AJMAS
https://journal.utripoli.edu.ly/index.php/Alqalam eISSN 2707-7179

Review Article

Color Image Compression Scheme with Reduced Computational Complexity

Sabriya Alfitouri[1]*, Najat Ali[2], Braika Alameen[3], Abdussalam Ali[4]

[1]*Department of Communications, College of Electronic Technology, Bani Walid, Libya*
[2]*Department of Computers, College of Electronic Technology. Bani Walid, Libya*
[3]*Department of Electric and Electronic Engineering, Bani Waleed University, Bani Walid, Libya*
[4]*Department of Mechanical and Industrial Engineering, Bani Waleed University, Bani Walid, Libya*

ARTICLE INFO

Corresponding Email. *Sabriya.sallheen97@gmail.com*
Received: *06-03-2022* ***Accepted***: *20-03-2022* ***Published***: *24-03-2022*
Keywords: *Color Image, Absolute Block Truncation Coding, Quadtree, Interpolative.*

ABSTRACT

Block Truncation Coding "BTC" is a simple and fast algorithm for coding digital images, which achieves constant bit rate of 2 bits per pixel. The compression may be improved by coding only a half of the bits in the BTC bit plane of each block; the other half will be interpolated. The resulting bit rate will be 1.5 bits per pixel. A low computational complexity compression scheme for coding color images based on Absolute Moment Block Truncation Coding "AMBTC" is presented in this paper. Four techniques are employed in this compression scheme. They are quad tree segmentation, absolute moment block truncation coding bit plane omission, bit plane coding using 32 predefined visual patterns and one of the interpolative bit plane coding techniques. The algorithm has been investigated and applied to different still color images. The simulation results show that the scheme achieves an average bit rate of 0.385 bits per pixel for color images with an average PSNR of 30.71 dB

Cite this article: *Alfitouri S, Ali N, Alameen B, Ali A. Color Image Compression Scheme with Reduced Computational Complexity. Alq J Med App Sci. 2022;5(1):172-176. https://doi.org/0.5281/zenodo.6378415*

INTRODUCTION

Image data compression is necessary to reduce space for storing images and save the bandwidth needed for transmitting them. There several algorithms and techniques for image data compression. The performance of an image compression algorithm may be evaluated in terms of computational complexity, compression ratio, and fidelity. A good algorithm has low-computational complexity, high-compression ratio and high fidelity. Unfortunately, not all three can be achieved simultaneously. Block Truncation Coding (BTC) is an efficient and fast compression technique applicable for gray- scale images [1]. This method gives a relatively good compression ratio and has a simple and fast algorithm [2].

In the basic BTC scheme, the image is first divided into a set of non-overlapping blocks, and the first two statistical moments, mean and variance and the bit plane are computed. In the decoder, each block of the image is reconstructed using the bit plane and two statistical moments [2]. The bit rate achieved is *2* bits/pixel instead of the original 8 bits/pixel, which corresponds to a compression ratio of 4:1. A variant of BTC called Absolute Moment Block Truncation Coding (AMBTC) is introduced by Lema, Mitchell [3] .This version preserves the higher and lower means of the blocks, and the bit rate is the same as that of the original BTC. Several techniques based on the BTC have been introduced to reduce the bit rate and computational requirements while maintaining an acceptable image quality.

In this paper, a low bit rate and low computational complexity color image compression scheme based on AMBTC is presented. This scheme employs four techniques. These techniques are quad tree segmentation, bit plane omission, bit plane coding using 32 predefined visual patterns and interpolative bit plane coding.

Absolute Moment Block Truncation Coding (AMBTC)

In basic BTC the image is divided into non-overlapping blocks. The first and second sample moments of the block are calculated as follows:

$$\overline{X} = \frac{1}{m}\sum_{i=1}^{m} X_i \quad (1)$$

$$\overline{X^2} = \frac{1}{m}\sum_{i=1}^{m} X_i^2 \quad (2)$$

Where X_i represents the pixel value in the image block and m is, the total number of pixels in one block. The variance is given by

$$\sigma^2 = \overline{X^2} - \overline{X}^2 \quad (3)$$

The next step is to find a threshold which is taken as the block mean $\overline{X}$ and two reconstruction levels, *a* and *b* such that:

$$\hat{X} = a \quad if \quad X_i < \overline{X} \quad (4)$$

$$\hat{X} = b \quad if \quad X_i \geq \overline{X} \quad (5)$$

*For i=1, 2, 3… m.*Where $\hat{X}$ is the value of the pixel in the reconstructed block and a and b are given by

$$a = \overline{X} - \sigma\sqrt{\frac{q}{m-q}} \quad (6)$$

$$b = \overline{X} + \sigma\sqrt{\frac{m-q}{q}} \quad (7)$$

Where *q is the* number of pixels greater than or equal to the block mean and *m*- is the total number of pixels in the block. The output of the BTC for each block include two numbers, *a* and *b*, which specify the pixel values greater than the block mean, and pixel values less than or equal to the block mean, respectively. A 4x4 bit plane is formed by replacing a bit (1) for each *b* and a bit (0) for each *a*. The bit plane is transmitted (16 bits) along with the levels *a* and *b* (8 bits each). The total number of bits transmitted is 32 and the resulting bit rate is 2 bits/pixel. The AMBTC algorithm is a simple and fast variant of the BTC that preserves the higher mean and lower mean of an image block [1]. In this algorithm the image is divided into non-overlapping blocks of size 4x4 or 8x8, etc., The average gray level of the block is calculated using equation(1). Pixels in the image block are then classified into two ranges of values. The upper range is the gray levels which are greater than the block mean $\overline{X}$ and the remaining represent the lower range. The mean of higher range X_H and the lower range X_L are calculated as follows:

$$X_H = \frac{1}{k}\sum_{X_i \geq \overline{X}}^{n} X_i \quad (8)$$

$$X_L = \frac{1}{m-k}\sum_{X_i < \overline{X}}^{n} X_i \quad (9)$$

Where m is the total number of pixels in the block and k is the number of pixels with gray levels greater than the block mean $\overline{X}$.

As in the BTC a bit plane is formed by replacing a bit (1) for each pixel whose gray level is greater than or equal to block mean $\overline{X}$ and a bit (0) for each pixel whose gray level is less than the block mean $\overline{X}$ .The bit plane , X_H and X_L are transmitted to the decoder. In the decoder, the image block is reconstructed by replacing each bit (1) by X_H and each bit (0) by X_L. The total number of bits transmitted is 32 and the resulting bit rate is 2 bits/pixel, which is the same as in BTC, however, AMBTC, requires less computations.

Low Computational Computational Complexity Codind Scheme

The low computational complexity color image-coding scheme makes use of quadtree segmentation, AMBTC bit plane omission, bit plane coding using 32 predefined visual patterns and one of the interpolative techniques. In this case, the absolute difference between the higher mean and lower mean of AMBTC method is used as a controlling value to segment the given image. In bit plane omission technique, bit plane of the AMBTC method is omitted in the encoding procedure if the difference between the higher mean and lower mean is less than a threshold value and only the block mean is retained. At the time of decoding bit plane, omitted blocks are replaced by the respective block means. Bit plane coding with visual patterns is the technique to encode bit plane using the predefined 32 visual patterns as in Figure 1. Thus, it needs only 5 bits to code the bit plane. The interpolative technique is the method that drops half of the bit plane at the time of encoding and at the time of decoding the dropped bits are recovered by taking the arithmetic mean of the adjacent pixel values [6]. Referring to Fig.2, the dropped bits are calculated as follows:

B=1 iff two or more the surrounding bits, A, F, and C are equal 1.

D=C.

E=1 iff two or more the surrounding bits, A, F, and I are equal to 1.

G= 1 iff two or more of surrounding bits F, K, H, and C are equal to 1.

J=1 iff two or more of the surrounding bits ,F,I ,N and K are equal to 1.

L=1 iff two or more of the surrounding bits, H, K , and P are equal to 1.

M=N.

O= 1 iff two or more of the surrounding bits, N,K, and P are equal to 1.

0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1	1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0	1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 0 1 1 0 0 1 1 1 1 1 1 1 1	0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1	1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0	0 0 0 0 0 0 0 1 1 0 0 1 1 0 0
1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1	0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0	1 1 1 0 1 1 0 0 1 0 0 0 0 0 0 0	1 1 1 1 1 1 1 0 1 1 0 0 0 1 0 0	1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0
0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0	0 1 1 1 0 0 1 1 0 0 0 1 0 0 0 0	1 1 1 1 0 1 1 1 0 0 1 1 0 0 0 1	1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 1
0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 0	0 0 0 0 1 0 0 0 1 1 0 0 1 1 1 0	1 1 0 0 1 1 0 0 1 1 1 0 1 1 1 1	1 1 0 0 1 1 1 0 1 1 1 0 1 1 1 1
0 0 1 1 0 1 1 1 0 1 1 1 1 1 1 1	0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1	0 0 0 1 0 0 1 1 0 0 1 1 0 1 1 1	0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1

Fig 1. Predefined 32 visual patterns.

D	C	B	A
H	G	F	E
L	K	J	I
P	O	N	M

Fig 2. Bit plane: Bold-faced letters represent the dropped bits.

The steps involved in the compression process are as follows:
1-Divided the given color image into a set of non over lapping blocks, say x of size n=16×16 pixels.
2-Compute the block mean $\overline{X}$, lower mean $\overline{X_L}$ and higher mean $\overline{X_H}$ for a block.
3-Fix a threshold Th1 and if $\left|\overline{X_H} - \overline{X_L}\right| \le Th1$ encode the block x with the block mean $\overline{X}$, put (0) an indicator bit as prefix code for decoding purpose and go to step 14 else continue to step 4**.**
4- Divided the 16x16 block into four non-overlapping blocks (x_b) of size n=8×8 pixels.
5-Compute the block mean $\overline{X_b}$, lower mean $\overline{X_{bL}}$ and higher mean $\overline{X_{bH}}$ for the 8x8 block.
6-If $\left|\overline{X_{bH}} - \overline{X_{bL}}\right| \le Th2$ **,** encode the block (x_b) with the block mean $\overline{X_b}$, put (1) an indicator bit as prefix code for decoding purpose and go to step 7 for the next block else go to step 8**.**
7-Rpeat steps 5 and 6 until all four-sub blocks in this block are encoded, then go to step 14,
8-Divided the 8x8 block into four non-overlapping blocks (x_c) of size n=4×4 pixels.
9- Compute the block mean $\overline{X_c}$, lower mean $\overline{X_{cL}}$ and higher mean $\overline{X_{cH}}$ for the 4x4 block
10-If $\left|\overline{X_{cH}} - \overline{X_{cL}}\right| \le Th3$ **,** encode the block (x_c) with the block mean $\overline{X_c}$, put (01) an indicator bit as prefix code for decoding purpose and go to step 11 for the next block else go to step 12**.**
11-Rpeat steps 9 and 10 until all four-sub blocks in this block are encoded, then go to step 7.
12-Construct the bit plane by taking "1" for the pixels with values greater than or equal to the mean $\overline{X_c}$ and the rest of the pixels are presented by "0". Encode the bit plane using 32 predefined bit plane pattern of 4×4 given in Fig.1 along with lower $\overline{X_{cL}}$ mean and higher mean. $\overline{X_{cH}}$, put (10) an indicator bit as prefix code for decoding purpose and go to step 11 for the next block else go to step 11**.** If the bit plane does not match with 32 visual patterns then go to step 13.
13-Drop a pattern of bits as shown in Fig 2, encode the block by the remaining bits with lower mean $\overline{X_{cL}}$ and higher mean $\overline{X_{cH}}$, put (11) an indicator bit as prefix code for decoding purpose and go to step 11**.**
14-Go to step 2 until all the blocks are processed.

SIMULATION AND RESULT

Four techniques were used to reduce bit rate and computational complexity in coding color images. They are quad tree segmentation, AMBTC bit plane omission, bit plane coding using 32 visual patterns and Interpolative bit plane coding. Several color test image of size 512x512 were used in this simulation. The results are shown in Table (1). From Table (1) it can be seen that the threshold values Th1 =10, Th2=15, Th3=20 give good quality images with a bit rate 0.385 bpp and PSNR of 30.71 dB. Some of the reconstructed images using the proposed technique are illustrated in Figure (3).

Table 1. BPP and PSNR for different thresholds and different test images.

	Threshold									
	5-10-20		10-10-10		10-15-20		15-15-15		15-10-15	
Image	bpp	PSNR	bpp	PSNR	bpp	PSNR	bpp	PSNR	bpp	PSNR
Lena	0.432	30.87	0.532	31.34	0.356	30.51	0.386	30.48	0.401	30.66
Barbie	0.698	25.97	0.784	26.11	0.648	25.89	0.683	25.94	0.729	25.99
Zelda	0.398	32.76	0.499	33.49	0.276	31.85	0.306	31.66	0.403	31.96
House	0.240	34.02	0.281	34.38	0.207	33.55	0.219	33.51	0.238	33.65
Peppers	0.388	31.66	0.461	32.07	0.322	31.17	0.341	31.03	0.399	31.27
Boat	0.758	29.20	0.508	28.53	0.549	31.51	0.550	28.24	0.625	28.76
Average	0.470	31.18	0.564	31.81	0.385	30.71	0.412	30.69	0.475	30.89

Fig 3. Reconstructed color images using the proposed method with threshold 10-15-20.

CONCLUSION

A low bit rate compression scheme for coding color images based on AMBTC is presented in this paper. The compression scheme make use of AMBTC bit plane omission, bit plane coding using 32 predefined visual patterns and one of the interpolative bit plane coding techniques. Simulation results using real color images have proved that this algorithm achieves low bit rate using low computational complexity.

Disclaimer

The article has not been previously presented or published, and is not part of a thesis project.

Conflict of Interest

There are no financial, personal, or professional conflicts of interest to declare.

REFERENCES

1. Delp E, Mitchell O. Image compression using block truncation coding," IEEE Trans on.Comm. 1979;27(9):1335-1341.
2. Mitchell O, Delp E. Multilevel Graphics Representation Using Block Truncation Coding. Proceedings of the IEEE. 1980;68(7):868-873.
3. Lema M. Mitchell O. Absolute moment block truncation coding and its application to color images," IEEE Trans. on Comm. 1984;32:1148-1157.
4. Ramana Rao Y, Eswaran C. A new algorithm for BTC image bit plane coding," IEEE Trans. on Comm. 1995;43(6);2010-2011.
5. Rhoma E, Belgassem F. Improved Algorithms for BTC Image Bit-map Coding with Reduced Bit-rate, Department of Communication, The Higher Institute of Electronics, Bani Walid, Libya, 2006.
6. Somasundaram K. Kasper Raj J. Low Computational Image Compression Scheme Based on Absolute Moment Block Truncation Coding" Proceedings of World Academy of Science, Engineering and Technology. 2006.