Digital Signal Processing 20 (2010) 97–110

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

High efficiency ordered dither block truncation coding with dither array LUT and its scalable coding application

Jing-Ming Guo

Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

ARTICLE INFO

Article history:
Available online 3 May 2009

Keywords:
Halftone
Ordered dithering
Bit-interleaving
Progressive coding
Least-mean-squares

ABSTRACT

Block truncation coding is an efficient compression technique while offering good image quality. Nonetheless, the blocking effect inherent in BTC causes severe perceptual artifact in high compression ratio applications. In this study, the ordered dither block truncation coding (ODBTC) is proposed to solve this problem, where the dither array look up table (LUT) approach is proposed to significantly reduce the complexity of the BTC. Moreover, a novel progressive coding scheme is presented for efficiently displaying ODBTC images. The ODBTC utilizes the characteristic of the bit-interleaved bit map to determine the transmitting order and then progressively reconstructs the ODBTC image. In addition, the progressive ODBTC can provide higher compression ratio by toggling and entropy coding treatments. The experimental results demonstrate high-quality reconstructions while maintaining low transmitted bit rates.

1. Introduction

Block truncation coding (BTC) was first introduced by Delp and Mitchell in 1979 [1]. The basic concept is to divide the image into non-overlapped blocks, each pixel in a block is replaced by either high or low mean while preserving the first and second moments of the block. The main advantages of the method are the good image quality and the relatively lower complexity compared to the modern compression techniques, e.g. JPEG or JPEG2000. However, the image quality obtained by the traditional BTC degrades rapidly with the increasing in coding gain. Several investigations have addressed in the issue of further improving the image quality or coding gain of the BTC [2–7]. More details about these approaches and comparisons are organized later in Experimental results section.

Digital halftoning [8,9] is a technique for converting gray level images into two-tone binary images. These halftone images can resemble the original images when viewed from a distance by the low-pass filtering of the human visual system. Halftoning is commonly used in printing books, newspapers, and magazines because these printing processes can only generate two tones, black and white (with and without ink). There are several kinds of halftoning methods, including ordered dithering [8,9], error diffusion [10–21], iteration-based [22–26], and dot diffusion [27,28]. Among these, order dithering offers good visual quality and lowest computational complexity. The two important features make it quite suited in combining with BTC, namely ordered dither block truncation coding (ODBTC), to solve the low quality problem in high compression ratio application while maintaining the high efficiency feature of the traditional BTC. In fact, the complexity of the ODBTC is much lower than BTC.

The proposed ODBTC has an extra progressive coding functionality. Consider the case that a user is to select one from a number of images in a remote database. Assume the images are sequentially compressed prior to transmission. The lossless compression required gives an average compression ratio ranging from 1 to 2. In narrow bandwidth wireless environment,

E-mail address: jmguo@seed.net.tw.

1051-2004/$ – see front matter
doi:10.1016/j.dsp.2009.04.007

Table 1
Complexity comparisons between traditional BTC and the proposed ODBTC (total operations in a block of size $M \times N$).

	Addition/subtraction	Multiplication/division	Square root
Traditional BTC	$2 \times (M \times N) + 3$	$M \times N + 9$	2
ODBTC	$M \times N$	0	0

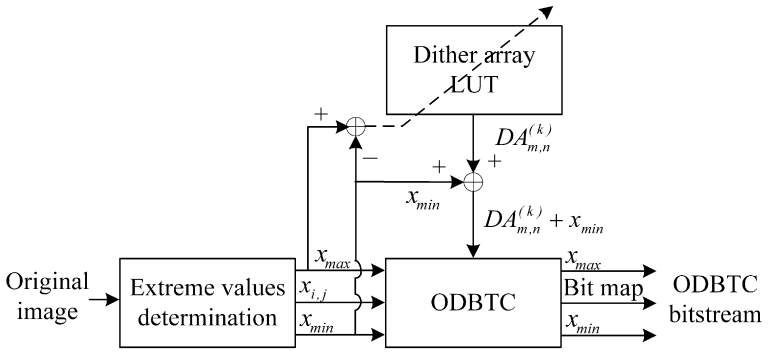


Fig. 1. Schematic diagram of the ODBTC.

the compression ratio may not be practical. Hence, a more effective approach is to use progressive coding to transmit an approximation of the images first at low bit rates. If the user identifies the image of interest, further refinement can be requested. To my knowledge, there is no progressive coding BTC scheme was presented in the literature. In this study, a novel progressive coding scheme is introduced for ODBTC images, since it has an exclusive bit-interleaved structure in bit map. Moreover, the coding gain can be improved with toggling and entropy coding treatments. Since the complexity of the progressive ODBTC is higher than BTC, it is suited for coding gain demanding applications. Conversely, the baseline ODBTC is suited for efficiency demanding applications.

The rest of the paper is organized as following. The baseline ODBTC is presented in Section 2. The progressive coding with toggling and entropy coding is introduced in Section 3. Section 4 demonstrates some experimental results. The conclusions are drawn in Section 5.

2. Ordered dither block truncation coding (ODBTC)

We now briefly describe the algorithm of the ordered dithering, and then introduce the proposed ODBTC. Suppose the dither array of the ordered dithering is of size $M \times N$ [8,9]. Each pixel $x_{i,j}$ in the original grayscale image is mapped to a member in dither array with value $DA_{i \bmod M,\, j \bmod N}$. The dithered output $b_{i,j}$ is determined as

$$b_{i,j} = \begin{cases} 255 \text{ (white)} & \text{if } x_{i,j} \geqslant DA_{i \bmod M,\, j \bmod N}, \\ 0 \text{ (black)} & \text{if } x_{i,j} < DA_{i \bmod M,\, j \bmod N}. \end{cases} \tag{1}$$

The dithered results can resemble the original grayscale image when view from a distance because of the low-pass nature of the human visual system.

On the other hand, the BTC also divides the original image into non-overlapped blocks of size $M \times N$. Suppose the variable $m = M \times N$, and $x_1, x_2, \ldots, x_m$ are the pixel values in a block. The traditional BTC replaces each pixel value with the evaluated high mean and low mean by comparing if the pixel value is higher or lower than the average of the block [1]. Herein, the algorithm is modified as below.

Suppose the maximum and minimum values of the block are denoted as and $x_{\max}$, $x_{\min}$, respectively. The size of the dither array is the same as the divided block in BTC. The proposed ODBTC can be written as

$$o_{i,j} = \begin{cases} x_{\max}, & \text{if } x_{i,j} \geqslant DA^{(k)}_{i \bmod M,\, j \bmod N} + x_{\min}, \\ x_{\min}, & \text{if } x_{i,j} < DA^{(k)}_{i \bmod M,\, j \bmod N} + x_{\min}, \end{cases} \tag{2}$$

where $o_{i,j}$ denotes the output pixel value, and $k = x_{\max} - x_{\min}$. The reason that the previous high mean and low mean are replaced with the two extreme values, $x_{\max}$ and $x_{\min}$, is that the threshold values in dither array may higher than the high mean or lower than the low mean. This phenomenon causes the reconstructed results perceived quite dull. A significant feature of the proposed ODBTC is the dither array LUT, where each specific size dither array has its corresponding 255 different scaling versions. The 255 scaling versions are obtained by

$$DA^{(k)}_{m,n} = k \times \frac{DA_{m,n} - DA_{\min}}{DA_{\max} - DA_{\min}}, \tag{3}$$

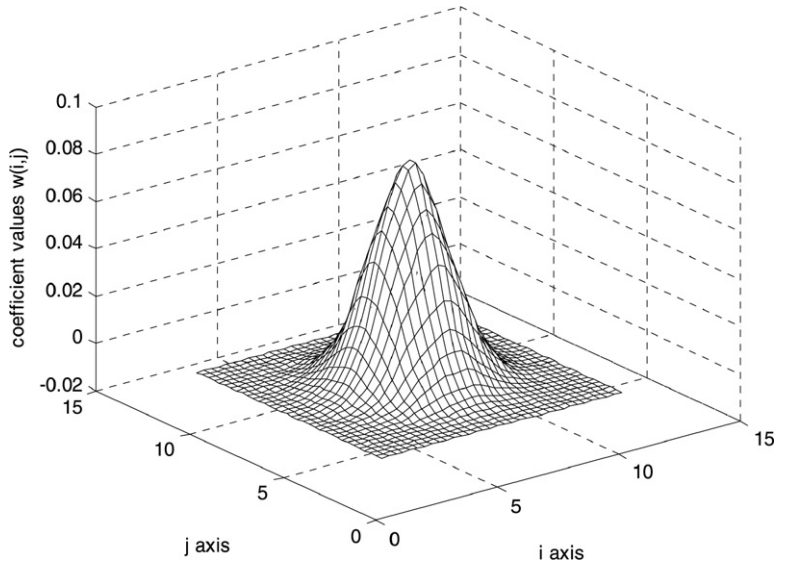


Fig. 2. Gaussian filter of size 7×7 ($\sigma = 1.3$).

where $1 \leqslant k \leqslant 255$, $1 \leqslant m \leqslant M$, and $1 \leqslant n \leqslant N$; $DA_{\min}$ and $DA_{\max}$ denote the minimum and maximum values in dithered array. The dynamic range of $DA_{m,n}^{(k)}$ is k, and the minimum value is 0. Consequently, the member values in $DA_{m,n}^{(k)}$ must be added by $x_{\min}$ to provide a fair thresholding with the pixel values in a block. Since the dither arrays $DA_{m,n}^{(k)}$ can be pre-calculated off-line as a LUT for usage, the complexity can be significantly reduced. The complexity comparisons between the BTC and the ODBTC are organized in Table 1, where the number $M \times N$ in addition/subtraction field of ODBTC is the value shifting caused by adding $x_{\min}$ as indicated in Eq. (2). The operation numbers in Table 1 are the sums in a block of size $M \times N$, which are proportional to the overall operation numbers of an image. The schematic diagram of the ODBTC is illustrated in Fig. 1.

Let the size of the original image be $P \times Q$. The error criterion involved in this study is defined as below.

$$PSNR = 10 \log_{10} \frac{P \times Q \times 255^2}{\sum_{i=1}^{P} \sum_{j=1}^{Q} [\sum_{m,n \in R} \sum w_{m,n}^2 (x_{i+m,j+n} - o_{i+m,j+n})^2]} \tag{4}$$

where the variables $x_{i,j}$ and $o_{i,j}$ represent the original grayscale image and the corresponding ODBTC image at position (i, j), respectively; R is the support region of the Gaussian distribution $w_{m,n}$, which is used to resemble the human low-pass nature as shown in Fig. 2. In this study, the standard deviation and the size are fixed at 1.3 and 7×7, respectively.

3. Progressive coding with toggling and entropy coding

3.1. Progressive coding

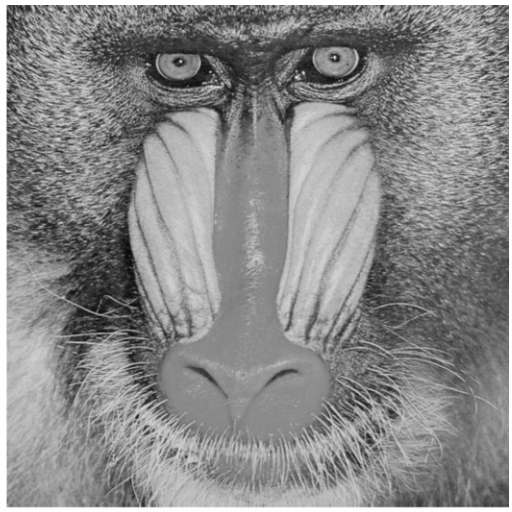
Let us start exploring the progressive coding ODBTC with the example given in Fig. 3. Fig. 3(c) shows the ODBTC result by processing the original image in Fig. 3(b) using the dither array in Fig. 3(a). The bit-interleaving [29] extracts and gathers all the pixels in bit map which map to the same threshold together to form a sub-image, and then resort each sub-images from left to right, bottom to top corresponding to the threshold values from lower to higher levels in Fig. 3(a) to form the image shown in Fig. 3(d), which composes of 64 sub-images. Without losing generality, the dither array is fixed at 8×8 for demonstration simplification.

A two-dimensional image such as Fig. 3(d) can be rearranged into the one-dimension data sequence form, as in Fig. 4(a), where each block represents a sub-image. Those sub-images in both extreme sides have more likelihood to be all white or all block. Suppose sub-images 1 to 8 are all white, and sub-images 57 to 64 are all black for example. Thus, the pixel values of these sub-images are not required to be transmitted, because the receiver recognizes it when it receives an overhead bit stream of 64 bits. In this bit stream, the bit "1" appears in the interval $(1, 32)$, which represents an all-white sub-image. On the other hand, the bit "1" appears in the interval $(33, 64)$, representing an all-black sub-image. By doing so, we can reduce a quarter of data set in this example and the receiver can reconstruct it by filling all white (or black) in the relative positions of image after receiving this bit stream. The advantage of this treatment will be enhanced when it comes to the lossy application which will be introduced in next sub-section.

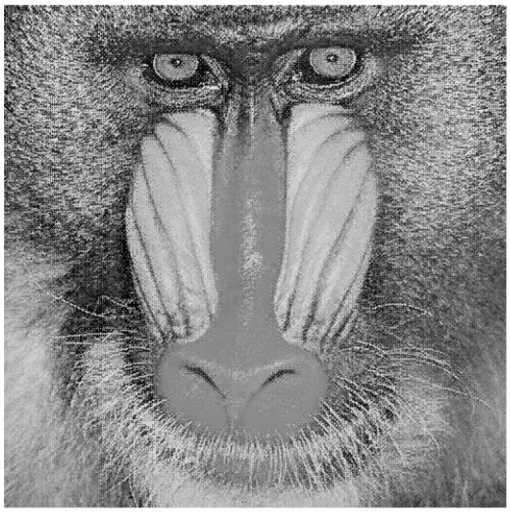
Fig. 3(d) shows that the sub-images in position "32" or "33" preserve a significant portion of the original image's features. In other words, the sub-images in position "32" or "33" should be transmitted and reconstructed first (the 32th sub-image

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

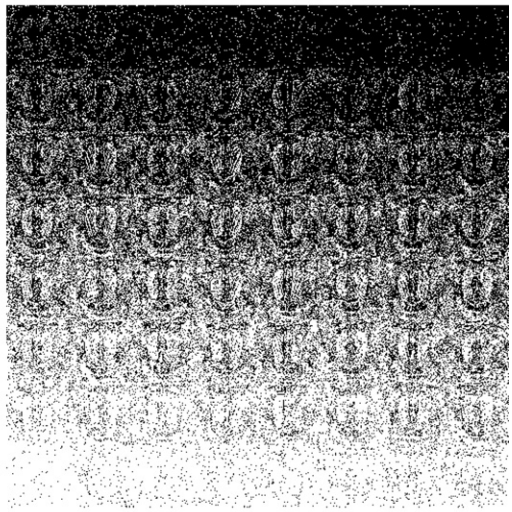
(a)



(b)

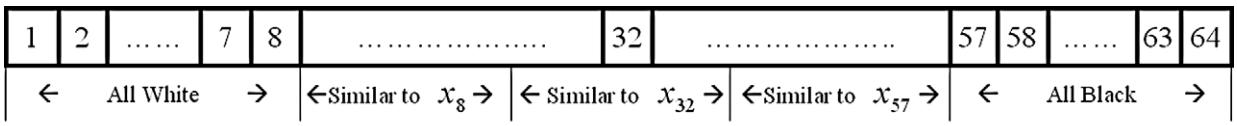


(c)

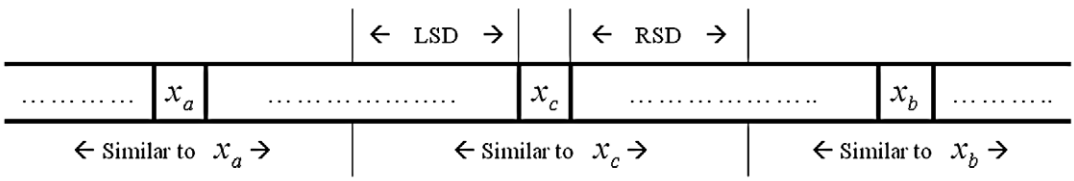


(d)

Fig. 3. ODBTC and the bit-interleaving results. (a) 8×8 dither array. (b) Original 512×512 grayscale Mandrill image. (c) ODBTC result. (d) Result of bit map bit-interleaving (all printed at 150 dpi).



(a)



(b)

Fig. 4. 1-D representation of subimages. (a) Order of 64 sub-images. (b) LSD and RSD of x_c.

is adopted in this work). The receiver reconstructs the image by duplicating it and filling in positions 9 to 56, and then conduct the inverse bit-interleaving. The reconstructed bit map is then replaced with the corresponding $x_{\max}$ and $x_{\min}$ in each block.

The algorithm for the determination of the subsequent sub-images to be transmitted is described as follows.

1. Let $x_1, x_2, \ldots, x_m$ represent the sub-images. ($m = n^2$, here $n = 8$, $m = 64$.)
2. Initialize $\text{flag}(x_1) = \text{flag}(x_2) = \cdots = \text{flag}(x_m) = 0$.
3. Set $\text{flag}(x_{\text{upper}}) = \text{flag}(x_{\text{lower}}) = 1$. ($x_{\text{lower}}$ represents the sub-image before the sub-image that a black pixel first appears if counting from x_1, where x_{upper} represents the sub-image before the sub-image that the white pixel first appears if counting down from x_m. When $\text{flag}(\bullet) = 1$, means the sub-image inside the parentheses should be transmitted or already sent.)
4. Set $\text{flag}(x_{\text{middle}}) = 1$, here x_{middle} represents x_{32} in this study.
5. Calculate the difference D_{ab} of every two adjacent sub-images, x_a and x_b, such that $\text{flag}(x_a) = \text{flag}(x_b) = 1$, define $D_{ab} = D_{ab}^{(1)} + D_{ab}^{(2)}$, where

$$D_{ab}^{(1)} = H(x_a, x_{a+1}) + H(x_a, x_{a+2}) + \cdots + H(x_a, x_c), \tag{5}$$

$$D_{ab}^{(2)} = H(x_{c+1}, x_b) + H(x_{c+2}, x_b) + \cdots + H(x_{b-1}, x_b), \quad a \leqslant c \leqslant b, \tag{6}$$

$D_{ab}^{(1)}$, $D_{ab}^{(2)}$ satisfy the condition $D_{ab}^{(1)} \cong D_{ab}^{(2)}$. ($\cong$ means as closer as possible, and $H(x_a, x_{a+1}) = \sum_{i=1}^{P'} \sum_{j=1}^{Q'} \{\sum_{m,n \in R} \sum w_{m,n}^2 [x_a(i+m, j+n) - x_{a+1}(i+m, j+n)]^2\}$ represents the weighted square distance between x_a and x_{a+1}, where $P' \times Q'$ stands for the dimension of a sub-image, and $w_{m,n}$ denotes the Gaussian filter described in Section 2.) Suppose $D_{ab}^{\max}$ is the maximum difference of all the adjacent sub-image pairs that $\text{flag}(x_a) = \text{flag}(x_b) = 1$, then x_c is the next sub-image to be transmitted, and set $\text{flag}(x_c) = 1$.
6. Fig. 4(b) shows the reconstructed left-side distance (LSD) and right-side distance (RSD) in the receiver, which satisfy the following conditions:

$$H(x_a, x_{a+1}) + H(x_a, x_{a+2}) + \cdots + H(x_a, x_{c-\text{LSD}-1}) \cong H(x_{c-\text{LSD}}, x_c) + H(x_{c-\text{LSD}+1}, x_c) + \cdots + H(x_{c-1}, x_c), \tag{7}$$

$$H(x_c, x_{c+1}) + H(x_c, x_{c+2}) + \cdots + H(x_c, x_{c+\text{RSD}}) \cong H(x_{c+\text{RSD}+1}, x_b) + H(x_{c+\text{RSD}+2}, x_b) + \cdots + H(x_{b-1}, x_b). \tag{8}$$

LSD and RSD are then represented by five bits and transmitted to the receiver as side information for reconstruction (the bit length is increased if the size of dither array is increased). In the receiver, the sub-images covered by LSD and RSD should be placed the same values as sub-image x_c.

3.2. Toggling processing and entropy coding

In the previous section, we introduced the procedure of progressive transmission for lossless reconstruction of an ODBTC images. In some practical applications, perfect reconstruction may not be as important as the compression ratio. Thus, it is of importance to analyze the trade-off between reconstruction errors and compression ratios. Observe that if there are more all black (white) sub-images in Fig. 3(d), a great deal of pixels in bit map are not required to be coded, since those uniform sub-images can be represented by the overhead bit stream as described above. Here in this sub-section, the toggling treatment in bit map is proposed to improve the coding gain as below:

1. Define a threshold N_{th}, as the total number of pixels of which the values can be changed in bit map in a reconstructed progressive ODBTC image.
2. Find the minimum minor pixel (minor pixel means black pixel in $x_1 \sim x_{\text{middle}}$ or white pixel in $x_{\text{middle}+1} \sim x_m$) number of the sub-image simultaneously searching up from x_1 and down form x_{64}.
3. Toggle a minor pixel value (white to black, or black to white) of the sub-image with minimum minor pixel as given in Step 2 and then subtract 1 from N_{th}.
4. Repeat Steps 2 and 3 until $N_{\text{th}} = 0$ (when $N_{\text{th}} = 0$ means already have amount of N_{th} pixels toggled in the bit map).

Thus we can produce a bit-interleaved image with more all black (white) sub-images, or generally speaking, an image with lower entropy.

Finally, we employ the extended lossless Huffman coder to perform further entropy coding with sub-images to be transmitted. Because the pixels are average white in the lower half plane and black in the upper half plane, as shown in Fig. 3(d), the coding gain can be improved if we separate the Huffman coding process into the lower plane and upper plane.

4. Experimental results

Eight test images of size 512×512 are involved in experiments as shown in Fig. 5(a), which include Lena, Mandrill, House, Boat, Castle, Peppers, Milk, and Lake images. The quality comparison between the traditional BTC and the proposed

Fig. 5. Thumbnail of 8 tested images. Left to right, top to bottom: Lena, Mandrill, House, Boat, Castle, Peppers, Milk, and Lake.

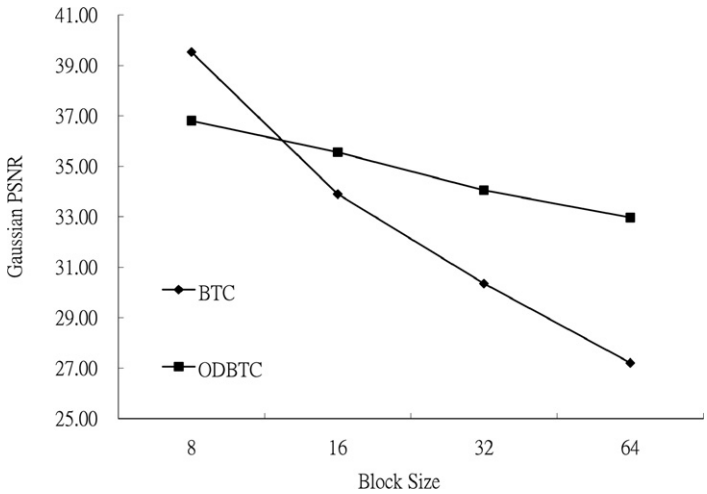


Fig. 6. Quality comparison between traditional BTC and the proposed ODBTC.

(a)

Fig. 7. BTC and ODBTC results under different block size. (a) Original image of size 512×512. (b) BTC result of bock size 16×16. (c) BTC result of block size 32×32. (d) ODBTC result of block size 16×16. (e) ODBTC result of block size 32×32.

ODBTC is shown in Fig. 6. The ODBTC scheme outperforms the BTC when block size is equal or higher than 16×16. The quality degrading speed of the ODBTC is much slower than that in BTC when coding gain is increased, which means that the ODBTC performs more stable than BTC. The stable coding feature is an important element for a good coder. The inside

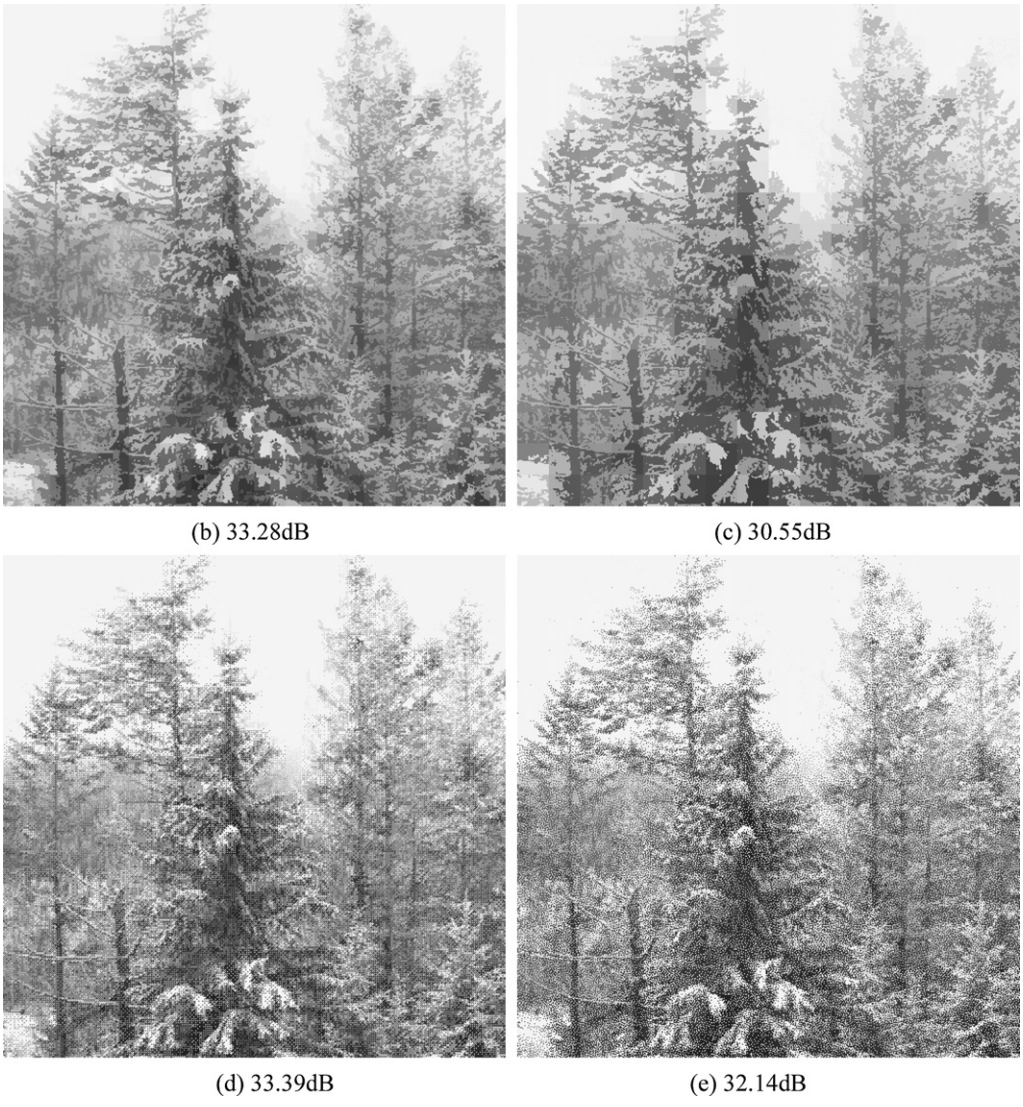


Fig. 7. (*continued*)

exploration that the proposed ODBTC can provide better image quality than that of BTC can be explained as below: In traditional BTC, the gray level of each block is reduced from 256 to 2, which results in prominent false contour and blocking effect. The dithering approach enjoys the benefit of easing false contour effect by alternately rendering the two-tone values in a block. In other words, a cluster region with high means in BTC will involve some low means by using ODBTC instead, and a cluster region with low means in BTC will involve some high means by using ODBTC instead. When view from a distance, the low pass nature of human vision can resemble the two-tone dithered region as a continuous-tone region, which also makes the boundary between two blocks look smoother and then reduces the blocking effect as well.

Fig. 7 shows an example of the comparison between BTC and ODBTC, where two types of block of sizes 16×16 and 32×32 are given. It is clear that the ODBTC has more stereo rendition than that in BTC. The objective quality of the ODBTC is also superior to BTC as 33.28 dB and 30.55 dB in BTC, and 33.39 dB and 32.14 dB in ODBTC, respectively, with block of sizes 16×16 and 32×32.

The complexity comparisons between the BTC and the ODBTC are organized in Table 1, where the number $M \times N$ in addition/subtraction field of ODBTC is the value shifting caused by adding $x_{\min}$ as indicated in Eq. (2). The operation numbers are the sums in a block of size $M \times N$, which are proportional to the overall operation numbers of an image.

The results of progressive reconstruction are shown in Figs. 8(a)–8(g), corresponding to steps 1 to 7, which are the reconstructions by receiving 1, 2, 4, 8, 16, 32, and 64 sub-images.

Fig. 9 shows the experimental PSNR versus number of steps, where five test images, Mandrill, Lena, Castle, Boat, and Peppers, are employed to show the convergent speed of the progressive reconstruction.

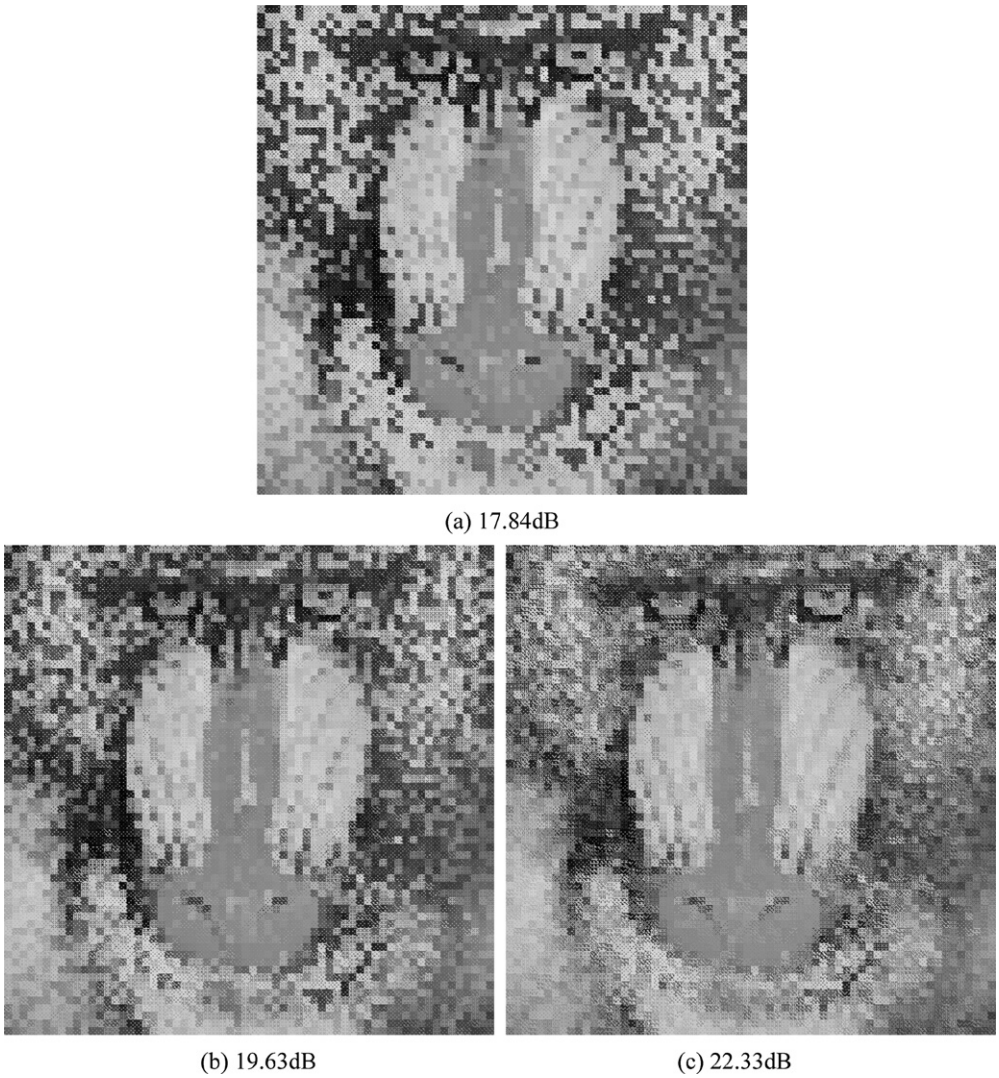


Fig. 8. Progressive reconstruction of an ODBTC image. (a) Original 512 × 512 grayscale Mandrill image. (b)–(h) Reconstructed ODBTC results in 7 steps (all printed at 150 dpi).

Table 2 shows the Huffman entropy compression ratio, and the occurrence number of the upper or lower sub-images in each coding step of the nine test images. The average bit rate of the eight test images is 0.69 bits/pixel.

Figs. 10 and 11 illustrate image quality and compression ratio versus a number of toggling numbers, where the unit of the horizontal axis is 1000. The result shows that the qualities of the test images are all satisfactory with PSNRs higher than 30 dB even under 10,000 toggling in bit map pixels. According to the experimental results, the degradation is not explicit. Fig. 12 shows a practical example to demonstrate this observation. Herein, it is suggested that moderate toggling number (e.g., 3000) for quality oriented applications, and more toggling number for coding gain oriented applications (e.g., 10,000). Table 3 shows the $(x_{\text{lower}}, x_{\text{upper}})$ of five test images under various toggling numbers. As expected that the range of the difference between x_{lower} and x_{upper} is shrinking when the toggling number is increasing, and the corresponding entropy is reducing.

Finally, comparisons among various former related approaches are organized in Table 4. In Udpikar and Paina [2], the vector quantization (VQ) is employed to further compress the overhead information of the BTC outputs. Although a high coding gain is achieved, the complexity is increased dramatically. On the other hand, since VQ itself is a lossy compression scheme, the reconstructed image quality is inferior to the typical BTC. In Wu and Coll [3], a hybrid coding model using VQ and DCT are applied for bit map and the high–low means, respectively, which increases the complexity as well. Nonetheless, the correlations of the bit map and high–low means between blocks are removed with the hybrid model. Thus, the coding gain is significantly improved. In Huang and Lin [4], the universal Hamming codes and the differential pulse code modulation (DPCM) are used to further compress the bit map and the high–low means to reduce bit rate and preserving moderate computational complexity. According to experimental results, the coding gain is close to that of Udpikar and

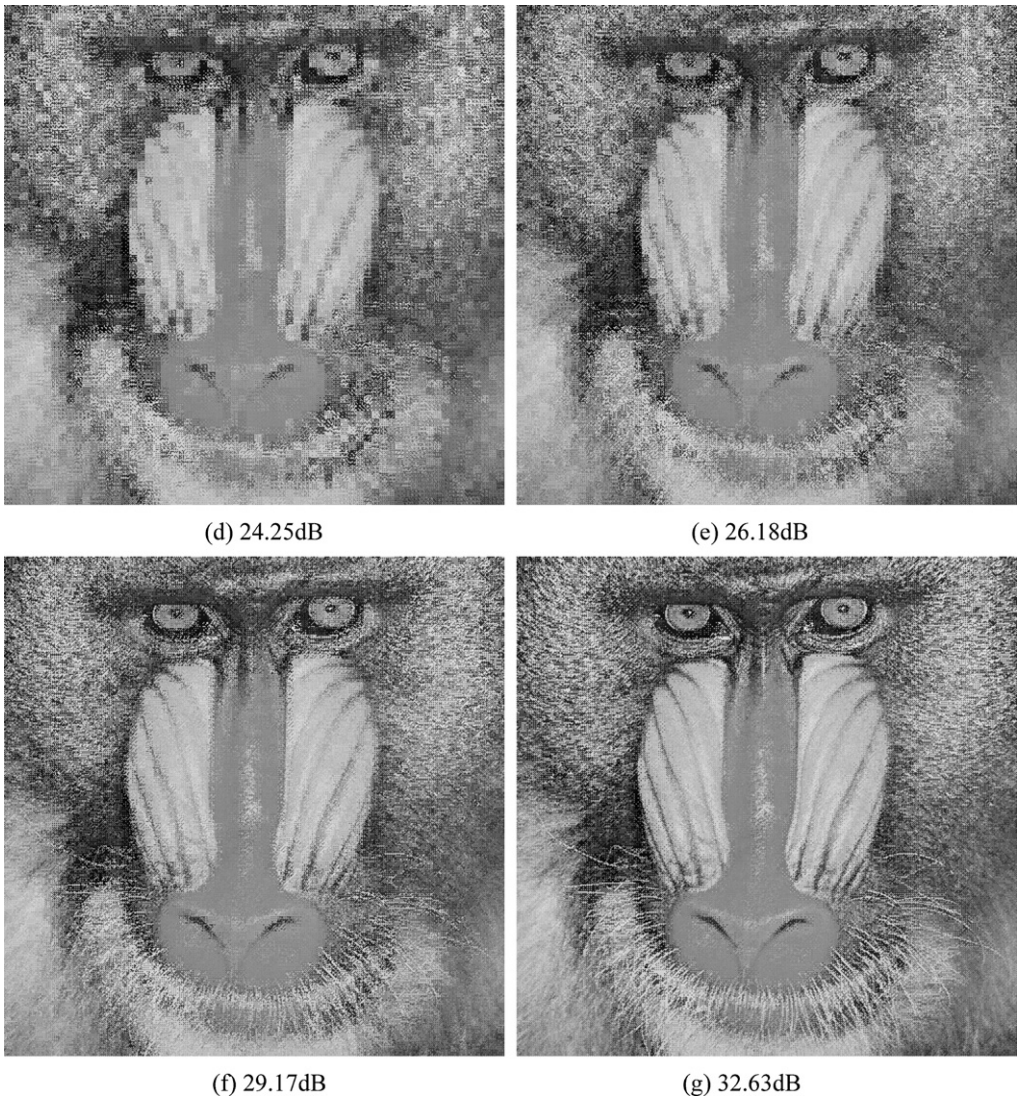


Fig. 8. (*continued*)

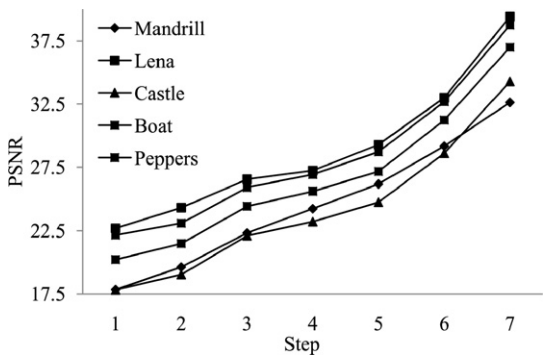


Fig. 9. PSNR vs. reconstructed steps of five test images.

Paina's method [2] with slightly sacrificing in image quality. In Wu and Tai [5], the feature extraction and visual significance selection steps are used to determine whether a block is further processed by AMBTC and entropy coding or discarded to flexibly adjust computation overhead and bit rate. The image quality is lower than [2] for the side effect "distortion propagation" introduced by the significance discarding. In Hu [6], a two-step criterion is employed to determine if a block is

Table 2
Bit rates with extended lossless Huffman coding.

Lena		Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
Upper	Transmitted subimages number	0	1	1	2	4	7	17
	Compression ratio	–	1.46	1.48	1.36	1.38	1.34	1.48
Lower	Transmitted subimages number	1	0	1	2	4	9	15
	Compression ratio	1.42	–	1.45	1.48	1.33	1.24	1.56
Average rate				0.7 bits/pixel				
Mandrill		Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
Upper	Transmitted subimages number	0	0	1	3	3	9	16
	Compression ratio	–	–	1.46	1.35	1.3	1.3	1.43
Lower	Transmitted subimages number	1	1	1	1	5	7	16
	Compression ratio	1.43	1.45	1.43	1.5	1.27	1.26	1.63
Average rate				0.71 bits/pixel				
House		Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
Upper	Transmitted subimages number	0	1	1	2	4	8	16
	Compression ratio	–	1.48	1.49	1.39	1.33	1.34	1.42
Lower	Transmitted subimages number	1	0	1	2	4	8	16
	Compression ratio	1.45	–	1.45	1.69	1.32	1.28	1.57
Average rate				0.7 bits/pixel				
Boat		Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
Upper	Transmitted subimages number	0	0	1	3	3	9	16
	Compression ratio	–	–	1.45	1.34	1.31	1.83	1.4
Lower	Transmitted subimages number	1	1	1	1	5	7	16
	Compression ratio	1.44	1.5	1.46	1.62	1.33	1.32	1.69
Average rate				0.66bits/pixel				
Castle		Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
Upper	Transmitted subimages number	0	0	1	2	5	9	15
	Compression ratio	–	–	1.47	1.34	1.31	1.32	1.39
Lower	Transmitted subimages number	1	1	1	2	3	7	17
	Compression ratio	1.43	1.47	1.44	1.46	1.27	1.25	1.57
Average rate				0.72 bits/pixel				
Peppers		Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
Upper	Transmitted subimages number	0	0	1	3	3	8	17
	Compression ratio	–	–	1.46	1.33	1.37	1.3	1.4
Lower	Transmitted subimages number	1	1	1	1	5	8	15
	Compression ratio	1.43	1.49	1.46	1.57	1.36	1.31	1.68
Average rate				0.7 bits/pixel				
Milk		Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
Upper	Transmitted subimages number	0	1	1	2	4	8	16
	Compression ratio	–	1.47	1.61	1.46	1.5	1.54	1.67
Lower	Transmitted subimages number	1	0	1	2	4	8	16
	Compression ratio	1.45	–	1.45	1.42	1.28	1.19	1.47
Average rate				0.68 bits/pixel				
Tree		Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
Upper	Transmitted subimages number	0	1	1	2	3	9	16
	Compression ratio	–	1.52	1.49	1.36	1.32	1.31	1.41
Lower	Transmitted subimages number	1	0	1	2	5	7	16
	Compression ratio	1.46	–	1.49	1.65	1.35	1.35	1.62
Average rate				0.69 bits/pixel				

encoded from the neighboring predefined 12 coded blocks to reduce the bit rate. However, since 12 neighboring blocks are used to calculate Euclidean distance with the current block, extra computational overhead are required in the approach. One recent work proposed by Guo [7] which coordinates error diffusion and BTC to improve the image quality. The error diffusion can effectively diffuse the error between the original gray tone and the replaced two-tone value to the neighboring pixels and then maintains the average grayscale in a local region. However, since the error diffusion needs additional multiplications/divisions and additions/subtractions for its error kernel, the complexity is still higher than that of the proposed ODBTC. The proposed ODBTC needs neither multiplication/division nor square root, and thus has the lowest complexity among various approaches. The inside exploration that the proposed ODBTC can provide better image quality than that of BTC can be explained as below: In traditional BTC, the gray level of each block is reduced from 256 to 2, which results in prominent false contour and blocking effect. The dithering approach enjoys the benefit of easing false contour effect by alternately rendering the two-tone values in a block. In other words, a cluster region with high means in BTC will involve

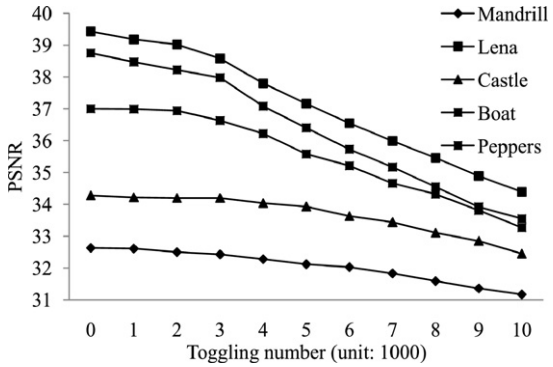


Fig. 10. PSNR vs. toggling number of five test images.

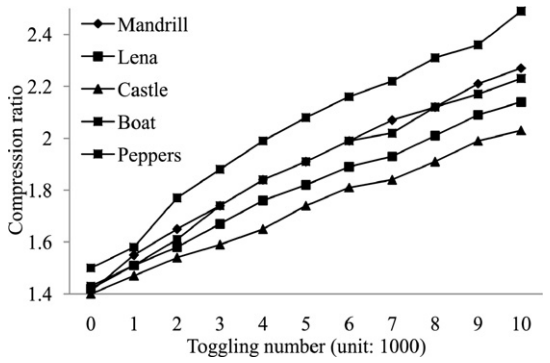


Fig. 11. Compression ratio vs. toggling number of five test images.

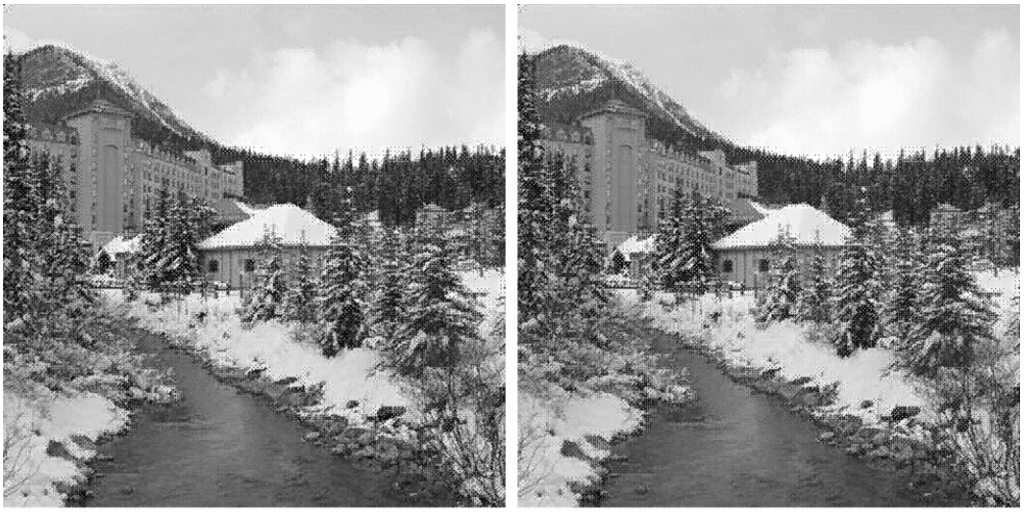


Fig. 12. Toggling processing results with Castle image. (a) No toggling. (b) 2000 toggles. (c) 4000 toggles. (d) 6000 toggles. (e) 8000 toggles. (f) 10,000 toggles.

some low means by using ODBTC instead, and a cluster region with low means in BTC will involve some high means by using ODBTC instead. When view from a distance, the low pass nature of human vision can resemble the two-tone dithered region as a continuous-tone region, which also makes the boundary between two blocks look smoother and then reduces the blocking effect as well.

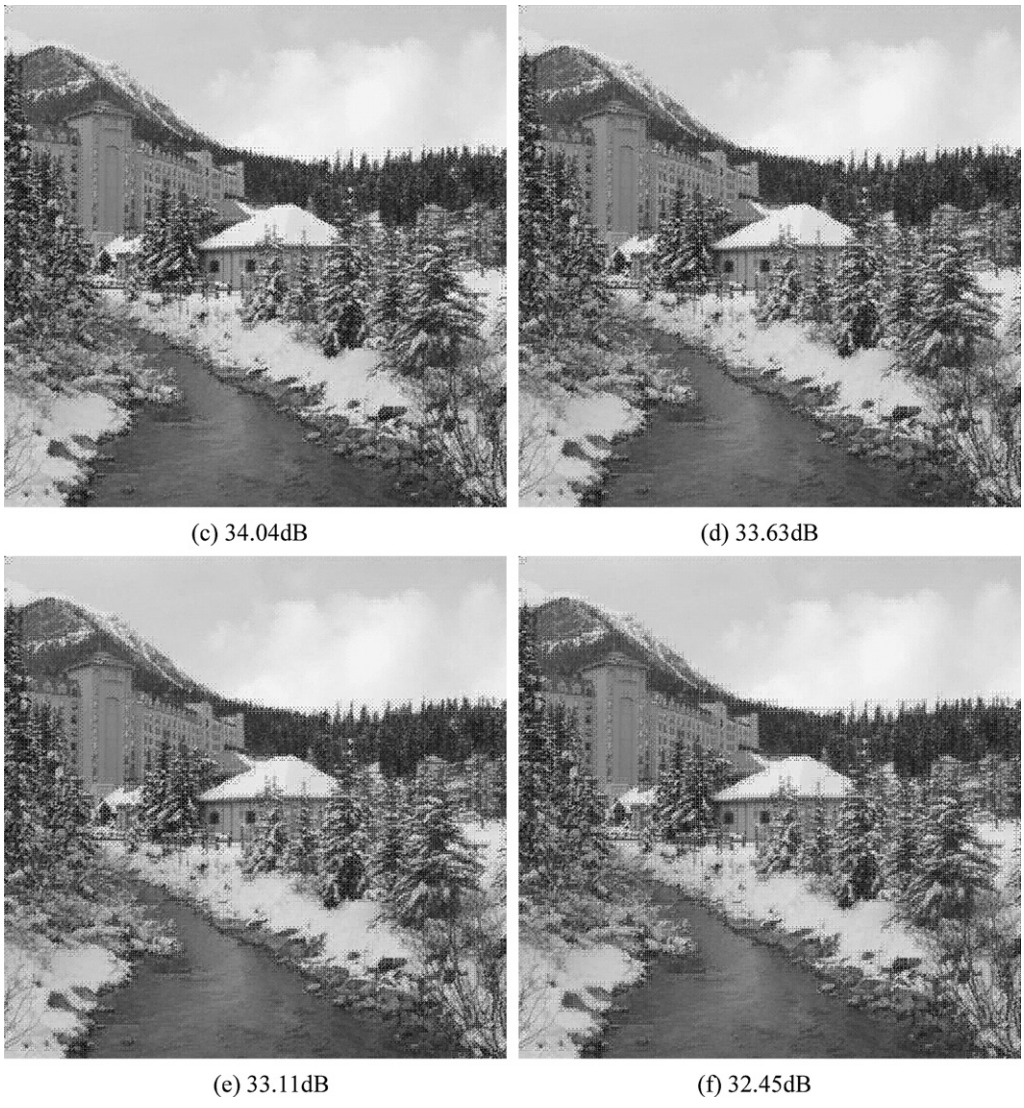


Fig. 12. (*continued*)

Table 3
$(x_{\text{lower}}, x_{\text{upper}})$ of five test images under various toggling numbers (unit: 1000).

	Mandrill	Lena	Castle	Boat	Peppers
1	(4, 61)	(1, 60)	(3, 63)	(3, 61)	(1, 60)
2	(6, 59)	(4, 58)	(6, 63)	(5, 58)	(3, 58)
3	(8, 58)	(7, 58)	(7, 62)	(8, 58)	(7, 58)
4	(9, 56)	(8, 56)	(8, 61)	(9, 56)	(9, 57)
5	(9, 54)	(9, 55)	(9, 59)	(11, 56)	(10, 56)
6	(11, 54)	(10, 54)	(10, 58)	(11, 54)	(11, 55)
7	(12, 53)	(11, 54)	(11, 58)	(12, 54)	(11, 54)
8	(12, 52)	(11, 52)	(11, 56)	(12, 52)	(13, 54)
9	(13, 51)	(11, 50)	(11, 54)	(12, 51)	(14, 54)
10	(14, 51)	(11, 49)	(12, 54)	(14, 51)	(15, 54)

5. Conclusions

In this study, an improved version of BTC, namely ordered dither block truncation coding (ODBTC), is proposed to provide better image quality and much lower complexity. Moreover, based on the observation that the bit map of the ODBTC can offer lower entropy after rearranging with bit-interleaving, the ODBTC image can be extended to progressive coding application. The coding gain can be further improved by adequately cooperating entropy coding with the

Table 4
Performance comparisons among various methods.

Method	Quality	Complexity	Coding gain	Additional feature
Udpikar and Paina [2]	Lower than typical BTC	Very high (VQ is used)	High	None
Wu and Coll [3]	Lower than typical BTC	Very high (VQ and DCT are used to compress bit map and high–low means, respectively)	Very high	None
Huang and Lin [4]	Slightly lower than [2]	Moderate (universal Hamming codes and DPCM are used to compress bit map and high–low means)	Similar to [2]	None
Wu and Tai [5]	Lower than [2] (distortion propagation by the significance discarding)	Moderate (additional feature extraction preprocessing and entropy coding are used)	High (additional feature extraction preprocessing and entropy coding are used)	Variable bit rate (when low information content block discarding is enable)
Hu [6]	Slightly lower than typical BTC	Moderate (12 neighboring blocks are used to calculate Euclidean distance with the current block)	High (BOC and smooth-complex blocks separation are applied)	None
Guo [7]	Highest (higher than typical BTC)	Low (without high and low mean calculation)	Low (no further compression for bit map an high–low means)	None
Proposed method	Very good (higher than typical BTC)	1. Lowest (without high and low mean calculation)	High (when progressive coding is enable)	1. Progressive reconstruction
		2. Moderate (when progressive coding is enable)		2. Variable bit rate (toggling number in bit map is controllable)

important feature of the bit-interleaved bit map that the majority pixel is opposite in upper and lower image. Finally, the coding gain can be even increased with the proposed toggling treatment to lower down the overall entropy of the bit map. The experimental results show that good image quality is available even under a huge bit map pixel toggling.

References

[1] E.J. Delp, O.R. Mitchell, Image compression using block truncation coding, IEEE Trans. Commun. 27 (1979) 1335–1342.
[2] V.R. Udpikar, J.P. Raina, BTC image coding using vector quantization, IEEE Trans. Commun. 35 (1987) 352–356.
[3] Y. Wu, D. Coll, BTC-VQ-DCT hybrid coding of digital images, IEEE Trans. Commun. 39 (1991) 1283–1287.
[4] C.S. Huang, Y. Lin, Hybrid block truncation coding, IEEE Signal Process. Lett. 4 (12) (1997) 328–330.
[5] Y.G. Wu, S.C. Tai, An efficient BTC image compression technique, IEEE Trans. Consum. Electron. 44 (2) (1998) 317–325.
[6] Y.C. Hu, Improved moment preserving block truncation coding for image compression, Electron. Lett. 39 (19) (2003) 1377–1379.
[7] J.M. Guo, Improved block truncation coding using modified error diffusion, Electron. Lett. 44 (7) (2008) 462–464.
[8] R. Ulichney, Digital Halftoning, MIT Press, Cambridge, MA, 1987.
[9] D.L. Lau, G.R. Arce, Modern Digital Halftoning, Marcel Dekker, Inc., New York/Basel, 2000.
[10] R.W. Floyd, L. Steinberg, An adaptive algorithm for spatial gray scale, Digest. Soc. Inform. Displ. (1975) 36–37.
[11] J.F. Jarvis, C.N. Judice, W.H. Ninke, A survey of techniques for the display of continuous-tone pictures on bilevel displays, Comp. Graph. Image Process. 5 (1976) 13–40.
[12] P. Stucki, MECCA—A multiple-error correcting computation algorithm for bilevel image hardcopy reproduction, Res. Rep. RZ1060, IBM Res. Lab., Zurich, Switzerland, 1981.
[13] V. Ostromoukhov, A simple and efficient error-diffusion algorithm, in: Proceedings of SIGGRAPH 2001, Computer Graphics, 2001, pp. 567–572.
[14] T.C. Chang, J.P. Allebach, Memory efficient error diffusion, IEEE Trans. Image Process. 12 (11) (2003) 1352–1366.
[15] P. Li, J.P. Allebach, Block interlaced pinwheel error diffusion, J. Electron. Imaging 14 (2) (2005), 023007-1-13.
[16] D.L. Lau, G.R. Arce, Robust halftoning with green-noise, in: Proceedings of the IS&T's Image Processing, Image Quality, Image Capture Systems Conference, Savannah, GE, April 25–28, 1999, pp. 315–320.
[17] D.L. Lau, G.R. Arce, N.C. Gallagher, Green-noise digital halftoning, Proc. IEEE 86 (12) (1998) 2424–2444.
[18] D.L. Lau, G.R. Arce, N.C. Gallagher, Digital halftoning via green-noise masks, J. Opt. Soc. Am. 16 (7) (1999) 1575–1586.
[19] D.L. Lau, G.R. Arce, N.C. Gallagher, Digital color halftoning via generalized error-diffusion and vector green-noise masks, IEEE Trans. Image Process. 9 (5) (2000).
[20] R. Levien, Output dependant feedback in error diffusion halftoning. in: IS&T's Eighth International Congress on Advances in Non-Impact Printing Technologies, Williamsburg, VA, USA, October 25–30, 1992, pp. 280–282.
[21] D.L. Lau, G.R. Arce, R. Ulichney, Blue and green noise halftoning models, IEEE Signal Process. Mag. 20 (4) (2003) 28–38.
[22] A. Zakhor, S. Lin, F. Eskafi, A new class of B/W and color halftoning algorithm, in: Proc. ICASSP-91, vol. 4, Toronto, Canada, May 1991, pp. 2801–2804.
[23] M. Analoui, J.P. Allebach, Model based halftoning using direct binary search, in: Human Vision, Visual Proc., Digital Display III, San Jose, CA, Proc. SPIE, vol. 1666, Feb. 1992, pp. 96–108.
[24] D. Anastassion, Error diffusion coding for A/D conversion, IEEE. Trans. Circuits Syst. 36 (9) (1989) 1175–1186.
[25] D. Anastassiou, S. Kollias, Digital image halftoning using neural networks, in: Proc. of SPIE, vol. 1001, 1988.
[26] J.B. Mulligan, A.J. Ahumada Jr., Principled halftoning based on models of human vision, in: Human Vision, Visual Proc., Digital Display III, San Jose, CA, Proc. SPIE, vol. 1666, Feb. 1992, pp. 109–121.
[27] D.E. Knuth, Digital halftones by dot diffusion, ACM Trans. Graph. 6 (4) (1987).

[28] M. Mese, P.P. Vaidyanathan, Optimized halftoning using dot diffusion and methods for inverse halftoning, IEEE. Trans. Image Process. 9 (4) (2000) 691–709.
[29] C.N. Judice, Data reduction of dither coded images by bit interleaving, Proc. Soc. Inform. Display IT-17 (2) (1976) 91–97.

Jing-Ming Guo was born in Kaohsiung, Taiwan, in 1972. He received the B.S.E.E. and M.S.E.E. degrees from National Central University, Taoyuan, Taiwan, in 1995 and 1997, respectively, and the Ph.D. degree from the Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan, in 2004.

From 1998 to 1999, he was an Information Technique Officer with the Chinese Army. From 2003 to 2004, he was granted the National Science Council scholarship for advanced research from the Department of Electrical and Computer Engineering, University of California, Santa Barbara. He is currently an Associate Professor with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei. His research interests include multimedia signal processing, multimedia security, digital halftoning, and digital watermarking.

Dr. Guo is a member of the IEEE Signal Processing Society. He received 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 and 2006, and the Outstanding Faculty Award in 2002 and 2003.