



ISSN: 0067-2904

## Small Binary Codebook Design for Image Compression Depending on Rotating Blocks

Maha A. Hameed<sup>1</sup>, Saif B. Al-Khoja<sup>2</sup>, Rafah Rasheed Ismail<sup>1</sup>

<sup>1</sup>Department of Astronomy and space, College of Science, University of Baghdad, Baghdad, Iraq

<sup>2</sup>Department of Biology, College of Science, University of Baghdad, Baghdad, Iraq

Received: 18/1/2020

Accepted: 9/1/2021

### ABSTRACT

The searching process using a binary codebook of combined Block Truncation Coding (BTC) method and Vector Quantization (VQ), i.e. a full codebook search for each input image vector to find the best matched code word in the codebook, requires a long time. Therefore, in this paper, after designing a small binary codebook, we adopted a new method by rotating each binary code word in this codebook into  $90^0$  to  $270^0$  step  $90^0$  directions. Then, we systematized each code word depending on its angle  $\theta$  to involve four types of binary code books (i.e. Pour when  $\theta = 0$ , Flat when  $0 < \theta \leq 30$ , Vertical when  $60 \leq \theta \leq 90$ , or Zigzag). The proposed scheme was used for decreasing the time of the coding procedure, with very small distortion per block, by designing small binary codebook, then rotating each block in it. Moreover, it can be used for improving the efficiency of the coding process even further with the decrease in the bit rate (i.e. increasing the compression ratio).

**Keywords:** Image compression, Block Truncation Coding method, Vector Quantization method VQ, binary code book and block angle.

### كتاب ترميز ثنائي صغير مصمم بالاعتماد على تدوير البلوكات لضغط الصورة

مها احمد حميد<sup>1</sup>، سيف بشير<sup>2</sup>، رفاه رشيد اسماعيل<sup>1</sup>

<sup>1</sup>قسم فضاء وفلك، كلية العلوم، جامعه بغداد، بغداد، العراق

<sup>2</sup>قسم علوم الحياة، كلية العلوم، جامعه بغداد، بغداد، العراق

الخلاصة:

بما ان عملية البحث في كتاب الترميز الثنائي عند جمع طريقي بتر المقاطع وتكميم المتجهات تستغرق وقت طويل ( عملية البحث الكلي لكتاب الترميز لكل بلوك صورته داخل لإيجاد أفضل كلمة ترميز له في كتاب الترميز). لذا في هذا البحث، بعد تصميم كتاب ترميز ثنائي صغير تبيننا طريقه جديدة تتلخص بتدوير كل كلمة ترميز في كتاب الترميز بزوايه  $90^0$  و  $180^0$  و  $270^0$  درجة بعدها تمييز كل كلمة ترميز بالاعتماد على الزاويه  $\theta$  الخاصه بها لإيجاد أربع انواع لكتاب الترميز ( Pour, Flat, Vertical and Zigzag). الطريقة المتبنيه قادره على تقليل زمن عملية الترميز مع تدهور صغير جدا في الانجاز لكل بلوك ( التشوه لكل بلوك) وذلك بتوليد كتاب ترميز ثنائي صغير ثم تدوير كل بلوك فيه. ويمكن استخدامها لتحسين كفاءة عملية الترميز أيضا وتقليل عدد البتات ( اي زيادة نسبة الضغط).

### Introduction

Image compression methods help to reduce the space required to transmit or store image

\*Email: Maha.Hameed@sc.uobaghdad.edu.iq

data by choosing the technique to be applied on the input image. There are numerous methods for compressing data of digital images, each having its own advantages and disadvantages [1]. The BTC method for image data compression was introduced by Delp and Mitchell [2]. It is a lossy and simple image compression method. It has the advantage of being simple to implement, compared with the vector quantization method [3]. In the encoding system of BTC, the image is firstly partitioned into a set of blocks, and then the two statistical moments and the binary image (i.e. bit plane) are computed. After that, in the decoding system, the encoded image blocks are reconstructed using the binary image and the two statistical moments. This method achieves 2 bits per pixel, with a low computational complexity [1]. In this work, vector quantization (VQ) method is used to code binary images. VQ is a very popular method for data compression. The major advantage of VQ is that it achieves high compression ratio with an easy decoding process (look-up table), but it requires a relatively larger time to perform the matching process [3]. Therefore, in this paper, a small binary codebook is designed from a tested image. Then, each code word in this codebook is rotated into  $0^0$  to  $270^0$  step  $90^0$  directions. Finally, all these code words are organized depending on angle  $\theta$  to involve four code books (i.e. Pour, Flat, Vertical, and Zigzag). This approach leads to a decrease in the bit rate, then an increase in the compression ratio, with a reduction in the search time during the coding process.

### Block Truncation Compression method

This technique can easily explained by several steps. In general, a " $pxp$ " pixels image is divided into blocks of size " $n \times n$ " pixels. The mean and the standard deviation values for each block are computed, then two values (i.e. for level quantization for each block) are compared; when a pixel value is greater than the value of mean, it is assigned "1", otherwise "0" [4,5].

### Vector Quantization

This method can be described as a look-up matching process. Let  $X_i$  be an input vector and  $C$  be a given codebook which has  $K$  code words stored in its  $Cw_1, Cw_2, \dots, Cw_K$ . An input vector is represented by the index of the codeword in the codebook that achieves the best matching with the input block [3]. The best matching is identified using the minimum square distortion for the vector  $X_i$  which is found by:

$$d_i = d_{\min(X_i, Cw_j)} \quad \dots\dots(1)$$

A major advantage of VQ method is that the hardware of the decoding process is very simple, but the encoding process is, in fact, a computationally intensive procedure, which limits the applicability of this compression method [6, 7]. An idea which is projected in this paper enables a sensible decrease in the time of coding procedure, with a small deterioration of performance, where  $\theta$  for an input binary image block is calculated to determine the type of this block. Then, a search is conducted in the corresponding code book that has the same type to find the best matched code word; moreover, the proposed method can be used for improving the efficiency of coding even further because the search will be in one type instead of all code book types.

### Block Rotation

To surmount the problem of the longer time needed for performing the matching process (code book search), this paper uses block rotation (i.e. multi code book design) which depends on the edges involved in each block. This design shows the way to reduce the search time required to reach the best matching between input image block and the blocks in the code book.

The proposed scanning method is based on rotating the code book blocks  $0^0$  to  $270^0$  step  $90^0$  angles. For every block, the direction of the main edge is found by computing the value of totality gradient in the vertical and flat directions, as follows;

At first, let  $B(i,j)$  represents a  $n \times n$  element in block (i.e. size  $n \times n$ ).

The Vertical gradient is;

$$V_{(i,j)} = B_{(i,j+1)} - B_{(i,j)} \dots \dots \dots (2), \text{ where } i=1 \dots n, j=1 \dots n-1.$$

But, the Flat gradient is;

$$F_{(i,j)} = B_{(i+1,j)} - B_{(i,j)} \dots \dots \dots (3), \text{ where } i=1 \dots n-1, j=1 \dots n.$$

For each processed block, the effects of total Vertical edge and Flat edge are computed as;

$$EV = \sum_{i=1}^n \sum_{j=1}^{n-1} |V_{(i,j)}| \dots \dots \dots (4)$$

$$EF = \sum_{j=1}^n \sum_{i=1}^{n-1} |F_{(i,j)}| \dots \dots \dots (5)$$

Using equations 4 and 5, the rotation procedure can be determine as follows;

$$\theta = \text{TAN}^{-1} (EV/EF) \dots \dots \dots (6)$$

Each block can be rotated as Pour when  $EV$  and  $EF$  are equal to zero (i.e.  $\theta=0$ ); it is flat when  $0 < \theta \leq 30$ , but it is Vertical when  $60 \leq \theta \leq 90$ , otherwise it is Zigzag.

**The procedure of the new fast method**

The proposed method (i.e. block rotation and multi code book design based on block edge) is described as in the following steps;

**STEP 1.** Initialization, which includes the selection of binary codebook size (no. of code vectors,  $N$ ), block size, and distortion threshold.

**STEP 2.** Rotation of each codeword in this code book by  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$ . In this case, the number of codewords will be  $(32*4)$ .

**STEP 3.** Calculation of  $EV$  and  $EF$  to find  $\theta$  for each codeword, then organizing all these code words depending on  $\theta$  to involve four code books ( i.e. Pour, Flat, Vertical, and Zigzag). Then, a label or an index " $i_c$ " is put for each code book; 0 for Pour, 1 for Flat, 2 for Vertical, and 3 for Zigzag.

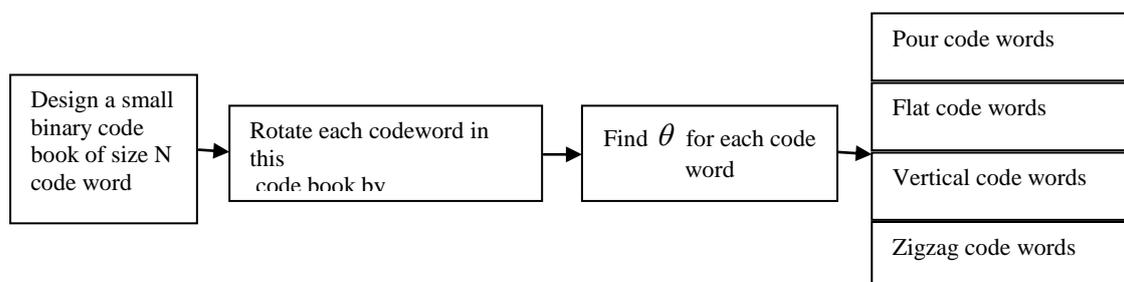
**STEP 4.** The image is divided into blocks  $(4x4)$ . After that, the BTC is applied and, for each block, the two values of reconstruction and binary block ( bit map) are computed.

**STEP 5.** Obtaining an input block from the binary image and calculating its  $EV$ ,  $EF$ , and  $\theta$ .

**STEP 6.** The employment of eq. 6 to classify the input block. It is classified as Pour when both  $EV$  and  $EF$  are equal to zero, Flat when  $0 < \theta \leq 30$ , Vertical when  $60 \leq \theta \leq 90$ , or otherwise it is Zigzag. Then, the input block is given an index " $i_c$ " of a sub code book (0, 1, 2, 3).

**STEP 7.** Determination of the distortion value for the matching of the input vector and the code words of the equivalent class. Then, the input block is given an index " $i_b$ " of the nearest code word, using eq.1.

**STEP 8.** Transforming the two values of reconstruction with two indexes " $i_c$  and  $i_b$ ", then we go to step 5.



**Figure 1-A** diagram illustrating the proposed method.

In other words, one can design a multi code book of larger size than the primary codebook.

### Experimental results

In this paper, we developed a gray image compression scheme with a high-compression ratio, characterized by a decrease in search time. We utilized a new method based on rotating each code word in a small binary codebook of size  $N$  (i.e. code word) to three directions, namely  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$ . In this case, the result was a decrease in code book size (i.e. the number of code words) to  $4N$ . All of these code words were then organized by calculating the angle  $\theta$  for each one (see eq. 6) to involve four code books (i.e. Pour, Flat, Vertical, and Zigzag).

In the experimental setting, we performed several simulations by applying the proposed method on a standard image "RMB1" in different block and codebook sizes. Then, we calculated the compression ratio, bit rate, peak signal to noise ratio, distortion /block, and elimination efficiencies by applying the original scheme (without rotation) and the proposed scheme (with rotating), as shown in table 1.

**Table 1-** Performance of the proposed scheme

RMB1 Image		Original scheme without rotation				The proposed scheme (with rotation)				Efficiencies of Proposed
Codebook size	Block size	Bit rate	Compression ratio	Distortion /block	PSNR	Bit rate	Compression ratio	Distortion /block	PSNR	
32	4x4	1.32	6.06	5.54	35.34	1.31	6.10	5.37	33.11	33%
64	4x4	1.39	5.75	4.15	35.17	1.38	5.83	4.77	33.68	45%
32	8x8	.359	22.30	22.29	31.84	.33	24.38	25.91	30.02	51%
64	8x8	.406	19.71	19.72	32.09	.34	23.26	25.02	30.13	67%

### Conclusions

In the proposed scheme, block rotation was developed to find the best substitution for the codebook. According to the experimental results, the image quality of the image obtained by using the proposed scheme is the same as that achieved using the original scheme (i.e. without rotation), but the proposed scheme enables a sensible decrease in the time of coding procedure, with very small deterioration of performance (i.e. Distortion /block). Moreover, the proposed scheme can be used for improving the efficiency of coding even further, where the number of code words in the codebook was four times higher; therefore, the chance can be faster to find the best-matched code word for each input vector. In other words, our method made the search process faster than the original scheme, without being clearly different in the value of PSNR. Our method may be useful for reducing the bit rate, then increasing the compression ratio and reducing the search time.

The elimination efficiencies of applying our scheme are listed in table 1. It can be seen that percent code words matching are eliminated by using the proposed scheme. The proposed method will be efficient further if we can initially find a code word which has a small distortion to the encoding input vector. This is because many distortion computations can then be eliminated.

**References**

- [1] Rafael C. Gonzalez and Richard E. Woods. 1987. *Digital image processing*, Fourth edition.
- [2] E. Delp & O. Mitchell. 1979. Image compression using block truncation coding, *IEEE Transactions on Communications*, Vol.27, No.9, pp.1335-1342.
- [3] Y. Linde, A. Buzo, and R.M Gray. 1980. An algorithm for vector quantizer design, *IEEE Trans. Commun*, Vol. com-28, pp. 84-95.
- [4] Yu-Chen Hu. 2003. Low complexity and low bit-rate image compression scheme based on Absolute Moment Block Truncation Coding, *Optical Engineering*, Vol. 42, No. 7, pp. 1964-1975.
- [5] C. C. Chang, C. Y. Lin, and Y. H. Fan. 2008. Lossless Data Hiding for Color Images Based on Block Truncation Coding, *Pattern Recognition*, Vol.41, No.7, pp. 2347-2357.
- [6] Maha A. Hameed. 2015. Fast vector quantization algorithm - based on Absolute Moment Block Truncation coding (AMBTC), *International Journal of Application or Innovation in Engineering & Management (IJAIEEM)*, Vo. 4, Issue 6, pp. 4-7.
- [7] Sindhu M1 and Rajkamal R2. 2009. Images and Its Compression Techniques- A Review, *International Journal of Recent Trends in Engineering*, Vol. 2, No. 4, pp. 71-75 .