Eliminating blocking artifacts in halftoning-based block truncation coding

Zi-Xin XU and Yuk-Hee CHAN

Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong
Email: ben.x.xu@connect.polyu.hk and enyhchan@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. In this work, a real-time halftoning-based BTC algorithm is proposed to solve this problem by eliminating the cause of blocking artifacts while maintaining a complexity comparable to the best state-of-the-art halftoning-based BTC algorithm. Both objective and subjective comparisons demonstrate the visual quality improvement in its encoding outputs.**

***Keywords* - Block truncation coding; halftoning; blocking artifacts**

I. Introduction

Block truncation coding (BTC) is a lossy compression scheme 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. Although its coding gain is lower than modern compression methods like JPEG, its complexity is significantly lower to make it applicable in some fast scenes.

Since its introduction, various modifications to BTC have been proposed to improve its coding performance[2]. For examples, algorithms were proposed to encode the binary quantization results of BTC with various coding schemes such as vector quantization (VQ) [3], entropy coding[4], and bit-plane coding. Furthermore, block classification was employed to improve the image quality [5,6] and adaptive BTC algorithms allow variable block sizes to minimize the mean square error (MSE) of compressed images[7,8]. There were reports on how to optimize the output quality of a BTC algorithm by adjusting the boundary values and the quantization thresholds of a block [9]. The idea of BTC was extended to encode color images [10] and allow multi-bit quantization[11,12]. Hybrid coding schemes in which BTC works with discrete cosine transform [13] or differential pulse code modulation (DPCM) [14,15] were also developed to improve the rate-distortion performance. For the early stage of the evolution of BTC, one can refer to a detailed summary provided in [2].

This work was supported by Center for Multimedia Signal Processing and a grant from The Hong Kong Polytechnic University (PolyU Grant G-YBAN).

Recently, researches on BTC shift their focus towards how to enhance the visual quality of its outputs with halftoning techniques[16-20]. Halftoning is a powerful tool to quantize a multi-level image into a bi-level output image of desirable blue noise characteristics [21,22] 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, the high frequency blue noise can be effectively removed and the visual quality of the compressed image can be significantly improved.

The first halftoning-based BTC algorithm was proposed by Guo [16]. It is referred to as error-diffused BTC (EDBTC) as the halftoning process is achieved by error diffusion. EDBTC was later modified to support parallel processing in [20]. In [17], ordered dithering BTC (ODBTC) was introduced to reduce the coding complexity by replacing error diffusion with ordered dithering. The quality of the encoding outputs is sacrificed as a consequence. As dot diffusion is able to support parallel processing and provide good halftoning performance simultaneously, dot diffusion BTC (DDBTC) [18] was proposed to support low-cost real-time implementation recently. Like the original BTC proposed in [1], halftoning-based BTC algorithms can work with other coding techniques such as VQ and entropy coding to improve their rate-distortion performance [23].

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 [24,25]. Low pass filtering removes high frequency contents and hence it blurs the image. When removing the blocking artifacts in JPEG coding, one can only filter the block boundaries to minimize the blurring effect. However, the same idea does not work here. A halftoning-based BTC coding output contains strong high frequency components, which makes the contrast between unfiltered and filtered regions very outstanding as shown in Figure 1. As filtering unavoidably distorts the visual quality of block boundary regions, a solution that can proactively eliminate the introduction of blocking artifacts during the coding process would be attractive.

In this work, a novel algorithm is proposed to achieve this goal. The proposed algorithm is developed based on DDBTC [18] in which dot diffusion works with BTC to produce outputs of good image quality efficiently. By changing the halftoning method used in the proposed algorithm, it can be easily modified to produce various halftoning-based BTC algorithms to suit different needs in different circumstances.

The rest of this paper is organized as follows. Section 2 presents the proposed method. Its complexity is addressed in Section 3. Simulation results are provided in Section 4 for evaluation study and a conclusion is given in Section 5.

II. Proposed algorithm

Without loss of generality, the input gray-level image I is assumed 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)th pixel of the input image I, $I_{m,n}$ be the (m,n)th block of I, and $I_{m,n}(k,l)$ be the (k,l)th pixel of block $I_{m,n}$. Then we have $I_{m,n}(k,l)=I(sm+k,sn+l)$.

For each block, the maximum and the minimum values of the pixels are, respectively, given as

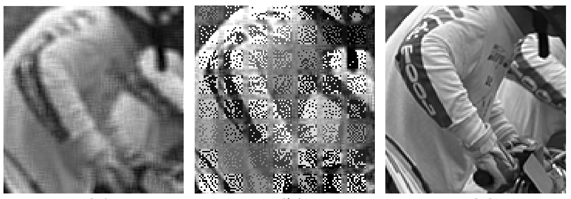
$$I_{m,n}^{\max} = \max(\{ I_{m,n}(k,l) \mid s > k,l \geq 0\}) \quad (1)$$

and

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

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

In our algorithm, 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. The element values of CM are all different and they are integer values bounded in $[0,\ s^2-1]$. Each pixel in a block is positionally mapped to an element of the CM. The pixel associated with a smaller CM element value is processed earlier.


(a) (b) (c)
Figure 1. Post-filtered DDBTC coding results obtained with a 3×3 average filter: (a) filter the whole image, (b) filter block boundaries only, and (c) the original.

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.

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

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

$$T_{m,n}(k,l)=\frac{\widehat{I}^{\max}(sm+k,sn+l)+\widehat{I}^{\min}(sm+k,sn+l)}{2}$$
$$\text{for } (m,n)\in\Lambda \text{ and } s>k,l\geq 0. \quad (4)$$

$\widehat{I}^{\max}(i,j)$ and $\widehat{I}^{\min}(i,j)$ are two non-stepwise bounding functions of (i,j). They are obtained by

$$\widehat{I}^{\max}(sm+\frac{s}{2}-1+k,sn+\frac{s}{2}-1+l)=\left(1-\frac{k}{s}\right)\left(1-\frac{l}{s}\right)I_{m,n}^{\max}+\left(1-\frac{k}{s}\right)\left(\frac{l}{s}\right)I_{m,n+1}^{\max}+\left(\frac{k}{s}\right)\left(1-\frac{l}{s}\right)I_{m+1,n}^{\max}+\left(\frac{k}{s}\right)\left(\frac{l}{s}\right)I_{m+1,n+1}^{\max}$$
$$\widehat{I}^{\min}(sm+\frac{s}{2}-1+k,sn+\frac{s}{2}-1+l)=\left(1-\frac{k}{s}\right)\left(1-\frac{l}{s}\right)I_{m,n}^{\min}+\left(1-\frac{k}{s}\right)\left(\frac{l}{s}\right)I_{m,n+1}^{\min}+\left(\frac{k}{s}\right)\left(1-\frac{l}{s}\right)I_{m+1,n}^{\min}+\left(\frac{k}{s}\right)\left(\frac{l}{s}\right)I_{m+1,n+1}^{\min}$$
$$\text{for } s>k,l\geqq 0 \quad \text{if } s \text{ is even} \quad (5)$$

$$\widehat{I}^{\max}(sm+\frac{s-1}{2}+k,sn+\frac{s-1}{2}+l)=\left(1-\frac{k}{s}\right)\left(1-\frac{l}{s}\right)I_{m,n}^{\max}+\left(1-\frac{k}{s}\right)\left(\frac{l}{s}\right)I_{m,n+1}^{\max}+\left(\frac{k}{s}\right)\left(1-\frac{l}{s}\right)I_{m+1,n}^{\max}+\left(\frac{k}{s}\right)\left(\frac{l}{s}\right)I_{m+1,n+1}^{\max}$$
$$\widehat{I}^{\min}(sm+\frac{s-1}{2}+k,sn+\frac{s-1}{2}+l)=\left(1-\frac{k}{s}\right)\left(1-\frac{l}{s}\right)I_{m,n}^{\min}+\left(1-\frac{k}{s}\right)\left(\frac{l}{s}\right)I_{m,n+1}^{\min}+\left(\frac{k}{s}\right)\left(1-\frac{l}{s}\right)I_{m+1,n}^{\min}+\left(\frac{k}{s}\right)\left(\frac{l}{s}\right)I_{m+1,n+1}^{\min}$$
$$\text{for } s>k,l\geqq 0 \quad \text{if } s \text{ is odd} \quad (6)$$

$$X(sm+k+i,sn+l+j)=X(sm+k+i,sn+l+j)$$
$$+\frac{\big(X(sm+k,sn+l)-Y(sm+k,sn+l)\big)\cdot h(i,j)\cdot K(sm+k+i,sn+l+j)}{\sum_{(p,q)\in\Omega\setminus\{(0,0)\}} h(p,q)K(sm+k+p,sn+l+q)}$$
$$\text{for } (i,j)\in\Omega\setminus\{(0,0)\} \text{ under the condition that } K(sm+k+i,sn+l+j)=1 \quad (8)$$

interpolating $I_{m,n}^{\max}$ and $I_{m,n}^{\min}$ for $(m,n)\in\Lambda$, respectively, with a two-dimensional bilinear interpolation as shown in eqns. (5) and (6).

Note that $I_{m,n}^{\max}$ and $I_{m,n}^{\min}$ are only defined for $(m,n)\in\Lambda$. To interpolate $\hat{I}^{\max}(i,j)$ for the boundary pixels of the image with eqns. (5) and (6), one may need $I_{-1,n}^{\max}$, $I_{m,-1}^{\max}$, $I_{N/s,n}^{\max}$ and $I_{m,N/s}^{\max}$ for $-1\le m,n\le N/s$. They can be duplicated as $I_{-1,n}^{\max}=I_{0,n}^{\max}$, $I_{m,-1}^{\max}=I_{m,0}^{\max}$, $I_{N/s,n}^{\max}=I_{N/s-1,n}^{\max}$ and $I_{m,N/s}^{\max}=I_{m,N/s-1}^{\max}$ for $-1\le m,n\le N/s$ to make eqns. (5) and (6) function properly. Similar arrangement is made to handle the interpolation of $\hat{I}^{\min}(i,j)$.

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 then be determined as

$$Y(sm+k,sn+l)=\begin{cases}\hat{I}^{\min}(sm+k,sn+l) & \textit{if } B_{m,n}(k,l)=0\\ \hat{I}^{\max}(sm+k,sn+l) & \textit{else}\end{cases} \quad (7)$$

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 shown in eqn. (8) 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

$$K(x,y)=\begin{cases}0 & \text{if pixel } (x,y) \text{ has been processed}\\ 1 & \text{else}\end{cases} \quad (9)$$

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. 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. (7). The compression ratio achieved by the proposed algorithm is the same as that of DDBTC.

Theoretically, bounding functions $\hat{I}^{\max}(i,j)$ and $\hat{I}^{\min}(i,j)$ can also be obtained via a higher-order interpolation scheme. However, our simulation results show that it does not guarantee a coding output of better quality. A bilinear interpolation scheme is hence suggested in our proposal.

III. Complexity

BTC is created as a low-cost simple method to compress images. Complexity should hence be a major concern when one tries to modify BTC to improve its output quality.

The realization of the two-dimensional bilinear interpolation defined in eqns.(5) and (6) can be decomposed into two phases each of which involves one-dimensional interpolation only. In practical applications, for better memory management and more efficient implementation of BTC, block size is generally selected to be $s=2^r$, where r is an integer. In such a case, the one-dimensional interpolation along a row or a column can be realized with a limited number of shift-additions and hence no floating point operation is required. Besides, most of the shift-additions are independent such that they can be realized in parallel. In general, when all independent operations can be carried out in parallel, it takes $2\log_2 s$ time slots to complete the interpolation of $\hat{I}^{\max}(i,j)$ and $\hat{I}^{\min}(i,j)$ for all (i,j), where one time slot corresponds to the time required to complete an integer shift-addition.

By counting each shift-addition or comparison as an addition, DDBTC takes 11.5 additions and 2 multiplications to encode one pixel while the proposed algorithm takes 13.5 additions and 2 multiplications. Their complexity is much lower than JPEG and real-time realization can easily be achieved. Our simulation results show that, as compared with DDBTC, on average the proposed algorithm takes, respectively, around 6% and 15% more time to encode an image when s=8 and s=16. Note that DDBTC can achieve a frame rate up to 61236.99 fps (frames per second) [18]. Both methods are efficient enough to be applied in areas which are sensitive to computation complexity.

IV. Performance analysis

Simulations were carried out to evaluate the performance of the proposed algorithm in terms of output quality. The luminance planes of the 24 images available in the Kodak image database [26] were used as the test images in our evaluation study. For comparison, the performance of other halftoning-based BTC algorithms including EDBTC[16], ODBTC[17] and DDBTC[18] was also evaluated. When realizing DDBTC and the proposed algorithm in our simulation, the CM and DM pairs that are optimized in [18] for different block sizes were used. Figure 2 shows some regions of the encoded test images for subjective comparison. In Figure 2, rows 2 and 4 are, respectively, enlarged versions of rows 1 and 3. The enlarged regions are for easier comparison.

As shown in Figure 2, noticeable blocking artifacts can be found in ODBTC's, EDBTC's and DDBTC's outputs while the proposed algorithm can effectively remove the blocking artifacts appeared in their outputs. Spatial features of the test images can also be well preserved in both smooth and textured regions of the outputs of the proposed algorithm.

The performance of the halftoning-based BTC algorithms is also evaluated in terms of various objective quality measures including PSNR, SSIM, HPSNR [27], Information Content Weighted PSNR (IW-PSNR) [28], Multi-Scale Structural Similarity Index (MS-SSIM) [29], and Information Content Weight Structural Similarity Index (IW-SSIM) [28]. Table 1 shows the averages of the evaluation results of all test images when different block sizes are used. The best results are bolded. The proposed algorithm always provides the best results in terms of HPSNR, IW-PSNR and PSNR. Though EDBTC generally performs better in terms of SSIM-based measures, the difference between EDBTC and the proposed algorithm is little and the subjective visual quality of EDBTC's output is actually lower than that of the proposed algorithm as shown in Figure 2. Besides, EDBTC does not support parallel processing while the proposed algorithm does, which makes EDBTC inferior when coding speed is a major concern. The current proposed algorithm provides a good balance between efficiency and quality.

V. 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 of an encoded image. The improvement is conceptually achieved through an interpolation of bounding functions that allows one to remove the sudden contrast change across the block boundaries. This problem-solving approach forms a framework for one to develop halftoning-based BTC algorithms with different halftoning techniques (e.g. [30-35]) as their cores.

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.

Block size	Alg.	HPSNR	IW-PSNR	PSNR	SSIM	IW-SSIM	MS-SSIM
8x8	ODBTC	36.355	28.172	19.479	0.886	0.918	0.922
	EDBTC	38.880	32.220	20.163	0.902	**0.946**	0.935
	DDBTC	40.617	32.963	19.933	0.899	0.941	0.935
	ours	**41.323**	**34.757**	**20.246**	**0.904**	0.943	**0.937**
16x16	ODBTC	35.032	26.892	16.371	0.797	0.875	0.870
	EDBTC	37.689	29.671	16.768	**0.817**	**0.916**	**0.885**
	DDBTC	38.481	31.664	16.631	0.794	0.904	0.878
	ours	**39.940**	**34.244**	**16.821**	0.795	0.908	0.877

Table 1 Coding performance of halftoning-based BTC algorithms

REFERENCES

[1] E.J.Delp and O.R.Mitchell, "Image compress-ion

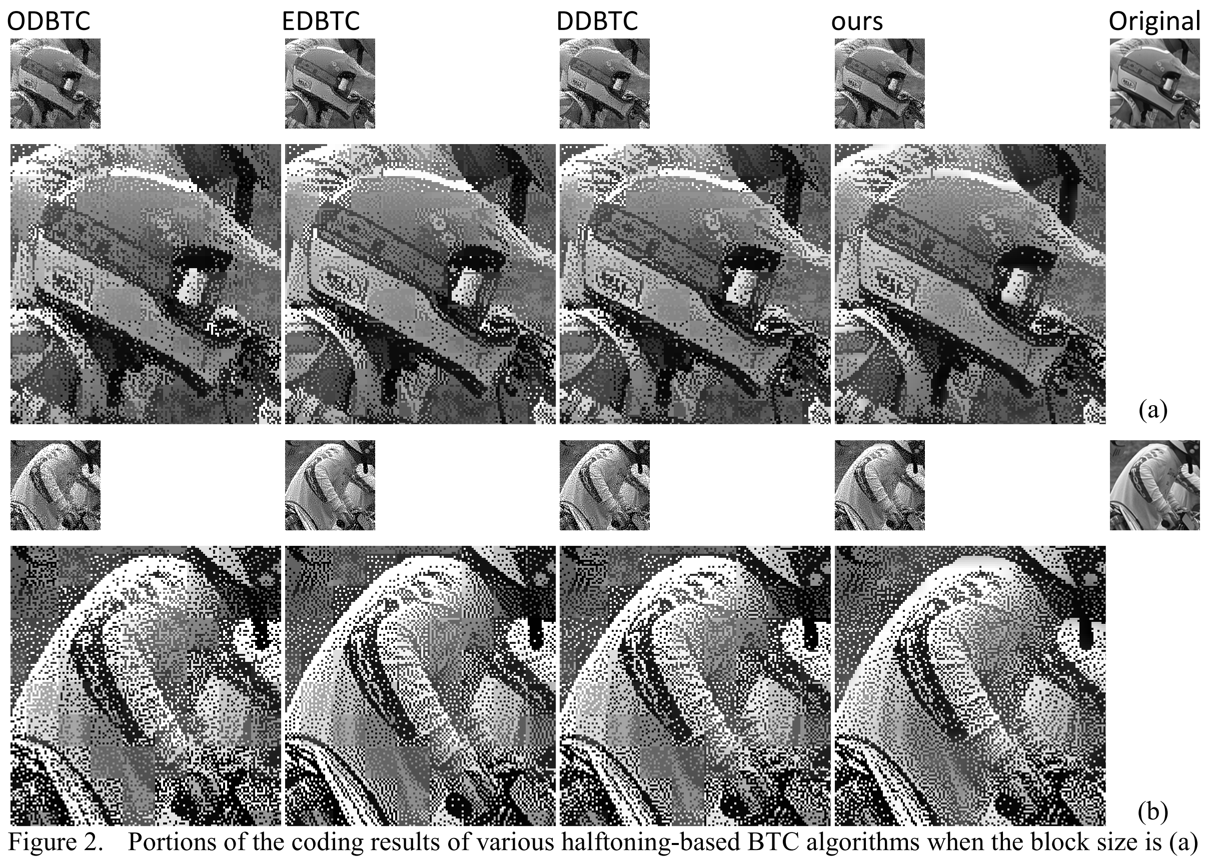


Figure 2. Portions of the coding results of various halftoning-based BTC algorithms when the block size is (a) 8×8 and (b) 16×16

using block truncation coding," *IEEE Trans. on Communications,*, vol. 27, pp. 1335-1342, 1979.

[2] 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

[3] B. Zeng and Y. Neuvo, "Interpolative BTC image coding with vector quantization," *IEEE Trans. on Communications,* vol. 41, pp. 1436-1438, 1993.

[4] P. Fränti , Olli Nevalainen, "Block Truncation Coding with Entropy Coding ," *IEEE Trans. on Communications*, Vol.43, pp.1677-1685, 1995

[5] Q.Kanafani, A.Beghdadi and C.Fookes, "Segmentation-based image compression using BTC-VQ technique," in, Proceedings. 7th Int. Symposium on Signal Processing and its Applications, 2003, vol.1, pp. 113-116.

[6] S.I. Olsen, "block truncation and planar image coding", *Pattern Recognition Letters*, vol.21, Issue 13-14, pp.1141-1148, 2000

[7] L. Hui, "An adaptive block truncation coding algorithm for image compression," in Proc., ICASSP'90, 1990, vol.4, pp. 2233-2236.

[8] S.Vimala, P.Uma and B.Abidha, "Article: Improved Adaptive Block Truncation Coding for Image Compression," *International Journal of Computer Applications*, vol. 19, pp.1-5, 2011.

[9] J. Mathews, M. S. Nair and L. Jo, "Modified BTC Algorithm for Gray Scale Images using max-min Quantizer", Proceedings, iMac4s 2013 (India), IEEE Computer Society Press, Mar 2013, pp.377-382.

[10] G. Campbell, T. A. DeFanti, J. Frederiksen, S. A. Joyce and L. A. Leske, "Two bit/pixel full color encoding," SIGGRAPH Comput.Graph., vol. 20, pp. 215-223, 1986.

[11] Y. Wu, and D.C. Coll, "Multilevel block truncation coding using a minimax error criterion for high-fidelity compression of digital images," *IEEE Trans. on Communications*, pp.1179-1191, 1993

[12] K.W. Chan, K.L. Chan, "Optimisation of multi-level block truncation coding," *Signal Processing: Image Communication*, vol. 16, Issue 5, pp.445–459, 2001

[13] Y.Wu and D.C. Coll, "BTC-VQ-DCT hybrid coding of digital images," *IEEE Trans. on Communications*, vol.39, pp.1283-1287, 1991

[14] 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.

[15] K. Choi, "Bit plane modification for improving MSE-near optimal DPCM-based block truncation coding," *Digital Signal Processing*, vol. 23, pp. 1171-1180, 2013.

[16] J. M. Guo, "Improved block truncation coding using modified error diffusion," *Electronics Letters*, vol. 44, pp. 462-464, 2008.

[17] 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.

[18] 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.

[19] 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*, pp. 694-697, 2011.

[20] J. M. Guo and C. Y. Lin, "Parallel and element-reduced error-diffused block truncation coding," *IEEE Trans. on Communications*, vol.58, no.5, pp.1667-1673, 2010.

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

[22] Y.H. Fung and Y.H. Chan, "Tone-dependent noise model for high-quality halftones," *Journal of Electronic Imaging*, 22 (2), 023004, 2013. doi:10.1117/1.JEI.22.2. 023004

[23] J.M. Guo, "High efficiency ordered dither block truncation coding with dither array LUT and its scalable coding application," *Digital Signal Processing*, vol.20, Issue 1, pp.97–110, 2010

[24] J.M. Guo, C.C.Su, H.J.Kim, "Blocking effect and halftoning impulse suppression in an ED/ OD BTC image with optimized texture-depend-ent filter sets," Proc. Int. Conf. on Sys. Sci. and Engg., Macau, China, 2011, pp.593-596

[25] Y. Wan, Y. Yang and Q.Q. Chen, "Multitone block truncation coding," *Electronics Letters*, 18th March 2010, vol.46, no.6, pp.414-416

[26] Kodak true color image suite [Online]. Available: http://r0k.us/graphics/kodak/

[27] 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,* pp. 2056-2069, 2010.

[28] Z. Wang and Q. Li, "Information Content Weighting for Perceptual Image Quality Assessment," *IEEE Trans. on Image Processing*, pp. 1185-1198, 2011.

[29] 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.

[30] 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

[31] Y.H. Fung and Y.H. Chan, "Green Noise Digital Halftoning with Multiscale Error Diffusion," *IEEE Trans. on Image Processing*, Jul 2010, pp.1808-1823.

[32] K.C. Lui, Y.H. Fung and Y.H. Chan, "A Low-complexity High-performance Multiscale Error Diffusion Technique for Digital Halftoning," *Journal of Electronic Imaging*, Vol.16, Issue 1, 013010, 12 pages, Jan-Mar 2007

[33] Y.H. Chan and Sin-Ming Cheung, "Feature-preserving multiscale error diffusion for digital halftoning," *Journal of Electronic Imaging*, Vol.13, No.3, Jul 2004, pp.639-645

[34] Y.H.Chan and S.M. Cheung, "A row-oriented multiscale error diffusion technique for digital halftoning," *IEE Electronics Letters*, Vol.37, No.10, 10 May 2001, pp.626-627.

[35] Y.H.Chan, "A modified multiscale error diffusion technique for digital halftoning", *IEEE Signal Processing Letters*, Vol.5, No.11, Nov 1998, pp.277-280.