International Journal of Electronics and Electrical Engineering

Volume 3 | Issue 1 Article 8

July 2014

LOSSLESS AND LOSSY IMAGE COMPRESSION BASED ON DATA FOLDING

GAYATRI. B. TIGADI
Department of Telecommunication Engineering , DSCE, Visvesvaraya Technological University Bangalore, India, gayatri.mp009@gmail.com

MANJULADEVI T.H.
Department of Telecommunication Engineering , DSCE, Visvesvaraya Technological University Bangalore, India, thm0303@gmail.com

LOSSLESS AND LOSSY IMAGE COMPRESSION BASED ON DATA FOLDING

GAYATRI.B.TIGADI[1] & MANJULADEVI.T.H.[2]

[1,2]Department of Telecommunication Engineering , DSCE, Visvesvaraya Technological University Bangalore, India
Email:gayatri.mp009@gmail.com, thm0303@gmail.com

Abstract: Image compression plays a very important role in image processing especially when we have to send the image on the internet. Since imaging techniques produce prohibitive amounts of data, compression is necessary for storage and communication purposes. Many current compression schemes provide a very high compression rates but with considerable loss of quality. On the other hand, in some areas in medicine, it may be sufficient to maintain high image quality only in the region of interest, i.e., in diagnostically important regions called region of interest. In the proposed work images are compressed using Data folding technique which uses the property of adjacent neighbour redundancy for prediction. In this method first column folding is applied followed by the row folding iteratively till the image size reduces to predefined value, then arithmetic encoding is applied which results the compressed image at the end before transmitting the data. In this paper lossless compression is achieved only at the region of interest and it is mainly suitable for medical images.

Keywords: *Image compression, Data folding, ROI*
.

1. INTRODUCTION

Data compression plays an important role in computer storage and transmission. The purpose of data compression is that we can reduce the size of data to save storage and reduce time for transmission. The same thing goes for image compression; namely, we can save storage by using data compression, data could be text, image, audio and video. In this I am dealing with images.

There are two main domains of image compression are:

- Lossless compression
- Lossy compression

The compression process is termed lossless (also reversible or noiseless) if the reconstructed image is identical to the original; otherwise, it is called lossy compression (also irreversible or noisy).Lossy compression is used for network related application and Lossless compression is used for storage related application [4].

The general principle of image compression algorithms is to transform binary digits into a new one that contains the same information but with fewer digits, so the file can be as small as possible. Any lossless coding system basically consists of three steps. They are: transformation, data-to-symbol mapping and lossless symbol coding [2]. Transformation converts the image data into a form that can be compressed more efficiently by further stages. But sometimes, it might be desirable to operate directly on the original data without incurring the additional cost of applying transformation; in this case transformation is set to the identity mapping. Data-to- symbol mapping converts the input data into symbols that can be efficiently coded by final stage. Lossless symbol coding generates a binary bit stream by assigning binary code words to the input symbol. The first two stages can be regarded as pre-processing stages for mapping the data into a form that can be coded more efficiently by the last stage.

The efficient image compression algorithm or any data compression is chosen according to scales, such as compression size, compression ratio, compression efficiency and computational complexity. The proposed method works fast for compressing the images which gives image compression with good compression efficiency and low computational complexity. The remainder of this paper is organized as follows:

Section II describes the concept of region of interest. Section III describes the proposed method, Section IV experimental results, Section V conclusion and future scope followed by references.

2. CONCEPT OF ROI

A Region of Interest, often abbreviated ROI, is a selected subset of samples within a dataset identified for a particular purpose. The concept of an ROI is commonly used in medical imaging. Medical imaging has a great impact on medicine, especially in the fields of diagnosis and surgical planning [11]. However, imaging devices continue to generate large amounts of data per patient, which require long-term storage and efficient transmission. Current compression schemes produce high compression rates if loss of quality is affordable. An approach that brings a high compression rate with good quality in the ROI is thus necessary. The general theme is to preserve quality in diagnostically critical regions while allowing lossy encoding of the other regions For example; the boundaries of a tumour may be defined on an image or in a volume, for the purpose of measuring its size. So in case of ROI we separate the region of interest from the complete image in which the user is interested.

3. PROPOSED METHOD

In the proposed method the compression idea is based on spatial resolution for lossless image compression called corrugation/ data folding [6, 7, 8, 9, 10]. The idea is to subtract even pixels from odd pixels and store the difference data in a buffer. Odd pixels are stored in another buffer for further iterations. In this Image compression method first column folding is applied followed by the row folding iteratively till the image size reduces to predefined value. In column folding, pixels used for subtraction are column adjacent whereas in row folding, the pixels are row adjacent. The pixel redundancies are rearranged in a tile format and Arithmetic encoding technique is applied which results the compressed image at the end before transmitting the data. The idea is to reduce the image size iteratively in terms of dimensions - rows or columns by 2. At the decoder, Arithmetic decoding is applied followed by data unfolding which is similar to data folding. Difference data thus obtained through all iterations is stored in the tile format, as shown in Fig. 1.

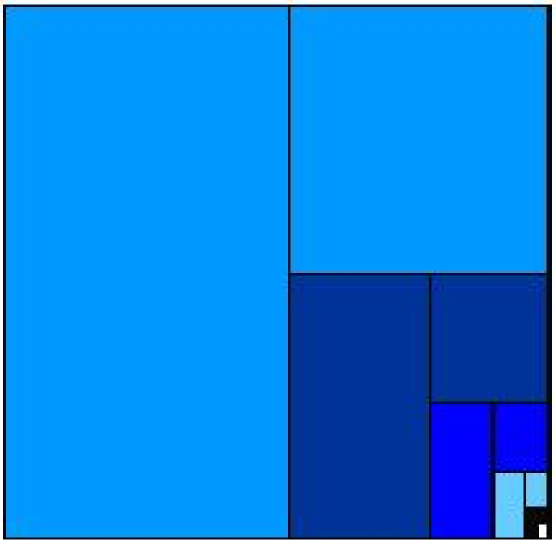

Fig. 1: Difference data arranged in the tile format

3.1 Encoder

3.1.1 Data Folding

Data folding is an iterative procedure, column folding followed by row folding, that is repeated at every image level. Original image (i.e. input image) must be square.

Define a buffer 'F' (also called image matrix) whose size is equal to that of original image. Original image is considered as input image for the first iteration. Initially, the buffer 'F' is empty. In column folding, odd columns of the input image are subtracted from its right adjacent even columns and stored in first half columns of the empty portion of the buffer 'F'. Odd columns are stored in a different buffer S' which is taken as input image to row folding. Following equations depicts column folding technique [1].

$F(X+a, Y+b) = S(a, 2b-1) – (a, 2b)$
$S'(a, b) = S(a, 2b-1)$
$a \in [1, W]$ and $b \in [1, W/2]$
Where,
S = input image
W = width of input image
S' = modified input image
X = starting x-coordinate of empty portion of 'F'
Y = starting y-coordinate of empty portion of 'F'

Row folding is similar to column folding. In row folding, odd rows are subtracted from its adjacent even rows and stored in first half rows of empty portion of 'F'. Odd rows are stored in a different buffer S'' which is taken as input to next iteration. Observe that, input image to column folding is always square whereas it is rectangular to row folding.

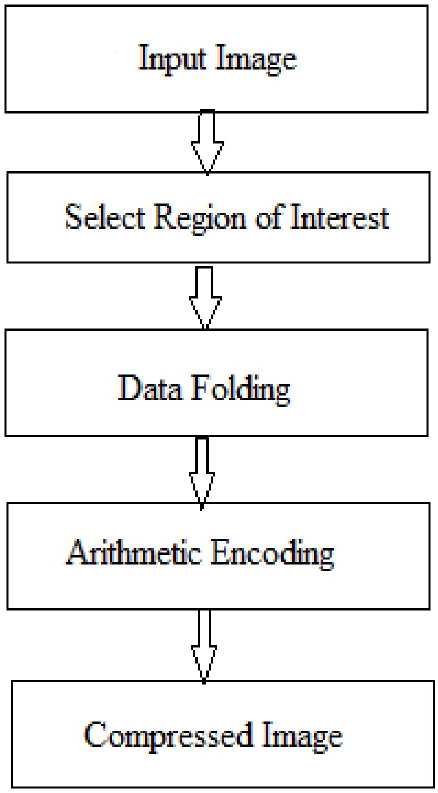

Fig 2: Flow chart of data folding

3.1.2 Coding:

This difference matrix can be entropy encoded by using various algorithms like Huffman Coding (HC), Arithmetic encoding (AC). I have encoded it using both Huffman algorithm and Arithmetic encoding and compared both encoding techniques in each level of iteration. The final result obtained after encoding the data of each level would be compressed data for the input image. The ratio of output compressed file to input image is the quantitative measure for the algorithm. The compression is measured as number of bits used per pixel and is calculated as input file size divided by sum of size of the output bit file (i.e. the matrix obtained after source encoding), sum of sizes of the dictionaries and y bits (to send the iterations to the decoder) [7].

$$\text{Compression Ratio} = \frac{I}{I' + D + Y}$$

Where,
I = Raw image size (image size at start)
I'= Size of output bit file
D= Sum of sizes of all dictionaries
y= no. of bits required to encode no. of iterations

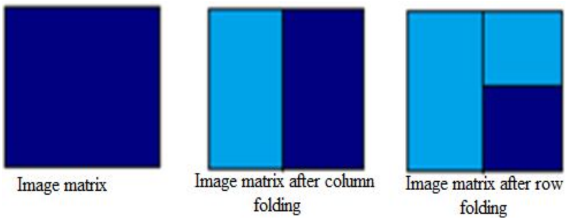

Fig 3: Illustration of data folding for the first iteration

3.2 Decoder

The decoder will perform Arithmetic decoding followed by data unfolding. Data unfolding is an iterative procedure similar to data folding; row unfolding followed by column unfolding that is repeated at each image level. Here, no. of iterations is equal to that of data folding. Matrix elements which were folded in higher iterations during encoding will be unfolded first. Data folding starts with the first element of the image whereas data unfolding starts with bottom right elements of the image.

The procedure is illustrated for a 16x16 image in Fig. 4. Assume that only 3 levels of folding is applied at the encoder side.

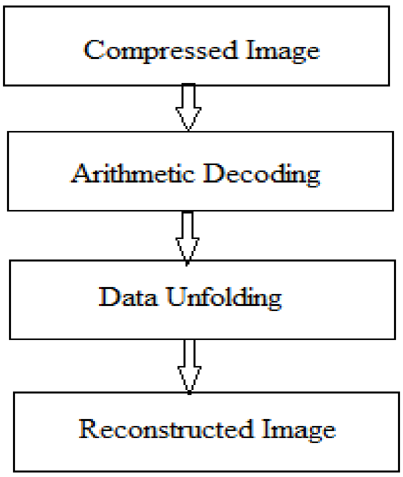

Fig 4: Flow chart of data Unfolding

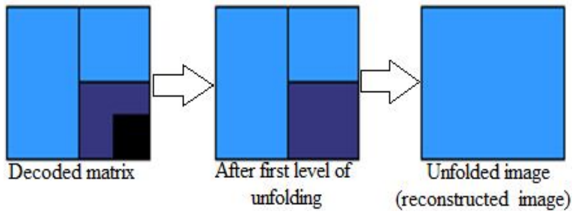

Fig. 5: Illustration of data unfolding algorithm

4. EXPERIMENTAL RESULTS

First Input image size is calculated based on number of rows and columns, then maximum possible iteration is calculated using k= log2 (min value between rows and column) = 8, after this encoding method is applied, which will give the following information of the image. Here I have compared results from both encoding techniques (i.e, Huffman and arithmetic encoding methods)

Total bits of input image = 256*256*8 = 5, 24,288

- The maximum iteration is 8
- Level = 4
- Total bits after Arithmetic compression = 65,821
- Total bits after Huffman compression = 93,633
- BPP = 1.428726
- The PSNR is 5.543254

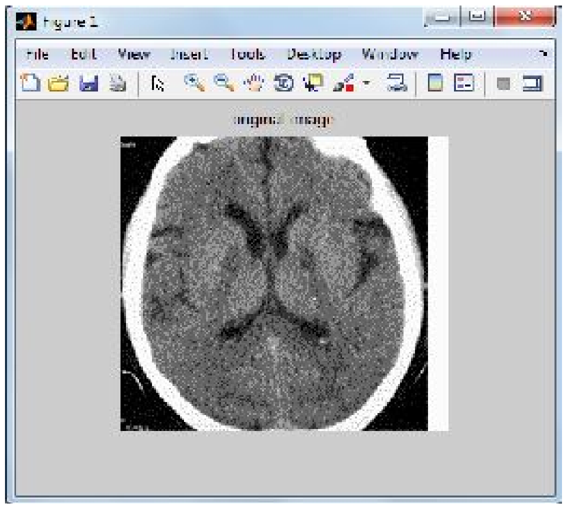

Fig 6: Original image

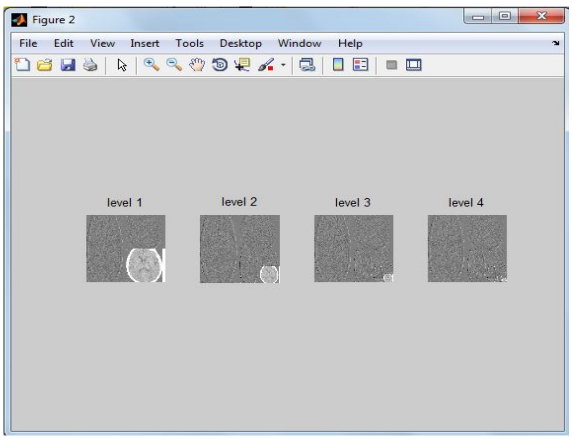

Fig 7: Levels of folding

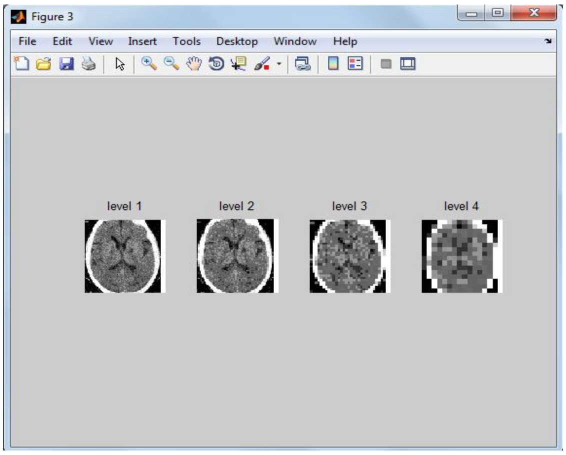

Fig 8: Extracted low frequency coefficients of fig 7

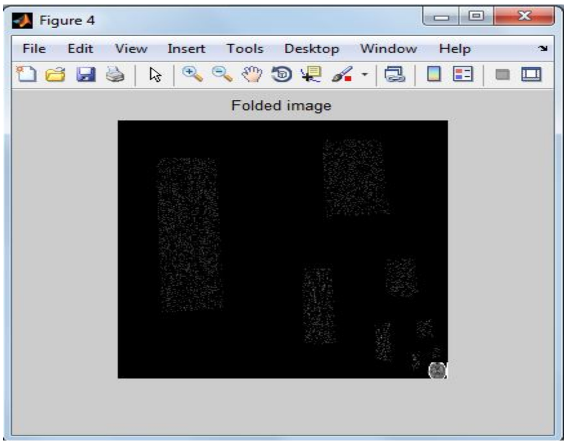

Fig 9: Folded image

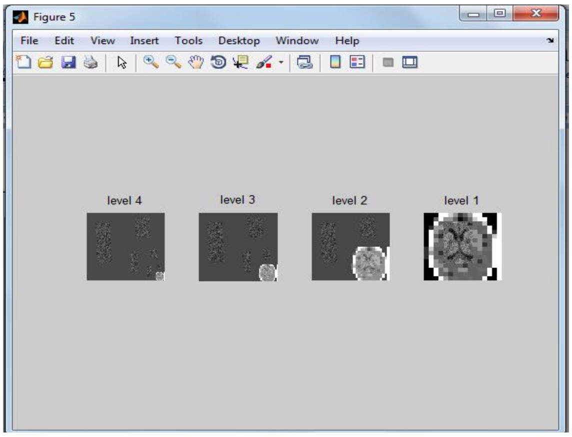

Fig 10: Levels of unfolding

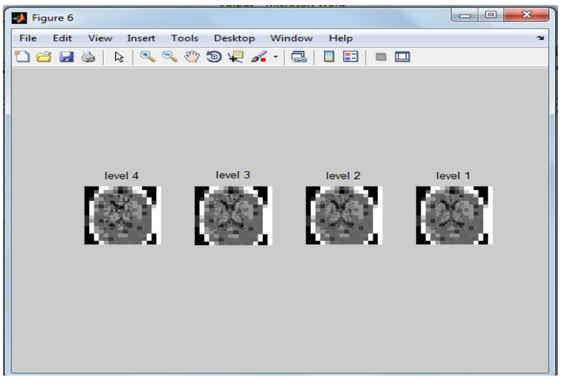

Fig 11: Extracted low frequency coefficients of f ig 10

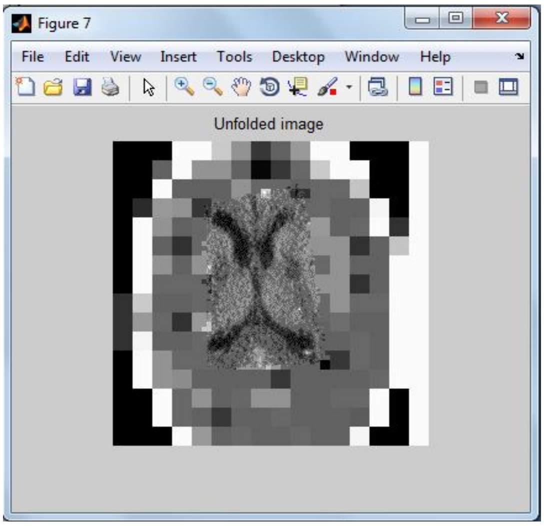

Fig 12: Unfolded image (Reconstructed image)

Total number of bits obtained after both Huffman and Arithmetic encoding techniques is as shown below for which input image chosen is cameramen.jpeg of resolution 256*256.
Input image = 256*256*8 = 5, 24,288

Table 1: Comparison of total number of bits between Huffman and Arithmetic coding

Level of iteration	Total number of bits after compression using Huffman coding	Total number of bits after compression using arithmetic coding
1	263895	199969
2	178591	156878
3	153357	143086
4	146128	139233
5	144122	138269
6	143604	138042
7	143486	137996
8	143461	137988

5. CONCLUSION AND FUTURE SCOPE

Data Folding is an algorithm for lossless image compression at the region of interest and lossy compression means quality of image degrades in the rest part of the image. It is simple and faster method which gives good bpp (bits per pixel) value, high PSNR and lower computational complexity. Arithmetic coding is better than the Huffman coding because it gives fewer bits as level of folding increases. It works comparatively better for smooth images. The proposed work is mainly suitable for medical images.

The proposed data folding algorithm can be implemented using transformation technique such as DCT, DWT etc where the computation complexity is affordable. This results high compression ratio and achieves lossless compression for complete image.

REFERENCES

1. Suresh Yerva, Smita Nair, Krishnan Kutty, *"Lossless Image Compression based on Data Folding"*, IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011.
2. Alan Conrad Bovik, The Essential guide to Image Processing, vol. 16, pp. 385-419, 2009.
3. Marcelo J. Weinberger, Senior Member, IEEE, Gadiel Seroussi, Fellow, IEEE, and Guillermo Sapiro, Member, IEEE *"The LOCO-I Lossless Image Compression Algorithm:Principles and Standardization into JPEG-LS"*, VOL.9, NO. 8, AUGUST 2000.
4. R.Sudhakar,Ms R Karthiga, S.Jayaraman, "Wavelet Based Image Compression using SPIHT algorithm" ICGST-GVIP Journal, Volume (5), Issue(6), June 2005.
5. Amir Said, Member, IEEE, and William A. Pearlman, Senior Member, IEEE, *"An Image Multiresolution Representation for Lossless and Lossy Compression"*, VOL. 5, NO. 9, SEPTEMBER 1996.
6. Bin Zhu, Mitchell D. Swanson, and Ahmed H. Tewfik, *"Image Coding by Folding"*, 1997 IEEE.
7. Alan Conrad Bovik, The Essential guide to Image Processing, vol. 16,pp. 385-419, 2009.
8. Chin-Chen Chang, *Chi-Shiang Chan and **Ju-Yuan Hsiao *"Lossless Image Compression Based on Two-Way Smaller Difference"*, 2003 IEEE.
9. Michael Ciavarella Alistair Moffat ,*"Lossless Image Compression Using Pixel Reordering"* Department of Computer Science and Software Engineering The University of Melbourne, Australi
10. Bruno carpentieri, member, IEEE, Marcelo j. Weinberger, senior member, IEEE, and Gadiel seroussi, fellow, IEEE, *"lossless compression of continuous-tone images"*, proceedings of the IEEE, vol. 88, no. 11, November 2000.
11. Yair Wiseman, "*Burrows-Wheeler Based Jpeg"*, Data Science Journal, Volume 6, 5 March 2007.
12. Balpreet Kaur, Deepak Aggarwal, Gurpreet Kaur, Amandeep Kaur, "Efficient Image Compression based on Region of Interest", IJCSt vol . 2, issue 1, march 2011.

◈◈◈