Adaptive Bit Plane Quadtree-based Block Truncation Coding for Image Compression

Shenda Li*, Jin Wang, Qing Zhu
Faculty of Information Technology
Beijing University of Technology, Beijing, China
Beijing Key Laboratory of Internet Culture and Digital Dissemination Research
* lsd1994@163.com

ABSTRACT

Block truncation coding (BTC) is a fast image compression technique applied in spatial domain. Traditional BTC and its variants mainly focus on reducing computational complexity for low bit rate compression, at the cost of lower quality of decoded images, especially for images with rich texture. To solve this problem, in this paper, a quadtree-based block truncation coding algorithm combined with adaptive bit plane transmission is proposed. First, the direction of edge in each block is detected using Sobel operator. For the block with minimal size, adaptive bit plane is utilized to optimize the BTC, which depends on its MSE loss encoded by absolute moment block truncation coding (AMBTC). Extensive experimental results show that our method gains 0.85 dB PSNR on average compare to some other state-of-the-art BTC variants. So it is desirable for real time image compression applications.

Keywords: image compression, block truncation coding, hierarchical decomposition, adaptive bit plane.

1. INTRODUCTION

Block truncation coding (BTC) is an efficient image compression technique. Delp and Mitchell first introduced the concept of BTC in 1979 [1]. This is the original BTC, which partitions an image into non-overlapping blocks and processes them individually. For each block, the mean value and the standard deviation of the block is first calculated. Then, two quantization levels are preformed based on these parameters. Besides, it uses mean value for thresholding, pixels with values equal to or greater than the threshold are quantized to high quantization level, while others will be quantized to low quantization level. This approach is relatively fast, requiring only a little encoding bits and computational complexity.

After that, several modifications and improvements of the basic BTC were proposed [2]. In 1984, Lema and Mitchell proposed absolute moment block truncation coding (AMBTC) [3]. In AMBTC, instead of using standard deviation, they use high mean or low mean to quantize, which are calculated by averaging the pixels whose value is greater or smaller than block mean. This method is proved to be more efficient than the basic BTC, because square root and multiplication operations are omitted. Healy and Mitchell proposed joint quantization [4], using fewer bits to encode the quantization data with slightly sacrificing the image quality. There are some schemes to reduce the bit plane transmission. Several approaches use visual patterns (VQ) to quantize the bit plane [5,6]. Their main idea is to predefine the codebook to represent those blocks which have the highest probability to occur. This technique significantly reduces the bit rate while maintains the image quality at the same time. In recent years, Chandravadhana proposed least mean square error-based BTC [7]. Mean square error is calculated iteratively for each block to ensure optimal visual quality. Xu proposed halftoning-based BTC to eliminate blocking artifacts [8]. Their method can maintain the computational complexity and get better result. Dega proposed multi-partition BTC [9], harmony search algorithm is used to create multi-quantization level. This method is better than AMBTC. Guo proposed dot diffusion BTC [10], they distributed the quantization error to neighbor blocks, which decreases the average MSE in each block.

In order to further reduce the bit rate, variable block size of BTC is considered [11,12]. The original image is first partitioned into non-overlapping blocks. Then, hierarchical decomposition technique is used. If the current block satisfies the threshold criterions, it can be replaced by its mean value, otherwise it will be subdivided into four subblocks. Each subblock is encoded using the same way until the minimal block size is reached. With a series of indicator bits sent to decoder, it can reconstruct the image based on the quadtree structure. Combined with quadtree segmentation and

advanced BTC techniques, this method achieves good image quality while requiring low computational complexity at a low bit rate [13]. In recent years, Daga proposed k-d tree-segmented BTC [14], they utilized variable size of blocks with different row and column to further reduce the bit rate, while slightly decreasing the image quality. Chen proposed quadtree-based BTC using two adjustable thresholds to balance the high image quality and low bit rate [15]. This method is flexible, and able to fit in many situations. Yang combined error diffusion and quadtree-based BTC [16]. Their method is quite fast, but not suitable for high texture image. Although the hierarchical decomposition seems to outperform other BTC variants, it also has some disgusting drawbacks too. The blocking effect inherent in BTC cannot be eliminated, especially at low bit rate. Besides, current quadtree-based BTC is not able to compress high texture image, which may cause severe degradation in reconstructed image.

In this paper, a quadtree-based block truncation coding algorithm combined adaptive bit plane transmission is proposed. We calculate the variance of current block, deciding if it should be subdivided or not. We used Sobel operator to detect the direction of edge in each block. According to absolute response of Sobel operator, the current block is further subdivided horizontally or vertically. For the minimal block size, we try to encode it by AMBTC and decode it using two quantization levels immediately. If the MSE loss originated from the encoding-decoding procedure is unbearable, we quantize the bit plane by 2 bit per pixel to enhance the performance.

The rest of the paper is organized as follows: Section 2 illustrates some basic concepts and techniques used in this paper. Section 3 introduces the proposed method in details, and gives some examples to demonstrate. Experimental results are shown in section 4, it compares the proposed scheme with several state-of-the-art techniques. Finally, some discussions and conclusions are given in section 5.

2. BACKGROUND

2.1 AMBTC

The goal of AMBTC is to preserve the mean and the first absolute central moment of image blocks. It first partitions the image into 16 × 16 non-overlapping blocks. For each block, the block mean $\bar{x}$ is first calculated by averaging the k pixel values in the current block. Then, using the mean value for thresholding, two quantization levels are calculated by averaging the pixels whose values are greater or smaller than block mean (formula 1). The two quantization levels are then sent to the decoder using 8+8 bits.

$$a = \frac{1}{k-q}\sum_{x_i<\bar{x}} x_i \,, b = \frac{1}{q}\sum_{x_i\geq\bar{x}} x_i \tag{1}$$

Here q denotes the number of pixels greater or equal to threshold. For each pixel in the block, there is a corresponding bit in the bit plane that records which quantization level is used to encode this pixel. Thus, a trio consisting of two quantization levels and a bit plane, 32 bits per block, is sent to the decoder. AMBTC needs very low computation ability and memory space, while the reconstructed image is acceptable for most applications. However, the bit rate is fixed at 2 bits per pixel when encoding, which prevents it from being an excellent image codec.

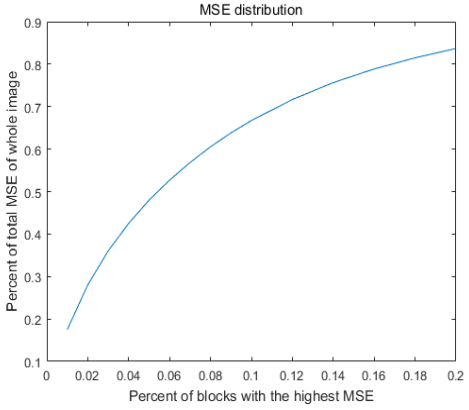


Figure 1. MSE distribution of Lena

Typically, during the encoding-decoding procedure, the majority of MSE loss is caused by only a few blocks, as shown in Fig.1. We encode grayscale image "Lena" by AMBCT and calculate the MSE between original image and decoded image. The result shows that 10% of all the blocks with the highest MSE reach approximately 65% of the total MSE of the whole image. It means that high variance blocks are the leading cause of image quality degradation, which are not suitable to be encoded by ABMTC.

2.2 Quadtree-based BTC

The quadtree segmentation technique is a hierarchical decomposition technique that partitions image into variable sized blocks based on the quadtree structure (Fig.2). It first determines the maximum and minimum block size and processes block by block. For each block, if the current block satisfies the threshold criterions, that means the current block is inactive, so it can be replaced by its mean value. Otherwise, it will be subdivided into four subblocks. Then each subblock is processed recursively until threshold criterions is satisfied or the minimal block size is reached. The most important part in quadtree-based BTC is how to encode minimal active blocks. One can use all kinds of BTC variants, even some new schemes specially designed to deal with active block. All of these methods include encoding one indicator bit, quantization data and bit plane information. In decoding phase, the indicator bit is first decoded, deciding whether the current block is active or not. Using this information, the decoder knows how many bits should be read from the bit stream and their exact meaning. This procedure is done block by block until the whole image has been reconstructed.

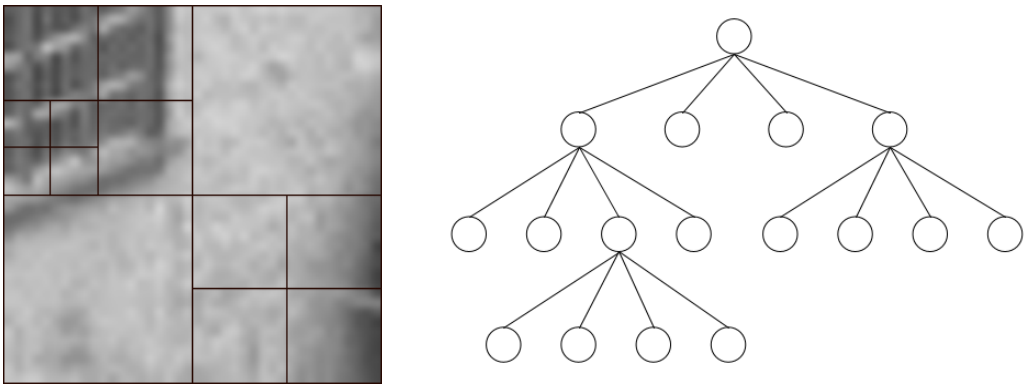


Figure 2. Original image and its quadtree structure

The bit rate in quadtree-based BTC is significantly lower than that in AMBTC, since it is possible to represent an entire block with its block mean value. Correspondingly, due to the loss of image detail, image quality of reconstructed image is slightly inferior to AMBTC. So the threshold should be adjusted to balance the rate-distortion performance.

3. PROPOESD METHOD

In this section, we will illustrate our method. First, the original image is partitioned into non-overlapping blocks. Then, Sobel operator is applied on each block to detect the direction of edge in each block, and the current block is decided whether it should be further subdivided horizontally or vertically. For blocks with different row and column, divide it into two subblocks of the same size. For the minimal block, we first encode it by AMBTC and attempt to decode it using two quantization levels in encoding phase. If the MSE generated from this procedure is greater than the post-threshold, a 2-bit-plane quantization scheme is used. Thus, we utilize adaptive bit plane to optimize the BTC, which depends on its MSE. The framework of our proposed method is shown in Fig.3.

3.1 Edge detection

We use Sobel operator to detect possible edge in the block. The correlation between current block and two Sobel operators (Fig.4(a) and 4(b)) is first calculated by using formula 2.

$$w(x,y) * f(x,y) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} w(s,t) f(x+s, y+t) \tag{2}$$

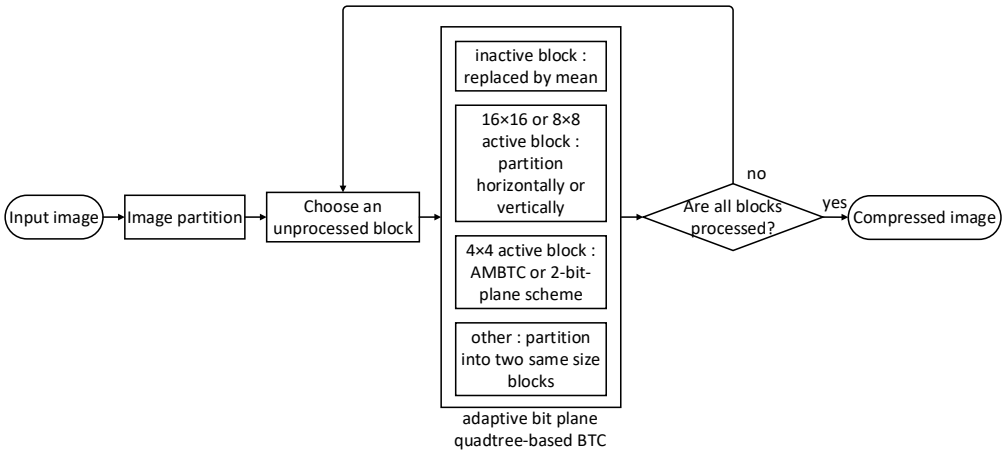


Figure 3. Framework of proposed method

Here $w(x, y)$ denotes 3 × 3 Sobel operator and denotes the pixel value in the image. To simplify the computation, we use valid part of pixel in the original image without padding. Thus, the size of correlation result is smaller than that of the original block. Then, the summation of absolute of result are calculated. If one is greater than the other one, it has higher probability of having an edge in the corresponding direction. So we utilize the edge information to divide the current block, making its subblocks more likely to be an inactive one, which could be replaced by their mean value. Thus, there may be less active blocks, therefore reducing the bit rate efficiently. Note that any block segmentation using the quadtree can be achieved using our method, though more levels in the tree structure may be needed. However, the reverse is not true.

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

(a) (b)

Figure 4. Sobel operator

We give an example to illustrate it. Suppose that Fig.5(a) is the current block to be processed. We use formula 2 to calculate the correlation of this block and two Sobel operators shown in Fig.4(a) and 4(b). The correlation result are shown in Fig.5(b) and 5(c). Then, we sum up all absolute value of the result, and get 6455 and 3151 respectively. The block correlated with first Sobel operator is larger than the second one, which means there is a horizontal edge in this block. As a result, the current block should be divided vertically into two 4 × 8 subblocks. An additional indicator bit is needed to record how to divide the current block, 1 for vertically and 0 for horizontally.

111	89	71	51	51	45	55	49
105	92	64	57	51	52	56	53
107	96	65	50	47	48	54	51
131	112	74	64	62	59	55	53
176	157	134	114	99	80	69	59
198	193	182	175	156	139	113	90
194	198	201	201	194	188	175	150
197	199	199	200	198	198	200	196

-4	6	12	6	-1	-3
-76	-47	-35	-36	-24	-5
-260	-263	-249	-200	-131	-70
-337	-408	-424	-379	-312	-233
-167	-262	-336	-385	-417	-411
-28	-65	-109	-168	-247	-339

164	154	64	18	-21	-9
182	175	61	14	-12	-1
198	185	77	46	37	30
157	152	108	109	110	97
67	76	94	119	135	157
0	11	41	64	79	127

(a) current block (b) correleation with Fig.4(a) (c) correleation with Fig.4(b)

Figure 5. Example of edge detection

3.2 Two bit plane quantization

For 2 bit plane, there are 4 quantization levels for a block, denoted by quan(i) (i=1,2,3,4). We use uniform quantization levels to save the bit rate. In order to prevent the maximum or minimum pixel value dominating the result, we use average of the four largest pixels for the highest quantization level quan(4), and average of the four smallest pixels for the lowest quantization level quan(1). Then, other quantization level can be calculated by formula 3.

$$quan(i) = \frac{4-i}{3} quan(1) + \frac{i-1}{3} quan(4)(i = 1,2,3,4) \tag{3}$$

Thus, we only need to encode the lowest quantization level and step length obtained in formula 4.

$$l = \frac{quan(4) - quan(1)}{3} \tag{4}$$

We use 8 bits to represent quan(1) and 7 bits for step length, since its value is no more than 127. As for bit plane, each pixel is quantized to nearest quantization level using formula 5 and represented by 00,01,10,11, respectively.

$$index(i) = \min(abs(x_t - quan(i)))(t = 1,2,...16, i = 1,2,3,4) \tag{5}$$

Here x denotes each pixel in the block. Adding two additional indicator bits (one for active block another one for 2 bit plane coding decision), the active 4 × 4 block is encoded by totally 49 bits. Indeed, the bit rate of this block is rather high. However, just like what we analyzed in section 2.1, there are only a few blocks occupy most of the MSE loss. So we can adjust the post-threshold to control how many blocks are allowed to be encoded by our scheme.

In decoding phase, two indicator bits are first decoded to decide how to reconstruct the current block. Then 15 bits are read as first quantization level and step length, and four quantization levels are calculated using formula 6.

$$quan(i) = quan(1) + (i-1) * l(i = 1,2,3,4) \tag{6}$$

Finally, the bit plane is decoded and the block is reconstructed by corresponding quantization level.

We give an example to illustrate the whole procedure. Fig.6(a) is an active 4 × 4 block with variance of about 5967. We first try to encode it by AMBTC as introduced in section 2.1. The result is shown in Fig.6(b). Then the encoder calculates the MSE between reconstructed block and original block, about 1337, which is unacceptable. So we use 2 bit plane quantization scheme. The average of the four largest pixel value is 226.5, and the average of the four smallest ones is 40. So four quantization levels (40,102.2,164.3,226.5) are obtained by formula 3. Then the first quantization level 40 is encoded by 8 bits, and step length calculated by formula 4 is rounded to nearest integer 62 and encoded by 7 bits. Then bit plane is encoded by (11111000111110001111100011111000) for 32 bits by formula 5 and corresponding bit code. In decoding phase, indicator bits, quantization data and bit plane are decoded sequentially. Then quantization data is used to calculate four quantization levels (40,102,164,226) by using formula 6. Finally, the block is reconstructed according to bit plane and corresponding quantization level, as shown in Fig.6(c). As we can see, the reconstructed block only has 148.5 MSE, nearly one tenth of that encoded by AMBTC.

227	214	148	40
229	212	146	42
226	221	142	38
224	221	134	40

(a) original

222	222	91	91
222	222	91	91
222	222	91	91
222	222	91	91

(b) AMBTC

226	226	164	40
226	226	164	40
226	226	164	40
226	226	164	40

(c) proposed

Figure 6. Example of two bit plane quantization

3.3 Proposed compression algorithm

Combined with edge detection and adaptive bit plane quantization, the proposed adaptive bit plane quadtree-based BTC algorithm is as follow:

Step 1: If the image is color image, decompose it into red, green and blue channel and process each channel as grey image separately. Otherwise, proceed to Step 2.

Step 2: Partition the image into 16 × 16 non-overlapping blocks.

Step 3: For each unprocessed 16 × 16 image block, if its variance is smaller than pre-threshold, encode this 16 × 16 block using its block mean and proceed to Step 6. Otherwise, proceed to Step 4.

Step 4: According to current block size and its variance, choose how to process it.

4.1 If the variance of current block is smaller than pre-threshold, encode this block using its block mean and proceed to Step 6. Otherwise, the current block is active.

4.2 If the current block size is 4 × 4, proceed to Step 5.

4.3 If the current block size is 16 × 16 or 8 × 8, use the edge detection technique introduced in section 3.1, and divide the current block into two subblocks. For each subblock, return to Step 4.

4.4 Otherwise, the current block has different sizes of row and column. Just divide it into two subblocks with the same size and return to Step 4.

Step 5: The 4×4 active block is first encoded by AMBTC introduced in section 2.1. Then use two quantization levels and bit plane information to decode immediately. If the MSE generated from reconstructed block is smaller than post-threshold, encode it by AMBTC. Otherwise, encode it by 2-bit-plane scheme introduced in section 3.2.

Step 6: If there are still any 16 × 16 blocks to be processed, return to Step 3.

4. EXPERIMENTAL RESULTS

In this section, extensive experiments are carried out to validate the performance of our compression scheme. We choose eight standard grey test images ‘‘Lena’’, ‘‘Peppers’’, “Boat”, “Couple”, “Hill”, “Airplane”, ‘‘Barbara’’, ‘‘Baboon’’, downloaded from Department of Computer Science and Artificial Intelligence of University of Granada [17]. The standard test images are shown in Fig.7.

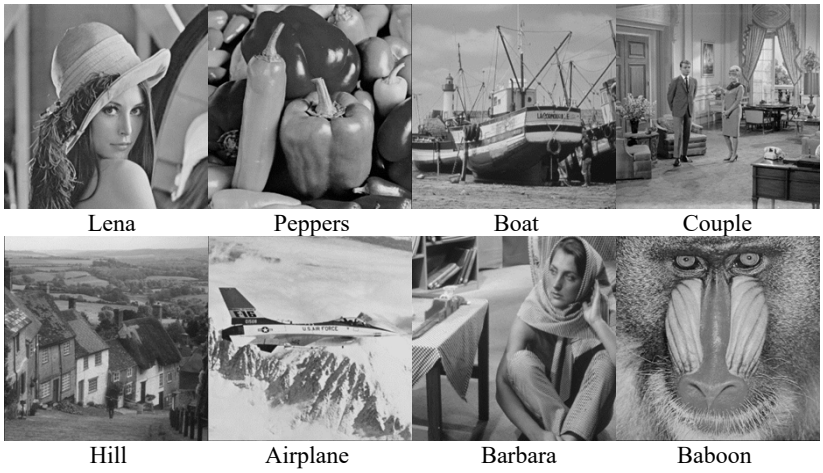


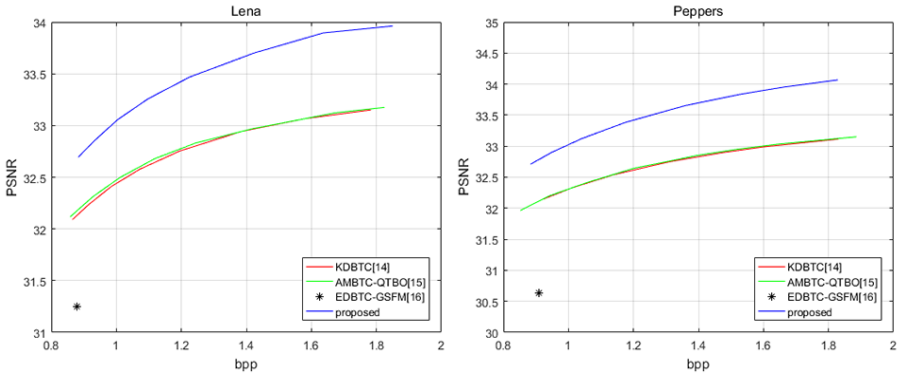
Figure 7. Standard test images

By trying different combinations with pre- and post-threshold, the rate-distortion performance of Lena is shown in Table 1. From the results, we can find some key points: (1) Bit rate mainly depends on pre-threshold. The larger pre-threshold is, the lower bit rate will be due to more blocks are replaced by mean values. But the influence of changing the post-threshold is insignificant, even in large post-threshold region; (2) At the same pre-threshold, PSNR doesn’t change uniformly following the change of post-threshold. When post-threshold is slightly higher than pre-threshold, PSNR decreases sharply as the post-threshold increases. However, it changes more smoothly when the post-threshold is relatively high; (3) Computational complexity mainly depends on post-threshold. Because the 2 bit plane quantization is more time-consuming than AMBTC, when post-threshold is high, there are only several blocks encoded by 2 bit plane. It’s obvious that if the post-threshold is positive infinite, it is just identical to AMBTC.

Table 1. bit rate and PSNR of proposed scheme on image “Lena”

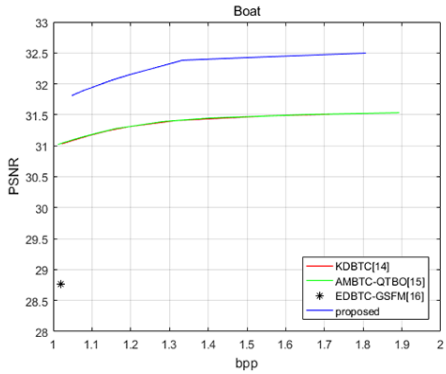
post \ pre		5	10	15	20	25	30
50	bpp	1.97	1.55	1.35	1.22	1.13	1.06
	psnr	36.19	35.77	35.40	35.07	34.77	34.47
100	bpp	1.92	1.49	1.29	1.16	1.07	1.00
	psnr	35.46	35.09	34.78	34.49	34.23	33.96
200	bpp	1.87	1.44	1.24	1.11	1.02	0.95
	psnr	34.49	34.20	33.94	33.70	33.48	33.26
300	bpp	1.85	1.43	1.23	1.10	1.00	0.93
	psnr	33.96	33.70	33.47	33.26	33.05	32.85
500	bpp	1.84	1.41	1.22	1.09	0.99	0.92
	psnr	33.54	33.28	33.07	32.87	32.69	32.50
800	bpp	1.84	1.41	1.21	1.08	0.99	0.92
	psnr	33.20	32.98	32.78	32.60	32.43	32.25

To further validate the efficiency of our method, it’s compared with some start-of-the-art BTC variants [14,15,16]. We fixed the post-threshold at 300 and adjust pre-threshold to change the bit rate, the result is shown in Fig.8. The result prove that our method gains 0.853 dB and 0.841 dB PSNR on average compare to [14] and [15]. It improves more for high texture image ‘‘Barbara’’ and “Baboon”.

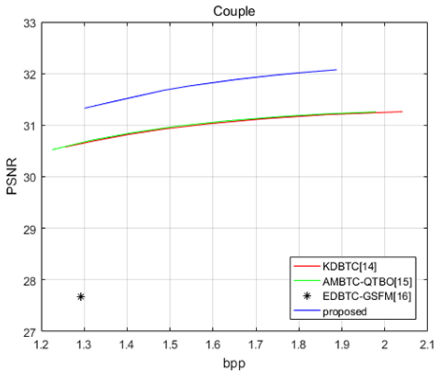


(a) Lena

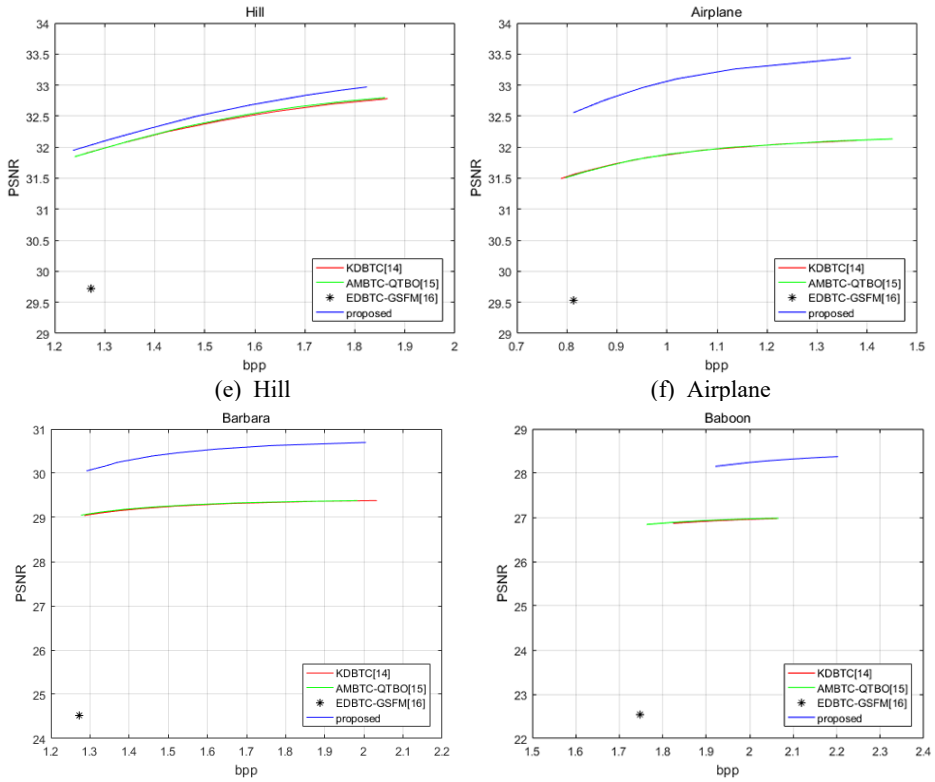
(b) Peppers



(c) Boat



(d) Couple



(e) Hill (f) Airplane

(g) Barbara (h) Baboon

Figure 8. Comparison with other BTC variants in rate-distortion curves
(EDBTC [16] has no adjustable parameter, so its bit rate is fixed)

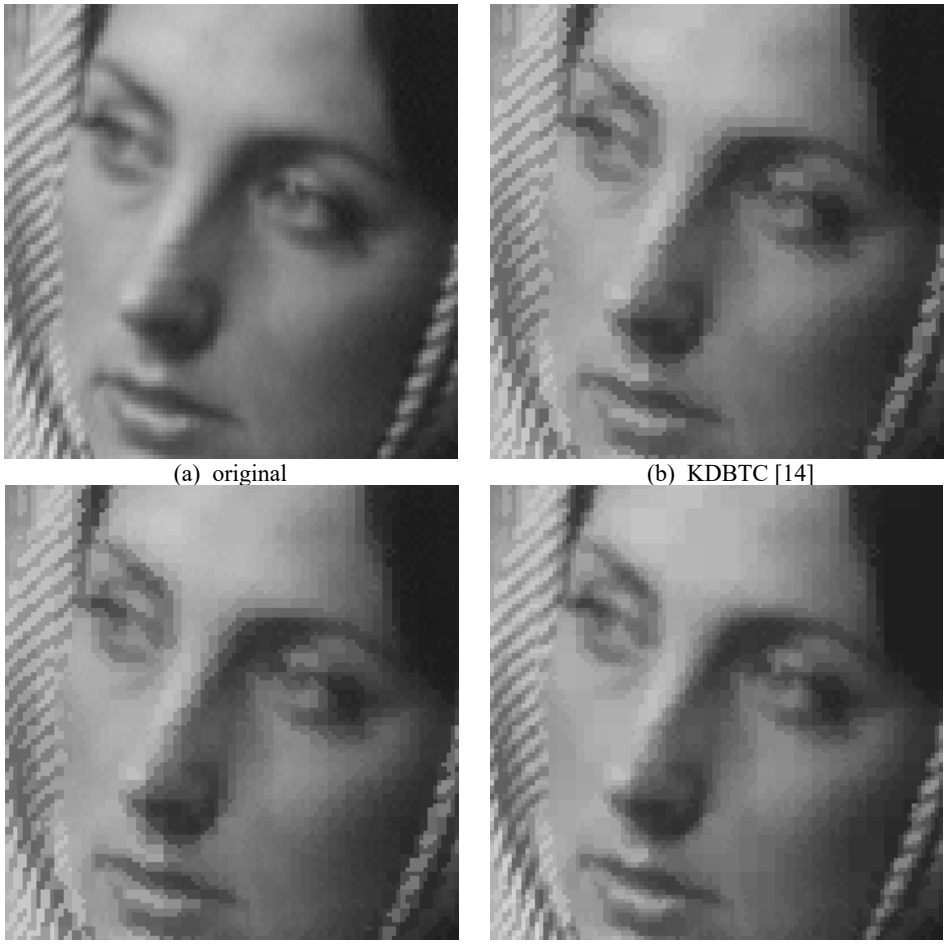
We also compare the performance of different methods in terms of subjective image quality. We choose test image ‘‘Barbara’’ and fixed the bit rate at 2.0 bpp. For clarity of comparison, the decoded image is zoomed in specified region (100 × 100 pixels). The result is shown in Fig.9. It’s obvious that our method outperforms other methods in terms of subjective quality, the reason is that four quantization levels are adaptively used in some high variance blocks, which is more consistent with human visual system.

5. CONCLUSION

In this paper, we proposed a quadtree-based block truncation coding algorithm combined with adaptive bit plane quantization. By adjusting the pre- and post-threshold, we balance the PSNR, bit rate and computation complexity, and get the rate-distortion performance of the proposed method.

On one hand, the edge detection technique takes different block size into consideration. It makes more blocks become inactive and be replaced by its mean value, which significantly reduce the bit rate. On the other hand, the use of adaptive bit plane, targeted at high MSE loss block, can compensate the degradation of image quality and occurrence of blocking effect. As a result, the proposed method combines those two techniques, which can both improve the image quality and decrease bit rate at the same time. Although more computation is needed in encoding phase, decoding time is almost the same as other BTC variants.

Extensive experimental results show that our method outperforms other state-of-the-art BTC variants in terms of both objective and subjective quality, especially for images with rich texture. However, the proposed method is still inferior to JPEG [18], which processes the image in frequent domain. The most advantage of our method compared with JPEG is much lower computational complexity and memory space, which is desirable for some single-core devices or real time applications.



(a) original (b) KDBTC [14]

(c) AMBTC-QTBO [15] (d) proposed

Figure 9. Comparison with other BTC variants in terms of subjective image quality

ACKNOWLEDGMENT

This work was supported by Beijing Natural Science Foundation (4164079, Z2025001201502), and National Science Foundation of China (61672068), and the Opening Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research.

REFERENCES

[1] Delp, E. J. and Mitchell, O. R., "Image coding using block truncation coding," IEEE Trans. Comm. 27, 1335–1342, (1979).

[2] Fränti, P., Nevalainen, O., and Kaukoranta, T., "Compression of Digital Images by Block Truncation Coding: A Survey," Computer Journal 37(4), 308-332, (1994).

[3] Lema, M., and Mitchell, O. R., "Absolute Moment Block Truncation Coding and Its Application to Color Images," Comm. IEEE Trans. on 32(10), 1148-1157, (1984).

[4] Healy, D., and Mitchell, O. R., “Digital Video Bandwidth Compression Using Block Truncation Coding,” IEEE Trans. on Comm. 29(12), 1809-1817, (1981).

[5] Chen, D., and Bovik, A. C., “Visual pattern image coding,” Comm. IEEE Trans. on 38(12), 2137-2146, (1990).

[6] Chan, T. P., Zeng B., and Liou, M. L., “Visual pattern BTC with two principle colors for color images,” IEEE International Symposium on Circuits and Systems, 1, 235-238, (1995).

[7] Chandravadhana, S., and Nithiyanandam, N., “Least mean square error-based image compression using block truncation coding,” International Journal of Information and Communication Technology, 11(1), 25-37, (2017)

[8] Xu, Z. X., and Chan, Y. H., “Eliminating blocking artifacts in halftoning-based block truncation coding,” European Signal Processing Conference, 7760381, 913-917, (2016)

[9] Daga, R. R. M., and Fernandez, P.”Multi-Partition Block Truncation Coding for Image Compression,” NCITE, (2015).

[10] Guo, J. M., and Liu, Y. F., “Improved block truncation coding using optimized dot diffusion,” IEEE Trans. on Image Proc. A Publication of the IEEE Signal Proc. Society 23(3), 1269-1275, (2014).

[11] Vaisey, J., and Gersho, A., “Image compression with variable block size segmentation,” IEEE Trans. on Signal Proc.,40(8), 2040-2060, (1992).

[12] Hu, Y. C., and Chang, C. C., "Quadtree-segmented image coding schemes using vector quantization and block truncation coding." Optical Engi. 39(2), 464-471, (2000).

[13] Hu, Y. C., "Low-complexity and low-bit-rate image compression scheme based on absolute moment block truncation coding." Optical Engi. 42(7), 1964-1975, (2003).

[14] Daga, R. R. M., and Fernandez, P. L., “k-d Tree-Segmented Block Truncation Coding for Image Compression,” International Conference on Automation Sciences, (2016).

[15] Chen, W. L., Hu, Y. C., Liu, K. Y., Lo, C. C., and Wen, C. H., “Variable-Rate Quadtree-segmented Block Truncation Coding for Color Image Compression,” International Journal of Signal Proc. Image Proc. & P 7(1), 65-76, (2014).

[16] Yang, F. J., Lien, C. Y., Chen, P. Y., and Hsu, C. L., “An Efficient Quadtree-Based Block Truncation Coding for Digital Image Compression,” International Conference on Advanced Information Networking and Applications Workshops, 939-942, (2016).

[17] http://decsai.ugr.es/cvg/index2.php This is the computer vision group of Department of Computer Science and Artificial Intelligence of University of Granada.

[18] Wallace, G. K., “The JPEG still picture compression standard,” Comm. of the Acm 38(1), xviii-xxxiv, (1991).