A LOSSLESS COMPRESSION ALGORITHM FOR COLOR-INDEXED IMAGES USING ADAPTIVE PALETTE REORDERING

Ka-Chun Lui and Yuk-Hee Chan

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

ABSTRACT

In this paper, an adaptive palette reordering method is proposed to reshape the statistical properties of the color index map of a color-indexed image with a dynamic palette. The proposed method extracts information from both the palette and the color index map to achieve the objective. It reduces the zero-order entropy of the image and removes the spatial correlation among pixels significantly. The compression performance of JPEG-LS can be significantly improved when the proposed method is used. An even better compression performance can be achieved when being compatible with JPEG-LS is not the concern.

1. 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.

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[2]. Various reordering methods were proposed for this purpose. Some of them assign indices to palette colors based on the attributes of the palette colors [3] or the distance among the palette colors [4]. 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[5-10]. All of them can effectively improve the compression rate when their outputs are encoded with JPEG-LS[11].

In general, these methods assign indices to palette colors with the information extracted from either the palette or the index map exclusively, and the assignment is based on the global characteristics or statistics of the image non-adaptively. In this paper, we proposed a new reordering method. Unlike those aforementioned reordering methods, 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 little spatial correlation. By so doing, the index map can be encoded more efficiently as compared with the outputs of other reordering methods[2-10].

2. PROPOSED METHOD

The proposed method can be used to pre-process a given color index map to generate an input to a JPEG-LS codec and, at the receiver, post-process 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.

2.1. Core of the method

Let the input color-indexed image be **X** and the associated palette be $\Omega=\{\vec{c}_i|i=0,1\ldots N-1\}$, 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. 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. The MED predictor is used because the reordering method is proposed to support JPEG-LS and MED is used in JPEG-LS[11].

Suppose the prediction result of the three color components of the current pixel is $\vec{v}=(r',g',b')$. $\vec{v}$ is then quantized with palette Ω. Let the quantization result be $\vec{c}_p$, the p^{th} palette color in Ω. $\vec{c}_p$ could be different from the real color of the pixel, $\vec{c}_r$, which is assumed to be the r^{th} palette color in Ω without losing the generality. In the proposed scheme, the occurrence of this discrepancy is recorded and cumulated for improving the prediction performance in the future. In particular, a table is constructed for storing the values of $\{H(m,n)|m,n=0,1\ldots N-1\}$, 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 after a pixel is processed. For reference, this table is referred to as *discrepancy frequency table* (DF-Table) hereafter.

After $\vec{v}$ and $\vec{c}_p$ are determined, the colors in palette Ω are adaptively reordered based on $\{H(p,i)|i=0,1\ldots N-1\}$ and

$\| \vec{c}_i - \vec{v} \|^2$. 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 $\vec{v}$. 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, H(p,r) is incremented by 1 to update the frequency count of this event.

For each pixel, 3 MED prediction processes, an N-codeword VQ process and a sorting process are required. In practice, the sorting effort can be neglected as a sequence of $\vec{c}_i$'s which are sorted by H(p,i) can be easily updated when H(p,r) is updated after processing a pixel. As all we need is the position of $\vec{c}_r$ in the queue, in most cases we do not need to sort $\vec{c}_i$'s by $\| \vec{c}_i - \vec{v} \|^2$.

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 predicted color, the quantized predicted color and the real color of the pixel are, respectively, $\vec{v}$, $\vec{c}_4$ and $\vec{c}_3$. In such a case, {$\vec{c}_i$|i=0,1…7} are sorted by H(4,i) and then by $\| \vec{c}_i - \vec{v} \|^2$. It results in $\{\vec{c}_5, \vec{c}_4, \vec{c}_3, \vec{c}_0, \vec{c}_1, \vec{c}_7, \vec{c}_6, \vec{c}_2\}$. 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 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 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.

2.2. DF-Table Merging

The DF-Table 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.

At the early stage of palette reordering, most of H(m,n) entries are of zero values. Such a premature DF-Table provides no or little statistical information and hence does not contribute to the reordering performance. A DF-Table merging scheme is exploited in the proposed reordering method to solve this problem.

In the proposed method, by making use of LBG algorithm[12], 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.

Without loss of generality, let us assume $\vec{c}_p$ and $\vec{c}_k$ belong to the same group. When the quantized predicted color $\vec{c}_p$ is determined, $H_p = \sum_{n=0}^{N-1} \mathrm{H}(p,n)$ 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 this instance, as $\vec{c}_p$ and $\vec{c}_k$ belong to the same group, {$\vec{c}_i$|i=0,1…7} are 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 $\vec{v}$. If they are still not distinguishable, their order will be determined by their ranking in reference palette Ω.

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 {$\vec{c}_i$|i=0,1…7} by H(4,i)+H(7,i) and then by $\| \vec{c}_i - \vec{v} \|^2$, 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.

Table of {H(m,n)}

$\vec{c}_i$	$\vec{c}_0$	$\vec{c}_1$	$\vec{c}_2$	$\vec{c}_3$	$\vec{c}_4$	$\vec{c}_5$	$\vec{c}_6$	$\vec{c}_7$
i	0	1	2	3	4	5	6	7
H(0,i)	29	7	6	5	4	3	2	1
H(1,i)	0	88	1	2	0	0	0	1
H(2,i)	0	2	65	1	2	1	0	0
H(3,i)	0	2	10	56	1	1	0	1
H(4,i)	3	1	0	**8**	8	8	0	0
H(5,i)	0	0	3	2	3	23	5	1
H(6,i)	0	0	0	1	1	2	23	2
H(7,i)	0	2	0	3	0	2	6	9

(a)

$\vec{c}_i$	$\vec{c}_0$	$\vec{c}_1$	$\vec{c}_2$	$\vec{c}_3$	$\vec{c}_4$	$\vec{c}_5$	$\vec{c}_6$	$\vec{c}_7$
Prediction Error	0.6	0.2	0.3	0.3	0.2	0.1	0.2	0.1

(b)

$\vec{c}_i$	$\vec{c}_0$	$\vec{c}_1$	$\vec{c}_2$	$\vec{c}_3$	$\vec{c}_4$	$\vec{c}_5$	$\vec{c}_6$	$\vec{c}_7$
New index	3	4	7	2	1	0	6	5

(c)

$\vec{c}_i$	$\vec{c}_0$	$\vec{c}_1$	$\vec{c}_2$	$\vec{c}_3$	$\vec{c}_4$	$\vec{c}_5$	$\vec{c}_6$	$\vec{c}_7$
*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)

*Assume that $\vec{c}_4$ and $\vec{c}_7$ are in the same group.

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

A merged DF-Table can further be merged into an even smaller DF-Table in the same manner when it is necessary. The codec can use a smaller DF-Table whenever a large DF-

Table has not yet been mature. As it needs fewer samples to build up a smaller DF-Table, 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.

3. SIMULATION RESULTS

Simulations were carried out to evaluate the performance of the proposed reordering method. A set of 24 standard full-color testing images of size 768x512 (the Kodak image set) were color-quantized to 256-color images with MATLAB 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.

DF-Table merging was on when realizing our method. Starting from 256, the number of groups into which all quantized predicted colors were clustered was halved progressively until either it reached 8 or $\sum_{\vec{c}_l \in \Phi_p} \sum_{n=0}^{N-1} \mathrm{H}(l,n) > T$, where $\Phi_p \subset \Omega$ is the group to which $\vec{c}_p$ belongs, was satisfied. In particular, the threshold T was selected to be 0.1N.

Before encoding the index map obtained with our method with JPEG-LS, the indices of the index map are remapped to other values with a bijective mapping $M(\bullet)$ as follows.

$$M(i) = \lceil N/2 \rceil - 1 - (-1)^i \lceil i/2 \rceil \quad \text{for } i = 0,1 \ldots N\text{-}1 \quad (1)$$

This mapping shifts the peak of the index distribution from 0 to $\lceil N/2 \rceil$-1, which improves the efficiency of JPEG-LS.

Table 1 lists the compression performance achieved by JPEG-LS when encoding the outputs of various reordering methods[3-10]. The proposed method improves the compression performance more significantly than the others.

Figure 2 shows the histograms of the color index maps obtained with different reordering methods, and Figure 3 shows some reindexed index maps for visual inspection. One can see that the peak of the histogram of our reindexed output is very sharp while those of the others are not. This explains why the zero-order entropy of our result is significantly lower than the others. Specifically, the proposed reordering method reduces the zero-order entropy of the original 256-color index map of *Lena* from 7.7756 to 4.0217 bits per pixel (bpp). Other reordering methods[2-8] exploit bijective mappings to reindex the palette colors and hence their zero-order entropy will remain the same after mapping.

In data compression, effectiveness in removing redundancy usually implies compression performance. Figure 4 shows the correlation coefficient of the index values of two pixels which are (x,y) apart in the output of a particular reordering method. It was evaluated with the results shown in Figure 3. One can see that the remapped indices in the output of our approach are highly uncorrelated after mapping. This implies that the proposed method can remove most spatial correlation in the reordered index map while the others cannot. The proposed method is much more effective than the others in eliminating the statistical redundancy of indices.

Since most of the correlation is removed and the zero-order entropy of the indices are significantly reduced in our reordering result, our reindexed output can also be compressed remarkably with some other lossless compression algorithms. As a matter of fact, if being compatible with JPEG-LS is not the concern, an even higher compression ratio can be achieved with some other means.

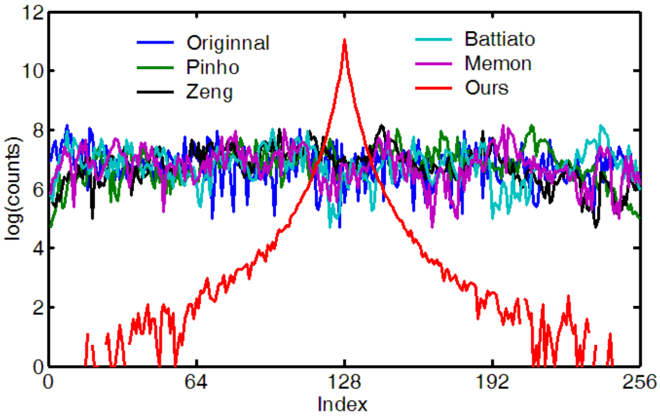

Fig. 2 Histograms of color index maps of *Lena*

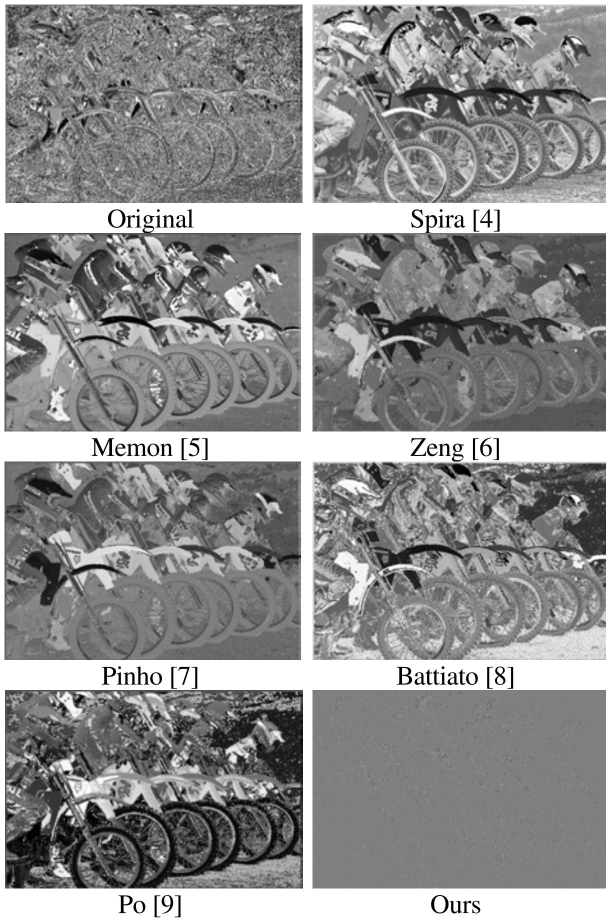

Fig. 3 Reindexed *Cycle*

In particular, when the reindexed output of the proposed reordering method (the one without being further mapped with mapping (1) for JPEG-LS) are separated into $\lceil \log_2 N \rceil$ bit planes (Each of which carries the j^{th} most significant bits of all indices in the reindexed output for j=1,2… $\lceil \log_2 N \rceil$.) and each of the bit planes is encoded with JBIG [18] individually, an average compression performance of 3.702 bits per pixel can be achieved. For comparison, Table 2 shows the compression performance of some other lossless compression algorithms [13-17] for coding color-indexed images. The column of Ours-J shows the case when the bit planes are encoded with JBIG [18].

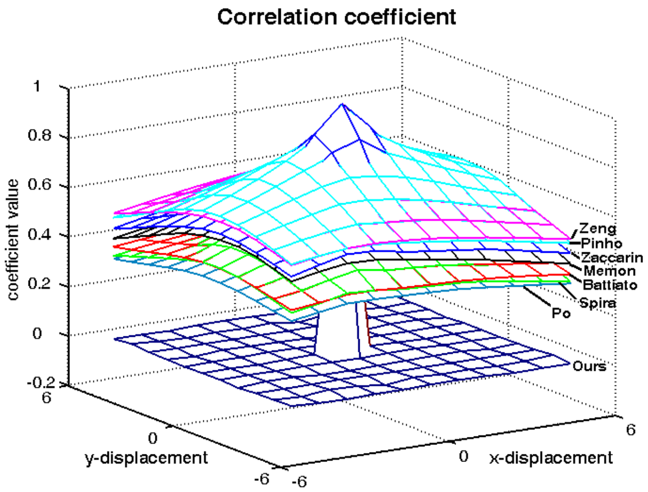

Fig. 4 Performance of different algorithms in terms of correlation among pixels in their reindexed results

4. CONCLUSIONS

An adaptive palette reordering method is proposed in this paper to reshape the statistical properties of a color index map. This method adaptively reorders the palette based on both the palette and the index map to produce a new index map. This method can remove most of the spatial correlation among pixels and reduce the zero-order entropy of the reindexed index map significantly. Simulation results reveal that its performance is better than other JPEG-LS-compatible reordering methods. When being compatible with JPEG-LS is not the concern, an even better compression performance can be achieved.

5. ACKNOWLEDGEMENT

This work was supported by the RGC of Hong Kong Special Administrative Region (PolyU5205/04E) and The Hong Kong Polytechnic University (PolyU Grant G-YE90).

REFERENCES

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

[2] 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.

[3] Zaccarin and B. Liu, “A novel approach for coding color quantized images,” IEEE Trans. on Image Processing, vol. 2, Oct. 1993, pp. 442-453.

[4] 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.

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

[6] W. Zeng, J. Li, and S. Lei, “An efficient color re-indexing scheme for palette-based compression,” Proc. IEEE ICIP’00, Vol. III, Sept. 2000, pp. 476-479.

[7] 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.

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

[9] L.M.Po and W.T.Tan, "Block address predictive colour quantisation image compression," IEE Electronics Letters, vol.30, no.2, Jan 1994, pp.120-121.

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

[11] 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.

[12] Y. Linde, A. Buzo, and R. M. Gray, “An algorithm for vector quantizer design,” IEEE Trans. on Commun., vol. COM-28, Jan. 1980, pp. 84-95.

[13] F. G. B. De Natale, G. S. Desoli, D. D. Giusto, and G. Vernazza, “A framework for high-compression coding of color images,” Proc. SPIE, vol. 1199, Nov. 1989, pp. 1430-1439.

[14] V. Ratnakar, "RAPP: lossless image compression with runs of adaptive pixel patterns", Conference Record of the 32nd Asilomar Conference on Signals, Systems & Computers, 1-4 Nov. 1998, Vol 2, pp.1251-1255.

[15] 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.

[16] Armando J. Pinho, and Antonio J.R. Neves, "A Context Adaptation Model for the Compression of Images with a Reduced Number of Colors,” Proceedings, IEEE ICIP’05, 11-14 Sept. 2005, Vol.2, pp.738-741.

[17] John A. Robinson, "Adaptive Prediction Trees for Image Compression", IEEE Trans. on Image Processing, Vol 15, No 8, Aug 2006, pp.2131-2145.

[18] http://www.jpeg.org/jbig/index.html

Method	Zaccarin [3]	Spira [4]	Memon [5]	Zeng [6]	Pinho [7]	Battiato [8]	Po [9]	Memon [10]	Ours
Bits per pixel	4.854	5.085	4.392	4.905	4.537	5.555	5.340	4.414	3.799

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

Method	GIF	PNG	[13]*	[14]	[15]	[16]	[17]	Ours-J
Bits per pixel	5.156	4.488	4.612	4.242	4.018	3.955	4.691	3.702

* Sub-codebook size = 8

Table 2. Performance of various lossless image coding algorithms for coding color-indexed images.