An adaptive palette reordering method for compressing color-indexed image

Ka-Chun Lui and Yuk-Hee Chan

Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong

Abstract - An adaptive palette reordering method is proposed in this paper to reshape the statistical properties of the color index map of a color-indexed image with a dynamic palette. Unlike other reordering methods, the proposed method extracts information from both the color index map and the palette to achieve the objective. The compression performance of JPEG-LS can be significantly improved when the proposed method is used.

I. INTRODUCTION

Color-quantized images [1] are widely used in various applications especially Internet applications nowadays to reduce communication bandwidth and storage requirement. A color-quantized image is generally represented with a color index map each element of which serves as an index to select a color from a predefined set of colors to represent the color of a pixel in the image. The predefined set of colors is called a palette.

To reduce the size of a color-indexed image further, lossless compression techniques are generally used because the index used to pick a particular palette color must be exact in decoding. A minor difference between two index values may result in a serious color shift.

Predictive coding technique is widely used in lossless compression. In fact, most lossless coding algorithms such as CALIC [2] and JPEG-LS [3] are based on predictive coding and entropy coding techniques. CALIC and JPEG-LS raster-scan an image and use the intensity values of some encoded local pixels to predict the intensity value of the pixel being encoded. The estimation error between the actual value and the estimated value is then encoded with entropy coding algorithms to compress the image. In particular, the predictors used in CALIC and JPEG-LS are, respectively, Gradient Adaptive Predictor (GAP) and Median Edge Detection (MED) predictor. Both predictors make use of the high spatial correlation in a natural image. In general, the smaller the average prediction error over the image, the higher compression ratio can be achieved.

Predictive coding techniques do not perform well when they are used to encode a color-quantized image. Encoding a color-quantized image implies encoding its color index map. In general, a palette is generated in a way without concerning the order of the colors in the resultant palette. Accordingly, the numerical values of two indices that point to similar colors may be very different. Most predictors assume that neighboring pixels have similar attributes and, when these attributes are quantitatively measured, their values are similar. This assumption is valid for the intensity values of an image, but not for the index values in a color index map due to the aforementioned reason.

Palette reordering is a remedial process aiming at finding a permutation of the color palette to make the resulting color index map more suitable for predictive coding. In general, palette reordering attempts to minimize the index difference between adjacent pixels such that the prediction error would be as small as possible. Various reordering methods were proposed for this purpose. Some of them assign indices to palette colors based on the attributes of the palette colors [4] or the distance among the palette colors [5]. Some of them assign indices to palette colors based on the number of occurrences of having two particular palette colors in two spatially adjacent pixels[6-9]. All of them can effectively improve the compression rate when their outputs are encoded with JPEG-LS.

However, after studying these algorithms, we have two observations. First, these methods use the information extracted from either the palette or the index map exclusively to assign indices to palette colors. Second, the assignment is based on the global characteristics or statistics of the image and it is non-adaptive.

In this paper, we proposed a new reordering method. This method adaptively reorders the palette based on both the palette and the index map to produce an index map of lower zero-order entropy and higher spatial correlation. The resultant index distribution is very biased. By so doing, the index map can be encoded with JPEG-LS more efficiently as compared with the outputs of other reordering methods.

II. PROPOSED ALGORITHM

Unlike conventional palette reordering methods[4-9] in which the palette is reordered based on some global properties of the input image, the proposed approach reorders the palette adaptively according to the current statistics and local characteristics of the image such that more spatial redundancy can be removed in the produced index map.

The proposed method pre-processes a given color index map to generate an input to a JPEG-LS codec and, at the receiver, post-processes the output from a JPEG-LS decoder to reconstruct the original color-quantized image. It is fully compatible with JPEG-LS in a way that no modification to the JPEG-LS codec is required to compress an image when the proposed reordering method is exploited.

1-4244-0549-1/06/$20.00 ©2006 IEEE.

In the pre-processing stage, the input image is raster scanned and pixels are processed sequentially. For each pixel, its color is predicted with its processed neighboring pixels. The prediction error is recorded for future reference. Based on the prediction errors of all previously processed pixels and the Euclidean distance between the predicted color and each palette color, all palette colors are sorted to form a queue. The position of the real color in the queue is then used as the new index for the pixel in the output. Similar approach is realized in the post-processing stage to generate a transient color queue for each pixel to determine the original pixel color with the received index.

A. Core of the method

Let the input color-indexed image be **X** and the associated palette be $\Omega=\{\vec{c}_i|i=0,1...N-1\}$, where N is the size of the palette. Without lose of generality, we assume all $\vec{c}_i$ in Ω are sorted according to their luminance and $\vec{c}_0$ 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. In the proposed method, the three color planes of the constructed full-color image are separated and each of them is treated as a grey-level image for prediction. The three color planes are raster scanned and processed in parallel synchronously. For each pixel, the intensity values of its three color components are individually predicted with their corresponding color planes by using a MED predictor. Note that some other predictors such as GAP can also be used for the prediction. In fact, our simulation results show that both MED and GAP can provide a good prediction performance in this application. That MED is suggested in the method is because the reordering method is proposed to support JPEG-LS and MED is used in JPEG-LS.

Suppose the prediction result of the three color components of the current pixel is $\vec{v}_p=(r_p,g_p,b_p)$. $\vec{v}_p$ is then quantized with palette Ω. Let the quantization result be $\vec{c}_p$. $\vec{c}_p$ could be different from the real color of the pixel, $\vec{c}_r$. The occurrence of this discrepancy is recorded and cumulated for improving the prediction performance in the future as history is useful in predicting an event. In particular, in the proposed scheme, a table is constructed for storing the values of $\{H(m,n)|m,n=0,1...N-1\}$, where $H(m,n)$ is defined as the number of occurrences when the quantized predicted color and the real color of a pixel are, respectively, $\vec{c}_m$ and $\vec{c}_n$. All $H(m,n)$ values are initialized to zero at the very beginning and the table is updated whenever a pixel is processed. For reference, this table is referred to as *discrepancy frequency table* (DFT) hereafter.

After $\vec{v}_p$ and $\vec{c}_p$ are determined, the colors in palette Ω are adaptively reordered based on $\{H(p,i)|i=0,1...N-1\}$ and $\|\vec{c}_i-\vec{v}_p\|^2$. In particular, $\vec{c}_i$'s are sorted according to the values of $\{H(p,i)|i=0,1...N-1\}$ in descending order. If there exist two different colors $\vec{c}_l$ and $\vec{c}_j$ such that $H(p,l)=H(p,j)$, $\vec{c}_l$ and $\vec{c}_j$ will be sorted according to their Euclidean distances to $\vec{v}_p$. The closer one is 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, $H(p,r)$ is incremented by 1 to update the frequency count of this event.

Figure 1 shows an example of how an index is adaptively determined for a pixel when the current status of $H(m,n)$ is shown in Fig. 1a. In this example, the palette Ω is of size 8. Assume that the quantized predicted color and the real color of the pixel are, respectively, $\vec{c}_4$ and $\vec{c}_3$. $\{\vec{c}_i|i=0,1...7\}$ are then sorted by $H(4,i)$ and then by $\|\vec{c}_i-\vec{v}_4\|^2$. The position of $\vec{c}_3$ in the sorted sequence is 2 and hence the output index for $\vec{c}_3$ is 2.

In the decoder, to decode a pixel, the same process is carried out to determine a 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 static color-index map which uses a fixed palette such as Ω to generate a full-color image, or even (ii) the full-color image directly.

B. DFT Merging

The DFT helps to improve the performance of palette reordering. Based on the values of $\{H(m,n)|n=0,1..N-1\}$, it shows how likely that $\vec{c}_n$ is the real color when $\vec{c}_m$ is the quantized predicted color. This statistical information is useful for the codec to assign indices to palette colors. In our case, the color which is more likely to occur is assigned a smaller index. Eventually, the indices of the output index map are highly biased to zero. This helps to reduce the zero-order entropy of the index map.

At the early stage of palette reordering, most of $H(m,n)$ entries are of zero values. Working with a premature DFT degrades the reordering performance. As it takes time to process sufficient pixels for building up a meaningful DFT, a DFT merging scheme is exploited in the proposed reordering method to solve this problem.

In the proposed method, by making use of LBG algorithm[10], a palette of a size smaller than palette Ω is generated with all colors in Ω as the training vectors. All colors in Ω are then color quantized with this smaller palette. In consequence, all $\vec{c}_i$ in Ω are clustered into a few groups.

Let $\vec{c}_p$ be the quantized predicted color for the pixel being processed and $H_m=\sum_{n=0}^{N-1}H(m,n)$ for $m=0,1...N-1$. When $\vec{c}_p$ is determined, H_p is checked against a predefined threshold value T. If it is smaller than T, which implies insufficient samples were collected for predicting the real color $\vec{c}_r$ based on $\vec{c}_p$, the statistics of all colors in the same group with $\vec{c}_p$ will be merged to determine the new index of $\vec{c}_r$. For instance, if $\vec{c}_p$ and $\vec{c}_k$ belong to the same group, $\vec{c}_i$'s will be sorted according to the values of {H(p,i)+H(k,i) | i=0,1…N-1} in descending order. If there exist two different colors $\vec{c}_l$ and $\vec{c}_j$ such that H(p,l)+H(k,l) = H(p,j)+H(k,j), $\vec{c}_l$ and $\vec{c}_j$ will be sorted according to their Euclidean distance to $\vec{v}_p$.

Let's consider the example shown in Figure 1 again. Assume that $\vec{c}_4$ and $\vec{c}_7$ belong to the same group and H_4 is now smaller than the threshold. After sorting $\{\vec{c}_i \mid i=0,1\ldots 7\}$ by H(4,i)+H(7,i) and then by $\|\vec{c}_i - \vec{v}_4\|^2$, the new queue is $\{\vec{c}_3, \vec{c}_5, \vec{c}_7, \vec{c}_4, \vec{c}_6, \vec{c}_1, \vec{c}_0, \vec{c}_2\}$ and the new index of $\vec{c}_3$ is 0.

A merged DFT can further be merged into an even smaller DFT in the same manner when it is necessary. The codec can use a smaller DFT whenever a large DFT has not yet been mature. As it needs fewer samples to build up a smaller DFT, the proposed palette reordering method can provide a reasonable and steady performance after processing a few samples and its advantage can be seen even at a very early stage of the index reordering process.

C. Preparation for JPEG-LS

With the proposed palette reordering method, the indices of the output index map are highly biased to zero and the zero-order entropy of the output index map is lowered. For example, the zero-order entropy of the index map of a 256-color color-quantized 'Lena' can be reduced from 7.7756 to 4.0217 bits per pixel (bpp) with the proposed reordering method. In contrast, other reordering methods[4-9,11] cannot reduce the zero-order entropy of an index map as they exploit bijective mappings to reindex the palette colors.

However, the index map obtained at this stage is still not good enough for JPEG-LS to encode. To further reduce the index difference between adjacent pixels, the indices of the index map are remapped to other values with a bijective mapping $M(\bullet)$ as follows.

$$M(i) = \begin{cases} (N-2-i)/2 & \text{for } i = 0,2\ldots N-2 \\ (N-1+i)/2 & \text{for } i = 1,3\ldots N-1 \end{cases} \quad \text{if } N \text{ is even}$$

$$M(i) = \begin{cases} (N-1-i)/2 & \text{for } i = 0,2\ldots N-1 \\ (N+i)/2 & \text{for } i = 1,3\ldots N-2 \end{cases} \quad \text{if } N \text{ is odd} \qquad (1)$$

This mapping shifts the bias of the index from 0 to $\lceil N/2 \rceil - 1$.

III. Experimental Results

Simulations were carried out to evaluate the performance of the proposed reordering method. A number of standard full-color testing images including computer graphic images and natural images were color-quantized to 256-color images with MathLab function RGB2IND. No dithering was performed in the quantization. The resultant color quantized images were then processed with different index reordering methods. Finally, the reordered index maps were encoded with JPEG-LS.

Table 1 lists the compression ratios achieved by JPEG-LS when encoding the outputs of various reordering methods [4-9,12]. The proposed method boosts up more compression performance than the others.

Figure 2 shows the histograms of the color index maps obtained with different reordering methods. One can see that our result is very biased while the others are not. This explains why the zero-order entropy of our result is lower than the others.

Table of {H(m,n)}

*Index m	*Index n							
	0	1	2	3	4	5	6	7
0	29	7	6	5	4	3	2	1
1	0	88	1	2	0	0	0	1
2	0	2	65	1	2	1	0	0
3	0	2	10	56	1	1	0	1
4	3	1	0	**8**	8	8	0	0
5	0	0	3	2	3	23	5	1
6	0	0	0	1	1	2	23	2
7	0	2	0	3	0	2	6	9

(a)

*Index i	0	1	2	3	4	5	6	7
Prediction Error	0.6	0.2	0.3	0.3	0.2	0.1	0.2	0.1

(b)

*Index i	0	1	2	3	4	5	6	7
New index	3	4	7	2	1	0	6	5

(c)

*Index i	0	1	2	3	4	5	6	7
**H(4,i)+H(7,i)	3	3	0	11	8	10	6	9
New index	6	5	7	0	3	1	4	2

(d)

*Refer to the reference palette   ** $\vec{c}_4$ and $\vec{c}_7$ are in the same group.

Fig. 1 Example of how to assign indices to a dynamic palette when $\vec{v}_4$, $\vec{c}_4$ and $\vec{c}_3$ are, respectively, the predicted, the quantized predicted and the real colors: (a) current status of the DFT, (b) prediction error $\|\vec{c}_i - \vec{v}_4\|^2$, (c) index assignment without DFT merging, and (d) index assignment with DFT merging.

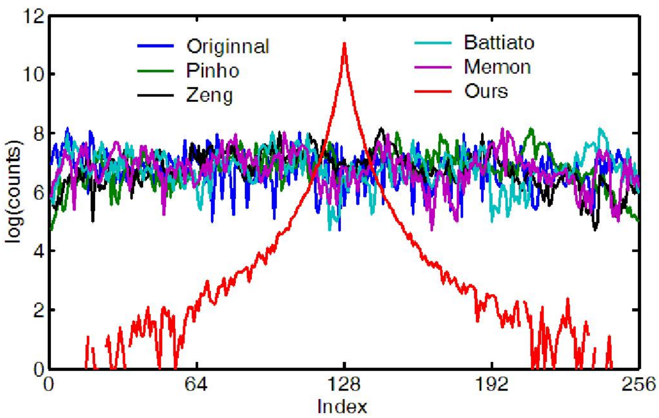

Fig. 2 Histograms of color index maps of *Lena*

IV. Conclusion

An adaptive palette reordering method is proposed in this paper to reshape the statistical properties of a color index map such that, when the index map is encoded with

JPEG-LS, the compression ratio can be significantly improved. Unlike other reordering methods, this method adaptively reorders the palette based on both the palette and the index map to produce a new index map of very low zero-order entropy and high spatial correlation. Simulation results reveal that its performance is better than other JPEG-LS-compatible reordering methods.

ACKNOWLEDGEMENT

This work was supported by a grant from The Hong Kong Polytechnic University (G-YE90).

REFERENCE

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

[2] X. Wu and N. D. Memon, "Context-based adaptive lossless image coding," IEEE Trans. Commun., Vol. 45, Apr. 1997, pp. 437-444.

[3] 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.

[4] A. Zaccarin and B. Liu, "A novel approach for coding color quantized images," IEEE Trans. Image Processing, Vol. 2, Oct. 1993, pp. 442-453.

[5] A. Spira and D. Malah, "Improved lossless compression of color-mapped images by an approximate solution of the traveling salesman problem," Proc. IEEE ICASSP'01, Vol.III, May 2001, pp.1797-1800

[6] N. D. Memon and A. Venkateswaran, "On ordering color maps for lossless predictive coding," IEEE Trans. Image Processing, Vol. 5, Nov. 1996, pp. 1522-1527.

[7] W. Zeng, J. Li, and S. Lei, "An efficient color re-indexing scheme for palette-based compression," in Proc. 7th IEEE Int. Conf. Image Processing Vancouve, BC, Canada, vol. III, Sept. 2000, pp. 476-479.

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

[9] S. Battiato, G. Gallo, G. Impoco and F. Stanco, "An Efficient Re-Indexing Algorithm for Color-Mapped Images," IEEE Trans. Image Processing, Vol.13, Nov 2004, pp.1419-1423

[10] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," IEEE Trans. Commun., Vol. COM-28, Jan. 1980, pp. 84-95.

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

[12] http://www.hpl.hp.com/loco/

Group	Image	Size (pixel²)	Bits per pixel						
			[12]	[4]	[6]	[7]	[8]	[9]	Ours
CG image	pool	510x383	1.563	1.965	1.464	1.560	1.462	1.786	1.402
	watch	1027x768	2.111	2.436	2.093	2.273	2.169	2.632	1.964
	water	1024x768	5.568	5.673	5.487	6.207	5.621	7.089	4.887
Natural image	Kodak 01	768x512	5.489	5.956	5.403	5.969	5.599	7.055	4.926
	Kodak 02	768x512	4.259	5.277	4.234	4.605	4.383	5.418	3.748
	Kodak 03	768x512	2.603	3.587	2.723	2.875	2.742	3.548	2.526
	Kodak 04	768x512	4.002	4.765	3.920	4.953	4.312	5.447	3.468
	Kodak 05	768x512	5.071	5.669	5.029	5.390	5.204	6.408	4.485
	Kodak 06	768x512	4.758	4.997	4.976	5.413	5.153	5.921	4.247
	Kodak 07	768x512	3.525	4.208	3.667	4.069	3.857	4.698	3.082
	Kodak 08	768x512	5.517	5.554	5.797	5.965	5.852	6.404	4.603
	Kodak 09	768x512	3.873	4.423	3.952	4.514	4.100	5.506	3.497
	Kodak 10	768x512	4.683	4.551	4.505	5.339	4.826	5.969	3.668
	Kodak 11	768x512	4.461	5.024	4.334	4.890	4.458	4.799	3.787
	Kodak 12	768x512	3.739	4.316	3.868	4.403	3.960	5.329	3.318
	Kodak 13	768x512	6.009	6.482	5.968	6.509	6.203	7.016	5.608
	Kodak 14	768x512	4.468	5.369	4.497	5.140	4.717	6.482	4.106
	Kodak 15	768x512	3.397	4.089	3.537	3.883	3.636	4.332	3.060
	Kodak 16	768x512	4.104	4.558	4.254	4.817	4.386	5.995	3.661
	Kodak 17	768x512	4.654	4.510	4.335	5.526	4.818	5.763	3.733
	Kodak 18	768x512	5.398	5.906	5.276	5.777	5.220	6.762	4.583
	Kodak 19	768x512	4.565	5.018	4.656	5.104	4.845	5.987	3.988
	Kodak 20	768x512	3.085	3.201	3.144	3.326	3.053	3.690	2.912
	Kodak 21	768x512	4.509	5.231	4.422	4.916	4.544	5.563	4.012
	Kodak 22	768x512	5.043	5.242	4.990	5.708	5.119	5.969	4.095
	Kodak 23	768x512	2.926	3.489	3.091	3.252	2.890	3.297	2.448
		Average	**4.207**	**4.673**	**4.216**	**4.707**	**4.351**	**5.341**	**3.685**

Table 1. Performance of different palette reordering methods when working with JPEG-LS