Stochastic Modelling and Computational Sciences
Vol. 1 No. 1 (June, 2021)

Received: 15th March 2021 Revised: 25th May 2021 Accepted: 09th June 2021

Modified Coding for image Compression Algorithm

P.K. Kaniadakis, Department of Mathematics, Polytechnic University of Turin, Italy

Abstract

Block Truncation Coding (BTC) is a simple and fast compression technique suitable for image compression and video compression and it has good reconstructed image quality. The drawback of the original BTC algorithm is the high bit rate (2 bits/pixel). In order to reduce the bit rate, we have proposed two compression schemes, one is Look-up table technique for coding the mean and the absolute moment and another is Relative Encoding (RC) for coding the higher mean and the lower mean. An improved bitplane coding is also presented in this paper. The test results show the effectiveness of our proposed method.

Key Words: Block Truncation Coding, Extended Golomb Code, Prediction technique

1. INTRODUCTION

Block Truncation Coding (BTC) [1] is a block based computation free lossy image compression technique. It can be applied for both gray scale images and colour images. Even though BTC is a fast compression technique; it requires 2 bits/pixel. So many techniques: such as prediction technique [7], transform coding [11], vector quantization [10] and entropy coding [6], have been proposed to reduce the bit rate.

In BTC, the bit rate reduction can be achieved in two ways, coding the quantization data (higher mean (a), lower mean (b)) and coding the bit plane. The quantization data is calculated either using statistical moments: mean ($\bar{x}$) and standard deviation (σ) or by using mean ($\bar{x}$) and absolute moment (α) in the case of AMBTC [4]. The quantization data to be stored in the compressed file is either the pair ($\bar{x}$, σ) or ($\bar{x}$, α). They are usually expressed by 8+8 bits or by 6+4 bits, they can also be stored by 10 bits using joint quantization [3], [5]. The other variants of BTC store (a, b) using vector quantization [10], discrete cosine transform [11] and prediction technique with entropy coding [6]. Likewise the bits required for the bit plane can be reduced by prediction technique [7], [9], vector quantization and entropy coding [6]. However the low bit rate can be achieved at the cost of either image degradation or computational complexity.

In this paper, two algorithms are presented to store the quantization data, the first one is a look-up table technique and the second one is a variable length integer coding. An improved bit plane coding is also given.

Presentation of the work in the paper is as follows, section 2 contains a brief survey of BTC. Variable length integer coders are discussed in the section 3, use of variable length integer codes to reduce the bits required for the quantization data is studied in the section 4 and Look-up table technique is also discussed in it. In section 5, an improved bit plane coding and the experimental results are given. Finally conclusions are drawn in section 6.

2. BLOCK TRUNCATION CODING

Here we consider the image matrix of *n* pixels, each represented by 8 bits. In the original BTC [1] the image is divided into sub images of size 4x4 each. For each sub image (block), the mean value ($\overline{x}$) and the standard deviation (σ) are computed and encoded.

$$\overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{1}$$

$$\sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - \overline{x})^2} \tag{2}$$

Using $\overline{x}$ and σ, two quantized values (a, b) are calculated for each block and these are representative of that block. For a block, pixels with values greater than the mean will be represented by 1-bit which indicates the quantized value *a* and for the rest of pixels, 0-bit is stored to denote the quantized value *b.* These 0's and 1's form bit plane for each block. So each block is represented by <a, b, bit plane >.

$$a = \overline{x} + \sigma\sqrt{\frac{m-q}{q}} \tag{3}$$

$$b = \overline{x} - \sigma\sqrt{\frac{q}{m-q}} \tag{4}$$

where m (=16) is the total number of pixels in the block and *q* is the number of 1-bits in the bitplane. A variant of BTC is Absolute moment block truncation coding

(AMBTC) [4] which uses mean and absolute moment (α) to compute the quantized values instead of mean and standard deviation.

$$\alpha = \frac{1}{m}\sum_{i=1}^{m} | x_i - \overline{x} | \tag{5}$$

$$a = \overline{x} + \frac{m\alpha}{2q} \tag{6}$$

$$b = \overline{x} - \frac{m\alpha}{2(m-q)} \tag{7}$$

Another variant of BTC is:

$$a = \frac{1}{q}\sum_{x_i \geq \overline{x}} x_i \tag{8}$$

$$b = \frac{1}{(m-q)}\sum_{x_i < \overline{x}} x_i \tag{9}$$

It has been shown that this technique is an optimal technique among BTC variants.

3. VARIABLE LENGTH INTEGER CODES (VLC)

We propose a new compression technique to further compress the η and the α obtained in AMBTC. We make use of VLC compression technique. VLC is a simple one employed in both text compression and image compression. Since it requires fewer computations in encoding and decoding, it has been used in many compression methods [2], [12]. Our method makes use of Extended Golomb code (EGC) [8] to reduce compression rate.

EGC is a variable length encoder to represent non-negative integers. The EGC for a given integer is generated by dividing it by a selected divisor (*d*) successively until the quotient (q) becomes 0. Count the number of divisions made and retain the remainder obtained at each division. The count is coded by Unary code [8] and the remainders are coded in $\log_2 d$ bits when $q \neq 0$ and in $\log_2 (d-1)$ bits when $q = 0$. Sample EGC for few integers are given in Table 1.

A modified version of EGC (MEGC) for non-negative integer is generated by dividing the integer by a selected *d* until the quotient (*q*) becomes either 0 or 1. The remaining steps to generate MEGC are the same as that of EGC except that the last remainder is coded in $\log_2$ (d-2) bits when the count >=2 and q = 0, and in $\log_2$ (d-1) bits when the count >= 2 and q = 0. MEGC for few integers are given in Table 2.

Table 1
EGC for10 integer numbers

	EGC		
n	*d=2*	*d=3*	*d=4*
1	1	10	11
2	010	11	110
3	011	0100	111
4	00100	01010	01000
5	00101	01011	01001
6	00110	0110	01010
7	00111	01110	01011
8	0001000	01111	011000
9	0001001	001000	011001
10	0001010	0010010	011010

Table 2
MEGC for 10 integer numbers

n	*d=2*	*d=3*	*d=4*
1	0	010	010
2	110	011	0110
3	111	110	0111
4	10100	1110	1100
5	10101	1111	1101
6	10110	0010	1110
7	10111	00110	1111
8	1001000	00111	001000
9	1001001	10100	001001
10	1001010	101010	001010

4. THE PROPOSED METHODS TO CODE THE STATISTICAL MOMENTS (SM) $(\bar{x}, \alpha)$ AND THE QUANTIZATION DATA (a, b)

In this paper, we adopt the look-up table technique (LUT) and Relative Encoding (RC) to code the statistical moments (SM) and the quantization data (a,b) respectively. The LUT method represents the SM in 14 bits with no loss of information and RC method represents *a* and *b* together in 10.05 bits at no loss.

4.1. Look-Up table technique

All block based image compression algorithms decompose images into small blocks of size 4x4 (16 pixels). The reason is the high correlation exiting among the pixels in a block of size 16. We consider only gray scale images and blocks of size 16 in this paper. Our LUT method exploits the property of predicting of maximum possible absolute moment value of a block for the mean of an image block. Using this property, $\overline{x}$ and α are coded together as a Statistical Moment Pair (SMP). For gray scale images, the possible mean values of a block are $\overline{x} = 0, 1, 2, 3\ldots 255$. For each mean value of a block, there will be a possible maximum value (M) of α, so there are $\overline{x} \times M$ SMPs for each $\overline{x}$. In this case, we can easily prove that there are totally 20894 values as shown in Table 3. These pairs are quantized into 16384 (2^{14}) possible pairs, so that we can represent the pairs by 14 bits. Since the value of á will not be greater than 64 in most of the case, there will be no extra distortion when quantizing 20894 SMP into 16384 SMP. Possible maximum value of α (*Pma*) for $\overline{x}$ is calculated in different steps given below.

Table 3
The totally possible statistical moment pair (,á)

Mean Value	*Pma*	*Total SMPs for the given Mean Value*
0	0	0
1	1	1
2	3	3
3	5	5
5	9	9
10	18	18
...	...	...
50	76	76
100	116	116
127	127	127
128	127	127
116	116	116
...	...	...
204	76	76
255	0	0
Total SMPs for all α		20894

Step 1: $T_m = \overline{x} \times bz$

Step 2: $N_m = T_m$ / Max (pixel value of the image): For gray scale, it is 255

Step 3: $R_p = T_m - N_m \mathrm{Max}(P_x)$

Step 4: $Pma = ((N_m - 1) + \overline{x}\ (bz - N_m - 1)\ (\mathrm{Max}(P_x) - \overline{x}) + |R_p|)\ /\ bz\ (N_m \neq 16)$

Where T_m is the total sum of a block, N_m is the number of highest pixel value (255 for gray scale image) that the block might have, R_p is one of the pixel values in the block and *bz* is the block size (16). We can prove that the value of α for a given $\overline{x}$ will not exceed its *Pma* value.

4.2 Relative encoding of Quantization data (a, b)

LUT method represents the SM at the bit rate of 14 bits without extra distortion. Our plan is to reduce the bits required as much as possible without extra loss. In this paper, application of VLC to compress the quantization data (*a*, *b*) is described. We don't apply it to compress the SM. If SM is used in encoding phase, quantization data should be calculated in the decoding phase, so that we don't consider SM in the relative coding and it can be considered in the LUT method alone. Most of the compression techniques have two phases to achieve compression. They are modeling of data and coding of data. In such compression techniques, modeling is significant, so that the compression will be effective. In this paper, quantization levels (*a*, *b*) are coded by simple relative coding by taking advantage of intra block correlations, and the relative values are compressed by MEGC. The blocks used for relative coding are shown in Fig 1. Subtract the lower mean (*b*) from the higher mean (*a*) to make *b* as a small value before using the steps for relative coding. Using of (*a*, *a*-*b*) for relative coding instead of (*a*, *b*), there will be some improvement in compression. The steps used for relative coding for (*a*, *a*-*b*) are given below.

(1) Subtract the values of the pair (a, a-b) of the current block from the corresponding values of the Left Block (if Left Block is present for the current block, otherwise consider only Top Block and skip step 2) and Top Block (if Top Block is available for the current block, otherwise consider only Left Block and skip step 2) and take the difference as $(\Delta a_L, \Delta b_L)$ for Left Block and $(\Delta a_T, \Delta b_T)$ for Top Block.

(2) Find $d_L = |\Delta a_L| + |\Delta b_L|$ and $d_T = |\Delta a_T| + |\Delta b_T|$ and take the difference values which satisfies min (d_L, d_T) for coding. Repeat steps 1 and 2 for all the image blocks.

(3) Code the difference values using simple MEGC.

Not used	Top Block (a, a-b)
Left Block (a, a-b)	Current Block (a, a-b)

Figure 1: Blocks used for Relative Coding

5. AN IMPROVED BIT PLANE CODING

In AMBTC, the bit plane contains 16 bits to represent the quantization data (a, b). The bits used for the bit plane can be reduced using the technique proposed in [7] by Ramana and Eswaran (RE) and [9] by Somasundaram and Kaspar raj (SK). The bits used for prediction in [7], [9] are shown in the Fig. 2. In [7], the bits without encircled are dropped and the dropped bits are predicted from their neighboring bits which are encircled. In [9], the bits without encircled are dropped in encoding phase and in decoding phase, respective pixel values of the bits without encircled are reconstructed from the corresponding neighboring quantized pixel values of the encircled bits. The reconstructed image generated by SK method is having better quality than that of RE method always. RE method may produce better quality image, provided the prediction is a perfect prediction. But the perfect prediction is not possible for all the image blocks. If the prediction of the bits without encircled (except D and M) is exactly as in the AMBTC bitplane, it is called as perfect prediction. The algorithm used in RE method and SK method is given in the section 5.1 and 5.2 respectively.

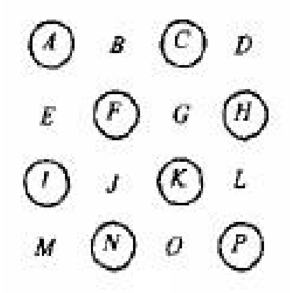

Fig. 2. Bit plane with bits Encircled used for Prediction and others are Dropped

5.1 RE method

The criteria followed for prediction is

1. B = 1 if two or more of neighboring encircled bits A , F and C are 1 otherwise B = 0;

2. D = C;
3. E = 1 if two or more of neighboring encircled bits A, E and I are 1 otherwise E = 0;
4. L = 1 if two or more of H, K and P are 1 otherwise L = 0;
5. O = 1 if two or more of K, P and N are 1 otherwise O = 0;
6. G = 1 if two or more of C, F, H and K are 1 otherwise G = 0;
7. M=N;
8. J = 1 if two or more of F, I, K and N are 1 otherwise J=0;

The above steps are followed for all the blocks of the image.

5.2. SK method

The criteria used for reconstruction of quantized pixel values with respect to the bits without encircled are.

1. $rp(B) = \dfrac{qp(A) + qp(C) + qp(F)}{3}$
2. $rp(D) = \dfrac{qp(C) + qp(H)}{2}$
3. $rp(E) = \dfrac{qp(A) + qp(I) + qp(F)}{3}$
4. $rp(G) = \dfrac{qp(H) + qp(C) + qp(F) + qp(K)}{4}$
5. $rp(J) = \dfrac{qp(I) + qp(N) + qp(F) + qp(K)}{4}$
6. $rp(L) = \dfrac{qp(H) + qp(K) + qp(P)}{3}$
7. $rp(M) = \dfrac{qp(I) + qp(N)}{2}$
8. $rp(O) = \dfrac{qp(N) + qp(K) + qp(P)}{3}$

Where *rp* (*) is the reconstructed value of the corresponding bit in the bit plane and *qp* (*) is the respective quantized value (a or b) of the bit in the bit plane. The above steps are employed for all the blocks of the image.

Example 1:

Original

2	9	12	15
2	11	11	9
2	3	12	15
3	3	4	14

Bit plane

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

Reconstructed (AMBTC)

2	12	12	12
2	12	12	12
2	2	12	12
2	2	2	12

Bit plane after prediction

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

Reconstructed (RE method)

2	12	12	12
2	12	12	12
2	12	12	12
2	2	12	12

Reconstructed (SK method)

2	9	12	12
5	12	12	12
2	7	12	12
2	2	5	12

In AMBTC, the value of $\overline{x}$ = 8; α = 5; *q* = 9; a = 12; b = 2.

MSE = 11.81 (in RE method).

MSE = 3.81 (in SK method).

The quality of the image is measured by M*ean Square Error (MSE)* which is the average of the sum of the square of the errors. The error is measured by subtracting reconstructed pixel value from the original value. Minimum of MSE gives the quality image. So SK method is better than RE method.

5.3. The proposed method

In this section, a new idea is proposed based on the RE method and SK method. In our method, either RE method or SK method is used for bit plane coding. In the case of perfect prediction, RE method is better than SK method (see example 2). So we use both the methods in our algorithm. The steps used in our algorithm are:

Step 1: Select the image block and perform AMBTC and get the bit plane.

Step 2: Apply RE method for the bit plane coding, find out whether it gives the perfect. prediction. If there is perfect prediction, use RE method, otherwise, use SK method. Repeat the steps 1 and 2 for all the image blocks.

The above two steps are repeated for all the image blocks. There will be a small modification in the prediction of D and M. if RE method is employed. D and M are not predicted, but the actual quantized pixel value of D and M is reconstructed using steps 2 and 7 given in the section 5.2.

We have tested our algorithms (coding the quantization data (a, a-b) and bit plane coding algorithm) with four test images Lena512.bmp, Lenna.bmp, Jet.bmp and Peppers.bmp) of size 512x512 with block size16. In the experiment, Relative coding (RC) is used with bit plane coding and LUT method is not employed with bit plane coding (no bit plane compression). From Table 4, It has been shown that LUT method gives better bit rate than AMBTC, and RC method achieves lower bit rate than the other methods given in table 4 and produce good reconstructed images than that of RE and SK methods (Note: higher PSNR gives better quality).

Original

2	9	12	15
2	11	11	9
2	3	3	15
12	3	4	14

Bit plane

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

Reconstructed (AMBTC)

2	12	12	12
2	12	12	12
2	2	12	12
2	2	2	12

Bit plane after prediction

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

Reconstructed (RE method)

2	12	12	12
2	12	12	12
2	2	2	12
2	2	2	12

Reconstructed (SK method)

2	9	12	12
5	12	12	12
2	7	12	12
2	2	9	12

Example 2:

MSE = 9.31 (RE method)

MSE = 16.18 (SK method)

Table 4
Experimental Results

	RE method		*SK method*		*The proposed method for the pair (a, a-b)*		*AMBTC*		*LUT method for the pair (mean, α)*	
Images	*Bitrate (bpp)*	*PSNR*	*Bitrate (bpp)*	*PSNR*	*Bitrate (bpp)*	*PSNR*	*Bitrate (bpp)*	*PSNR*	*Bitrate (bpp)*	*PSNR*
Lena 512	1.50	30.63	1.50	31.97	1.19	32.09	2	33.83	1.875	33.83
Lenna	1.50	31.34	1.50	32.68	1.20	32.81	2	33.10	1.875	33.10
Jet	1.50	30.34	1.50	31.16	1.19	31.64	2	32.90	1.875	32.90
Peppers	1.50	31.09	1.50	31.95	1.20	32.39	2	34.00	1.875	34.00

CONCLUSION

In this paper, we make use of Look-up table method to code the statistical moments and variable length integer code to code the higher mean and the lower mean without any distortion. An improved bitplane coding is also given in this work. The test results prove the effectiveness of our proposed algorithm.

REFERENCES

[1] Delp E.J., Mitchell O.R., Image Coding Using Block Truncation Coding. *IEEE Trans on Comm*, vol. **27 (9)**, pp. 1335-1342, 1979.

[2] Fenwick P. M, "Burrows Wheeler Compression with Variable Length Integer Codes", *Software-Practice and Experience*, Vol. **32** (13), pp. 1307-1316, 2002.

[3] Healy D., Mitchell O.R., "Digital Bandwidth Compression Using Block Truncation Coding," *IEEE Trans on Comm*, vol. **29** (12), pp. 1809-1817, 1981.

[4] Lema M. D. and Mitchell O. R., "Absolute moment block truncation coding and applications to color images," *IEEE Trans on Comm*, vol. **32** (10), pp. 1148-1157, 1984.

[5] Mitchell O.R., Delp E.J., "Multilevel Graphics Representation Using Block Truncation Coding," *Proceedings of the IEEE*, vol. **68 (7)**, pp. 868-873, 1980.

[6] Pasi Franti and Olli Nevalainen, "Block Truncation Coding with Entropy Coding," *IEEE Trans on Comm* Vol. **43**(2-4) Part-3, pp. 1677-1685, 1995.

[7] Ramana Y V, and Eswaran H, "A new algorithm for BTC image bit plane coding," *IEEE Trans on Comm*, **32(10)**, 1148-1157, 1984.

[8] Somasundaram *K. and* Domnic S., "Extended Golomb code for Integer Representation," *IEEE Trans on Multimedia (accepted)*, 2006.

[9] Somasundaram K. and Kaspar raj I., "A Modified Block Truncation Coding for Image Compression, "*Proc .of NCCMM-'05*, Narosa publishers, New Delhi-2006.

[10] Udpikar V., Raina J., "BTC Image Coding Using Vector Quantization," *IEEE Trans on comm*, vol. **35 (10)**, pp. 352-356, 1987.

[11] Wu Y., Coll D.C., "BTC-VQ-DCT Hybrid Coding of Digital Images," *IEEE Trans on comm*, vol. **39 (9)**, pp. 1283-1287, 1991.

[12] Williams H. E. and Zobel J. "Compressing integers for fast file access". *Computer Journal*, **42(3)**, pp. 193-201, 1999.

S. Domnic & K. Somasundaram
Depart ent of Computer Science & Applications
Gandhigram Rural Institute, Gandhigram
Tamilnadu, India 624 302
E-mail: to_domnic@yahoo.co.in