Adaptive Bit-Map Correlation in the Ambtc-Vq Techniques Using an Algorithm for Binary Codebook Design

Maha A. Hameed, Ban S. Ismael and Hasanain H. Ali
Department of Astronomy, College of Science, University of Baghdad.

Abstract

In this paper, The coding performance of the VQ method (which is suggested to code the output bit-map produced from the AMBTC technique) will be improved by finding an adaptive method for binary codebook designation which is totally depends on an individual local original image statistics; that is generate a new codebook for each individual coded binary image (bit-map). An application of this coding method is that performed on digital video image sequence. In this paper, we have introduced anew calibration (i.e. adaptive bit-map correlation ***(ρ)***) which must distinguish between the decoding bit-map using VQ technique and the original bit-map (the output from the AMBTC method). From the experimental results affixed binary codebook of size 64 code words, utilizing block of size 4x4, the expected rate will be fall from 2 b/p into 1.375 b/p**.**

Keywords: Image compression, absolute moment block truncation coding, vector quantization, codebook construction, adaptive bit-map correlation *(ρ)*.

Introduction

Image compression is application where images are stored in a data-base such as transmission of TV images, video conferencing, archiving images. In effect, the objective is to reduce redundancy of the image data and to reduce the bit rate in order to be able to store or transmit data in an acceptable fidelity (good image quality), [1].

The block truncation coding (BTC) is the simple technique that is firstly introduced by Delp and Mitchell, [2]. This technique is an efficient image coding method that has been adapted to obtain the statistical properties of the block in an image compression. Another variation of BTC is Absolute Moment Block Truncation Coding or AMBTC, which was proposed by Lema and Mitchell [2]. It is obtained by preserving absolute moments instead of standard moments. This algorithm preserves spatial detail in the image content and has low computation of complexity but it has a bit rate equal to 2 b/p could be achieved with block of size 4x4. Therefore, an adaptation is required to improve the compressibility of the BTC method by utilizing the vector quantization method (VQ) to represent the binary form of the coded image produced from the BTC or AMBTC [2].

A vector quantization method is proposed and carried out as early as 1975 by Hilbert. This quantization method is achieved by quantizing group of numbers together instead of one at a time [3]. In the present paper we study the use of vector quantization in the AMBTC-VQ image compression system. The quality of decoding image depends radically on the codebook used in the VQ technique, here an adaptive method will be improved to generate anew binary codebook from each individual coded bit-map (binary image), this designation is totally depends on an individual local original image statistics, and an adaptive bit-map correlation ***(ρ)*** is generated to find the different between the decoding bit-map using VQ technique and the original bit-map.

Absolute moment block truncation coding (AMBTC):

AMBTC has the same general characteristics as BTC which include low storage requirements and simple coding and decoding stage. To compare the computational complexity of the BTC and AMBTC, we will see that in case of the AMBTC, it is only necessary to calculate the sum of (*m x n*) block pixels values (i.e. this algorithm hasn't involved multiplications) while in the convention BTC case, it is necessary to calculate the sum of (*m x n*) block pixels values and calculate the square of them, thus when using AMBTC the total calculation time at the coding (transmitter) is less, [2].

In the AMBTC method an image is

divided into blocks of size (*m x n*) pixels. For each block the high and low mean (the two reconstruction levels) are calculated, these values change from block to block. If a value of pixel is greater than or equal to the mean of block *(M)*, the corresponding pixel position of the binary image (bit-map) will have "1" value, otherwise it will have "0" value. The two reconstruction levels (i.e. the high and low mean) M_h and M_l, the first for the pixels greater than or equal to *(M)* and the second for the pixels smaller than *(M)* are computed. In decoding, for each block, the pixel positions where the corresponding bit-map has "1" value is replaced by M_h value and those pixel position where the corresponding bit-map has "0" value is replaced by M_l value. When the block with size (*m x n*), the pixels values of its are $(f(i,j), i=1,2,\ldots,m, j=1,2,\ldots,n)$ and the block mean M is [4, 5]:

$$M = \frac{1}{mxn}\sum_{i=1}^{m}\sum_{j=1}^{n} f(i,j) \quad \ldots\ldots (1)$$

$$B(i,j) = \begin{cases} 1 & when\ f(i,j) \geq M \\ 0 & when\ f(i,j) < M \end{cases} \quad \ldots\ldots (2)$$

$(B(i,j), i=1,2,\ldots,m, j=1,2,\ldots,n)$ is a bit-map (binary image), and the two mean values M_h and M_l are calculated as [5]:

$$M_h = \frac{1}{\sum_{i=1}^{m}\sum_{j=1}^{n} B(i,j)} \sum_{i=1}^{m}\sum_{j=1}^{n} B(i,j) f(i,j) \quad \ldots\ldots (3)$$

$$M_l = \frac{1}{mxn - \sum_{i=1}^{m}\sum_{j=1}^{n} B(i,j)} \sum_{i=1}^{m}\sum_{j=1}^{n} (1 - B(i,j)) f(i,j) \quad \ldots\ldots (4)$$

$$g(i,j) = \begin{cases} M_h & if \quad B(i,j) = 1 \\ M_l & if \quad B(i,j) = 0 \end{cases} \forall\ i,j \quad \ldots\ldots (5)$$

Where $(g(i,j), i=1,2,\ldots,m, j=1,2,\ldots,n)$, represents the decoded (output) block.

Vector quantization:

Vector quantization technique is a well-known compression method; it is also called block quantization which is often used in lossy data compression. The idea in this method, it works by encoding values from a multidimensional vector space into a finite set of values from a discrete subspace of lower dimension. The compression or transformation data are usually compressed or transformed using a codebook (i.e. lower-space vectors) which requires less storage space [3, 6].

There are two operations in the vector quantization method, the encoder, and the decoder;

1- The encoder takes an input vector and outputs the index of the codeword that identified the lowest distortion between the input vector and closest codeword in the codebook from this input vector, where the lowest distortion is found by calculating the Euclidean distance between the input vector and each codeword in the codebook. The Euclidean distance is [3, 6]:

$$d(x, y_i) = \sqrt{\sum_{j=1}^{k} (x_j - y_{ij})^2} \quad \ldots\ldots (6)$$

Where x_j is the j th component of the input vector, and y_{ij} is the j th component of the codeword y_i.

The closest vector in the codebook of size N codeword from the input vector is defined the representative codeword (i.e. identified the lowest distortion) eq. (7).

$$d(x, y) = \min_{i=1,\ldots,N} d(x, y_i). \quad \ldots\ldots (7)$$

After the closest codeword is found, the index of that codeword is sent through a channel.

2- When the decoder receives this index, it replaces this index with the associated codeword in the codebook.

The codebook designation is mostly based upon the minimization of the distortion measure eq. (6&7) [6,7]. In this paper a new method is generating to design a binary codebook for each coded bit-map. This algorithm can be described as follows:

An algorithm for a new binary codebook design:

The AMBTC-VQ technique may be improved by finding an adaptive method to design a binary codebook which is totally depends on an individual local original image statistics (i.e. generating a new binary codebook from each individual coded binary image (bit-map)).

An algorithm:

Determine binary codebook size (i.e. no. of code words (no. of classes)). Let no. of iteration ***IT = 0***, the threshold value ***Th = 0*** and **μ** is the increasing value on the threshold.

1. Take the overall lowest distortion ***LD*=0**, ***IT*=*IT*+1, *F*=0, *T*=0**.
2. Choose the first input binary image vector from the bit-map image as the first codeword in the codebook, and let its index ***i*** equal to **1.**
3. Compare all the input binary image vectors that have been not labeled yet with this labeled vector. The best match is based on the nearest neighbor condition (i.e. ***dn*** the lowest distortion between this labeled vector and the input vector eq. 7).
4. If ***dn*** ≤ ***Th*** in this case the input matched vectors should be given the same label (index) of the codeword vector and ***LD=LD+ dn***.
5. Take the next input vector that is not labeled as the next codeword vector and give it a new label (i.e. ***i=i+1***).
6. If ***i*** no. of code words (no. of classes) is greater than the chosen codebook size, then the threshold value ***Th*** should be increased by a small value **μ**, then repeat the process from step 1 again. Other wise go to step 3.
7. From all input vectors which are referred as class ***i*** (have the same index ***i)***, the average vector of each class should be calculated and used to represent all the vectors have the same index in that class.
8. The average of the lowest distortion per block is calculated.
9. To evaluate the performance of this method, a criterion called an adaptive bit-map correlation ***(ρ)*** is used, where, the number of true elements ***(T)*** and the number of false elements ***(F)*** between the decoding bit-map (using above mentioned algorithm in VQ technique) and the original bit-map are computed then an adaptive bit-map correlation ***(ρ)*** is calculated and as follows:

Adaptive bit-map Correlation *(ρ)*:

To evaluate the performance of above algorithm, one way to correct the false element in the bit-map would be a bit-map correlation ***(ρ)*** which is difference between the decoding bit-map (using above mentioned algorithm in VQ technique) and the original bit-map (the output from the conventional AMBTC method), a bit-map correlation ***(ρ)*** can be calculated as:

r = *(Number of true elements - Number of false elements) / Total number of elements.*

$$r = (T - F)/(MXM) \quad ....(8)$$

Where (MXM) is an image size.

T/F represents the ratio between the true and false elements between the decoding bit- map and the original bit-map (binary image).

Table (1)
The ratio between the true and false elements as a function of adaptive bit- map correlation(ρ), when image size 4×4 pixels.

T	*F*	*T / F*	*(ρ)*
16	0	∞	1
12	4	3	0.5
10	6	1.66	0.25
8	8	1	0
6	10	0.6	–.25
4	12	0.33	–0.5
0	16	0	–1

Adaptive bit-map correlation (ρ) is always ranged between 1 & -1,but it can never be 1 because of number of false elements ≠ 0.

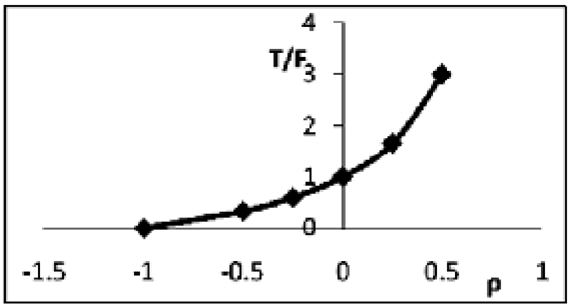

Fig. (1) The ratio between true and false elements as a function of Adaptive bit-map correlation (ρ), when an image of size 4×4 pixels.

Experimental Work

The absolute moment block truncation coding using vector quantization is more efficient than using fixed block truncation coding and that is increased the compression ratio (decreased in bit rate) for an image. For affixed binary codebook of size 64 code words, utilizing block of size 4x4, the expected rate will be fall from 2 b/p into 1.375 b/p.

In the present paper, the original input image (A 64×64 “boy” image) is firstly divided into 4x4 blocks (sub images) and the AMBTC is implemented on blocks, then the VQ technique is implemented on the output bit-map produced from the AMBTC technique to remove the intra-block redundancy.

Here an adaptive method will be improved to generate a new binary codebook which is designed by using a above mentioned algorithm, this designation is totally depends on an individual local image statistics, where the quality of decoding image depends radically on the binary codebook used in VQ.

An adaptive bit-map correlation ***(ρ)*** is generated to use as anew calibration which distinguishes between the decoding bit-map (using above mentioned algorithm in VQ technique) and the original bit-map (the output from the conventional AMBTC method).

Conclusions and Results

Our conclusions will consider the coding results of the mentioned adaptive technique on "boy" image. Table (2), illustrate the decoding image quality is highly affected by the utilized size of the codebook; i.e. As the number of code words within a binary codebook is increased, the quality of decoding image will be increased too, but the distortion per block will be decreased, see Fig. (2&3). However, improving image quality and increasing the compression ratio (decreasing in bit rate) is a trade-off mater, see Table (2) and Fig. (4).

Fig. (5), illustrate when the size of codebook within a binary codebook is increased, adaptive bit-map correlation ***(ρ)*** will be increased too, see Table (2).

Table (2)
Results of the mentioned algorithm on "boy" image, Lowest distortion per block, Th, IT, T/F, ρ and PSNR using different codebooks with block size (4x4) and μ=1.

Binary codebook size	*B.r. b/p*	*Lowest distortion / block*	*Th (Threshold value)*	*IT(no. of iteration)*	*PSNR (dB)*	*(T/F)*	*(ρ)*
128	1.437	.29	1	2	27.19	15.61	.96
64	1.375	.51	3	4	25.24	8.87	.79
32	1.313	.92	4	5	24.66	5.73	.70
16	1.250	1.41	5	6	23.76	4.14	.61

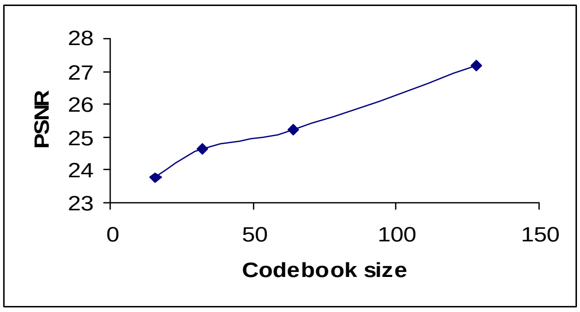

Fig. (2) PSNR as a function of codebook size, implementing above mentioned algorithm to generate different binary codebooks.

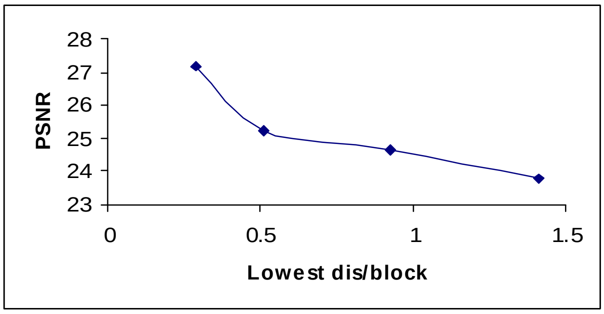

Fig. (3) PSNR as a function of distortion/ block, implementing above mentioned algorithm to generate different binary codebooks.

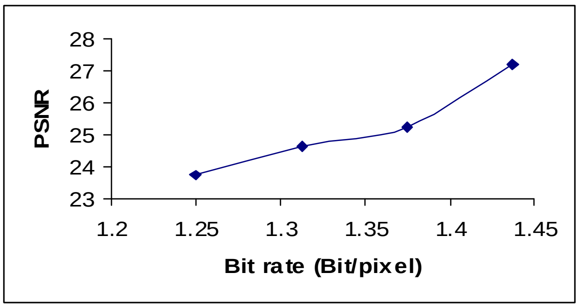

Fig. (4) PSNR as a function of bit rate, implementing above mentioned algorithm to generate different binary codebooks.

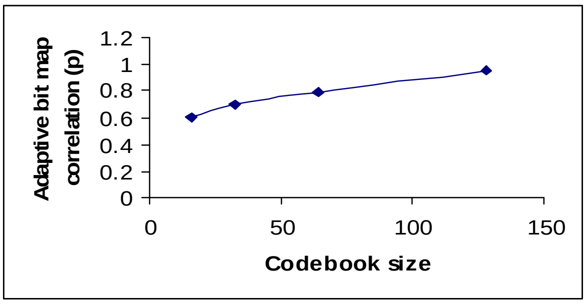

Fig. (5) Adaptive bit-map correlation (ρ)as a function of codebook size, implementing above mentioned algorithm to generate different binary codebooks.

a. Original image. **b. Binary image.** **c. Decoding Binary image.** **d. Decoding image.**

e. Original image histogram. **f. Decoding image histogram.**

Fig. 6: Coding– decoding A 64x64 "boy" image, using 4x4 block size, (128) binary codebook size, μ=1.

a. Original image. **b. Binary image.** **c. Decoding Binary image.** **d. Decoding image.**

e. Original image histogram. **f. Decoding image histogram.**

Fig. (7) Coding–decoding A 64x64 "boy" image, using 4x4 block size, (64) binary codebook size, μ=1.

References

[1] Amarunnishad T. M., Govindan V. K. and Abraham T. Mathew, " Use of Fuzzy edge image in block truncation coding for image compression", International Journal of Signal Processing, Vol. 4, No. 3, 2008, pp. 215-221.

[2] Z. M. Lu, C. H. Liu, and H. Burkhardt, "Block truncation coding based histograms for colour image retrieval", IJCSES International Journal of Computer Sciences and Engineering Systems, Vol. 1, No. 1, 2007.

[3] Y. Linde, A. Buzo, and R. M. Gray, " An algorithm for vector quantizer design", IEEE Transactions on Communications, Vol. com- 28, No. 1, 1980, pp. 84-95.

[4] Y. Wu and D. C. Coll, " BTC-VQ-DCT Hybrid coding of digital images", IEEE Transactions on Communications, Vol. 39, No. 9, 1991, pp. 1283-1287.

[5] Y. - Gi Wu, " Block truncation image bit plane coding", Optical Engineering, Vol. 41, No. 10, 2002, pp. 2476-2478.

[6] N. M. Nasrabadi and Y. Feng, "Vector quantization of images based upon a neural-network clustering algorithm", Visual Communications and Image Processing, SPIE Vol. 1001, 1988, pp. 207-213.

[7] M. A. Hameed, "Fast encoding algorithm for vector quantization - based on block truncation coding technique", Iraqi Journal of Science, Vol. 46, No. 1, 2005, pp. 296-298.

الخلاصة

في هذا البحث تم تطوير انجاز طريقة تكميم المتجهات (والذي استخدم لتشفير الصوره الثناثيه الناتجه من تنفيذ طريقة بتر المقاطع) بتوليد كتاب تشفير ثنائي والذي يعتمد كليا على طبيعة الصورة الأصلية المستخدمة أي توليد كتاب تشفير ثنائي جديد لكل صوره ثنائيه مشفره. التطبيق العملي لمثل هذا العمل هو انه ينجز على الصور الرقمية الفديويه المتتابعة. وهنا تم إدخال معيار جديد والذي يبين الفرق بين الصوره الثنائيه المسترجعه بتنفيذ طريقة تكميم المتجهات عليها والصوره الثناثيه الاصليه الناتجه من تنفيذ طريقة بتر المقاطع. من النتائج العملية ولكتاب التشفير الثنائي حجم 64 وبحجم البلوك4×4 لاحظنا ان نسبة البتات قلت من 2 بت لكل عنصر الصورة الى 1.375 بت لكل عنصر صورة.