Enhancing Image Compression by Adaptive Block Truncation Coding with Edge Quantization

Ching-Nung Yang*, Yi-Cheng Chen*, Yung-Chien Chou*, Tao-Ku Chang*, and Cheonshik Kim†
*National Dong Hwa University, Taiwan
†Sejong University, Korea

Abstract— **Recently, Mathews and Nair proposed an image compression using adaptive block truncation coding based on edge quantization (ABTC-EQ). Their approach deals with an image by two types of blocks, edge blocks and no-edge blocks. Different from using bi-clustering approach on all blocks in previous BTC-like schemes, ABTC-EQ adopts tri-clustering to tackle edge blocks. The compression ratio of ABTTC-EQ is reduced, but the visual quality of reconstructed image is significant improved. However, it is observed that ABTC-EQ uses 2 bits to represent the index of three clusters in a block. We can only use an average 5/3 bits by variable-length code to represent the index of each cluster. On the other hand, there are two observations on the quantization levels in a block. The first observation is that the difference between two quantization values is often smaller than the quantization values themselves. The second observation is that more clusters may enhance the visual quality of reconstructed image. Based on variable-length coding and the above observations, we design variants of ABTC-EQ to enhance the visual quality of reconstructed image and compression ratio.**

I. INTRODUCTION

This document shows guidelines for preparing a final camera-ready manuscript in the proceedings of SISA 2019. The manuscript for the initial submission should also be formatted in this style. The format here described allows for a graceful transition to the style required for that publication. Rapid improvements in the area of network and information technology increase the services of digital multimedia, especially digital image, in today's digitalized and informationized world. For example, consider storage and transmission. File compression reduces the amount of space needed to store data, and also speeds the time to send over the Internet. About compressing digital images, there are two main types of compressing digital images, lossless and lossy. In this paper, we deal with block truncation coding (BTC) and its variants [1-4], which are lossy compression algorithms. Because of their stable compression rates and low computation efforts, BTC-like schemes are widely used in cryptography, e.g., data hiding [5-9], watermarking [10], secret image sharing and visual cryptography [11-14].

The BTC was first proposed by Delp and O. Mitchell [1]. It is a block-based lossy image compression technique for grayscale or color images, where a quantizer is adopted to reduce the number of gray levels in each block. An image is subdivided into non-overlapping blocks. Then, each block is coded by the mean, the standard deviation and the bit map consisting of 0's and 1's. Suppose that a lock size has (4×4) pixels, and then these 16 pixels (i.e., 128 bits) can be represented by a trio with (8+8+16)=32 bits. Therefore, the compression ratio (CR) of BTC is determined from CR= (16×8)/32=4 for this case. BTC provides good compression ratio without degrading the visual quality of reconstructed image much. However, the processing time for BTC algorithm has significant computational complexity, and thus it is not generally recommended for time consuming applications. In [2], Lema and Mitchell proposed absolute moment BTC (AMBTC), a variation of BTC, which has a simple computation. AMBTC adopts bi-clustering approach, and thus still uses two quantization levels in a block like BTC. Afterwards, the authors in [3] proposed a modified BTC (MBTC) to further enhance the visual quality of reconstructed image by using the so-called max-min quantizer.

Because of using the simple bi-clustering approach in AMBTC and MBTC, the details near edges will be removed from the reconstructed image. As we know, edges are important information for human visual perception. Moreover, the information of edges are also necessary for some image processing applications, e.g., pattern recognition and optical character recognition. Therefore, the loss of details near edges will compromise the availability of BTC-like compression, and this leads a motivation for designing an edge-based BTC compression algorithm. Recently, Mathews and Nair proposed an adaptive BTC based on edge quantization (ABTC-EQ) [4]. The authors first use edge detector to divide an image into edge blocks and non-edge blocks. Afterwards, by applying MBTC on non-edge blocks, and adopting tri-clustering algorithm to tackle edge blocks, ABTC-EQ may provide the better visual quality of reconstructed image, but has the worse compression ratio when compared with AMBTC or MBTC. This is because edge blocks have to use three pixels to represent the quantization values for three clusters. However, it is observed that ABTC-EQ uses 2 bits to represent the index of each cluster. We can only use an average 5/3 bits by variable-length code to represent the index of each cluster. Note, in fact, 2 bits can be used to represent four clusters. On the other hand, there are two observations on the quantization levels in a block. The first observation is that the difference between two quantization values is often smaller than the quantization values themselves. The second observation is that more clusters may enhance the visual quality of reconstructed image. Based on variable-length coding and the above observations, we design variants of ABTC-EQ to enhance peak signal-to-noise ratio (PSNR) and

CR. The rest of this paper is organized as follows. Section II briefly reviews AMBTC, MBTC, and Mathews and Nair's ABTC-EQ. The proposed ABTC-EQ with theoretical analyses are formulated in Section III. Experiment and comparison are given in Section IV. And, Section V draws some conclusions.

II. Previous Works

A. AMBTC and MBTC

In [2], Lema and Mitchell propose AMBTC, a variation of BTC, which has a simple computation. Compared with BTC, it requires less computation time. AMBTC still preserves the higher mean and lower mean of each block and uses these two values to quantize output. It provides better image quality than BTC, as well as reasonable computational complexity. In AMBTC, an image is first subdivided into non-overlapping (k×k)-sized blocks, where k may be set to be (4×4), (6×6), (8×8) and so on. AMBTC adopts block-wise operation. For each block, the mean pixel value $\bar{x}$ is calculated by

$$\bar{x} = \frac{1}{k \times k} \sum_{i=1}^{k^2} x_i, \tag{1}$$

where x_i denotes the i-th pixel value in this block. Each pixel value x_i is compared with this mean value $\bar{x}$ in such a way that a bitmap $M = [b_i]$ of the same block size which consists of two clusters is generated according to Eq. (2).

$$b_i = \begin{cases} 1, \text{ if } x_i \geq \bar{x}, \\ 0, \text{ if } x_i < \bar{x}. \end{cases} \tag{2}$$

Let t be the number of "1" in the bitmap M, i.e., the number of pixels under $x_i \geq \bar{x}$. The means μ_1 and μ_0 are, respectively, the higher and lower ranges computed with Eq. (3).

$$\mu_1 = \left\lfloor \frac{1}{t} \sum_{x_i \geq \bar{x}} x_i \right\rfloor \text{ and } \mu_0 = \left\lfloor \frac{1}{(k \times k) - t} \sum_{x_i < \bar{x}} x_i \right\rfloor. \tag{3}$$

Finally, the image block is compressed into two quantization levels μ_0, μ_1 and a bitmap M, and can be represented as a trio (μ_0, μ_1, M). A bitmap M contains the bit-planes that represent the pixels, and the values μ_0 and μ_1 are used to decode the AMBTC compressed image. For the case k=4, i.e., we deal with an image by (4×4) block-wise operation. Sixteen pixels in a block are represented as a trio (μ_0, μ_1, M) of 8+8+16=32 bits, and thus the CR is (16×8)/32=4. Consider the example of a 512×512-pixel image. The file size of 2M bits can be reduced to 0.5 M bits. In decoding phase, when two quantization levels and the bitmap obtained, the corresponding image block can be easily reconstructed by replacing every "1" in a bitmap M with μ_1, while every "0" is replaced with μ_0.

Because AMBTC provides better image quality and the fat computation, most BTC-based data hiding schemes and secret image sharing schemes adopt AMBTC approach. In [3], the authors proposed a MBTC by using max-min quantizer to further enhance the quality of reconstructed image. In AMBTC, the threshold value used for distinguishing two clusters is simply using the mean value $\bar{x}$ in a block. A threshold value x_{th} of MBTC in Eq. (4) is obtained by calculating the average value of the maximum value ($x_{\max}$), minimum value ($x_{\min}$), and mean value ($\bar{x}$) in a block.

$$x_{th} = \frac{(x_{\max} + x_{\min} + \bar{x})}{3}. \tag{4}$$

Afterwards, by the same argument of AMBTC but using x_{th} instead of $\bar{x}$, we may obtain a trio for each block in MBTC.

B. Mathews and Nair's ABTC-EQ

In previous BTC and its variants, e.g., AMBTC and MBTC, the quantization approaches are all the same. They all use bi-clustering approaches (two quantization levels) in all blocks. ABTC-EQ is an edge-based block truncation scheme. Its quantization is based on the edge information. In ABTC-EQ, we find the edged image from the input image by Canny edge detector [15]. We have an edge map $E = [e_i]$ for a (k×k)-sized block, and then classify blocks into edge block and non-edge blocks under the following condition. If any of the edge values e_i, $1 \leq i \leq k^2$, in E is "1" and not all the edge values are "1" then the image block is defined as an edge block, otherwise a non-edge block.

For these two types of blocks, we use various quantization approaches. For non-edge blocks, we use MBTC. Therefore, a non-edge image block can be represented as a trio (μ_0, μ_1, M_n), where we intentionally use M_n notation to represent a bit map for the non-edge block. On the other hand, for edge blocks, we use tri-clustering approach. The pixels in a lock are classified into three clusters (c_0, c_1, c_2), which similar pixels are grouped into the same cluster, by k-means clustering algorithm [16]. A bitmap $M_e = [b_i]$ of the edge block is generated by Eq. (5).

$$b_i = \begin{cases} 00, \text{ if } x_i \in c_0, \\ 01, \text{ if } x_i \in c_1, \\ 10, \text{ if } x_i \in c_2. \end{cases} \tag{5}$$

The mean μ_i' of each cluster c_i, $0 \leq i \leq 2$, is calculated with Eq. (6), where $\mu_0' \leq \mu_1' \leq \mu_2'$. Thus, an edge image block is represented as $(\mu_0', \mu_1', \mu_2', M_e)$.

$$\mu_i' = \left\lfloor \frac{1}{|c_i|} \sum_{x_i \in c_i} x_i \right\rfloor, 0 \le i \le 2. \quad (6)$$

Bi-clustering and tri-clustering approaches are performed for non-edge blocks and edge blocks, respectively. To discriminate edge blocks from non-edge blocks, an identifier flag f should be defined and assigned with the value 0 (respectively, 1) for the edge block (respectively, non-edge block). Finally, k^2 pixels may be represented as $(f=1, \mu_0, \mu_1, M_n)$ or $(f=0, \mu_0', \mu_1', \mu_2', M_e)$. Therefore, the CR of ABTC-EQ is dynamic not static like previous BTC schemes.

III. The Proposed ABTC-EQ

A. Design Concept

The pixel grayscale values in an edge block vary greatly from each other, and thus Mathews and Nair's ABTC-EQ [4] uses tri-clustering approach to deal with edge blocks to enhance the visual quality of reconstructed image. BTC and its variants use (μ_0, μ_1, M) to represent a block, while Mathews and Nair's ABTC-EQ uses $(f=1, \mu_0, \mu_1, M_n)$ or $(f=0, \mu_0', \mu_1', \mu_2', M_e)$ instead. Thus, Mathews and Nair's ABTC-EQ enhances the visual quality of reconstructed image but reduces the CR due to using an extra flag bit f and extra quantization value μ_2'. To enhance BTC-like approaches, we obviously should improve the CR as well as with a high PSNR. Therefore, when further enhancing Mathews and Nair's ABTC-EQ, we have to simultaneously take two features, CR and PSNR, into account.

By observation of Mathews and Nair's ABTC-EQ, its PSNR enhancement of reconstructed image comes from using tri-clustering approach on edge blocks. However, the ineluctable weakness accompanies with an advantage. The CR is reduced due to using extra f and μ_2'. Some in-depth observations on ABTC-EQ are listed below.

Observation 1: *It is not necessarily to use two bits to represent a pixel in bit map M_e for edge block.*

As shown in Eq. (5), we use (00, (01), and (10) for three clusters c_0, c_1, and c_2, respectively. However, we may use Huffman code, a variable-length code, to represent three clusters, by (0), (10), and (11) with average length 5/3 bits for clusters c_0, c_1, and c_2. Note: the Huffman code can be uniquely decoded. By this approach, the size of bit map M_e is reduced from $2k^2$ to $(5/3)k^2$. Finally, the CR can be enhanced.

Observation 2: *Consider two quantization values (say μ_i' and μ_j', where $\mu_i' < \mu_j'$). The difference $(\mu_j' - \mu_i')$ between two quantization values is often smaller than the quantization values μ_i' and μ_j' themselves.*

By observing quantization value, we herein use a homologous way to describe the difference between two quantization values with the help of the coder in converting voice. A well-known coder, differential pulse code modulation (DPCM), is described as follows: obtain the pulse of analog signals by sampling and then convert the difference of pulses into binary sequences using the non-uniform coding scale. This property is also true for the quantization levels in ABTC-EQ, i.e., the large difference of quantization values does not occur frequently. Thus, we could carefully design our quantization ranges for the small difference between two quantization values.

Observation 3: *More clusters may enhance the PSNR of reconstructed image.*

In previous BTC-like schemes, all blocks adopt bi-clustering, i.e., using two quantization ranges for each block. Mathews and Nair's ABTC-EQ performed tri-clustering approach on edge blocks. Because there are three values μ_0, μ_1, and μ_2 to approximate the pixel grayscale values, it can reduce the mean square error. It is obvious that we may use more clusters (say four clusters) to more precisely approximate pixel values.

B. The Proposed Scheme

Scheme A: Same as ABTC-EQ, while we determine the elements of bitmap $M_e' = [b_i]$ from Observation 1, via Eq. (7). Finally, we use $(f=0, \mu_0', \mu_1', \mu_2', M_e')$ for an edge block.

$$b_i = \begin{cases} 0, & \text{if } x_i \in c_0, \\ 10, & \text{if } x_i \in c_1, \\ 11, & \text{if } x_i \in c_2. \end{cases} \quad (7)$$

Theorem 1: Suppose that the percentages of edge blocks and non-edge blocks in an image be p_e and p_n, where $p_e + p_n = 1$, when dealing with (k×k)-pixel block in an image. The CR of Mathews and Nair's ABTC-EQ is $\mathrm{CR}_{\mathrm{MN}} = \frac{8k^2}{(17+k^2)+(8+k^2)\times p_e}$, and Scheme A has the $\mathrm{CR}_{\mathrm{A}} = \frac{8k^2}{(17+k^2)+(8+(2k^2/3))\times p_e}$, where $\mathrm{CR}_{\mathrm{A}} > \mathrm{CR}_{\mathrm{MN}}$. Meanwhile, they have the same PSNR of reconstructed image, i.e., $\mathrm{PSNR}_{\mathrm{MN}} = \mathrm{PSNR}_{\mathrm{A}}$.

Proof: By the compression data formats $(f=1, \mu_0, \mu_1, M_n)$ and $(f=0, \mu_0', \mu_1', \mu_2', M_e)$ of ABTC-EQ format, and the formats of Scheme A $(f=1, \mu_0, \mu_1, M_n)$ and $(f=0, \mu_0', \mu_1', \mu_2', M_e')$, we may easily derive $\mathrm{CR}_{\mathrm{MN}}$ and CR_{A} in Eq. (8) and Eq. (9), respectively.

$$\begin{cases} \mathrm{CR}_{\mathrm{MN}} = \dfrac{8\times k^2}{\underbrace{(1+2\times 8+k^2)\times p_n}_{\text{non-edge block}} + \underbrace{(1+3\times 8+2\times k^2)\times p_e}_{\text{edge block}}} \\ = \dfrac{8k^2}{(17+k^2)+(8+k^2)\times p_e}. \end{cases} \quad (8)$$

$$\begin{cases} \mathrm{CR}_{\mathrm{A}} = \dfrac{8\times k^2}{\underbrace{(1+2\times 8+k^2)\times p_n}_{\text{non-edge block}} + \underbrace{(1+3\times 8+(5/3)\times k^2)\times p_e}_{\text{edge block}}} \\ = \dfrac{8k^2}{(17+k^2)+(8+(2k^2/3))\times p_e}. \end{cases} \quad (9)$$

By Eqs. (8) and (9), since $(8+(2k^2/3)) < (8+k^2)$ we obviously have $\mathrm{CR}_{\mathrm{A}} > \mathrm{CR}_{\mathrm{MN}}$. Except using different bit map M_e' from M_e, Scheme A uses the same approaches of ABTC-EQ. Thus, both schemes have the same PSNR, i.e., $\mathrm{PSNR}_{\mathrm{MN}} = \mathrm{PSNR}_{\mathrm{A}}$. □

Scheme B: We use the same bit map $M_e' = [b_i]$ from Observation 1. Moreover, via Observation 2, we use a new $(\delta_0,\ \delta_1,\ \delta_2)$ to represent $(\mu_0',\ \mu_1',\ \mu_2')$. In Scheme B, we define four classifications of quantization ranges. For each value $\delta_i,\ 0 \le i \le 2,$ we use n_i bits with a radix $R_i : (r_{n_i},\ r_{n_i-1},\ \ldots,\ r_1)$ to represent its grayscale value.

$$\begin{cases} \text{(I) } (n_0=7, n_1=7, n_2=7) \text{ with } \begin{cases} R_0 : (r_7, r_6, \ldots, r_1) = (128\ 64\ 32\ 16\ 8\ 4\ 2), \\ R_1 : (r_7, r_6, \ldots, r_1) = (64\ 32\ 16\ 8\ 4\ 2\ 1), \\ R_2 : (r_7, r_6, \ldots, r_1) = (64\ 32\ 16\ 8\ 4\ 2\ 1). \end{cases} \\ \text{(II) } (n_0=6, n_1=6, n_2=6) \text{ with } \begin{cases} R_0 : (r_6, r_5, \ldots, r_1) = (128\ 64\ 32\ 16\ 8\ 4), \\ R_1 : (r_6, r_5, \ldots, r_1) = (64\ 32\ 16\ 8\ 4\ 2), \\ R_2 : (r_6, r_5, \ldots, r_1) = (64\ 32\ 16\ 8\ 4\ 2). \end{cases} \\ \text{(III) } (n_0=5, n_1=5, n_2=5) \text{ with } \begin{cases} R_0 : (r_5, r_4, \ldots, r_1) = (128\ 64\ 32\ 16\ 8), \\ R_1 : (r_5, r_4, \ldots, r_1) = (64\ 32\ 16\ 8\ 4), \\ R_2 : (r_5, r_4, \ldots, r_1) = (64\ 32\ 16\ 8\ 4). \end{cases} \\ \text{(IV) } (n_0=4, n_1=4, n_2=4) \text{ with } \begin{cases} R_0 : (r_4, r_3, r_2, r_1) = (128\ 64\ 32\ 16), \\ R_1 : (r_4, r_3, r_2, r_1) = (64\ 32\ 16\ 8), \\ R_2 : (r_4, r_3, r_2, r_1) = (64\ 32\ 16\ 8). \end{cases} \end{cases} \quad (10)$$

The value $\delta_i,\ 0 \le i \le 2,$ is then iteratively determined based on radix R_i with the condition $\min\{\mu_i' - \sum_{j=0}^{i} \delta_j\}$. Finally, we use $(f=0,\ \delta_0,\ \delta_1,\ \delta_2,\ M_e')$ for an edge block.

Theorem 2: Suppose that the percentages of edge blocks and non-edge blocks in an image be p_e and p_n, where $p_e + p_n = 1$, when dealing with (k×k)-pixel block in an image. Scheme B has the $\mathrm{CR}_{\mathrm{B}} = \dfrac{8k^2}{(17+k^2)+(\sum_{i=1}^{3} n_i + (2k^2/3))\times p_e}$, and $\mathrm{CR}_{\mathrm{B\text{-}IV}} > \mathrm{CR}_{\mathrm{B\text{-}III}} > \mathrm{CR}_{\mathrm{B\text{-}II}} > \mathrm{CR}_{\mathrm{B\text{-}I}} > \mathrm{CR}_{\mathrm{A}}$.

Proof: By the compression data formats of Scheme B $(f=1,\ \mu_0,\ \mu_1,\ M_n)$ and $(f=0,\ \delta_0,\ \delta_1,\ \delta_2,\ M_e')$, we may derive CR_{B} in Eq. (11).

$$\begin{cases} \mathrm{CR}_{\mathrm{B}} = \dfrac{8\times k^2}{\underbrace{(1+2\times 8+k^2)\times p_n}_{\text{non-edge block}} + \underbrace{(1+\sum_{i=1}^{3} n_i + (2/3)\times k^2)\times p_e}_{\text{edge block}}} \\ = \dfrac{8k^2}{(17+k^2)+(\sum_{i=1}^{3} n_i - 16 + (2k^2/3))\times p_e} \end{cases} \quad (11)$$

Via Eqs. (10) and (11), we have compression ratios for these four ranges $\mathrm{CR}_{\text{B-I}} = \dfrac{8k^2}{(17+k^2)+(5+(2k^2/3))\times p_e}$, $\mathrm{CR}_{\text{B-II}} = \dfrac{8k^2}{(17+k^2)+(2+(2k^2/3))\times p_e}$, $\mathrm{CR}_{\text{B-III}} = \dfrac{8k^2}{(17+k^2)+((2k^2/3)-1)\times p_e}$, and $\mathrm{CR}_{\text{B-IV}} = \dfrac{8k^2}{(17+k^2)+((2k^2/3)-4)\times p_e}$. From the above and Eq. (9), we have $\mathrm{CR}_{\text{B-IV}} > \mathrm{CR}_{\text{B-III}} > \mathrm{CR}_{\text{B-II}} > \mathrm{CR}_{\text{B-I}}$. □

All compression ratios of Scheme B are larger than $\mathrm{CR}_{\mathrm{MN}}$ (ABTC-EQ). By Observation 3, we may obtain the approximated $(\mu_0',\ \mu_1',\ \mu_2')$ with a tolerant distortion from $(\delta_0,\ \delta_1,\ \delta_2)$. Moreover, Scheme B may have the higher PSNR than those of AMBTC and MBTC.

Scheme C: By Observation 3, we use four clusters $(c_0,\ c_1,\ c_2,\ c_3)$ for edge blocks, and use a bit map $M_e'' = [b_i]$, as shown in Eq. (12). The mean μ_i'' of each cluster $c_i,\ 0 \le i \le 3,$ is calculated with Eq. (13), where $\mu_0'' \le \mu_1'' \le \mu_2'' \le \mu_3''$. Moreover, via Observation 2, we use a new $(\delta_0',\ \delta_1',\ \delta_2',\ \delta_3')$ to represent $(\mu_0'',\ \mu_1'',\ \mu_2'',\ \mu_3'')$. For each value $\delta_i',\ 0 \le i \le 3,$ we use n_i bits with a radix $R_i : (r_{n_i},\ r_{n_i-1},\ \ldots,\ r_1)$ to represent its grayscale value. In Scheme C, we define a quantization range in Eq. (14). The value $\delta_i',\ 0 \le i \le 3,$ is then iteratively determined based on radix R_i with the condition $\min\{\mu_i'' - \sum_{j=0}^{i} \delta_j'\}$. Finally, we use $(f=0,\ \delta_0',\ \delta_1',\ \delta_2',\ \delta_3',\ M_e'')$ for an edge block.

$$b_i = \begin{cases} 00, \text{ if } x_i \in c_0, \\ 01, \text{ if } x_i \in c_1, \\ 10, \text{ if } x_i \in c_2, \\ 11, \text{ if } x_i \in c_3. \end{cases} \quad (12)$$

$$\mu_i'' = \left\lfloor \frac{1}{|c_i|} \sum_{x_i \in c_i} x_i \right\rfloor,\ 0 \le i \le 3. \quad (13)$$

$$(n_0=6, n_1=6, n_2=6, n_3=6) \text{ with } \begin{cases} R_0 : (r_6, r_5, \ldots, r_1) = (128\ 64\ 32\ 16\ 8\ 4), \\ R_1 : (r_6, r_5, \ldots, r_1) = (64\ 32\ 16\ 8\ 4\ 2), \\ R_2 : (r_6, r_5, \ldots, r_1) = (32\ 16\ 8\ 4\ 2\ 1), \\ R_3 : (r_6, r_5, \ldots, r_1) = (32\ 16\ 8\ 4\ 2\ 1). \end{cases} \quad (14)$$

Theorem 3: Suppose that the percentages of edge blocks and non-edge blocks in an image be p_e and p_n, where $p_e + p_n = 1$, when dealing with (k×k)-pixel block in an image. Scheme C has the $\mathrm{CR_C} = \dfrac{8k^2}{(17+k^2)+(8+k^2)\times p_e}$, where $\mathrm{CR_C} = \mathrm{CR_{MN}}$.

Proof: By the compression data formats of Scheme C $(f=1,\ \mu_0,\ \mu_1,\ M_n)$ and $(f=0,\ \delta_0',\ \delta_1',\ \delta_2',\ \delta_3',\ M_e'')$, we may easily derive $\mathrm{CR_C}$ in Eq. (15).

$$\begin{cases} \mathrm{CR_C} = \dfrac{8\times k^2}{\underbrace{(1+2\times 8+k^2)\times p_n}_{\text{non-edge block}} + \underbrace{(1+(\sum_{i=1}^{4} n_i)+2\times k^2)\times p_e}_{\text{edge block}}} \\ = \dfrac{8k^2}{(17+k^2)+(24-16+k^2)\times p_e} \quad (\because \sum_{i=1}^{4} n_i = 24) \\ = \dfrac{8k^2}{(17+k^2)+(8+k^2)\times p_e} = \mathrm{CR_{MN}} \end{cases} \tag{15}$$

□

IV. Experiment and Comparison

A. Experimental Results

Five test images, Lena, Butterfly, Cameraman, Lake, and Peppers are used for evaluating all BTC-like schemes: AMBTC, MBTC, ABTC-EQ, and the proposed schemes (Scheme A, Scheme B and Scheme C). To properly deal with all (*k*×*k*) blocks, where *k*=4, 6 and 8, we use all test images of the size 504×504 pixels. The evaluation metrics, PSNR, CR, structural similarity (SSIM) index, and feature similarity (SSIM) index, are used to compare the performance of all these schemes. Table I illustrates the comparison of all BTC-like Schemes.

For the test image Lena, consider dealing with (4×4) blocks by all schemes. Scheme C adopts quad-clustering, and also uses 24 bits to represent four quantization values by the approach of using difference. Therefore, Scheme C has the best visual quality (PSNR=39.62 dB), and meanwhile has the same CR=3.09 like ABTC-EQ. Although AMBTC and MBTC have the high CR, they have poor PSNR because they only use bi-clustering approach. The PSNR=33.87 dB of MBTC is slightly greater than the PSNR=33.42 dB of AMBTC. This slight enhancement comes from using the more precise threshold value for MBTC. Scheme A uses tri-clustering and same quantization ranges like ABTC-EQ, and thus Scheme A and ABTC-EQ have the same PSNR=37.49 dB. Because of using a variable-length code to record the index of cluster, Scheme A has the higher CR=3.24 than the CR=3.09 of ABTC-EQ. On the other hand, Scheme B may trade off PSNR for CR by using different quantization ranges. Scheme B-I has PSNR=37.47 dB almost the same to PSNR=37.49 dB of ABTC-EQ, and has the higher CR than Scheme A. If we want to achieve a high CR and meanwhile retain a moderate PSNR, we may choose Scheme B-IV, which has the CR=3.62 PSNR=35.96 dB. Also, all the values of SSIM and FSIM demonstrate a consistency with the performance of PSNR.

TABLE I
COMPARISON OF BTC-LIKE SCHEMES ON PSNR, CR, SSIM AND FSIM

	Method	Block size (4×4) pixels				Block size (6×6) pixels				Block size (8×8) pixels			
		PSNR	CR	SSIM	FSIM	PSNR	CR	SSIM	FSIM	PSNR	CR	SSIM	FSIM
Lena	AMBTC	33.42	4.00	0.9901	0.9946	31.30	5.54	0.9753	0.9814	29.99	6.40	0.9587	0.9697
	MBTC	33.87	4.00	0.9901	0.9942	31.77	5.54	0.9763	0.9800	30.55	6.40	0.9614	0.9647
	ABTC-EQ	37.49	3.09	0.9944	0.9973	35.29	3.98	0.9885	0.9915	34.07	4.36	0.9826	0.9855
	Scheme A	37.49	3.24	0.9944	0.9973	35.29	4.29	0.9885	0.9915	34.07	4.80	0.9826	0.9855
	Scheme B-I	37.47	3.33	0.9945	0.9974	35.28	4.38	0.9886	0.9917	34.05	4.87	0.9827	0.9857
	Scheme B-II	37.39	3.42	0.9940	0.9970	35.21	4.47	0.9879	0.9911	34.00	4.94	0.9819	0.9849
	Scheme B-III	37.14	3.52	0.9928	0.9962	35.01	4.56	0.9862	0.9896	33.82	5.01	0.9799	0.9832
	Scheme B-IV	35.96	3.62	0.9869	0.9917	34.17	4.66	0.9787	0.9841	33.13	5.09	0.9717	0.9770
	Scheme C	39.62	3.09	0.9954	0.9979	37.40	3.98	0.9911	0.9939	36.26	4.36	0.9874	0.9907
Butterfly	AMBTC	32.26	4.00	0.9877	0.9944	30.27	5.54	0.9702	0.9818	29.09	6.40	0.9520	0.9687
	MBTC	32.66	4.00	0.9877	0.9939	30.74	5.54	0.9721	0.9806	29.61	6.40	0.9552	0.9644
	ABTC-EQ	36.26	2.99	0.9939	0.9971	34.20	3.78	0.9866	0.9908	33.04	4.09	0.9793	0.9845
	Scheme A	36.26	3.15	0.9939	0.9971	34.20	4.12	0.9866	0.9908	33.04	4.57	0.9793	0.9845
	Scheme B-I	36.24	3.25	0.9939	0.9971	34.18	4.21	0.9867	0.9910	33.02	4.64	0.9795	0.9847
	Scheme B-II	36.17	3.35	0.9935	0.9967	34.12	4.31	0.9861	0.9904	32.97	4.72	0.9786	0.9840
	Scheme B-III	35.96	3.46	0.9925	0.9961	33.96	4.42	0.9846	0.9891	32.81	4.80	0.9769	0.9823
	Scheme B-IV	34.99	3.58	0.9874	0.9922	33.24	4.53	0.9779	0.9835	32.19	4.89	0.9692	0.9759
	Scheme C	38.32	2.99	0.9951	0.9976	36.29	3.78	0.9901	0.9933	35.19	4.09	0.9852	0.9895
Cameraman	AMBTC	32.13	4.00	0.9920	0.9934	29.82	5.54	0.9782	0.9762	28.67	6.40	0.9659	0.9609
	MBTC	32.43	4.00	0.9915	0.9914	30.24	5.54	0.9795	0.9739	29.13	6.40	0.9678	0.9559
	ABTC-EQ	36.73	3.12	0.9961	0.9971	34.33	4.07	0.9914	0.9909	33.12	4.56	0.9872	0.9844
	Scheme A	36.73	3.26	0.9961	0.9971	34.33	4.37	0.9914	0.9909	33.12	4.97	0.9872	0.9844
	Scheme B-I	36.71	3.35	0.9961	0.9972	34.31	4.45	0.9916	0.9911	33.11	5.03	0.9874	0.9848
	Scheme B-II	36.65	3.44	0.9957	0.9967	34.27	4.54	0.9910	0.9903	33.07	5.10	0.9867	0.9838
	Scheme B-III	36.43	3.53	0.9946	0.9957	34.13	4.62	0.9896	0.9888	32.95	5.16	0.9851	0.9820
	Scheme B-IV	35.30	3.63	0.9891	0.9892	33.42	4.72	0.9828	0.9812	32.41	5.23	0.9779	0.9737
	Scheme C	39.05	3.12	0.9968	0.9978	36.73	4.07	0.9939	0.9944	35.56	4.56	0.9913	0.9910
Lake	AMBTC	30.48	4.00	0.9869	0.9918	28.29	5.54	0.9674	0.9747	27.30	6.40	0.9519	0.9611
	MBTC	30.93	4.00	0.9866	0.9904	28.82	5.54	0.9698	0.9714	27.94	6.40	0.9557	0.9529
	ABTC-EQ	34.68	3.04	0.9928	0.9958	32.50	3.88	0.9853	0.9882	31.55	4.25	0.9796	0.9808
	Scheme A	34.68	3.19	0.9928	0.9958	32.50	4.21	0.9853	0.9882	31.55	4.71	0.9796	0.9808
	Scheme B-I	34.67	3.29	0.9928	0.9959	32.49	4.30	0.9854	0.9883	31.54	4.78	0.9798	0.9811
	Scheme B-II	34.62	3.39	0.9925	0.9956	32.46	4.40	0.9849	0.9877	31.50	4.86	0.9792	0.9804
	Scheme B-III	34.47	3.49	0.9919	0.9949	32.34	4.49	0.9839	0.9868	31.39	4.93	0.9778	0.9787
	Scheme B-IV	33.87	3.60	0.9887	0.9922	31.86	4.60	0.9790	0.9822	30.93	5.01	0.9715	0.9730
	Scheme C	36.71	3.04	0.9940	0.9966	34.74	3.88	0.9890	0.9919	33.81	4.25	0.9853	0.9877
Peppers	AMBTC	33.57	4.00	0.9908	0.9956	31.22	5.54	0.9772	0.9826	29.73	6.40	0.9619	0.9689
	MBTC	34.05	4.00	0.9904	0.9946	31.87	5.54	0.9777	0.9802	30.49	6.40	0.9629	0.9621
	ABTC-EQ	37.59	3.17	0.9939	0.9975	35.40	4.05	0.9878	0.9915	34.07	4.39	0.9818	0.9850
	Scheme A	37.59	3.31	0.9939	0.9975	35.40	4.35	0.9878	0.9915	34.07	4.83	0.9818	0.9850
	Scheme B-I	37.57	3.39	0.9939	0.9975	35.37	4.43	0.9879	0.9916	34.03	4.89	0.9819	0.9851
	Scheme B-II	37.50	3.47	0.9936	0.9972	35.30	4.52	0.9873	0.9909	33.98	4.96	0.9811	0.9843
	Scheme B-III	37.27	3.56	0.9926	0.9963	35.12	4.61	0.9859	0.9895	33.81	5.04	0.9793	0.9822
	Scheme B-IV	36.11	3.65	0.9873	0.9916	34.25	4.70	0.9786	0.9832	33.09	5.11	0.9710	0.9749
	Scheme C	39.26	3.17	0.9946	0.9977	37.06	4.05	0.9899	0.9932	35.77	4.39	0.9856	0.9891

For simplicity, we only show experimental results for Lena. The original image is given in Fig. 1(a), and the reconstructed images from AMBTC, MBTC, ABTC-EQ, Scheme A, Scheme B-I, Scheme B-II, Scheme B-III, Scheme B-IV, and Scheme C using (4×4) blocks are, respectively, illustrated in Figs.1(b)~(j). Scheme C (Fig. 1(j)) has the best PSNR 39.62 dB.

ABTC-EQ and the proposed schemes deal with edged blocks, and thus may have the better performance near edges. The edge images of the original image Cameraman and the reconstructed images from AMBTC, MBTC, ABTC-EQ, and Scheme C are shown in Fig. 2. Obviously, the non-edge based schemes (AMBTC and MBTC) do not retain the details of selected portions, as shown in dashed circle. However, both Scheme C and ABTC-EQ have better details. Moreover, it is observed that Scheme C demonstrates more edges in the circle area than ABTC-EQ. Scheme C clearly depicts the improvement in the visual quality near edges, and its edge image is very similar to the original image.

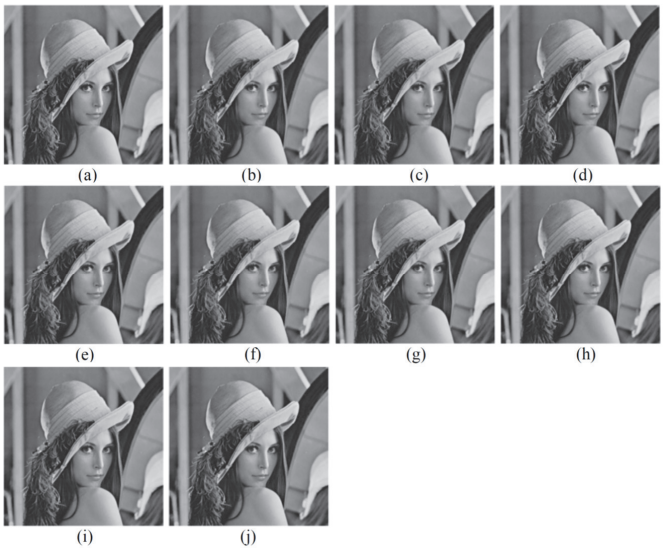

Fig. 1. The reconstructed images for Lena: (a) original image (b) AMBTC: 33.42 dB (c) MBTC: 33.87 dB (d) ABTC-EQ: 37.49 dB (e) Scheme A: 37.49 dB (f) Scheme B-I: 37.47 dB (g) Scheme B-II: 37.39 dB (h) Scheme B-III: 37.14 dB (i) Scheme B-IV: 35.96 dB (j) Scheme C: 39.62 dB.

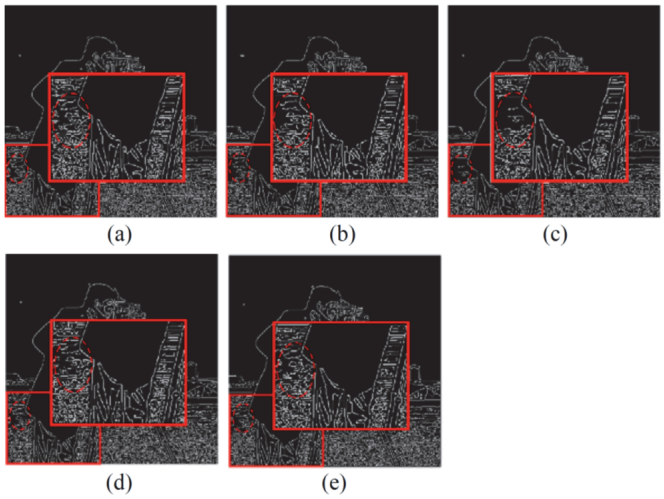

Fig. 2. Selected portion of edge image for Cameraman: (a) original image (b) AMBTC (c) MBTC (d) ABTC-EQ (e) Scheme C.

V. Conclusions

Based on variable-length coding and the observation that the difference between two quantization values is often smaller than quantization values, we design variants of ABTC-EQ to enhance PSNR and CR. Three variant schemes are given, as well as sufficient theoretical analyses. When compared with ABTC-EQ, Scheme A enhances CR and does not change the PSNR, while Scheme C enhances PSNR without reducing CR. Scheme B trades off PSNR for CR.

Acknowledgment

This work was partially supported by Ministry of Science and Technology, under Grant MOST 108-2221-E-259-009-MY2.

References

[1] E. Delp and O. Mitchell, “Image compression using block truncation coding,” IEEE Tran. On Communication, vol. 27, pp. 1335-1342, 1979.
[2] M. Lema and O. Mitchell, “Absolute moment block truncation coding and its application to color images,” IEEE Tran. on Communication, vol. 32, pp. 1148-1157, 1987.
[3] J. Mathews, M.S. Nair, and L. Jo, “Modified BTC algorithm for gray scale images using max-min quantizer,” International multi conference on automation, computing, control, communication and compressed sensing – iMac4s 2013 (India). IEEE Computer Society Press, pp. 377-82, 2013.
[4] J. Mathews and M.S. Nair, “Adaptive block truncation coding technique using edge-based quantization approach,” Computers & Electrical Engineering, vol. 43, 169-179, 2015.
[5] C.C. Chang, C.Y. Lin, and Y.H. Fan, “Lossless data hiding for color images based on block truncation coding,” Pattern Recognition, vol. 41, pp. 2347-2357, 2008.
[6] C.C. Lin and X.L. Liu, “A reversible data hiding scheme for block truncation compressions based on histogram modification,” Sixth International Conference on Genetic and Evolutionary Computing, pp. 157-160, 2012.
[7] I.C. Chang, Y.C. Hu, and W.L. Chen, and C.C. Lo, “High capacity reversible data hiding scheme based on residual histogram shifting for block truncation coding,” Signal Processing, vol. 108, pp. 376-388, 2015.
[8] S. Zhang, T. Gao, and L. Yang, “Reversible data hiding scheme based on histogram modification in integer DWT domain for BTC compressed images,” International Journal of Network Security, vol.18, No.4, pp.718-727, July 2016.
[9] J. Bai and C.C. Chang, “A high payload steganographic scheme for compressed images with Hamming code,” International Journal of Network Security, vol.18, pp. 1122-1129, 2016.
[10] C. Kim, D. Shin, and C.N. Yang, “Self-embedding fragile watermarking scheme to restoration of a tampered image using AMBTC,” Personal and Ubiquitous Computing, doi: 10.1007/s00779-017-1061-x, published online July, 2017.
[11] C. Kim, D. Shin, D, Shinn, and C.N. Yang, “A (2, 2) secret sharing scheme based on Hamming code and AMBTC,” Asian Conference on Intelligent Information and Database Systems (ACIIDS 2012), vol. LNCS 7197, pp. 129-139, 2012.
[12] C. Kim, D. Shin, Dongil Shin, R. Tso, and C.N. Yang, “A secret sharing scheme for EBTC using steganography,” Journal of Intelligent Manufacturing, vol. 25, pp 241-249, April, 2014.
[13] D. Ou, L. Ye, and W. Sun, “User-friendly secret image sharing scheme with verification ability based on block truncation coding and error diffusion,” Journal of Visual Communication and Image Representation, vol. 29, pp. 46-60, 2015.
[14] C.N. Yang, X. Wu, Y.C. Chou, and Z. Fu, “Constructions of general (k, n) reversible AMBTC-based visual cryptography with two decryption options,” Journal of Visual Communication and Image Representation, vol. 48, pp. 182-194, 2017.
[15] J. Canny, “A computational approach to edge detection,” IEEE Tran. on Pattern Analysis and Machine Intelligence, vol. PAMI-8, 679-698, 1986.
[16] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu, “An efficient k-means clustering algorithm: analysis and implementation,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, 881-92, 2002.