Journal of Computers Vol. 30 No. 1, 2019, pp. 15-22
doi:10.3966/199115992019023001002

# Visual Quality Improvement for Ordered Dither Block Truncation Coding

Chyi-Hwa Chu*

Center for Educational Resources and Publishing, National Academy for Educational Research,
New Taipei City 237, Taiwan, ROC
chu@mail.naer.edu.tw

Received 9 January 2018; Revised 30 October 2018; Accepted 30 October 2018

Abstract. Ordered dither block truncation coding (ODBTC) employs the ordered dither technique to produce the bitmap required in traditional block truncation coding (BTC). It can reduce the complexity of computations in the coding phrase. However, the blocking artifacts and undesired noises are still in the decoded images by ODBTC. In this paper, the properties of thresholds stored in the dither matrices are considered to estimate the decoded pixel intensities. That is, the dither matrices are used in both coding and decoding. Experimental results tell that the proposed schema can not only reduce quantization error but also improve the visual quality for the reconstructed images by ODBTC.

Keywords: block truncation coding, digital halftoning, image compression, ordered dither block truncation coding, ordered dither technique

## 1 Introduction

Block truncation coding (BTC) is a simple and efficient technique for image compression [1]. An image is first partitioned into equal-sized block and each image block is processed separately. Duration the encoding phase, a two-level quantization is performed. Pixels with values less than the quantization threshold are set to “1” (white) and the others are set to “0” (black). That is, each image block is converted to a bitmap B. Then two quantization levels, a and b, are calculated to maintain the desired properties of the image blocks. At last, a triple (B, a, b) is obtained for each image block. The decompressed image can be easily constructed by replacing the value “0” of the bitmap with a and the value “1” of the bitmap with b. The original BTC introduced by Delp and Mitchell uses the mean value as a threshold in each image block and finds the quantization levels such that the first and second moments are preserved. Many variants of BTC were proposed to improve the performance of the compression and decompression [2]. For example, absolute moment BTC (AMBTC) calculates the quantization levels to preserve the first absolute central moment [3]. Lloyd quantization [4] and neural network [5] were also employed to improve the coding performance of BTC.

Recently, digital halftoning are introduced to enhance the efficiency of coding for BTC [6-10]. Digital halftoning is a technique to display the gray images with only black and white colors [11]. Hence, it can be used to produce the bitmap required in BTC. Error diffusion [12] produces the binary image by compares the sum of pixel value and the accumulated quantization error to a fixed threshold to determine the output value. Then the quantization error is distributed to the unprocessed pixels. Error-diffused BTC (EBTC) introduces the schema of quantization error distribution to find the bitmap and uses the maximum and minimum values of each image block as the quantization levels [6]. In such a way, the complexity of calculating for high and low means in the original BTC can be greatly reduced. Ordered dither BTC (ODBTC) uses the predefined dither matrix to generate the bitmap [7]. Only one comparison is needed for each pixel. Both of low complexity and better visual quality are introduced by ODBTC. Dot diffusion and other modifications of digital halftoning techniques have also been proposed to enhance

* Corresponding Author

BTC [9-10]. These techniques of halftoning-based BTC (HBTC) treat the halftoning technique as a binary quantizer to find the bitmap during the coding phase. However, the properties of the halftoning technique are not applied to decompress the images. For example, dither matrix contains the thresholds used for finding the bitmap. Thus, the upper bound and lower bound of each pixel in decompressed image can be determined. The visual quality of the decoded images by ODBTC can be further improved with the properties of dither matrices.

In this paper, we take the advantages of dither patterns generated by ordered dither technique to decrease the quantization error and improve the visual quality for ODBTC. That is, the schema of order dithering can not only be applied in compression but also in decompression.

## 2 Proposed Method

### 2.1 Ordered Dither Technique

A dither matrix contains a set of thresholds. Ordered dither technique [13] compares the pixel intensity with the threshold number stored in the dither matrix to determine the output value. Repeating the dither matrix over the image throughput, the output halftoning image is obtained.

Let $x_{i,j}$ be the $(i, j)$-th pixel intensity in the original grayscale image $x$ and $y_{i,j}$ be the $(i, j)$-th pixel intensity in the corresponding output image $y$. By ordered dither technique, the output $y_{i,j}$ can be determined as

$$y_{i,j} = \begin{cases} 1 & \text{if } x_{i,j} \geq d_{i \bmod M,\, j \bmod N} \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where $d_{i,j}$ is the $(i, j)$-th element in the dither matrix $d$ having size of $M$ by $N$ and $0 \leq i \leq M$-1, $0 \leq j \leq N$-1.

It is noted that only one dither matrix is needed for the original schema of the ordered dither technique. From the location of the pixel, we can find the corresponding threshold number. The output of a pixel is dependent on its intensity and location only. Therefore, each pixel can be processed independently.

### 2.2 Ordered Dither Block Truncation Coding

ODBTC first partition the original image into non-overlapped blocks of size $M \times N$. That is, the image block has the same size of the dither matrix. Suppose the maximum and minimum value of the image block are denoted as $x_{\max}$ and $x_{\min}$. The bitmap $b$ can be computed by

$$b_{i,j} = \begin{cases} 1 & \text{if } x_{i,j} \geq d^{k}_{i \bmod M,\, j \bmod N} + x_{min} \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where $b_{i,j}$ is the $(i, j)$-th element in the bitmap $b$, $k = x_{\max} - x_{\min}$, and $d^k$ is the scaling version of the dither matrix $d$ by $k$. It can be obtained by

$$d^{k}_{m,n} = k \times \frac{d_{m,n} - d_{min}}{d_{max} - d_{min}}, \tag{3}$$

where $1 \leq k \leq 255$, $0 \leq m \leq M$-1, and $0 \leq n \leq N$-1; $d_{\max}$ and $d_{\min}$ denote the maximum and minimum values in the dither matrix. At last, we have a binary image $b$ and two quantization levels ($x_{\min}$, $x_{\max}$) for each image block. The reconstructed image can be obtained by replacing black pixels with $x_{\min}$ and white pixels with $x_{\max}$ for each image block.

From the algorithm of ODBTC, we can find that the dither matrix needed for thresholding is dependent on the difference of the maximum and minimum values in an image block. That is, each image block needs one dither matrix and more than one dither matrices are needed for a whole image.

### 2.3 Our Method

Due to the block-based nature, blocking artifacts are commonly found in the decompressed image by BTC and ODBTC. In this paper, we employ the concept of reconstruction of the compressed image by ordered dither technique [14] to decrease the quantization error and enhance the visual quality for ODBTC.

Let $y$ be the decoded image of the original image $x$ by ODBTC. Now we attempt to find the output image $z$ from image $y$ such that image $x$ and image $z$ are close as possible. In ODBTC, the element in a dither matrix is treated as a threshold to find the bitmap. From the quantization levels stored in each image block, we can find the dither matrix used in thresholding for each image block. That is, we can find the threshold value $d_{i,j}$ at location $(i, j)$. In such a way, the upper bound and lower bound of the output pixel intensity can be obtained by

$$u(z_{i,j}) = \begin{cases} x_{max} & if\ y_{i,j} = x_{max} \\ d_{i,j} & if\ y_{i,j} = x_{min} \end{cases} \quad \textbf{(4)}$$

and

$$l(z_{i,j}) = \begin{cases} d_{i,j} & if\ y_{i,j} = x_{max} \\ x_{min} & if\ y_{i,j} = x_{min} \end{cases}, \quad \textbf{(5)}$$

where $u(z_{i,j})$ and $l(z_{i,j})$ are the upper bound and lower bound of $z_{i,j}$, respectively. In addition, $x_{\max}$ and $x_{\min}$ are the maximum and minimum value of the image block which contains location $(i, j)$. It is noted that the quantization levels $x_{\max}$ and $x_{\min}$ for each image block are coded in image $y$, the upper bound and lower bound of each element $z_{i,j}$ can be obtained from image $y$ instead of $x$.

Next, the neighbors of $z_{i,j}$ in a 4×4 windows are considered to estimate the value of $z_{i,j}$. Element $z_{i,j}$ and its 15 neighbors are shown in Fig. 1. It is noted that $z_{i,j}$ and its neighbors may not be contained in the same image block during the encoding phase. They may have different upper bounds or lower bounds. Let $L$ denote the maximum value of the 16 lower bounds and $U$ denote the minimum value of the 16 upper bounds. That is,

| | | | |
|---|---|---|---|
| $z_{i-2,j-2}$ | $z_{i-1,j-2}$ | $z_{i,j-2}$ | $z_{i+1,j-2}$ |
| $z_{i-2,j-1}$ | $z_{i-1,j-1}$ | $z_{i,j-1}$ | $z_{i+1,j-1}$ |
| $z_{i-2,j}$ | $z_{i-1,j}$ | $z_{i,j}$ | $z_{i+1,j}$ |
| $z_{i-2,j+1}$ | $z_{i-1,j+1}$ | $z_{i,j+1}$ | $z_{i+1,j+1}$ |

**Fig. 1.** Current pixel $z_{i,j}$ and its 15 neighbors in a 4x4 window

$$L = \max_{-2 \le m,n, \le 1} \{l(z_{i+m,j+n})\} \quad \textbf{(6)}$$

and

$$U = \min_{-2 \le m,n, \le 1} \{u(z_{i+m,j+n})\}.. \quad \textbf{(7)}$$

If $U > L$, we set the output $z_{i,j} = (L+U) / 2$. Otherwise, $z_{i,j}$ can be calculated by

$$z_{i,j} = \begin{cases} l(z_{i,j}) & if\ (z_{i,j}) > g \\ g & if\ (z_{i,j}) \le g \le u(z_{i,j}) \\ u(z_{i,j}) & if\ g > u(z_{i,j}) \end{cases}, \quad \textbf{(8)}$$

where

$$g = \frac{1}{18} \Sigma_{-1 \le m,n \le 1} \{l(z_{i+m,j+n}) + u(z_{i+m,j+n})\}. \quad \textbf{(9)}$$

## 3 Experimental Results

Twelve 256-level images are involved in our simulations. They have the same size of 512×512 and can be retrieved from web [15]. The reduced versions of test images are shown in Fig. 2. Four dither matrices size of 2×2, 4×4, 8×8, and 16×16 are used in ODBTC and our proposed schema. Fig. 3 gives the matrices of varying sizes from 2×2 to 8×8. The dither matrix size of 16×16 is not shown here due to the length limits. The matrices of larger sizes can be derived from the smaller ones [14].

**Fig. 2.** Test images

| | |
|---|---|
| 32 | 160 |
| 224 | 96 |

(a) 2×2

| | | | |
|---|---|---|---|
| 8 | 136 | 40 | 168 |
| 200 | 72 | 232 | 104 |
| 56 | 184 | 24 | 152 |
| 248 | 120 | 216 | 88 |

(b) 4×4

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 130 | 34 | 162 | 10 | 138 | 42 | 170 |
| 194 | 66 | 226 | 98 | 202 | 74 | 234 | 106 |
| 50 | 178 | 18 | 146 | 58 | 186 | 26 | 154 |
| 242 | 114 | 210 | 82 | 250 | 122 | 218 | 90 |
| 14 | 142 | 46 | 174 | 6 | 134 | 38 | 166 |
| 206 | 78 | 238 | 110 | 198 | 70 | 230 | 102 |
| 62 | 190 | 30 | 158 | 54 | 182 | 22 | 150 |
| 254 | 126 | 222 | 94 | 246 | 118 | 214 | 86 |

(c) 8×8

**Fig. 3.** Dither matrices

ODBTC and our schema are simulated with image blocks having the same sizes of dither matrices. In order to evaluate the performance of ODBTC and our method, Mean Square Error (MSE), Mean Absolute Error (MAE), Peak Signal to Noise Ratio (PSNR), and Structure Similarity (SSIM) are used to measure the quality of the reconstructed images. MAE, MSE, and PSNR of the original image $x$ and the reconstructed image *z* are calculated as follows.

$$MAE = \frac{1}{512 \times 512} \Sigma_{0 \le 1, j < 512} \, | x_{i,j} - z_{i,j} |, \tag{10}$$

$$MSE = \frac{1}{512 \times 512} \Sigma_{0 \le 1, j < 512} [x_{i,j} - z_{i,j}]^2, \tag{11}$$

$$PSNR = 10 \log_{10} MSE. \tag{12}$$

SSIM for two signals *x* and *z* can be calculated by

$$SSIM = \frac{(2\mu_X \mu_z + C_1)(2\sigma_{xz} + C_2)}{(\mu_x^2 + \mu_z^2 + C_1)(\sigma_x^2 + \sigma_z^2 + C_2)}, \tag{13}$$

where μ means the average value and σ is the variance, $C_1$ and $C_2$ are constants. The more SSIM approaches one, the better the reconstruction quality. The details of computations for SSIM can be found in [16].

The averages of the quality measurements of the test images with different dither matrices by ODBTC and our method are plotted in Fig. 4 to Fig. 7. Solid lines with square marks represent our method, while dashed lines with circle marks represent ODBTC. From the experimental results, we can find that our method has better results than those by ODBTC in MAE, MSE and PSNR. For SSIM, the difference between the reconstructed images by ODBTC and our method is very close for block size of 2×2, 4×4 and 8×8. It is apparent that our method has better results than those by ODBTC for block size of 16×16 in SSIM. Furthermore, these curves tell that the larger the dither matrix is, the more it becomes that our proposed schema outperforms ODBTC.

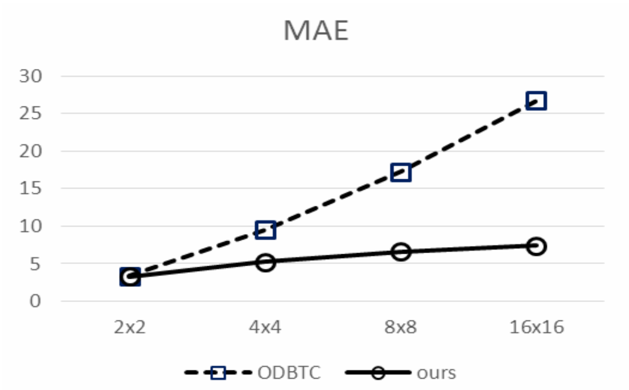


**Fig. 4.** MAE for ODBTC and our schema

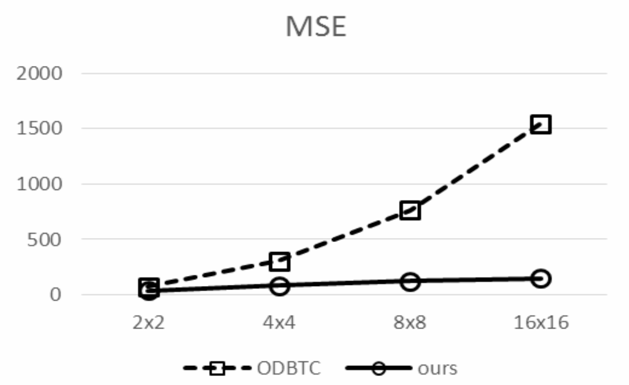


**Fig. 5.** MSE for ODBTC and our schema

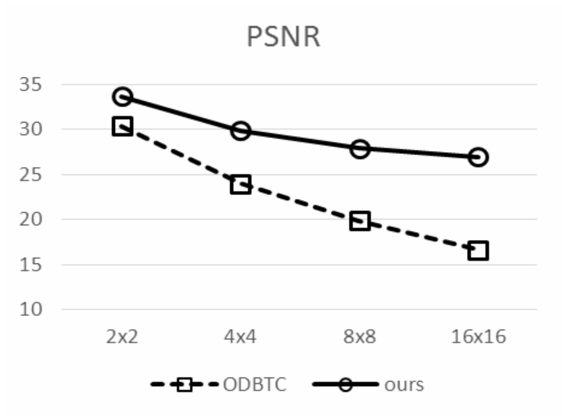


**Fig. 6.** PSNR for ODBTC and our schema

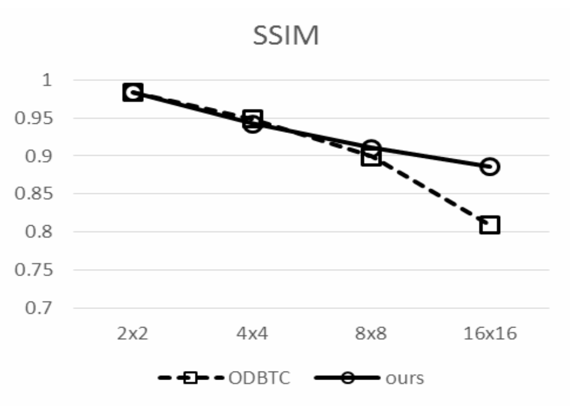


**Fig. 7.** SSIM for ODBTC and our shema

In visual quality test, Fig. 8 and Fig. 9 show the coding results for image LENA and PEPPERS, respectively. Portions of the results by ODBTC with 16×16 dither matrices are shown in Fig. 5(a) and Fig. 6(a). Our results with the same dither matrices are shown in Fig. 5(b) and Fig. 6(b). As shown in Fig. 5 to Fig. 6, blocking artifacts and undesired noises are noticeable in the reconstructed image by ODBTC than those by our method. The same results appear in other ten test images. Due to the paper length limitation, only the reconstructed versions of test image LENA and PEPPERS are shown in this paper.

(a) ODBTC (b) ours

**Fig. 8.** Portitions of the reconstructed images LENA

## 4 Discussions & Conclussions

ODBTC introduced the adaptive dither matrices to find the bitmap for traditional BTC. It can reduce the complexity of computations in coding phrase. However, the properties of the thresholds stored in the dither matrices are not considered in decoding phrase. In this paper, we employ the main concept of reconstruction for dithered images to decompress the coded images by ODBTC.

According to the simulation results, our method can improve the visual quality of encoded images by ODBTC. It is expected that since we use not only the maximum and minimum values of an image block but also the elements of dither matrices to estimate the pixel intensities. The larger the dither matrix is, the more it becomes that our proposed schema outperforms ODBTC.

Variants of digital halftoning have been proposed for many applications, for example, multilevel image rendering, image compression and image security. Each digital halftoning technique has its own characteristics. In this paper, we take the advantage of the ordered dither technique to improve the visual quality of the coding images by ODBTC. The halftoning-based block truncation coding may have the potential to improve its performance by the properties of the halftoning methods.

## References

[1] E.J. Delp, O.R. Mitchellm, The use of block truncation coding in DPCM image coding, IEEE Trans, Signal Process 39(1991) 967-971.

[2] P. Franti, O. Nevalainen, T. Kaukoranta, Compression of digital images by block truncation coding: a survey, The Computer Journal 37(4)(1994) 308-332.

[3] M. Lema, O. Mitchell, Absolute moment block truncation coding and its application to color images, IEEE Transactions on Communications 32(10)(1984) 1148-1157.

[4] W.W. Lu, M.P. Gough, P.N.H. Davies, Scientific data compression for space: a modified block truncation coding algorithm, SPIE 1470(1991) 197-205.

[5] G. Qiu, M.R. Varley, T.J. Terrell, Improved block truncation coding using Hopfield neural network, Electronics Letters 27(21)(1991) 1924-1926.

[6] J.M. Kuo, Improved block truncation coding using modified error diffusion, Electronics Letters 44(2008) 462-464.

[7] J.M. Kuo, M.F. Wu, Improved block truncation coding based on the void-and-cluster dithering approach, IEEE Trans. On Image Processing 44(2009) 211-213.

[8] J.M. Kuo, High efficiency ordered dither block truncation coding with dither array LUT and its scalable coding application, Digital Signal Processing 20(2010) 97-110.

[9] J.M. Guo, Y.F. Liu, Improved block truncation coding using optimized dot diffusion, IEEE Trans. on Image Processing 23(2014) 1269-1275.

[10] J.M. Guo, C.Y. Lin, Parallel and element-reduced error-diffused block truncation coding, IEEE Trans. on Communications 58(5)(2010) 1667-1673.

[11] J. Allebach, T. Pappas, Digital halftoning, IEEE Signal Processing Magazine 20(2003) 12-13.

[12] R.W. Floyd, L. Steinberg, An adaptive algorithm for spatial grayscale, Proc. SID 17(2)(1976) 75-77.

[13] B.E. Bayer, An optimum method for two-level rendering of continuous-tone pictures, in Proc. IEEE International Conference on in Communications 1973.

[14] P. Pirsch, A.N. Netravali, Transmission of gray level image by multilevel dither techniques, Computers & Graphics 7(1)(1983) 31-44.

[15] Image Databases. <http://imageprocessingplace.com/root_files_V3/image_databases.htm>, 2017 (accessed 20.10.2017).

[16] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error measurement to structural similarity, IEEE Trans. Image Processing 13(2004) 600-612.