Special Issue Published in International Journal of Trend in Research and Development (IJTRD), ISSN: 2394-9333, www.ijtrd.com

A Compact Image Compression Technique for Data Communication

[1]Sk. Abdul Vaheed and [2]R Bala Dinakar
[1]PG Student, [2]Assistant Professor,
[1,2]Department of Computer Applications, Godavari Institute of Engineering and Technology, Rajahmundry, India

***Abstract*--** This paper presents a state-of-the-art implementation of lossless image compression algorithm LOCO-R, which is based on the LOCO-I (low complexity lossless compression for images) algorithm developed by Weinberger, Seroussi and Sapiro, with modifications and betterment, the algorithm reduces obviously the implementation complexity. Experiments illustrate that this algorithm is better than Rice Compression typically by around 15 percent. Having identified the difficulties with the storage and transmission of data over long distances, research has led to a number of efficient image compression techniques to conserve bandwidth. A study of the literature available on the current problem reveals a large number of available techniques. Some of these techniques are mentioned in the Introduction. Each technique differs from the other technique in terms of Peak Signal to Noise Ratio (PSNR) and Compression Ratio (CR).

***Keywords*--** *Image Compression; LZW; BCH*

I. INTRODUCTION

Lossless image compression is well-established as a means of reducing the volume of image data without compromising the data quality. Missions often desire hardware to perfom such compression, the aim is to reduce the demand on processors and to increase the speed at which images can be compressed. Currently, the available space-qualified hardware designed for lossless compression is based algorithm Rice Compression has limitations, however, and there are other algorithms that achieve better lossless image compression. An algorithm known as LOCO-I (low complexity lossless compression for images i) is an appealing choice. LOCO-I is at the core of JPEG-LS, the algorithm selected by the Joint Photographic Experts Group as the International Standards Organization/International Telecommunications Union (lSOIITU) standard for lossless and near-lossless compression of continuous-tone images .The paper have improved LOCOI to reduce the complexity of implementation, regarded as LOCO-R. For natural images, the compression is achieved by the LOCO-Rexceeds Rice Compressionin compression rate LOCO-R ALGORITHM The LOCO-R algorithm described in this section is based on the LOCO-I algorithm[5] , it takes as input a rectangular image with 8-bit pixel values (the pixel values are within the range ° to 255). The compressed image produced is a Sequence of bits from which the original image can be Reconstructed Let wd be the image width and ht be the image height. Pixels are identified by coordinates (x, y) with x in the range [O,wd-l] and y in the range [O,ht-l]. This paper supposes: (0,0) corresponds to the upper left comer of the image. The LOCO-R algorithm is based on predictive compression. During compression, the pixels of the image are processed in raster scan order. Specifically, y is incremented through the range[O,ht-l], and for each y value, x is incremented through the range[O,wd-l]. (Thus, the ydimensions, the slowly varying dimension.)The first two pixels, with coordinates (0,0) and (1,0), are simply put into the output bit stream uncoded. For all other pixels of the image, the processing that occurs can be conceptually divided into four steps: (1) Classify the pixel into one of several contexts according to the values of (usually 5) previously encoded pixels. (2) Estimate the pixel value from (usually 3) previously encoded pixels, and add a correction(called the bias), which depends on the context. (3) Map the difference between the estimate and the actual pixel value to a non-negative integer, and encode this integer usingGolomb's variable length codes. (4) Update the statistics for the context based on the new pixel value. These steps are explained in detail below.

II. RELEATED WORK

Medical diagnosis becomes effective if it identifies the defective areas in limited processing. In medical images, some structures in the data areof interest. These structures typically occupy a small percentage of the data, buttheir analysis requires contextual information like locations within a specificorgan or adjacency to sensitive structures. Therefore, while focusing on aparticular region of the data, designated as a region of interest (ROI), contextualinformation surrounding that region is important. However, the same amount of detail is not required for the context and the ROI. Fuzzy C- means logic is used to separate out the Region of Interest from the whole image. After performing segmentation contourlet and wavelet transformsare applied to significant region(ROI) and to the whole image. Several algorithms likeEZW (Embedded Zerotree Wavelet)(Shapiro's, 1993) based on identifying image regions containing low coefficient values for all subbands, and code using a special symbol ,the SPIHT (Set Partitioning in Hierarchical Trees) (Said Pearlman 1996), and the SPECK (Set Partitioning Embedded Block)(Islam. A.Pearlman 1999) exists for image compression.The existence of tumors in MRI, CT scan is one of the most important signs, considered by radiologist is the biggest challenge in interpreting this huge amount of data precisely and effectively (Sebastien Piccand et al 2008). In this ROI technique, a special data structure that is based on a zerotree model is used to manipulate the region of interest moreeasily. The region of interest especially the tumor region in different modalities is of taken more care because of information content in that portion. If the regions of interests are usually of low contrast and noisy nature of the images, then preservation of image content in that region has been very difficult. Hence an image denoising and enhancement may be required to preserve the image quality, highlighting image features and suppressing thenoise (Osher and Rudin 1990, Gilboa et al 2004). These enhancement techniques make very small tumor structures into more visible and the features of the tumors are made more evident hence the radiologist can be aided to retain the information content of the image named as ROI.

III. EXISTING SYSTEM

The Lossy compression is a data encoding method which discards (loses) some of the data, in order to achieve its goal, with the result that decompressing the data yields content that

is different from the original, though similar enough to be useful in some way.It is possible to compress many types of digital data in a way which reduces the size of a computer file needed to store it or.

IV. PROPOSED SYSTEM

The LOCO-R algorithm works as LOCO-I and takes rectangular image block with 8 pixel values. The compressed image produced is a sequence of bits from which the original image can be reconstructed. Let wd be the image width and ht be the image height. Pixels are identified by coordinates (x, y) with x in the range [O, wd-l] and y in the range [O, ht-l].Lossless image compression is well-established as a means of reducing the volume of image data without compromising the data quality. LOCO-R (LOw ComplexityLossless Compression for Images) is the algorithm at the core of the new ISO/ITU standard for lossless and near-lossless compression of continuous-tone images, JPEG-LS.The aim is to reduce to increasethe speed at which images can becompressed. Currently, the available space-qualifiedhardware designed for lossless compression is based on the Rice Compression algorithm.

V. IMPLEMENTATION

Before starting the actual coding phase, it is highly important to understand the requirements of the end user and have an idea of how should the product looks like. The requirement specifications from the first phase are studied in this phase and a system design is prepared. System design helps in specifying hardware and system requirements and helps in defining the overall system architecture. The system design specifications serve as an input for the next phase of the model.

A. Divide the Image

Attempting to compress an entire image would not yield optimal results. Therefore, JPEG divides the image into matrices of 8 x 8 pixel blocks. This allows the algorithm to take advantage of the fact that similar colors tend to appear together in small parts of an image. Blocks begin at the upper left part of the image, and are created going towards the lower right. If the image dimensions are not multiples of 8, extra pixels are added to the bottom and right part of the image to pad it to the next multiple of 8 so that we create only full blocks. The dummy values are easily removed during decompression. From this point on, each block of 64 pixels is processed separately from the others, except during a small part of the final compression step.

B. Conversion to the Frequency Domain

At this point, it is possible to skip directly to the quantization step. However, we can greatly assist that stage by converting the pixel information from the spatial domain to the frequency domain. The conversion will make it easier for the quantization process to know which parts of the image are least important, and it will de-emphasize those areas in order to save space.

C. Quantization

Having the data in the frequency domain allows the algorithm to discard the least significant parts of the image. The JPEG algorithm does this by dividing each cosine coefficient in the data matrix by some predetermined constant, and then rounding up or down to the closest integer value. The constant values that are used in the division may be arbitrary, although research has determined some very good typical values. However, since the algorithm may use any values it wishes, and since this is the step that introduces the most loss in the image, it is a good place to allow users to specify their desires for quality versus size.

D. Entropy Coding

After quantization, the algorithm is left with blocks of 64 values, many of which are zero. Of course, the best way to compress this type of data would be to collect all the zero values together, which is exactly what JPEG does. The algorithm uses a zigzag ordered encoding, which collects the high frequency quantized values into long strings of zeros.

VI. ALGORITHM STEPS

The proposed method compression the original image by implements a number of steps Figure 2 represents the flowchart of the proposed method. The algorithm steps are:

Input: image (f) Output: compressed file

Begin

Initialize parameters

SET round to zero

READ image (f)

Convert (f) to gray scale

SET A = ( ) // set empty value to matrix A

A = image (f)

Bn = convert matrix A into binary

Out1 = Compress matrix by LZW algorithm function norm 2l zw (Bn);

Convert matrix compress by LZW into binary

Set N = 7, k = 4

WHILE (there is a codeword) and (round ≤3) xxx = the size of the (Out 1) remd = matrix size mod N; div = matrix size /N; FOR i = 1 to xxx-remd step N FOR R = i to i□ □N 1□ □ divide the image into blocks of size 7 save into pa-rametermsg = out 1 [R] END FOR R

c2 = convert (msg) to Galoris field; origin= c2

d2 = decoding by BCH decoder (bchdec (c2, n, k,)) c2 = Encode by BCH encoder for test bchenc (d2, n, k) IF (c2 == origin) THEN // original message parameter

INCREMENT the parameter test (the number of cod- eword found) by 1;

add the compressed block d2 to the matrix CmprsImg

add 1 to the map[round] matrix ELSE

add the original block (origin) to the matrix CmprsImg

add 0 to the map[round] matrix

ENDIF

END FOR

Pad and Add remd bits to the matrix CmprsImg and encode it

Final map file = map [round] to reuse map file in the iteration

FOR stp = 1 to 3

Compress map by RLE encoder and put in parameter map_RLE [stp] = RLE (map [stp])

END FOR stp

INCREMENT round by 1

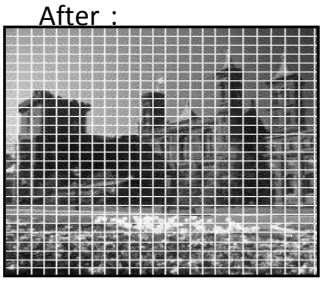
ENDWHILE

END

VII. RESULTS

Before :

After :



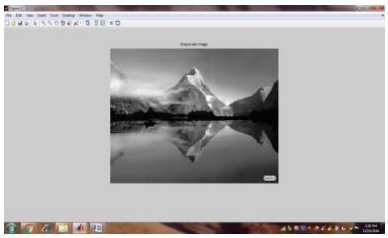


Figure 1: After Appling Dct

Figure 2: (a) INPUT IMAGE (b) GRAYSCALE IMAGE (C) AFTER APPLING DCT

Figure 3: Input Image

Figure 4: Input Image

Figure 5: Input Image

Figure 6: Grayscale Image

Figure 7: After Appling Dct

References

[1] R. F. Rice, Some Practical Universal Noiseless Coding Techniques, Part III,Module PSIl4,K+, JPL Publication 91-3, Jet Propulsion Laboratory, Pasadena,Califomia, November 1991.

[2] Consultative Committee for Space Data Systems, "CCSDS Recommendation for Lossless Data Compression", CCSDS 121.0-BI, Blue Book, issue I, May I 997.http://www.ccsds.orglblue books.html

[3] M. J. The paperinberger, GSeroussi, and G. Sapiro, "The LOCO-I Lossless ImageCompression Algorithm: Principles and Standardization into JPEG-LS", IEEETransactions on Image Processing, vol. 9, no. 8, pp. \309-\324, August 2000.

[4] Information Technology-Lossless and Near-Lossless Compression of Continuous-Tone Still Images, ISOIlEC 14495-1, ITU Recommendation T.87, 1999.

[5] M. J. The paperinberger, G. Seroussi, and G. Sapiro, "LOCO-I: A Low Complexity, Context-Based, Lossless Image Compress- ion Algorithm", Proc. of the 1996 DataCompression Conference (DCC '96), Snowbird, Utah, pp. 141-149, March 1996.

[6] M. Rabbani and P. Jones, Digital Image Compression Techniques, Bellingham, Washington: SPIE Publications, 1991. [7] S. W. Golomb, "Run-Length Encodings", IEEE Transactions on Information Theory, vol. IT-12, no. 3, pp. 399-401, July 1996.