Efficient Block Truncation Coding

K.Somasundaram
Department of Computer Science and Applications
Gandhigram Rural Institute
Gandhigram – 624 302
Tamilnadu, India

MS.S.Vimala
Department of Computer Science
Mother Teresa Women's University
Kodaikanal – 624 102
Tamilnadu, India

Abstract – **Block Truncation Coding (BTC) is one of the lossy image compression techniques. The computational complexity involved in this method is very simple. In the proposed method, the feature of inter-pixel correlation is exploited to further reduce the requirement of bits to store a block. The proposed method gives very good performance in terms of bit-rate and PSNR values when compared to the conventional BTC.**

Keywords – image compression; mean, standard deviation; bit-plane; bit-rate;

I. INTRODUCTION

The amount of image data grows day by day. Large storage and bandwidth are needed to store and transmit the images, which is quite costly. Hence methods to compress the image data are essential now-a-days. The image compression techniques are categorized into two main classifications namely Lossy compression techniques and Lossless compression techniques [1]. Lossless compression yields only less compression ratio but gives good quality of compressed images, whereas the lossy compression techniques lead to loss of data with higher compression ratio [2]. Vector Quantization and Block Truncation Coding are few of the lossy image compression techniques. Block Truncation Coding (BTC) is a simple technique which involves less computational complexity [3]. BTC is a recent technique used for compression of monochrome image data. It is one-bit adaptive moment-preserving quantizer that preserves certain statistical moments of small blocks of the input image in the quantized output [2]. The original algorithm of BTC preserves the standard mean and the standard deviation [4]. The statistical overheads Mean and the Standard deviation are to be coded as part of the block. The truncated block of the BTC is the one-bit output of the quantizer for every pixel in the block. [4] and [5] report other important efforts to improve the coding efficiency of BTC.

Various BTC methods have been proposed during last twenty years [6]. Several single bit-map color BTC algorithms [7]-[9] have been proposed .

A. *Block Truncation Coding*

A gray level image of size 256 x 256 requires 256 x 256 x 8 bits. The input image is divided into small blocks of size 4 x 4 pixels. The mean of the block $\bar{x}$ is calculated using the equation (1).

$$\bar{x} = \frac{1}{16}\sum_{i=1}^{16} Xi \qquad (1)$$

The standard deviation is calculated using the equation (2).

$$\sigma = \sqrt{\frac{\left(\bar{x} - x_i\right)^2}{n}} \qquad (2)$$

The block of gray levels is converted in to a bit-plane that comprises of only 0's and 1's using the equation (3). If the current pixel value is greater than the mean value $\bar{x}$, it is coded as 1 otherwise 0.

$$\left.\begin{array}{l} \text{if } x_i = 1, \text{ if } x_i > \bar{x} \\ \text{else } x_i = 0 \end{array}\right\} \qquad (3)$$

The two quantizing levels g_1 and g_2 for each block are calculated using the equations (4) and (5).

$$g1 = \bar{x} - \sigma\left[\sqrt{\frac{p}{m-p}}\right] \qquad (4)$$

$$g2 = \bar{x} + \sigma\left[\sqrt{\frac{m-p}{p}}\right] \qquad (5)$$

.

$\bar{x}$ is the mean of the block
σ is the deviation of the block
p, the number of pixels in a block whose values are greater than or equal to x and
m, the total number of pixels in the block

The two quantizing levels g_1, g_2 and bit-plane for each block are transmitted or stored for the compressed image. For an original image, a block of pixels requires 4 x 4 x 8 = 192 bits. But the same block after compression using BTC requires only 16 + 8 + 8 = 32 bits. The bit-rate is reduced to 16.67%. The bit-rate reduction achieved using BTC can further be reduced by the proposed method. While encoding the compressed image in the receiving end, all the 1's are replaced with g_2 and 0's are replaced with g_1.

The performance of the image compression technique is measured by computing the Mean Square Error (MSE). MSE gives the difference between the original image and the reconstructed image and is calculated using the equation (6).

$$MSE = \frac{1}{M}\sum_{i=1}^{M}\left(x_i - \bar{x}\right)^2 \qquad (6)$$

where M is the number of elements in the image. The PSNR (Peak Signal to Noise Ratio) is the quality of the reconstructed image and is the inverse of MSE. The PSNR is defined as follows:

$$PSNR = 10\log_{10}\left[\frac{\left(2^n - 1\right)^2}{MSE}\right] \qquad (7)$$

Where n is the number of bits per pixel. As an example, if we want to find the PSNR between two 256 gray level images, then we set n to 8 bits.

A novel idea is introduced in this paper to improve the coding efficiency of BTC. The remaining part of the paper is organized as follows: The proposed method is explained in section 2. The results are discussed in section 3 and the conclusion is given in section 4.

II PROPOSED METHOD (EBTC)

In this method Efficient Block Truncation Coding (EBTC), we exploit the feature of inter-pixel redundancy to further reduce the bit-rate. It is a known fact that the nearest pixels have more or less same intensity values. After splitting the image into small blocks of size 4 x 4 pixels, the blocks are categorized into low detail blocks and high detail blocks. When the intensity values of the nearest pixels are different, the block is referred to as high detail block and if the difference between the intensity values is less, the blocks are referred to as low detail block. Categorization of blocks is done using the equations (8) and (9)

$$S = \sum_{i=1}^{k} abs(x_i - \bar{x}) \qquad (8).$$

where S is the sum of the absolute difference between the individual components of the block and the mean of the block.

$$\text{if } S <= T. \text{ then } B_h \qquad \text{else } B_l \qquad (9)$$

where B_h is the high detail block, B_l is the low detail block and $\bar{x}$ is the mean of the block. The threshold values range from 50, 100, 200, to 1000. For low detail blocks, only the mean value $\bar{x}$ is stored instead of the two quantizing levels g1 and g2. The bit plane is also discarded, since 0 is assumed for every pixel. Hence for a low detail block, it requires only 8 bits and for a high detail block, 32 bits are required. The bit rate can significantly be reduced using the proposed method. Depending on the number of low detail blocks, the amount of storage required to store the compressed image varies.

A. *Algorithm for EBTC*

Step1: Input the image of size n x n pixels.
Step2: Divide the image into N blocks, each of size 4 x 4 pixels.
Step3: Compute the mean for each block using the equation (1).
Step4: Compute the two quantizing levels g1 and g2 using equations (4) and (5).
Step5: Compute S using the equation (8)
Step6: Determine the threshold value T.
Step7: if S > T then
Step8: Generate the bit plane for the high detail block B_h using the equation (3) and store the bit plane.
Step9: Store the two quantizing levels g_1 and g_2 and go to Step11.
Step10: else Store only the Mean $\bar{x}$ for the low detail block B_l.
Step11: Repeat the steps 3 through 10 for all blocks.

III RESULTS AND DISCUSSION

Experiments were carried out with the standard images Lena, Bridge, Boats and Cameraman of size 256 x 256 pixels. The PSNR value is taken as a measure of reconstructed image quality. The algorithms are implemented using Matlab 7.0 on Windows Operating System. The hardware used is the Intel Core 2 E7400@ Duo 2.8 GHz Processor with 2 GB RAM.

The PSNR values and the bit rate obtained with the conventional BTC and the proposed BTC methods are presented in Tables1 thru. Table5.

Table1: The PSNR values and the Bit-rate obtained with the normal BTC for images of size 256 x 256 pixels.

Image	PSNR	Bitrate
Cameraman	18.51	2.00
Lena	22.15	2.00
Boats	19.07	2.00
Bridge	18.44	2.00

In Table1, it is observed that the bit rate obtained is 2 for all images. But in the proposed BTC method, the bpp varies with the nature of the image.

In the proposed method, to categorize the image blocks in to high detail and low detail blocks, the sum value S that is computed using the equation (8) is compared against the threshold value T as in equation (9). The initial threshold value was set to 100. The threshold value was incremented by 100 every time until the bit rates of consecutive iterations converge.

Table 2: The PSNR values and the bit-rates obtained with different threshold values for the cameraman image of size 256 x 256 pixels with respect to the proposed method.

Image	Threshold	PSNR	Bitrate
Cameraman	100	18.54	1.05
	200	18.74	0.84
	300	19.00	0.72
	400	19.18	0.68
	500	19.45	0.65
	600	19.80	0.62
	700	20.26	0.59
	800	20.93	0.57
	900	21.70	0.54
	1000	22.50	0.53
	1100	23.24	0.51
	1200	23.97	0.50
	1300	24.34	0.50

In Table2, the bit rate obtained with the threshold values of 1200 and 1300 converge. In every iteration, as the threshold value increases, both the PSNR and the bit rate values are improved. In Table1, the PSNR obtained with the conventional BTC is 18.51 and the bit rate obtained is 2 for the cameraman. In table2, with the threshold value of 100, the PSNR and the bit rate obtained are 18.54 and 1.05 respectively.

The PSNR is increased to 18.54 and the bit rate is reduced to 1.05 (nearly 1 bit) from the conventional BTC method. For the same image, with the threshold value of 1300, the PSNR is increased to 24.34 and the bit rate is reduced to 0.50 which shows significant improvement in both PSNR and the bit rate values. The same trend is noticed for the other images also. But the rate of reduction in bit rate varies with the nature of the input image.

Table-3: The PSNR values and the bit-rates obtained with different threshold values for the Lena image of size 256 x 256 pixels with respect to the proposed method.

Image	Threshold	PSNR	Bitrate
Lena	100	22.30	1.06
	200	22.69	0.80
	300	23.25	0.68
	400	24.09	0.59
	500	25.15	0.55
	600	25.91	0.52
	700	26.50	0.51
	800	26.78	0.51
	900	27.40	0.50
	1000	27.73	0.50
	1100	28.04	0.50

Table-4: The PSNR values and the bit-rates obtained with different threshold values for the Boats image of size 256 x 256 pixels with respect to the proposed method.

Image	Threshold	PSNR	Bitrate
Boats	100	19.13	1.25
	200	19.41	0.96
	300	19.79	0.82
	400	20.38	0.71
	500	21.14	0.63
	600	22.07	0.57
	700	22.90	0.53
	800	23.40	0.52
	900	24.81	0.51
	1000	25.73	0.50
	1100	25.78	0.50

Table-5: The PSNR values and the bit-rates obtained with different threshold values for the Bridge image of size 256 x 256 pixels with respect to the proposed method.

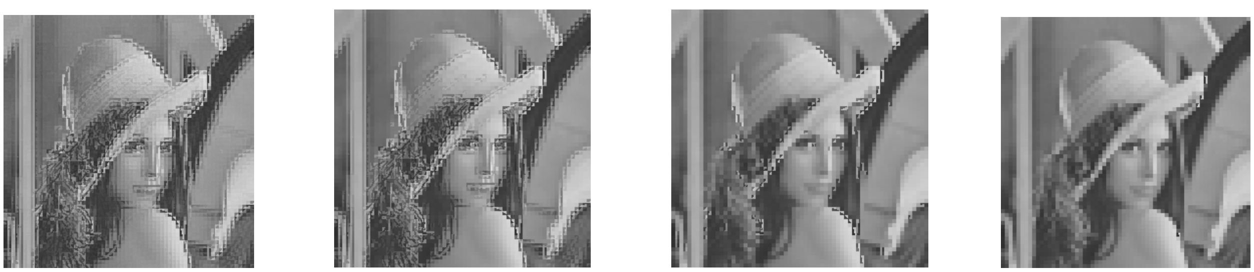
Image	Threshold	PSNR	Bitrate
Bridge	100	18.45	1.78
	200	18.84	1.26
	300	19.62	0.88
	400	2.047	0.69
	500	21.15	0.61
	600	21.89	0.56
	700	22.59	0.54
	800	23.26	0.52
	900	23.80	0.51
	1000	24.25	0.50
	1100	24.40	0.50

From the above tables, it is observed that when the threshold value reaches 1000, the bit rates started converging. When compared to the results obtained with the BTC method, the proposed method better PSNR values as well better compression ratio.

Fig.1 shows the images taken for the study.

(a) Lena (b) Cameraman (c) Boats (d) Bridge

Fig.1: Images taken for the study



(a) PSNR : 22.15
bpp : 2.00

(b) PSNR : 22.30
bpp : 1.06
Thresh : 100

(c) PSNR : 25.13
bpp : 0.55
Thresh : 500

(d) PSNR : 27.73
bpp : 0.50
Thresh : 1000

Fig.2: Reconstructed images obtained with different threshold values – (a) reconstructed image of BTC method, (b), (c) and (d) are the reconstructed images of EBTC method.

Fig.2 represents the visual comparison of the reconstructed images obtained with the BTC and EBTC methods.

IV CONCLUSION

In this paper, the inter-pixel redundancy feature is incorporated in the conventional BTC method and this has lead to further reduction in the bit rate and improvement in the PSNR of the reconstructed images. The EBTC method was tried with different threshold values in categorizing the low and high detail blocks. As the threshold value increases, the PSNR is improved and the bit-rate is reduced. The technique was tried with different images and yielded better results with less computational effort.

REFERENCES

[1] Paul W.Jones and Majid Rabbani, “Digital Image Compression”, Eastman Kodak Company, Rochester, New York.

[2] M.D.Lema, O.R.Mitchell, “Absolute Moment Block Truncation Coding and its Application to Color Image”, IEEE Trans. Coomun., Vol. COM-32, No. 10, pp. 1148-1157, Oct. 1984.

[3] E.J.Delph and O.R.Mitchell, “Image Compression using BTC”, IEEE Trans. Commun., Vol. COM-27, pp. 1335-1342, Sept. 1979.

[4] Vishwas R., Udpikar and Jewan P.Raina, “BTC Image Coding Using Vector Quantization”, IEEE Trans. On Commun., Vol., COM-35, No. 3, March 1987.

[5] D.J.Healy and O.R.Mitchell, “Digital Video Bandwidth Compression using Block Truncation Coding”, IEEE Trans. Coomun., Vol. COM-29, pp. 1809-1817, Dec. 1981.

[6] P.Franti, O.Nevaleinan, and T.Kaukoranta, “Compression of Digital Images by Block Truncation Coding: A Survey”, Comput. J., Vol. 37, No. 4, pp. 308-332, 1994.

[7] G.Campbell, T.A.Defanti, J.Frederiksen, S.A.Joyce, A.L.Lawrence, J.A.Lindberg, and D.J.Sandin, “Two bit/pixel Full Color Encoding”, Comput. Graph., Vol. 20, pp. 215-223, Aug. 1986.

[8] Y.Wu, and D.C.Coll, “Single Bit-map Block Truncation Coding of Color Images”, IEEE J. Select. Areas Commun., Vol. 10, pp. 952-959, June 1992.

[9] T.Kurita and N.Otsu, “A Method of Block Truncation Coding for Color Image Compression”, IEEE Trans. Commun., Vol. COM-41, pp. 1270-1274, Sept. 1992.