A NOVEL BLOCK TRUNCATION CODING OF COLOR IMAGES BY USING QUATERNION-MOMENT-PRESERVING PRINCIPLE

Soo-Chang Pei[1] *Ching-Min Cheng*[2]

[1]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R. O. C.
Email address: pei@cc.ee.ntu.edu.tw
[2]Telecommunication Lab, Ministry of Communications, Taiwan, R.O.C.

ABSTRACT

To compress color-pixel blocks, a novel color BTC algorithm, called the quaternion-moment block truncation coding (QMBTC), is presented in this paper. The QMBTC are derived by using the quaternion arithmetic and moment-preserving principle. The proposed color BTC algorithm can adaptively truncate a pixel block into one or two output classes according to the distribution of color values inside the blocks. The experimental results show that the compression ratio will increase as compared with existing color BTC algorithms and the picture quality of reconstructed images is satisfactory.

1. INTRODUCTION

Block truncation coding (BTC) was first proposed by Delp and Mitchell [1] to compress monochrome images. Unlike other image compression methods such as transform coding and vector quantization, the BTC requires less computation efforts. It also has good capability of combating against channel-errors. Lema and Mitchell [2] have extended the BTC method to color images by applying the BTC technique to each of color planes. Since the BTC is a two-level quantizer that adapts to local properties of the image, the three resultant bit-maps produced by method [2] will be quite similar or almost identical. This motivates the usage of one bit-map to quantizer all three of the color planes in order to save the output bit rate.

Several single bit-map color BTC algorithms [3,4,5] have thus been proposed. However, they process on only one transformed component of input color data and truncate each pixel block into two output classes for transmitting a bit-map. Different from these color BTC algorithms, we propose in this paper a novel color BTC algorithm, called the quaternion-moment block truncation coding (QMBTC). The QMBTC generalizes conventional monochrome BTC [1] to color BTC by expressing input color space as a quaternion-valued space. Through the definition of quaternion moments of input color data, QMBTC extends the moment-preserving principle of [1] from one-dimensional (1D) monochrome data to three-dimensional (3D) color data. One feature of the QMBTC is that it can determine the number of output classes according to the distribution of color values inside the pixel block. The pixel block with similar color values produces only one output class and the associated bit-map can be replaced by one-bit reference indicating one color clustering happened. Thus the QMBTC can achieve better compression ratio than the algorithms of [3,4,5].

2. QUATERNION MOMENTS

The algebra of the quaternions is the generalization of complex numbers [6]. Considering a 4D real-valued data set $H = \{(q_0(n), q_1(n), q_2(n), q_3(n))\}_{n=1}^{N}$, a quadruple data point $(q_0(n), q_1(n), q_2(n), q_3(n))$ can be expressed as a quaternion number $\hat{q}(n)$

$$\hat{q}(n) = q_0(n) + q_1(n) \cdot i + q_2(n) \cdot j + q_3(n) \cdot k \qquad (1)$$

with i, j, and k denote the operation units of quaternion number. Any vector $\mathbf{v} \in \mathbf{R^3}$, can be expressed as a quaternion with q_0 set to be zero. For example, a color value (R, G, B) can be shown as a quaternion with $q_1 = \mathrm{R}$, $q_2 = \mathrm{G}$, $q_3 = \mathrm{B}$, $q_0 = 0$. And any vector $\mathbf{v} \in \mathbf{R^2}$ can be expressed like a complex number. A quaternion can also be denoted as $\hat{q}(n) =< \mathbf{a}, \mathbf{b} >$ where $\mathbf{a} = (\mathbf{q_1(n)}, \mathbf{q_2(n)}, \mathbf{q_3(n)})$ and $b = q_0(n)$. The operation of quaternion number has the following properties:

a) The addition and subtraction rules of the quaternions are the same as for complex numbers.

b) Using the cross product of vector space ($\times$) one can define multiplication of two quaternions, $\hat{q}$ and $\hat{q}'$, as

$$\hat{q} \cdot \hat{q}' =< \mathbf{a}, \mathbf{b} > \cdot < \mathbf{a}', \mathbf{b}' >=< \mathbf{a} \times \mathbf{a}' + \mathbf{b} \cdot \mathbf{a}' + \mathbf{b}' \cdot \mathbf{a}, \mathbf{b} \cdot \mathbf{b}' - \mathbf{a} \cdot \mathbf{a}' > \qquad (2)$$

c) The conjugate $\hat{q}^*$ of $\hat{q}$ is defined as

$$\hat{q}^* = - < \mathbf{a}, \mathbf{b} >= \mathbf{q_0} - (\mathbf{q_1} \cdot \mathbf{i} + \mathbf{q_2} \cdot \mathbf{j} + \mathbf{q_3} \cdot \mathbf{k}) \qquad (3)$$

and the norm of the quaternion is denoted as $\|\hat{q}\|^2 = \hat{q} \cdot \hat{q}^*$.

d) The reciprocal of $\hat{q}$ is

$$(\hat{q})^{-1} = \frac{\hat{q}^*}{\|\hat{q}\|^2} \qquad (4)$$

With the help of the $(\hat{q})^{-1}$, the division of the quaternions is denoted as

$$\frac{\hat{q}'}{\hat{q}} = \hat{q}' \cdot (\hat{q})^{-1} \qquad (5)$$

Based on the above definition of the quaternion, we will designate the quaternion moments as follows in order to explicitly express the statistical parameters of 4D data point:

$$\begin{aligned} \hat{m}_1 &= \mathbf{E}[\hat{\mathbf{q}}] \\ \hat{m}_2 &= \mathbf{E}[\hat{\mathbf{q}} \cdot \hat{\mathbf{q}}^*] \\ \hat{m}_3 &= \mathbf{E}[\hat{\mathbf{q}} \cdot \hat{\mathbf{q}}^* \cdot \hat{\mathbf{q}}] \end{aligned} \qquad (6)$$

0-7803-3073-0/96/$5.00

with $\mathbf{E}[\bullet]$ representing the expectation.

The definitions of $\hat{m}_1$ and $\hat{m}_2$ are the extension of complex moments. And the definition of third order quaternion moment $\hat{m}_3$ is adopted from the high order statistics.

3. QUATERNION-MOMENT BLOCK TRUNCATION CODING

In this section, we present the QMBTC algorithm and its application to do color image compression.

3.1 The Algorithm

The QMBTC can be described as follows:

Step 1. Divide input color image into small non-overlapping blocks, $M \times M$ size.

Step 2 Express the color value of each pixel, (I_1, I_2, I_3), by the quaternion number denoted by (1) where $q_1 = I_1$, $q_2 = I_2$, $q_3 = I_3$, $q_0 = 0$.

Step 3. Obtain two quantized levels, $\hat{z}_0$ and $\hat{z}_1$, for each pixel block by solving the quaternion-moment-preserving equations

$$\begin{aligned} p_0 \cdot \hat{z}_0 + p_1 \cdot \hat{z}_1 &= \hat{m}_1 \\ p_0 \cdot \hat{z}_0 \cdot \hat{z}_0^* + p_1 \cdot \hat{z}_1 \cdot \hat{z}_1^* &= \hat{m}_2 \\ p_0 \cdot \hat{z}_0 \cdot \hat{z}_0^* \cdot \hat{z}_0 + p_1 \cdot \hat{z}_1 \cdot \hat{z}_1^* \cdot \hat{z}_1 &= \hat{m}_3 \\ p_0 + p_1 &= 1 \end{aligned} \tag{7}$$

where p_0 and p_1 denote the probabilities of each pixel being assigned as $\hat{z}_0$ and $\hat{z}_1$, respectively.

Step 4. Choose the hyperplane l' perpendicular to and bisecting the line segment $\overline{\hat{z}_0 \hat{z}_1}$ as the decision boundary of the pixel block.

Step 5. Construct a class-indicating bit-map such that each pixel location is coded as a "one" or a "zero" depending on whether that pixel is on the right of the decision boundary or not.

In Step 3, moment-preserving principle is employed in order to keep the first three quaternion moments of the pixel block, $\hat{m}_1, \hat{m}_2, \hat{m}_3$, unchange after $\hat{z}_0$ and $\hat{z}_1$ are obtained. Instead of using $\hat{z}_0$ and $\hat{z}_1$, we select in this paper the centroid of each output class as reproduction colors of the truncated block in order to reduce the minimum mean square error.

To illustrate coding a pixel block, we select a 4×4 pixel block from an image and arrange the color values,$(I_1, I_2, I_3) = (R, G, B)$, in the block as a quaternion-valued matrix $\mathbf{X}$. In $\mathbf{X}$, each element (q_0, q_1, q_2, q_3) is set to $(0, R, G, B)$ of the corresponding pixel location.

$$\mathbf{X} = \begin{bmatrix} (0,228,133,105)(0,230,132,111)(0,228,136,120)(0,229,136,110) \\ (0,230,136,111)(0,226,130,97)(0,231,137,113)(0,232,135,115) \\ (0,234,138,115)(0,229,137,99)(0,229,131,98)(0,230,133,106) \\ (0,230,143,117)(0,231,141,105)(0,228,136,112)(0,228,131,106) \end{bmatrix}$$

so

$$\begin{aligned} \hat{z}_0 &= (0., 229.6, 135.5, 109.2) \\ \hat{z}_1 &= (0., 206.1, 87.6, 9.4) \end{aligned}$$

and the bit-map is

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

This example shows that only one output class is generated. Another example will demonstrate a case of two output classes:

$$\mathbf{X} = \begin{bmatrix} (0,147,79,90)(0,134,74,93)(0,144,79,101)(0,150,86,99) \\ (0,157,83,98)(0,148,63,90)(0,16,16,39)(0,147,91,106) \\ (0,142,89,92)(0,13,14,34)(0,15,17,38)(0,155,97,117) \\ (0,17,14,19)(0,145,93,93)(0,143,92,117)(0,142,90,122) \end{bmatrix}$$

so

$$\begin{aligned} \hat{z}_0 &= (0., 146.0, 85.2, 102.2) \\ \hat{z}_1 &= (0., 14.1, 13.3, 29.3) \end{aligned}$$

and the bit-map is

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

3.2 Application to Color Image Compression

As it is known, the RGB space has extensive correlation among color components. Besides, pixels within the block are likely to have spatial correlation, which results in similar color values, except for the edge blocks. These two factors cause too many one-class pixel blocks by applying the QMBTC on the RGB space. Even though the compression ratio is high in this space, the picture quality of the reconstructed image is not acceptable from our empirical results. To alleviate this situation, we first transform the RGB space to Yrg space.

$$\begin{aligned} Y &= \frac{(R+G+B)}{3} \\ r &= \frac{R}{(R+G+B)} \\ g &= \frac{G}{(R+G+B)} \end{aligned} \tag{8}$$

In this space, the color components are decorrelated into two parts, the achromatic and chromatic components. The achromatic Y component represents intensity of image and approximates the principal axis of the KL transform on the RGB space. The r and g chromatic components represent the normalized chromatic information respectively.

Then, the weights w_v for the associated color components, $(I_1, I_2, I_3) = (Y, r, g)$, are determined by the smoothness of Y values inside the pixel block. If sample mean value of Y inside the pixel block is denoted as $\bar{Y}$, w_v are defined as

$$\begin{aligned} w_1 &= 1 - \exp(-\frac{T}{\rho}) \\ w_2 &= \frac{\exp(-\frac{T}{\rho})}{2} \\ w_3 &= \frac{\exp(-\frac{T}{\rho})}{2} \\ T &= \frac{1}{N}\sum_{n=1}^{N} |(Y(n) - \bar{Y}| \end{aligned} \tag{9}$$

where ρ is a scaling factor and $Y(n)$ is the Y value of nth pixel in the block. The weighted color components are thus $w_v \cdot I_v$, which are applied to the QMBTC. From (9), we see that T indicate the variation of Y values inside the pixel block. When the case of $T = 0$ is happened, there is a constant Y value inside the pixel block. The CCC algorithm [3], which processes Y component only, would not produce satisfactory results in this situation since there might be different colors existed within the pixel block. Nevertheless, the weighted color component approach of the QMBTC can solve this problem. From (9), we understand that the weighted r and g components would dominate QMBTC and help it judge whether pixels inside the block should be truncated into two different classes or not.

4. EXPERIMENTAL RESULTS AND CONCLUSIONS

In Table I, we illustrate the distribution of thresholded pixel blocks by using the QMBTC on four test images, when the input color spaces are the RGB space and weighted Yrg space with $\rho = 3.0$, respectively. The choice of ρ is based on empirical results that $\rho = 3.$ is a good compromise between bit rate and APSNR performance. $\rho = 3.$ is also selected in the following experiments of the proposed QMBTC. It can be seen from Table I that the arrangement of w_v by (9) assists in improving the situation of too many one-class pixel blocks for each test image.

To evaluate the performance of applying the QMBTC to color image compression, we conducted the experiments on four test images. Table II illustrates the performance comparison among the QMBTC, CCC algorithm [3] and Kurita and Otsu's algorithm [5]. We observe that the bit rate will save 30% on average by the proposed QMBTC algorithm as compared with the other testing algorithms. However, the Average Peak Signal-to-Noise-Ratio (APSNR) of the proposed algorithm is close to those of the other testing algorithms. In addition, Fig. 1 shows the reconstructed images created by using these color BTC algorithms on 'Lena'. It is noticed that there is no significant image quality degradation between the reconstructed image produced by the proposed algorithm and those of the other testing algorithms.

Therefore, the proposed QMBTC is an efficient color BTC which can produce good compression ratio and output picture quality.

Images	Color Spaces	One-Class Blocks	Two-Class Blocks
Lena	I	11858	4526
	II	9408	6976
Peppers	I	12199	4185
	II	9408	6976
Scene	I	10396	5988
	II	7576	8808
Jet	I	11430	4954
	II	9670	6714

Table 1. Thresholded blocks distribution of the QMBTC under different color spaces: I - RGB, II - Yrg.

Images	Algorithms	APSNR(dB)	Bit-Rate(bits/pixel)
Lena	I	32.2	4.0
	II	32.3	4.0
	III	32.0	2.56
Peppers	I	31.1	4.0
	II	31.6	4.0
	III	30.8	2.56
Scene	I	28.1	4.0
	II	28.2	4.0
	III	28.2	2.84
Jet	I	31.8	4.0
	II	32.0	4.0
	III	32.2	2.52

Table 2. Performance comparison of various color BTC algorithms: I - CCC algorithm [3], II - Kurita and Otsu's algorithm [5], III- the QMBTC algorithm.

Fig. 1. Original and reconstructed images of 'Lena'. (a) original image.

REFERENCES

[1] E. J. Delp and O. R. Mitchell, "Image Compression Using Block Truncation Coding," *IEEE Trans. Commu.*, vol. COMM-27, pp. 1335-1341, Sep. 1979.

[2] M. D. Lema and O. R. Mitchell, "Absolute Moment Block Truncation Coding and Application to Color Image," *IEEE Trans. Commu.*, vol. COMM-32, pp. 1148-1157, Oct. 1984.

[3] G. Campbell, T. A. Defanti, J. Frederiksen, S. A. Joyce, A. L. Lawrence, J. A. Lindberg, and D. J. Sandin, "Two Bit/Pixel Full Color Encoding ," *Comput. Graph.*, vol. 20, no. 4, pp. 215-223, Aug. 1986.

[4] Y. Wu and D. C. Coll, "Single Bit-Map Block Truncation Coding of Color Images," *IEEE Journal on Selected Areas in Commu.*, vol. 10, no. 5, pp. 952-959, Jun. 1992.

[5] T. Kurita and N. Otsu, "A Method of Block Truncation Coding for Color Image Compression," *IEEE Trans. Commu.*, vol. COMM-41, pp. 1270-1274, Sep. 1993.

[6] J. B. Fraleigh, *A First Course in Abstract Algebra*, Addison-Wesley, 1982.

(b) (c)

(d) (e)

Fig. 1. (Continued)

(b) reconstructed image by the QMBTC with block size 4×4.

(c) thresholded block distribution by the QMBTC.

(d) reconstructed image by the CCC Algorithm with block size 4×4.

(e) reconstructed image by the Kurita and Otsu's Algorithm with block size 4×4.