Improved Block Truncation Coding Using Optimized Dot Diffusion

Jing-Ming Guo, *Senior Member, IEEE*, and Yun-Fu Liu, *Member, IEEE*

***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.

In the previous studies, many approaches have endeavored to improve BTC. The first category of such 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 [3] proposed a modified BTC algorithm, which 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 category involves improving image quality and reducing blocking effect. Kanafani *et al*. [4] 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 [5]. 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*. [6] 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 [7] 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 [8] is a technique for

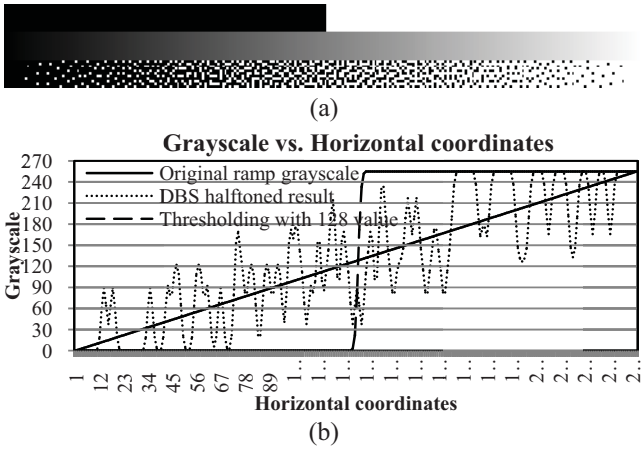


Fig. 1. Comparison of (a) ramp pattern representation, where the first row and third row are obtained by coarse quantization with fixed 128 threshold value and DBS halftoning [9] (printed at 75 dpi) and (b) corresponding HVS simulation results.

Manuscript received September 9, 2012; revised April 2, 2013 and December 17, 2012; accepted April 2, 2013. Date of publication April 12, 2013; date of current version February 4, 2014. This work was supported by the National Science Council, Taiwan, under Contract NSC 101-2221-E-011-136-MY3. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Joan Serra-Sagrista.

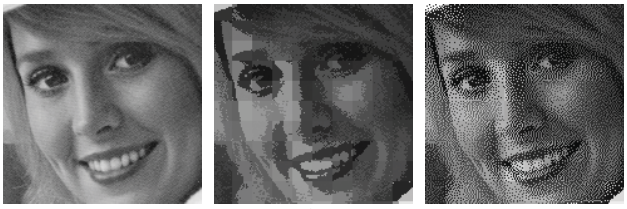
The authors are with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan (e-mail: jmguo@seed.net.tw; yunfuliu@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2013.2257812

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. For instance, Fig. 1(a) shows two types of ramp representation by thresholding with a fixed value 128 (top) and by the DBS halftoning [9] (bottom). It is clear that the ramp structure is completely eliminated when coarse quantization is applied, which is similar to the mechanism of the traditional BTC. In contrast, the halftoning has a property that it can control the dot density to cope with the desired grayscale. To provide another perspective, the mid-rows of the three different patterns, top, middle, and bottom, as shown in Fig. 1(a) are re-drawn in histogram fashion as shown in Fig. 1(b), in which the distributions are smoothed by a 1D Gaussian filter of size nine and standard deviation of 1.3 for simulating the low-pass nature of the HVS. Once again the halftone result can yield better grayscale representation capability than that of the coarse binarization with a fixed threshold which is used in traditional BTC. Moreover, the high frequency texture of the halftone histogram as shown in Fig. 1(a) is inversely proportional to viewing distance, which means that when viewing from a distance, the dithered pattern can approximate the original continuous-tone pattern. In another case, when the pixel/inch (PPI) is increased, the halftone pattern can approximate the original image as well when the viewing distance is fixed, and this is the case for the evolution of modern displayers. Thus, it is not hard to imagine that the halftoning-based BTC can render better image quality than that of the traditional BTC. Fig. 2 shows a compressed result comparison between the traditional BTC and the halftoning-based BTC, and it is obvious that those artifacts are significantly eased by the involvement of the halftoning.

Some famous halftoning-based BTC techniques have been proposed, for instance Guo [10] exploited the error diffusion [11]–[17] 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 [18] 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 [8], 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.



(a) (b) (c)

Fig. 2. Cropped Elaine BTC results of size 159×159. (a) Original grayscale. (b) Traditional BTC [1]. (c) EDBTC [10] with Jarvis *et al.*'*s* error kernel [12]. (all printed at 150 dpi).

In this paper, the dot diffusion, i.e., [19] or [20], is employed to cooperate with BTC to yield the proposed dot-diffused BTC (DDBTC) image compression technique. Notably, the diffused matrix and class matrix are co-optimized to obtain even better image quality than that of the EDBTC which has been proved of achieving the best image quality so far, while maintaining the parallelism similar to that of ODBTC. These two properties endorse the proposed DDBTC for efficient compression applications and low power systems, e.g., low-end embedded systems or handheld devices for image/video recording. One peculiar example recently we conducted is to embed the proposed scheme into a global positioning system (GPS) device which is used on a car. The low complexity advantage enables the additional coding feature using the rest CPU resource of the GPS device. Thus, the proposed method can be considered as a powerful candidate for use in most of the low power image/video codec system. Comparing with the former work [21], more theoretical analyses/explanations and more comparisons with novel image quality assessment (IQA) are provided for supporting the effectiveness of the proposed scheme.

The rest of this paper is organized as follows. Section II introduces the traditional BTC, and Section III describes the proposed DDBTC technique in detail. Experimental results are summarized in Section IV, and conclusions are drawn in Section V.

II. Overview of the Traditional Block Truncation Coding

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 $P \times Q$, and which is divided into many non-overlapped blocks of size $M \times N$, 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

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

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

$$\sigma^2 = \overline{x^2} - (\overline{x})^2, \tag{3}$$

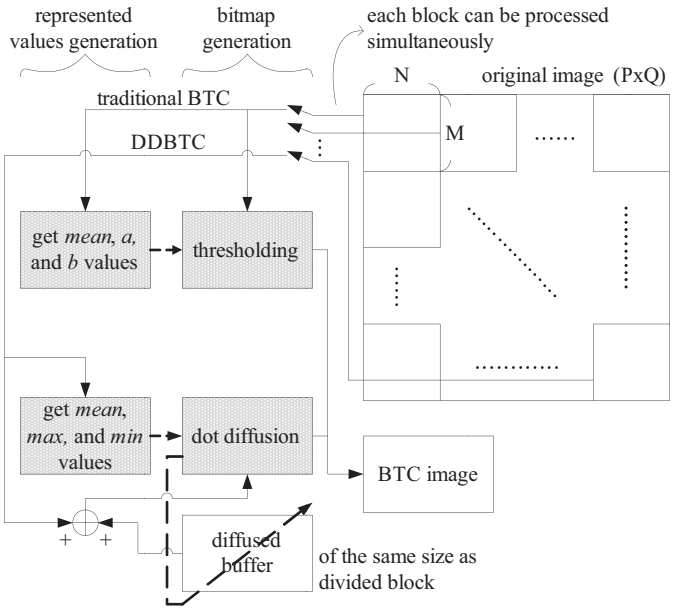


Fig. 3. Algorithm comparison between the traditional block Truncation Coding (BTC) and the proposed Dot-Diffused BTC (DDBTC).

where the variable $x_{i,j}$ denotes the grayscale pixel value in a block. Since BTC is a one-bit quantizer with a threshold $\overline{x}$ to binarize the block, the block is then replaced by bitmap as defined below

$$h_{i,j} = \begin{cases} 1, & \text{if } x_{i,j} \geq \overline{x} \\ 0, & \text{if } x_{i,j} < \overline{x}' \end{cases} \tag{4}$$

and the reconstructed result is obtained as

$$y_{i,j} = \begin{cases} b, & \text{if } h_{i,j} = 1 \\ a, & \text{if } h_{i,j} = 0' \end{cases} \tag{5}$$

where the variable $h_{i,j}$ 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.

$$m\overline{h} = (m-q)\,a + qb, \tag{6}$$

$$m\overline{h^2} = (m-q)\,a^2 + qb^2, \tag{7}$$

where $m = M \times N$, and q denotes the number of pixels greater than $\overline{x}$. The high and low means can be evaluated as follows

$$a = \overline{x} - \sigma\sqrt{\frac{q}{m-q}}, \tag{8}$$

$$b = \overline{x} + \sigma\sqrt{\frac{m-q}{q}}. \tag{9}$$

III. Dot-Diffused Block Truncation Coding

The structure of the proposed DDBTC algorithm is similar to the traditional BTC algorithm, in which the detail comparisons between the two methods are shown in Fig. 3. The variable *mean* can be calculated by (1), and variables a and b are obtained by (8) and (9), respectively. Notably, the proposed DDBTC totally complies with the flow of the traditional BTC,

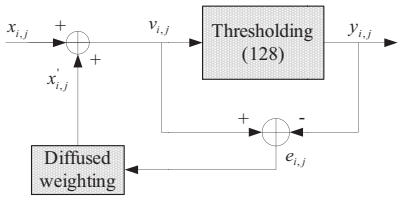


Fig. 4. Dot diffusion algorithm.

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.

$$x_{\max} = \max(B), \tag{10}$$

$$x_{\min} = \min(B), \tag{11}$$

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 ($x_{\max}$) and minimum ($x_{\min}$) in a block, because the high dynamic range ($x_{\max} - x_{\min}$) can easily destroy the blocking effect and false contour, and 2) the manner of bitmap generation is replaced by the dot-diffused halftoning as detailed below.

Suppose 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 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. Fig. 4 shows the processing concept, and the corresponding equations are given below.

$$v_{i,j} = x_{i,j} + x'_{i,j}, \quad \text{where } x'_{i,j} = \sum_{(m,n)\in R} \frac{e_{i+m,j+n} \times k_{m,n}}{sum}, \tag{12}$$

$$e_{i,j} = v_{i,j} - y_{i,j}, \quad \text{where } y_{i,j} = \begin{cases} 0, & \text{if } v_{i,j} < 128 \\ 255, & \text{if } v_{i,j} \geq 128 \end{cases}, \tag{13}$$

where the variable $x_{i,j}$ denotes the current input grayscale value, variable $x'_{i,j}$ denotes the diffused error accumulated from neighboring processed pixels, and variable $v_{i,j}$ denotes the modified grayscale output. 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 of diffused weighting, with a suggested size of 3×3 as in Knuth's [19] and Mese-Vaidyanathan's dot diffusion [20]. The diffused weighting can be represented as

$$\begin{bmatrix} k_{-1,-1} & k_{-1,0} & k_{-1,1} \\ k_{0,-1} & x & k_{0,1} \\ k_{1,-1} & k_{1,0} & k_{1,1} \end{bmatrix}. \tag{14}$$

The variable x denotes the pixel currently being processed. Notably, the error can only diffuse to neighboring pixels that

TABLE I

TWO TRAINED CLASS MATRICES WITH DIFFERENT SIZES, 8×8 AND 16×16, AND THE CORRESPONDING DIFFUSED MATRICES

(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

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.

$$sum = \sum_{m=-1}^{1} \sum_{n=-1}^{1} \begin{cases} k_{m,n}, & \text{if } c_{i+m,j+n} > c_{i,j} \\ 0, & \text{if } c_{i+m,j+n} < c_{i,j}, \end{cases} \tag{15}$$

where the variable $c_{i,j}$ denotes the coefficient value in the class matrix. Fig. 5 shows an example which includes four independent blocks split by the thick lines. Herein, 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 $sum = k_{-1,-1} + k_{-1,0} + k_{0,-1} + k_{1,1}$ in this case. Notably, the error not only can diffuse to the self block, but also can diffuse to its neighboring blocks.

Since the dot diffusion in DDBTC is different from the traditional one, such as the pixel depth of the represented

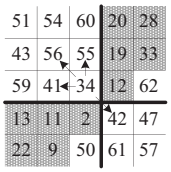


Fig. 5. Demonstrated example of Diffusion between blocks using the class matrix in Table I(a).

values, the threshold and the two replaced values in (13) are modified as below

$$e_{i,j} = v_{i,j} - y_{i,j}, \quad \text{where } y_{i,j} = \begin{cases} x_{\min}, & \text{if } v_{i,j} < \overline{x} \\ x_{\max}, & \text{if } v_{i,j} \geq \overline{x}, \end{cases} \tag{16}$$

where the variables $\overline{x}$, $x_{\min}$, and $x_{\max}$ are defined in (1), (10), and (11), respectively. 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 [19], [20] 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

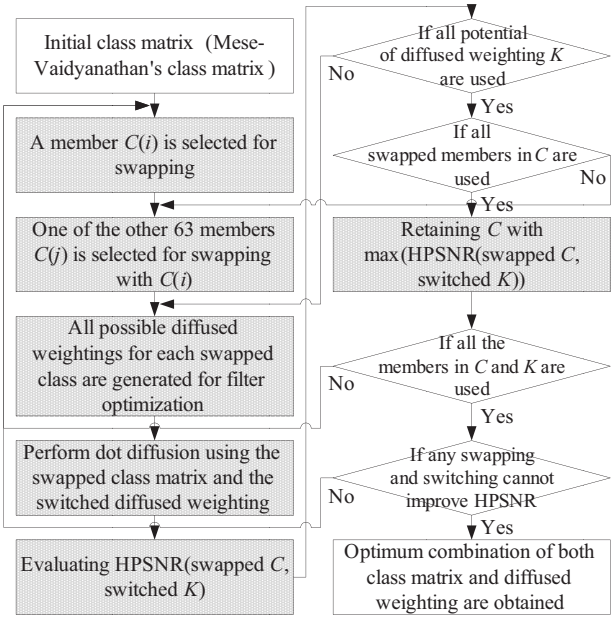


Fig. 6. Class matrix and diffused matrix co-optimization algorithm.

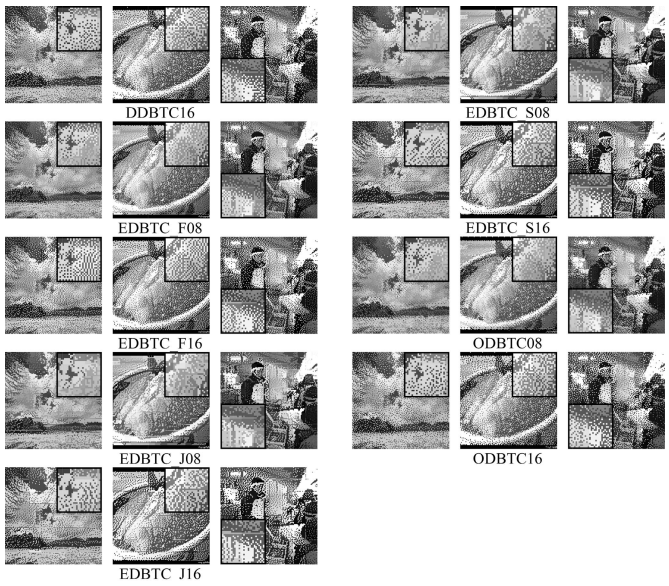


Fig. 7. BTC results of size 128 × 128 with various BTC techniques. (all printed at 150 dpi).

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 [22]) is adopted as the cost function for IQA as defined in (17)

$$\mathrm{HPSNR} = 10\log_{10} \times \frac{P \times Q \times 255^2}{\sum_{i=1}^{P}\sum_{j=1}^{Q}\left[\sum_{(m,n)\in R} w_{m,n}\left(x_{i+m,j+n} - y_{i+m,j+n}\right)\right]^2}, \tag{17}$$

where $x_{i,j}$ and $y_{i,j}$ denote the original image and reconstructed image of size $P \times Q$, respectively; the variable $w_{m,n}$ denotes the filter to simulate the low-pass property of HVS. To define this filter, the Gaussian filter with two parameters defined in [22], the standard deviation 1.3 and support region R of size 9×9, is adopted. According to the experimental results, slight modification of the parameter will not significantly change the results. Fig. 6 shows the procedure of the co-optimization algorithm, and the detailed steps are explained below.

Step 1. The Mese-Vaidyanathan's class matrix [20] 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 $i \neq j$.

Step 3. Generate all potential diffused weightings $k_{m,n} \in K$ 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 (17) 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.

Table I shows the two optimized class matrices of different sizes, 8×8 and 16×16, and their corresponding diffused matrices. For the sake of the co-optimization treatment, the proposed DDBTC can achieve even better image quality than that of the EDBTC and ODBTC. Notably, the bigger size in class matrix leads to less parallel advantage. For example, the whole bitmap of an image can be obtained in 64 unit times when a class matrix of size 8×8 is employed, while when the size of the class matrix grows to 16×16, then 256 unit times are required to obtain the whole bitmap of an image. Hence, there is a tradeoff between the size of the class matrix and processing efficiency.

IV. Experimental Results

The performance of the proposed DDBTC technique is analyzed and discussed in this section. Notably, in the following experiments, the optimized class matrix and diffused matrix as exhibited in Table I are used for evaluation. Those who desire to implement our system can simply use the matrices shown in Table I directly, and the implementation for that optimization is not needed. Fig. 6 shows three test images

TABLE II
AVERAGE OBJECTIVE IMAGE QUALITY COMPARISON (100 NATURAL IMAGES OF SIZE 512 × 512 ARE AVERAGED; THE BEST VALUE UNDER EACH IQA METHOD IS CIRCLED FOR VISUALIZATION)

	BTC Methods	IW-SSIM [23]	MS-SSIM [24]	IW-PSNR [23]	VIF [25]	HPSNR [22]
Block size = 8 × 8	EDBTC_F	0.9458	0.9409	34.38	**0.468159**	40.83
	EDBTC_J	0.9345	0.9318	30.64	0.40615	36.19
	EDBTC_S	0.9383	0.9366	31.25	0.419732	36.77
	ODBTC	0.9251	0.9344	28.18	0.364198	36.80
	Proposed DDBTC	**0.9491**	**0.9451**	**34.44**	0.464446	**41.95**
Block size = 16 × 16	EDBTC_F	0.9134	0.8882	31.03	**0.394108**	38.78
	EDBTC_J	0.8929	0.8707	26.45	0.3334	33.52
	EDBTC_S	0.9000	0.8817	27.09	0.3490	34.18
	ODBTC	0.8897	0.8820	26.96	0.2978	35.32
	Proposed DDBTC	**0.9147**	**0.8983**	**32.21**	0.3820	**39.23**

of size 128 × 128 to perceptually demonstrate the image quality. Moreover, a set of 100 other test grayscale images of size 512 × 512 is also employed to objectively evaluate the average performance. The reason that we chose a smaller test image for perceptual comparison is to magnify the detail texture with a small dot/inch (dpi) configuration, say 75 dpi. Throughout this paper, the results are yielded with Windows XP Professional Edition SP2 operating system, Intel Core (TM) 2 CPU 2.13 GHz, RAM 1.98 GB.

Fig. 7 shows the compressed results by various BTC techniques for subjective evaluation, in which the proposed DDBTC and other BTC methods the EDBTC [10] and ODBTC [18] are also involved. To fully estimate the EDBTC, three different error kernels, the Floyd-Steinberg [11] (abbr.: EDBTC_F), Jarvis *et al*. [12] (abbr.: EDBTC_J), and Stucki [13] (abbr.: EDBTC_S), are also considered. In addition, two types of block sizes, 8 × 8 and 16 × 16 (abbr.: for instance, DDBTC08 or DDBTC16), are also involved in the experiment. According to the results, the proposed DDBTC and the EDBTC_F offer not only the best but quite similar visual qualities under careful observation. In contrast, the results obtained by the rest methods still exhibit some prominent artifacts such as blocking effect or false contour. Among thee three EDBTCs, EDBTC_J and EDBTC_S possess the worst quality, since the error kernels of Jarvis *et al*. and Stucki are bigger than that of Floyd-Steinberg. This property leads to edge enhancement effect (see details in [12], [13]) and then strengthens the blocking effect.

To provide an objective image quality comparison, five different popular IQA methods, including information content weight structural similarity (IW-SSIM) [23], multi-scale SSIM (MS-SSIM) [24], IW-PSNR [23], visual information fidelity (VIF) [25], and HPSNR [22] are considered. The corresponding comparisons are shown in Table II. According to the comparison with the ODBTC, obvious improvements on all measures are demonstrated, which also matches the subjective comparison through Fig. 7's results. Moreover, pertaining the comparison with the EDBTC_F, although an inferior result of the proposed method is evaluated under the VIF metric, according to the article [23], the confidences of the IW-SSIM and MS-SSIM are both higher than that of the VIF. Thus, the above results yield a conclusion that the proposed DDBTC is still superior to that of the EDBTC.

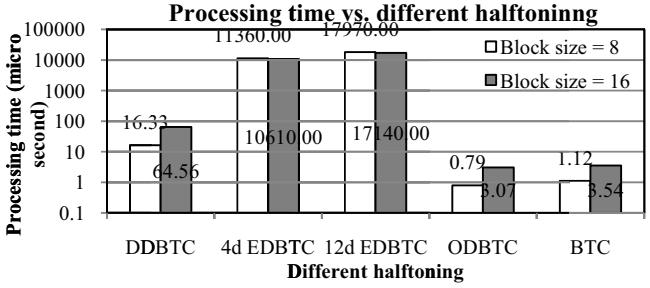


Fig. 8. Objective average processing efficiency comparisons (from 100 test images of size 512 × 512) among various BTC schemes.

Pertaining the processing efficiency comparison as shown in Fig. 8, the proposed method can yield excellent processing speed at 16.33μ and 64.56μ seconds for class matrices of sizes 8 × 8 and 16 × 16, respectively. These results are also equivalent to the frame rates of 61236.99 and 15489.47 fps (frame/second), respectively, for 512 × 512 grayscale image compression (far faster than the modern JPEG or JPEG2000 schemes). The notation #d in this figure denotes the number of diffusing directions of the employed error kernel. For instance, Floyd-Steinberg's error kernel has four diffused directions, while it is 12 directions for Jarvis *et al*.'s and Stucki's error kernels, respectively. These results demonstrate that the proposed DDBTC has important values and impacts in highly efficient or low power compression communication and related applications.

V. CONCLUSION

This paper presented 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. Although the proposed DDBTC provides satisfactory image quality, future work can be put to develop a better optimization algorithm to avoid class matrix and diffused matrix being trapped in local optimal states.

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.

[8] R. Ulichney, *Digital Halftoning*. Cambridge, MA, USA: MIT Press, 1987.

[9] D. J. Lieberman and J. P. Allebach, "A dual interpretation for direct binary search and its implications for tone reproduction and texture quality," *IEEE Trans. Image Process.*, vol. 9, no. 11, pp. 1950–1963, Nov. 2000.

[10] J. M. Guo, "Improved block truncation coding using modified error diffusion," *Electron. Lett.*, vol. 44, no. 7, pp. 462–464, Mar. 2008.

[11] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial gray scale," in *Int. Symp. Soc. Inf. Display, Dig.*, 1975, pp. 36–37.

[12] 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.

[13] P. Stucki, "MECCA: A multiple-error correcting computation algorithm for bilevel image hardcopy reproduction," IBM Res. Lab., Zurich, Switzerland, Res. Rep. RZ1060, 1981.

[14] 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.

[15] V. Ostromoukhov, "A simple and efficient error-diffusion algorithm," in *Proc. 28th Annu. Conf. Comput. Graph.*, 2001, pp. 567–572.

[16] P. Li and J. P. Allebach, "Tone-dependent error diffusion," *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 201–215, Feb. 2004.

[17] P. Li and J. P. Allebach, "Block interlaced pinwheel error diffusion," *J. Electron. Imag.*, vol. 14, no. 2, p. 023007, Apr.–Jun. 2005.

[18] 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.

[19] D. E. Knuth, "Digital halftones by dot diffusion," *ACM Trans. Graph.*, vol. 6, no. 4, pp. 245–273, Oct. 1987.

[20] M. Mese and P. P. Vaidyanathan, "Optimized halftoning using dot diffusion and methods for inverse halftoning," *IEEE Trans. Image Processing*, vol. 9, no. 4, pp. 691–709, Apr. 2000.

[21] J. M. Guo and Y. F. Liu, "Improved block truncation coding using optimized dot diffusion," in *Proc. IEEE Int. Symp. Circuits Syst.*, May–Jun. 2010, pp. 2634–2637.

[22] J. M. Guo and Y. F. Liu, "Joint compression/watermarking scheme using majority-parity guidance and halftoning-based block truncation coding," *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 2056–2069, Aug. 2010.

[23] Z. Wang and Q. Li, "Information content weighting for perceptual image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1185–1198, May 2011.

[24] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multi-scale structural similarity for image quality assessment," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, vol. 2. 2003, pp. 1398–1402.

[25] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.

Jing-Ming Guo (M'06–SM'10) received the Ph.D. degree from the Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan, in 2004. He is currently a Professor with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan. His research interests include multimedia signal processing, biometrics, computer vision, and digital halftoning.

Dr. Guo had acted as the Technical program Chair for IEEE International Symposium on Intelligent Signal Processing and Communication Systems in 2012, IEEE International Symposium on Consumer Electronics in 2013, and IEEE International Conference on Consumer Electronics in Taiwan in 2014. He has been invited as a lecturer for the IEEE Signal Processing Society summer school on Signal and Information Processing in 2012 and 2013. He has been elected as the Chair of the IEEE Taipei Section GOLD group in 2012. He has served as a Guest Co-Editor of two special issues for Journal of the Chinese Institute of Engineers and Journal of Applied Science and Engineering. He serves on the Editorial Board of the Journal of Engineering, The Scientific World Journal, International Journal of Advanced Engineering Applications, and Open Journal of Information Security and Applications. Currently, he is Associate Editor of the IEEE Transactions on Multimedia, IEEE Signal Processing Letters, the Information Sciences, and the Signal Processing.

Dr. Guo is a senior member of the IEEE, and a Fellow of IET. He has been promoted as a Distinguished Professor in 2012 for his significant research contributions. He received the Outstanding youth Electrical Engineer Award from Chinese Institute of Electrical Engineering in 2011, the Outstanding young Investigator Award from the Institute of System Engineering in 2011, the Best Paper Award from the IEEE International Conference on System Science and Engineering in 2011, the Excellence Teaching Award in 2009, the Research Excellence Award in 2008, the Acer Dragon Thesis Award in 2005, the Outstanding Paper Awards from IPPR, Computer Vision and Graphic Image Processing in 2005-2006 and 2013, and the Outstanding Faculty Award in 2002 and 2003.

Yun-Fu Liu (S'09–M'13) received the master's degree in electrical engineering from the Chang Gung University, Taoyuan, Taiwan, in 2009, and the PhD degree in electrical engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2013.

In 2012, he was awarded being involved in research with the Department of Electrical and Computer Engineering, University of California, Santa Barbara. In 2013, he joined the Multimedia Signal Processing Lab at the National Taiwan University of Science and Technology, Taipei, Taiwan, as a postdoctoral fellow. He has worked on foreground segmentation, biometrics, digital halftoning, watermarking, image compression and enhancement. His general interests lie in pattern recognition and image processing and their related applications.