

Article

An Enhanced Adaptive Block Truncation Coding with Edge Quantization Scheme [†]

Ching-Nung Yang ^{1,†}, Yung-Chien Chou ^{1,†}, Tao-Ku Chang ^{1,†} and Cheonshik Kim ^{2,*,†}

¹ Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien 97401, Taiwan; cnyang@gms.ndhu.edu.tw (C.-N.Y.); 610521232@gms.ndhu.edu.tw (Y.-C.C.); tkchang@gms.ndhu.edu.tw (T.-K.C.)

² Department of Computer Engineering, Sejong University, Seoul 05006, Korea

* Correspondence: mipsan@sejong.ac.kr

† A preliminary conference version of this paper appeared under the title “Enhancing Image Compression by Adaptive Block Truncation Coding with Edge Quantization”, In Proceedings of the 2019 International Workshop on Smart Info-Media Systems in Asia (SISA 2019), Tokyo, Japan, September 2019; pp. 125–130.

‡ These authors contributed equally to this work.

Received: 22 September 2020; Accepted: 14 October 2020; Published: 20 October 2020



Abstract: Recently, image compression using adaptive block truncation coding based on edge quantization (ABTC-EQ) was proposed by Mathews and Nair. Their approach deals with an image for two types of blocks, edge blocks and non-edge blocks. Different from using the bi-clustering approach on all blocks in previous block truncation coding (BTC)-like schemes, ABTC-EQ adopts tri-clustering to tackle edge blocks. The compression ratio of ABTC-EQ is reduced, but the visual quality of the reconstructed image is significantly 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 of 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 the two quantization values is often smaller than the quantization values themselves. The second observation is that more clusters may enhance the visual quality of the reconstructed image. Based on variable-length coding and the above observations, we design variants of ABTC-EQ to enhance the visual quality of the reconstructed image and compression ratio.

Keywords: block truncation coding; edge detection; Huffman code; k-means clustering; lossy compression

1. Introduction

Rapid improvements in the area of network and information technology increase the services of digital multimedia, especially digital image, in today’s digitalized and information 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 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 block 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. That is, 16 pixels are represented as two quantization levels and a bitmap. Therefore, the compression ratio (CR) of BTC is determined from $CR = (16 \times 8)/32 = 4$ for this case. BTC provides a good compression ratio without degrading the visual quality of the 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 the 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 is 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 to 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. Afterward, 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 the reconstructed image, but has the less 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.

Compared with existing ABTC-EQ [4], MBTC [3], and AMBTC [1], the proposed enhanced ABTC-EQ method takes full advantage of the image compression and achieves a higher image quality. The main contributions of this paper are as follows: (1) The proposed method improves the compression rate by using a variable-length code to represent the index of each cluster. As a result, we can only use an average of $5/3$ bits. That is, our proposed method less compression ratio when compared with AMBTC or MBTC. (2) Increasing the number of quantization levels improves image quality, but degrades compression efficiency. For this problem, we reduce the number of bits by using the difference between the quantization levels for three or more quantization levels. (3) The proposed method enhances the visual quality of the reconstructed image by exploiting the edges of a compressed image. Variable-length coding used for our method enable to enhance the peak signal-to-noise ratio (PSNR) and CR.

The rest of this paper is organized as follows. Section 2 briefly reviews AMBTC, MBTC, and Mathews and Nair's ABTC-EQ. The proposed ABTC-EQ with theoretical analyses are formulated in Section 3. Experimental results and discussions are given in Section 4. Section 5 draws some conclusions.

2. Previous Works

2.1. AMBTC and MBTC

In [2], Lema and Mitchell propose AMBTC, the 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 $(k \times k)$ -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

$$\bar{x} = \frac{1}{k \times k} \sum_{i=1}^{k^2} x_i \quad (1)$$

where x_i denotes the i -th pixel in this block. Each pixel value x_i is compared with the mean value \bar{x} using Equation (2). If x_i is greater than or equal to \bar{x} , b_i becomes 1, otherwise b_i becomes 0. That is, a bitmap $M = [b_i]$ of the same block size which consists of two clusters is generated.

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

AMBTC preserves two quantization values per block and the higher mean and the lower mean. Equation (3) describes the method of generating two quantized values in each block. Here, t denotes the number of "1" in each bitmap M , i.e., the number of pixels under $x_i \geq \bar{x}$. $\lfloor \cdot \rfloor$ is the floor function is the function that takes as input as real number x , and gives as output the greatest integer less than or equal to x , denoted $\text{floor}(x)$, or $\lfloor x \rfloor$. The means μ_1 and μ_0 are, respectively, the higher and lower means based on \bar{x} .

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

Finally, a block of the image is compressed into two quantization levels (μ_0, μ_1) and bitmap M , i.e., 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 $(16 \times 8)/32 = 4$. Consider the example of a 512×512 -pixel image. The file size of $2M$ bits can be reduced to $0.5M$ 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 fast 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 Equation (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.

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

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.

2.2. 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 given input image by Canny edge detector [15], where the process of the detection algorithm is composed of 4 different steps: (1) smooth the image with Gaussian filter to remove the noise. (2) find the intensity gradients of the image using finite-difference approximations for the partial derivatives. (3) apply non-maximum suppression to the gradient magnitude. (4) use the

double thresholding algorithm to determine potential edges.

For a block of the edge map (bitmap) $E = [e_i]$ (sized $(k \times k)$) obtained from the given input image based on the process, we classify the block into an edge block or a non-edge block using the criteria. That is if all pixel values in the block E are '0', it classifies to a non-edge block, otherwise, it classifies to an edge block. Notes: The image created after applying the edge detector algorithm is a bitmap and includes edges representing the shape of objects. By applying this feature, the set of blocks may be classified into edge blocks or non-edge blocks.

For these two types of blocks, we use various quantization approaches. In case of 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 block 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 Equation (5).

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

The mean value μ'_i of each cluster c_i , $0 \leq i \leq 2$, is calculated with Equation (6), where $\mu'_0 \leq \mu'_1 \leq \mu'_2$. Thus, an edge image block is represented as $(\mu'_0, \mu'_1, \mu'_2, M_e)$.

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

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.

3. The Proposed ABTC-EQ

3.1. Design Concept

The advantage of ABTC-EQ [4] is to improve the quality of the reconstructed image because it can represent edge and non-edge blocks. However, it is impossible that BTC and its variants cannot represent edge block, because they only use two quantization levels, (μ_0, μ_1, M) , to represent a block. However, ABTC-EQ enhances the visual quality of the reconstructed image as well as 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. As a result of studying the ABTC-EQ, it was found that the PSNR improvement of the reconstructed image was due to the tri-clustering approach to the edge block. The weakness of this approach is that the CR decreases due to the increase of the additional bits to represent the edge block. Some in-depth observations on ABTC-EQ are listed below.

Observation 1. *It is not necessary to use two bits to represent a pixel (i.e., '00', '01', and '10') in bit map M_e for edge block.*

As shown in Equation (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 the reconstructed images.

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 a 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. We may use more clusters (say four clusters) to more precisely approximate pixel values.

3.2. The Proposed Schemes

We aim to achieve the high CR and the reasonable PSNR of the reconstructed image. For the purpose, we proposed three schemes: (1) Scheme A motivated from Observation 1, (2) Scheme B is based on Observations 1 and 2, and (3) Scheme C is based on Observations 2 and 3. Compared with Mathews and Nair’s ABTC-EQ [4], Scheme A has the same PSNR, while enhances the CR. However, Scheme C has the same CR, while enhances the PSNR. On the other hand, Scheme B further enhances the CR than Scheme A but still retain a reasonable PSNR compared with AMBTC [2] and MBTC [3].

(1) Scheme A: The algorithm is the same as that of ABTC-EQ. If a block belongs to the edge block as described in Observation 1, the bitmap (M'_e) is composed of $\{‘0’, ‘01’, ‘10’\}$ (see Equation (7)). In this case, we show the compression performance of the bitmap when a variable-length coding is applied to the bitmap. The proof is demonstrated by the Equations (8) and (9). Finally, we use $(f = 0, \mu'_0, \mu'_1, \mu'_2, M'_e)$ for an edge block.

$$b_i = \begin{cases} 0, & \text{if } x_i \in c_0, \\ 01, & \text{if } x_i \in c_1, \\ 10, & \text{if } x_i \in c_2. \end{cases} \tag{7}$$

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 $(k \times k)$ -pixel block in an image. The CR of ABTC-EQ is $CR_{MN} = \frac{8k^2}{(17+k^2)+(8+k^2) \times p_e}$, and Scheme A has the $CR_A = \frac{8k^2}{(17+k^2)+(8+(2k^2/3)) \times p_e}$, where $CR_A > CR_{MN}$. Meanwhile, they have the same PSNR of reconstructed image, i.e., $PSNR_{MN} = PSNR_A$.

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, 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 CR_{MN} and CR_A in Equations (8) and (9), respectively.

$$CR_{MN} = \left\{ \begin{array}{l} \frac{8 \times k^2}{\underbrace{(1 + 2 \times 8 + k^2) \times p_n}_{\text{non-edge block}} + \underbrace{(1 + 3 \times 8 + 2 \times k^2) \times p_e}_{\text{edge block}}} \\ = \frac{8k^2}{(17+k^2)+(8+k^2) \times p_e} \end{array} \right. \quad (8)$$

$$CR_A = \left\{ \begin{array}{l} \frac{8 \times k^2}{\underbrace{(1 + 2 \times 8 + k^2) \times p_n}_{\text{non-edge block}} + \underbrace{(1 + 3 \times 8 + (5/3) \times k^2) \times p_e}_{\text{edge block}}} \\ = \frac{8k^2}{(17+k^2)+(8+(2k^2/3)) \times p_e} \end{array} \right. \quad (9)$$

By Equations (8) and (9), since $(8 + (2k^2/3)) < (8 + k^2)$ we obviously have $CR_A > CR_{MN}$. 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., $PSNR_{MN} = PSNR_A$. Notes: In Equations (8) and (9), $(1 + 2 \times 8 + k^2)$ represents data format of the edge block, i.e., “1” is a flag bit, “ 2×8 ” is 2 pixels \times 8 bits (two quantization levels), and “ k^2 ” is $k \times k \times 1$ bits (bitmap) in non-edge block. In each block, $(1 + 3 \times 8 + 2 \times k^2)$ denotes the format of the non-edge block, i.e., “ 3×8 ” is 3 pixels \times 8 bits and “ $2 \times k^2$ ” is $k \times k \times 2$ bits in edge block. The 5/3 of Equation (9) indicates that the number of cluster is 3 and the total bit length is 5 when a block belongs to an edge block. By this approach, the size of bit map M_e is reduced from $2k^2$ to $(5/3)/k^2$. Finally, we proved that the CR performance is enhanced by Scheme A. \square

(2) Scheme B: Basically, this scheme is based on the compressed bitmap $M'_e = [b_i]$ derived from Observation 1. Moreover, as explained in Observation 2, a new compressed quantization levels $(\delta_0, \delta_1, \delta_2)$ is exploited. This is derived from the idea that compression performance can be improved by exploiting the difference between the two quantization levels. For this, we define a way of classifying the range of quantization levels into four categories. That is, for the tri-cluster $\delta_i, 0 \leq i \leq 2$, we use n_i bits with a radix $R_i : (r_{n_i}, r_{n_i-1}, \dots, r_1)$ to represent its quantization levels.

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

The new quantization levels due to the tri-cluster, $\delta_i, 0 \leq i \leq 2$, is then iteratively determined based on radix R_i with the equation $\min\{\mu'_i - \sum_{j=0}^i \delta_j\}$. Finally, we obtain new quantization levels ($f = 0, \delta_0, \delta_1, \delta_2, M'_e$) for an edge block. This process can greatly reduce the bits of the quantization level.

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 $(k \times k)$ -pixel block in an image. Scheme B has the $CR_B = \frac{8k^2}{(17+k^2)+(\sum_{i=1}^3 n_i(2k^2/3)) \times p_e}$, and $CR_{B-IV} > CR_{B-III} > CR_{B-II} > CR_A$.

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 in Equation (11).

$$\left\{ \begin{aligned} CR_B &= \frac{8k^2}{\underbrace{(1 + 2 \times 8 + k^2) \times p_n}_{\text{non-edge block}} + \underbrace{(1 + \sum_{i=1}^3 n_i + (2/3) \times k^2) \times p_e}_{\text{edge block}}} \\ &= \frac{8k^2}{(17+k^2)+(\sum_{i=1}^3 n_i - 16 + (2k^2/3)) \times p_e} \end{aligned} \right. \quad (11)$$

Via Equation (11) and four quantization ranges in Equation (10), we have compression ratios for these four quantization ranges $CR_{B-I} = \frac{8k^2}{(17+k^2)+(\sum_{i=1}^3 n_i - 16 + (2k^2/3)) \times p_e}$, $CR_{B-II} = \frac{8k^2}{(17+k^2)+(2+(2k^2/3)) \times p_e}$, $CR_{B-III} = \frac{8k^2}{(17+k^2)+(2k^2/3) \times p_e}$, and $CR_{B-IV} = \frac{8k^2}{(17+k^2)+((2k^2/3)-4) \times p_e}$. From the above and Equation (9), we have $CR_{B-IV} > CR_{B-III} > CR_{B-II} > CR_{B-I} > CR_A$. □

All compression ratios of Scheme B are larger than CR_{MN} (ABTC-EQ). By Observation 3, we may obtain the approximated (μ'_0, μ'_1, μ'_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.

(3) Scheme C: As the number of clusters for a block increases, the PSNR of an image increases proportionally like the case of Observation 3. In this scheme, four clusters (c_0, c_1, c_2, c_3) are introduced for edge blocks and the bitmap ($M''_e = [b_i]$) shown in Equation (12) is used. The mean μ''_i of each cluster $c_i, 0 \leq i \leq 3$, is calculated with Equation (13), where $\mu''_0 \leq \mu''_1 \leq \mu''_2 \leq \mu''_3$. Moreover, via Observation 2, we use a new quantization levels $(\delta'_0, \delta'_1, \delta'_2, \delta'_3)$ to represent $(\mu''_0, \mu''_1, \mu''_2, \mu''_3)$. Here, the quantization range is defined like Equation (14) and the value $\delta'_i, 0 \leq i \leq 3$ is then iteratively determined based on radix R_i with the criteria $\min\{\mu''_i - \sum_{j=0}^i \delta'_j\}$. Finally, we use new format ($f = 0, \delta'_0, \delta'_1, \delta'_2, \delta'_3, M''_e$) for an edge block. Scheme C is a method to improve image quality while maintaining the same level of compression as ABTC-EQ, and here, it is proved that the compression ratios of Scheme C and ABTC-EQ are the same for this case.

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

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

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

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 $(k \times k)$ -pixel block in an image. Scheme C has the $CR_C = \frac{8k^2}{(17+k^2)+(8+k^2) \times p_e}$, where $CR_C = CR_{MN}$.

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 CR_C in Equation (15).

$$\left\{ \begin{aligned}
 CR_C &= \frac{8 \times k^2}{\underbrace{(1 + 2 \times 8 + k^2) \times p_n}_{\text{non-edge block}} + \underbrace{(1 + \sum_{i=1}^4 n_i + 2 \times k^2) \times p_e}_{\text{edge block}}} \\
 &= \frac{8k^2}{(17+k^2)+(24-16+k^2) \times p_e} (\because \sum_{i=1}^4 n_i = 24 \text{ via Equation (14)}) \\
 &= \frac{8k^2}{(17+k^2)+(8+k^2) \times p_e} = CR_{MN}
 \end{aligned} \right. \tag{15}$$

□

3.3. Examples

An example, dealing with a (4 × 4)-pixel image block, is given in this sub section to easily understand all proposed schemes: Scheme A, Scheme B-I ~ Scheme B-IV, and Scheme C. Moreover, we will show the stored bits for this block and average mean square error (AMSE) of a single block for all proposed schemes, and the AMBTC, MBTC, and ABTC-EQ.

Suppose that a (4 × 4)-pixel image block is
$$\begin{bmatrix}
 124 & 89 & 124 & 60 \\
 135 & 114 & 120 & 86 \\
 120 & 144 & 68 & 82 \\
 100 & 104 & 55 & 78
 \end{bmatrix}.$$

Obviously, by using AMBTC [2], this block can be represented as a compressed trio (77, 123, 1010111011000100), which has to store 32 bits for this block and the AMSE is 167.56. The trio is (74, 120, 1010111011001100) when using MBTC, which differs with clustering the thirteen pixel in this block, and its AMSE is slightly reduced as 160.44.

Example 1. Compress the image block by Scheme A.

By Canny edge detector, we first find out the edge map $E =$

$$\begin{bmatrix}
 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 \\
 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0
 \end{bmatrix}.$$

According to the definition of an edge block, this image block is an edge block. We then obtain three clusters from these 16 pixels of block via k-means clustering algorithm, and assign (0), (10), and (11) for each cluster to

establish a bit map $M'_e =$

$$\begin{bmatrix}
 11 & 10 & 11 & 0 \\
 11 & 11 & 11 & 10 \\
 11 & 11 & 0 & 10 \\
 10 & 10 & 0 & 10
 \end{bmatrix}.$$

Via Equation (7), we determine $\mu'_0 = 61$, $\mu'_1 = 89$, and $\mu'_2 = 125$. Therefore, the compressed data ($f = 0, \mu'_0, \mu'_1, \mu'_2, M'_e$) is (0, 61, 89, 125, 1110110111111011110101010010) of 54 bits. Moreover, the AMSE is 77.81. Mathews and Nair’s ABTC-EQ uses ($f = 0, \mu'_0, \mu'_1, \mu'_2, M_e$) with the same (μ'_0, μ'_1, μ'_2), and thus it has the same AMSE. However, it uses the bit map M_e of 32 bits and it requires a total of 57 bits to store this block. Therefore, we showed that Scheme A provides an advantage to reduce 3-bit in compression of the block compared to ABTC-EQ.

Example 2. Compress the image block by Scheme B.

Consider using the quantization range in Scheme B-I. By the range $R_0 = (128 \ 64 \ 32 \ 16 \ 8 \ 4 \ 2)$ and the condition satisfying $\min\{\mu'_0 - \delta_0\}$, we derive $\delta_0 = 60$; by the range $R_1 = (64 \ 32 \ 16 \ 8 \ 4 \ 2 \ 1)$ and the condition satisfying $\min\{\mu'_1 - \delta'_1 - \delta_0\}$, we derive $\delta_1 = 29$; by the range $R_2 = (64 \ 32 \ 16 \ 8 \ 4 \ 2 \ 1)$ and the condition satisfying $\min\{\mu'_2 - \delta'_2 - \delta'_1 - \delta_0\}$, we derive $\delta_2 = 36$. Therefore, the compressed data ($f = 0, \delta_0, \delta_1, \delta_2, M'_e$) is (0, 60, 29,

36, 1110110111111011110101010010) of 51 bits ($\because |\delta_0| + |\delta_1| + |\delta_2| = 21$ bits). In reconstruction phase, the recovered values of are (60, 89, 125). Thus, the AMSE = 78 is slightly larger than that of Scheme A.

Consider using the quantization range in Scheme B-II. By the same argument of Scheme B-I, we may derive ($\delta_0 = 60, \delta_1 = 28, \delta_2 = 36$). The recovered values of (μ'_0, μ'_1, μ'_2) are (60, 88, 124). Thus, Scheme B-II has AMSE=80.19, but further reduces the compressed data to 48 bits.

When using quantization ranges of Scheme B-III and Scheme B-IV, we have and ($\delta_0 = 64, \delta_1 = 24, \delta_2 = 36$) and ($\delta_0 = 64, \delta_1 = 24, \delta_2 = 40$), respectively. In reconstruction, Scheme B-III and Scheme B-IV can recover (μ'_0, μ'_1, μ'_2) = (64, 88, 124) and (μ'_0, μ'_1, μ'_2) = (64, 88, 128). Finally, Scheme B-III (respectively, Scheme B-IV) has AMSE=81.69 (respectively, 82.19) and stores 45 (respectively, 42) bits for this block.

Example 3. Compress the image block by Scheme C.

By k-means clustering algorithm, we subdivide this block into four clusters, and assign (00), (01), (10), and (11)

for each cluster to establish bit map $M''_e = \begin{bmatrix} 10 & 01 & 10 & 00 \\ 11 & 10 & 10 & 01 \\ 10 & 11 & 00 & 01 \\ 10 & 10 & 00 & 01 \end{bmatrix}$. Via Equation (13), we determine $\mu''_0 = 61,$

$\mu''_1 = 83, \mu''_2 = 115$ and $\mu''_3 = 139$. Using quantization range in Scheme C, by the range $R_0 = (128\ 64\ 32\ 16\ 8\ 4)$ and the condition satisfying $\min\{\mu''_0 - \delta'_0\}$, we derive $\delta_0 = 60$; by the range $R_1 = (64\ 32\ 16\ 8\ 4\ 2)$ and the condition satisfying $\min\{\mu''_1 - \delta'_1 - \delta'_0\}$, we derive $\delta'_1 = 22$; by the range $R_2 = (32\ 16\ 8\ 4\ 2\ 1)$ and the condition satisfying $\min\{\mu''_2 - \delta'_2 - \delta'_1 - \delta'_0\}$, we derive $\delta'_2 = 33$; by the range $R_3 = (32\ 16\ 8\ 4\ 2\ 1)$ and the condition satisfying $\min\{\mu''_3 - \delta'_3 - \delta'_2 - \delta'_1 - \delta'_0\}$, we derive $\delta'_3 = 24$.

Thus, the compressed data ($f = 0, \delta'_0, \delta'_1, \delta'_2, \delta'_3, M''_e$) is (0, 60, 22, 33, 24, 100110001110100 11011000110100001) of 57 bits ($\because |\delta'_0| + |\delta'_1| + |\delta'_2| + |\delta'_3| = 24$ bits), which is the same as Mathews and Nair’s ABTC-EQ (see Example 1). In reconstruction phase, the recovered values of ($\mu''_0, \mu''_1, \mu''_2, \mu''_3$) are (60, 82, 115, 139). Thus, the AMSE is 48.13.

The AMBTC has the worst AMSE = 167.56, but it needs the least bits (32 bits) for representing a block. The above three examples imply that the AMSE = 48.13 of Scheme C is much lesser than those of other schemes (note: this significant improvement comes from the quad-clustering approach), and its number of required bits is the same as ABTC-EQ. Compared with ABTC-EQ, Scheme A has the same AMSE but has fewer bits for a block. About Scheme B, it can make a trade of the number of required bits for AMSE. For example, Scheme B-IV only needs 42 bits for a block, and meanwhile, the AMSE = 82.19 is far less than AMSE = 167.56 of AMBTC.

4. Experiment and Comparison

4.1. 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 ($k \times k$) 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 (FSIM) index, are used to compare the performance of all these schemes. Table 1 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 as ABTC-EQ. While AMBTC and MBTC have high CR, they have poor PSNR because they only use the 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 the cluster, Scheme A has a 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. Moreover, all the values of SSIM and FSIM demonstrate consistency with the performance of PSNR.

Table 1. Comparison of block truncation coding (BTC)-like Schemes on peak signal-to-noise ratio (PSNR), compression ratio (CR), structural similarity (SSIM) and feature similarity (FSIM).

Tested Image	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 Figure 1a, 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 Figure 1b–j. Scheme C (Figure 1j) has the best PSNR 39.62 dB.



Figure 1. The reconstructed images for Lena: (a) original image (b) absolute moment BTC (AMBTC): 33.42 dB (c) modified BTC (MBTC): 33.87 dB (d) adaptive block truncation coding based on edge quantization (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.

ABTC-EQ and the proposed schemes deal with edged blocks and thus may have 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 Figure 2. The non-edge based schemes (AMBTC and MBTC) do not retain the details of selected portions, as shown in the 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 depicts the improvement in the visual quality near edges, and its edge image is very similar to the original image.

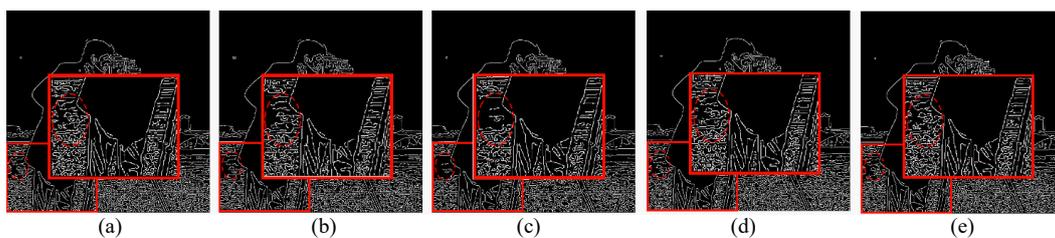


Figure 2. Selected portion of edge image for Cameraman: (a) original image (b) AMBTC (c) MBTC (d) ABTC-EQ (e) Scheme C.

4.2. Discussion

We further discuss three important issues in-depth: (i) the visual quality of reconstructed image, i.e., PSNR, (ii) the size of a compressed rate, i.e., CR, and (iii) an appropriate way of using Scheme A, Scheme B, and Scheme C for applications.

(1) Visual Quality of Reconstructed Image:

In Schemes B and C, we adopt various quantization range for each cluster. Scheme B has three quantization ranges R_i for $\delta_i, 0 \leq i \leq 2$, while Scheme C has four quantization ranges R_i for $\delta_i, 0 \leq i \leq 3$. Consider Scheme C with four ranges $R_0 = (128\ 64\ 32\ 16\ 8\ 4)$, $R_1 = (64\ 32\ 16\ 8\ 4\ 2)$, $R_2 = (32\ 16\ 8\ 4\ 2\ 1)$, and $R_3 = (32\ 16\ 8\ 4\ 2\ 1)$. By the definition $\mu''_0 = \delta'_0$, we have the quantization error ± 2 ($\because R_0$ can represent the values 0, 4, 8, ..., and 252). Because of $\mu''_1 = \delta'_0 + \delta'_1 = \mu''_0 + \delta'_1$ with minimum distortion,

if the values of quantization range R_1 can catch up the difference δ'_1 then the quantization error of μ'_1 is ± 1 ($\because R_1$ can represent the values 0, 2, 4, ..., and 126). By the same argument, we have no quantization errors for μ'_2 and μ'_3 , because R_2 and R_3 can represent the values 0, 1, 2, ..., and 63.

Therefore, when comparing with the original values of four clusters (c_0, c_1, c_2, c_3) , our recovered values have very small distortion. In Example 3, the original values are $(\mu''_0, \mu''_1, \mu''_2, \mu''_3) = (61, 83, 115, 139)$, and the recovered values are (60, 82, 115, 139), almost the same to the original one. For this case, we still use 24 bits to represent four quantization values, which are the same to ABTC-EQ using 24 bits to represent three quantization values.

By using the above analysis, we may derive quantization errors of (μ'_0, μ'_1, μ'_2) for Schemes B-I, B-II, B-III and B-IV as $(\pm 1, 0, 0)$, $(\pm 2, \pm 1, \pm 1)$, $(\pm 4, \pm 2, \pm 2)$, and $(\pm 8, \pm 4, \pm 4)$, respectively. For the original values $(\mu'_0, \mu'_1, \mu'_2) = (61, 89, 125)$, Example 2 demonstrates that the recovered values for Schemes B-I, B-II, B-III and B-IV are (60, 89, 125), (60, 88, 124), (64, 88, 124) and (64, 88, 128). These recovered results are consistent with the theoretical quantization errors. Now, we give an analysis on the probability of whether a quantization range can catch up the previous difference. For example, for the case using Scheme C, there are four clusters. On average, we may have four $(\mu''_0, \mu''_1, \mu''_2, \mu''_3) = (51, 102, 153, 204)$, and the first value μ'_0 and the differences $(\delta'_1, \delta'_2, \delta'_3)$ are no larger than 51.

The quantization ranges of Scheme C $R_0 : 0 \sim 252$, $R_1 : 0 \sim 126$, $R_2 : 0 \sim 63$, and $R_3 : 0 \sim 63$ satisfy the requirement. For a $(\delta'_0, \delta'_1, \delta'_2, \delta'_3)$, the difference δ'_1 is bounded with the value $(255 - s)$, where $s = (\mu'_0 + \delta'_2 + \delta'_3) = (\delta'_0 + \delta'_1 + \delta'_2)$. Because the quantization range $R_1 : 0 \sim 126$ is used for δ'_1 , for an extreme case which s is very small, the value of $(255 - s)$ may be larger than 126, out of range R_1 . This may give rise to large error of recovered value. As we know, k-means clustering algorithm is a good *quantization* algorithm. Therefore, for most data, we may recover the original $(\mu''_0, \mu''_1, \mu''_2, \mu''_3)$ with small distortion. Experimental results in Table 1 also confirms the above statement.

(2) Size of Compressed Rates (CRs):

Here, we deal with the enhancement of three modified ABTC-EQ schemes such as Scheme A, Scheme B, and Scheme C. Except the Scheme C (using two bits to represent four clusters), the other two schemes enhance the CR. As we know about compression technology, the CR is the most important key property. Better CR implies that compression technology has a better performance. Therefore, we showed a theoretical analysis of the estimated CRs in Theorems 1, 2, and 3. In addition, to prove the accuracy for theorem, we show a comparison of the simulation results (Table 2) and the estimated CRs derived by the Theorems. That is when k is {4,6,8} for all values of the given p_e , the expected compression ratio of the proposed method is shown in Table 2.

Moreover, the average of CRs for five test images (Lena, Butterfly, Cameraman, Lake, Pepper) are listed. Consider the CRs using Scheme A. For the case $k = 4$, the average CR of experimental values is 3.23 near the theoretical CR = 3.24 ($p_e = 0.35$). The average CR of experimental values is 4.27 (respectively, 4.78) for $k = 6$ (respectively, $k = 8$), which is near the theoretical CR = 4.27 ($p_e = 0.45$) (respectively, CR = 4.78 ($p_e = 0.50$)). This result consists with the increment of p_e for a large k . As we know, if 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. The number of edge blocks is increased when the value of k is increased, and thus the probability is increase for the large k .

Table 2. Estimated CRs of all proposed schemes with $k = 4, 6$ and 8 for $0.05 \leq p_e \leq 0.6$.

p_e	Scheme A			Scheme B-I			Scheme B-II			Scheme B-III			Scheme B-IV			Scheme C (AMBTC-EQ)		
	k=4	k=6	k=8	k=4	k=6	k=8	k=4	k=6	k=8	k=4	k=6	k=8	k=4	k=6	k=8	k=4	k=6	k=8
0.05	3.77	5.27	6.13	3.79	5.29	6.14	3.81	5.30	6.15	3.82	5.32	6.16	3.84	5.33	6.17	3.74	5.22	6.05
0.10	3.67	5.12	5.95	3.70	5.15	5.97	3.74	5.18	5.99	3.77	5.21	6.01	3.80	5.24	6.03	3.62	5.02	5.80
0.15	3.58	4.98	5.78	3.62	5.02	5.81	3.67	5.06	5.84	3.72	5.10	5.87	3.76	5.14	5.90	3.50	4.83	5.58
0.20	3.48	4.85	5.62	3.54	4.90	5.66	3.60	4.95	5.69	3.66	5.00	5.73	3.73	5.05	5.77	3.39	4.66	5.37
0.25	3.40	4.72	5.47	3.47	4.78	5.51	3.54	4.84	5.56	3.61	4.90	5.60	3.69	4.97	5.65	3.28	4.50	5.17
0.30	3.32	4.60	5.32	3.40	4.67	5.37	3.48	4.74	5.42	3.57	4.81	5.48	3.66	4.88	5.53	3.18	4.35	4.99
0.35	3.24	4.49	5.19	3.33	4.56	5.24	3.42	4.64	5.30	3.52	4.72	5.36	3.62	4.80	5.42	3.09	4.21	4.82
0.40	3.16	4.38	5.06	3.26	4.46	5.12	3.36	4.54	5.18	3.47	4.63	5.24	3.59	4.72	5.31	3.00	4.08	4.66
0.45	3.09	4.27	4.93	3.20	4.36	5.00	3.31	4.45	5.06	3.43	4.55	5.13	3.56	4.65	5.20	2.92	3.96	4.51
0.50	3.02	4.17	4.82	3.13	4.27	4.88	3.25	4.36	4.95	3.38	4.47	5.03	3.52	4.57	5.10	2.84	3.84	4.38
0.55	2.96	4.08	4.70	3.08	4.18	4.78	3.20	4.28	4.85	3.34	4.39	4.93	3.49	4.50	5.01	2.77	3.73	4.25
0.60	2.90	3.99	4.60	3.02	4.09	4.67	3.15	4.20	4.75	3.30	4.31	4.83	3.46	4.43	4.91	2.70	3.63	4.12
Avg. values	3.23	4.27	4.78	3.32	4.35	4.84	3.41	4.45	4.92	3.51	4.54	4.99	3.62	4.64	5.07	3.08	3.95	4.33

Notes: "Avg. values" show averages experimental values as generating given algorithms such as Scheme A ... Scheme C.

(3) Time Complexity:

The time complexity of all the proposed methods is an important criterion for the performance evaluation of compression algorithms. The suggested method fits this criterion very well. Because the only difference between Scheme A and the original ABTC-EQ is that Scheme A adopts Huffman code for representing three clusters, namely using (0), (10), and (11). This is a very simple Huffman code. In fact, we do not need encoding/decoding for this Huffman code in Scheme A. From another viewpoint, the original ABTC-EQ uses (00), (01), and (10) for representing three clusters, while Scheme A uses (0), (10), and (11) instead. The Huffman code in Scheme A is only used to represent the index of clusters c_0, c_1 , and c_2 , and we do not need encoding/decoding of Huffman code. At this time, the (0), (10), and (11) are to only assure the unique representation of the index of the cluster.

Scheme B uses the same approach as Scheme A, represent clusters using Huffman code, and this approach combines various quantization ranges. Quantized pixels are used as representative pixels (8 bits) of each block when the image is decompressed. We use Scheme B-I as an example for describing the quantization representation is the same as a binary representation.

For a pixel with 8 bits (b_8, b_7, \dots, b_1), its pixel value is $\sum_{i=1}^8 (b_i \times 2^{i-1})$. For Scheme B-I, the value in R_0 is $\sum_{i=1}^7 (b_i \times 2^i)$, while the values in R_1 and R_2 are $\sum_{i=1}^7 (b_i \times 2^{i-1})$ (note: R_0, R_1 and R_2 use the following quantization ranges: $R_0 : (r_7, r_6, \dots, r_1) = (128\ 64\ 32\ 16\ 8\ 4\ 2)$, $R_1 : (r_7, r_6, \dots, r_1) = (64\ 32\ 16\ 8\ 4\ 2\ 1)$, $R_2 : (r_7, r_6, \dots, r_1) = (64\ 32\ 16\ 8\ 4\ 2\ 1)$). From the above description, Scheme A and Scheme B have almost the same execution time as the original ABTC-EQ. However, for Scheme C, we use four clusters. Thus, we need to use k -means clustering algorithm to subdivide a $(k \times k)$ -pixel block into four clusters. Running a fixed number t of iterations of the standard algorithm takes $O(t \times c \times k^2)$ for k^2 pixels in a block, where c is the number of clusters.

The original ABTC-EQ and Scheme C use $c = 3$ and 4 , respectively. The proposed Scheme C uses four clusters and the quantization ranges in Equation (14). As described above the quantization approach has almost the same execution time as the original ABTC-EQ, but subdividing more clusters in every block for Scheme C is slightly greater than ABTC-EQ (note: time complexity order of clustering is $O(t \times c \times k^2)$).

(4) Appropriate Way Using Our Schemes:

Our goal is for consumers to understand and appropriately use application scenarios for the three proposed ABTC-EQ schemes. To clearly describe their respective application scenarios, we create a radar chart to illustrate the multiple performances (the number of clusters, the PSNR of a recovered image, and the size of compressed file) and the variations among all the schemes (Scheme A, Scheme B-I~IV, and Scheme C). The values of PSNR and CR are adopted from Table 1 using the test image Lena and the block size 4×4 pixels. From Figure 3, we conclude the following results showing how to appropriately use Scheme A, Scheme B-I~B-IV, and Scheme C to develop their specialties according to

applications. Because Scheme A has the same PSNR as ABTC-EQ, if we want to obtain a good PSNR along with a good CR, we should choose Scheme A. On the other hand, Scheme C has a very high PSNR. Therefore, if we want to have a significant improvement in PSNR, we may choose Scheme C, which still has the same CR as ABTC-EQ. For the application that a high CR is a requirement, we may use Scheme B to trade off the PSNR for the CR.

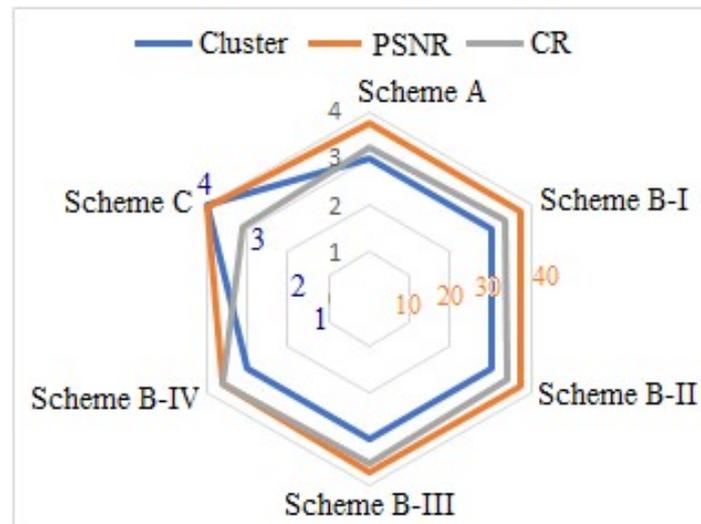


Figure 3. Radar chart for the proposed schemes using the variables of the number of clusters, the PSNR of recovered image, and the CR.

5. Conclusions

In this paper, we propose a method to improve the compression performance of ABTC-EQ, a method that overcomes the edge loss problem of the existing BTC-like image. It is very reasonable that the tri-clustering approach is preferable to the bi-clustering approach for the quality of the image. On the other hand, the problem of introducing the tri-clustering approach is that the file size increases. In this paper, by introducing variable-length coding, we found a method that satisfies both the image compression and image quality. In addition, we have mentioned in detail in the paper a sufficient theoretical analysis of the proposed method. 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. From these properties, we demonstrate how to properly use these schemes for intended applications. Moreover, experimental results are given to illustrate the effectiveness and advantages of the proposed schemes.

Author Contributions: Conceptualization, C.-N.Y.; writing—original draft preparation, C.-N.Y.; Writing—review & editing, C.K.; Validation, C.-N.Y., Y.-C.C., and C.K.; formal analysis, C.-N.Y.; methodology, C.-N.Y., Y.-C.C., T.-K.C., and C.K.; data curation, T.-K.C.; funding acquisition, C.-N.Y., C.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by Ministry of Science and Technology (MOST), under Grant 108-2221-E-259-009-MY2 and 109-2221-E-259-010, and by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by (2015R1D1A1A01059253), and was supported under the framework of international cooperation program managed by NRF (2016K2A9A2A05005255).

Acknowledgments: Thank you to the reviewers who reviewed this paper and the MDPI editor who edited it professionally.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Delp, E.; Mitchell, O. Mitchell: Image compression using block truncation coding. *IEEE Trans. Commun.* **1979**, *27*, 1335–1342. [[CrossRef](#)]

2. Lema, M.; Mitchell, O. Absolute moment block truncation coding and its application to color images. *IEEE Trans. Commun.* **1984**, *32*, 1148–1157. [[CrossRef](#)]
3. Mathews, J.; Nair, M.S.; Jo, J. Modified BTC algorithm for gray scale images using max-min quantizer. In *Proceedings of the International Multi Conference on Automation, Computing, Control, Communication and Compressed Sensing—iMac4s, Kottayam, India, 22–23 March 2013*; IEEE Computer Society Press: Los Alamitos, CA, USA, 2013; pp. 377–382.
4. Mathews, J.; Nair, M.S. Adaptive block truncation coding technique using edge-based quantization approach. *Comput. Electr. Eng.* **2015**, *43*, 169–179. [[CrossRef](#)]
5. Chang, C.C.; Lin, C.Y.; Fan, Y.H. Lossless data hiding for color images based on block truncation coding. *Pattern Recognit.* **2008**, *41*, 2347–2357. [[CrossRef](#)]
6. Lin, C.; Liu, X. A reversible data hiding scheme for block truncation compressions based on histogram modification. In *Proceedings of the 2012 Sixth International Conference on Genetic and Evolutionary Computing, Kitakushu, Japan, 25–28 August 2012*; pp. 157–160.
7. Chang, I.C.; Hu, Y.C.; Chen, W.L. High capacity reversible data hiding scheme based on residual histogram shifting for block truncation coding. *Signal Process.* **2015**, *108*, 376–388. [[CrossRef](#)]
8. Zhang, S.; Gao, T.; Yang, T.L. Reversible data hiding scheme based on histogram modification in integer DWT domain for BTC compressed images. *Int. J. Netw. Secur.* **2016**, *18*, 718–727.
9. Bai, J.; Chang, C.C. A high payload steganographic scheme for compressed images with Hamming code. *Int. J. Netw. Secur.* **2016**, *18*, 1122–1129.
10. Kim, C.; Shin, D.; Yang, C.N. Self-embedding fragile watermarking scheme to restoration of a tampered image using AMBTC. *Pers. Ubiquitous Comput.* **2018**, *22*, 11–22. [[CrossRef](#)]
11. Kim, C.; Shin, D.; Yang, C.N. A (2, 2) secret sharing scheme based on Hamming code and AMBTC. In *Proceedings of the Asian Conference on Intelligent Information and Database Systems (ACIIDS 2012), Kaohsiung, Taiwan, 19–21 March 2012*; Volume 7197, pp. 129–139.
12. Kim, C.; Shin, D.; Shin, D.; Tso, R.; Yang, C.N. A secret sharing scheme for EBTC using steganography. *J. Intell. Manuf.* **2014**, *25*, 241–249. [[CrossRef](#)]
13. Ou, D.; Ye, L.; Sun, W. User-friendly secret image sharing scheme with verification ability based on block truncation coding and error diffusion. *J. Vis. Commun. Image Represent.* **2015**, *29*, 246–60. [[CrossRef](#)]
14. Yang, C.N.; Wu, X.; Chou, Y.C.; Fu, Z. Constructions of general (k, n) reversible AMBTC-based visual cryptography with two decryption options. *J. Vis. Commun. Image Represent.* **2017**, *48*, 182–194. [[CrossRef](#)]
15. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *6*, 182–194. [[CrossRef](#)]
16. Kanungo, T.; Mount, D.M.; Netanyahu, N.S.; Piatko, C.D.; Silverman, R.; Wu, A.Y. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 881–892. [[CrossRef](#)]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).