

Edge Enhanced Dot Diffusion Block Truncation Coding

Chandiea L
Final year student
ECE Dept, SMVEC
Pondicherry University
Pondicherry, India

LakshmiPriya K
Final year student
ECE Dept, SMVEC
Pondicherry University
Pondicherry, India

Gangadharani N
Final year student
ECE Dept, SMVEC
Pondicherry University
Pondicherry, India

Under the guidance of
Dr L M Varalakshmi
Associate Professor
ECE Dept, SMVEC
Pondicherry University
Pondicherry, India

Abstract: In recent years, the development of multimedia product is rapidly growing which contributes to insufficient bandwidth of network and storage. Thus the theory of image compression gains more importance for reducing the storage space and transmission bandwidth needed. Dot Diffused Block Truncation Coding (DDBTC) is a type of lossy moment preserving quantization method for compressing digital grayscale images. Its advantages are simplicity, relatively high compression efficiency with good image quality, but the major disadvantage is its blocking effect. In this work, an edge enhanced DDBTC technique is proposed that overcomes the blocking effect and also offers an improved image quality. Minimal gradient method for edge enhancement and Thresholding method for image smoothening is used in this work.

Keywords: Block truncation coding, dot diffusion, edge enhanced, smoothening.

I. INTRODUCTION

BTC is a lossy type image compression technique used for compression of grayscale images. It divides the images into blocks and then uses a quantizer to reduce the number of grey levels in each block while it maintains the same mean and standard deviation. The first step of the algorithm is to divide the image into non-overlapping rectangular regions. For a two level (1 bit) quantizer, it selects two luminance values to represent each pixel in the block.

These values are chosen such that the sample mean and standard deviation of the reconstructed block are identical

$$a = m_1 - \sigma \sqrt{\frac{q}{k-q}} \quad (4)$$

$$b = m_1 + \sigma \sqrt{\frac{k-q}{q}} \quad (5)$$

to those of the original block. An $n \times n$ bit map is then used

to determine whether a pixel luminance value is above or below a certain threshold. In order to illustrate how BTC works, we will let the sample mean of the block be the threshold; a "1" would then indicate if an original pixel value is above this threshold, and "0" if it is below [1]. The BTC produces a bitmap to represent a block. By knowing the bit map for each block, the decompression/reconstruction algorithm knows whether a pixel is brighter or darker than the average.

Thus, for each block two gray scale values, a and b are needed to represent the two regions. These are obtained from the sample mean and sample standard deviation of the block, and are stored together with the bit map [4].

To understand how a and b are obtained, let k be the number of pixels of an $N \times N$ block, where $k = N^2$ and x_1, x_2, \dots, x_k be the intensity values of the pixels in a block of the original image. The first two sample moments m_1 and m_2 are given by

$$m_1 = \bar{x} = \frac{1}{k} \sum_{i=1}^k x_i \quad (1)$$

$$m_2 = \frac{1}{k} \sum_{i=1}^k x_i^2 \quad (2)$$

The standard deviation σ is given by

$$\sigma^2 = m_2 - m_1^2 \quad (3)$$

The mean can be selected as the quantizer threshold. Once a threshold, x_{th} , is selected the output levels of the quantizer (a and b) are found such that the first and second moments are preserved in the output. The a and b values are obtained by

In the previous studies, many approaches have attempted to improve BTC. Earlier approaches involves preserving the moment characteristic of the original image.

Halverson et al. [2] generalized a family of moment-preserving quantizers by employing moments higher than three. Udpikar and Raina [5] proposed a modified BTC algorithm, that preserves only the first-order moment. The algorithm is optimum in the mean-square sense, and it is also convenient for hardware implementation. The second set of works tried to improve the image quality and reducing blocking effect.

Kanafani et al. [6] decomposed images into homogeneous and non-homogeneous blocks and then compressed them using BTC or vector quantization (VQ). Block classification was achieved by image segmentation using the expectation maximization (EM) algorithm. The new EM-BTC-VQ algorithm can significantly improve the quality and fidelity of compressed images when compared with BTC or VQ. A new video codec algorithm combined with discrete cosine transform (DCT) was proposed by Horbelt-Crowcroft [7]. The basic concept of this algorithm is that traditional BTC provides excellent performance in high-contrast and detailed regions, while DCT works better for smooth regions. A problem with BTC is its poor image quality under low bit rate conditions. Some studies have attempted to address this issue.

Kamel et al. [8] proposed two modifications of BTC. The first one allows the partitioning of an image into variable block sizes rather than a fixed size. The second modification involves the use of an optimal threshold to quantize the blocks by minimizing the mean square error. Chen and Liu [9] proposed a visual pattern BTC (VPBTC), in which the bitmap is employed to compute the block gradient orientation and match the block pattern. Another refinement is the classification of blocks according to the properties of human visual perception. However, most of the improvements described above increase the complexity substantially.

Recently, some halftoning-based BTC schemes have been developed to effectively improve image quality while minimizing computational complexity. Halftoning [10] is a technique for printing newspapers, books, and magazines with two distinct colors in a color channel. This technique is employed to render the bitmap of a BTC block. When the human visual system (HVS) is involved, the halftoning-based BTC schemes can effectively ease the inherent annoying blocking effect and the false contour artifacts of the traditional BTC. The difference between the halftoning-based BTC and traditional BTC is similar to the halftoning and coarse quantization with a fixed threshold value.

Some famous halftoning-based BTC techniques have been proposed, for instance Guo [11] exploited the error diffusion [12]–[18] to diffuse the quantization errors to the neighboring pixels in the bitmap to maintain local average tone, which is called error-diffused BTC (EDBTC). The adopted error diffusion has an inherent property which can compensate the quantized error of the currently processing position by diffusing the error to its neighboring positions, yet this type of halftoning technique has no parallelism characteristic.

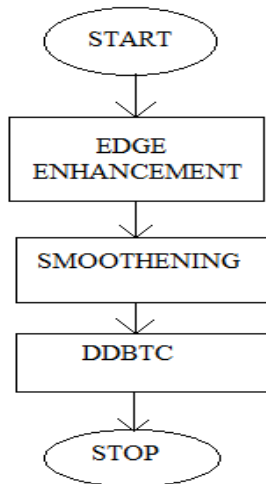
Thus, although good image quality can be achieved by the EDBTC, the prolonged processing time is still an issue. For this, Guo -Wu [19] later proposed the ordered dither BTC (ODBTC) to improve the processing efficiency of the EDBTC by employing look-up-table dither arrays. Yet, the quantization error cannot be compensated with the ordered dithering halftoning [10], and thus the ODBTC yields lower image quality compared to that of the EDBTC. In summary, although both of these halftoning-based methods have apparently coped with the blocking and false contour artifacts of the traditional BTC, there is still room left in terms of the image quality and processing efficiency.

Later Dot Diffused Block Truncation Coding was proposed [2]. Dot diffused BTC employs halftoning method that simulates shades of gray by varying the size of tiny black dot arranged in a regular pattern. The structure of DDBTC algorithm is similar to the traditional BTC algorithm. The DDBTC has two main differences to that of the traditional BTC: 1) The high mean and low mean are replaced by the local maximum and minimum in a block and 2) the manner of bitmap generation is replaced by the dot-diffused halftoning. The original image and the divided block are of sizes $P \times Q$ and $M \times N$, respectively, and each block can be processed independently. For each block, the processing order of pixels is defined by the class matrix. 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. The processing pixel is compared with the threshold value and the error obtained is taken as the diffused weighting. This weighting is diffused to the next processing pixel in the order. 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. But the blocking effect is still a drawback. The work proposed improves the image quality of DDBTC and also overcomes the blocking effect through edge enhancement technique. It also maintains the parallelism advantage of DDBTC.

This paper is organized as follows.

II. MINIMAL GRADIENT METHOD

The basic manipulation tool used is edge extraction from natural images and enhancing it. Natural image editing and high-level structure inference can be done using structured edges. It is very difficult to produce high quality results that are continuous, accurate, and thin due to high susceptibility of edge detectors to complex structures and inevitable noise. A flow chart of the proposed work is shown below.



The minimal gradient method suppresses low-amplitude details, thus remarkably stabilizing the extraction process. It enhance highest-contrast edges by confining the number of non-zero gradients, and smoothing is achieved in a global manner. To begin with, we denote the input discrete signal by s and its smoothed result by d . The method counts amplitude changes discretely, given by

$$U(d) = \# \{p \mid |dp - dp+1| = 0\} \quad (6)$$

Where p and $p+1$ denotes neighboring samples (or pixels). $|dp - dp+1|$ is a gradient w.r.t. p in the terms of forward difference. $\#\{\}$ is the counting operator, outputting the number of p that satisfy $|dp - dp+1| = 0$, $u(d)$ does not count on gradient magnitude, and hence not affected if an edge only alters its contrast. The resulted signal flattens details and sharpens main edges.

$$\min f \sum p (dp - sp)^2 \text{ s.t. } u(d) = c. \quad (7)$$

$u(d)=c$ indicates that c non-zero gradients exist in the result. This equation is abstracts structural information. The overall shape is also in line with the original image because intensity change must arise along significant edges to reduce the total energy as much as possible. This method can remove statistically insignificant details by global optimization. It first apply bilateral filtering, which lowers the amplitudes of noise-like structures more than those of long coherent edges, followed by global sharpening of prominent edges. Result only contains large-scale salient edges, profiting main structure extraction and understanding.

III. EDGE EXTRACTION AND ENHANCEMENT

Pictures contain characters together with background textures. Directly computing gradients on the input image acquires many unwanted small-amplitude structures. The purpose of detecting sharp changes is to capture important details of the image. It has been shown that under rather general assumptions for an image

formation model, discontinuities in image brightness corresponds to

- Discontinuities in depth
- Discontinuities in orientation of surface
- Changes in material properties and
- Variations in scene illumination.

In the ideal case, applying an edge detector to an image may lead to a set of connected curves that indicate the boundaries of objects, surface markings as well as curves that correspond to discontinuities in surface orientation. Thus, the amount of data to be processed is significantly reduced by applying an edge detection algorithm to an image as it filter out information that may be regarded as less relevant, meanwhile preserving the important structural properties of an image. If edge detection is successful, then interpreting the information contents in the original image may be simplified. In most of the case, it is not always possible to obtain such ideal edges from real life images of moderate complexity. The filters are employed in the process of identifying the image by locating the sharp edges which are discontinuous. These discontinuities changes the pixels intensities which define the boundaries of the object. Steps for edge extraction and enhancement are as follows

- (1) Localization: determine the exact location of an edge.
- (2) Detection: determine which edge pixels should be discarded as noise and which should be retained.
- (3) Enhancement: apply a filter to enhance the quality of the edges in the image
- (4) Smoothing: suppress as much noise as possible, without destroying the true edges.

IV. THRESHOLDING

The purpose of thresholding is to extract those pixels from the image which represent an object. The pixels represent a range of intensities even though they are binary. And the objective of binarization is to mark pixels that belong to true foreground regions with a single intensity and background regions with different intensities.

3.2.1 Thresholding algorithm

For a thresholding algorithm to be effective, it should preserve logical content. There are two types of thresholding algorithm

1. Global thresholding algorithm
2. Local adaptive thresholding algorithm

In global thresholding, a single threshold is used for all the pixels in an image. When the pixel values of the components and that of background are fairly consistent in their respective values over the entire image then global

thresholding could be used. In local thresholding, different threshold values for different local areas are used. The work in this paper uses global threshold. The steps involved in thresholding are given below.

- 1) Select an initial T
- 2) Compute the average intensity m_1 and m_2 for the pixels above and below T.
- 3) Compute a new threshold:

$$T = (m_1 + m_2) / 2$$

- 4) Continue the process until the difference between T values of consecutive iteration is less.
- 5) When the final threshold is arrived, pixels less than that are assigned '0' and pixels greater than that are assigned '1'.

3. Simulation results

The simulation tool used is MATLAB R2012a. The comparison between the existing DDBTC and edge enhanced DDBTC in terms of MSE, PSNR, SSIM, compression ratio is shown below. The input image used is of size 480 600. It is JPEG type image. The figure 4.1 shows the input image, figure 4.2 represents the existing DDBTC result. Fig 4.3 and 4.4 shows the result of the proposed work.

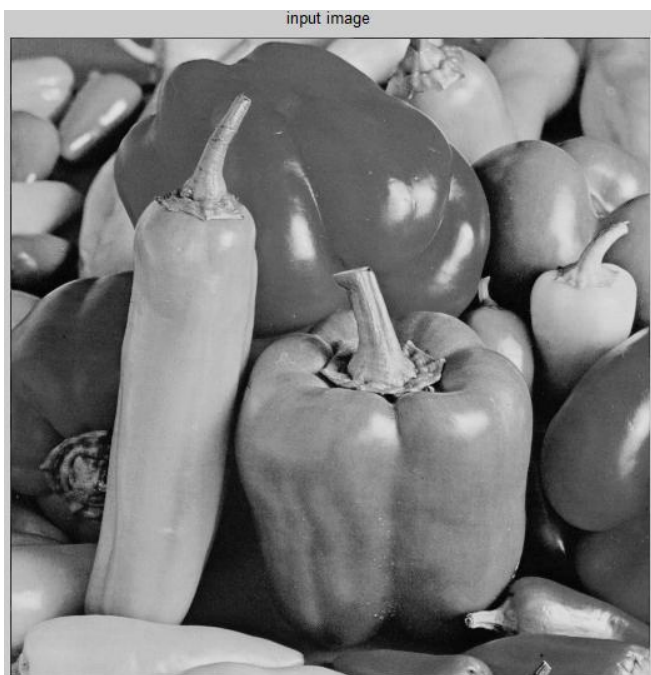


Fig 4.1 Input image

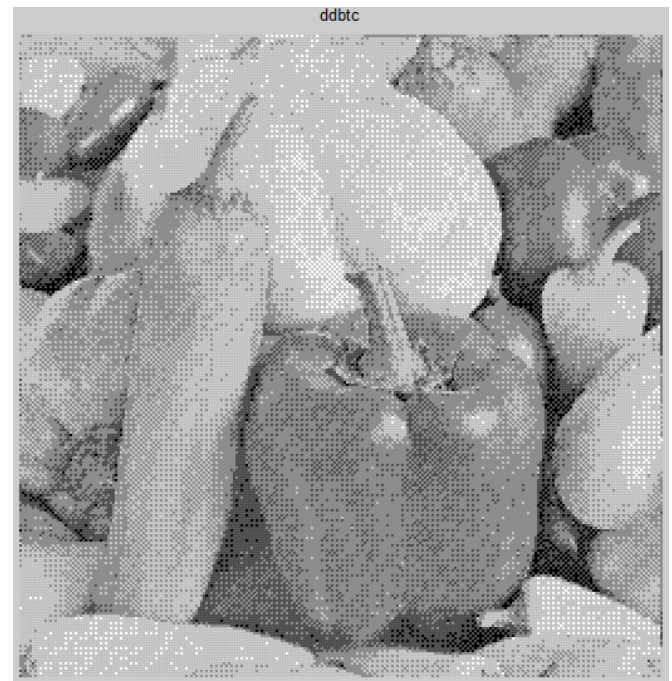


Fig 4.2 DDBTC

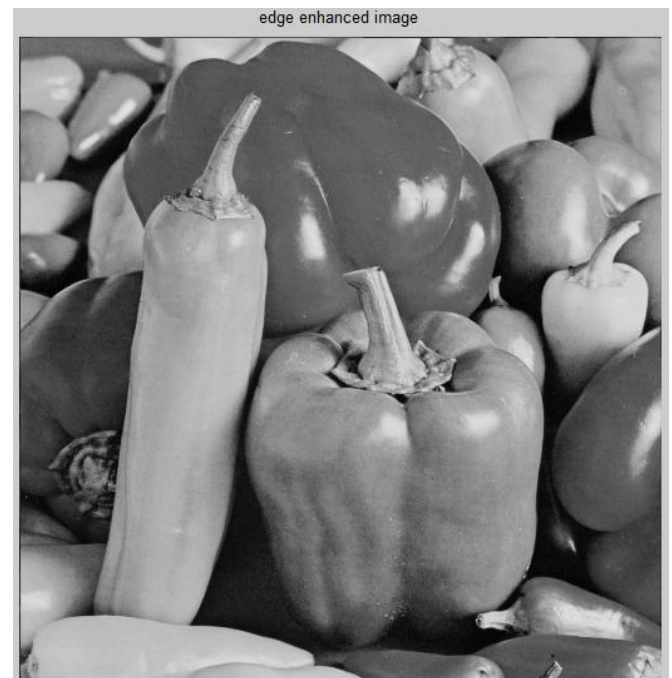


Fig 4.3 Edge enhanced input image

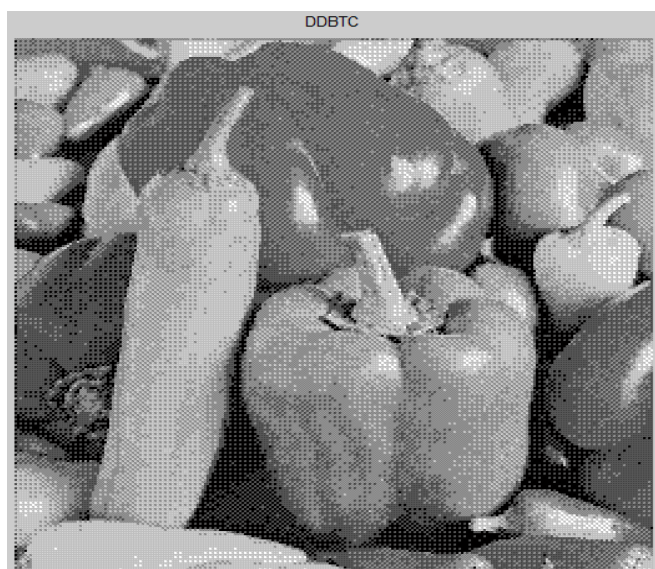


Fig 4.4 Edge enhanced DDBTC

4 PARAMETER COMPARISON

TABLE 4.1

Parameter	DDBTC	Edge Enhanced DDBTC
MSE	4915.4	1060.2
PSNR	11.58	17.8770
SSIM	0.6279	0.8395
Compression ratio	2.8806	2.9963

The parameters of DDBTC and Edge enhanced DDBTC are compared in table 4.1.

5. CONCLUSION

To conclude, the edge enhancement and smoothening of the image has been proposed to improve the quality of the image obtained using dot diffused block truncation coding. The minimal gradient method performs edge enhancement over the original image and thresholding performs the smoothening Operation which results in higher quality of the image. The DDBTC then performed gives higher quality than obtained by the existing method. Edge enhancement enhances the edge contrast of an image in an attempt to improve its acutance. In the thresholding method a global value is obtained and it is compared with the processing pixel and the smoothening operation is carried out. A drawback of edge enhancement is that the image may begin to look less natural, because the apparent acutance of the overall image has increased but the level of detail in flat, smooth areas has not. And in thresholding a single threshold will not work well when we have uneven illumination due to shadows or due to the direction of illumination.

6. REFERENCE

- (1) E. Delp and O. R. Mitchell, "Image compression using block truncation coding"; *IEEE Trans. Commun.*; vol. 27; No.9 :pg-1335-1342; sep 2013.
- (2) J. Guo and Y. Liu, "Improved Block Truncation Coding using Optimized Dot Diffusion"; *IEEE Trans. on image processing*; vol.23; NO. 3; March 2014.
- (3) 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. 2011.
- (4) Li Xu, Cewu Lu, Yi Xu, Jiaya Jia, "Image Smoothing via L_0 Gradient minimization"; *ACM Transactions on Graphics*; Vol. 30; No. 5 (*SIGGRAPH Asia 2011*); Dec 2011.
- (5) 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.
- (6) 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.
- (7) 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.
- (8) 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.
- (9) 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.
- (10) 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.
- (11) R. Ulichney, "Digital Halftoning" Cambridge, MA, USA: MIT Press, 1987.
- (12) J. M. Guo, "Improved block truncation coding using modified error diffusion," *Electron. Lett.*, vol. 44, no. 7, pp. 462–464, Mar. 2008.
- (13) R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial gray scale," in *Int. Symp. Soc. Inf. Display, Dig.*, 1975, pp. 36–37.
- (14) J. F. Jarvis, C. N. Judice, and W. H. Ninke, "A survey of techniques for the display of continuous-tone pictures on bilevel displays," in *Proc. Comp. Graph. Image Process.*, vol. 5. 1976, pp. 13–40.
- (15) P. Stucki, "MECCA: A multiple-error correcting computation algorithm for bilevel image hardcopy reproduction," IBM Res. Lab., Zurich, Switzerland, Res. Rep. RZ1060, 1981.
- (16) J. N. Shiau and Z. Fan, "A set of easily implementable coefficients in error diffusion with reduced worm artifacts," *Proc. SPIE*, vol. 2658, pp. 222–225, Jan. 1996.
- (17) V. Ostromoukhov, "A simple and efficient error-diffusion algorithm," in *Proc. 28th Annu. Conf. Comput. Graph.*, 2001, pp. 567–572.
- (18) P. Li and J. P. Allebach, "Tone-dependent error diffusion," *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 201–215, Feb. 2004.
- (19) P. Li and J. P. Allebach, "Block interleaved pinwheel error diffusion," *J. Electron. Imag.*, vol. 14, no. 2, p. 023007, Apr.–Jun. 2005.
- (20) J. M. Guo and M. F. Wu, "Improved block truncation coding based on the void-and-cluster dithering approach," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 211–213, Jan. 2009