A Review of Block Truncation Coding Using Digital Halftoning

Riyaz Ahmad Dar

M.Tech Student, Dept. of ECE, Srinivas Institute of Technology, Mangalore, India.

ABSTRACT: Images are an important part of today's digital world. However, due to the large quantity of data is needed to represent modern imagery; the storage of such data can be expensive. Thus, the work on efficient image storage (image compression) has the potential to reduce storage costs and enable new applications. Block Truncation Coding (BTC) is a lossy image compression technique which uses moment preserving quantization method for compressing digital grayscale images. Even though this method retains the visual quality of the reconstructed image with good compression ratio, it shows some artifacts like blocking effect and false contouring. In this paper, various schemes of block truncation coding using digital halftoning proposed to improve the quality of BTC will be discussed.

KEYWORDS: Block Truncation, Digital Halftoning, Dot Diffusion, Error Diffusion.

I. INTRODUCTION

Block truncation coding (BTC) was first proposed by Delp and Mitchell in 1989. Block truncation coding is a lossy type of image compression. In block truncation coding (BTC), the original image is divided into fixed-size non overlapping blocks of size $M \times N$. The block size chosen is usually small to avoid the edge blurring and blocking effect. Each block is independently coded using a two level (1-bit) quantizer. The two values preserve the first and the second moment characteristic of the original block. BTC does not provide a higher gain than any of the modern image compressing algorithms like JPEG or JPEG-2000, but it is much lesser complex.

Digital Halftoning is a technology of converting a continuous tone image to a two tone image. A continuous tone image and a halftoning image are similar when the low-pass nature of the Human Visual System (HVS) is the perceived device. There are many kinds of halftoning techniques, including ordered dithering, error diffusion, and dot diffusion.

II. BLOCK TRUNCATION CODING

Block truncation coding (BTC) was first proposed by Delp and Mitchell in 1989 [1]. As stated earlier, given an image of size $P \times Q$, BTC divides the original image into non overlapped blocks of size $M \times N$. These blocks are processed independently and coded using a two-level quantizer. Thus each block will be transmitted as a set of two values. The first- and second-moment of the block are calculated as

$$\bar{x} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} x_{i,j} ,$$ eq. (1)

$$\overline{x^2} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} x^2{}_{i,j}.$$ eq. (2)

The corresponding variance is obtained by

$$\sigma^2 = \overline{x^2} - (\bar{x})^2,$$ eq. (3)

where the variable $x_{i,j}$ represents the pixel values in the block. BTC is a 1-bit quantizer with threshold $\bar{x}$, therefore, the bitmap is calculated as

$$h_{i,j} = \begin{cases} 1, if \ x_{i,j} \geq \bar{x} \\ 0, if \ x_{i,j} < \bar{x}, \end{cases}$$ eq. (4)

and the reconstructed result is given by

$$y_{i,j} = \begin{cases} a, if \ h_{i,j} = 0 \\ b, if \ h_{i,j} = 1. \end{cases}$$ eq. (5)

The variables a and b are given by:

$$a = \bar{x} - \sigma \sqrt{\frac{q}{m-q'}} \qquad\qquad\qquad \text{eq. (6)}$$

and

$$b = \bar{x} + \sigma \sqrt{\frac{m-q}{q}}. \qquad\qquad\qquad \text{eq. (7)}$$

where $m = M \times N$, q denotes the number of pixels greater than $\bar{x}$ and q' is the number of pixels less than $\bar{x}$.
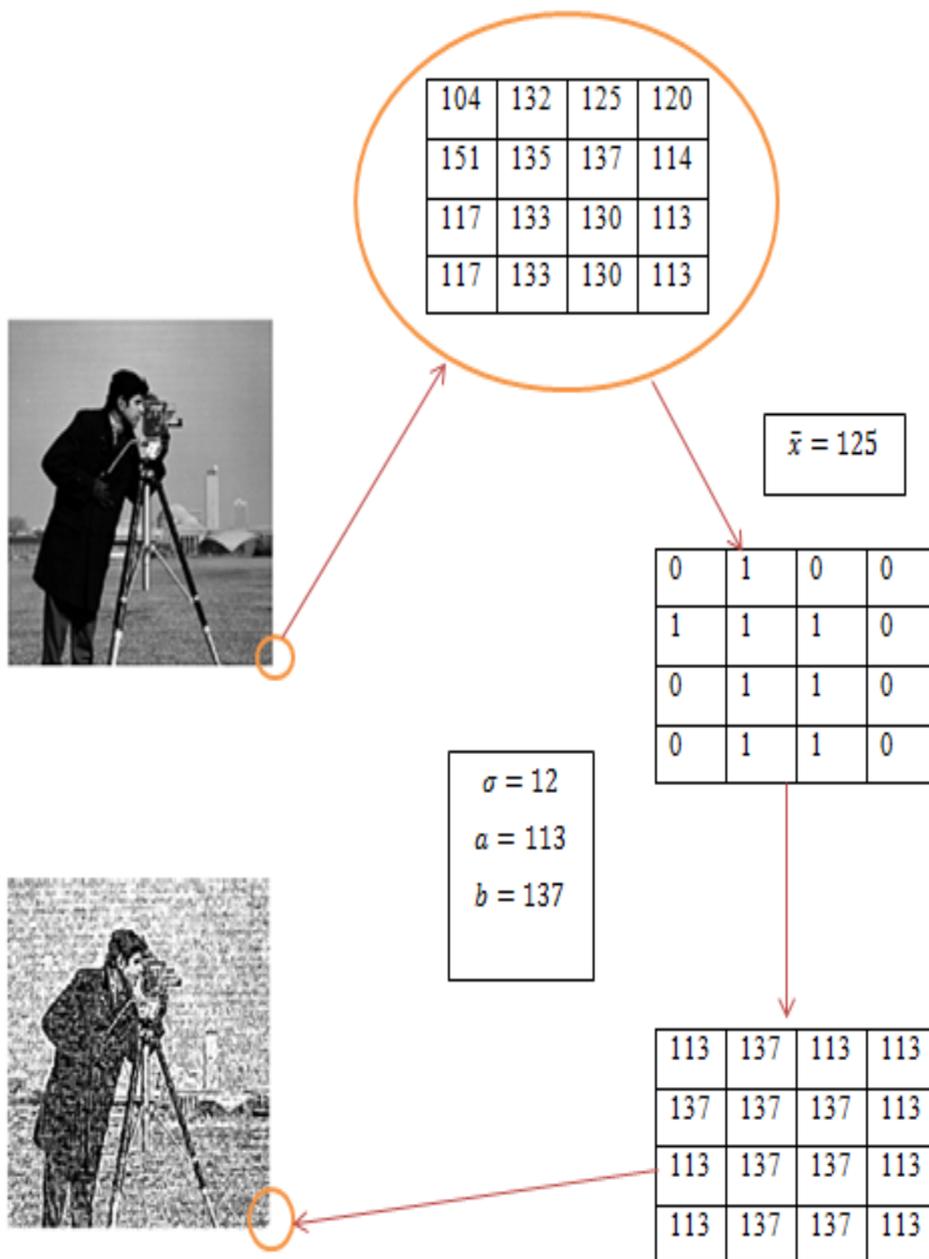
The above figure shows the processing of a block in BTC. The mean is calculated for each block and each pixel in a block is compared to the mean. If the pixel value is greater than the mean, it is recorded as a "1" and if the value is smaller than the mean, it is recorded as a "0" in the bitmap. While converting the bitmap into the final compressed image; for a "0" in the bitmap, it is replaced by the value of "a" which is calculated using eq. 6 and for a "1" in the bitmap, it is replaced by the value of "b" which is calculated using eq. 7.

III. BLOCK TRUNCATION CODING USING DIGITAL HALFTONING

Many endeavors have been addressed in the literature to improve the computational complexity, compression ratio, and image quality of the BTC. In [2], the author used the error diffusion technique. Error diffusion enjoys the benefit of diffusing the quantised error into the neighbouring pixels. The error diffusion can effectively diffuse the error between the neighboring pixels and then maintains the average grayscale in a local region.

The original image is first divided into $M \times N$ blocks. The average value of the block is given by eq. 1. The maximum and minimum are given by:

$$x_{max} = \max(B), \qquad \text{eq. (8)}$$
$$x_{min} = \min(B), \qquad \text{eq. (9)}$$

where the B denotes the divided original block.

The pixels are processed according to the equations given below:

$$v_{i,j} = x_{i,j} + x'_{i,j}, where\ x'_{i,j} = \sum_{m=0}^{2} \sum_{n=-2}^{2} e_{i+m,j+n} \times h_{m,n} \qquad \text{eq. (10)}$$

$$e_{i,j} = v_{i,j} - o_{i,j}, where\ o_{i,j} = \begin{cases} x_{min} & if\ v_{i,j} < \bar{x} \\ x_{max} & if\ v_{i,j} \geq \bar{x} \end{cases} \qquad \text{eq. (11)}$$

The variable $x_{i,j}$ denotes the current input pixel value, and $x'_{i,j}$ denotes the diffused error sum added up from the neighboring processed pixels. The variable $o_{i,j}$ denotes the binary output in position (i,j), and the error kernel $h_{m,n}$ is used to diffuse the error caused by the difference, $e_{i,j}$ between the output binary value and the input grey level value. The variable $v_{i,j}$ denotes the modified value. EDBTC uses a threshold of 128.

In [3], the authors used the void and cluster dithering approach. Using the void-and-cluster halftoning, the image quality is improved when operated in high coding gain applications. The dither array in order dithering is employed to substitute the fixed average threshold in BTC, and the extreme pixel values in a block are adopted to substitute the high mean and low mean.

Consider an image of size $P \times Q$. ODBTC will divide the image into blocks of size $M \times N$. The maximum and minimum values of the block are given by equations 8 and 9 respectively. The size of the dither array is $M \times N$, which the same as the divided block in BTC. ODBTC can be written as

$$r_{i,j} = \begin{cases} x_{max}, & if\ x_{i,j} \geq DA^{(k)}_{i\ mod\ M, j\ mod\ N} + x_{min} \\ x_{min}, & if\ x_{i,j} < DA^{(k)}_{i\ mod\ M, j\ mod\ N} + x_{min} \end{cases}, \qquad \text{eq. (12)}$$

where $r_{i,j}$ represents the ODBTC result; $x_{i,j}$ represents the original image, and $DA^{(k)}_{i\ mod\ M, j\ mod\ N}$ represents void-and-cluster dither array, and $k = x_{max} - x_{min}$.

A significant feature of the ODBTC is the dither array lookup table (LUT), where each specific size dither array has its corresponding 255 different scaling versions. The 255 scaling versions are obtained by

$$DA^{(k)}_{m,n} = k \times \frac{DA_{m,n} - DA_{min}}{DA_{max} - DA_{min}} \qquad \text{eq. (13)}$$

where $1 \leq k \leq 255, 1 \leq m \leq M,$ and $1 \leq n \leq N; DA_{min}$ and DA_{max} represent the minimum and maximum values in dithered array.

The dot diffusion method for halftoning, is an attractive method which attempts to retain the good features of error diffusion while offering substantial parallelism. The dot diffusion method for halftoning has only one design parameter, called the class matrix C. It determines the order in which the pixels are processed. Thus, the pixel positions (n_1, n_2) of an image are divided into *IJ* classes For example, if we use a class matrix of size I = J = 8, then there will be 64 class members in that class matrix. We then look at the eight neighbours of (n_1, n_2) and replace each constant tone pixel with an adjusted version for those neighbours which have a higher class number (i.e., those neighbours that have not been processed yet) [4].

DDBTC algorithm [5] is similar to the traditional BTC algorithm with a few changes. This algorithm has two main differences to that of the traditional BTC:

1.   The high mean and low mean are replaced by the maximum x_{max} and minimum x_{min} in a block.
2.   The bitmap generation is done using the dot-diffused halftoned.

The maximum and minimum values of the block are given by equations 8 and 9 respectively. Consider an image of size $P \times Q$. DDBTC will divide the image into blocks of size $M \times N$. These blocks are then independently coded. The order of processing of a pixel in a block is predefined by a class matrix. Each divided block corresponds to the same 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

The above table is the class matrix for an 8×8 block which determines the order of processing of pixels which was generated by Guo and Liu in their paper for optimized dot diffusion [5].The pixels numbered "0" will be processed first.

The pixels in a block are processed according to the equations given below:

$$v_{i,j} = x_{i,j} + x'_{i,j}, where \ x'_{i,j} = \sum_{(m,n) \in R} \frac{e_{i+m,j+n \times k_{m,n}}}{sum} \qquad \text{eq. (14)}$$

$$e_{i,j} = v_{i,j} - y_{i,j}, where \ y_{i,j} = \begin{cases} x_{min}, if \ v_{i,j} < \bar{x} \\ x_{max}, if \ v_{i,j} > \bar{x} \end{cases}. \qquad \text{eq. (15)}$$

Where the variables $\bar{x}, x_{min}$, and x_{max} are given by equations 1, 8 and 9 respectively. $x_{i,j}$ is the input original image pixel, $x'_{i,j}$ is the diffused error accumulated from the neighboring pixels, and the variable $v_{i,j}$ is the processed output pixel. The variable $y_{i,j}$ denotes the binary output in the bitmap, and variable $e_{i,j}$ denotes the difference between the modified grayscale output $v_{i,j}$ and the binary output $y_{i,j}$. The variable $k_{m,n}$ denotes the diffused weighting, and R denotes the support region [6].

The variable *sum* is given by:

$$sum = \sum_{m=-1}^{1} \sum_{n=-1}^{1} \begin{cases} k_{m,n}, & if\ c_{i+m,j+n} > c_{i,j}, \\ 0, & if\ c_{i+m,j+n} < c_{i,j}, \end{cases} \qquad \text{eq. (16)}$$

where the variable $c_{i,j}$ denotes the coefficient value in the class matrix.

IV. **EXPERIMENTAL RESULTS**

The test was conducted on a set of three grayscale images: Cameraman, Tire and Coins. The simulation was carried in Matlab[TM].

The following figure shows the cropped results of the test images. The images are compressed by both BTC and DDBTC algorithms and the results are displayed for comparison.

Original image   BTC   DDBTC

Cameraman

Tire

Coins
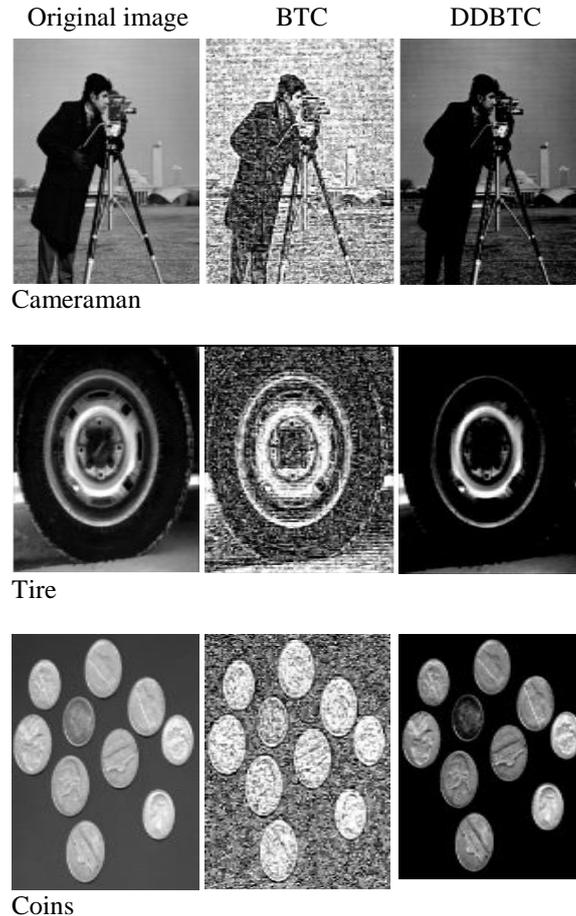
The first column in the above figure shows the thumbnails of the original images, the second column shows the thumbnails of these images compressed by the BTC algorithm, and the third column shows the thumbnails of the same images compressed by the DDBTC algorithm.

V. CONCLUSION

This study presented various digital halftoning-based BTC image compression techniques which can provide an excellent image quality and artifact-free results (inherent blocking effect and false contour artifacts of the traditional BTC) simultaneously.

Two algorithms were selected namely, the original block truncation coding (BTC) and Dot Diffused Block Truncation Coding (DDBTC) to generate the experimental results. The two algorithms are based on dividing the image into non overlapping blocks. The two techniques were applied to different grey level test image each containing 256×256 pixels.

As is evident from the experimental results, the image quality of the DDBTC is superior to BTC. The performance can be attributed to parallel processing of the blocks in the dot diffusion.

REFERENCES

1.   E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," IEEE Trans. Communications, vol. 27, no. 9, pp. 1335-1342, Sept. 1979.
2.   J. M. Guo "Improved block truncation coding using modified error diffusion," Electronics Letters, vol. 44, no.7, pp. 462-464, Mar. 2008.
3.   Guo and Wu, "Improved Block Truncation Coding based on the Void and Cluster Dithering Approach," IEEE Transactions on image processing, vol. 18, no. 1, pp. 211-213, Jan. 2009.
4.   M. Mese and P. P. Vaidyanathan, "Optimized halftoning using dot diffusion and methods for inverse halftoning," IEEE Trans. Image Processing, vol. 9, pp. 691–709, Apr. 2000.
5.   Guo and liu, "Improved Block Truncation Coding using Optimized Dot Diffusion," IEEE Trans. Image Processing (To be published in a forthcoming issue of IEEE).
6.   D. E. Knuth, "Digital halftones by dot diffusion," ACM Trans. Graph., vol. 6, no. 4, Oct. 1987.

BIOGRAPHY

Riyaz Ahmad Dar was born in Srinagar, India, on January 06, 1988. He received the B.E. degree from the Department of Computer Engineering, S.S.M College of Engineering, Srinagar, India, in 2010. He is currently pursuing the M.Tech. degree with the Department of Electronics and Communication Engineering, Srinivas Institute of Technology, Mangalore, India.