Far East Journal of Electronics and Communications
100 *Proceedings of SHANNON* - 3rd *Int. Conf. Comp. Sci.*, 2016

http://dx.doi.org/10.17654/ECSV3PII16493
Special Volume 3, Part II, 2016, Pages 493-503 **ISSN: 0973-7006**

FRAMEWORK OF ADAPTIVE BLOCKWISE LOSSLESS IMAGE COMPRESSION

Deepa Raj and Seema Gupta

Baba Saheb Bhimrao Ambedkar University
Lucknow, U. P., India

Abstract

Bandwidth, time and power are three important constraints of image communication system which holds transmission of uncompressed image data. Dictionary based compression and model based compression are two main approaches of adaptive lossless image compression. Dictionary based image compression is known as LZ coding. This paper is based on model based adaptive Huffman coding where the word adaptive comes from dynamically updating. We have already worked with adaptive Huffman lossless text compression based model. We discuss adaptive lossless image compression to reduce bit errors and to calculate compression ratio of the image dynamically.

1. Introduction

In today's digital world, storage and transmission of image data are two prime factors for compression techniques. Compression efficiency is directly proportionate to the quality of the image. Lossless image compression retains the source data which is more important than compression efficiency [5]. Main applications of lossless image compression are in medical, astronomical, scientific, professional video processing application images,

Conference held during April 8-9, 2016 at Lovely Professional University, Punjab, India.
Guest Editor: G. Geetha; Division of Research and Development, Lovely Professional University, Punjab, India.

etc. [4]. Lossless image compression technique utilizes adaptive block size to obtain high compression ratios in neighboring pixels [8]. Image is divided into several subblocks that have interpixel redundancy due to correlations between the pixels in the image. Adaptive lossless image compression is a user defined set of functions that allows the transmission of data. The level of compression is constantly adapted according to the platform provided. Image compression is a viable approach towards preserving energy by reduction of the height. Adapting refers changing the compression level during the transmission. Compression time is directly proportionate to the compression level.

The concept of three entropy coding methods in lossless image compression using digitized radiographs provides a speed of 4 to 5 bits per pixel bitrates. Pixels of lossless coding techniques are used to magnetic resonance images. Later on, prediction, linear transformation, and multiresolution methods for non-correlated data got invented. Chu [2] proposed a lossless (LS) image compression technique combining a prediction step with the integer wavelet transform. The prediction step proposed in this technique is a simplified version of the median edge detector algorithm used with JPEG-LS. First, the image is transformed using the prediction step and a difference image is obtained. The difference image goes through an integer wavelet transform and the transform coefficients are used in the lossless codeword assignment. The algorithm is simple and test results show that it yields higher compression ratios than competing techniques. Computational cost is also kept close to competing techniques. Oh et al. [3] gave visually lossless image compression method aimed at to compression images while ensuring the compression distortions below perceptible levels. A pseudo lossless image compression which modifies the noise component of the bit data to enhance the compression without affecting the image quality is proposed in [6].

Section 2, in this paper, describes lossless image compression methods. Section 3 deals with framework for adaptive lossless compression blockwise. Section 4 tells about statistical analysis, and Section 5 about experimental performance.

2. Lossless Image Compression

Grey image of size $512 * 512$ and about .5Mb of disk space need approximately 7 minutes to transfer from one end to another end using 16Kbps high speed. To store these images and make them available after transmission, compression techniques are needed. Size reduction process is mainly based on the removal of redundant data. If the image compression ratio is $10:1$, then the space to storage reduced to 300kb and the time of transmission will drop to less than 2 minutes. Hence, it may be concluded that to reduce storage space and transmission time, image compression is needed. Compression method is applicable for images, audio and video compression. It also saves storage space and provides a higher level of security and monitoring. The lossless decompressor is inversely proportional to lossless compressor. In lossless image compression, medical images can be compressed to about half of their original size without destroying the data. Lossless image compression methods provide no loss in compression images, i.e., both encoding and decoding are the same. Compression ratio is the main measurement of lossless image compression. Other factors are bit rate, mean square error and the sum of absolute difference. Bit rate due to its independence of the data storage. A bit rate defines the average pixel of bits used to represent each pixel of the image in a compressed form. Bit rates are measured in bits per pixel, where a lower bit rate corresponds to a greater amount of compression. Lossless compression methods are divided into four categories: run length coding, lossless predictive coding, entropy coding and the development arithmetic coding. Digital image processing consists of three types of redundancies: coding, interpixel and psychovisual redundancy.

3. Framework of Adaptive Blockwise Lossless Image Compression

We describe here the complete framework of adaptive lossless compression blockwise. First we explain the propos method of creation of adaptive Huffman tree by which code words of particular pixel has lesser pixel of bits. We also used a block wise data of fixed size of the data file to compress. We explain complete algorithm and methods to compress the image file. There are two types of compressors in lossless data compression, namely, statistical coders and dictionary based compression. Model of statistical based coder executes block dynamically and replaces single characters in the data block with a variable length code. Dictionary based compression create lookup table of the string in terms of code words. We explain the first method of compression to generate the code word of each pixel for generating compressed data.

3.1. Proposed adaptive tree

Pixel is the smallest unit of the image. All pixels have 8 bits for grey image of size 75×75. Total space acquired by this image is $75 \times 75 \times 8$ $= 5625$ bytes $= 5.49$kb.

Original image Compressed image Decompressed image

Figure 1. Boat (75×75).

For compression of the image, we divide the image in blocks of 3×3 size which is displayed in Table 1. In this way, for compression of whole image we need total 25 blocks.

Table 1. Image has a block of size $3 * 3$

126	128	132
126	130	131
126	130	131

In this table, 9 pixels needed 72 bits to store, (8 bits for each pixel), but by using the proposed tree, we can reduce pixel of bits for each pixel using coding redundancy.

Tree is generated without using not yet to transmitted node (NYT). There are two types of nodes, namely, internal and external. External nodes as a leaf node contain four information: name of pixel, weight of pixel and two address parts. Internal node contains only weight. Initially, pixel 126 is scanned, root node is created and pixel name as '126' and weight are provided as 1 because it is the first pixel in the blockwise matrix. Next pixel is '128' since this pixel is not present in the tree so root node of tree has left side null so '128' node is attached at left hand side. Next pixel is '132'. Since this pixel is not present in the tree and left hand side has less weight, so new node is created as the root node of a subtree as internal node and new node attached at the left hand side. Next pixel is again '126', so weight of the node is incremented by 2 and root node is created and '126' node is at the right hand side. Next pixel is '130'. Since this pixel is not present in the tree, right child of root node and left child of root node have same weight and right child is external node, so node '130' is attached at the left hand side of root of subtree at right side of the root node of the tree. Next pixel is '131'. Since this pixel is not present in the tree, so new node is created through external node at the left side of the root node of the tree because of lesser weight is at the left hand side. Apply this procedure for each and every pixel of the blockwise matrix. The node which is attached at the left side of the parent is given the code word 0, while the node which is attached at the right side of the parent is given the code word 1. In this way, code words of each character can be determined.

3.2. Adative blockwise lossless compression method

Adaptive compression takes one pass to do encoding and decoding simultaneously unlike the Huffman compression. Huffman compression is static in nature, reads whole file, generates the code words of pixel, i.e., encoding and after that scans one by one pixel of image file then do the decoding. It takes lots of time during encoding but adaptive Huffman compression is dynamic in nature, reads one by one pixel, generates the adaptive Huffman tree both the sides and simultaneously encoding and decoding take place. In this compression method, we use a medium to store the online data for some small time to generate its variable length binary code and then send toward receiver to reduce transmission time and its bandwidth. Image data is taken as blockwise of some fixed length because Huffman tree is generated up to this block and for the next block regenerate the Huffman tree for reducing the searching time of the pixel present in the block. First pixel is ‘126’ before sending, it is searched in the Huffman tree, if not found, then send the 8bit code towards its receiver side and decode at receiver side. Then update the Huffman tree by inserting in the tree while satisfying sibling property. Next another pixel is 128, not found in the tree, then send the 8bits code towards its receiver side and modify the Huffman tree by inserting in the tree while satisfying sibling property. Another is ‘130’ not found in the tree send 8bits code and then update in the tree by proposed method. Again next pixel is ‘126’ found then update the Huffman tree, i.e., increment the weight of node, while satisfying sibling property. Same method is applied to complete sending of whole block of size 72kb. For sending another block, code word is determined on the basis of the adaptive Huffman tree if it occurs in the tree. Simultaneously regenerate the adaptive Huffman tree beginning from the empty tree and remove the old adaptive Huffman tree. In this way, searching time will reduce because height of the tree is reduced due to reinitializing the Huffman tree. Encoding and decoding will take place simultaneously because Huffman tree is updated automatically in both the sender side and the receiver side. So, in one pass, encoding and decoding takes place before next data is sent.

Figure 2 shows the framework of compression of one block of image file. If the probability of occurrence of repetition of character is high, then this algorithm gives better result.

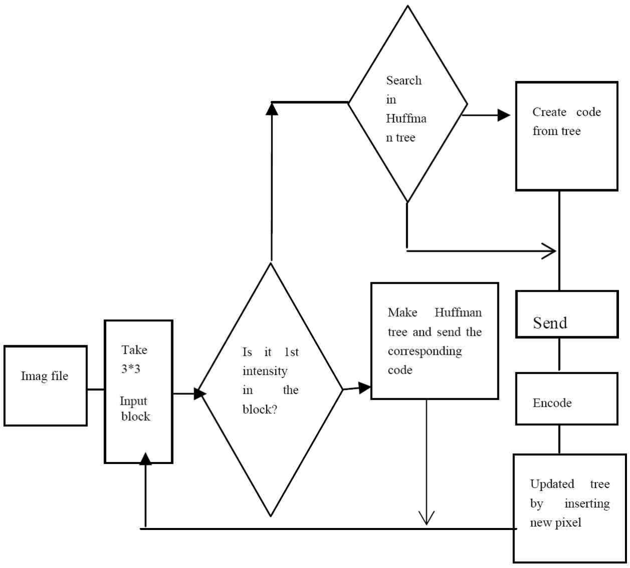


Figure 2. Framework of adaptive image compression for one block.

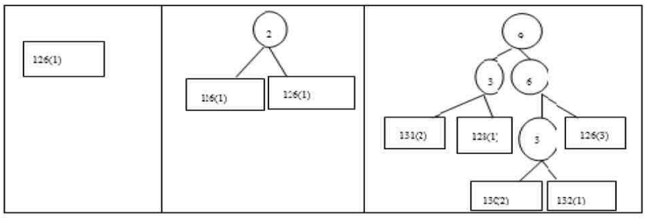


Figure 3. Formation of proposed tree in block 1.

Applying proposed method on the 3×3 block of an image in which each pixel has 8bits, we get the corresponding bits through the adaptive Huffman tree. Process as is depicted in Figure 4.

126(8)	128(8)	132(8)
126(8)	130(8)	131(8)
126(8)	130(8)	131(8)

Apply proposed adaptive →

126(8)	128(8)	132(8)
126(2)	130(8)	131(8)
126(2)	130(3)	131(2)

Total bits before compression $= 3 * 3 * 8 = 72.$

Total bits after compression $= 8 + 8 + 8 + 2 + 8 + 8 + 2 + 3 + 2 = 49.$

Same process is applied for 2nd block to 25th block of the image of size 75×75.

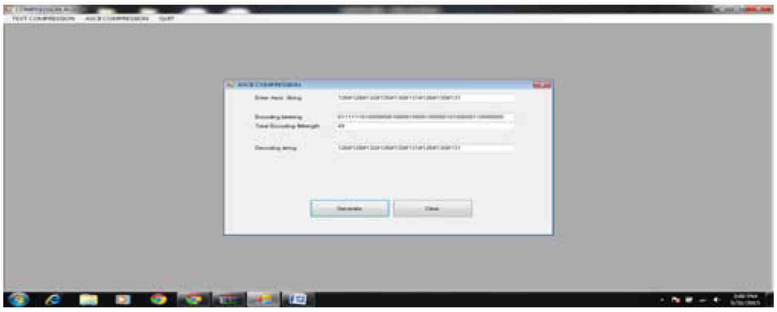


Figure 4.

4. Statistical Analysis of Lossless Image Compression

Lossless image compression analysis is based on three main parameters: compression ratio, mean square error and peak signal to noise ratio. Compression ratio is the ratio of total bits of the image before compression and total bits of the image after compression. For the lossless image compression, MSE is zero and PSNR value is calculated from MSE. So it gives infinite value. Equations of compression ratio, PSNR and MSE are given below:

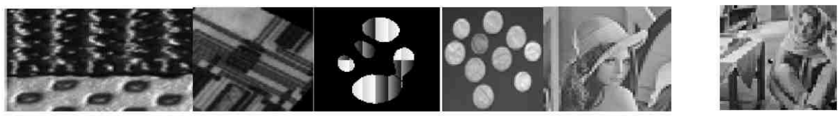
Compression ratio = before compression/after compression (1)

$$\text{MSE} = \frac{1}{mn\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[r(i,\ j) - K(i,\ j)]} 2, \tag{2}$$

$$\text{PSNR} = 10\log_{10}[65025/\text{MSE}]. \tag{3}$$

5. Experimental Performance

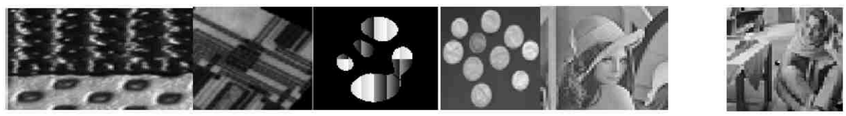
For experimental verification of digital image compression, we collected 6 different images of size 75×75 and applied the above adaptive compression on those images. Table 1 shows compression ratio, mean square error and peak signal to noise ratio. Experimental result shows that compression ratio is nearly equal to 2.



(a) (b) (c) (d) (e) (f)

Original images (75×75)

Compressed images



Decompressed images

Figure 5. Original, compressed and decompressed images.

This method compresses the image data half of that of the original size of the image data. Mean square error of all images is zero because the reconstructed image is the same as the original image due to lossless image compression. Another parameter is peak signal to noise ratio which has infinite value because it is inversely proportional to the mean square error.

Table 2. Performance analysis of grey scale image based on the compression ratio

Images	**Total bits before compression**	**Total bits after compression**	**Compression ratio**
Boat $(75 * 75)$	45000	23000	1.95
Lena $(75 * 75)$	45000	24000	1.875
Barbara $(75 * 75)$	45000	23500	1.91

Texture (75 ∗ 75)	45000	24500	1.83
Coins (75 ∗ 75)	45000	23000	1.95
Circuits (75 ∗ 75)	45000	23670	1.90

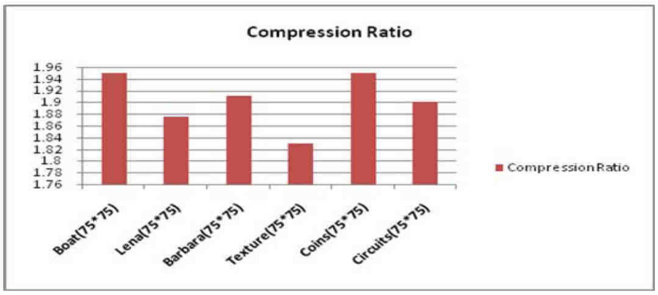


Figure 5. Compression ratio of all images.

6. Conclusion

By applying proposed method on the pixel of images, we found that compression ratio of all the images comes out approximately to 2.0. Size of all images is reduced up to half of the size of images. This method is dynamic in nature so block by block image is sent over the internet with calculated reduced bit. Simultaneously encoding and decoding process is in one pass. We found decompressed image block by block which reduced the processing as well as transmission time. More redundancy in the image makes the compression ratio large. In future, we will apply the same procedure for video images with some changes in this method.

References

[1] M. J. Weinberger, G. Seroussi and G. Sapiro, The LOCO-I lossless image compression algorithm: principles anstandardization into JPEG-LS, IEEE Transactions on Image Processing 9 (2000), 1309-1324.

[2] W. C. Chu, On lossless and lossy compression of step size matrices in JPEG coding, International Conference on Computing, Networking and Communications, 2013, pp. 103-107.

[3] H. Oh, A. Bilgin and M. Marcellin, Visually lossless encoding for JPEG2000,

IEEE Transactions on Image Processing 22 (2013), 189-201.

[4] K. Srinivasan, J. Dauwels and M. Reddy, Multichannel EEG compression: wavelet-based image and volumetric coding approach, IEEE Journal of Biomedical and Health Informatics 17 (2013), 113-120.

[5] K. Rajakumar and T. Arivoli, Implementation of multiwavelet transform coding for lossless image compression, International Conference on Information Comm. and Embedded Systems, 2013, pp. 634-637.

[6] Tzong-Jer Chen and Keh-Shih Chuang, A pseudo lossless image compression method, The 3rd International Congress on Image and Signal Processing, 2010, pp. 610-615.

[7] Y. F. Tan, W. N. Tan and K. Y. Tan, Lossless image compression utilizing reference points coding, Proceedings of the 3rd International Conference on Digital Image Processing, 2011, pp. 80092-E: 1-5.

[8] Michael D. Adams, The JPEG-2000 still image compression standard, Technical Report No. N2412, ISO/IEC JTC 1/SC 29/WG, Dept. of Electrical and Computer Engineering, University of Victoria, Canada, 2006.

[9] M. M. Fouad and R. M. Dansereau, A lossless image compression technique by combining integer wavelet transform with a simplified MED algorithm, International Conference on Electrical and Computer Systems, 2012, pp. 1-6.

[10] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Prentice-Hall, New Jersey, 2002.