International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 25 Issue 10 – NOVEMBER 2018.

Implementation of DDBTC algorithm for encoding and decoding the image

T. R. Usha [#1] and Ajay Betur P [*2]

[#] *PG Scholar, Dept. of DECS, JNNCE, Shivamogga, India.*

[*] *Asst. Prof., Dept. of ECE, JNNCE, Shivamogga, India*

***Abstract*— Block truncation coding (BTC) has been considered a highly efficient compression technique for decades. However, its inherent artifacts, blocking effect and false contour, caused by low bit rate configuration are the key problems. To deal with these, an improved BTC, namely dot-diffused BTC (DDBTC), is proposed in this paper. Moreover, this method can provide excellent processing efficiency by exploiting the nature parallelism advantage of the dot diffusion, and excellent image quality can also be offered through co-optimizing the class matrix and diffused matrix of the dot diffusion. According to the experimental results, the proposed DDBTC is superior to the former error-diffused BTC in terms of various objective image quality assessment methods as well as processing efficiency. In addition, the DDBTC also shows a significant image quality improvement comparing with that of the former ordered-dither BTC.**

***Index Terms*— Block truncation coding(BTC), image compression, Dot diffusion , Halftoning, Optimization**

I. INTRODUCTION

BLOCK truncation coding (BTC), which was proposed by Delp and Mitchell in 1979 [1], is a technique for image compression. The basic concept of this technique is to divide the original image into many non-overlapped blocks, each of which is represented by two distinct values. In traditional BTC, the two values preserve the first- and second-moment characteristics of the original block. When a BTC image is transmitted, a pair of values (2 × 8 bits/block) and the corresponding bitmap which addresses the arrangement of the two values in each block (1 bit/pixel) are required. Although BTC cannot provide a comparable coding gain to other modern compression techniques, such as JPEG or JPEG2000, the complexity of the BTC is much lower than that of the above techniques.

II. LITERATURE SURVEY

In paper [1] author E. J. Delp, O. R. Mitchell, et al described "Image compression using block truncation coding technique".

A technique for image compression called block coding is derived and compared with transform and other techniques. The BTC algorithm uses a two-level (one-bit) nonparametric quantizer that adapts to local properties of the image. The quantizer that shows great promise is one which preserves the local sample moments. This quantizer produces good quality images that appear to be enhanced at data rates of 1.5 bits picture element. No large data storage is required, and the computations name all. The quantizer is compared with standard (minimum mean-square error and mean absolute error) one-bit quantizers. Modifications of the basic BTC algorithm are discussed along with the performance of BTC in the presence of channel errors.

In paper [2] author D. R. Halverson, N. C. Griswold, G. L. Wise, et al described "A generalized block truncation coding algorithm for image compression,"

Block Truncation Coding is a recently developed approach to image compression whose design is specified by the appropriate moment preserving quantizer. In this paper we show how the basic Block Truncation Coding algorithm can be generalized to include a family of moment preserving quantizers with the potential for improved performance. We then illustrate by way of example that such improvement is indeed possible from the standpoint of peak signal to noise ratio. There is a subclass of this family of moment preserving quantizers for which practical difficulties in implementation exist; however, we show that frequently we can avoid this subclass and still obtain good performance.

In paper [3] authors V. Udpikar, J. P. Raina, et al described "Modified algorithm for block truncation coding of monochrome images,".

Block truncation coding (BTC) is a recent technique used in the coding of image data. In the letter a modified technique for BTC coding of image data is presented which is algorithmically simple and hence easy to implement. This new technique uses only the first-order statistical information as 'block overhead'. The new algorithm is shown to be optimum in the mean-square sense for a particular class of BTC algorithms. The letter presents the results of using the new algorithm for a typical image and compares the performance with that of the earlier algorithm for the same image.

In paper [4] author Q. Kanafani, A. Beghdadi, C. Fookes, et al described "Segmentation-based image compression using BTC-VQ technique,".

This paper proposes a new approach to image compression based on image segmentation using the EM algorithm and combined with BTC (Block Truncation Coding) and VQ (Vector Quantization). The main idea is to decompose the image into homogeneous and non-homogeneous blocks and then compress them using BTC or VQ. This block classification is achieved using an image segmentation based on the EM (Expectation-Maximization) algorithm. The use of the EM algorithm results in a good robust segmentation with well behaved boundaries. The segmented image is then used to specify whether BTC or VQ is used to encode a block by assessing if it contains all pixels from a homogeneous or non-homogeneous region. BTC provides a simple and effective method for coding blocks which contain a lot of information or distinct edges due to its two-level quantizer. However, its lowest attainable hit rate is limited and it often introduces blocking effect in homogeneous regions. VQ on the other hand is more efficient due to a multilevel quantizer and thus results in better compression ratios. However, it does not retain any spatial information about the edges, resulting in stair casing effects. Previous attempts to combine both techniques into a hybrid algorithm only make use of simple measures such as image variance. Results for medical images show that this approach yields significant improvements over traditional BTC or VQ coding when used alone.

In paper [5] author S. Horbelt , J. Crowcroft,ep al described "A hybrid BTC/ADCT video codec simulation bench,".

We compare the performance of video compression algorithms in terms of compression ratio, video distortion and computation time in order to study the feasibility of a real time software videophone for 64 kbits/s channels, with flexible user defined compression methods. We can combine and adjust the compression algorithms to the required bandwidth and video quality. We introduce the double logarithmic CORT Diagram, which is a novel, visual method to combine and evaluate video compression algorithms. The implemented algorithms are subsampling, block truncation coding (BTC), adaptive discrete cosine transform (ADTC) and conditional refreshment. 1 Introduction is We compare the performance of video compression algorithms in terms of compression ratio, video distortion and computation time in order to study the feasibility of a real time software videophone for 64 kbits/s channels, with flexible user defined compression methods.

III. Block Truncation Coding

Block truncation coding is a simple and fast lossy compression technique for digitized gray scale images. Compared to other image compression techniques, BTC requires less computational effort and has good capability of combating channel errors. The key idea of BTC is to perform moment preserving quantization for blocks of pixels so that the quality of image will remain acceptable and at the same time the demand for the storage space will decrease. Even if the compression gain of the BTC algorithm is inferior to the standard JPEG algorithm, BTC has gained popularity due to its special usefulness. Several improvements of the basic method have been recently proposed in the literature. In BTC, the input image of size m x m pixels is divided into blocks of size 4 x 4 pixels. Each block is processed independent of each other. The mean value are calculated for each block using the equations:σx and the standard deviation.

$$\bar{x} = \frac{1}{k} \sum_{i=1}^{k} x_i$$

$$\overline{x^2} = \frac{1}{k} \sum_{i=1}^{k} x_i^2$$

$$\sigma = \sqrt{\overline{x^2} - (\bar{x})^2}$$

where k is the number of pixels in a block.

A two-level quantization is done by transforming each pixel value into either 1 or 0. If the pixel value xi is greater than or equal to the mean x , then the pixel value xi is transformed to 1 otherwise the 0. This collection of 1's and 0's is called a bit-plane are preserved and are σ(B) of that block. The two statistical moments x and transmitted or stored along with the bit-plane. Hence, a compressed block is a set of and B}. For an uncompressed image, each block of pixels requires 16 x 8 =σ{ x , 128 bits (in case of a gray scale image, where the bpp is 8) per block. But for an σimage compressed using BTC, a block requires only 32 bits (8 bits for x , 8 bits for and 16 bits for the bit-plane) leading to a bit-rate of only 2 bpp.

In general, the bit-rate is computed as:
(b + b + k)/k= 1 + 2m/k bits per pixel (bpp)

Where, b = Log2(L), L is the maximum gray level intensity and k is the number of pixels in a block. The value of L is 256 for a gray scale image with 256 gray levels. While decoding, two quantizing levels q1 and q2 are calculated using the equations

IV. BTC Algorithm

Step1: Divide the image into small non-overlapping blocks of size 4 x 4 pixels. of the block using the equationsσ
Step2: Compute the statistical moments x and 1, 2 and 3.
Step3: Generate the bit-plane with 0's and 1's. .σ
Step4: Transmit or store the bit-plane, x
Step5: Repeat the steps 2 through 4 for all the blocks of the input image.

V. DDBTC

The proposed DDBTC is an improved version of the traditional BTC algorithm, thus the traditional algorithm will be firstly introduced for a better comprehension. Given an original image of size P × Q, and which is divided into many non-overlapped blocks of size M × N, then each block can be processed independently and eventually represented by two values. The independent processing property yields the additional excellent parallelism advantage. To begin with, the first-, second-moment, and the corresponding variance are obtained by

$$\overline{x} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} x_{i,j},$$

$$\overline{x^2} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} x_{i,j}^2,$$

$$\sigma^2 = \overline{x^2} - (\overline{x})^2,$$

where the variable xi, j denotes the grayscale pixel value in a block. Since BTC is a one-bit quantizer with a threshold x to

$$q_1 = \overline{x} - \sigma\sqrt{\frac{q}{(16-q)}}$$

$$q_2 = \overline{x} + \sigma\sqrt{\frac{(16-q)}{q}} ,$$

Where, q denotes the number of 1's in the bit plane. Each 1 in the bit-plane is replaced with q2 and 0 is replaced with q1. This gives the reconstructed image. This method is fast, easy to implement and has low computational complexity. binarize the block, the block is then replaced by bitmap as defined below

$$h_{i,j} = \begin{cases} 1, & \text{if } x_{i,j} \geq \overline{x} \\ 0, & \text{if } x_{i,j} < \overline{x}' \end{cases}$$

$$h_{i,j} = \begin{cases} 1, & \text{if } x_{i,j} \geq \overline{x} \\ 0, & \text{if } x_{i,j} < \overline{x}' \end{cases}$$

and the reconstructed result is obtained as

$$y_{i,j} = \begin{cases} b, & \text{if } h_{i,j} = 1 \\ a, & \text{if } h_{i,j} = 0' \end{cases}$$

where the variable hi, j denotes the bitmap, which is employed to address the arranged positions of low mean (a) and high mean (b). The concept of the BTC is to preserve the first and second-moments of a block when the original value is substituted by its high or low means. Thus, the following two equations should be maintained.

$$m\overline{h} = (m-q)\,a + qb,$$

$$m\overline{h}^2 = (m-q)\,a^2 + qb^2,$$

where m = M×N, and q denotes the number of pixels greater than x. The high and low means can be evaluated as follows

$$a = \overline{x} - \sigma\sqrt{\frac{q}{m-q}},$$

$$b = \overline{x} + \sigma\sqrt{\frac{m-q}{q}}.$$

such as parallel processing characteristic and the substitution of the two distinct values in a block. First, the corresponding maximum and minimum are obtained for the proposed method as below:

$$x_{\max} = \max(B),$$

$$x_{\min} = \min(B),$$

where the vector B denotes a divided original block. The proposed algorithm has two main differences to that of the traditional BTC: 1) The high mean and low mean are replaced by the local maximum (xmax) and minimum (xmin) in a block, because the high dynamic range (xmax − xmin) can easily destroy the blocking effect and false contour, and 2) the manner of bitmap generation is replaced by the dot-diffused half toning as detailed below. Suppose the original image and the divided block are of sizes P × Q and M × N, respectively, and each block can be processed independently. For each block, the processing order of pixels is defined by the class matrix as shown in Table I(a) and (c). For example, if the class matrix in Table I(a) is adopted, the original image is divided into blocks of the same size 8 × 8 as that of the class matrix. Each divided block maps to the same class matrix, and all of pixels associated with number zero in the class matrix are processed firstly

$$v_{i,j} = x_{i,j} + x'_{i,j}, \quad \text{where } x'_{i,j} = \sum_{(m,n)\in R} \frac{e_{i+m,j+n} \times k_{m,n}}{sum},$$

$$e_{i,j} = v_{i,j} - y_{i,j}, \quad \text{where } y_{i,j} = \begin{cases} 0, & \text{if } v_{i,j} < 128 \\ 255, & \text{if } v_{i,j} \geq 128 \end{cases}$$

where the variable xi, j denotes the current input grayscale value, variable xi, j denotes the diffused error accumulated from neighboring processed pixels, and variable vi, j denotes the modified grayscale output. The variable yi, j denotes the binary output in the bitmap, and variable ei, j denotes the difference between the modified grayscale output vi, j and the binary output yi, j. The variable km,n denotes the diffused weighting, and R denotes the support region of diffused weighting, with a suggested size of 3 × 3 as in Knuth's [2] and MeseVaidyanathan's dot diffusion [3]. The diffused weighting can be represented as:

$$\begin{bmatrix} k_{-1,-1} & k_{-1,0} & k_{-1,1} \\ k_{0,-1} & x & k_{0,1} \\ k_{1,-1} & k_{1,0} & k_{1,1} \end{bmatrix}.$$

The variable x denotes the pixel currently being processed. Nota associates to the numbers in the class matrix with a greater value than its own associated value.These are the pixels that have yet to be thresholded. The variable sum is the summation of the diffused weights corresponding to those unprocessed pixels.

$$sum = \sum_{m=-1}^{1} \sum_{n=-1}^{1} \begin{cases} k_{m,n}, & \text{if } c_{i+m,j+n} > c_{i,j} \\ 0, & \text{if } c_{i+m,j+n} < c_{i,j}, \end{cases}$$

where the variable ci, j denotes the coefficient value in the class matrix. Fig. 1 shows an example which includes four independent blocks split by the thick lines.

The class matrix in Table I(a) is employed. In this example, the central position with number 34 is the current processing position, and those numbers in gray represent the processed pixels. The arrows represent the possible diffusing directions. Since the pixels with numbers smaller than 34 are processed, the total number of diffusing directions is four, and thus the variable sum = k−1,−1 + k−1,0 + k0,−1 + k1,1 in this case. Notably, the error not only can diffuse to the self block, but also can diffuse to its neighboring blocks

(a) 8 × 8 Class Matrix

42	47	46	45	16	13	11	2
61	57	53	8	27	22	9	50
63	58	0	15	26	31	40	30
10	4	17	21	3	44	18	6
14	24	25	7	5	48	52	39
20	28	23	32	38	51	54	60
19	33	36	37	49	43	56	55
12	62	29	35	1	59	41	34

(b) Diffused Matrix for the 8 × 8 Class Matrix

0.27163	1	0.27163
1	x	1
0.27163	1	0.27163

(c) 16 × 16 Class Matrix

6	7	20	10	53	55	66	87	137	142	143	144	172	122	175	164
3	9	23	50	60	51	65	74	130	145	138	148	179	180	214	221
0	14	24	37	67	79	96	116	39	149	162	198	12	146	224	1
15	26	43	28	71	54	128	112	78	159	177	201	208	223	225	242
22	4	48	32	94	98	80	135	157	173	113	182	222	226	227	16
40	85	72	83	104	117	163	133	168	184	200	219	244	237	183	21
47	120	101	105	123	132	170	176	190	202	220	230	245	235	17	41
76	73	127	109	97	134	178	181	206	196	229	231	246	19	42	49
103	99	131	147	169	171	166	203	218	232	243	248	247	33	52	68
108	107	140	102	185	167	204	217	233	106	249	255	44	45	70	69
110	141	88	75	192	205	195	234	241	250	254	38	46	77	5	100
111	158	160	174	119	215	207	240	251	252	253	61	62	93	84	125
151	136	189	199	197	216	236	239	25	31	56	82	92	95	124	114
156	188	191	209	213	228	238	29	36	59	64	91	118	139	115	155
187	194	165	212	2	13	30	35	58	63	90	86	152	129	154	161
193	210	211	8	11	27	34	57	18	89	81	121	126	153	150	186

(d) Diffused Matrix for the 16 × 16 Class Matrix

0.305032	1	0.305032
1	x	1
0.305032	1	0.305032

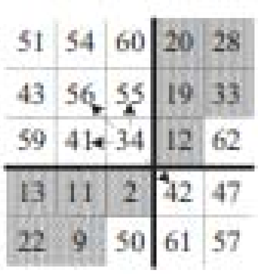

Fig 1: Diffusion between blocks using class matrix in Table I(a)

Above figure demonstrated example of diffusion between blocks using the class matrix in Table I(a). Since the dot diffusion in DDBTC is different from the traditional one, such as the pixel depth of the represented values.

$$e_{i,j} = v_{i,j} - y_{i,j}, \quad \text{where } y_{i,j} = \begin{cases} x_{\min}, & \text{if } v_{i,j} < \bar{x} \\ x_{\max}, & \text{if } v_{i,j} \geq \bar{x}' \end{cases}$$

For the current stage, DDBTC cannot provide better image quality than that of EDBTC for the following two reasons:

1) The class matrix and the diffused matrix employed in traditional dot diffusion [2], [3] are designed for two-tone output, while the DDBTC generates multi-tone output when the bitmap is replaced with the maximum and minimum values of the block.

2) The threshold employed in the traditional dot diffusion is a fixed number 128. This is different from that used in DDBTC, in which adaptive mean values are employed for each block. Consequently, we proposed a co-optimization procedure for the class matrix and diffused matrix to further improve image quality. Herein, the human-visual peak signal-to-noise ratio (HPSNR; named HVS-PSNR in [5]) is adopted as the cost function for IQA as defined in

$$\text{HPSNR} = 10\log_{10} \times \frac{P \times Q \times 255^2}{\sum_{i=1}^{P} \sum_{j=1}^{Q} \left[\sum_{(m,n)\in R} w_{m,n} \left(x_{i+m,j+n} - y_{i+m,j+n} \right) \right]^2},$$

where xi, j and yi, j denote the original image and reconstructed image of size P × Q, respectively; the variable wm,n denotes the filter to simulate the low-pass property of HVS

Algorithm:

Step 1. The Mese-Vaidyanathan’s class matrix [3] is adopted as the initial Class matrix (C) in this optimization.

Step 2. Suppose the coefficients in the class matrix are collocated as a 1D sequence. Successively swap each member C(i) in the class matrix with one of the other 63 members C(j) (suppose the size of the class matrix is 8 × 8), where i _= j .

Step 3. Generate all potential diffused weightings km,n K by adjusting 10−6 within a range of 0 to 1. During the generation of diffused weighting, the nearest vertical and horizontal weights are fixed as 1, and the other four diagonals are kept at the same value.

Step 4. Evaluate the average HPSNR using (6) with eight natural grayscale images of size 512 × 512, Lena, Mandrill, Peppers, Milk, Tiffany, Airplane, Lake, and Shuttle, with swapped class matrices (Step 2) and switched diffused weightings (Step 3).

Step 5. The successive combination of both swapped class matrices and switched diffused weightings is capable of achieving the highest DDBTC image quality by max(HPSNR (swapped C, switched K). These are then employed as the new class matrix and diffused matrix candidate.

Step 6. Select another member C(i), and perform Steps 2 to 5.

Step 7. If all swapped class matrices and switched diffused weightings cannot improve HPSNR, then terminate this optimization. Otherwise, perform Steps 2 to 6.

VI. Results and Discussion

In this literature survey, the content based image retrieval techniques using Block Truncation Coding based compression schemes for color image has been scrutinized. Four techniques were selected for compression specifically, the typical Block Truncation Coding (BTC), Error Diffusion Block Truncation Coding (EDBTC), Ordered Dither Block Truncation Coding (ODBTC) and Dot Diffusion Block Truncation Coding (DDBTC). The BTC based techniques are suitable for image retrieval requiring fast execution since their simplicity, low computational burden, fault tolerance, the relatively high compression efficiency and good image quality of the decoded image. This survey starts with a brief introduction about CBIR basic concepts and its usage. In this survey find out that BTC based CBIR not only applied for gray scale image it can be extended for color image. For future study, these image retrieval schemes can be applied to Satellite Image Retrieval and Video Retrieval. New features can be added by extracting the BTC based compressed images, not only CCF/CHF and BPF/BHF, to enhance the retrieval performance. Since similarity distance is the heart of image retrieval system, an effective and suitable similarity distance can be applied to improve the overall performance. The user relevance feedback scheme can be added to retrieve more meaningful result.

VII. Conclusion

This project presents a dot-diffused-based BTC image compression technique which can yield excellent image quality (even superior to that of the EDBTC_F), processing speed (faster than that of the EDBTC_F about 696× for block of size 8×8, and around 164× for block of size 16×16), and artifact free results (inherent blocking effect and false contour artifacts of the traditional BTC) simultaneously. The performance can be attributed to the use of the inherent parallelism of the dot diffusion and the proposed co-optimization procedure over the class matrix and diffused matrix. As documented in the experimental results, the proposed DDBTC is superior to EDBTC in terms of image quality and processing efficiency, and has much better image quality than that of the ODBTC. Thus, the proposed DDBTC has important values and impacts in prospective highly efficient or low powerless compression communication and related applications.

Acknowledgment

The Authors would like to thank the management and the Principal, the Dept. of E&CE, JNNCE, Shivamogga for providing all the resources and support to carry out the work.

References

[1] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Trans. Commun.*, vol. 27, no. 9,pp. 1335–1342, Sep. 1979.

[2] D. R. Halverson, N. C. Griswold, and G. L. Wise, "A generalized block truncation coding algorithm for image compression," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 3, pp. 664–668, Jun. 1984.

[3] V. Udpikar and J. P. Raina, "Modified algorithm for block truncation coding of monochrome images," *Electron. Lett.*, vol. 21, no. 20, pp. 900–902, Sep. 1985.

[4] Q. Kanafani, A. Beghdadi, and C. Fookes, "Segmentation-based image compression using BTC-VQ technique," in *Proc. Int. Symp. Signal Process. Appl.*, vol. 1. Jul. 2003, pp. 113–116.

[5] S. Horbelt and J. Crowcroft, "A hybrid BTC/ADCT video codec simulation bench," in *Proc. 7th Int. Workshop Packet Video*, Mar. 1996, pp. 18–19.

[6] M. Kamel, C. T. Sun, and G. Lian, "Image compression by variable block truncation coding with optimal threshold," *IEEE Trans. Signal Process.*, vol. 39, no. 1, pp. 208–212, Jan. 1991.

[7] L. G. Chen and Y. C. Liu, "A high quality MC-BTC codec for video signal processing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 1, pp. 92–98, Feb. 1994.

Author

Ms.T R Usha is pursuing Master of Technology in Digital Electronics and Communication System at JNNCE, Shivamogga. under VTU, Belagavi. Her area of interest in Image Processing, VLSI design, CMOS designs, Computer communication network and wireless communication.