A New Approach To Compress Color-indexed Images

Yuk-Hee Chan, Ka-Chun Lui and P. K. Lun

Centre for Multimedia Signal Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong SAR

Abstract

A lossless compression scheme for coding color-indexed images is proposed in this paper. This two-stage scheme first turns the index map of a color-indexed image into a new index map each element of which serves as an index to a pixel-dependently reordered version of the palette. Consequently, it reduces the zero-order entropy and the energy of the index map and removes the spatial correlation among pixels significantly. A matching coding scheme is then exploited to encode the new index map effectively. Simulation results show that the proposed compression scheme is superior to other state-of-art lossless compression schemes for coding color-indexed images in terms of compression performance.

I. Introduction

A color-indexed image is represented with a color index map each element of which serves as an index to select a color from a predefined set of colors called palette to represent the color of a pixel in the image[1]. Two completely different colors can be of similar index values in a palette. Hence, it is always a challenging task to compress a color-indexed image as the compression must be lossless and predictive coding techniques are generally not effective to predict an index based on the spatial correlation of the index map.

A new lossless compression scheme for coding color-indexed images is proposed in this paper. This scheme can be decomposed into two stages. In stage 1, it raster scans the image to be encoded and, on the fly, reassigns indices to palette colors pixel by pixel adaptively based on the predicted color of the current pixel and an updated statistical study of the processed image pixels. As a result, it produces an index map of very low zero-order entropy, little energy and little spatial correlation. In stage 2, the resultant index map is further decomposed into a number of holey binary bit planes and then each of them is encoded with a context-based binary arithmetic coder. Simulation results show that the performance of the proposed compression scheme outperforms the other state-of-art lossless compression schemes in terms of output bit rate[2-11].

Some preliminary results of this work were reported in [12]. In contrast to the scheme presented in [12], the compression scheme presented in this paper builds extra context-based probability models and merges them adaptively such that one can assign a smaller index value to a more likely palette color when processing a pixel. Besides, it decomposes the output index plane into binary bit planes based on the index value instead of the bit significance and adopts a matched binary coding scheme. With such advancements, it further reduces the average bit rate of a compression output by 0.4 bits per pixel.

II. Pixel-wise Palette Reordering

Stage 1 of the proposed coding scheme is inspired by the palette reordering methods. Since the index assignment of the palette is pixel-dependent, the proposed reordering method used in this stage is referred to as pixel-wise palette reordering method (PPR) so as to discriminate it from the conventional palette reordering methods in which a palette of fixed index assignment is designed for all pixels to share.

A. Algorithm

Let the input color-indexed image be **X** and the associated palette be $\Omega=\{\vec{c}_k \mid k=0,1\ldots M-1\}$, where M is the size of the palette and $\vec{c}_k$ is the k^{th} palette color in Ω. Without loss of generality, we assume all $\vec{c}_k$ in Ω are sorted according to their luminance and $\vec{c}_0$ (i.e. k=0) is the one of the minimum luminance. Note that this criterion can be easily satisfied through an initialization process. This sorted palette is used as a reference palette in the codec.

Based on the index map of **X**, a full-color image can be constructed with palette Ω. The image is raster scanned and processed. For each pixel, the intensity values of its three color components are individually predicted with their own corresponding color planes by using the MED predictor used in JPEG-LS[13].

Suppose the prediction results of the red, the green and the blue color components of the current pixel (i,j) are, respectively, $r(i,j)$, $g(i,j)$ and $b(i,j)$. In vector form, the prediction result of pixel (i,j) is $\vec{v}(i,j)=(r(i,j),g(i,j),\ b(i,j))$. $\vec{v}(i,j)$ is then color quantized with palette Ω. In practice, the quantization result could be different from the original color of pixel (i,j). Without loss of generality, we assume that the quantization result and the original color of pixel (i,j) are, respectively, the p^{th} and the r^{th} palette colors in Ω. For the purpose of reference, they are denoted as $\vec{c}_p$ and $\vec{c}_r$ respectively.

In the proposed reordering method, statistics about the frequency of the occurrence of specific events are collected when processing the image such that the scheme can learn from the experience to improve its reordering performance adaptively. Specifically, we build five simple probability models and then merge them to form a combined probability model. Five separate tables are constructed for building these five probability models.

Proceedings of the Second APSIPA Annual Summit and Conference, pages 318–321,
Biopolis, Singapore, 14-17 December 2010.

The first table $\{T_d(m,n)|m,n=0,1\ldots M\text{-}1\}$ is referred to as *discrepancy frequency table* (DF-Table) as its entry $T_d(m,n)$ records the occurrences of encountering a pixel whose quantized predicted color and real color are, respectively, $\vec{c}_m$ and $\vec{c}_n$ up to the moment when pixel (i,j) is processed. With this table, the conditional probability that $\vec{c}_k$ is the real color of pixel (i,j) given that $\vec{c}_p$ is the quantized predicted color of the pixel can then be estimated as

$$P_p(k\,|\,p) = T_d(p,k)/\Sigma_{u=0}^{M\text{-}1}T_d(p,u) \qquad \text{for } k=0,1\ldots M\text{-}1 \quad (1).$$

All $T_d(m,n)$ values are initialized to 1 at the very beginning and the DF-table is updated after a pixel is processed.

The other four tables, which are denoted as $\{T_N(m,n)|m,n=0,1\ldots M\text{-}1\}$, $\{T_W(m,n)|m,n=0,1\ldots M\text{-}1\}$, $\{T_{NW}(m,n)|\ m,n=0,1\ldots M\text{-}1\}$ and $\{T_{NE}(m,n)|m,n=0,1\ldots M\text{-}1\}$, are constructed to reflect how likely that a particular color in Ω is the real color of a pixel when the color of one of the four processed connected neighbors (i.e. the northern, the western, the northwestern and the northeastern) of the pixel is given. For example, $\{T_N(m,n)|m,n=0,1\ldots M\text{-}1\}$ records the occurrences of encountering a pixel whose northern neighbor's real color is $\vec{c}_m$ while its own real color is $\vec{c}_n$ up to the moment when pixel (i,j) is processed. Entry values of all these tables are also initialized to 1 at the very beginning and all tables are updated after a pixel is processed.

Without loss of generality, here we assume that the real colors of pixels $(i,j\text{-}1)$, $(i\text{-}1,j\text{-}1)$, $(i\text{-}1,j)$ and $(i\text{-}1,j+1)$ are, respectively, the r_1^{th}, the r_2^{th}, the r_3^{th} and the r_4^{th} palette colors in Ω. Based on the four tables, four additional context-based probability models can then be derived as

$$P_{c1}(k\,|\,r_1) = T_W(r_1,k)/\Sigma_{u=0}^{M\text{-}1}T_W(r_1,u)$$
$$P_{c2}(k\,|\,r_2) = T_{NW}(r_2,k)/\Sigma_{u=0}^{M\text{-}1}T_{NW}(r_2,u)$$
$$P_{c3}(k\,|\,r_3) = T_N(r_3,k)/\Sigma_{u=0}^{M\text{-}1}T_N(r_3,u)$$
$$P_{c4}(k\,|\,r_4) = T_{NE}(r_4,k)/\Sigma_{u=0}^{M\text{-}1}T_{NE}(r_4,u) \qquad \text{for } k=0,1\ldots M\text{-}1 \quad (2).$$

Each of them tells how likely that $\vec{c}_k$ is the real color of pixel (i,j) when the real color of a particular neighbor of pixel (i,j) is given.

In our approach, when assigning a new color index to a palette color, probability models $P_p(k\,|\,p)$, $P_{c1}(k\,|\,r_1)$, $P_{c2}(k\,|\,r_2)$, $P_{c3}(k\,|\,r_3)$ and $P_{c4}(k\,|\,r_4)$ are merged to form a single probability model to determine the likeliness that the concerned palette color is the real color of the pixel currently processed. In particular, the combined probability model is given as

$$P(k) = L(k)/\Sigma_{u=0}^{M\text{-}1}L(u) \qquad \text{for } k=0,1\ldots M\text{-}1 \quad (3),$$

where

$$L(k) = w_0T_d(p,k) + w_1T_W(r_1,k) + w_2T_{NW}(r_2,k) + w_3T_N(r_3,k) + w_4T_{NE}(r_4,k) \qquad \text{for } k=0,1\ldots M\text{-}1 \quad (4).$$

Weighting factors w_0, w_1, w_2, w_3 and w_4 are used to weight the contribution of the five probability models according to their significance in the merging. The determination of their values will be discussed in the following section. The larger the value of $L(k)$, the more likely that $\vec{c}_k$ is the real color of pixel (i,j).

It is possible that some neighbors of pixel (i,j) are out of the image boundary. In that case, corresponding items are removed from eqn. (4) when computing $L(k)$. For example, when one processes a pixel on the left boundary of an image, the western and the northwestern neighbors are missing and hence eqn. (4) becomes $L(k)= w_0T_d(p,k) + w_3T_N(r_3,k) + w_4T_{NE}(r_4,k)$.

Once $L(k)$ is determined for pixel (i,j), the colors in palette Ω are adaptively reordered based on $L(k)$. In particular, $\vec{c}_k$'s are sorted according to the values of $L(k)$ for $k=0,1\ldots M\text{-}1$ in descending order. If there exist two different colors $\vec{c}_l$ and $\vec{c}_d$ such that $L(l)=L(d)$, $\vec{c}_l$ and $\vec{c}_d$ will be sorted according to their Euclidean distances to $\vec{c}_p$. The closer one is put in front of the other. If they are still not distinguishable, their order will be determined by their ranking in reference palette Ω.

The position of $\vec{c}_r$ in the newly reordered queue can be used as an index to the queue and is used to represent the pixel in the output of the reordering method. Note the queue forms a transient version of palette Ω. After processing this pixel, $T_d(p,r)$, $T_W(r_1,r)$, $T_{NW}(r_2,r)$, $T_N(r_3,r)$ and $T_{NE}(r_4,r)$ are incremented by 1 to update the frequency count of corresponding events. If necessary, weighting factors w_l for $l=0,1,..4$ can also be updated accordingly. The algorithm proceeds to process next pixel in the scanning path by repeating the same procedures until all pixels are processed.

In the decoder, to decode a pixel, the same process is carried out to determine the same transient version of palette Ω. As soon as the index for the pixel is received, it can be used to fetch the corresponding color in the transient version of palette Ω to reconstruct (i) a color index map all elements of which share a single palette of fixed assignment of color indices such as Ω, or even (ii) the full-color image directly.

B. Determination of weighting factors w_l

Weighting factors w_l for $l=0,1,..4$ can be adjusted according to a cost function to favor the models that provide better performance after assigning a new color index to the color of the pixel being processed.

In particular, the cost function of the combined probability model $P(k)$ can be defined as

$$\begin{aligned} J &= -\log_2 P(r) = -\log_2\left(L(r)/\Sigma_{u=0}^{M\text{-}1}L(u)\right) \\ &= \log_2\left(\Sigma_{u=0}^{M\text{-}1}L(u)\right) - \log_2 L(r) \end{aligned} \qquad (5),$$

which can be regarded as the estimated information contained in the color index of pixel (i,j). Accordingly, J will be the optimal codeword length with respect to the combined probability model for coding the color index directly.

J should be minimized with respect to w_l with the restriction that $w_l \geq 0$ for all l=0,1…4. To achieve this, in our proposal, all weights are initialized as w_l =1 and adjusted along the cost gradient with gradient descent method to reduce J as follows.

$$w_l = \max\left(0, w_l - \ln 2 \cdot \frac{\partial J}{\partial w_l}\right) \qquad \text{for all } l=0,1\ldots4 \quad (6)$$

where $\frac{\partial J}{\partial w_l}$ is given by

$$\frac{\partial J}{\partial w_0} = \left(\frac{\Sigma_{u=0}^{M-1} \mathrm{T}_d(p,u)}{\Sigma_{u=0}^{M-1} L(u)} - \frac{\mathrm{T}_d(p,r)}{L(r)}\right)\frac{1}{\ln 2}$$

$$\frac{\partial J}{\partial w_1} = \left(\frac{\Sigma_{u=0}^{M-1} \mathrm{T}_W(r_1,u)}{\Sigma_{u=0}^{M-1} L(u)} - \frac{\mathrm{T}_W(r_1,r)}{L(r)}\right)\frac{1}{\ln 2}$$

$$\frac{\partial J}{\partial w_2} = \left(\frac{\Sigma_{u=0}^{M-1} \mathrm{T}_{NW}(r_2,u)}{\Sigma_{u=0}^{M-1} L(u)} - \frac{\mathrm{T}_{NW}(r_2,r)}{L(r)}\right)\frac{1}{\ln 2}$$

$$\frac{\partial J}{\partial w_3} = \left(\frac{\Sigma_{u=0}^{M-1} \mathrm{T}_N(r_3,u)}{\Sigma_{u=0}^{M-1} L(u)} - \frac{\mathrm{T}_N(r_3,r)}{L(r)}\right)\frac{1}{\ln 2} \qquad \text{and}$$

$$\frac{\partial J}{\partial w_4} = \left(\frac{\Sigma_{u=0}^{M-1} \mathrm{T}_{NE}(r_4,u)}{\Sigma_{u=0}^{M-1} L(u)} - \frac{\mathrm{T}_{NE}(r_4,r)}{L(r)}\right)\frac{1}{\ln 2} \qquad (7).$$

The step size ln2 in eqn.(6) is selected to compensate for the factor 1/ln2 that appears in eqn.(7), which helps to reduce the realization effort of eqn.(6).

III. Coding of Reindexed Indexed Maps

To achieve a higher compression ratio, the reindexed index map is decomposed into M-1 bit planes as follows.

$$B_k(i,j) = \begin{cases} 1 & \text{if } I(i,j) > k \\ 0 & \text{if } I(i,j) = k \\ \text{don't care} & \text{if } I(i,j) < k \end{cases} \qquad \text{for } k=0,1,\ldots M\text{-}2 \qquad (8)$$

where $B_k(i,j)$ is the $(i,j)^{\text{th}}$ element of the k^{th} bit plane and $I(i,j)$ is the new index value of pixel (i,j).

Starting from k=0, bit planes are gradually constructed as k increases. Once a bit plane is defined, its bits are raster scanned and encoded with context-based entropy coding. In other words, bit planes of lower k values are encoded first. If $B_k(i,j)$ is a don't care bit, there must be $B_l(i,j)$=0 for a particular $l<k$ and hence it need not be encoded. As a consequence, the bits left behind to be encoded form a holey binary bit-plane B_k.

Each holey binary bit-plane B_k is encoded with a context-based binary arithmetic coder. Figure 1 shows the general form of the context templates used in the proposed coding scheme. It is possible that the binary context of $B_k(i,j)$ contains some don't care pixel bits when $B_k(i,j)$ is encoded with context-based entropy coding. In that case, the don't care bits are filled with 0.

As the value of k gets larger, there are more don't care bits in bit plane B_k and, accordingly, the context template covers more don't care bits when it moves around. To avoid context dilution problem, a context template of smaller size is used to encode a bit plane of larger k.

As shown in Figure 1, the pixel positions of the context template are numbered. Instead of using all template locations, only first L positions are used, where L is a function defined as

$$L = \lceil 9 - \log_2(k+1) \rceil, \qquad \text{for } k=0,1,\ldots M\text{-}2 \quad (9).$$

In the suggested binary arithmetic coder, the forgetting factor α and the biasing constant Δ for updating the conditional probability of having bit '1', $P(1|context)$, are, respectively, 0.985 and 0.006. In particular, the probability of having bit '1' when the context is binary pattern i again is updated by

$$P(1 \mid context = binary\ pattern\ i) = (t_i + \Delta)/(s_i + 2\Delta) \qquad \text{for } i=0,1\ldots4095 \quad (10)$$

where t_i and s_i are updated whenever a context of binary pattern i is encountered using

$$t_i := \begin{cases} \alpha t_i + 1 & \text{if the current pixel value is } 1 \\ \alpha t_i & \text{else} \end{cases} \qquad (11)$$

and

$$s_i := 1 + \alpha s_i \qquad (12)$$

The initial values of t_i and s_i are, respectively, 1 and 2 for all context patterns i.

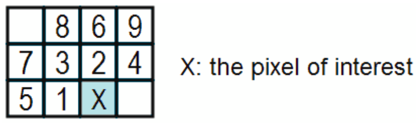

Fig 1 The context template used in our bit-plane coding scheme.

IV. Simulation Results

Simulations were carried out to evaluate the performance of the proposed coding scheme. To have a fair comparison with the performance of other state-of-art lossless coding schemes, all the testing images used in the simulations were obtained from the ftp site ftp://ftp.ieeta.pt/~ap/images. They are organized into 7 groups: (1) a set of 18 synthetic images referred to as *Synthetic*, (2)-(4) 6-, 7- and 8-bit color-quantized versions of image set *Natural1* which contains 23 natural images in the "Kodak" database, and (5)-(7) 6-, 7- and 8-bit color-quantized versions of image set *Natural2* which contains 12 popular natural images.

Table 1 lists the compression performance achieved by various lossless compression schemes for coding color-indexed images in terms of bits per pixel (bpp). Among these evaluated compression schemes, [3], [4] and [7] are based on conventional palette reordering. Their outputs can be encoded with either JPEG-LS or JPEG-2000. Table 1 shows the results using JPEG-LS as JPEG-LS generally provides a better compression performance than JPEG-2000.

Each figure in the table shows the weighted average of a particular testing image group. To compute the weighted average of a group, the bpp value of each compressed testing image in the group is weighted by the ratio of the number of the image's pixels to the total number of pixels of all images in the group, and then all weighted bpp values in the group are summed. The proposed method outperforms the others in all situations.

V. Conclusions

A pixel-wise palette reordering method is proposed in this paper to reshape the statistical properties of the index map of a color-indexed image. This method scans the index map pixel by pixel. At each pixel, it adaptively reorders the colors in the image palette and then replaces the pixel's original index in the index map with the position of the pixel's color in the reordered palette as the new index. Eventually it generates a new index map each element of which serves as an index to a pixel-dependently reordered version of the palette. The resultant index map contains very low zero-order entropy and energy. Besides, most of the spatial correlation among pixels can be removed in the new index map.

A dedicated coding scheme is also proposed to encode the outputs of the proposed pixel-wise palette reordering methods. Both bit-plane coding and entropy coding techniques are exploited to make a good use of the statistical properties of the reindexed index maps. Simulation results reveal that the coding performance of the proposed approach is better than other state-of-art lossless image compression schemes when coding color-indexed images.

Acknowledgement

This work was supported by a grant from the research Grants Council of the Hong Kong Special Administrative Region (PolyU 5123/08E) and Centre for Multimedia Signal Processing, The Hong Kong Polytechnic University.

References

[1] M.T. Orchard and C.A. Bouman, "Color quantization of images," IEEE Trans. on SP, Dec 1991, Vol.39, No.12, pp.2677-2690.

[2] A.J. Pinho and A. J. R. Neves, "A survey on palette reordering methods for improving the compression of color-indexed images," IEEE Trans. IP, Vol.13, No.11, Nov 2004, pp.1411-1418.

[3] A. J. Pinho and A. J. R. Neves, "A note on Zeng's technique for color reindexing of palette-based images," IEEE SPL, vol. 11, Feb. 2004, pp. 232-234.

[4] S. Battiato, G. Gallo, G. Impoco, and F. Stanco, "An efficient Re-indexing algorithm for color-mapped images," IEEE Trans. on IP, Vol.13, Nov. 2004, pp.1419-1423.

[5] N. D. Memon and R. Rodila, "Transcoding GIF images to JPEG-LS," IEEE Trans. on CE, Vol.43, No.3, Aug 1997, pp.423-429.

[6] S.C. Pei, Y.T. Chuang and W.H. Chuang, "Effective palette indexing for image compression using self-organization of Kohonen feature map," IEEE Trans on IP, Vol.15, No.9, Sept. 2006, pp.2493-2498.

[7] S. Battiato, F. Rundo and F. Stanco, "Self organizing motor maps for color-mapped image re-indexing," IEEE Trans. on IP, Vol.16, No.12, Dec 2007, pp.2905-2915.

[8] J. Z. C. Lai and Y. C. Liaw, "A novel approach of reordering color palette for indexed image compression," IEEE SPL, Vol.14, No.2, Feb. 2007, pp.117-120

[9] N. Kuroki, T. Yamane and M. Numa, "Lossless coding of color quantized images based on Pseudo distance," Proceedings, 47th IEEE International Midwest Symposium on Circuits and Systems, 2004, Vol.I, pp.245-247.

[10] Z. Arnavut and F. Sahin, "Lossless compression of color palette images with one-dimensional techniques," Journal of Electronic Imaging, Vol.15, No.2, 023014, Apr-Jun 2006

[11] X. Chen, S. Kwong, and J. Feng, "A New Compression Scheme for Color-Quantized Images", IEEE Trans. on CASVT, Vol. 12, No 10, Oct 2002, pp.904-908.

[12] K. C. Lui and Y.H. Chan, "A lossless compression algorithm for color-indexed images using adaptive palette reordering," Proceedings, EUSIPCO'07, 2007, Poznan, Poland, pp.812-815.

[13] ISO/IEC JTC 1/SC 29/WG 1 JPEG LS image coding system, ISO Working Document ISO/IEC JTC1/SC29/WG1 N399 –WD14495, July 1996.

Image group	No. of colors	PNG	JPEG-LS	JPEG2000	Pinho[3]	Battiato[4]	Arnavut[10]	Battiato[7]	Chen[11]	Kuroki[9]	Ours
(1)	6 - 256	1.926	2.854	3.438	2.333	2.650	1.532	2.306	1.565	2.438	**1.425**
(2)	256	4.604	6.077	6.574	4.475	5.530	3.549	4.440	3.633	4.275	**3.316**
(3)	128	3.742	4.838	5.358	3.552	4.287	2.954	3.526	2.915	3.553	**2.749**
(4)	64	2.952	3.661	4.208	2.709	3.144	2.327	2.710	2.247	2.862	**2.195**
(5)	256	5.136	6.433	6.959	4.881	5.983	4.023	4.779	3.799	4.447	**3.600**
(6)	128	4.017	5.318	5.903	3.979	4.729	3.220	3.785	2.969	3.652	**2.857**
(7)	64	3.190	3.967	4.596	2.912	3.281	2.497	2.921	2.298	2.974	**2.290**

Table 1. Performance of different lossless image compression schemes for coding color-indexed images in terms of bpp