COLOR IMAGE COMPRESSION BY MOMENT-PRESERVING AND BLOCK TRUNCATION CODING TECHNIQUES†

Chen-Kuei Yang, Ja-Chen Lin[Δ], and Wen-Hsiang Tsai

Department of Computer and Information Science, National Chiao Tung University
Hsinchu, Taiwan 300, Republic of China.

ABSTRACT

A new color image compression technique based on moment-preserving and block truncation coding (BTC) is proposed. An input image is divided into non-overlapping blocks. And each block is coded by two representative colors, or equivalently, by two sets of tristimulus values (R_1, G_1, B_1) and (R_2, G_2, B_2). To compute the two representative colors of each block, some color moments of the block are preserved, and the two sets of desired tristimulus values are computed accordingly using analytic formulas. The compression ratio is 6 when the block size is 4×4. Furthermore, a simple method is applied to quantize the bit map into 64 predefined bit maps and requantize each color component from 8 bits into 4 bits. An average compression ratio of 14.00 can be achieved.

I. INTRODUCTION

The block truncation coding (BTC) algorithm developed by Delp and Mitchell [1] is to use a two-level non-parameteric quantizer that adapts over the local regions of the input image. Many trichromatic models for color have been developed [2]. The components of the RGB model are highly correlated with one another, but most monitors use them as input signals. Lema and Mitchell [3] use the absolute the moment BTC technique on the three color planes of a color image separately. The resulting bit maps occupy more than 61% of output code, and a compression ratio of 4 can be obtained for a 4×4 image block. Wu and Coll [4] use a single bit map to quantize all the three color planes. This means that only one out of three bit planes need be preserved. Kurita and Otsu [5] use the mean vector and covariance matrix of color vectors to compute the principal score for each pixel in the block and classify the pixels into two classes. The two classes' mean vectors and a bit map are preserved. In these methods, a compression ratio of 6 can be obtained for a 4×4 block.

The remainder of this paper is organized as follows. In Section II, proposed color image compression using the moment-preserving and BTC techniques is described. To improve the compression ratio, the three modifications of the proposed compression algorithm are described in Section III. In Section IV, we present several experimental results to show the feasibility of the proposed method. Finally, some conclusions are made in Section V.

II. PROPOSED COLOR IMAGE COMPRESSION USING MOMENT-PRESERVING AND BTC TECHNIQUES

Given an image f with (r, g, b) tristimulus values, let F_j be a 4×4 block of f and P_i be a pixel in F_j with tristimulus values (r_i, g_i, b_i). Suppose that H is the compressed version of F_j and the pixels of H have only two sets of representative tristimulus values (R_1, G_1, B_1) and (R_2, G_2, B_2). Let p_1 be the fractions of pixels in H with tristimulus values (R_1, G_1, B_1), and p_2 the fractions of pixels in H with tristimulus values (R_2, G_2, B_2).

The proposed algorithm for color image compression using the moment-preserving and BTC techniques (abbreviated as CICMPBTC) is described as follows.

<u>Algorithm</u> <u>CICMPBTC</u>

Step 1: Read in an $M\times N$ color image G.

Step 2: Partition G into non-overlapping 4×4 blocks F_j.

Step 3: For each block F_j perform the following steps:

(a) preserve a set of color moments of F_j to obtain a set of equations involving as variables the values of (R_1, G_1, B_1) and (R_2, G_2, B_2);

† This work was supported partially by Na-tional Science Council, Republic of China under Grant NSC83-0408-E009-010.

Δ Also with the Department of Information Management, Ming Chuan College, Taipei, R. O. C.

(b) solve the equations to obtain the desired values of (R_1, G_1, B_1) and (R_2, G_2, B_2) if the spectral condition is 3-spectral; otherwise compute the desired tristimulus values using different methods for F_j according to the spectral conditions (the details are described later) ;

(c) generate a 4×4 bit map for F_j by assigning "0" or "1" to each element P_i of F_j according to whether (r_i, g_i, b_i) is closer to (R_1, G_1, B_1) or (R_2, G_2, B_2);

(d) allow 8 bits for each of the six tristimulus values (R_1, G_1, B_1) and (R_2, G_2, B_2);

(e) take the bit map and the six representative tristimulus values as output.

Several color moments of each block are preserved according to the moment-preserving principle which has also been applied to several other image processing tasks [6-9]. A set of analytic formulas can be obtained accordingly to compute two sets of (R, G, B) tristimulus values for each block. They are described as follows:

$$m_r = \sum_i p_i r_i,\ m_g = \sum_i p_i g_i,\ m_b = \sum_i p_i b_i,$$
$$m_{r^2} = \sum_i p_i r_i^2,\ m_{g^2} = \sum_i p_i g_i^2,\ m_{b^2} = \sum_i p_i b_i^2,\quad (1)$$
$$m_{rgb} = \sum_i p_i r_i g_i b_i,$$

where $p_i = \frac{1}{n}$ for all i; n is the total number of pixels in F_j; m_r, m_g, and m_b are the means of the (r, g, b) tristimulus values, respectively; m_{r^2}, m_{g^2}, and m_{b^2} are related to the variances of the (r, g, b) tristimulus values, respectively; and m_{rgb} is related to the correlation within the (r, g, b) tristimulus values.

By preserving the above seven moments of F_j in the compressed version H, the resulting equations are as follows:

$$p_1R_1 + p_2R_2 = m_r,\quad p_1R_1^2 + p_2R_2^2 = m_{r^2},$$
$$p_1G_1 + p_2G_2 = m_g,\quad p_1G_1^2 + p_2G_2^2 = m_{g^2},$$
$$p_1B_1 + p_2B_2 = m_b,\quad p_1B_1^2 + p_2B_2^2 = m_{b^2},\quad (2)$$
$$p_1R_1G_1B_1 + p_2R_2G_2B_2 = m_{rgb},$$
$$p_1 + p_2 = m_0,$$

where all the terms on the left-hand sides of the equalities are the moments of H and all those on the right-hand sides are the corresponding ones of F_j computed according to (1). There are eight unknowns p_1, p_2, R_1, G_1, B_1, R_2, G_2, and B_2. Note that we define $m_0 = 1$ so that $p_1 + p_2 = 1 = m_0$.

Without loss of generality, we shift the values of the tristimulus values such that the three mean values m_r, m_g, and m_b are zero. After (R_1, G_1, B_1) and (R_2, G_2, B_2) are obtained from solving (2), these six values of color components should be translated back by adding to them the mean values m_r, m_g, and m_b, respectively.

Let om_{r^2}, om_{g^2}, om_{b^2}, and om_{rgb} denote the shifted moments of m_{r^2}, m_{g^2}, m_{b^2}, and m_{rgb}, respectively. Assume that $p_1 \geq p_2$. Then (R_1, G_1, B_1), (R_2, G_2, B_2), p_1, and p_2 can be solved (with details omitted) to be:

$$p_1 = \frac{1}{2} + \frac{1}{2}\sqrt{\frac{om_{rgb}^2}{om_{rgb}^2 + 4om_{r^2}^2 om_{g^2}^2 om_{b^2}^2}},$$
$$p_2 = 1 - p_1,$$

$$R_2 = \pm\sqrt{om_{r^2} k},\quad G_2 = \pm\sqrt{om_{g^2} k},\quad B_2 = \pm\sqrt{om_{b^2} k},$$
$$R_1 = -\frac{om_{r^2}}{R_2},\quad G_1 = -\frac{om_{g^2}}{G_2},\quad B_1 = -\frac{om_{b^2}}{B_2},\quad (3)$$

where

$$k = \frac{om_{rgb}^2 + 2om_{r^2}^2 om_{g^2}^2 om_{b^2}^2 + \sqrt{om_{rgb}^4 + 4om_{r^2}^2 om_{g^2}^2 om_{b^2}^2 om_{rgb}^2}}{2om_{r^2}^2 om_{g^2}^2 om_{b^2}^2}.$$

Note here that it is the appropriate selection of the seven color moments of (1) that leads to the possibility of obtaining the analytic solutions for (R_1, G_1, B_1) and (R_2, G_2, B_2) as described in (3).

Different techniques are applied to different color conditions (1-spectral, 2-spectral, or 3-spectral) in the 4×4 block, such that one of the two sets of (R, G, B) tristimulus values is assigned to each pixel in the block according to the Euclidean distance measure. The method of Lin and Tsai [10] is used to compute suitable values for the two non-uniform color components for the 2-spectral condition. And the method of Tsai [11] is used to select a suitable value for the single non-uniform color component for the 1-spectral condition. We focus on the 3-spectral case in this study. The mean of the original color component values of the block is used as a substitute for each uniform color plane. A compression ratio of 6 can be obtained by allowing 8 bits for each component of the two sets of (R, G, B) values and 16 bits for the 4×4 bit map.

III. THREE MODIFICATIONS OF PROPOSED CICMPBTC ALGORITHM

To improve the compression ratio, three modifications of the CICMPBTC algorithm (denoted as CICMPBTC-1, CICMPBTC-2, and CICMPBTC-3) are proposed. The CICMPBTC-1 algorithm is designed to remove the redundancy found in specifying the two sets of representative tristimulus values (R_1, G_1, B_1) and (R_2, G_2, B_2) when the image block has uniform color planes. For example, when the image block has uniform red component values, the two tristimulus values sets are (R_1, G_1, B_1) and (R_2, G_2, B_2) with $R_1 = R_2$ being computed as the mean of all the red component values of the block. The proposed algorithm CICMPBTC-1 uses a 3-bit SI code to represent the various spectral cases and saves the bits used to represent the duplicated tristimulus values when the image blocks have uniform color planes. This coding method is found effective in improving the compression ratio. In order to improve the compression ratio further , we code the bit map more efficiently using the method of Yang and Tsai [12] which is based on the observation of the human visual system's sensitivity to edge and line patterns in small image blocks. This results in Algorithm CICMPBTC-2, which uses 6 bits to code the bit map of each image block after classifying the bit map into one of sixty-four predefined bit maps which includes almost all possible variations of edge and line patterns of size 4×4.

Algorithm CICMPBTC-2 follows the steps of CICMPBTC-1 until the bit map is generated. Then the bit map is matched with the sixty-four predefined bit maps and the best-match bit map is chosen as the desired bit map. Each block is finally coded by the 3-bit SI code, some of six 8-bit tristimulus values, and a bit-map index except the case with the SI code being 000. From our experimental results, the average compression ratio was found to be about 8.05.

Furthermore, we apply further a simple method to requantize each color component from 8 bits into 4 bits. This is equivalent to equally dividing the value range of each color component from 256 levels into 16 levels, and thus requantizing the color image into 4096 colors $(16 \times 16 \times 16)$. And based on this simple method, we propose a third modification version (denoted as CICMPBTC-3) of the CICMPBTC algorithm, which requantizes the truncated data in the above way from the result produced by the CICMPBTC-2 algorithm. As a result, an average compression ratio of 14.00 can be achieved. For some randomized image contents, the compression ratio can be raised to about 20.

IV. EXPERIMENTAL RESULTS

The proposed algorithms have been tested on an IRIS Indigo workstation on several color images. Each color image has 24 bits/pixel and is 512×512 in size. A smaller value of the MAE (Mean Absolute Error) means that the reconstructed image is less distorted. Table 1 shows the comparative MAE values of the proposed algorithms CICMPBTC, CICMPBTC-1, CICMPBTC-2, and CICMPBTC-3 (with color component requantization bit number ξ=4), respectively. For each method, the images "Lenna," "pepper," "jet," and "house," are tested. Table 2 shows the compression ratios of the images mentioned above. Tables 3 and 4 show the comparative MAE values and compression ratios yielded by Algorithm CICMPBTC-3 for which each color image mentioned above is requantized using ξ=3, 4, 5, 6, or 7 bits for each color component.

Fig. 1 shows the standard images of "Lenna," "pepper," and "jet" of size 512×512 in (a), (b), and (c), respectively. Fig. 2, shows the compression results of "Lenna" using Algorithms CICMPBTC-1, CICMPBTC-2, and CICMPBTC-3 (using 4 bits to requantize each color component), respectively. Fig. 3 shows the compression results of "Lenna," "pepper," and "jet" using Algorithm CICMPBTC-3 with 3 bits to requantize each color component in (a), (b), and (c), respectively.

From Tables 1 and 2 several facts can be observed. First, Algorithm CICMPBTC-1 is an error-free improvement on Algorithm CICMPBTC; reconstructed image qualities from CICMPBTC and CICMPBTC-1 are exactly the same. Second, Algorithm CICMPBTC-2 can both preserve reconstructed image quality and gain reasonable compression ratios due to the use of the sixty-four predefined bit maps. It requires fewer bits to code the bit map without sacrificing too much quality in the reconstructed image. From Figs. 2(a) and 2(b), we can find that the image quality is almost the same as that of the original. This proves the feasibility and efficiency of the proposed algorithm CICMPBTC. It can be seen from Figs. 2(c) and 3 that the resulting image quality is still good and acceptable (only with some slightly zigzag edges).We can find slight color difference only in gradually changing image regions (see image "Lenna," for example). This is due to the use of fewer bits to requantize the color components which makes more colors being requantized to identical ones. Note that when the image content is randomized, it will introduce less visual color difference in the reconstructed image (see images "pepper" and "jet," for examples).

V. CONCLUSIONS

A new approach to compressing color images has been proposed. The approach is based on the moment-preserving principle and the BTC technique. The proposed method preserves the spatial details in the image content and requires little memory. It has low computational complexity because analytic formulas have been provided for some of the compression steps. RGB images can be compressed directly without any transformation. The proposed method is simple but produces compression results both high in compression ratios and good in reconstructed image quality. It is applicable not only to the RGB color model but also to any other trichromatic model.

REFERENCES

[1] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Trans. on Commun.*, vol. COM-27, pp. 1335-1342, 1979.
[2] W. K. Pratt, *Digit Image Processing*, 2nd edition, New York, U. S. A. : Wiley, 1991, Ch. 4, pp. 195-218.
[3] M. D. Lema and O. R. Mitchell, "Absolute moment block truncation coding and its application to color images," *IEEE Trans. on Commun.*, vol. COM-32, pp. 1148-1157, 1984.
[4] Yiyan Wu and David C. Coll, "Single bit map block truncation coding for color image," *IEEE Trans. on Commun.*, vol. 10, no. 5, pp. 952-959, June 1992.
[5] Takio Kurita and Nobuyuki Otsu, "A method of block truncation coding for color image compression," *IEEE Trans. on Commun.*, vol. 35, pp. 352-356, Mar. 1987.
[6] S. T. Liu and W. H. Tsai, "Moment-preserving corner detection," *Pattern Recognition*, vol. 23, No. 5, pp. 441-460,1990.
[7] L. H. Chen and W. H. Tsai, "Moment-preserving line detection," *Pattern Recognition*, vol. 21, pp. 45-53, 1988.
[8] S. C. Cheng and W. H. Tsai, "Image compression using moment-preserving edge detection," *Proc. of IEEE Data Compression Conference*, Snowbird, Utah, U. S. A., March 1992.
[9] C. K. Yang, C. T. Wu, J. C. Lin, and W. H. Tsai, "Color image sharpening by moment-preserving technique," submitted.
[10] J. C. Lin and W. H. Tsai, "Feature-preserving clustering of 2D data for two-class problems using analytical formulas," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, to appear.
[11] W. H. Tsai, "Moment-preserving thresholding," *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 377-393, 1988.
[12] C. K. Yang and W. H. Tsai, "Improving block truncation coding using line and edge information for gray-scale image compression," submitted.

Table 1. The comparative MAE values of Algorithms CICMPBTC, CICMPBTC-1, CICMPBTC-2, and CICMPBTC-3, respectively, with color component requantization bit number $\xi=4$.

image	CICMPBTC	CICMPBTC-1	CICMPBTC-2	CICMPBTC-3
Lenna	4.05	4.05	4.82	8.75
pepper	4.65	4.65	5.51	9.56
jet	3.77	3.77	4.31	8.86
house	2.03	2.03	2.17	6.94

Table 2. The compression ratios of Algorithms CICMPBTC, CICMPBTC-1, CICMPBTC-2, and CICMPBTC-3, respectively, with color component requantization bit number $\xi=4$.

image	CICMPBTC	CICMPBTC-1	CICMPBTC-2	CICMPBTC-3
Lenna	6.0	6.22	7.38	12.63
pepper	6.0	6.03	6.75	12.24
jet	6.0	8.56	9.60	16.48
house	6.0	7.36	8.48	14.65
average	6.0	7.04	8.05	14.00

Table 3. The comparative MAE values of Algorithm CICMPBTC-3 with color component requantization bit number $\xi=3$, 4, 5, 6, and 7, respectively.

image	3 bits	4 bits	5 bits	6 bits	7 bits
Lenna	15.75	8.75	5.91	4.99	4.82
pepper	17.30	9.56	6.64	5.74	5.53
jet	15.93	8.86	5.73	4.56	4.41
house	15.07	6.94	3.65	2.54	2.18

Table 4. The comparative compression ratios of Algorithm CICMPBTC-3 with color component requantization bit number $\xi=3$, 4, 5, 6, and 7, respectively.

image	3 bits	4 bits	5 bits	6 bits	7 bits
Lenna	15.34	12.63	10.73	9.33	8.25
pepper	14.90	12.24	10.39	9.02	7.97
jet	20.24	16.48	13.89	12.01	10.58
house	17.96	14.65	12.40	10.74	9.48
average	17.11	14.00	11.85	10.28	9.07

Fig. 1 The standard images of (a) "Lenna," (b) "pepper," and (c) "jet" of size 512×512.

(a) Compression ratio = 6.22. (b) Compression ratio = 7.38. (c) Compression ratio = 12.63.

Fig. 2 The coded images of "Lenna," using CICMPBTC-1, CICMPBTC-2, and CICMPBTC-3 (with color component requantization bit number ξ=4) in (a), (b), and (c), respectively.

(a) Compression ratio = 15.34 (b) Compression ratio = 14.90. (c)Compression ratio = 20.24.

Fig. 3 The coded images of (a) "Lenna," (b) "pepper," and (c) "jet" using CICMPBTC-3 (with color component requantization bit number ξ=3), respectively.