Reversible Data Hiding in Highly Efficient Compression Scheme

Jing-Ming Guo, Member, IEEE and Jia-Jin Tsai

Department of Electrical Engineering,
National Taiwan University of Science and Technology
Taipei, Taiwan
E-mail: jmguo@seed.net.tw, M9607212@mail.ntust.edu.tw

ABSTRACT

Nowadays, most multimedia is stored in compressed bit stream format to save the storage apace or transmission time. This study proposes a novel technique for embedding flexible amounts of data in the bitmap of the improved Ordered Dither Block Truncation Coding (ODBTC) image, where the ordered dithering is used to dither the quantized BTC image to avoid the annoying false contour and blocking effect inherently existed in BTC image. Moreover, the LUT strategy is also used to significantly reduce the complexity. The inverse halftoning and the second round of halftoning are employed as the key steps in locating the embedded information bits. Experimental results demonstrate that an objective good quality image with flexible capacity and reasonable complexity is obtained. Moreover, the correct decoding rate of 100% is maintained, and the original host ODBTC image can also be reconstructed in the decoder when needed, which significantly boosts the flexibility in image quality control.

***Index Terms*–block truncation coding, ordered dithering, false contour, blocking effect, halftoning, data hiding.**

1. INTRODUCTION

With the rapid advancement of information technology, the Internet has become a convenient channel for data transmission. The impetus for digital watermarking in order to maintain control over one's ownership and data integrity during transmission of images thus arises [1].

Block truncation coding (BTC) [2, 3], presented by Delp and Mitchell in 1979 provides a simple and efficient compression solution. To encode each non-overlapping image block, all pixels in each block are replaced by two characteristic values which preserve the first and second moment. For every block, only its high mean, low mean and bitmap need to be transmitted for an image of high visual quality to be reconstructed at the receiving end. To improve image quality, many schemes have been proposed. These includes BTC algorithms combined with vector quantization (VQ) [4] or discrete cosine transform (DCT) [5]; modified algorithms of BTC such as the absolute moment BTC (AMBTC) have also been proposed [6]. Yang and Tsai [7] proposed an Improved BTC (IBTC) method using a set of predefined line and edge bit planes adaptively selected to yield lower bit rates and better reconstructed image quality.

In the first part of this paper, we present a novel method of BTC which incorporates the use of ordered dithering [8, 9]. Ordered dithering has originally been used in digital halftoning whereby multi-tone images are converted into two-tone formats. These two-tone formats are widely used in computer printouts, books, newspapers, and magazines [8], [9] as they still resemble a continuous image due to the low-pass nature of the Human Visual System (HVS).There are many kinds of halftoning techniques, including ordered dithering, error diffusion, direct binary search, and dot diffusion. Among these, ordered dithering provides good image quality and the lowest computational complexity. In this paper, we have employed the dither array in ordered dithering to use in thresholding instead of the block average in BTC, for the generation of the bitmap. The extreme pixel values, the maxima and minima, in a block have also used in place of the high mean and low mean to optimize the effects of a dither screen.

Data hiding schemes using BTC have also been looked into previously. For example, in a paper by Tu and Hsu [10], an ownership share, which is a combination of the image and its watermark, is generated as a key and leaves the original image unmodified. The ownership share is then required in the process of decoding. Lin and Chang [11] also proposed a data hiding scheme for BTC compressed images, by embedding information into the high and low means, alternating one bit value according to the messages' value, as well as into the bitmap using the minimum distortion algorithm. This creates sufficient hiding capacity, yet maintaining traditional BTC image quality. However, these approaches cannot recover the original BTC image when needed, and the embedded capacity is not flexible.

The main part of this work looks into utilizing the bitmap generated by ODBTC for data hiding purposes based on the Embeddable Cells Selection (ECS) that provides good image quality and flexible capacity with reasonable computational complexity. Moreover, the original ODBTC image can also be reconstructed in the decoder.

The rest of the study is organized as follows. Section 2 introduces the performance evaluations used in this work. Section 3 introduces the proposed data hiding in ODBTC scheme. The experimental results are given in Section 4, and Section 5 draws conclusions.

2. PERFORMANCE EVALUATIONS

Suppose an image of size $P \times Q$, the quality assessment used in this study is defined as below:

$$PSNR = 10 \log_{10} \frac{P \times Q \times 255^2}{\sum_{i=1}^{P} \sum_{j=1}^{Q} (x_{i,j} - \sum_{m,n \in R} \sum w_{m,n} x'_{i+m,j+n})^2} \qquad (1)$$

where $x_{i,j}$ denotes the original grayscale image; $x'_{i,j}$ denotes the reconstructed image; $w_{m,n}$ denotes the Gaussian filter at position (m,n), and R denotes the support region of a Gaussian filter. In this work, the size of the filter is fixed at 7×7 with standard deviation σ=1.3, which is used to simulate the human visual response.

The criterion used to evaluate the Correct Decoding Rate (CDR) of the secret message is defined as below:

$$CDR = \frac{M \times N - HD(BM_{i,j}, BM'_{i,j})}{M \times N} \times 100\%, \qquad (2)$$

where $BM_{i,j}$ and $BM'_{i,j}$ denote the original binary message and the decoded message, respectively. Both of the messages are of size $M \times N$. The symbol $HD(\bullet)$ denotes the Hamming distance between the two messages. The CDR is a measurement to determine the similarity between the original message and the corresponding decoded message. A good data hiding scheme provides a higher CDR.

3. PROPOSED DATA HIDING IN ODBTC

A. Ordered Dither Block Truncation Coding (ODBTC)

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

$$o_{i,j} = \begin{cases} x_{max}, & if \quad x_{i,j} \geq DA^{(k)}_{i \bmod M,\, j \bmod N} + x_{min} \\ x_{min}, & if \quad x_{i,j} < DA^{(k)}_{i \bmod M,\, j \bmod N} + x_{min}, \end{cases} \quad (3)$$

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 be higher than that of the high mean or lower than that of 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

$$DA^{(k)}_{m,n} = k \times \frac{DA_{m,n} - DA_{min}}{DA_{max} - DA_{min}}, \quad (4)$$

where $1 \leq k \leq 255$, $1 \leq m \leq M$, and $1 \leq n \leq N$; DA_{min} and DA_{max} denote the minimum and maximum values in dithered array. The dynamic range of $DA^{(k)}_{m,n}$ is k, and the minimum value is 0. Consequently, the member values in $DA^{(k)}_{m,n}$ must be added by x_{min} to provide a fair thresholding with the pixel values in a block. Since the dither arrays $DA^{(k)}_{m,n}$ 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. 3. 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. 2.

B. Data Hiding in ODBTC

Since the bitmap of the ODBTC is similar to a dithering halftone image after thresholding with the proposed LUT dither arrays. In this work, the secret information is embedded in its bitmap. Figure 3(a) shows the encoded flow chart. The procedure of the proposed encoding is briefly introduced as below. The embedding is a cell-wise method. The secret information bits should be embedded into the high frequency cell of the bitmap. The thresholds in the dither array are arranged in a dispersed fashion. If a bitmap pattern belongs to the low frequency region of the image, then the dots in the pattern have higher likelihood of being in a checkerboard arrangement (black and white dots appear alternatively). Due to the low pass nature of the human eye, the alteration is easily perceived when pixel values are changed, since the modified portions are in clustered black or white. Conversely, if the bitmap pattern is in the high-frequency region of the image, the alteration is more difficult to be perceived. Hence, in the encoder, the high frequency cells in bitmap are first located with the proposed technique as introduced below. All the pixel values in the cell that corresponds to information bit 1 in the secret message are then reversed. The cells with close black and white pixel numbers can intuitively be exploited, so that even when all the pixel values in the cell are reversed, the average gray level is remained similar. For this reason, the cells with close black and white pixel numbers are reserved in the encoding. The detailed encoding is introduced as below.

Because the halftone screen (Bayer-5) employed has diagonal symmetry property, the original bitmap is divided into cells of size 4×4. The first step of the embedding is to select and reverse the cell with nearly the number of black and white pixels satisfying the following equation:

$$\left| \#black\ pixel – \#white\ pixel \right| < BW_{Th} \quad (5)$$

where BW_{Th} denotes an adjustable threshold parameter. The number of resersable cells rises as the values of BW_{Th} rises, and vice versa. In the second step, the cells are continuously extracted by filtering the reserved cells obtained from Eq. 5 with Eq. 6.

$$CON_{max} \geq \#connect \geq CON_{min} \quad (6)$$

Equation 6 searches for the cell with the highest frequency from the reserved cells retained by Eq. 5. The variable $\#connect$ is defined as follows. Consider a pixel P at location (m, n) and its neighbors in a 3×3 region, as shown in Fig. 4. A variable con(m, n) is defined as follows:

$$con(m,n) = \sum_{i=0}^{7} w(i) h(P, P_i) \quad (7)$$

where

$$h(P,P_i) = \begin{cases} 1 & P = P_i, \\ 0 & P \neq P_i, \end{cases}$$

$$w(i) = \begin{cases} 1 \quad , & for \quad i = 1, 3, 4, 6 \\ 0 \quad , & for \quad i = 0, 2, 5, 7 \end{cases}$$

The total sum of $con(m,n)$ in a cell is defined as：

$$\#connect = \sum_{m,n \in \frac{M}{2} X \frac{N}{2}} \sum con(m,n) \quad (8)$$

where $M \times N$ stands for the size of the halftone screen, and equals 8x8 since the Bayer-5 halftone screen as shown in Fig. 1 is applied in this example. The variables CON_{max} and CON_{min} denote the two other tunable parameters. The larger CON_{max} and smaller CON_{min} generally acquire more cells to embed secret messages.

In the third step, the pseudo random permutation secret message is embedded into the cells retained by Eq. 5 and Eq. 6. If an attempt is made to embed information bit 1, then all pixel values in the corresponding cell are reversed. If an information bit 0 is to be embedded, then all pixels values in a cell remain the same.

The maximum and minimum values in every 4×4 cell then in turn substitute the embedded bitmap to yield the embedded ODBTC image.

C. Decoding

Figure 3(b) shows the decoding flow chart and is introduced below. The key steps in decoding are the inverse halftoning and the second round halftoning. The cells corresponding to the information bit 1 can be located by comparing the bitmap in embedded ODBTC image with the second round dithered result of the bitmap (with high Hamming distance). However, the cells embedded with information 0 cannot be distinguished from the non-embeddable cells via the procedures described above. Hence, some side information such as the halftone screen, the parameters BW_{Th}, CON_{min}, and CON_{max}, are required to conduct the embeddable cell selection process as in the encoder to locate the embeddable cells. The details of the detection technique comprises of five steps as follows.

Step 1: The bitmap of the embedded ODBTC image is processed with Eq. 5 to reserve cells with similar numbers of black and white pixels.

Step 2: Searching for the cells associate to high frequency regions from the reserved cells retained by Eq. 6.

Step 3: These selected cells are then processed with inverse halftoning [12] and second round ordered dithering (using the same halftone screen as in the encoder). This step is an attempt to produce a temporary approximation version of the original dithered bitmap.

Step 4: The Hamming distance, denoted as H_d, between the "second round dithering cells" in step 3 and the corresponding cells in the original embedded ODBTC bitmap is calculated. If the H_d is greater or equal to half of the number of pixels in a cell, then this cell is confirmed to be embedded with an information bit "1". Otherwise, this cell represents an information bit "0".

Step 5: The original bitmap can be recovered by reversing the pixel values of the embedded cell that corresponds to the information bit "1" in the secret message.

4. EXPERIMENTAL RESULTS

In this section, the proposed technique is adopted to demonstrate the performance of the proposed algorithm. The three parameters BW_{Th}, CON_{min} and CON_{max} are adjusted to control the capacity. The secret messages of size 64×64 and 128×128 as shown in Figs. 5(a) and 5(b) are embedded into host images of size 512×512 and 1024×1024, respectively. Figure 6 shows the embedded bit numbers of the eight tested images with different parameters (CON_{min}=0; CON_{max}=30 and 34; BW_{Th}=12 and 16). The result indicates that using variables BW_{Th} of 12 and 16 yields average embedded bits of 1255 and 1663, which equal to capacity 0.48% and 0.63%, respectively.

Figures 7(a) shows the original Barbara image. Figures 7(b) shows the ODBTC results with objective PSNR value 39.95dB. Figures 7(c) shows the embedded ODBTC results with objective PSNR value 32.17dB. The corresponding decoded data is exhibited as white and black dots in Figs. 7(d), where the gray region contains no embedded information. These white and black dots associate to the positions of embedded cells as selected by the proposed embeddable cell selection technique. Notably, the correct decoding rates in all cases are 100%. Notably, the embeddable cell selection method is also applied to the other test images, and all the results achieve decoding rate 100%. Although the PSNR of these embedded ODBTC results are comparatively lower than that of the original ODBTC results, the original ODBTC results can be reversely obtained when needed in the receiver using the decoding rule introduced in Section 3C.

Table 2 shows the average quality of the embedded ODBTC results using eight test images. The tested blocks are of size 16x16 and 32x32, and the three parameters $BW_{th} = 148$, $CON_{max} = 237$ and $CON_{min} = 0$. Since the proposed ODBTC provides higher image quality than that of the BTC, the embedded ODBTC results achieve even higher quality than BTC when block size 32x32 is employed as shown in the third row of Table 2.

Some former approaches related to data hiding in BTC include Tu and Hsu [10] as well as Lin and Chang [11] providing the same secret message communication applications as the proposed method. However, unlike the proposed ODBTC-based approach, their approaches are based on the traditional BTC, and thus lower image quality is yielded. Moreover, both of the two former approaches cannot reconstruct the original BTC image. The proposed reversible technology outperforms to their methods when a high quality original ODBTC image is demanded in the decoder.

5. CONCLUSIONS

A novel data hiding technique is presented for embedding flexible amounts of data into an ODBTC image by tuning the parameters. The traditional BTC provides an efficient way to compress an image. However, the annoying false contour and blocking effect inherently existed in BTC image are the two major deficiencies. In this work, the ordered dithering is used to dither the quantized BTC image to avoid the above problems. Moreover, the LUT strategy is also used to significantly reduce the complexity. In data hiding application, the bitmap of an ODBTC image is adopted for message embedding. By dividing the bitmap into cells, the high-frequency region cells found with similar number of black and white pixels and low pixel connections are reserved due to the low sensitivity of the human eye to high frequencies. The secret data is then embedded into these cells. In the decoder, a blind decoding technique is applied with some side information, such as haftone screen, BW_{Th}, CON_{min}, and CON_{max}. The experimental results show that the average decoding rate of 100% can be achieved by suitably adjusting the parameters defined in this paper. Moreover, the lossless ODBTC image can be reconstructed in the decoder, which provides significant flexibility in image quality control.

REFERENCES

[1] D. Augot, J.-M. Boucqueau, J.-F. Delaigle, C. Fontaine, and E. Goray, "Secure Delivery of Images over Open Networks," Proceedings of the IEEE, vol. 87, no. 7, pp. 1251-1266, July 1999.

[2] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," IEEE Trans. Commun., vol. 27, pp. 1335-1342, Sept. 1979.

[3] O. R. Mitchell and E. J. Delp, "Multilevel graphics representation using block truncation coding," Proc. IEEE, vol. 68 , no. 7, pp. 868-873, July 1980.

[4] Q. Kanafani, A. Beghdadi and C. Fookes, "Segmentation-based image compression using BTC-VQ technique," IEEE Proc. of ISSPA 2003, vol.1, pp.113-116, Jul. 2003.

[5] S. Horbelt and J. Crowcroft, "A hybrid BTC/ADCT video codec simulation bench," Seventh International Workshop on Packet Video, Mar. 1996.

[6] M. D. Lema and O. R. Mitchell, "Absolute moment block truncation coding and its applications to color images," IEEE Trans. Commun., vol. 32, pp. 1148-1157, 1984.

[7] C. K. Yang and W. H. Tsai, "Improving Block Truncation Coding by Line and Edge Information and Adaptive Bit Plane Selection for Gray-Scale Image Compression," Pattern Recognition Letters, pp. 67-75, 1995.

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

[9] D. L. Lau and G. R. Arce, "Modern Digital Halftoning," New York: Marcel Dekker, q 2001.

[10] S. F. Tu and C. S. Hsu, "A BTC-based watermarking scheme for digital images," International Journal on Information & Security, pp. 216-228, 2004.

[11] M. H. Lin and C. C. Chang, "A novel information hiding scheme based on BTC," International Conference on Computer and Information Technology, 2004.

[12] P. C. Chang, C. S. Yu, and T. H. Lee, "Hybrid LMS-MMSE inverse halftoning technique," IEEE Trans. Image Processing, vol. 10, no. 1, pp. 95-103, Jan. 2001.

131	70	185	124	139	77	178	116
39	193	23	247	46	201	31	240
162	100	147	85	170	108	155	93
15	224	54	209	8	232	62	216
139	77	178	116	131	70	185	124
46	201	31	240	39	193	23	247
170	108	155	93	162	100	147	85
8	232	62	216	15	224	54	209

Fig. 1 Dispersed-dot dithering halftone screen (Bayer-5).

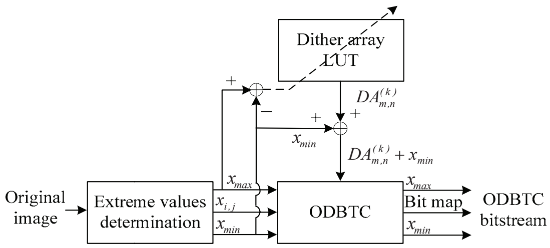
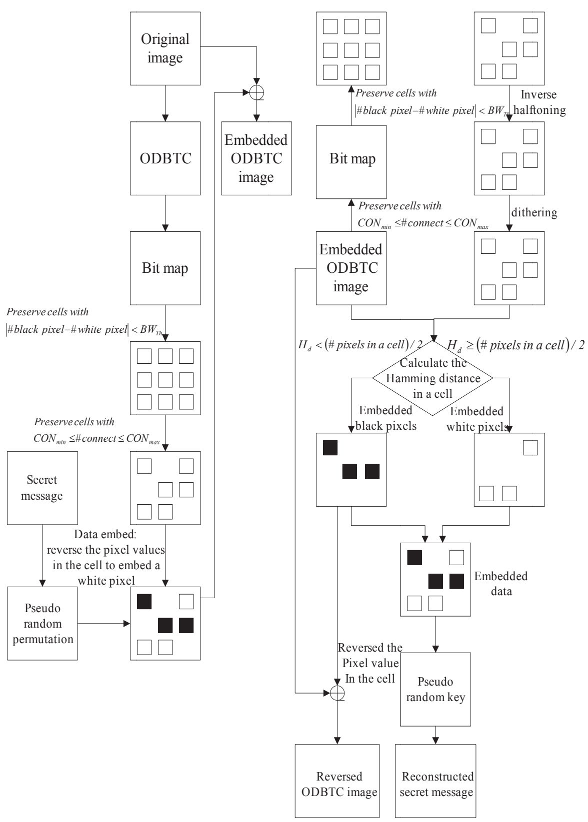


Fig. 2 Schematic diagram of the ODBTC.



(a) (b)

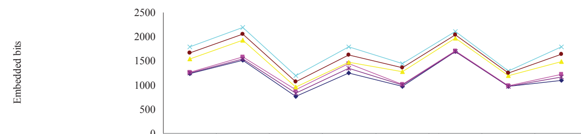
Fig. 3 Codec flow chart. (a) Encoder. (b) Decoder.

P_0	P_1	P_2
P_3	P	P_4
P_5	P_6	P_7

Fig. 4 3x3 connection evaluation covered region.

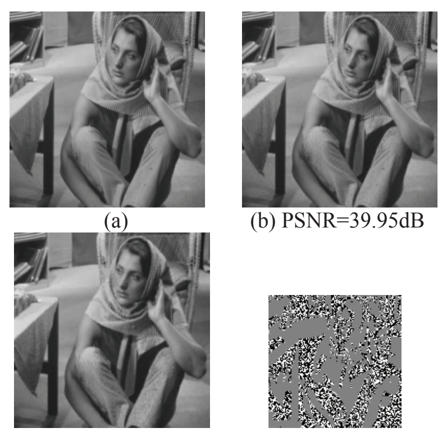
(a) (b)

Fig. 5 Embedded secret messages (a) message of size 64×64 (for host image of size 512×512) (b) message of size 128×128 (for host image of size 1024×1024).



	Lena	Boat	Peppers	Earth	Mandrill	Milk	Cameraman	Barbara
BWth=12,CONmax=30	1242	1513	758	1254	979	1693	970	1099
BWth=12,CONmax=34	1267	1581	903	1451	1020	1707	984	1225
BWth=16,CONmax=30	1537	1937	953	1473	1275	1969	1196	1487
BWth=16,CONmax=34	1789	2188	1189	1788	1449	2112	1297	1786
Average bits of BWth=12	1255	1547	831	1353	1000	1700	977	1162
Average bits of BWth=16	1663	2063	1071	1631	1362	2041	1247	1637

Fig. 6 The number of average encoded bits using eight testing images. (Average embedded bits=1255, which equals to capacity 0.48%, using $BW_{th} = 12$, and average embedded bits=1663, which equals to capacity 0.63%, using $BW_{th} = 16$)



(a) (b) PSNR=39.95dB

(c) PSNR=32.17dB (d) 7157 bits embedded

Fig. 7 (a) Original grayscale Barbara image (1024×1024). (b) ODBTC Barbara image (c) Embedded ODBTC Barbara image. (d) Decoded data from 7(b) of size 128×128. (Block size=8x8, $BW_{th} = 16, CON_{max} = 34, CON_{min} = 0$)

TABLE 1 COMPLEXITY COMPARISONS BETWEEN TRADITIONAL BTC AND THE PROPOSED ODBTC. (TOTAL OPERATIONS IN A BLOCK OF SIZE M×N)

	Addition/ Subtraction	Multiplication / Division	Square root
Traditional BTC	2×(M×N)+3	M×N+9	2
ODBTC	M×N	0	0

TABLE 2 EMBEDDED ODBTC QUALITY (AVERAGE BY EIGHT TEST IMAGES, ($BW_{th} = 148, CON_{max} = 237, CON_{min} = 0$)

Block size	BTC	ODBTC	Embedded ODBTC
16x16	33.84	36.57	29.82 (average capacity 0.34%)
32x32	30.03	35.06	32.77 (average capacity 0.00572%)