practical and commercial interest.

Image compression addresses the problem of reducing the amount of data required to represent a digital image. The underlying basis of the reduction process is the removal of redundant data. From a mathematical viewpoint, this amount of transforming a 2-D pixel array into a statistically uncorrelated data set. The transformation is applied prior to storage or transmission of the image. At some later time, the compressed image is decompressed to reconstruct the original image or an approximation of it. [9, 10]. Feasibility of image compression can also be studied. [16].

II. METHODOLOGY

The methodology used for this experimental work is as follows:

- Step-1: Set the previous symbol equal to an unmatchable value.
- Step-2: Read the next symbol from the input stream.
- Step-3: If the symbol is an EOF exit.
- Step-4: Write out the current symbol.
- Step-5: If the symbol does not match with the previous symbol, set the previous symbol to the current symbol, and go to Step-2.
- Step-6: Read and count additional symbols until a non-matching symbol is found. This is the run length.
- Step-7: Write out the run length.
- Step-8: Write out the non-matching symbol.
- Step-9: Set the previous symbol to the non-matching symbol, and go to Step-2.

The experimental work starts with applying the Run Length Encoding program on the Noiseless images. First of all the image is converted into binary format in an array of xxxxxxx **x** 8. This gave us multiple rows each of 8 bits. Then every bit is written into a binary text file called temp.m. Then the binary text file temp.m is read and each row (line) in the binary text file is encoded using the RLE scheme and the same line is transferred to another text file called runtemp.m.

In this experimental work the following images are taken into account. The said images are taken from the Matlab software. Only the images having regions that contain patterns are selected for this experiment. [5]

- tire.tif
- testpat1.png
- moon.tif

International Journal of Advances in Embedded Systems
Volume 1, Issue 1, 2011, pp-01-03
Available online at: http://www.bioinfo.in/contents.php?id=99

- circles.png

For every image, we have added four types of noises and the resulting parameters are calculated. The noises that are taken into account are….

- Gaussian Noise
- Poisson Noise
- Salt & pepper Noise
- Speckle Noise [9,10,15]

The compression ratio is calculated with the help of the formula:

CR=n1/n2;

Where,

n1 is the original file size and n2 is the compressed file size.

III. RESULTS

After processing each and every image with the help of developed RLE program in the Matlab, the temp.m and runtemp.m files are obtained with respect to each image. The temp.m file gives the exact representation of the binary version of the image in text. While the file runtemp.m gives the Run Length Encoded representation. The number of bytes needed for the corresponding files are calculated with the help of dir() function in the MATLAB. Also the same number of bytes can be calculated by observing the properties of the files.

The numbers of bytes needed for both the files for each image are calculated and also the Compression ratios are calculated.

For every image, the image, observation table containing image name and compression ration with respect to each considered noise is prepared. Also, to have the precise idea, the same observation table is exhibited in the form of graph.

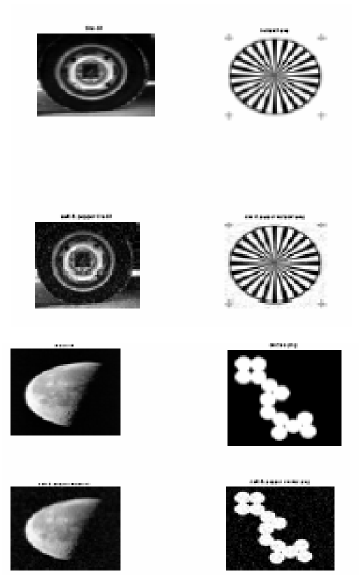
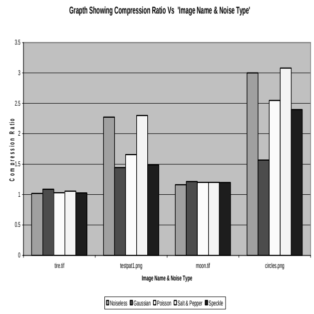


Fig. 1: Original & Sample Noise Added Images

TABLE 1: SHOWING COMPRESSION RATIO OF ALL IMAGES.

Noise Type	Compression Ratio			
	Tire	Testpat1	Moon	Circles
Noiseless	1.0207	2.274	1.16	3
Gaussian	1.0869	1.4407	1.2145	1.5655
Poisson	1.0281	1.6547	1.1992	2.5488
Salt & Pepper	1.0548	2.3001	1.1998	3.08
Speckle	1.0265	1.4867	1.1972	2.3962



Graph 1: Graph Showing Compression Ration Vs "Image Name & Noise type'

IV. DISCUSSION

In many pictures we have observed that the large number of pixels in the picture have the same intensity or color. Considering this fact the picture can be defined by pixel intensity and number of successive pixels on a given scan line with the same intensity. Such a representation of a picture is called Run Length Encoding.

Pictures or images, which are tending extremely towards either the upper or lower value of the gray scale, yield the better compression ratios. This can be applied on medical images, Xerox documents and fax documents also.

V. CONCLUSION

It is found that for all of the noiseless mages, number of bytes needed to encode the binary files is less than the existing binary file.

From 'compression ratio CR>1 and RLE scheme', it clearly indicates that there is no loss of data when the compression is done.

Images containing the regions filled with patterns give a better compression ratio.

For the noiseless circles.png file, a very good compression ratio CR=3 is found. This means that the file is compressed $1/3^{rd}$ times the original one.

When the RLE process is applied to the Gaussian noise added image file (tire.tif) in binary format, it yielded a good compression ratio rather than Poisson, Salt & Pepper and Speckle.

The compression ratio for Gaussian noise added image gave the good compression ratio greater than Noiseless, Poisson, Salt & Pepper and Speckle.

The images tire.tif and moon.tif could give a better compression ratio in response to Gaussian noise. Also by observing these images, it is found that majority of its bits are tending to gray scale values.

Among CR of noisy images, CR of image with Speckle Noise is less than the CR of image with Gaussian Noise.

The CR of Speckle & Poisson noisy image is nearly equal.

It is found that CR of Salt & Pepper noisy images is greater.

Future Work: Decoder and Variants on Run-Length Encoding has to be developed.

Another alternative RLE schemes can also be written to encode data down the length of a bitmap (the Y axis) along the columns, to encode a bitmap into 2D tiles, or even to encode pixels on a diagonal in a zigzag fashion.

ACKNOWLEDGMENT

The author thanks to Dr. Sadhana Deshpande, Principal, Vidya Pratishthan’s Arts, Science and Commerce College, Vidyanagari, Baramati, Dr. Shivanand Gornale, Head, Government College, Mandya (K.S.) for giving the moral support.

REFERENCES

[1] Adrian Low, “Computer Vision & Image Processing”, McGraw Hill (1991)

[2] A.P. Godse, “Computer Graphics”, Technical publications (2nd Edition)

[3] B.Chanda & Dutta Majumdar, “Digital Image Processing and Analysis”, PHI (2001)

[4] Encyclopedia of Graphics File Formats, O'Reilly

[5] Foly,Van Dam, Feiner, “Computer Graphics – principles & practice”, Pearson education.(2nd Edition)

[6] Jain Anil. K. “Fundaments of Digital Image Processing”, Prentice Hall, Upper Saddle River (1989).

[7] Kenneth R.Castleman, “Digital Image processing”, *PHI. (2002)*

[8] Pratt, W.K. “Digital Image Processing”, *3rd ed., PIKS Inside, Awiley-interscience publication. (2001)*

[9] Rafael C.Gonzalez & Richard E.Woods, “Digital Image Processing”, *Second edition, (2005).*

[10] Rafael C.Gonzalez & Richard E.Woods, “Digital Image Processing using MATLAB”, *Pearson education (2004).*

[11] Ranjan Bose, “Information theory, coding and cryptography”, *Tata McGraw-Hill publication.(2003)*

[12] Scott E. Umbaugh, “Computer Vision and Image processing”, *PHI international (1998).*

[13] Shalkoff R.J. John wiley and sons, New York, “Digital Image Processing and computer vision”. *(1989).*

[14] Stewan W.Smit,”Scientist and Engineers Guide to Digital Image processing”, *Second Edition, (1999).*

[15] www.mathworks.com

[16] Dr. S.N. Singh, “ Hardware Image Compression with FPGA”, *International Journal of Recent Trends in Engineering, vol 2, No. 8, November 2009.*

[17] Markus Holzer, “Improving Raster Image Run-Length Encoding Using Data Order”, *B.W. Watson and D. Wood (Eds.): CIAA 2001.*

[18] David A. Clunie, “Losless Compression of Gray Scale Medical Images- Effectiveness of Traditional and State of Arts Approaches”.