Journal of Computer Science 5 (5): 388-391, 2009
ISSN 1549-3636

Binary Merge Coding for Lossless Image Data Compression

[1]N. Subhash Chandra, [1]M. Bala Raju, [1]M. Arya bahanu, [2]B. Raja Vikram,
[3]A. Govardhan and [4]S. Mahaboob Basha
[1]Department of Computer Science and Engineering,
Jawaharlal Nehru Technological University, Hyderabad, AP., India
Department of Electronic and Communication Engineering,
Vidya Jyothi Institute of Technology, Hyderabad, AP., India
[2]Department of Computer Science and Engineering, College of Engineering,
Jawaharlal Nehru Technological University, Hyderabad, AP., India
[3]Department of Computer Science and Engineering,
Al-Habeeb College of Engineering, Chevella, Ranga Reddy

Abstract: Problem statement: Image processing applications were drastically increasing over the years. In such a scenario, the fact that, the digital images need huge amounts of disk space seems to be a crippling disadvantage during transmission and storage. So, there arises a need for data compression of images. **Approach:** This study proposed a novel technique called binary merge coding for lossless compression of images. This method was based on spatial domain of the image and it worked under principle of Inter-pixel redundancy reduction. This technique was taken advantage of repeated values in consecutive pixels positions. For a set of repeated consecutive values only one value was retained. **Results:** The proposed binary merge coding achieved the compression rate of the brain image was 1.6572479. Comparatively, it is 100% more than the compression rate achieved by standard JPEG. **Conclusion/Recommendations:** This technique was simple in implementation and required no additional memory area. The experimental results of binary merge coding were compared with standard JPEG and it showed that, the binary merge coding improved compression rate compared to JPEG. The same algorithm can be extending to color images. This algorithm can also used for lossy compression with few modifications.

Key words: Huffman coding technique, JPEG, bit plane, data table

INTRODUCTION

Data Compression is a technique of encoding information using fewer bits than an un encoded representation would use through specific encoding or compression algorithms[2,4]. All forms of data which includes text, numerical and image contain redundant elements. Through compression, the data can be compressed by eliminating the redundant elements

The History of image data compression started probably about a half of century ago with the works on predictive coding and variable length codes[6]. The technological breakthrough that took place in 60’s, 70’s and 80’s resulted in efficient compression algorithms[8] that have been standardized in early 1990’s and currently are in common use together with the improvements achieved during the last decade. These advances have brought substantial increase in efficiency of earlier basic techniques. Nevertheless, the last decade was also a period of strenuous search for new technologies of image data compression[7].

Image compression technique is divided into two major categories[3], which lossless compression technique and Lossy compression technique. In lossless compression, no information is lost and the decompressed data are identical to the original un-compressed data. While in Lossy compression, the decompressed data may be an acceptable approximation to the original un compressed data[5].

This study focuses Loss less compression and proposed Binary merge coding Technique which works under the principle of removing Inter-pixel redundancy in spatial domain of the image

Corresponding Author: N. Subhash Chandra, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Hyderabad, AP., India

MATERIALS AND METHODS

Proposed method: Binary Merge Coding is based on spatial domain of the image and is suitable for compression of medical images [1]. The main objective of this technique is to take advantage of repeated values in consecutive pixels positions. For a set of repeated consecutive values only one value is retained.

In the binary merge coding two codes are used to build the bit plane[1]. The codes are as given below:

- Code 1 (one) is used to indicate that current pixel is different from previous pixel. In this case the current pixel is moved to the data table
- Code 0 is used to indicate that the current pixel is exactly same as previous pixel. This eliminates the storage of current pixel

After generating and merging the Bit Plane and data table, Huffman coding is applied to generate final form of compressed file.

In Binary Merge Coding Compression and Reconstruction model, as shown in Fig. 1 and Fig.2 added one popular Coding Redundancy technique Huffman Coding to get good Compression rate over Binary Merge Coding Technique alone. The results given in table 1 are achieved as per this model to reach real need.

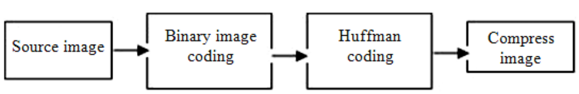

Fig. 1: Binary merge coding compression model

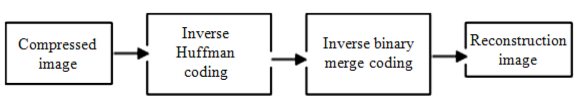

Fig. 2: Binary merge coding reconstruction model

Compression algorithm for binary merge coding:

```
Procedure binary_ Merge // main procedure
Begin
        // Generates bit plane and data tables
        call BINARY_DATA_COMPRESS()
        // Merges the Bit Plane and Data table

        call BINARY_DATA_MERGE ()
        // Generates the final form of compressed file
        call HUFFMANCODE()
End
```

Procedure binary_ data _compress ():

```
//subroutine to generate bit plane and data
     /* Data Items Used */

            prev_pixel // holds previous pixel
            cur_pixel // holds current pixel
            bit_plane /*8 bit number to hold the status
            bits to indicate whether pixel is retained or
            not retained. */

Begin
          open raw image file
          open bitplane file
          open data table file

          cur_pixel = read (image)
          write cur_pixel to data table file
          append bit 1 to bit_plane
          prev pixel = cur pixel
          while((cur_pixel = read(image))! = eof)
        Begin
          /* if repeated consecutive pixel value append 0
             to bit plane to
             indicate that pixel duplicate so not retained */
          if (cur_pixel = prev_pixel) then
             append bit 0 to bit planee
          else

Begin
          /*otherwise append 1 to bit plane to indicate
          that pixel is different so retained */
       append bit 1 to bit_plane
       write cur_pixel to datatable file
       prev_pixel=cur_pixel
   End

 if bit_plane is full then
       write bit_plane to bitplane file

End
       if bit_plane not empty then
             write bit_plane to bitplane file
       close raw image file
       close bitplane file
       close data table fil
End
```

Procedure binary data merge ():

```
/*To merge Bit Plane and Data table files and generate
intermediate compressed file */
         /* Data Items Used */
         cur_byte
```

```
Begin
        open bitplane file
        open data table file
        open bpds file
        while ((cur_byte = read(bitplane file))! = eof)
        Begin
                write cur_byte to bpds file
        End
        while ((cur_byte = read(data table file))! = eof)

        Begin
                write cur_byte to bpds file
        End
        close bitplane file
        close data table file
        close bpds file
End
```

Reconstruction of the image in binary merge coding: In the reconstruction of the image, first the intermediate file is generated from the compressed file. The bit plane and data tables are extracted from the intermediate file. By checking each bit of bit plane either a fresh byte from the binary plane is read and written to the reconstructed image file or earlier byte itself is written based on the current bit checked.

Reconstruction algorithm for binary merge coding:
Procedure binary_ merge_reconstruction:

```
Begin
        // To retrieve intermediate file from Huffman
        format call INVERSE_HUFFMANCODE()

        // To separate the Bit Plane and data tables
        call BINARY_DATA_DEMERGE ()

        // To build original image from Bit Plane and
        Data table call BINARY_DATA
        _DECOMPRESS()
End
```

Procedure binary data demerge ():

/* Subroutine to separate the Bit Plane and data tables

```
        // Data Items
        left// holds the no of bits in the last byte of the bit
        plane
        bpcount // holds no of bytes of bit plane
        cur_ byte

Begin
        open bitplane file
        open data table file
        open bpds file
        left = read(bpds file)
        bpcount = read(bpds file)

        for i = 1 to bpcound
        Begin
            cur byte = read(bpds file)
            write cur byte to bitplane file

        End

        while ((cur_byte = read(bpds file)! = eof)

        Begin
                write cur byte to datatable file
        End
        close bitplane file
        close data table file
        close bpds file

End
```

Procedure binary_data_ decompress ():

// Subroutine to build original image from Bit Plane and data table

```
        //Data Items
        cur_ pixel
        bit plane
        aBit//the current bit of current bit plane

Begin
        open bitplane file
        open data table file
        open image file.
        cur_pixel = read(data table file)
        while((bit_plane = read(bitplane file))! = eof)

        Begin
                for i = 1-8

                Begin
                        move ith bit of bitplane to aBit
                        // read fresh byte/pixel only
                        when the bit is 1
                        if aBit = 1 then

                            Begin
                                    cur pixel = read(data
                                    table file)
```

```
                End
                write cur_pixel to image file
            End
        End
        close raw image file
        close bitplane file
        close data table file
End
```

RESULTS AND DISCUSSION

The brain image which is taken as one sample source image, its histogram and statistical information are as shown in the Fig. 3. The histogram gives the distribution of the pixels in the range 0-255.

The generated results after executing Binary Merge Coding are shown in the Table 1. The memory requirement for BMC technique is very less compare to JPEG, because the processing is done byte by byte. In the case of JPEG the entire image needs to be brought into memory. As per as the process complexity is concerned, the Binary Merge Coding is simple to implement comparatively JPEG. The graph in Fig. 4 is drawn as the results shown in the Table1.

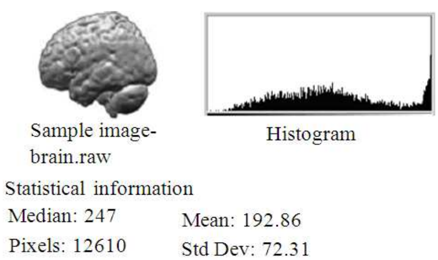

Fig. 3: The sample image for test

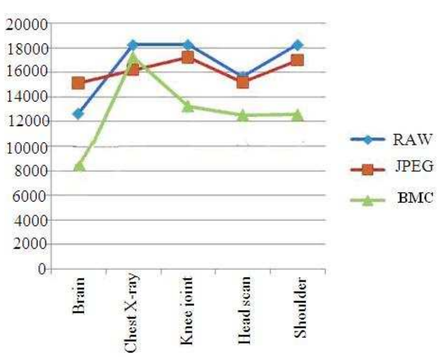

Fig. 4: Graph for the comparison between file sizes of BMC and JPEG compression techniques

Table 1: The size and compression rate of BMC and JPEG compression techniques

Image name	RAW Size	JPEG Size	JPEG Comp rate	BMC Size	BMC Comp rate
Brain	12610	15109	0.8346019	7609	1.6572479
Chest X ray	18225	16180	1.1263906	17207	1.0591619
Knee joint	18225	17193	1.0600244	13245	1.3759909
Head Scan	15625	15184	1.0290437	12532	1.2468081
Shoulder	18225	16962	1.0744606	12562	1.4508840

CONCLUSION

The compression rate of Binary Merge Coding is better than JPEG in medical images

The reconstructed image matches 100% with the original image because this is loss less Image compression technique.

The memory requirements for processing the images in this technique are significantly less compared to JPEG. The JPEG technique requires more memory because the entire image needs to be brought into memory. But for the Binary Merge Coding some sizable amount memory is required, because they process the image pixel by pixel.

The Binary Merge Coding is also applied for color images, which is producing equally good results in comparison with monochrome images.

REFERENCES

1. Subhash Chandra, N., 2009. An Hybrid approach of Lossless compression for Color Images. Asian J. Inform. Technol., 8: 472-472.
2. Skodras, A., C. Christopoulis and E. Ebrahimi, 2001. The JPEG2000 still image compression standard. IEEE. Signal Process. Mag., 18: 36-58. DOI: 10.1109/79.952804
3. Wallace, G.K., 1992. The JPEG still picture compression standard. IEEE. Trans. Consum. Elect., 38: xviii-xxxiv. DOI: 10.1109/30.125072
4. Jayant, N., 1992. Signal compression: Technology targets and research directions. IEEE. J. Select Areas Commun., 10: 796-818. DOI: 10.1109/49.138986
5. Rafael, C., G. Richard and E. Woods, 2007. Digital Image Processing. 3nd Edn., Pearson Education, Printice Hall, ISBN: 013168728X, pp: 954.
6. Huffman, D.A., 1952. A method for construction of min redundancy codes. Proc. IRE., 40: 1098-1101. DOI: 10.1109/JRPROC.1952.273898
7. Taskin, C. and S.K. Sarikoz, 2008. An overview of image compression approaches. Proceeding of the 3rd International Conference on Digital Telecommunications, June 29-July 5, IEEE Xplore Press, Bucharest, pp: 174-179. DOI: 10.1109/ICDT.2008.38
8. Salmon, D., 2007. Data Compression. 4th Edn., Springer-Verlag Ltd., London, pp: 125-128.