

Optimizing the Performance of Halftoning-Based Block Truncation Coding

Zi-Xin XU^{†*}, Yuk-Hee CHAN[†] and P.K. Daniel LUN[†]

[†]Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong

^{*}College of Electronic Engineering

Chengdu University of Information Technology, Chengdu, China

Email: ben.x.xu@connect.polyu.hk, enyhchan@polyu.edu.hk and enpklun@polyu.edu.hk

Abstract - Block Truncation Coding (BTC) is an effective lossy image coding technique that enjoys both high efficiency and low complexity especially when halftoning techniques are employed to shape the noise spectrum of its output. However, due to its block-based nature, blocking artifacts are commonly found in the coding outputs. Post-processing schemes are generally applied to soften the problem. Recently, a halftoning-based BTC algorithm was proposed to solve this problem by eliminating the cause of blocking artifacts. In this paper, through an optimization step, the performance of the algorithm is optimized in terms of a given objective measure. The idea can be adopted to work with other halftoning methods to optimize other measures for suiting different needs in different circumstances.

Key words: blocking effects, block truncation coding, halftoning, optimization

1. Introduction

Block truncation coding (BTC) was first introduced by Delp and Mitchell in 1979 [1]. In principle, it divides an image into non-overlapped blocks and then characterizes each block by two boundary values and a binary pattern that specifies the distribution of the two boundary values in a block. It provides a low-complexity yet effective means to achieve lossy image compression.

Since its introduction, various modifications to BTC have been proposed to improve its coding performance and various coding techniques such as vector quantization, entropy coding, predictive coding, discrete cosine transform, bit-plane coding and differential pulse code modulation [2-4] have been proposed to work with BTC to improve the rate-distortion performance. For the early stage of the evolution of BTC, one can refer to a detailed summary provided in [5].

Recent researches on BTC put their focus on how to enhance the visual quality of its outputs with halftoning techniques[6-10]. Halftoning is a powerful tool to quantize a multi-level image into a bi-level output image of desirable blue noise characteristics [11] and hence it can be used to produce the binary pattern required in BTC. With the help of the low-pass filtering effect of our human visual system(VHS), the high frequency blue noise can be effectively removed and the visual quality of the compressed image can be significantly improved.

Similar to all conventional BTC algorithms, halftoning-based BTC algorithms are block-based and hence blocking artifacts can generally be found in their outputs. Little effort has been devoted to address this issue and post-processing filtering seems to be the only remedial solution reported in literature [12,13] before the publication of [14]. Unlike the filtering approach that unavoidably distorts the visual quality of block boundary regions, the approach proposed in [14] (referred to as IDDBTC hereafter as it is an interpolation-involved dot-diffusion-based BTC algorithm) proactively eliminates the introduction of blocking artifacts during the coding process and hence no filtering is required. This paper presents a new halftoning-based BTC algorithm that modifies IDDBTC to optimize the visual quality of its encoding outputs at no extra cost in terms of bitrate.

The rest of this paper is organized as follows. Section 2 briefly reviews IDDBTC[14]. Section 3 presents an optimization scheme which can be exploited to further improve the performance of IDDBTC. Simulation results are presented in Section 4 for evaluation study and a conclusion is given in Section 5.

2. Review of IDDBTC

Without loss of generality, we assume that the input gray-level image I is of size $N \times N$, where N is a multiple of an integer s , and it is partitioned into non-overlapped blocks of size $s \times s$ each. Let $I(i, j)$ be the $(i, j)^{\text{th}}$ pixel of the input image I , $I_{m,n}$ be the $(m, n)^{\text{th}}$ block of I , and $I_{m,n}(k, l)$ be the $(k, l)^{\text{th}}$ pixel of block $I_{m,n}$. Then we have $I_{m,n}(k, l) = I(sm + k, sn + l)$.

The maximum and the minimum values of the pixels in block $I_{m,n}$ are, respectively, given as

$$I_{m,n}^{\max} = \max(\{I_{m,n}(k, l) \mid s > k, l \geq 0\}) \quad (1)$$

and

$$I_{m,n}^{\min} = \min(\{I_{m,n}(k, l) \mid s > k, l \geq 0\}) \quad (2)$$

where $\max(\cdot)$ and $\min(\cdot)$ are the maximum and the minimum operators, respectively.

In IDDBTC, an energy plane, which is denoted as X hereafter, is initialized as I and partitioned as in I . All the blocks are processed in parallel. For each block, pixels are processed sequentially according to a pre-defined order, which is specified by a $s \times s$ matrix called Class Matrix (CM), to produce the encoded image.

Consider the case that we are now processing $X_{m,n}(k,l) \equiv X(sm+k,sn+l)$, the $(sm+k,sn+l)^{th}$ element of X , to encode pixel $I(sm+k,sn+l)$ for $s>k,l \geq 0$ and $(m,n) \in \Lambda \equiv \{(p,q) | 0 \leq p,q < N/s\}$. A thresholding process is first applied to produce a binary output for the pixel as follows.

$$B_{m,n}(k,l) = \begin{cases} 0 & \text{if } X_{m,n}(k,l) < T_{m,n}(k,l) \\ 1 & \text{else} \end{cases} \quad (3)$$

where $T_{m,n}(k,l)$ is a threshold defined as

$$T_{m,n}(k,l) = \frac{\hat{I}^{max}(sm+k,sn+l) + \hat{I}^{min}(sm+k,sn+l)}{2} \quad (4)$$

for $(m,n) \in \Lambda$ and $s>k,l \geq 0$.

$\hat{I}^{max}(i,j)$ and $\hat{I}^{min}(i,j)$ are two non-stepwise bounding functions of (i,j) . They are obtained by interpolating $I_{m,n}^{max}$ and $I_{m,n}^{min}$ for $(m,n) \in \Lambda$, respectively, with a two-dimensional bilinear interpolation.

After $B_{m,n}(k,l)$ (i.e. $B(ms+k,ns+l)$) is determined, the corresponding pixel value of the encoded image Y can be determined as

$$Y(sm+k,sn+l) = \begin{cases} \hat{I}^{min}(sm+k,sn+l), & \text{if } B_{m,n}(k,l) = 0 \\ \hat{I}^{max}(sm+k,sn+l), & \text{else} \end{cases} \quad (5)$$

The difference between $Y(sm+k,sn+l)$ and $X(sm+k,sn+l)$ is considered as the coding error of the pixel and diffused to the neighbors of $X(sm+k,sn+l)$ as specified in eqn. (6) to update X before the next pixel is processed, where $h(i,j)$ is the $(i,j)^{th}$ coefficient of a non-causal diffusion filter and $K(x,y)$ is defined as

$$K(x,y) = \begin{cases} 0 & \text{if pixel } (x,y) \text{ has been processed} \\ 1 & \text{else} \end{cases} \quad (7)$$

Ω is the support of the diffusion filter. Coefficients $h(i,j)$ for $(i,j) \in \Omega$ are defined as elements in a matrix called Diffused Matrix (DM) whose center element corresponds to $h(0,0)$.

Pixels in a block are processed according to the order specified by the CM until all of them are processed. In

IDDBTC, the DM and CM suggested in [15] are used. At the end of encoding, $B(i,j)$ and $Y(i,j)$ for all pixel (i,j) form, respectively, a binary bitmap and the encoded version of I .

Though the encoded version of I is Y , the binary bit map B is stored instead of Y with the maximum and minimum values of the blocks as Y can be constructed with B , $I_{m,n}^{max}$ and $I_{m,n}^{min}$ as shown in eqn. (5).

3. HPSNR optimization

The human-visual peak signal-to-noise ratio (HPSNR) is an image quality assessment metric based on HVS [16]. When the pixel depth of the original image is 8 bit and the image size is $N \times N$, HPSNR is defined as

$$\text{HPSNR} = 10 \log_{10} \left(\frac{255^2}{\text{HMSE}} \right) \quad (8)$$

$$\text{where } \text{HMSE} = \frac{1}{N^2} \|G \otimes (Y - I)\|^2 \quad (9)$$

In eqn. (9), I and Y denote the original image and its encoded version respectively, G is a Gaussian filter that approximates the low-pass filtering nature of the HVS and \otimes symbolizes the convolution operation.

In IDDBTC, two continuous bounding functions $\hat{I}^{max}(i,j)$ and $\hat{I}^{min}(i,j)$ are derived by interpolating $I_{m,n}^{max}$'s and $I_{m,n}^{min}$'s respectively to eliminate the blocking artifacts.

Specifically, parameters $I_{m,n}^{max}$ and $I_{m,n}^{min}$ for block $I_{m,n}$ are computed with eqns. (1) and (2) respectively. As a matter of fact, the $\hat{I}^{max}(i,j)$ and $\hat{I}^{min}(i,j)$ interpolated with the $I_{m,n}^{max}$ and $I_{m,n}^{min}$ for $(m,n) \in \Lambda$ may not be optimal for producing an output Y of maximum HPSNR. In this paper, we suggest a better approach to derive two parameters, say $I_{m,n}^{ub}$ and $I_{m,n}^{lb}$, for each block such that, when $I_{m,n}^{ub}$ and $I_{m,n}^{lb}$ for $(m,n) \in \Lambda$ are respectively interpolated with bilinear interpolation to produce the bounding functions $\hat{I}^{max}(i,j)$ and $\hat{I}^{min}(i,j)$, the encoding image is optimal in terms of HPSNR.

HPSNR can be maximized by minimizing HMSE. In matrix formulation, the reconstructed Y obtained with our proposed algorithm is given as

$$\vec{y} = \mathbf{B}\mathbf{P}\vec{u} + \bar{\mathbf{B}}\mathbf{P}\vec{v} \quad (10)$$

where \vec{y} is a $N^2 \times 1$ column vector defined as the lexicographically-ordered image Y , \mathbf{B} is a $N^2 \times N^2$ diagonal matrix whose diagonal is the lexicographically-ordered binary

$$\begin{aligned} X(sm+k+i,sn+l+j) &= X(sm+k+i,sn+l+j) \\ &+ \frac{(X(sm+k,sn+l) - Y(sm+k,sn+l)) \cdot h(i,j) \cdot K(sm+k+i,sn+l+j)}{\sum_{(p,q) \in \Omega \setminus \{(0,0)\}} h(p,q) K(sm+k+p,sn+l+q)} \\ &\text{for } (i,j) \in \Omega \setminus \{(0,0)\} \text{ under the condition that } K(sm+k+i,sn+l+j) = 1 \end{aligned} \quad (6)$$

bitmap \mathbf{B} , \mathbf{I} is the identity matrix of size $N^2 \times N^2$, $\bar{\mathbf{B}} = \mathbf{I} - \mathbf{B}$, \vec{u} and \vec{v} are both $(N/s)^2 \times 1$ matrices and they are, respectively, the lexicographically-ordered two-dimensional arrays formed by $I_{m,n}^{ub}$ and $I_{m,n}^{lb}$ for $(m,n) \in \Lambda$, and \mathbf{P} is a $N^2 \times (N/s)^2$ matrix which is actually the matrix form of the two-dimensional bilinear interpolation operator that interpolates $I_{m,n}^{ub}$ and $I_{m,n}^{lb}$ respectively to produce the lexicographically-ordered $\hat{I}^{max}(i,j)$ and $\hat{I}^{min}(i,j)$.

Unlike IDDBTC in which the formulation of the bilinear interpolation depends on the nature (even/odd) of the value of s , the proposed method adopts a slightly different formulation that guarantees

$$\hat{I}^{max}(sm + \frac{s-1}{2}, sn + \frac{s-1}{2}) = I_{m,n}^{ub} \quad (11)$$

and

$$\hat{I}^{min}(sm + \frac{s-1}{2}, sn + \frac{s-1}{2}) = I_{m,n}^{lb} \quad (12)$$

for all m and n whatever the value of s is. In general, $\hat{I}^{max}(sm-1-k, sn-1-l)$, $\hat{I}^{max}(sm-1-k, sn+l)$, $\hat{I}^{max}(sm+k, sn-1-l)$ and $\hat{I}^{max}(sm+k, sn+l)$ for $k, l \in \{0, 1, \dots, [(s-1)/2]\}$ are computed as weighted sums of $I_{m,n}^{ub}$, $I_{m,n-1}^{ub}$, $I_{m-1,n}^{ub}$ and $I_{m-1,n-1}^{ub}$. Figure 1 shows an example in which $\hat{I}^{max}(sm, s(n-1)+3)$ is interpolated with $I_{m,n}^{ub}$, $I_{m,n-1}^{ub}$, $I_{m-1,n}^{ub}$ and $I_{m-1,n-1}^{ub}$ when $s=4$. $\hat{I}^{min}(sm, s(n-1)+3)$ is computed with the same approach based on $I_{m,n}^{lb}$, $I_{m,n-1}^{lb}$, $I_{m-1,n}^{lb}$ and $I_{m-1,n-1}^{lb}$ instead.

To look for the optimal $I_{m,n}^{ub}$ and $I_{m,n}^{lb}$ for $(m,n) \in \Lambda$, we define a HMSE-based cost function based on eqn. (9) as

$$J = \|\mathbf{H}(\vec{y} - \vec{x})\|^2 = \|\mathbf{H}(\mathbf{BP}\vec{u} + \bar{\mathbf{B}}\mathbf{P}\vec{v} - \vec{x})\|^2 \quad (13)$$

where \mathbf{H} is a $N^2 \times N^2$ matrix that represents Gaussian filter G and \vec{x} is a $N^2 \times 1$ column vector defined as the lexicographically-ordered image I , and then derive the optimal \vec{u} and \vec{v} that minimize J . In particular, by differentiating J with respect to \vec{u} and \vec{v} respectively, the optimal \vec{u} and \vec{v} can be obtained as

$$\vec{u} = (\mathbf{P}^t \mathbf{B}^t \mathbf{H}^t \mathbf{H} \mathbf{B} \mathbf{P})^{-1} \mathbf{P}^t \mathbf{B}^t \mathbf{H}^t \mathbf{H} (\vec{x} - \bar{\mathbf{B}} \mathbf{P} \vec{v}) \quad (14)$$

$$\vec{v} = (\mathbf{P}^t \bar{\mathbf{B}}^t \mathbf{H}^t \mathbf{H} \bar{\mathbf{B}} \mathbf{P})^{-1} \mathbf{P}^t \bar{\mathbf{B}}^t \mathbf{H}^t \mathbf{H} (\vec{x} - \mathbf{B} \mathbf{P} \vec{u}) \quad (15)$$

where \mathbf{M}^t and \mathbf{M}^{-1} denote the transpose and the inverse of a matrix \mathbf{M} respectively. If the inverse matrix of \mathbf{M} does not exist, then \mathbf{M}^{-1} is the pseudo inverse of \mathbf{M} . The optimized \vec{u} and \vec{v} are, respectively, the lexicographically-ordered two-dimensional arrays formed by $I_{m,n}^{ub}$ and $I_{m,n}^{lb}$ for $(m,n) \in \Lambda$.

Though eqns. (14) and (15) provide the formulations to compute the optimal $I_{m,n}^{ub}$ and $I_{m,n}^{lb}$ for block $I_{m,n}$, in practice it is impossible to derive them directly with eqns. (14) and (15) as the size of the involved matrices are huge and it is difficult to derive their inverses accurately. A gradient descent

method can be used instead to derive optimal $I_{m,n}^{ub}$ and $I_{m,n}^{lb}$ for $(m,n) \in \Lambda$ as follows.

Step 1: Initialize \vec{u} and \vec{v} as \vec{u}_0 and \vec{v}_0 which are respectively the lexicographically-ordered two-dimensional arrays formed by $I_{m,n}^{max}$ and $I_{m,n}^{min}$ for $(m,n) \in \Lambda$, and set index $k=0$.

Step 2: Carry out the following iterative procedures

$$\begin{aligned} \vec{u}_{k+1} &= \vec{u}_k - \frac{\beta}{2} \frac{\partial J}{\partial \vec{u}} \Big|_{\vec{u}=\vec{u}_k, \vec{v}=\vec{v}_k} \\ &= \vec{u}_k - \beta \mathbf{P}^t \mathbf{B}^t \mathbf{H}^t \mathbf{H} (\mathbf{B} \mathbf{P} \vec{u}_k + \bar{\mathbf{B}} \mathbf{P} \vec{v}_k - \vec{x}) \end{aligned} \quad (16)$$

$$\begin{aligned} \vec{v}_{k+1} &= \vec{v}_k - \frac{\beta}{2} \frac{\partial J}{\partial \vec{v}} \Big|_{\vec{u}=\vec{u}_k, \vec{v}=\vec{v}_k} \\ &= \vec{v}_k - \beta \mathbf{P}^t \bar{\mathbf{B}}^t \mathbf{H}^t \mathbf{H} (\mathbf{B} \mathbf{P} \vec{u}_k + \bar{\mathbf{B}} \mathbf{P} \vec{v}_k - \vec{x}) \end{aligned} \quad (17)$$

until termination criterion

$$\Delta J_{k+1} = \left| \frac{J|_{\vec{u}=\vec{u}_{k+1}, \vec{v}=\vec{v}_{k+1}} - J|_{\vec{u}=\vec{u}_k, \vec{v}=\vec{v}_k}}{J|_{\vec{u}=\vec{u}_{k+1}, \vec{v}=\vec{v}_{k+1}} - J|_{\vec{u}=\vec{u}_0, \vec{v}=\vec{v}_0}} \right| < \alpha \quad (18)$$

$$\text{or } J|_{\vec{u}=\vec{u}_{k+1}, \vec{v}=\vec{v}_{k+1}} \geq J|_{\vec{u}=\vec{u}_k, \vec{v}=\vec{v}_k} \quad (19)$$

is satisfied. Increase index k by 1 after each iteration.

Step 3: Extract the optimal $I_{m,n}^{ub}$ and $I_{m,n}^{lb}$ for $(m,n) \in \Lambda$ from the available \vec{u}_k and \vec{v}_k obtained so far.

While parameter α controls the termination criterion, parameter β is the step size which controls the rate of convergence. The smaller the value of β , the slower the convergence is but the more precise the optimal $I_{m,n}^{ub}$ and $I_{m,n}^{lb}$ can be obtained.

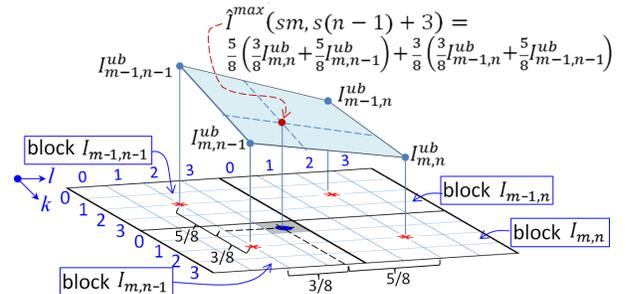


Figure 1. An example showing how $\hat{I}^{max}(sm, s(n-1)+3)$ is bi-linearly interpolated with $I_{m,n}^{ub}$, $I_{m,n-1}^{ub}$, $I_{m-1,n}^{ub}$ and $I_{m-1,n-1}^{ub}$ when $s=4$

4. Performance analysis

Simulations were carried out to evaluate the performance of the proposed algorithm. For comparison, the performance of BTC[1], EDBTC[6], ODBTC[7], DDBTC[8] and IDDBTC[14] was also evaluated. In the simulations, seven 512×512 grey-level testing images including Peppers,

Mandrill, Boat, Barbara, Lena, Airplane and Goldhill were encoded with different evaluated algorithms respectively.

Table 1 shows the performance of the evaluated algorithms in terms of various objective measures including HPSNR [16], Information Content Weighted PSNR (IW-PSNR) [17], Visual Information Fidelity (VIF) [18], Multi-Scale Structural Similarity Index (MS-SSIM) [19], Information Content Weight Structural Similarity Index (IW-SSIM) [17], and Gradient Magnitude Similarity Deviation (GMSD) [20]. The figures in the table are the averages of the evaluation results of all testing images under specific conditions. In the simulations, the results of the proposed algorithm were obtained with $\alpha=0.01$. Parameter β was selected to be 0.01 and 0.005 when the block size was 8×8 and 16×16 respectively. One can see that, whatever the block size is, the proposed algorithm performs better than other halftoning-based BTC algorithms in terms of almost all these objective measures.

Though BTC[1] can always perform better in terms of PSNR, IW-SSIM and MS-SSIM, the quality of its outputs is actually the lowest especially when the block size is large. The halftoning process in a halftoning-based BTC algorithm introduces high-frequency noise to its output, which lowers the PSNR remarkably. However, as HVS behaves as a low pass filter which is able to remove high frequency halftoning artifacts, conventional BTC is actually not preferable as compared with halftoning-based BTC algorithms. Figures 2 and 3 show some simulation results for subjective comparison. One can see that the proposed algorithm can effectively remove the blocking artifacts and preserve the spatial features in both smooth and textured regions of the original image. As shown in Figure 3, even when the block size is only 8×8 , blocking artifacts can be visible in the outputs of BTC, ODBTC, EDBTC and DDBTC but ours. One can easily observe this contrast in the regions around the windows and the field (the middle top) in Figures 3(b)-(f).

The encoding complexity of the proposed dot diffusion-based BTC algorithm is high as optimization is involved. However, the decoding process involves the interpolation of two planes (i.e. $\hat{I}^{max}(i, j)$ and $\hat{I}^{min}(i, j)$ for all i and j) and a selection process only. When block size $s=2^r$, where r is an

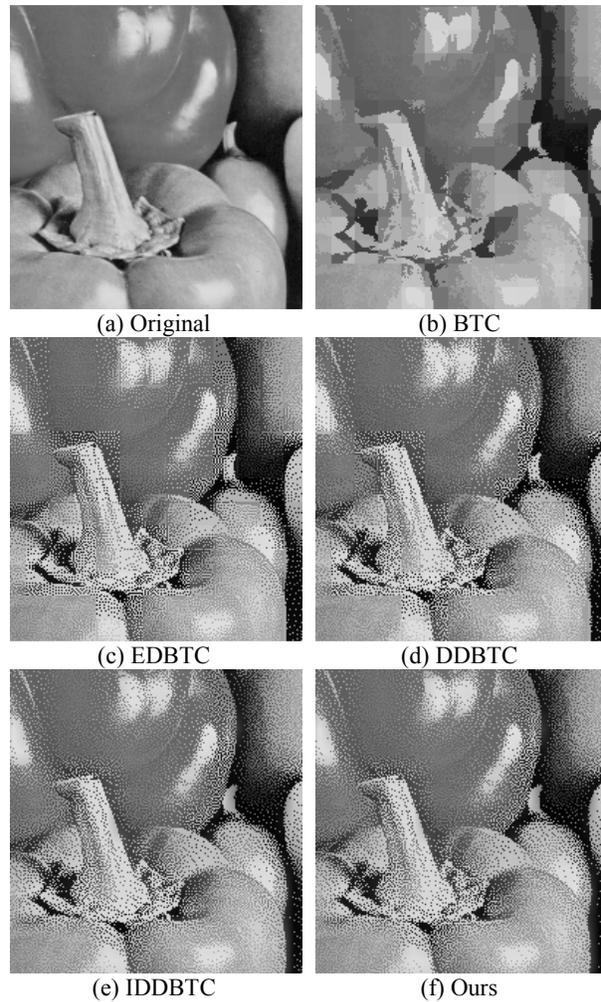


Figure 2. Enlarged regions of the coding results of test image *Peppers* using various halftoning-based BTC algorithms. (a) Original, (b) BTC[1], (c) EDBTC[6], (d) DDBTC[8], (e) IDDBTC[14] and (f) Ours. Block size is 16×16 .

Table 1 Coding performance of halftoning-based BTC algorithms

Block size	Algorithm	HPSNR	IW-PSNR	IW-SSIM	MS-SSIM	VIF	GMSD	PSNR
8x8	BTC [1]	39.9247	34.1837	0.9674	0.9777	0.4575	0.0517	28.1637
	ODBTC [7]	37.3828	29.7479	0.9149	0.9255	0.3712	0.0757	19.2997
	EDBTC [6]	40.5362	35.9514	0.9431	0.9394	0.4726	0.0729	20.1333
	DDBTC [8]	41.5265	34.3434	0.9424	0.9361	0.4747	0.0695	19.9211
	IDDBTC[14]	42.1071	35.6042	0.9429	0.9317	0.4872	0.0643	20.1964
	Ours	43.3236	41.6384	0.9456	0.9399	0.4999	0.0607	20.5164
16x16	BTC [1]	34.0370	27.0642	0.9238	0.9486	0.3302	0.1277	25.9355
	EDBTC [6]	38.3458	31.9609	0.9100	0.8901	0.3947	0.1153	16.7342
	DDBTC [8]	39.1629	33.6841	0.9030	0.8778	0.3814	0.1182	16.6073
	IDDBTC[14]	39.8976	34.5594	0.9041	0.8667	0.3952	0.1120	16.7824
	Ours	40.3908	36.3987	0.9072	0.8803	0.3965	0.1082	17.1052

integer, it takes at most six shift-additions and one binary selection to decode one pixel. Besides, parallel processing is easily achievable due to the block independent nature of the decoding algorithm. Its complexity is much lower than any existing image coding standards such as JPEG. When decoding complexity is a critical concern, BTC-based algorithms are obviously good options. If image quality is our next concern, the proposed algorithm will be an appropriate pick.

5. Conclusions

This paper presents a dot diffusion-based BTC algorithm that can improve the visual quality of encoded images by effectively eliminating the blocking artifacts and shaping the

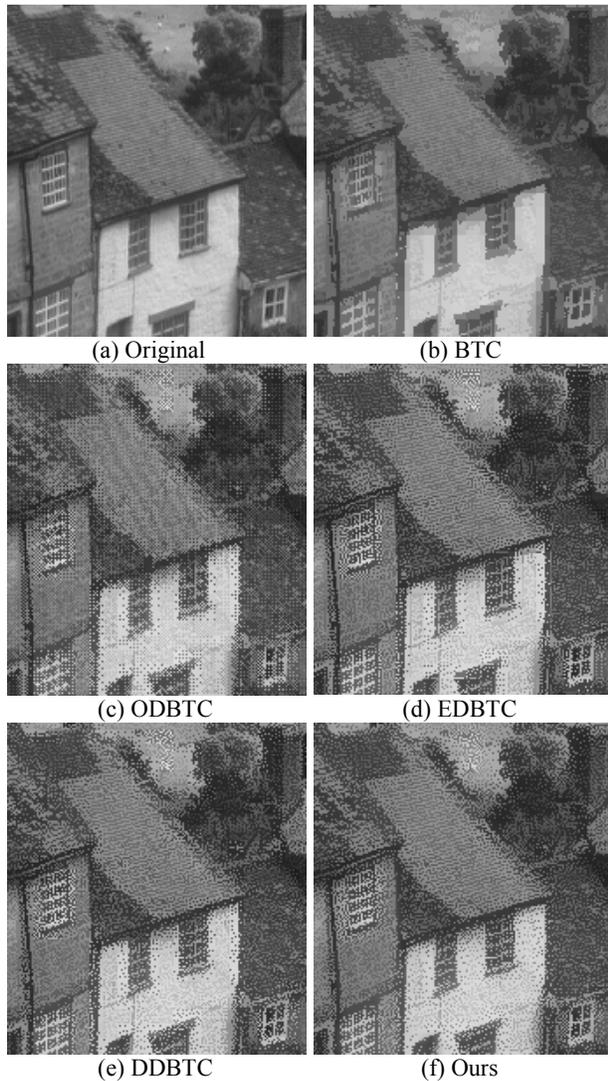


Figure 3. Enlarged regions of the coding results of test image *Goldhill* using various halftoning-based BTC algorithms. (a) Original, (b) BTC[1], (c) ODBTC[7] (d) EDBTC[6], (e) DDBTC [8], and (f) Ours. Block size is 8x8.

noise spectrum of an encoded image. The improvement is mainly achieved through an interpolation of bounding functions and a HPSNR optimization scheme.

As the interpolation and the optimization steps can be, respectively, considered as a pre-processing step and a post-processing step of the core part of the algorithm, one can develop other halftoning-based BTC algorithms by changing the halftoning method exploited in the core of the algorithm with some necessary modifications. In other words, the approach of the proposed algorithm forms a framework for one to develop halftoning-based BTC algorithms with different halftoning techniques as their cores.

Simulation results demonstrate that the BTC algorithms developed under this framework help to improve the visual quality of the encoded images for different block sizes.

The HPSNR optimization presented in Section 3 is based on the binary bitmap (i.e. B) derived with IDDBTC[14]. In IDDBTC, B is obtained with the conventional error diffusion technique based on a fixed scanning path (defined by the Class Matrix) and a spatial feature-independent diffusion filter (defined by the Diffused Matrix). In view of the noise model proposed in [21], the noise characteristics of B and hence the noise characteristics of our optimized output of IDDBTC are not ideal. To pursue an even better visual quality performance, a more advanced error diffusion technique such as [22] and [23] can be used to derive a better B for the optimization.

The algorithm presented in this paper is actually a barebone version. Similar to the case when the conventional BTC [1] evolves, by introducing other common coding techniques such as block classification, vector quantization, quadtree decomposition, discrete cosine transform, multi-bit quantization, entropy coding and block size adaption to the presented algorithm, the rate distortion performance can be further improved significantly.

Acknowledgment

This work was supported by a grant from The Hong Kong Polytechnic University (PolyU Grant G-YBM9) and a project from Ministry of Science and Technology of P.R.C. (Project No. 2017YFC1501701).

References

- [1] E.J. Delp and O.R. Mitchell, "Image Compression Using Block Truncation Coding," *Communications, IEEE Transactions on*, vol. 27, pp. 1335-1342, 1979.
- [2] E.J. Delp and O.R. Mitchell, "The use of block truncation coding in DPCM image coding," *IEEE Trans. on Signal Processing*, vol. 39, pp. 967-971, 1991.
- [3] K. Choi and S. Ko, "Improved differential pulse code modulation-block truncation coding method adopting two-level mean squared error near-optimal quantizers," *Optical Engineering*, vol. 50, pp. 047001, 2011.
- [4] K. Choi, "Bit plane modification for improving MSE-near optimal DPCM-based block truncation coding," *Digital Signal Processing*, vol. 23, pp. 1171-1180, 2013.
- [5] P. Fränti, O. Nevalainen and T. Kaukoranta, "Compression of Digital Images by Block Truncation

- Coding: A Survey," *The Computer Journal*, vol.37, Issue 4, pp.308-332, 1994
- [6] J.M. Guo, "Improved block truncation coding using modified error diffusion," *Electronics Letters*, vol.44, pp. 462-464, 2008.
- [7] J.M. Guo and M.F. Wu, "Improved Block Truncation Coding Based on the Void-and-Cluster Dithering Approach," *IEEE Trans. on Image Processing*, vol.18, pp. 211-213, 2009.
- [8] J.M. Guo and Y.F. Liu, "Improved Block Truncation Coding using Optimized Dot Diffusion," *IEEE Trans. on Image Processing*, vol.23, pp. 1269-1275, 2014.
- [9] J.M. Guo and C.C. Su, "Improved Block Truncation Coding Using Extreme Mean Value Scaling and Block-Based High Speed Direct Binary Search," *IEEE Signal Processing Letters*, 18(11), pp. 694-697, 2011.
- [10] J. M. Guo and C. Y. Lin, "Parallel and element-reduced error-diffused block truncation coding," *IEEE Trans. on Communications*, 58(5), pp.1667-1673, 2010.
- [11] R. Ulichney, *Digital Halftoning*. Cambridge, MA: MIT Press, 1987.
- [12] J.M.Guo, C.C.Su and H.J.Kim, "Blocking effect and halftoning impulse suppression in an ED/OD BTC image with optimized texture-dependent filter sets," Proc., Int. Conf. on Sys. Sci. and Engg., Macau, China, 2011, pp.593-596
- [13] Y. Wan, Y. Yang and Q.Q. Chen, "Multitone block truncation coding," *Electronics Letters*, 18th March 2010, 46(6), pp.414-416
- [14] Z.X. Xu and Y.H. Chan, "Eliminating blocking artifacts of halftoning-based block truncation coding," Proc. EUSIPCO'2016, 2016, Budapest, Hungary, pp.913-017.
- [15] J.M. Guo and Y.F. Liu, "Improved Dot Diffusion by Diffused Matrix and Class Matrix Co-Optimization," *IEEE Trans. on Image Processing*, vol.18, pp. 1804-1816, 2009.
- [16] J.M. Guo and Y.F. Liu, "Joint compression/watermarking scheme using majority-parity guidance and halftoning-based block truncation coding," *IEEE Trans.on Image Processing*, 19(8), pp. 2056-2069, 2010.
- [17] Z. Wang and Q. Li, "Information Content Weighting for Perceptual Image Quality Assessment," *IEEE Trans. on Image Processing*, vol.20, pp. 1185-1198, 2011.
- [18] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. on Image Processing*, vol.15, pp. 430-444, 2006.
- [19] Z. Wang, E. P. Simoncelli and A. C. Bovik, "Multi-scale structural similarity for image quality assessment," *IEEE Asilomar Conf. Signals, Systems and Computers*, 2003, pp.1398-1402.
- [20] W. Xue, L. Zhang, X. Moua and A.C. Bovik, Gradient Magnitude Similarity Deviation: A Highly Efficient Perceptual Image Quality Index," *IEEE Trans. on Image Processing*, vol.23, 2014, pp.684 – 695.
- [21] Y. H. Fung and Y. H. Chan, "Tone-dependent noise model for high-quality halftones," *Journal of Electronic Imaging*, 22(2), 023004-023004 (Apr 12, 2013). doi:10.1117/1.JEI.22.2.023004
- [22] Y. H. Fung and Y. H. Chan, "Tone dependent error diffusion based on an updated blue noise model," *Journal of Electronic Imaging*, 25(1), 013013 (Jan 25, 2016), doi: 10.1117/1.JEI.25.1.013013
- [23] Y. H. Chan and S. M. Cheung, "Feature-preserving multiscale error diffusion for digital halftoning," *Journal of Electronic Imaging*, 13(3), Jul 2004, pp.639-645