Novel Block Truncation Coding of Image Sequences for Limited-Color Display

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

[1]Dept. of Electrical Engineering, National Taiwan University,
Taipei, Taiwan, R. O. C. Email address: pei@cc.ee.ntu.edu.tw
[2]Telecommunication Lab, Chunghwa Telecom Co.,
Taiwan, R.O.C. Email address: cmc@ms.tl.gov.tw

Abstract. To compress color image sequences, a novel color Block truncation coding (BTC) algorithm, called quaternion-moment block truncation coding (QMBTC), is presented in this paper. The QMBTC is derived by using quaternion arithmetic and the moment-preserving principle. To compress color image sequences for limited-color display, we combine the QMBTC with the procedures of coded/uncoded block determination and color palette design. The advantage of the proposed coder is in the low decoder complexity. It can decode the compressed image in real-time without the assistance of specific hardware for limited-color display.

1 Introduction

The fast evolution of computers and telecommunications has made multimedia services become more and more popular. Multimedia is the combination of various forms of communication (e.g. audio, graphics, video) onto common platforms, PCs or workstations. One example of such multimedia services is video data retrieval system, which can be applied to digital library, video on demand, and video mail. To provide a useful video retrieval system, the storage and display of huge digitized video data have become the main challenges. For storage, some existing standards, such as H.261 and MPEG [1], have been proposed to compress the digitized video data. Both of H.261 and MPEG employ DCT and motion estimation techniques to reduce the spatial and temporal correlation, respectively [1]. Nevertheless, due to high computation complexity, implementations of these compression techniques should rely on special hardwares. On the other hand, full-color display of digitized video data utilizes 24 bits to specify the color of each pixel on the screen. In this way, the 16 million colors can be selected by the users. However, most of the current PCs and workstations usually don't support full-color display devices because of the costs of high speed video RAM. They generally have a single 8-bit frame buffer, capable of displaying 256 colors simultaneously at a time. The transformation of a original full-color image sequence into a limited-color image sequence requires the design of the color palette [2].

To release dependency of video retrieval applications from relative expensive hardwares, we present a software-only lossy compression scheme of color image sequences for limited-color display in this paper. This scheme is based on a novel color block truncation coding (BTC) algorithm, called quaternion-moment block truncation coding (QMBTC) proposed by Pei and Cheng [3]. In contrast to existed color BTC algorithms [5,6], which process on each transformed component of input color data, QMBTC generalizes conventional monochrome BTC [4] to color BTC by expressing input color space as a quaternion-valued space. To apply QMBTC, an input color image sequence is first partitioned into three dimensional (3D) pixel blocks in spatial-temporal domain. Then, QMBTC is used to quantize pixel blocks after a procedure of coded/uncoded block determination. In addition, for limited-color display, the color palette for the decoded sequence is obtained by detecting significant color frames and using dependent scalar quantization technique proposed by Pei and Cheng [7]. With the help of this color palette, the outputs of the proposed compression scheme can produce a compressed image sequence with limited colors.

This paper is organized as follows: Section 2 extends the QMBTC algorithm to compress color image sequences. In Section 3, we describe the color palette design for image sequence. Then implementation of the proposed limited-color coder is proposed in Section 4. Finally, the performance of proposed coded is evaluated and conclusion is made in Section 5.

2 Compression of Color Image Sequences

The algebra of the quaternions is the generalization of complex numbers [3]. Considering a 4D real-valued data set $\mathrm{H} = \{(q_0(n), q_1(n), q_2(n), q_3(n))\}_{n=1}^{N}$, a quadruple data point $(q_0(n), q_1(n), q_2(n), q_3(n))$ can be expressed as a quaternion number $\hat{q}(n) = q_0(n) + q_1(n) \cdot i + q_2(n) \cdot j + q_3(n) \cdot k$, where 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 = R, q_2 = G, q_3 =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 $\mathbf{a} = (\mathrm{q_1(n), q_2(n), q_3(n)})$ and $b = q_0(n)$. The conjugate $\hat{q}^*$ of $\hat{q}$ is defined as: $\hat{q}^* = - < \mathbf{a}, \mathbf{b} >= \mathrm{q_0} - (\mathrm{q_1} \cdot \mathrm{i} + \mathrm{q_2} \cdot \mathrm{j} + \mathrm{q_3} \cdot \mathbf{k})$

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:

$$\hat{m}_1 = \mathbf{E}[\hat{\mathrm{q}}], \qquad \hat{\mathrm{m}}_2 = \mathbf{E}[\hat{\mathrm{q}} \cdot \hat{\mathrm{q}}^*], \qquad \hat{\mathrm{m}}_3 = \mathbf{E}[\hat{\mathrm{q}} \cdot \hat{\mathrm{q}}^* \cdot \hat{\mathrm{q}}] \tag{1}$$

where $\mathbf{E}[\bullet]$ 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.

2.1 The Algorithm

In this section, we extend the QMBTC algorithm [3] to compress color image sequences.

After partitioning an input image sequence into several groups of frames (GOF) in temporal dimension as depicted in Fig. 1, the QMBTC can be described as follows:

Step 1. Divide each GOF into small non-overlapping 3D pixel blocks with size $S_x \times S_y \times S_t$, where S_x and S_y are spatial sizes of the pixel block, and S_t is the depth of the pixel block.

Step 2 Express the color value of each pixel **I**, (I_1, I_2, I_3), by the quaternion number, 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

$$\begin{aligned} p_0 \cdot \hat{z}_0 + p_1 \cdot \hat{z}_1 &= \hat{m}_1 \\ p_0 \cdot \hat{z}_0 \cdot \hat{z}_0^* + p_1 \cdot \hat{z}_1 \cdot \hat{z}_1^* &= \hat{m}_2 \\ p_0 \cdot \hat{z}_0 \cdot \hat{z}_0^* \cdot \hat{z}_0 + p_1 \cdot \hat{z}_1 \cdot \hat{z}_1^* \cdot \hat{z}_1 &= \hat{m}_3 \\ p_0 + p_1 &= 1 \end{aligned} \qquad (2)$$

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, $\hat{m}_1, \hat{m}_2, \hat{m}_3$, unchanged 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.

2.2 Application to Compression of Color Image Sequences

As it is known, the RGB space has extensive correlation among color components. Besides, pixels within the block, which does not contain edge or scene change, are likely to have spatial and temporal correlation, which results in similar color values. 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, where $Y = (R+G+B)/3, \quad r = R/(R+G+B), \quad g = G/(R+G+B)$.

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

$$\begin{aligned} w_1 = 1 - \exp(-\frac{T}{\rho}), \quad w_2 &= (\exp(-\frac{T}{\rho}))/2, \quad w_3 = (\exp(-\frac{T}{\rho}))/2 \\ where \quad T &= \frac{1}{N}\sum_{n=1}^{N} |(Y(n) - \bar{Y}| \end{aligned} \tag{3}$$

In (3), 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 (3), 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. But, there might be different colors existed within the pixel block. In this situation, 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.

3 Color Palette Design for Image Sequences

To design the color palette for a image sequence, we assume that the distribution of color values between consecutive frames doesn't change rapidly. Through this, we first detect the significant color frames. The color pixels of these significant color frames are then used to design the color palette by a color quantization method,called scalar dependent quantization (DSQ) proposed by Pei and Cheng [7]. The DSQ will partition the color space of image in a dependent way in order to fully utilize the correlations of color points of input image in color space. The dependent way partitions color space sequentially according to the order of color components.

3.1 Detection of Significant Color Frames

The color information used in the detection of significant color frames is from the Yrg space. Then an image processing method, which compares the normalized histograms of r and g between consecutive frames in order to detect significant color frames in a image sequence, is implemented. The principle behind this method is that two frames having a unchanging background and objects will show little difference in their respective normalized r and g histograms. The normalized histogram difference of chromatic components between the a-th and b-th frame is denoted by the following formula:

$$NHD_{a,b} = \sum_{l=1}^{L_r} \frac{(h_{r,a}(l) - h_{r,b}(l))^2}{h_{r,a}(l)} + \sum_{l=1}^{L_g} \frac{(h_{g,a}(l) - h_{g,b}(l))^2}{h_{g,a}(l)} \tag{4}$$

where $h_{r,a}(l)$ and $h_{g,a}(l)$ are the histograms of chromatic components r and g, whose values are designated as l, at frame a. If the sum of $NHD_{a,b}$ is larger than a given threshold Ω_h, the b-th frame is declared as a significant color frame. For empirical results shown in Sec. 6, both of L_r and L_g are equal to 255. and Ω_h is set to be 2000.

4 Proposed Limited-Color Coder

To avoid the time-consuming computation such as full-search motion estimation in the H.261 standard, the QMBTC-based coder proposed in this paper classifies pixel blocks into two types, changed blocks and unchanged blocks, which are then coded and uncoded respectively. The entire process of coding image sequence is illustrated in Fig. 2. We first describe the operation of coded/uncoded block decider. If the absolute $\bar{Y}$ difference between the processed pixel block $BK(x_B, y_B, t_B)$ and its preceding adjacent block in temporal dimension $BK(x_B, y_B, t_B - 1)$, $d_{\bar{Y}}(\ (x_B, y_B, t_B),\ (x_B, y_B, t_B - 1)\ )$, is beyond the threshold $\Omega_{\bar{Y}}$ and the color difference between these two blocks exceeds the threshold Ω_c, then the current pixel block is denoted as a changed one. Otherwise, current pixel block is announced as a unchanged one. The color difference $d_c(\ (x_B, y_B, t_B),\ (x_B, y_B, t_B - 1)\ )$ between processed blocks is defined as: $d_c(\ (x_B, y_B, t_B),\ (x_B, y_B, t_B - 1)\ ) = \sum_{n=0}^{N-1} |\mathbf{I}(n, t_B) - \mathbf{I}(n, t_B - 1)|$,where

N is the number of pixels in a block. Absolute value term in color difference d_c represents sum of absolute (R,G,B) color component difference between two pixels.

Only the changed blocks are coded and transmitted. This scheme is similar to conditional replenishment proposed by [8]. The coding is first performed by the QMBTC, which decides the number of classes in a pixel block and assigns the centroid of each class as a reproduction color. In addition, for limited-color display, a color palette for the processed image sequence is produced by the method described in Sec. 4. Using this color palette, a color index is assigned to each reproduction color in the sequence. This results in a compressed image sequence with limited colors. The search of the associated color index for each representation color is based on the well organized tree structure of color palette. Finally, the bit-maps and the color indices of reproduction colors of all coded pixel blocks will then be sent to the receiving end waiting for decoding. Through this coding scheme, a bit is required to represent whether the processed pixel block is coded or uncoded. And a coded one-class pixel block needs a bit and the associated color index. To decode each compressed pixel block, the bitmap, color indexes of each pixel block, and the color palette are the only required elements. It can be finished through a lookup table manner in real-time for limited-color display devices.

5 Experimental Results and Conclusions

To evaluate the performance of the proposed limited-color coder, we conducted the experiments on three standard test sequences with frame size 352x240, 'Miss America', 'Salesman', and 'Susie'. In Table 1, we illustrate the influence of block size on the proposed coder, which is the coding results on average of 150 frames with $\rho = 3.$. From empirical results, we notice that this choice of ρ is a good compromise between bit rate and APSNR performance. $\rho = 3.$ is also selected in the following experiments. In this table, $\Omega_{\bar{Y}}$ and Ω_c were set to be 1.0 and 45.0 respectively. We see that the larger the block size, the lower is the bit-rates with the exception of 4×4×3 size of 'salesman'. This incomparable case is mainly due to the motion part of 'salesman', in which color values of pixel blocks fluctuate more extensively when the depth of pixel block becomes larger. Therefore, more coded blocks are detected. Concerning the performance of APSNR, the larger the block size, the lower is APSNR value.

For the purpose of performance evaluation, the proposed coder with block size $4 \times 4 \times 2$ is compared with the H.261 standard. The motion estimation method used in the H.261 standard is the full-search sum-absolute-difference (SAD) algorithm [1] with search range set to be 15. Bit-rates of the proposed coder and the H.261 standard for test sequence 'Salesman' is displayed in Fig. 3(a). As we observe, the proposed coder outputs variable bit-rates. To obtain a limited-color decoded image sequence, the H.261 standard is passed through the well-known Heckbert's Median-Cut algorithm [2] for color quantization. The color palette for the full-color decoded image sequence of the H.261 standard is obtained by the same procedures as the proposed coder except that color quantization method is replaced by the Median-Cut algorithm.

For the same average bit-rates of the proposed coder and the H.261 standard with color quantization, APSNR of limited-color decoded image sequences are illustrated in Fig. 3(b). In these results, the size of color palette is 256. The average bit-rates of 150 frames for 'Salesman' is 0.28. From the APSNR performance curves of Fig. 3, we notice that the performance of the proposed coder is above 30 dB and comparable to that of the H.261 standard with color quantization. However, to encode image sequences by the H.261 standard, the computation cost of motion estimation will be overwhelming without the assistance of hardware. To store a 256 color decoded image sequence of the H.261 standard with color quantization, it still requires 8 bits/pixel. Nevertheless, the proposed coder only needs low bit-rates (below 0.6 bits/pixel) to store limited-color image sequences. And the computation cost of the proposed coder is reasonable when implemented in software.

References

[1] D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Application," *Comm. ACM*, vol. 34, no. 4, pp. 46-58, Apr. 1991.

[2] P. Heckbert, "Color Image Quantization for Frame Buffer Display," *Comput. Graph.*, vol. 16, no. 3, pp. 297-307, July 1982.

[3] S. C. Pei and C. M. Cheng, "A Novel Block Truncation Coding of Color Images by Using Quaternion-Moment-Preserving Principle," in *Proc. IEEE ISCAS'96*, pp. 684-687, Atlanta, Ga., U.S.A., May 1996.

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

[5] H. Huang and J. Wu, "Novel Real-time Software-based Video Coding Algorithms,"*IEEE Trans. Consumer Electronics*, vol. 39, no. 3, pp 570-580 , Aug. 1993.

[6] L. Chen and Y. Liu, " a High Quality MC-OBTC Codec for Video Signal Processing,"*IEEE Trans. Circuit and Syst. for Video Technol.*, vol. CSVT-4, no. 1, pp. 92-98, Feb. 1994.

[7] S. C. Pei and C. M. Cheng, "Dependent Scalar Quantization of Color Images," *IEEE Trans. Circuit and System for Video Technology*, vol. CSVT-5, no. 2, pp. 124-139, Apr. 1995.

[8] J. C. Candy, M. A. Franke, B. G. Haskell, and F. W. Mounts, " Transmitting Television as Clusters of Frame-to-frame Differences,"*Bell Syst. Tech. J.*, vol. 50, pp. 1889-1919, July. 1971.

Test Sequences	Block Size	APSNR (dB)	Bit-Rate (bits/pixel)
Miss America	I	33.9	0.468
	II	33.5	0.290
	III	33.2	0.247
	IV	31.7	0.209
Salesman	I	30.6	0.551
	II	30.6	0.285
	III	30.4	0.308
	IV	28.5	0.228
Susie	I	34.1	0.591
	II	33.9	0.564
	III	33.6	0.563
	IV	31.6	0.461

Table 1. Performance comparison of the proposed QMBTC-based coder with different block sizes: I - 4 × 4 × 1, II - 4 × 4 × 2, III - 4 × 4 × 3, IV - 8 × 8 × 2

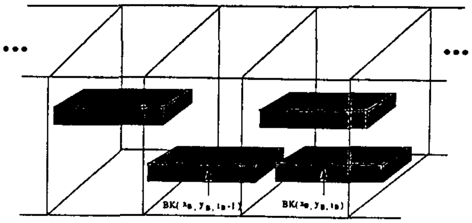

Fig. 1. Partitioning an input sequence into 3D pixel blocks.

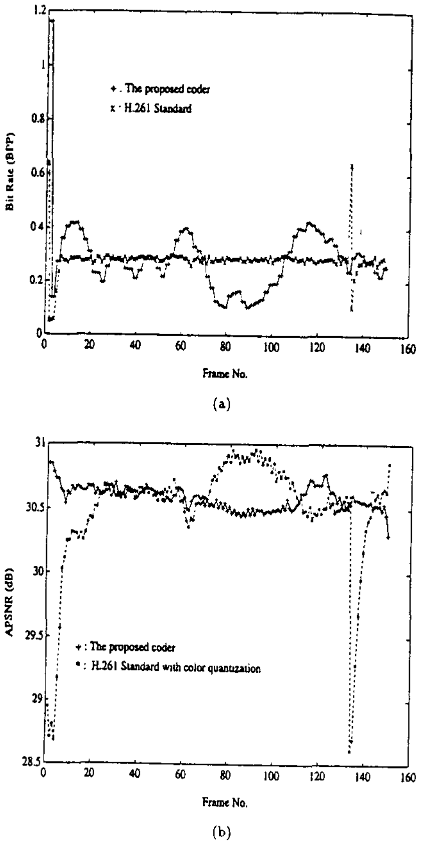

Fig.3. Testing results of 'Salesman' (a) bit-rate comparison of the proposed coder and the H.261 standard (b) APSNR comparison of the proposed coder and the H.261 standard with color quantization.

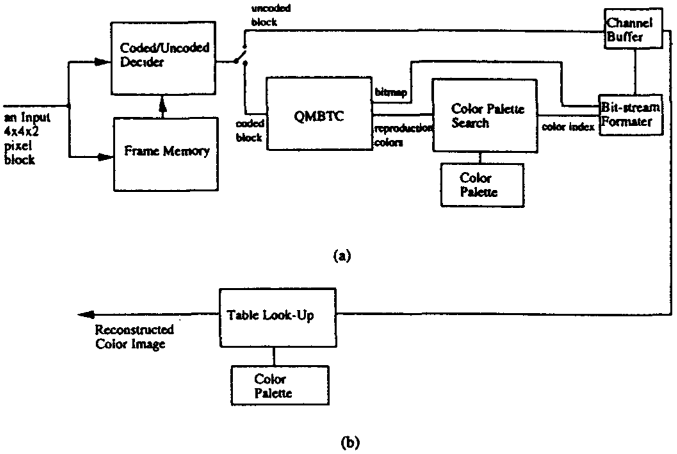

Fig. 2. The entire coding process of the proposed coder (a) encoder (b) decoder.