Journal of Information Hiding and Multimedia Signal Processing ©2016 ISSN 2073-4212
Ubiquitous International Volume **7**, Number **1**, January **2016**

An Image Compression Method Based on Block Truncation Coding and Linear Regression

Wan-Li Lyu[1,2] and Chin-Chen Chang[2,3]

[1]Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education
School of Computer Science and Technology
Anhui University, Hefei 230039, China

[2]Department of Information Engineering and Computer Science
Feng Chia University
100 Wenhwa Rd., Seatwen, Taichung 40724, Taiwan, ROC

[3]Department of Computer Science and Information Engineering
Asia University, Taichung 41354, Taiwan, ROC
(wanly_lv@163.com; alan3c@gmail.com)

Received December, 2014; revised September, 2015

Abstract. *This paper describes an image compression method based on block truncation coding (BTC) and linear regression coding (LRC) hybrid strategy. BTC is simple and easy to use, but only two representative values can not adequately represent the visual feature of image blocks. However, LRC can obtain different values of points with the line function by recording the coefficient of the regression line while the data distribution shows a linear relationship. The proposed method designs a hybrid strategy to combine the advantages of the two methods to improve image quality. In the paper, first, models are designed to rearrange the data of image blocks to satisfy a linear distribution, and then these models are used to code the image blocks. The experimental results indicated that the proposed method is superior to the BTC scheme in terms of image quality, and it can be used for image compression applications.*
Keywords: Block truncation coding, Image compression, Linear regression coding, Absolute moment block truncation coding.

1. **Introduction.** An image compression technique can represent an image with a smaller number of bits without introducing appreciable degradation of the visual quality of the decompressed image. The BTC algorithm is a simple, block-based, spatial-domain, image compression technique developed by Delp and Mitchell [1]. The basic idea of BTC is to substitute values of pixels by the high or low means of their blocks. Because BTC is simple and easy to use, there has been extensive interest in its continued development and application for image compression. Subsequently, many compression techniques developed have been inspired by BTC. Absolute moment block truncation coding (AMBTC) [2] preserves the higher mean and lower mean of each block and uses these quantities to quantize output, providing better image quality than image compression using BTC. Kamel et al. [3] proposed a variable block truncation coding method that can be applied hierarchically using various sizes of blocks.
However, BTC has a fundamental limit in that each block is reconstructed by only two representative values, and the annoying blocking effect and false contour has harassed researchers. The authors of [4] proposed an improved BTC image compression method

using a fuzzy complement edge operator. Many halftoning-based BTCs have been proposed [5, 6], however these methods introduce another impulse noise issue. Many research efforts have been focused on determining the use of the hybrid coding method to provide better image quality. Udpikar and Raina [7] used vector quantization (VQ) to further compress the overhead information of the BTC outputs. Wu and Coll [8] applied a hybrid coding model by using VQ to rapidly encode the bitmap and DCT to encode the high or low mean. Haung and Lin [9] used universal Hamming codes and a differential pulse code modulation (DPCM) to code the bit plane and reduce the bit rate and computational complexity. Chang and Hu [10] combined the VQ method and the BTC method to improve image quality and keep a low bit rate. In [11, 12], the BTC method was used for DPCM image coding, referred to as the DPCM-BTC method. In [13], an ordered dither block truncation coding (ODBTC) method was presented that sends the compressed information progressively. In [14], the proposed scheme is a progressive scheme based on BTC and pattern fitting (PBTC-PF)that transmits the most significant information to the receiver.

Linear regression [15] consists of determining the best-fitting straight line through the points. The best-fitting line is called a regression line. The linear regression method is usually used in the pattern recognition field [16-20]. The LRC algorithm has the advantage of allowing us to record the coefficients of the regression line and uses the line function to obtain the values of points. BTC is simple and easy to use, but only two representative values can not well represent the visual feature of image blocks. However, LRC can obtain different values of points with the line function by recording the coefficient of the regression line while the data distribution shows a linear relationship.

In this research, we propose a fairly simple, but efficient, hybrid strategy based on combining the advantages of BTC and LRC to improve image quality. The rest of the paper is organized as follows. Section 2 introduces block truncation coding in image compression and the linear regression method. In Section 3, our image compression technique based on the BTC and LRC hybrid strategyis described in detail. The experimental results are reported in Section 4. Our conclusions are presented in Section 5.

2. **Related Work.** In this section, we review the BTC and LRC methods to provide some background knowledge related to our new scheme.

2.1. **Review of the BTC Method.** In the traditional BTC method [1], an image is divided into non-overlapped blocks of size $n \times n$, which is always an integer that is a power of 2, and B_k is a block of the image, as shown in Fig.1.

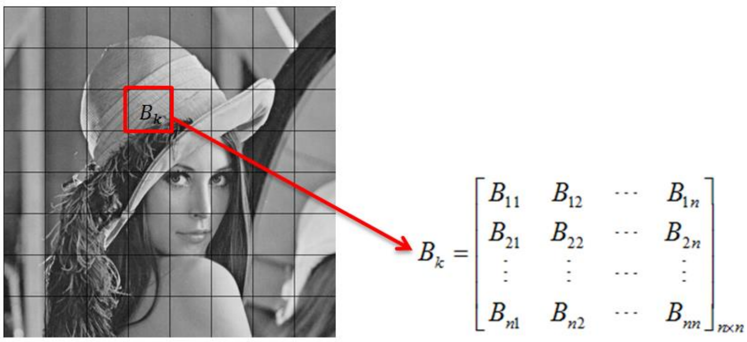

FIGURE 1. Image divided into non-overlapped blocks

Three elements are needed for each block, i.e., threshold x_{th}, max value h and min value l. If the x_{th} is chosen to be the mean value of the block, then the block will be divided into two parts, i.e., one part in which the pixel values are equal to or greater than x_{th} and one part in which the values are smaller than x_{th}. A bit map of the block with size of $n \times n$ is constructed and each element is a binary number with 0 denoting the pixel value in the position in the block that is smaller than x_{th} and 1 is greater than or equal to x_{th} Then the binary bitmap and the two values h and l will compose the compressed block. The values h and $l$are given by Equations (1) and (2):

$$h = x_{th} + \sigma\sqrt{\frac{n^2 - q}{q}} \tag{1}$$

$$l = x_{th} - \sigma\sqrt{\frac{q}{n^2 - q}} \tag{2}$$

where the standard deviation $\sigma$is given by Equation (3):

$$\sigma = \sqrt{\frac{1}{n \times n}\sum_{i=1}^{n}\sum_{j=1}^{n}(B_{ij}^2 - x_{th}^2)} \tag{3}$$

Here $B_{ij}, (1 \le i \le n,\ 1 \le j \le n)$ is the pixel value of the block B_k, and q is the number of pixels in the block whose pixel values are greater than or equal to x_{th}.

We use the sum of squared errors (SSR) to record the distance between B_k and B'_k, which is given by Equation (4):

$$SSR(B'_k, B_k) = \sum_{i=1}^{n}\sum_{j=1}^{n}(B'_{ij} - B_{ij})^2 \tag{4}$$

The SSR can show the difference between the original block and reconstructed block. If the value of SSR is bigger, the visual quality of reconstructed block is poorer. Fig. 2 shows an example of the BTC block compression and reconstruction procedure. In the BTC block, $n = 4$, $x_{th} = 151.75$, $h = 156$, and $l = 147$. The sum of the squared errors is $SSR(B'_k, B_k) = 207$. In the compression procedure, the binary bitmap, values h and

$$B_k = \begin{bmatrix} 149 & 141 & 141 & 147 \\ 148 & 151 & 150 & 152 \\ 153 & 152 & 159 & 152 \\ 158 & 161 & 155 & 159 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \Rightarrow B'_k = \begin{bmatrix} 147 & 147 & 147 & 147 \\ 147 & 147 & 147 & 156 \\ 156 & 156 & 156 & 156 \\ 156 & 156 & 156 & 156 \end{bmatrix}$$

(a) 4×4 block B_k (b) binary bitmap (c) reconstructed block B'_k

Figure 2. Example of a block compressed with the BTC method

l, are sent to the receiver. In Fig.1, the block before the BTC compression occupied $4 \times 4 \times 8 = 64$ bits, and, after compression, the block occupied $4 \times 4 \times 1 + 8 + 8 = 32$ bits.

Fig. 3 shows the use of the BTC method to organize the 32-bit unit to store the compressed data, i.e., the high value h, low value l, and the binary bitmap.

The image compression technique represents an image with a smaller number of bits, and, at the same time, it does so without introducing appreciable degradation of the visual quality of the decompressed image. The BTC method uses two integers between 0 and 255 to express 16 integers that also are between 0 and 255, which would lead to distortion especially in cases where the values of the pixels were different from each other. The aim of this paper is to determine a better way of expressing the pixel values of a block.

	31	30	29	28	27	26	25	24
Max value h	1	0	0	1	1	1	0	0

	23	22	21	20	19	18	17	16
Min value l	1	0	0	1	0	0	1	0

Bitmap	15	14	13	12	11	10	9	8
(part one)	0	0	0	0	0	0	0	1

Bitmap	7	6	5	4	3	2	1	0
(part two)	1	1	1	1	1	1	1	1

FIGURE 3. Organization of the 32-bit compressed unit with the BTC method

2.2. **Linear Regression Method.** In statistics, linear regression is an approach for modeling the relationship between two sets, X and Y, in which one or more explanatory variables belong to X. The case of one explanatory variable, which is denoted as x, is called simple linear regression, and $f(x)$ is a mapping from X into Y. Linear regression consists of determining the best-fitting straight line through the points, and this line is called the regression line.
The image is divided into non-overlapping blocks of size $n \times n$, and B_k is a block of the image, as shown in Fig.1. If the pixels in B_k are rearranged in regular rows, a row vector $\{B_{11}, B_{12}, \cdots, B_{nn}\}$ of the block B_k can be obtained. We use the formula in Equation 5 to express the row vector $\{B_{11}, B_{12}, \cdots, B_{nn}\}$:

$$f(x) = ax + r + \varepsilon, \tag{5}$$

where $x = 1, 2, \cdots, n \times n$, $f(x) = \{B_{11}, B_{12}, \cdots, B_{nn}\}$, a and r are coefficients, ε is residual, and $\varepsilon \sim N(0, \sigma^2)$ is a normally-distributed, random variable, and expectation of ε is $E(\varepsilon) = 0$. The values of a and r can be obtained by Equations (6) and (7), respectively.

$$a = \frac{n \times n \sum_{i=1}^{n\times n} x_i f(x_i) - (\sum_{i=1}^{n\times n} x_i)(\sum_{i=1}^{n\times n} f(x_i))}{n \times n \sum_{i=1}^{n\times n} x_i^2 - (\sum_{i=1}^{n\times n} x_i)^2}, \tag{6}$$

$$r = \overline{f(x)} - a\overline{x}. \tag{7}$$

So, when a and r are obtained, a regression line can be constructed by Equation (8).

$$f''(x) = ax + r, \tag{8}$$

where $x = \{1, 2, \cdots, n \times n\}$.

By rebuilding row vector $f''(x)$ to n rows and n columns, the decompression block B''_k can be obtained. The pixel values of a block can be reconstructed by a, r and regression line function $f''(x)$. The range of a is $\tan(-\frac{\pi}{2}) < a < \tan(\frac{\pi}{2})$, and r is the expected value of the regression line $f''(x)$ when $x = 0$. Values a and r are two real numbers. If the angle of the regression line, $\theta = \arctan(a)$, is recorded, then the range of θ is from $-90°$ to $90°$. Because $a = \tan(\theta)$ is a periodic function, the θ can be obtained by Equation (9):

$$\theta = \begin{cases} (-\pi + \tan^{-1} a) \times \dfrac{180}{\pi}, & a \geq 0 \\ \tan^{-1} a \times \dfrac{180}{\pi}, & a < 0 \end{cases}, (-180 \leq \theta < 0, \text{and } \theta \neq -90). \tag{9}$$

Because the regressed pixels' values are integers and the range is from 0 to 255, expressing r with two decimal pointsis adequate. For example, assume the data of the block $\begin{bmatrix} 149 & 141 & 141 & 147 \\ 148 & 151 & 150 & 152 \\ 153 & 152 & 159 & 152 \\ 158 & 161 & 155 & 159 \end{bmatrix}$, then the row vector $(x) = \{149, 141, 141, 147, 148, 151, 150, 152, 153, 152, 159, 152, 158, 161, 155, 159\}$, $x = 1, 2, \cdots, 16$. With the help of linear regression, we can obtain the coefficients $a = (0.82)_{10}$, $\theta = \tan^{-1} a \times \frac{180}{\pi} = (39)_{10} = (100111)_2$ and $r = (143.74)_{10} = (10001111.10111101)_2$. Then, we can reconstruct the equation $f''(x) = \{145, 145, 146, 147, 148, 149, 149, 150, \ 151, \ 152, \ 153, \ 154, 154, 155, 156, 157\}$, reconstruct block $B_k'' = \begin{bmatrix} 145 & 145 & 146 & 147 \\ 148 & 149 & 149 & 150 \\ 151 & 152 & 153 & 154 \\ 154 & 144 & 156 & 157 \end{bmatrix}$, and the sum of squared errors $SSR(B_k'', B_k) = 167$.

Fig. 4 shows the use of the linear regression method to organize the 32-bit unit to store the compressed data, i.e., coefficients θ and r.

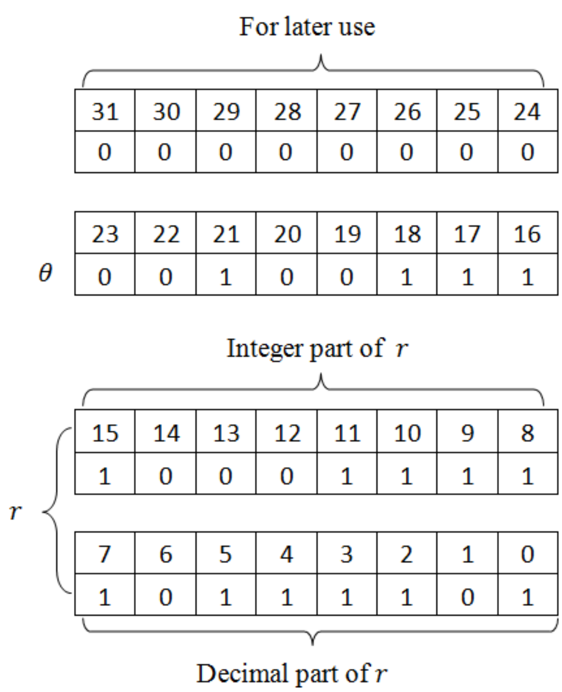

FIGURE 4. Organization of the 32-bitcompressed unit with the LRC method

Fig. 5 shows the difference figure of the original row vector data $f(x)$, the BTC-reconstructed block B_k' rearranged in regular rows $f'(x)$, and the linear regressed row vector data $f''(x)$. The MSE (mean square error) for a reconstructed gray level image I'' is defined by Equation (10):

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (I(i,j) - I''(i,j))^2 \tag{10}$$

where $M \times N$ is the total number of pixels of the original images I.

Use Equation (10) to calculate the MSE of BTC reconstructed block B_k' and LRC reconstruct block B_k''. $MSE(B_k, B_k') = 12.94$, $MSE(B_k, B_k'') = 10.44$. When the data

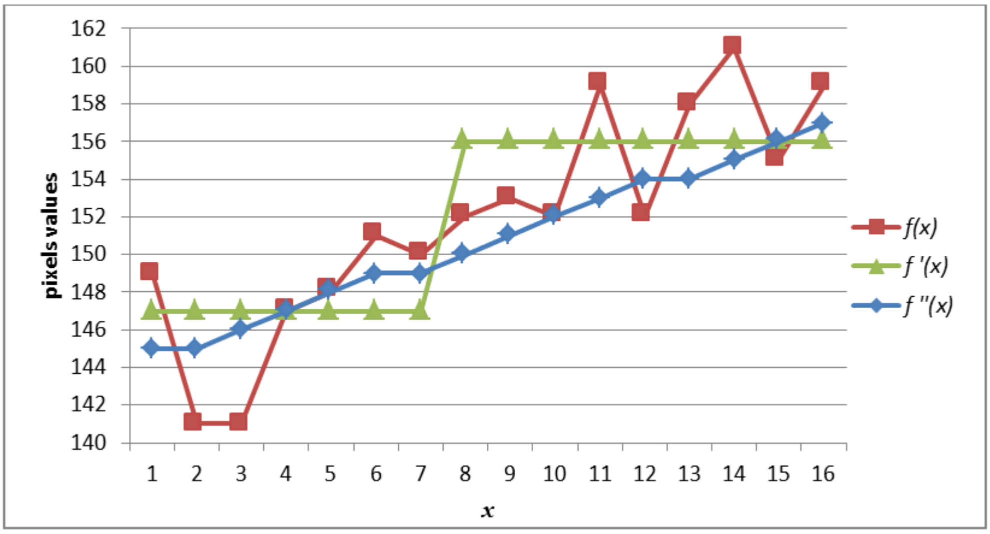

FIGURE 5. Difference figure of original row vector data $f(x)$, BTC reconstructed block $B'(x)$ rearranged in regular rows $f'(x)$, and linear regressed row vector data $f''(x)$s

distribution of a block shows a linear relationship, using LRC to compress a block can reduce the block effect incurred by BTC.

3. **Proposed Image Compression Method Based on the BTC and LRC Hybrid Strategy.** The proposed image compression method aims at uniting the advantages of BTC and LRC compression strategy to determine an effective way of better exressing the imagespixel values.

3.1. **Linear Regression Compression Method.** The data distribution influences the results of linear regression. Using only one model to rearrange the data of block B_k cannot satisfy the linear distribution of the data. Therefore, developing models to rearrange the data of block B_k to satisfy the linear distribution of the data is necessary.

For a pixel p in an image, there is also a strong correlation between the pixels surrouding pixel p. To gain row vectors, we designed 10 visited models to reorganize pixels within the 4×4 blocks. Fig. 6 shows these 10 visited models. In this section, we use four final models from the 10 visited models for later LRC compression.We use variables to sign the 10 visited models, so the range of s is $1 \leq s \leq 10$. For a 4×4 block in an image, if we start from the top-left of the block, the next pixel is adjacent to the pixel that was recently visited and not repeated, so we can gain the row vectors of the block, as Figs. 6 (a) to (d) show. Fig. 6 (e) shows the visited-by-row model, and Fig. 6 (f) shows the visited-by-column model. Figs. 6 (g) to (i) show the gradual changes in the luminance in the diagonal direction, the longitudinal direction, and the horizontal direction of the visited models of the block. Fig. 6 (j) shows the zigzag route starting from the bottom-left of the block. Fig. 7 shows the changes in the pixels luminance. With the visited models, the pixels in the 4×4 block can be organized into a row vector and a different visit model can make a different row vector. Fig.8 shows the flow chart of using the linear regression coding method to compress a block B_k of an image. In Fig. 8, t is used to mark the model number, and, with this model, the values of the regressed pixels have minimum SSR with original values between all 10 models. The LRC compression algorithm of an image is described as Algorithm 1. With Algorithm 1, we can count the frequency of the 10 travel models using linear regression coding to compress an image.

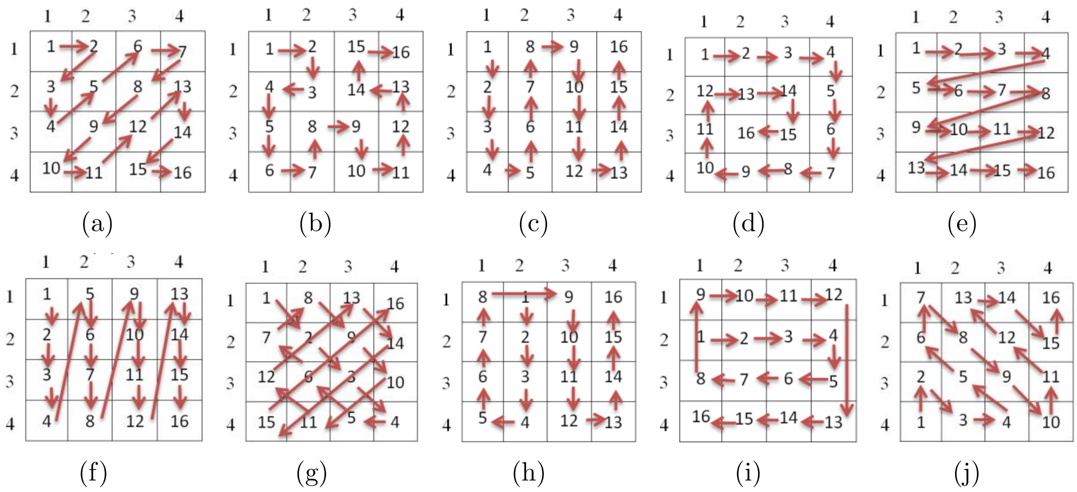

FIGURE 6. The 10 commonly-visited models: (a) Zigzag route;(b) Hilbert route; (c) W route;(d) Convolution route; (e) Arranged in regular rows; (f) Arranged in regular columns; (g) Slashes direction gradual change; (h)Longitudinal direction gradual change; (i) Horizontal direction gradual change; (j) Zigzagroute start from bottom-left

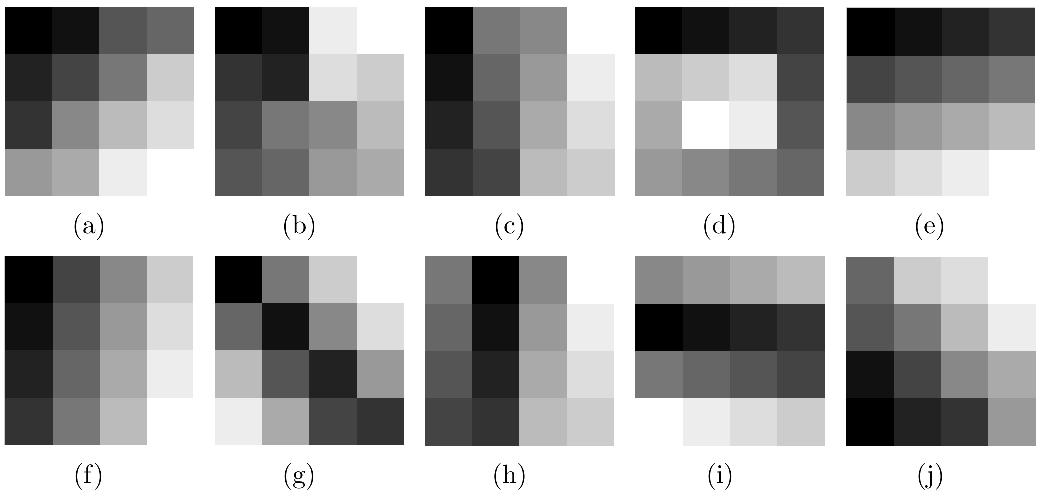

FIGURE 7. Step-by-step changes in the luminance of the pixels in the 10 visited models: (a) Zigzag route;(b) Hilbert route; (c) W route; (d) Convolution route; (e)Arranged in regular rows; (f) Arranged in regular columns; (g) Diagonal direction gradual change; (h) Longitudinal direction gradual change; (i) Horizontal direction gradual change; (j) Zigzag route starting from bottom-left

Here, we use 32 bits to store U_k, including the travel model serial number t. The organization of the 32-bit compressed unit in Algorithm 1 is shown in Fig.9.

The corresponding image linear regression coding decompression algorithm is described as Algorithm 2. We used 12 test images, the sizes of which were 512×512 as shown in Fig.13 in Section 4, to count the frequency of the 10 visited models to determine four common visited models for use in coding the later proposed method. Table 1 shows the statistical results.

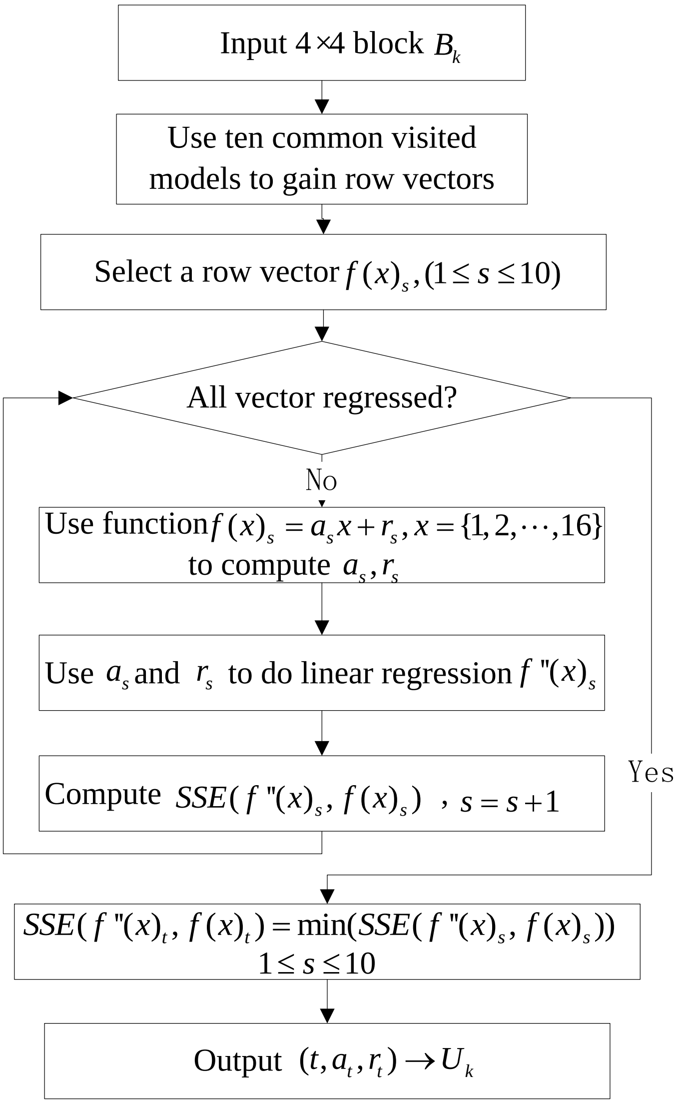

FIGURE 8. Flowchart using the linear regression coding method to compress a block B_k of an image

---

Algorithm 1 Linear regression coding compression algorithm with 10 travel models.

Input: Gray image I, which has the size of $M \times N$, and both M and N are exactly divisible by 4

Output: Compressed linear regression coding (LRC) $U = \{U_1, U_2, \cdots, U_{\frac{M}{4} \times \frac{N}{4}}\}$ of image I

1: Step 1. Segment image I into 4×4 blocks, $B_1, B_2, \cdots, B_{\frac{M}{4} \times \frac{N}{4}}$.

2: Step 2. If all of the blocks of image I are compressed, go to Step 4, else select an uncompressed block B_k, and goto Step 3.

3: Step 3. Use the linear regression coding method to compress block B_k, as shown in Fig.8; code and store U_k. Goto Step 2.

4: Step 4. Out put $U = \{U_1, U_2, \cdots, U_{\frac{M}{4} \times \frac{N}{4}}\}$.

Table 1 shows the frequency of 10 visited model in 12 test images.

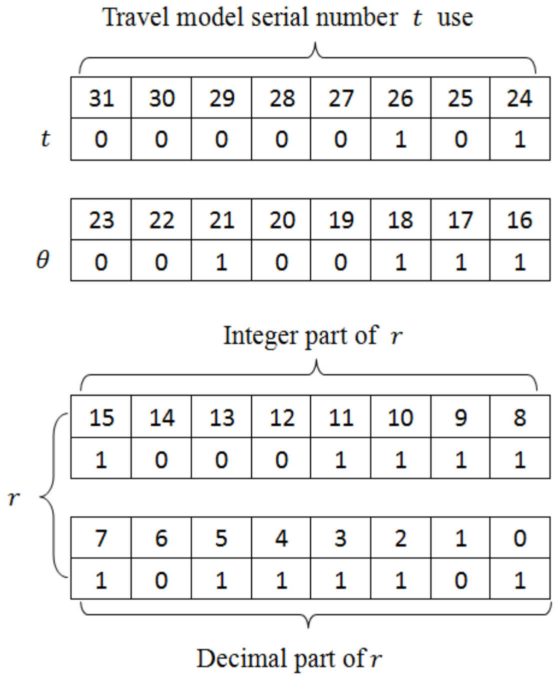

FIGURE 9. Organization of the 32-bit compressed unit in Algorithm 1

Algorithm 2 Linear regression coding decompression algorithm with 10 travel models.

Input: linear regression coding (LRC) $U = \{U_1, U_2, \cdots, U_{\frac{M}{4}\times\frac{N}{4}}\}$ of image I, which has $\frac{M}{4} \times \frac{N}{4} \times 32$ bits

Output: Gray image I, which has the size of $M \times N$

Step 1. Segment the linear regression coding U into 32-bit units $U_1, U_2, \cdots, U_{\frac{M}{4}\times\frac{N}{4}}$.

Step 2. If all units of U are decompressed, goto Step 5, else select an untreated unit $U_k = (t, a_k, r_k)$, goto Step 3.

Step 3. Reconstruct a regression line with (a_k, r_k) and Equation (8), where $x = 1, 2, \cdots, 16$. Get a regression vector $f''(x)_k$.

Step 4. Use the $t(1 \le t \le 10)$ model shown in Fig.6 to rearrange $f''(x)_k$ to 4×4 phalanx B''_k, then goto Step 2.

Step 5. Rearrange $\{B''_1, B''_2, \cdots, B''_{\frac{M}{4}\times\frac{N}{4}}\}$ to image I and output.

Fig.10 shows counts of top four used visited model in 12 test images and models (c), (e), (f), and (j) were used more frequently in gray images. Using only these four models, we can code modes (c), (e), (f), and (j) with 00, 01, 10, and 11.

3.2. **The Coding of Image Compression Blocks Based on the BTC and LRC Hybrid Strategy.** The proposed image compression method combines the advantages of BTC and LRC compression strategy, so an efficacious hybrid coding method is very important.

Using BTC, a block B_k should record h, l, and a 16-bit binary bitmap as compression data. Using LRC, θ, r and visited model serial number t should be recorded as compression data. Because we used only four models in the hybrid strategy, $0 \le t \le 3$. We use a 32-bit unit to store the hybrid strategy compressed data, and we mark the 32 bits with $U_k = [u_1, u_2, u_3, u_4]$, and each $u_v(1 \le v \le 4)$ occupies eight bits of storage space.

TABLE 1. Frequency of 10 visited model in 12 test images

Model Frequency Image	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
Lena	2438	1170	2785	627	1898	3690	364	453	330	2629
Tiffany	2061	1581	1851	1512	1736	2209	645	1223	751	2815
Zelda	1675	1845	2477	740	1873	3091	440	1574	167	2502
Boat	1335	1522	1844	1243	3098	2056	771	1815	440	2260
Barbara	2104	1186	2539	611	2065	2986	889	1120	297	2587
Bridge	1770	1974	1348	1858	3066	1853	424	656	1064	2371
Baboon	1748	1524	1314	2286	2392	1858	542	452	1562	2706
Elaine	1702	1752	1332	2633	1754	1787	816	654	986	2968
Pepper	1593	1807	1648	1845	1820	2397	862	915	764	2733
Goldhill	1272	1979	1302	1967	3332	1666	452	840	1141	2433
Airplane	1763	2019	1450	1454	2848	2090	446	731	628	2955
Couple	1466	1443	2148	901	3326	3427	239	619	547	2268
Average	1744	1650	1837	1473	2434	2426	574	921	723	2602

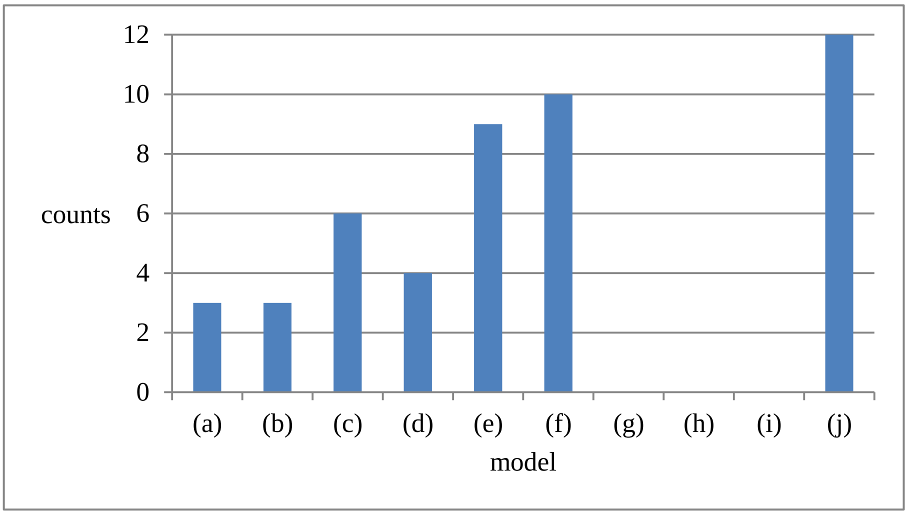

FIGURE 10. The counts of top four used visited model in 12 test images

Fig. 11 shows an example of using BTC and LRC hybrid strategy to organize the 32-bit unit to store the compressed data.

3.3. **Image Compression Procedure Based on BTC and LRC Hybrid Strategy.** Because the aim of this paper was to determine how to better express the pixel values of a block, the basic idea of the proposed image compression method based on BTC and LRC hybrid strategy was to start with an original image block and use BTC if it provided good performance or to use LRC if it provided good performance. Fig. 12 shows the flowchart of the image compression precedure based on BTC and LRC hybrid strategy.

3.4. **Image Decompression Procedure Based on BTC and LRC Hybrid Strategy.** In the proposed image decompression precedure, first, the compressed data U are divided with 32 binary bit length in every unit. The 32-bit unit is marked with $U_k = [u_1, u_2, u_3, u_4]_k$, and each $u_v, (1 \leq v \leq 4)$ occupies eight bits. If $u_1 = (00000000)_2$ and

Algorithm 3 The block B_k coding algorithm based on BTC and LRC hybrid strategy.

Input: Flag, (θ, r, t) or $(h, l, bitmap)$

Output: 32-bit compressed data U_k of block B_k

1: Step 1. Initialize $U_k = [u_1, u_2, u_3, u_4]_k$ with 32 bits 0, and each $u_v (1 \leq v \leq 4)$ occupies eight bits.

2: Step 2. If Flag = 0, use Equation (11) to generate compressed data U_k.

$$U_k = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}_k = \begin{bmatrix} (00000000)_2 \\ (\theta)_2 \\ (\text{integer part of } r)_2 \\ (\text{decimal part of } r)_1 || (t)_2 \end{bmatrix}_k \tag{11}$$

where the || in Equation (10) is the concatenation operation.

3: Step 3. If Flag = 1, use Equation (12) to generate compressed data U_k.

$$U_k = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}_k = \begin{bmatrix} (h)_2 \\ (l)_2 \\ (\text{bitmap part } 1)_2 \\ (\text{bitmap part } 2)_2 \end{bmatrix}_k \tag{12}$$

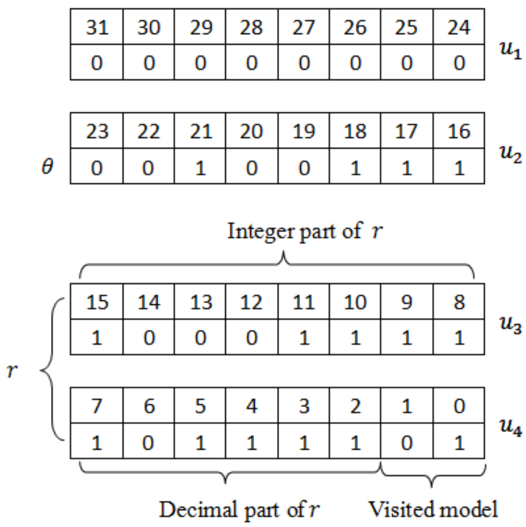

FIGURE 11. Organization of the 32 bits of the block compressed with the BTC and LRC hybrid strategy

$u_2 \neq (00000000)_2$, use the LRC method to decompress the U_k, else use the BTC method to decompress the U_k. Algorithm 4 decompressed the block U_k, which was compressed using the BTC and LRC hybrid strategy.

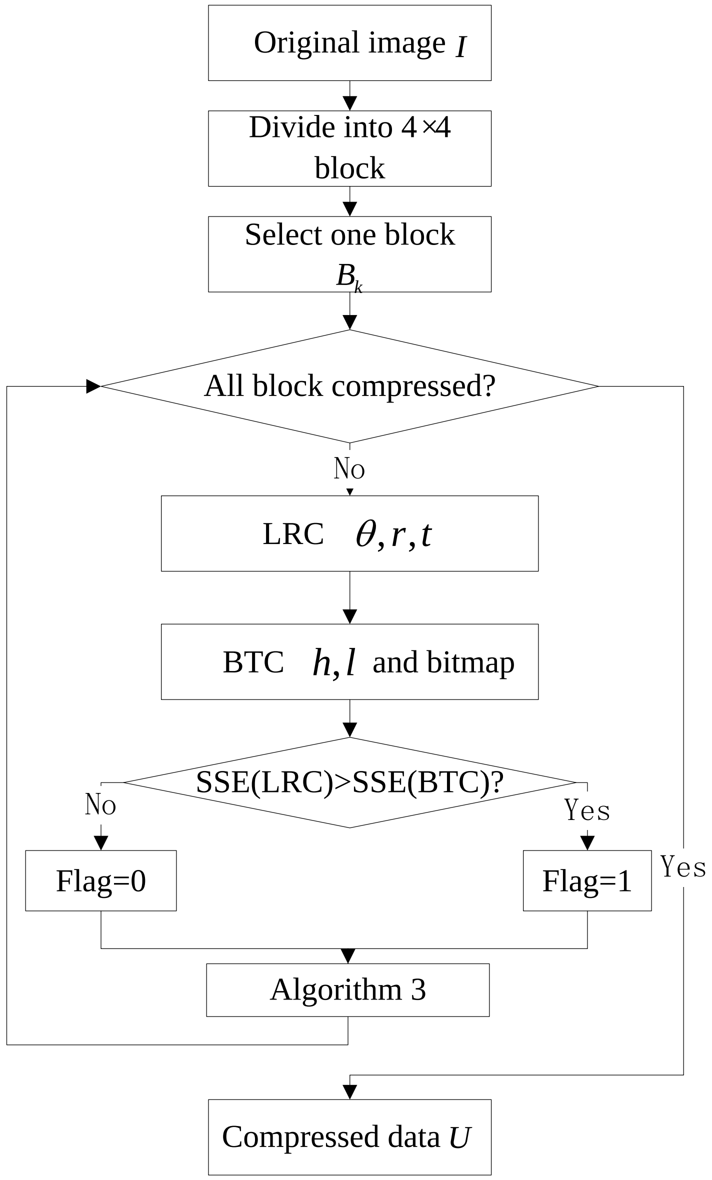

FIGURE 12. Flowchart of the proposed image compression method based on the BTC and LRC hybrid strategy

Fig. 13 shows a flow chart of the image decompression procedure based on the BTC and LRC hybrid strategy.

4. **Experimental Results.** The purpose of our scheme was to compress an image with a smaller number of bits and maintain acceptable visual quality. To prove the performance of the proposed scheme, the 12 gray level images in Fig. 14, each of which was 512×512, were used as original images for a later comparison of image compression quality. Fig. 15 shows the corresponding decompressed images using the proposed method.

The peak signal-to-noise ratio (PSNR) was used to evaluate the image quality. The PSNR of an $M \times N$ gray level image is defined by Equation (14):

$$PSNT = 10 \log_{10} \frac{255^2}{MSE} \tag{14}$$

where MSE (mean square error) for a reconstructed image is defined by Equation (10) in section 2:

Algorithm 4 The compressed block U_k decoding algorithm based on BTC and LRC hybrid strategy..

Input: Flag, U_k
Output: Decompressed block B_k''

1: Step 1. Mark $U_k = [u_1, u_2, u_3, u_4]_k$
2: Step 2. If Flag = 1, compute $h = u_1, l = u_2$, $bitmap = u_3||u_4$, then goto Step 6.
3: Step 3. If Flag = 0, compute $\theta = u_1$, $t = u_4 \bmod 4$, $r = u_3 + 2^{-8} \times (u_4 - t)$, then goto Step 4.
4: Step 4. $a = \tan(\theta)$, use a, r, and Equation (8) to regress row vector $f''(x)$, $x = \{1, 2, \cdots, 16\}$.
5: Step 5. Use model t to rearrange elements of $f''(x) = \{y_1'', y_2'', \cdots, y_16''\}$ to a 4×4 block B_k'' and output.
6: Rearrange the bitmap to 4×4 blocks in which the elements are binary bits. Use Equation (13) to construct 4×4 block B_k'' and output.

$$B_k'' = bitmap \cdot h + \overline{bitmap} \cdot l. \tag{13}$$

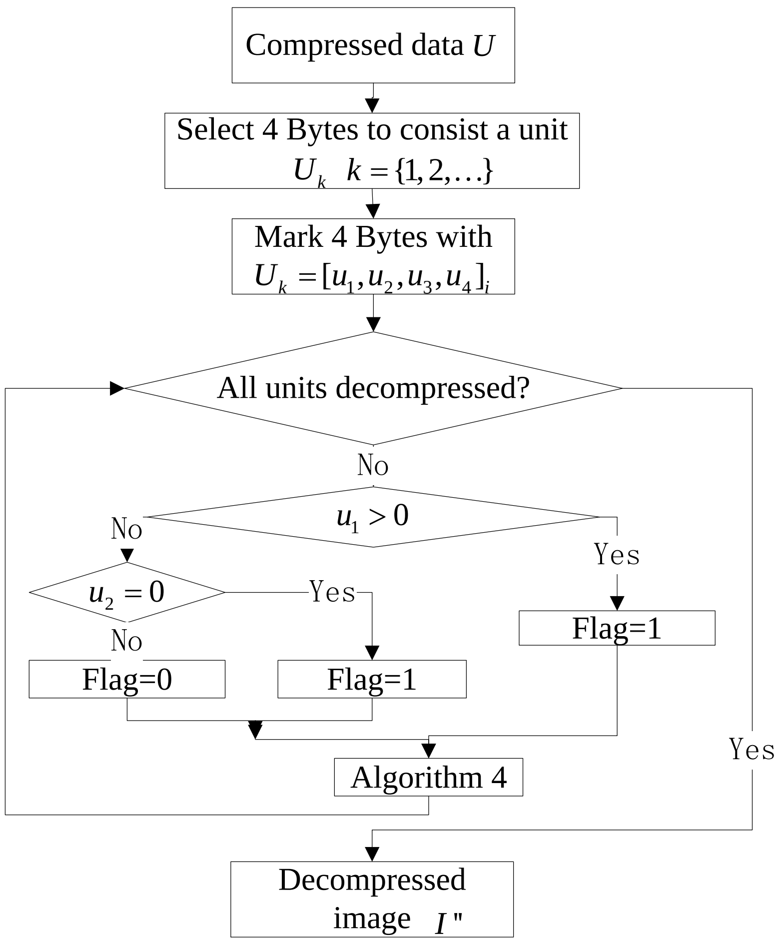

FIGURE 13. Decompression procedure of image compression method based on BTC and LRC hybrid strategy

Fig.16 shows the magnified details of the decompressed Lena image. In Fig. 16 (a), the BTC method was used, and the proposed hybrid method was used in Fig. 16(b).

FIGURE 14. Twelve original gray level test images: (a)Lena; (b)Tiffany; (c)Zelda; (d)Boat; (e)Barbara; (f)Bridge; (g)Baboon; (h)Elaine; (i) Pepper; (j)Goldhill; (k) Couple; (l) Airplane

Fig. 16 shows that, for the image compressed with the proposed hybrid method, the change in the luminance of the decompressed image is finer and smoother, and therefor so this method can effectively be used to compress images.

FIGURE 15. Twelve decompressed images using the proposed method: (a)Lena; (b)Tiffany; (c)Zelda; (d)Boat; (e)Barbara; (f)Bridge; (g)Baboon; (h)Elaine; (i) Pepper; (j)Goldhill; (k) Couple; (l) Airplane

Table 2 shows the comparative performances of BTC, LRC, and the proposed hybrid method based on PSNR. In Table 2, the PSNR using BTC or LTC was less than that of the proposed hybrid method.

(a) (b)

Figure 16. Decompressed 'Lena' image: (a) using BTC method; (b) using the proposed hybrid method

Table 2. Comparative performance results of BTC [1], LRC, PBTC-PF [14] and the proposed hybrid method in terms of PSNR values.

Image \ PSNR	BTC[1]	LRC only	PBTC-PF [14]	The proposed hybrid method
Lena	33.2357	32.2059	31.5082	34.0422
Tiffany	32.5305	28.9988	32.9271	32.7908
Zelda	36.3239	35.4218	33.7575	37.1138
Boat	31.1636	28.7793	30.3084	31.5772
Barbara	29.2453	25.4303	29.3928	29.6348
Bridge	28.5866	25.5987	28.5915	28.6773
Baboon	27.9045	24.4836	27.8127	27.9019
Elaine	33.9077	31.9847	34.2595	34.3761
Pepper	34.1316	31.3243	31.1163	34.6699
Goldhill	32.8608	31.8845	31.7122	33.0851
Airplane	30.0464	26.6581	30.7781	32.3525
Couple	33.0445	30.6399	31.356	36.1309
Average	31.9151	29.4508	31.1267	32.696

The compression bit rate is the number of bits per pixel (bpp) required by the compressed image, which can be calculated using Equation (14):

$$\text{bpp} = \frac{\text{size of compressed image in bits}}{\text{number of pixels}}. \tag{15}$$

The compression bit rate of the proposed method was 4 bpp, the same as the BTC method.

The proposed method invloves less number of simple computations and the time taken for encoding and decoding is a little more compared with BTC. Fig.17 shows the variation in encoding time taken by the proposed method and BTC for the twelve gray level test images with image size and block size and Fig.18 shows the variation in decoding time.

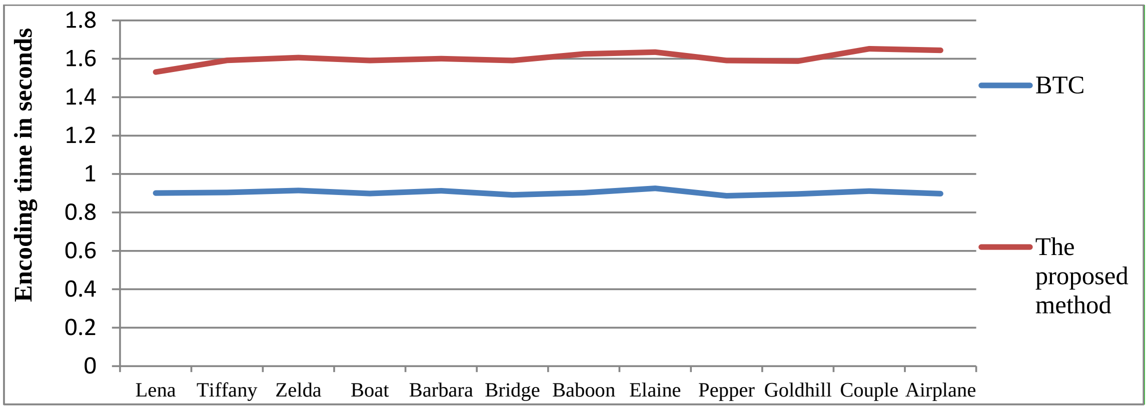

FIGURE 17. Encoding time in seconds compared with BTC

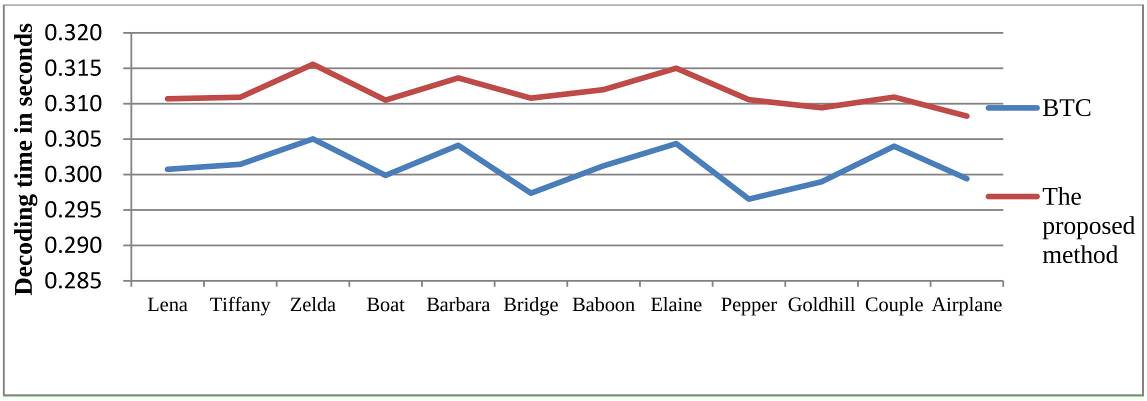

FIGURE 18. Decoding time in seconds compared with BTC

Fig.17 shows the encoding time taken by the proposed method is a little more than BTC method but the decoding time taken by the proposed method is approximately with BTC method as Fig.18 shows. It is deserving to expense a little time to gain a better image quality at lots of image compress cases.

5. **Conclusions.** BTC is an efficient tool for image compression. It is well known for its simplicity and ease of implementation. However, using only two representative values causes confused blocking and false contours, which can not accturately represent the visual features of the image blocks. LRC can gain different values of points with the line function only recording the coefficients of the regression line, while the data distribution shows a linear relationship. The proposed image compression method combines the advantages of the LRC and BTC methods to improve image quality. In this paper, we discussed the models we designed to rearrange the image data blocks to satisfy a linear distribution, and we used these models to code the image blocks. The results of the experiments showed that the proposed method was superior to the BTC scheme in terms of image quality, and it can be used for image compression applications. In future research, the images should be partitioned into 88 or other block sizes to reduce the compression bit.

REFERENCES

[1] E. J. Delp, O. R. Mitchell, Image Compression Using Block Truncation Coding, *IEEE Trans. Commun,* vol. 27, no. 9, pp. 1335-1342, 1979.

[2] M. Lema, O. R. Mitchell, Absolute Moment Block Truncation Coding and Its Application to Color Images *IEEE Trans.Commun*, vol.32, no. 10, pp. 1148-1157, 1984

[3] M. Kamel, C. T. Sun, L. Guan, Image Compression by Variable Block Truncation Coding with Optimal Threshold *IEEE Trans. Signal Processing*, vol. 39, no. 1, pp. 208-212, 1991.

[4] T. M. Amarunnishad, V. K. Govindan, A. T. Mathew, Improving BTC Image Compression Using a Fuzzy Complement Edge Operator *Signal Processing*, Vol. 88, no. 12, pp.2989-2997, 2008.

[5] J. M. Guo, M. F. Wu, Improved Block Truncation Coding Based on the Void-and-cluster Dithering Approach *IEEE Trans. Image Processing,* vol. 18no. 1, pp. 211-213, 2009.

[6] J. M. Guo, C. C. Su, Improved Block Truncation Coding Using Extreme Mean Value Scaling and Block-based High Speed Direct Binary Search *IEEE Signal Processing Letters*, vol. 18, no. 11, pp. 694-697, 2011.

[7] V. Udpikar, J. Raina, BTC Image Coding Using Vector Quantization *IEEE Trans. Commun*, vol. 35, no. 3, pp. 352-356, 1987.

[8] Y. Wu, D. C. Coll, BTC-VQ-DCT Hybrid Coding of Digital Images *IEEE Trans. Commun*, vol. 39, no. 9, pp. 1283-1287, 1991.

[9] C. S. Haung, Y. Lin, Hybrid Block Truncation Coding *IEEE Signal Processing Letters*, vol. 4,no. 12, pp. 328-330, 1997.

[10] C. C. Chang, Y. C.Hu, Hybrid Image Compression Methods Based on Vector Quantization and Block Truncation Coding *Optical Engineering*, vol. 38,no. 4, pp. 591-598, 1999.

[11] E. J. Delp, O. R. Mitchell, The Use of Block Truncation Coding in DPCM Image Coding , *IEEE Trans. Signal Processing*, vol. 39, no. 4, pp. 967-971, 1991.

[12] Y. Wan, Y. Yang, Q. Q. Chen, Multitone Block Truncation Coding *Electronics Letters*, vol. 46, no. 6, pp. 414-416, 2010.

[13] J. M. Guo, High Efficiency Ordered Dither Block Truncation Coding with Dither Array LUT and Its Scalable Coding Application *Digital Signal Processing*, vol. 20,no. 1, pp. 97-110, 2010.

[14] B. C. Dhara, B. Chanda, A Fast Progressive Image Transmission Scheme Using Block Truncation Coding by Pattern Fitting *Journal of Visual Communication and Image Representation*, vol. 23,No2, pp. 313-3222012

[15] G. A. Seber, A. J. Lee, *Linear regression analysis*, vol. 936, John Wiley & Sons 2012

[16] L. J. Revell, Phylogenetic Signal and Linear Regression on Species Data *Methods in Ecology and Evolution*, vol. 1, no. 4, pp. 319-329, 2010.

[17] G. Raskutti, M. J. Wainwright, B. Yu, Minimax Rates of Estimation for High-dimensional Linear Regression Over-balls *IEEE Trans. Information Theory*, vol. 57,no. 10, pp. 6976-6994, 2011.

[18] D. Kleinbaum, L. Kupper, A. Nizam, E. Rosenberg *Applied regression analysis and other multivariable methods* Cengage Learning 2013.

[19] I. Naseem, R. Togneri, M. Bennamoun, Linear Regression for Face Recognition *IEEE Trans. Pattern Anal Mach Intell*, vol. 32 no. 11, pp. 2106-2112, 2010.

[20] R. H. Myers, D. C. Montgomery, G. G. Vining, T. J. Robinson, *Generalized linear models: with applications in engineering and the sciences*, vol. 791, John Wiley & Sons 2012.