Oriental Journal of Computer Science & Technology Vol. **2**(1), 45-49 (2009)

Optimized Binary Merge Coding for Lossless Image Compression

N. SUBHASH CHANDRA[1], M. BALA RAJU [2], S. MAHABOOB BASHA[3] and A. GOVARDHAN[4]

[1,2,4]Department of Computer Science and Technology, College of Engineering, JNTU, Hyderabad, (India).
[3]Department of Computer Science and Engg., Al –Habeeb College of Engineering, Chevella, Ranga Reddy Dt (India).

(Received: March 18, 2009; Accepted: May 03, 2009)

ABSTRACT

Digital Spatial Image Processing is a rapidly evolving field with growing applications in Science and Engineering. Spatial Image data requires considerable storage capacity and transmission bandwidth. Despite rapid progress in mass-storage density, processor speeds, and digital communication system performance, demand for data storage capacity and data-transmission bandwidth continues to outstrip the capabilities of available technologies. This is a crippling disadvantage during transmission & storage. So, there arises a need of efficient Image Compression Techniques for compression .

This paper deals with Optimized Binary Merge Coding for data compression, which is a modification to the Binary merge coding. Like in BMC the Optimized Binary Merge Coding uses Huffman coding after the modified Binary Merge Coding. The results of the Optimized Binary Merge Coding are compared with Binary Merge Coding and JPEG. An experimental result shows that Optimized Binary Merge Coding improves compression rate compared to Binary Merge coding. The same algorithm can be extended to color images.

Keywords: Binary Merge Coding, Optimized Binary merge coding, Huffman Coding Technique, JPEG, Bit Plane, Data Table.

INTRODUCTION

The History of spatial image data compression started probably about a half of century ago with the works on predictive coding and variable length codes. The technological breakthrough that took place in 60's, 70's and 80's resulted in efficient compression algorithms[8,3] 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.

In this paper the effect of using the some modification to the Binary Plane Technique[2,5,4] is suggested to optimize the algorithm so the name Optimized Binary Plane Technique. This technique is spatial domain technique we found it better than the Binary Plane Technique.

The paper is organized as follows: the second and third sections are describes about the proposed Optimized Binary Merge Coding for encoding and Decoding. In the fourth section, we presented simulation results and finally the paper concludes with future work.

Optimized Binary Merge Coding and Encoding

The Optimized BMC encoding is involved with two stages i) Optimized binary Plane ii) Huffman coding in that order as given the Fig.1.

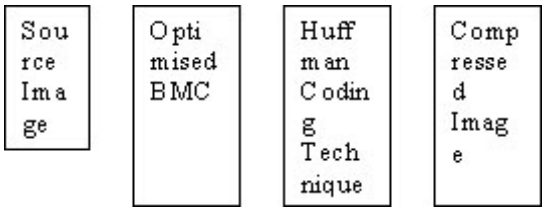


Fig. 1: Optimized BMC Image Compression Model

The Huffman coding[6,8,9] is popular and very widely used techniques so that is not explained here. But the Optimized Binary Merge Coding which is new explained in detail.

Optimized Binary Merge Coding

The Optimized BMC technique is used in the first stage. It is an improvement to BMC[1]. In both BMC and Optimized BMC two files namely bit plane and data table are created. The bit plane is collection of 1's and 0's to represent whether a pixel is repeated or not. The data table holds only the necessary pixel values. The bit plane and data table are later merged into one file.

On the data generated from OBMC, Huffman coding[6,7] is applied to further compress.

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[1] the first part 'bit plane' holds the bit 0 for each a pixel similar to previous pixel and the bit 1 for each pixel different from previous pixel. The second part 'data table' holds only the necessary pixel values, i.e. for a set of consecutive repeated values; one value is stored in the data table. After merging the bit plane and data table Huffman coding[6] is applied and final form of compressed file is generated.

The optimized binary merge coding like binary merge coding generates the 'bit plane' and 'data table'. But It is slightly different from binary merge coding. Instead of checking only for similar values, it is also checked for two successive values with difference in the range -8 to +7 with respect to previous value. If so the differences of the two successive pixels with respect to previous pixel, are merged and stored in one byte only. But this requires more than two codes. So we used 1 for two dissimilar values with difference outside the range -8 to +7, 00 for similar values and 01 with dissimilar values with the difference in the range -8 to 7. Like the above technique the Huffman code follows the optimized bit plane to further compress the file.

Optimized Binary Merge ALGORITHM

```
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 00
to bit plane to
                  indicate that pixel duplicate so not
retained */
    if (cur_pixel = prev_pixel) then
       append bit 00 to bit_planee
    else
     Begin
/*otherwise check whether the difference b/w two
successive pixels is in the range -8 to 7  */
       sec_pixel=read(image)
            if (diff(cur_pixel,prev_pixel)>-8 and
diff(cur_pixel,prev_pixel)<7) and
               (diff(sec_pixel,prev_pixel)>-8 and
diff(sec_pixel,prev_pixel)<7)
        begin
         append bit 01 to bit_plane
             write merge(diff(cur_pixel,prev_pixel),
merge(diff(cur_pixel,prev_pixel))
          to datatable file
       end
       else
         append bit 01 to bit_plane
         write cur_pixel to datatable file
         unread(image,sec_pixel)
```

```
        prev_pixel=cur_pixel
      end
    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 file
END
```

Optimized Binary Merge Decoding

In the reconstruction of the image the Inverse Huffman Technique and Inverse Optimized BPT are applied on compressed file respectively as in the Fig. 2.

Inverse Optimized Binary Merge Coding

In the Inverse Optimized Binary Merge Coding first the Bit Plane and Data Tables are extracted. Using the Data table and Bit Plane the source image is built as inverse of Optimized Binary Merge Coding algorithm.

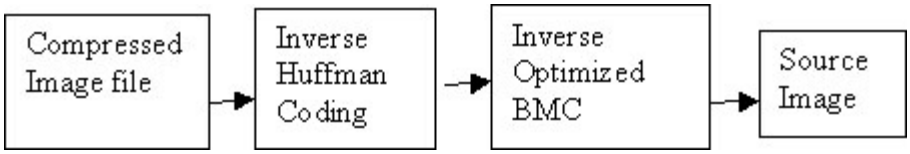


Fig. 2: Reconstruction Model in Optimized BMC

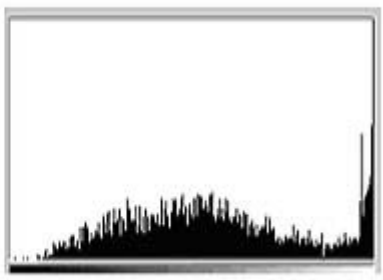
RESULTS

The brain image is taken as one of sample source images and applied with Optimized Binary Merge Coding to compress. The reconstructed image, its histogram and statistical information are as shown in the Fig. 3.

From the Table 1 which is generated from the results of the execution of the BMC and OBMC programs, It is clear that OBMC technique gives much better compression rate than BMC.

The memory requirement for both BMC & OBMC techniques is very less because the processing is done byte by byte. In case of the JPEG the entire image needs to be brought into memory.

As per as process complexity is concerned BMC and OBMC are simple to implement compared to JPEG. The graph in Fig. 4 is drawn based on the Table 1.



Mean	**:**	**192.86**
Std. Dev.	**:**	**72.22**
Median	**:**	**247**
Pixels	**:**	**12610**

Reconstructed Image **Histogram** **Statistical Information**

Fig. 3: Sample Image Brain with histograms

Table 1: Size and Compression rate between BMC vs OBMC vs JPEG

Image Name	RAW Size	JPEG		BMC		OBMC	
		Size	Comp Rate	Size	Comp Rate	SIZE	Comp Rate
Brain	12610	15109	0.8346019	7609	1.6572479	7144	1.76511758
Chest Xray	18225	16180	1.1263906	17207	1.0591619	12504	1.45753359
Knee joint	18225	17193	1.0600244	13245	1.3759909	11967	1.52293808
Head Scan	15625	15184	1.0290437	12532	1.2468081	10553	1.48062162
Shoulder	18225	16962	1.0744606	12562	1.450884	10805	1.68671911

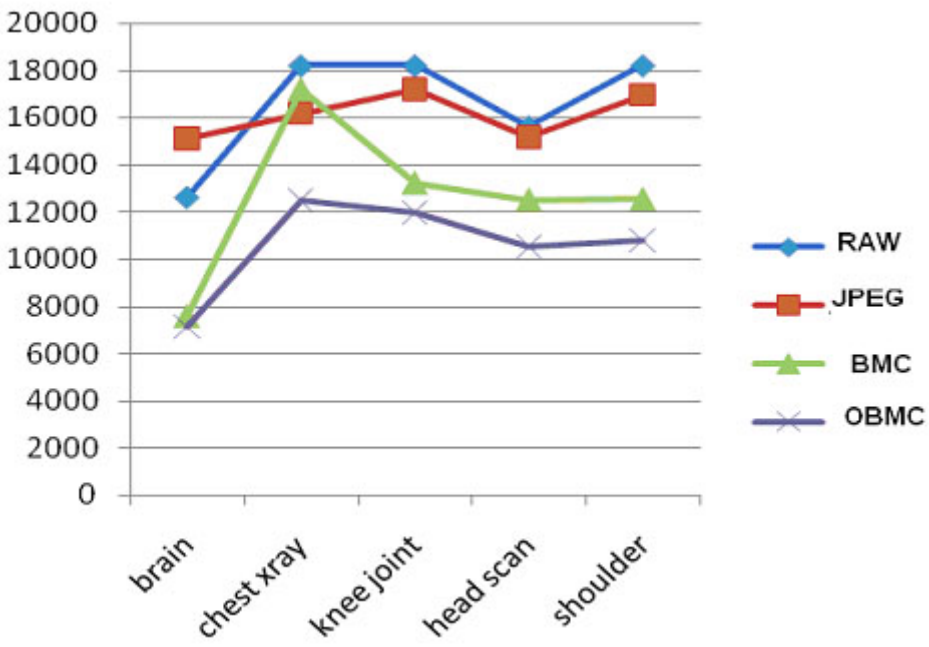


Fig. 4: Graph for comparison between file sizes of different compression techniques

CONCLUSIONS

The compression rate of BMC and OBMC is better than JPEG not necessarily in all cases. We have taken only the medical images where BMC & OBMC are better.

But in most of the cases the Optimized Binary Merge Coding is much better than Binary Merging Coding.

The BMC and OBMC techniques can be easily extended to color images by changing the algorithm accordingly.

REFERENCES

1. Subhash Chandra.N., An Hybrid approach Lossless compression for Color Images, Asian journal of information Technology, Medwell publishers, Vol.8, pp 472-472 (2009).
2. Skodras.A, Christopoulis.C, and Ebrahimi.E, "The JPEG2000 still image compression standard," IEEE Signal Processing Mag., vol. 18, no. 5, pp. 36-58, September (2001).
3. Wallace.G.K, " The JPEG still picture compression standard." IEEE Trans. Consumer Electron, Vol 38, No 1, Feb (1992).
4. Jayant.N, " Signal Compression: technology targets and research directions," IEEE J. Select Areas Commun., Vol. 10, No.5, pp 796 - 818 , June (1992).
5. Rafael C.Gonzalez, Richard E. Woods "Digital Image Processing " Second edition Pearson Education, Printice Hall.
6. Shannon,C.E." A Mathematical theory of communication". *Bell.Sys.Tech. J.* **27**, 379-423.
7. Huffman,D.A."A Method for construction of min. redundancy Codes" *Proc. IRE,* **40**(10): 1098-1101, (1951).
8. Abramson,N. "Information theory and coding",McGraw-Hill,New York.
9. Schwarz,J.W.,Barker,R.C."Bit Plane encoding: A Technique for source Encoding"IEEE Trans.aerospace Elec. Systems vol. AES-2 no.4, pp 385-392, (1966).