Color Image Compression by Moment-Preserving and Block Truncation Coding Techniques

Chen-Kuei Yang, *Member, IEEE*, Ja-Chen Lin, and Wen-Hsiang Tsai

***Abstract*— A new color image compression technique based on moment-preserving and block truncation coding is proposed. An input image is divided into nonoverlapping blocks and each block pixel is assigned one of two representative colors, which are computed with analytic formulas derived from preserving certain moments in the block. A bit map is then generated for each block to represent the pixels' colors. Different uniformity conditions in the representative colors are also identified and utilized to save code bits. Good average compression ratios up to about 13 can be achieved, as shown by experimental results.**

***Index Terms*—Bit map, block truncation coding, color image compression, moment-preserving, uniform color plane.**

I. Introduction

THE block truncation coding (BTC) algorithm developed by Delp and Mitchell [1] is a two-level nonparametric quantizer whose output levels are obtained by preserving the first- and the second-order moments of input samples. Lema and Mitchell [2] applied the absolute moment BTC technique to each of the three color planes of a color image. The compression ratio is image-independent and is always 4. To increase the compression efficiency, Wu and Coll [3] used a single bit map to quantize all of the three color planes so that only one out of three bit planes needs to be preserved. Kurita and Otsu [4] used the mean vector and the covariance matrix of color vectors to compute the principal score for each pixel in the block and classified the pixels into two classes. The two classes' mean vectors and a bit map are preserved. In [3] and [4], the compression ratio is always 6 when the block size is 4×4.

In this paper, a new method based on the moment-preserving principle and the block truncation coding procedure is proposed. An input image is divided into nonoverlapping 4×4 blocks. Several color moments of each block are preserved and a set of analytic formulas are derived accordingly [5]–[7] to compute two representative colors $C_1 = (R_1, G_1, B_1)$ and $C_2 = (R_2, G_2, B_2)$ for the block. Different techniques are applied to different uniformity conditions of the color components in the block. A block is said to be k-spectral (with $0 \leq k \leq 3$) if $3 - k$ of the three color components (R, G, B) of the block are nearly uniform, i.e., if the variations of the values of $3 - k$ color components are very small. Four types of color conditions in a block are identified: 3-spectral, 2-spectral, 1-spectral, and uniform (0-spectral). A 3-b spectral identification (SI) code is used to indicate different color uniformity conditions. This coding method is found to be effective in improving the compression ratio. A bit map of binary values is then generated, with each bit in the map indicating whether a block pixel is assigned to C_1 or C_2. An extended version of the proposed compression method to improve the compression ratio is also proposed.

The remainder of this paper is organized as follows. In Section II, the proposed color image compression method using the moment-preserving and the BTC techniques in the form of an algorithm is described. In Section III, the description of the extended version of the proposed method is included and some experimental results to show the feasibility of the proposed method are presented. Finally, some conclusions are made in Section IV.

II. Proposed Color Image Compression Using Moment-Preserving and BTC Techniques

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

Step 1: Read in a color image f and partition it into nonoverlapping 4×4 blocks f_j.

Step 2: Take a block f_j of f and perform the following steps.

1) Compute a set of moments of the color component values of the pixels in f_j.
2) Evaluate the number of uniform spectral bands based on the variances of the color component values (R, G, B) of the pixels in f_j.
3) Choose a method to compute two representative colors C_1 and C_2 for f_j:
   a) 0-spectral case—the three color planes of f_j are all nearly uniform; just compute the means of the three color planes to form a single color C set $C_1 = C_2 = C$ and set the SI code $= 000$;
   b) 1-spectral case—only one color plane is nonuniform; use the moment-preserving thresholding technique proposed by Tsai [5] to compute two representative color component values for the nonuniform color plane and compute the means of the two uniform color planes as well, to form C_1 and C_2 for f_j, and set the SI code $= 001, 010,$

Paper approved by M. R. Civanlar, the Editor for Image Processing of the IEEE Communications Society. Manuscript received February 15, 1994; revised May 9, 1996. This work was supported in part by the National Science Council, R.O.C., under Grant NSC83-0408-E009-010 and Grant NSC82-0408-E009-314.

C.-K. Yang is with the Department of Information Management, Ming Chuan College, Taipei, Taiwan 111, R.O.C. (e-mail: ckyang@mcu. edu.tw).

J.-C. Lin and W.-H. Tsai are with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 300, R.O.C. (e-mail: whtsai@cis.nctu.edu.tw).

Publisher Item Identifier S 0090-6778(97)08196-8.

TABLE I
CODING AND RECONSTRUCTION RULES USING THE SI CODE

SI Code	Coding rules		Reconstruction rules	
	block spectral condition	corresponding subset of six 8-bit color components	"0" in bit map	"1" in bit map
000	0-spectral	(R_1, G_1, B_1)	(R_1, G_1, B_1)	(none)
001	1-spectral	(R_1, G_1, B_1) and B_2	(R_1, G_1, B_1)	(R_1, G_1, B_2)
010	1-spectral	(R_1, G_1, B_1) and G_2	(R_1, G_1, B_1)	(R_1, G_2, B_1)
100	1-spectral	(R_1, G_1, B_1) and R_2	(R_1, G_1, B_1)	(R_2, G_1, B_1)
011	2-spectral	(R_1, G_1, B_1) and (G_2, B_2)	(R_1, G_1, B_1)	(R_1, G_2, B_2)
110	2-spectral	(R_1, G_1, B_1) and (R_2, G_2)	(R_1, G_1, B_1)	(R_2, G_2, B_1)
101	2-spectral	(R_1, G_1, B_1) and (R_2, B_2)	(R_1, G_1, B_1)	(R_2, G_1, B_2)
111	3-spectral	(R_1, G_1, B_1) and (R_2, G_2, B_2)	(R_1, G_1, B_1)	(R_2, G_2, B_2)

or 100 depending on which two color planes are uniform;

c) 2-spectral case—two color planes are nonuniform; use the feature-preserving clustering technique proposed by Lin and Tsai [7] to compute two representative color component values for each of the two nonuniform color planes and compute the mean of the uniform color plane as well, to form C_1 and C_2 for f_j, and set the SI code $= 011, 110$, or 101 according to which color plane is uniform;

d) 3-spectral case—use the following technique proposed in this study:

i) preserve certain moments and solve the resulting equations to compute the two representative colors $C_1 = (R_1, G_1, B_1)$ and $C_2 = (R_2, G_2, B_2)$;

ii) assign a new color to each pixel p_i in f_j according to whether the original color value (r_i, g_i, b_i) of p_i is closer to (R_1, G_1, B_1) or (R_2, G_2, B_2);

iii) set the SI code $= 111$.

4) Generate a 4×4 bit map for f_j by assigning 0 or 1 to each pixel p_i of f_j according to whether the new color value of p_i is C_1 or C_2.

5) Discard repetitive ones in the six 8-b component values of the two representative colors C_1 and C_2, and take the remaining ones, together with the SI code and the bit map, as output. According to the SI code, the block may be reconstructed using the output data in the future. A table of detailed coding and reconstruction rules using the SI code is shown in Table I.

Step 3: Stop when all blocks of f have been processed.

The set of preserved color moments of block f_j mentioned in Step 2-3di) are as follows:

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

where n is the total number of pixels in f_j; m_r, m_g, m_b are the means of the color component values (r, g, b), respectively; $m_{r^2}, m_{g^2}, m_{b^2}$ are related to the variances of the (r, g, b) values, respectively; and m_{rgb} is related to the correlation within the (r, g, b) values.

The proposed algorithm CICMPBTC uses the 3-b SI code to represent the various spectral cases and saves the bits used to represent the repetitive color component values when an image block has uniform color planes, as described in Step 2-5). The detailed rules are listed in Table I. The receiver can easily reconstruct an image block according to the table after reading the 3-b SI code, the corresponding subset of the six 8-b color component values, and the 16-b bit map. This coding method is found effective in improving the compression ratio.

III. EXTENDED VERSION OF PROPOSED ALGORITHM AND EXPERIMENTAL RESULTS

The human vision cannot distinguish so many colors (of the number of $2^8 \times 2^8 \times 2^8$) generated by the use of 8 b for each color component. Therefore, we may extend the proposed method to obtain better compression ratios with reasonable SNR values by employing a simple technique [8] to reduce the representative bits of each color component from 8 to 5 b. Besides, the resulting 16-b bit maps still occupy over 35% of the output codes. In order to reduce redundancy in the bit map, we may employ the method of [9] to use 6-b indices to represent 64 predefined standard bit maps [9], which are designed according to the observation of the human visual system's sensitivity to edge and line patterns in small image blocks, and to represent the bit map of each image block by one of the 64 indices after matching the bit map with the 64 standard ones.

More specifically, we revise Step 2-4) and Step 2-5) of algorithm CICMPBTC as follows (the resulting algorithm will be called CICMPBTC*):

Step 2-4):

a) Generate a 4×4 bit map M_j for f_j by assigning 0 or 1 to each pixel 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);

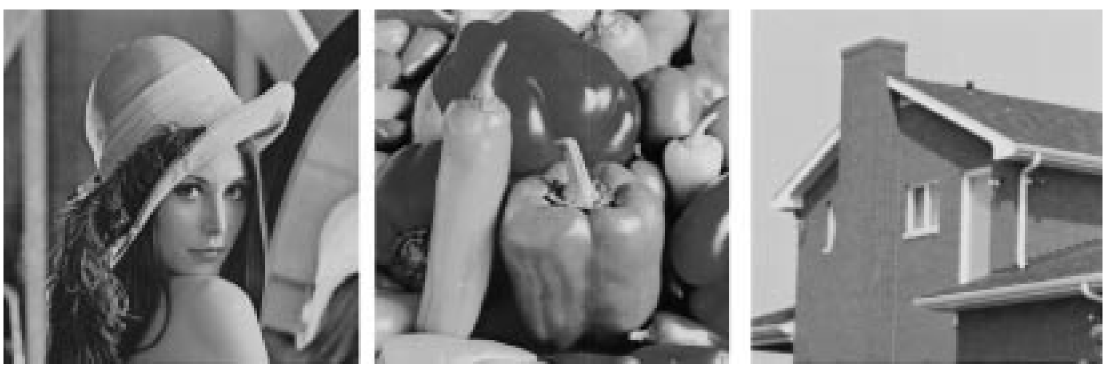
b) compare M_j with each of the 64 predefined standard bit maps by template matching, and let M_k be the one most similar to M_j in the sense of being with the

TABLE II
THE COMPARATIVE SNR VALUES AND COMPRESSION RATIOS OF CONVENTIONAL BTC, SINGLE-BIT-MAP BTC, AND PROPOSED ALGORITHMS CICMPBTC AND CICMPBTC*

	conventional BTC	single bit map BTC	CICMPBTC with 4×4 block size	CICMPBTC with 5×5 block size	*CICMPBTC** with 4×4 block size
Lena	32.85/4.0	29.03/6.0	31.55/7.03	30.95/9.27	29.14/12.29
pepper	32.46/4.0	27.43/6.0	30.93/6.60	30.12/8.80	28.54/11.47
house	39.93/4.0	31.16/6.0	36.77/7.75	33.32/9.81	33.47/13.67
jet	32.10/4.0	28.45/6.0	31.03/7.71	30.53/10.04	28.63/13.59
candy	39.60/4.0	31.04/6.0	37.52/8.20	33.87/10.51	33.20/14.57
balloon	34.93/4.0	28.32/6.0	33.53/7.36	32.26/9.55	31.16/12.92
average	**35.31/4.0**	**29.23/6.0**	**33.56/7.44**	**31.84/9.66**	**30.69/13.09**

(a) (b) (c)

Fig. 1. The standard images of "Lena" (left), "pepper" (middle), and "house" (right) of size 512×512.



(a) (b) (c)

Fig. 2. Images of (a) "Lena," (b) "pepper," and (c) "house" reconstructed from compression results using CICMPBTC with block size 4×4. (a) CR $= 7.03$. (b) CR $= 6.60$. (c) CR $= 7.75$.

smallest number of corresponding pixels with different bit values [9];

c) quantize each color component from 8 to 5 b.

Step 2-5):

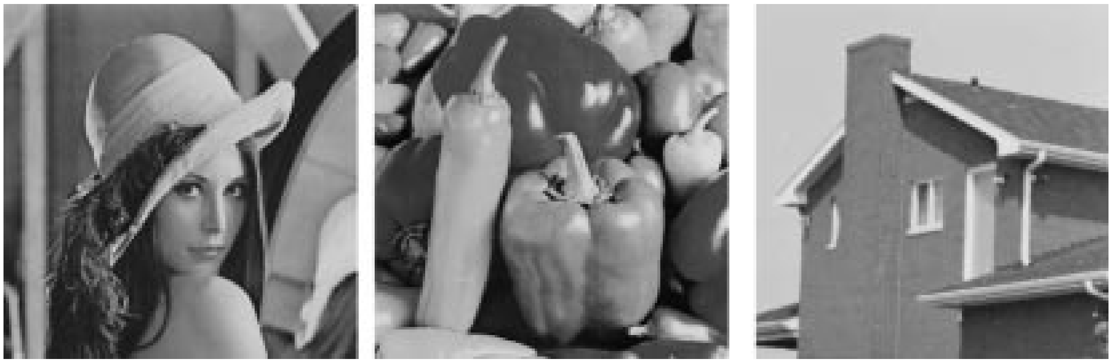
Take the nonrepetitive component values of the two representative colors C_1 and C_2, the 6-b index of M_k, and the SI code as output.

The proposed algorithms CICMPBTC and CICMPBTC*, together with two other methods, have been tested on an IRIS Indigo workstation on several color images. Each color image has 24 b per pixel and is 512×512 in size. We use the signal-to-noise ratio (SNR) as the criterion for performance evaluation of image reconstruction.

Table II shows the comparative SNR values and compression ratios produced by the methods of the conventional BTC with three separated bit maps [3], the BTC with a single bit map [4], and the proposed algorithms with blocks size 4×4 and 5×5. Fig. 1 shows the standard images of "Lena," "pepper," and "house." Figs. 2 and 3 show the images reconstructed from the compression results of Fig. 1 using algorithm CICMPBTC with block sizes 4×4 and 5×5, respectively. Fig. 4 shows the images reconstructed from the compression results of Fig. 1 using algorithm CICMPBTC* with block size 4×4. By carefully checking the third and the last columns of Table II, we can see that the experimental results produced by CICMPBTC* are better than those of [4] both in compression ratios and SNR values.

IV. CONCLUSION

A new approach to color image compression based on the moment-preserving principle and the BTC technique has been



(a) (b) (c)

Fig. 3. Images of (a) "Lena," (b) "pepper," and (c) "house" reconstructed from compression results using CICMPBTC with block size 5×5. (a) CR$=9.27$. (b) CR $=8.80$. (c) CR $=9.87$.

(a) (b) (c)

Fig. 4. Images of (a) "Lena," (b) "pepper," and (c) "house" reconstructed from compression results using CICMPBTC* with block size 4×4. (a) CR $=12.29$. (b) CR $=11.47$. (c) CR $=13.67$

proposed. The performance of the proposed method has been compared with those of the conventional BTC (with three bit maps) and the single-bit-map BTC methods and is found better than these methods. An average of 13.09 in compression ratio values and an average of 30.69 in SNR values can be achieved from the tested images if the extended version of the proposed algorithm is applied.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers of this paper for their insightful comments and valuable suggestions.

REFERENCES

[1] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Trans. Commun.*, vol. COM-27, pp. 1335–1342, 1979.

[2] M. D. Lema and O. R. Mitchell, "Absolute moment block truncation coding and its application to color images," *IEEE Trans. Commun.*, vol. COM-32, pp. 1148–1157, 1984.

[3] Y. Wu and D. C. Coll, "Single bit map block truncation coding for color image," *IEEE Trans. Commun.*, vol. COM-35, pp. 352–356, Mar. 1987.

[4] T. Kurita and N. Otsu, "A method of block truncation coding for color image compression," *IEEE Trans. Commun.*, vol. COM-35, pp. 352–356, Mar. 1987.

[5] W. H. Tsai, "Moment-preserving thresholding: A new approach," *CVGIP*, vol. 29, pp. 377–393, 1985.

[6] C. K. Yang, C. T. Wu, J. C. Lin, and W. H. Tsai, "Color image sharpening by moment-preserving technique," *Signal Processing*, vol. 45, no. 3, pp. 397–403, 1995.

[7] J. C. Lin and W. H. Tsai, "Feature-preserving clustering of 2D data for two-class problems using analytical formulas: An automatic and fast approach," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 554–560, 1994.

[8] B. J. Kurz, "Optimal color quantization for color displays," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1983, pp. 217–224.

[9] K. Yang and W. H. Tsai, "Improving block truncation coding using line and edge information for gray-scale image compression," *Pattern Recognit. Lett.*, vol. 16, pp. 67–75, 1995.