Small Binary Codebook Design for Image Compression Depending on Rotating Blocks

Maha A. Hameed¹, Saif B. Al-Khoja², Rafah Rasheed Ismail¹

¹Department of Astronomy and space, College of Science, University of Baghdad, Baghdad, Iraq
²Department 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° to 270° step 90° 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.

Introduction

Image compression methods help to reduce the space required to transmit or store image...
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 0⁰ to 270⁰ step 90⁰ directions. Finally, all these code words are organized depending on angle θ 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 "p pxp" pixels image is divided into blocks of size "nxn" 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 Xᵢ be an input vector and C be a given codebook which has K code words stored in its Cw₁, Cw₂,..., CwK. 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 Xᵢ which is found by:

\[ dᵢ = d_{\text{min}}(Xᵢ, Cw_j) \]       \[ \text{.......(1)} \]

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 θ 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 0⁰ to 270⁰ step 90⁰ 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 nxn element in block (i.e. size nxn).
The Vertical gradient is:

\[ V_{(i,j)} = B_{(i,j+1)} - B_{(i,j)} \] \hspace{1cm} \text{where } i=1...n, j=1...n-1. 

But, the Flat gradient is:

\[ F_{(i,j)} = B_{(i+1,j)} - B_{(i,j)} \] \hspace{1cm} \text{where } i=1...n-1, j=1...n.

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

\[ EV = \sum_{i=1}^{n} \sum_{j=1}^{n-1} |V_{(i,j)}| \] \hspace{1cm} \text{equation (4)}

\[ EF = \sum_{j=1}^{n} \sum_{i=1}^{n-1} |F_{(i,j)}| \] \hspace{1cm} \text{equation (5)}

Using equations 4 and 5, the rotation procedure can be determined as follows:

\[ \theta = TAN^{-1} \left( \frac{EV}{EF} \right) \] \hspace{1cm} \text{equation (6)}

Each block can be rotated as Pour when \( EV \) and \( EF \) are equal to zero (i.e. \( \theta = 0 \)); it is flat when \( 0 < \theta < 30 \), but it is Vertical when \( 60 \leq \theta < 90 \), 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 \( 0 < \theta < 30 \), Vertical when \( 60 \leq \theta < 90 \), 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 direction, namely 90°, 180°, and 270°. 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**

<table>
<thead>
<tr>
<th>RMB1 Image</th>
<th>Original scheme without rotation</th>
<th>The proposed scheme (with rotation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Codebook size</td>
<td>Block size</td>
<td>Bit rate</td>
</tr>
<tr>
<td>32</td>
<td>4x4</td>
<td>1.3 2</td>
</tr>
<tr>
<td>64</td>
<td>4x4</td>
<td>1.3 9</td>
</tr>
<tr>
<td>32</td>
<td>8x8</td>
<td>.35 9</td>
</tr>
<tr>
<td>64</td>
<td>8x8</td>
<td>.40 6</td>
</tr>
</tbody>
</table>

**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