

Improved Adaptive Block Truncation Coding for Image Compression

S.Vimala

Department of CSE
Mother Teresa Women's
University
Kodaikanal, Tamilnadu, India

P.Uma

Department of CS
Mother Teresa Women's
University
Kodaikanal, Tamilnadu, India

B.Abidha

Department of CS
Mother Teresa Women's
University
Kodaikanal, Tamilnadu, India

ABSTRACT

The Block Truncation Coding (BTC) is one of the lossy image compression algorithms. In this paper, we have proposed a method called the Improved Adaptive Block Truncation Coding (IABTC) based on Adaptive Block Truncation Coding (ABTC). The feature of inter-pixel redundancy is exploited to reduce the bit-rate further by retaining the quality of the reconstructed images. The proposed method outperforms the existing BTC based methods both in terms of bit-rate and PSNR values.

General Terms

Image Compression

Keywords

Compression, block, bit plane, bpp, PSNR, quantizers

1. INTRODUCTION

Data compression algorithms lead to reduction in transmission time and storage cost. Image compression techniques are classified into lossy and lossless techniques. In lossless techniques, the reconstructed image will be the exact replica of the original image whereas in lossy techniques, the reconstructed image will be an approximation of the original image.

One of the main goals for image data compression is to reduce redundancy in the image block as much as possible. That is, it is very important to represent an image with as few bits as possible while maintaining good image quality. Both compression and decompression algorithms should be simple and efficient. BTC is one of the simple and easy to implement image compression algorithms [1]. It comes under lossy image compression techniques. Though some data is lost, it does not make much difference to the Human Visual System (HVS).

One such lossy image compression techniques is Block Truncation Coding (BTC) technique. BTC has the advantage of being easy to implement when compared to Vector Quantization [8] and Transform Coding [9, 10]. BTC achieves 2 bits per pixel (bpp) with low computational complexity [11]. Many techniques with modifications to BTC have been proposed to reduce the bit-rate obtained with normal BTC: Gallagher [12] proposed a

Median Filter Roots method; Yung-Gi Wu [13] proposed Probability based Block Truncation Image Bitplane Coding; Yu-Chen Hu [14] presented a Modified BTC with Predictive Technique and Bitplane Coding with Edge Pattern. In this paper, we have presented a method named Improved Adaptive Block Truncation Coding (IABTC) based on the Adaptive Block Truncation Coding (ABTC) method. Few other existing Block Truncation Coding (BTC) based methods are discussed in Section 2. The proposed method is explained in Section 3. Results are discussed in Section 4 and the conclusion is given in Section 5.

2. EXISTING BTC BASED ALGORITHMS

Some of the existing BTC based algorithms are discussed in this section.

2.1. Standard BTC Algorithm

Block Truncation Coding (BTC) works by dividing the image into small sub blocks of size 4 x 4 pixels and then reducing the number of gray levels within each block. This reduction is performed by a quantizer that adapts to the local image statistics. The basic form of BTC divides the whole image into N blocks and codes each block using a two-level quantizer. The two levels, a and b are selected using the mean (\bar{X}) and standard deviation (σ) of the gray levels within the block and are preserved. Each pixel value within the block is then compared with the mean and then is assigned to one of the two levels [1].

The \bar{X} and σ are calculated using the equations (1) and (2).

$$\bar{X} = \frac{1}{m} \sum_{i=1}^m x_i \quad (1)$$

$$\sigma = \sqrt{\frac{\sum (y_i - x_i)^2}{m}} \quad (2)$$

where, m is the number of pixels in each block, and X_i is the original pixel value of the block. If the pixel value of each block is greater than or equal to mean, it is represented by '1' and if less than the mean, it is represented by '0'. The collection of 1s or 0s for each block is called a bit plane, B . In BTC, two statistical moments a and b are computed using the equations (3) and (4) and are preserved along with the bit plane for reconstructing the image. The compressed image is transmitted or stored as a set $\{B, a, b\}$

$$a = \bar{X} - \sigma \sqrt{\frac{q}{m-q}} \quad (3)$$

$$b = \bar{X} + \sigma \sqrt{\frac{m-q}{q}} \quad (4)$$

where, q is the number of pixel values greater than or equal to \bar{X} , and $(m-q)$ is the number of pixels whose gray levels are less than \bar{X} . While reconstructing the image, the θ in the bit plane is replaced by \mathbf{a} and the 1 in the bit plane is replaced by \mathbf{b} . The difference between the original image and reconstructed image is called Mean Square Error (MSE) and is calculated using the equation (5). The quality of the reconstructed image called the Peak Signal to Noise Ratio (PSNR) is calculated using the equation (6) and is the inverse of MSE.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - x_i)^2 \quad (5)$$

$$PSNR = 10 \cdot \log_{10} [MSE/255^2] \quad (6)$$

where, y_i is the reconstructed pixel value, x_i is the original pixel value and N is the number of pixels in an image.

Input image block	Bit Plane	Reconstructed
2 9 12 15	0 1 1 1	2 12 12 12
2 11 11 9	0 1 1 1	2 12 12 12
2 3 12 15	0 0 1 1	2 2 12 12
3 3 4 14	0 0 0 1	2 2 2 12

$\bar{X} = 9$	$q = 9$	$a = 2$
$\sigma = 4.91$		$b = 12.3$

Fig 1: The encoded and the decoded image block

2.2. Absolute Moment Block Truncation Coding (AMBTC)

AMBTC is the variant of BTC and introduced by Lema and Mitchell [2]. In this method, two statistical moments \mathbf{a} (lower mean) and \mathbf{b} (higher mean) calculated using the equations, (7) and (8) are preserved along with the bit plane.

$$\bar{X}_l = \frac{1}{m-q} \sum_{x_i < \bar{X}} x_i \quad (7)$$

$$\bar{X}_h = \frac{1}{q} \sum_{x_i \geq \bar{X}} x_i \quad (8)$$

The coding and decoding processes are very fast for AMBTC because the square root calculation and multiplications are omitted [5].

2.3. Minimum Mean Square Error (MMSE)

MMSE is the iterative process of AMBTC. This technique is used to reduce MSE value. In this method, the threshold value t is initialized by the average of minimum and maximum values of each block. The threshold value thus calculated is optimized through iterations as described in the Figure 3. The optimization process is terminated when the threshold values of consecutive iterations converge. The two statistical moments \mathbf{a} and \mathbf{b} are calculated using the equations (7) and (8).

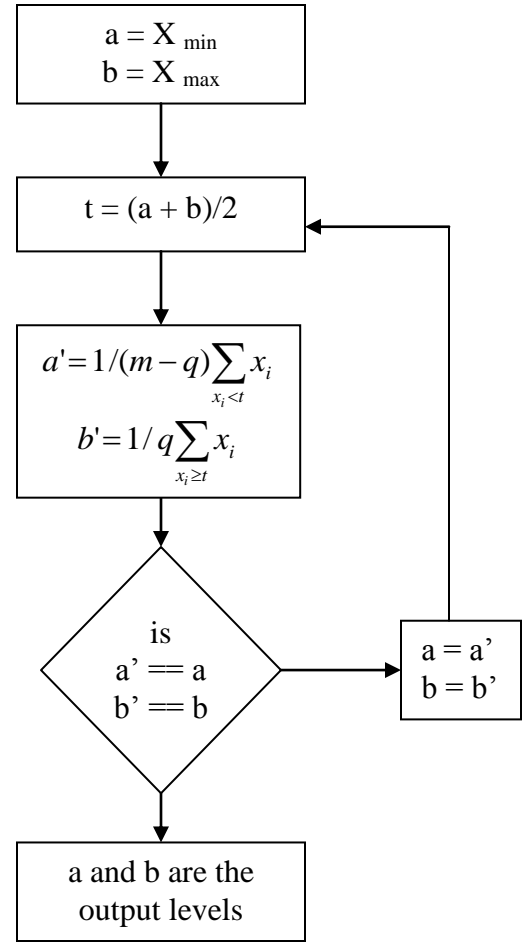


Fig 2: MMSE Method

2.4. Adaptive Block Truncation Coding (ABTC)

Adaptive Block Truncation Coding (ABTC) is based on multi level quantizer. In this method, the quality of the reconstructed image is improved with the increase in bit-rate. The input blocks are classified into three groups depending on the inter-pixel correlation within each block. (1) Low activity blocks - where all the pixel values inside the block are approximately the same and visually represent a flat area of gray, (2) Medium activity blocks - where there are some small transitions between pixels but these blocks do not represent any contrasting edges, and (3) High activity blocks - contain big pixel value changes with contrasting edges. A simple mathematical way of classifying the

blocks would be using the standard deviation in equation (2) where, class 1 contains all low σ blocks, class 2 contains medium σ blocks, and class 3 contains the high σ blocks [3].

In order to efficiently code the pixel blocks and to maintain better image quality, three set of quantizers are used in the ABTC method for all blocks: a 1-level quantizer for low activity blocks, a 2-level quantizer for the medium activity blocks, and a 4-level quantizer for the high activity blocks.

The output equation of the 1-level quantizer can be simply described using the equation (1) which is the sample mean \bar{X} of the block. If an 8-bit value is used for representing \bar{X} , the bit-rate of these blocks will be $8/16 = 0.5$ bits per pixel (bpp) [3].

The medium activity blocks are the kind of pixel blocks for which the standard BTC method is best designed. The bit-rate for this class is 2 bpp [3].

Based on the 2-level MMSE quantizer, a 4-level MMSE quantizer can be designed. By increasing the number of output quantization levels, the raggedness produced by standard BTC in the high activity edge blocks can be smoothed. Let the output levels of the quantizers be a, b, c and d and are calculated as $Xmin, (2Xmin+Xmax)/3, (Xmin+2Xmax)/3,$ and $Xmax$. Use the same iterative processes to optimize the thresholds t_1, t_2 and t_3 for the output levels a, b, c and d and vice versa. The results of the quantization will be a bit plane whose individual elements are of size 2 bits and the output levels are a, b, c and d . Hence, the bit-rate for this quantizer is 4 bpp [3].

3. IMPROVED ADAPTIVE BLOCK TRUNCATION CODING (IABTC)

The proposed method (IABTC) is based on Adaptive Block Truncation Coding and is used in further reducing the bit rate and to improve image quality. All the input image blocks are categorized into three groups based on sum value (S), which is calculated using the equation (9).

$$S = \sum_{i=1}^m abs(x_i - \bar{X}) \quad (9)$$

The blocks are categorized in to three groups as follows:

1. Low detailed block, if $S \leq t_1,$
2. Medium detailed block, is $S > t_1$ and $S \leq t_2$
3. High detailed block, if $S > t_2$ and $S \leq t_3$

The threshold values t_1, t_2 and t_3 are 50,170 and 256 respectively.

$$th_r = \min + ((\max - \min)r/n) \quad (11)$$

where, th_r represents the r^{th} value of threshold and n is the number of quantization levels and the four quantizing levels a, b, c and d are computed using the equation (12). min and max are the minimum and maximum intensities of the block respectively.

$$\left. \begin{aligned} a &= \min. \\ b &= (2\min + \max)/3. \\ c &= (\min + 2\max)/3. \\ d &= \max. \end{aligned} \right\} \quad (12)$$

While decompression, the pixels in the low detailed blocks are reconstructed using the mean of the block. The pixels in the medium detailed blocks are reconstructed using a two level quantizer with the two statistical moments being Min and Max which are optimized as in MMSE and the pixels in the high detailed blocks are reconstructed using the four quantizing levels.

The compression of image is done in two levels in IABTC: i) By making slight modifications to ABTC and ii) Dividing the statistical moments by 4.

3.1. First level compression

- Step1:** Input the image size $n \times n$ pixels
- Step2:** Divide the image into N blocks, each of size 4×4 pixels.
- Step3:** Compute the mean for each block using the equation (1).
- Step4:** Compute S for each block using the equation (9)
- Step5:** Categorize the block (Low or Medium or Detailed block) using the equations (10).
- Step6:** Store the set {B, statistical moments}

* The number of statistical moments differs depending on the type of input block.

3.2. Second level compression

In the second stage, all the statistical moments are divided by 4. Generally a statistical moment requires a maximum of 8 bits ($\log_2 256$) to get stored. But, when divided by 4, it requires only a maximum of 6 bits ($\log_2 64$).

While reconstructing the image, the statistical moments are multiplied back by 4 to get the approximate original values ranging from 1 to 256.

3.3. Bit-rate calculation

While encoding the image, for a low detailed block, only the mean (8 bits) alone is stored. For a medium detailed block, the Bit plane (size 16 bits) and the two quantizing values Max (8 bits) and Min (8 bits) are stored. Hence it requires 32 bits to store a medium detailed block. In the bit plane B of medium detailed block, if a pixel's intensity value is greater than or equal to mean, it is coded as 1 otherwise 0. For a high detailed block, the bit plane B requires 32 bits, i.e. 2 bits per pixel. The bit plane is generated by coding the individual pixel as

- 00, if $X_i \leq t_1$
- 01, if $X_i > t_1$ and $\leq t_2$
- 10, if $X_i > t_2$ and $\leq t_3$ and
- 11, if $X_i > t_3$

After the second level of compression, only 6 bits are required to store the quantizing levels. Hence the bpp of the image compressed using IABTC method is calculated as

$$n = n_1 * 6 + n_2 * (16 + 6 + 6) + n_3 * (32 + 6 + 6) \quad (13)$$

$$bpp = n / (256*256) \quad (14)$$

where, n_1 is the number of low detailed blocks, n_2 is the number of medium detailed blocks and n_3 is the high detailed blocks.

4. RESULTS AND DISCUSSION

Experiments were carried out with standard images Cameraman, Boats, Bridge, Baboon, Lena and Kush of size 256 x 256 pixels. The input images taken for the study are given in Fig 3.



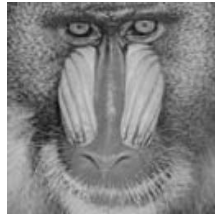
(a) Cameraman



(b) Boats



(c) Bridge



(d) Baboon



(e) Lena



(f) Kush

Fig 3: The Input Images taken for the study

The PSNR value is taken as a measure of the quality of the reconstructed image. The algorithm is implemented using MATLAB 7.0 on Windows Operating System. The hardware used is the Intel® 3.06GHz Processor with 512 MB RAM. The PSNR values and the bpp obtained with the proposed method are compared with that of the existing methods based on BTC in Table 1.

From Table 1, it is observed that a minimum bpp of 0.97 is achieved with the Baboon image and a maximum of 1.74 is achieved with the Bridge image. Since the Bridge image is high detailed block, it requires 1.74 bpp. Though the bpp obtained with ABTC varies with images, the average bpp is 2. But the average bpp obtained with IABTC is 1.16. As far as the PSNR is concerned, there is a significant improvement in IABTC. Both in terms of bpp and PSNR, IABTC outperforms the other techniques. For visual comparison, the reconstructed images by the discussed methods are given in Fig 4.

Table 1: The PSNR values and bit-rate obtained with different BTC methods and IABTC

Images	BTC		AMBTC		MMSE'		ABTC		IABTC	
	BPP	PSNR	BPP	PSNR	BPP	PSNR	BPP	PSNR	BPP	PSNR
Cameraman	2	32.07	2	32.13	2	32.70	1.90	30.19	1.01	34.67
Baboon	2	37.14	2	37.46	2	37.68	2.13	37.57	0.97	36.46
Boats	2	32.77	2	33.15	2	33.81	1.82	30.36	1.24	34.74
Bridge	2	30.68	2	30.94	2	31.47	2.09	31.00	1.74	32.02
Lena	2	34.62	2	34.85	2	35.39	2.18	35.60	0.98	35.79
Kush	2	35.41	2	35.55	2	36.11	1.88	34.50	1.00	35.77
Average	2	33.78	2	34.01	2	34.53	2.00	33.20	1.16	34.91

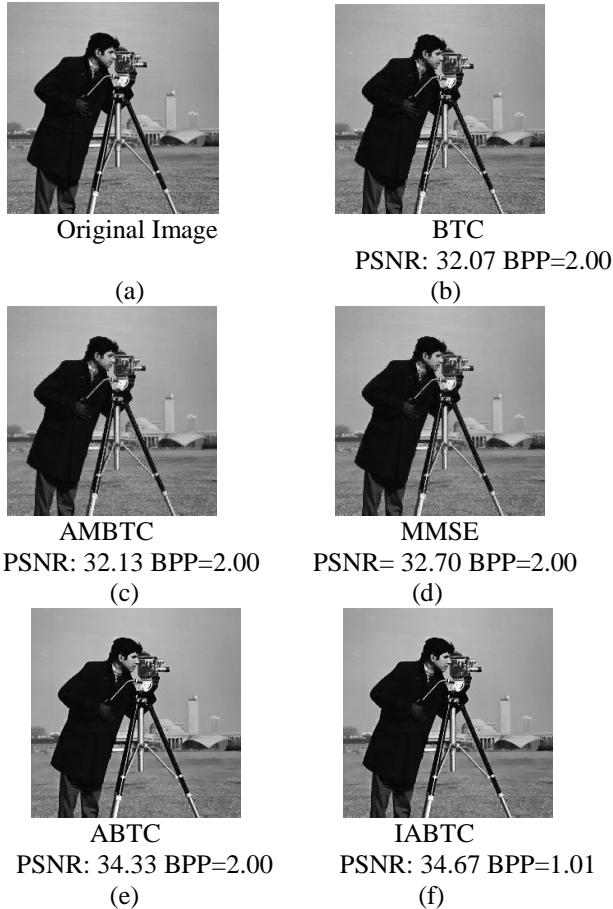


Figure 5: (a) Original Image, (b) – (f) Reconstructed images of BTC, AMBTC, MMSE, ABTC and the proposed (IABTC) methods.

5. CONCLUSION

In this paper, the feature of inter-pixel redundancy is exploited to reduce the bpp to significant level. The IABTC method was tried with different images. Depending on the type of image that is compressed, the number of low, medium and high detailed blocks, vary and thus results in different bpp for each image. But for all images, the bpp is reduced to a value that is less than 2. The average bpp obtained with the proposed method is 1.16. The proposed technique can also be applied to color images. This method is suitable for implementation in small hand held devices where image processing service has become inevitable.

6. REFERENCES

[1] Meftah M.Almrabet, Amer R. Zerek, Allaoua Chaoui and Ali A. Akash,” Image Compression using Block Truncation

Coding”, IJ-STA, Vol 3, No.2, PP.1046-1053, December 2009.

- [2] Maximo D.Lema, O.Robert Mitchell, ”Absolute Moment Block Truncation Coding and its Application to Color Images”, IEEE Transactions on Communications, VOL. COM-32, NO. 10, October 1984
- [3] Lucas Hui, “An Adaptive Block Truncation Coding Algorithm for Image Compression”, IEEE, PP.2233-2235, 1990.
- [4] Pasi Franti, Olli Nevalainen, Timo Kaukoranta, Compression of Digital Images by Block Truncation Coding:A Survey”, The Computer Journal, Vol.37, No.4, 1994.
- [5] Madhu Shandilya and Rajesh Shandilya, ”Implementation of Absolute moment Block Truncation Coding Scheme Based on Mean Square Error Criterion”, SDR 03 Technical Conference and Product Exposition.
- [6] Y.V.Ramana Rao and C.Eswaran, ”A New Algorithm for BTC Image Bit Plane”, IEEE Transactions on Communications, Vol.43, NO.6, June 1995.
- [7] K.Somasundram and S.Vimala, “Efficient Block Truncation Coding”, International Journal on Computer Science and Engineering, Vol. 02, No. 06, 2010, 2163-2166
- [8] N.M.Nasrabadi and R.B.King, “Image Coding using Vector Quantization: A Review”, IEEE Transactions on Communication, COM-36, pp. 957-971, 1998.
- [9] W.B.Pennebaker and J.L.Mitchell, “JPEG Still Image Data Compression Standard”, New York, Van Nostrand Reinhold, 1993.
- [10] M.Rabbani and R.Joshi, “An overview of the JPEG 2000 Still Image Compression Standard”, Signal Processing and Image Communication, Vol. 17, pp. 3-48, 2002.
- [11]K.Somasundaram and I.Kaspar Raj, “Low Computational Image Compression Scheme based on Absolute Moment Block Truncation Coding”, World Academy of Science, Engineering and Technoloy, Vol. 19, pp. 166-171, 2006.
- [12] G.Arce and N.C.Jr.Gallagher, “BTC Image Coding using Median Filter Roots”, IEEE Transactions on Communication, Vol. 31, No. 6, pp. 784-793, 1983.
- [13] Yung-Gi Wu, “Block Truncation Image Bitplane Coding”, SPOIE, Optical Engineering, Vol. 41, No. 10, pp. 2476-2478, 2002.
- [14] Yu-Chen Hu, “Predictive Moment Preserving Block Truncation Coding for Gray Level Image Compression”, Journal of Electronic Imaging, Vol. 13, No. 4, pp. 871-877, 2004.