Lossless compression of color-quantized images using block-based palette reordering

António J. R. Neves and Armando J. Pinho*

Dept. de Electrónica e Telecomunicações / IEETA
Universidade de Aveiro, 3810–193 Aveiro, Portugal
an@ieeta.pt, ap@det.ua.pt

Abstract. It is well-known that the lossless compression of color-indexed images can be improved if a suitable reordering of the palette is performed before encoding the images. In this paper, we show that, if this reordering is made in a block basis, then further compression gains can be attained. Moreover, we show that the use of block-based palette reordering can outperform a previously proposed block packing procedure. Experimental results using a JPEG-LS encoder are presented, showing how different reordering methods behave.

Keywords: Image compression, reordering techniques, lossless image coding

1 Introduction

Color-indexed images are represented by a matrix of indexes (the index image) and by a color-map or palette. For a particular image, the mapping between index values and colors is not unique — it can be arbitrarily permuted, as long as the corresponding index image is changed accordingly. Although equivalent in terms of representation, for most continuous-tone image coding techniques, such as JPEG-LS [1, 2] or lossless JPEG 2000 [3,4], different mappings may imply dramatic variations in the compression performance.

With the aim of minimizing this drawback, several techniques have been proposed. Basically, they rely on finding an appropriate reordering of the color-map, such that the corresponding image of indexes becomes more amenable to compression. Regrettably, the problem of finding the optimal mapping seems to be computationally intractable [5]. This motivated several sub-optimal, lower complexity, proposals, such as those reported in [5–10] (for a survey, that includes complexity comparison, see [11]).

In this paper, we show that the performance of palette reordering methods can be improved if used in a block basis. Moreover, we show that, in some cases, the use of block-based palette reordering can outperform the block packing procedure described in [12, 13]. To perform this evaluation, we used the JPEG-LS standard.

The remainder of this paper is organized as follows. In Section 2, we provide a brief presentation of the reordering methods that will be used. In Section 3 we describe the proposed method. In Section 4 we provide experimental results showing how the proposed method performs. Finally, in Section 5 we draw some conclusions.

* This work was supported in part by the Fundação para a Ciência e a Tecnologia (FCT).

2 Palette reordering

2.1 Luminance

The use of the luminance information conducts to a simple palette reordering method that is based only on the color palette. This method was proposed by Zaccarin *et al.* [6] in the context of lossy compression. It relies on the assumption that, generally, a given pixel has neighbors of similar luminance and, therefore, colors with similar luminance should have similar indexes. Reordering is performed by sorting the colors according to its luminance, computed according to

$$Y = 0.299R + 0.587G + 0.114B,$$

where R, G and B denote the intensities of the red, green and blue components, respectively.

2.2 The modified Zeng's method (mZeng)

Zeng *et al.* proposed an one-step look-ahead greedy approach for palette reordering, aiming at increasing the lossless compression efficiency of color-indexed images [7]. This algorithm starts by finding the index that is most frequently located adjacent to other indexes, and the index that is most frequently found adjacent to it. This pair of indexes is the starting base for an index list P_n that is constructed, one index at a time, during the operation of the reordering algorithm. If we denote by l_j the indexes already assigned to the index list and by u_i those still unassigned, then, just before starting the iterations, $P_2 = (l_1, l_2)$, where l_1 and l_2 form the pair of indexes mentioned above. New indexes can only be attached to the left or to the right extremity of the list.

We denote by u_L the index that satisfies

$$u_L = \arg\max_{u_i} D_L(u_i, N), \quad \text{with} \quad D_L(u_i, N) = \sum_{j=1}^{N} w_j \, C(u_i, l_j) \tag{1}$$

and by u_R the index satisfying

$$u_R = \arg\max_{u_i} D_R(u_i, N), \quad \text{with} \quad D_R(u_i, N) = \sum_{j=1}^{N} w_{N-j} \, C(u_i, l_j). \tag{2}$$

The function $C(i, j) = C(j, i)$ denotes the number of occurrences corresponding to pixels with index i that are spatially adjacent to pixels with index j. The weights w_j control the impact of the $C(i, j)$ on $D_L(u_i, N)$ and $D_R(u_i, N)$, and the summations are performed over all the N indexes already located in the index list $P_N = (l_1, l_2, \ldots, l_N)$.

The new index list will be given by $(u_L, l_1, l_2, \ldots, l_N)$ if $D_L(u_L, N) > D_R(u_R, N)$ or by $(l_1, l_2, \ldots, l_N, u_R)$ otherwise. Finally, the indexes in the index list are relabeled, creating $P_{N+1} = (l_1, l_2, \ldots, l_{N+1})$. This iterative process continues until all indexes are assigned to the index list. Then, the reordered image is constructed by applying the mapping $l_j \mapsto j - 1$ to all image pixels, and changing the color-map accordingly.

Zeng *et al.* [7] suggested that a reasonable choice for the weights w_j's is given by $w_j = \log_2(1+1/j)$, where j corresponds to the distance between the current left position of the index list and the position of index l_j in the index list.

A theoretical analysis of Zeng's method for the case of Laplacian distributed differences of neighboring pixels was presented in [10], leading to a set of parameters differing from that suggested in [7]. It was found that, under that Laplacian model, the process of building P_{N+1} from P_N has to be conducted in two steps. First, the index u satisfying (1) (or, equivalently, (2)) is determined, where the weights w_j (or w_{N-j}) are all equal (they can be set to one, for example). Then, the correct side of P_N to which the new index u should be attached is determined based on the sign of

$$\Delta = \sum_{j=1}^{N} (N+1-2j)C(u,l_j),$$

where if $\Delta > 0$ then the left side of P_N should be chosen, otherwise choose the right side.

2.3 Memon's method

Memon *et al.* formulated the problem of palette reordering within the framework of linear predictive coding [5], with the objective of minimizing the zero-order entropy of the prediction residuals. They noticed that, for image data, the prediction residuals are often well modeled by a Laplacian distribution and that, in this case, minimizing the absolute sum of the prediction residuals leads to the minimization of the zero-order entropy of those residuals. For the case of a first-order prediction scheme, the absolute sum of the prediction residuals reduces to

$$\sum_{i=1}^{M} \sum_{j=1}^{M} C(i,j)|i-j|, \tag{3}$$

where, in this case, $C(i,j)$ denotes the number of times index i is used as the predicted value for a pixel whose color is indexed by j.

The problem of finding a palette reordering that minimizes (3) can be formulated as the optimization version of the linear ordering problem (also known as the minimum linear arrangement), whose decision version is known to be NP-complete [5]. In fact, if we consider a complete non-directed weighted graph $G(V,E,w)$, where each vertex in $V = \{v_1, v_2, \ldots, v_{|V|}\}$ ($|V| = M$) corresponds to a palette color, and $w(v_i,v_j) = C(i,j) + C(j,i), (v_i,v_j) \in E$ corresponds to the weight associated to the edge defined between vertices v_i and v_j, then the goal is to find a permutation $\sigma : V \rightarrow \{1,2,\ldots,|V|\}$ satisfying

$$\sigma = \arg\min_{\sigma_n} \sum_{(v_i,v_j)\in E} w(v_i,v_j)|\sigma_n(v_i) - \sigma_n(v_j)|. \tag{4}$$

For finding approximate solutions to this problem, Memon *et al.* proposed a heuristic, called pairwise merge, which is based on repeatedly merging ordered sets of colors until obtaining a single reordered set. Initially, each color is assigned to a different set. Then, each iteration consists of two steps. First, the two sets A and B maximizing

$$\sum_{v_i \in A} \sum_{v_j \in B} w(v_i, v_j)$$

among all possible pairs of ordered sets are chosen. Then, a number of merging combinations of the sets A and B are tested (ideally, all possible combinations), and is chosen the one minimizing

$$\sum_{i=1}^{k} \sum_{j=1}^{k} w(u_i, u_j)|i - j|,$$

where $u_1, u_2, \ldots, u_k$ is the combined ordered set under evaluation. To alleviate the computational burden involved in selecting the best way of merging the two ordered sets, Memon *et al.* proposed to use a reduced number of configurations [5]. If $a_1, a_2, \ldots, a_r$ and $b_1, b_2, \ldots, b_s$ are the two ordered sets under evaluation, and if $r, s > 1$, then the following configurations are considered:

$$\begin{array}{c} a_1, a_2, \ldots, a_r, b_1, b_2, \ldots, b_s \\ a_r, \ldots, a_2, a_1, b_1, b_2, \ldots, b_s \\ b_1, b_2, \ldots, b_s, a_1, a_2, \ldots, a_r \\ b_1, b_2, \ldots, b_s, a_r, \ldots, a_2, a_1 \end{array}$$

Alternatively, if one of the sets has size one, then the following configurations are tested (without loss of generality, we consider $r = 1$):

$$\begin{array}{c} a_1, b_1, b_2, \ldots, b_s \\ b_1, a_1, b_2, \ldots, b_s \\ b_1, b_2, a_1, \ldots, b_s \\ \vdots \\ b_1, b_2, \ldots, b_s, a_1 \end{array}$$

3 Block-based approach

Previous work [12, 13] has shown that the performance of global palette reordering techniques can be improved if, in addition to reordering, we consider techniques that operate locally, in particular, block-based histogram packing methods.

Histogram packing can be seen as a preprocessing method that constructs a particular mapping of the image intensity values, where the order is preserved [14]. On the other hand, palette reordering also tries to find a mapping of the image intensity values but, in this case, more generic, and where the order is not necessarily preserved. This difference has a great impact on the side information necessary to reconstruct the original images. For histogram packing, only an indication of the existence of the intensities is needed (one bit per intensity/color suffices), whereas for the case of palette reordering the whole permutation has to be stored ($\log_2 M$ bits per intensity/color, for a M-color image).

Block-based palette reordering consists on applying reordering to each block of a given image partition. The basic idea is to address the palette reordering problem from a local point of view, i.e., on image regions, instead of addressing it globally.

4 Experimental results

The experimental results that we present in this paper are based on a set of 23 true color images (768 columns $\times$ 512 rows) that we refer to as the "Kodak" images[1].

Using version 1.2.5 of the "`gimp`" program,[2] each image was color-quantized based on an image-dependent palette of 256 colors (generated by "`gimp`"). After color quantization, the index images have been reordered using the methods described is this paper, in a block basis. Results using global reordering and block-based histogram packing are presented for comparison.

The compression results that are presented include, besides the size of the encoded index image, the (uncompressed) size of the color table (6 144 bits) and, for the block-based palette reordering approach, the overhead needed for storing the (uncompressed) permutation of the colors for each block (2 048 bits per block, a total of 49 152 bits for 128×128 blocks). The results referring to the block-based histogram packing procedure include the overhead needed for storing the (uncompressed) list of intensities that occur inside each image block (256 bits for each block).

Results using reordering by luminance are provided in Table 1. We note that, for this reordering method, if the palette is globally reordered, then it is also locally reordered, independently of the size and shape of the region. Therefore, it makes no sense to use block-based palette reordering by luminance, since only overhead is added without gaining in compression performance. However, the same does not apply to reordering methods that are based on statistics of the index images, such as Memon's and mZeng's methods, as the results in Tables 2 and 3 demonstrate. In fact, as can be observed, an increase in compression efficiency was obtained for all images, in comparison with both the globally palette reordered approach and the block-based histogram packing approach. The columns under the label "Block Reordered (Best)" in Tables 2 and 3 contain the compression values corresponding to the size of the blocks that provide the highest compression for each image.

5 Conclusions

Color palette reordering is a very effective approach for improving the compression of color-indexed images. In this paper, we provided experimental results showing that the lossless compression of color-indexed images can be further improved if palette reordering is applied in a block basis. With this new approach, additional lossless compression improvements of around 9% for the modified Zeng's technique and of 7% using Memon's technique have been attained, in relation to global palette reordering. In absolute terms, we also conclude that Memon's technique provides the best results. Nevertheless, we should have into account that the computational time necessary to perform the reordering task is proportional to the number of image blocks, which might put the other approaches into an overall competitive position.

---

[1] These images can be obtained from `http://www.cipr.rpi.edu/resource/stills/kodak.html`.

[2] `http://www.gimp.org`.

Table 1. Compression results, in bits per pixel (bpp), of the color-indexed "Kodak" images. The lossless compression of the index images was performed using JPEG-LS, after reordering the index images by luminance. "Global" indicates global palette reordering, whereas "Block Packing" indicates histogram block packing after global palette reordering by luminance.

Image	Global	Block Packing	
		bpp	%
01	5.973	5.286	11.5
02	5.762	5.384	6.5
03	3.633	2.434	33.0
04	4.771	3.723	21.9
05	5.667	4.612	18.6
06	4.956	4.268	13.8
07	4.202	3.251	22.6
08	5.566	4.869	12.5
09	4.510	3.650	19.0
10	4.652	3.805	18.2
11	5.119	4.116	19.5
12	4.327	3.295	23.8
13	6.452	5.766	10.6
14	5.347	4.219	21.0
15	4.058	3.255	19.7
16	4.669	3.910	16.2
17	4.550	3.951	13.1
18	5.899	4.917	16.6
19	5.080	4.101	19.2
20	3.371	3.192	5.3
21	5.257	4.330	17.6
22	5.223	4.357	16.5
23	3.599	2.631	26.8
Average	**4.898**	**4.058**	**17.1**

References

1. ISO/IEC 14495–1 and ITU Recommendation T.87: Information technology - Lossless and near-lossless compression of continuous-tone still images. (1999)
2. Weinberger, M.J., Seroussi, G., Sapiro, G.: The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. IEEE Trans. on Image Processing **9** (2000) 1309–1324
3. ISO/IEC International Standard 15444–1, ITU-T Recommendation T.800: Information technology - JPEG 2000 image coding system. (2000)
4. Skodras, A., Christopoulos, C., Ebrahimi, T.: The JPEG 2000 still image compression standard. IEEE Signal Processing Magazine **18** (2001) 36–58
5. Memon, N.D., Venkateswaran, A.: On ordering color maps for lossless predictive coding. IEEE Trans. on Image Processing **5** (1996) 1522–1527
6. Zaccarin, A., Liu, B.: A novel approach for coding color quantized images. IEEE Trans. on Image Processing **2** (1993) 442–453

Table 2. Compression results, in bits per pixel (bpp), of the color-indexed "Kodak" images. The lossless compression of the index images was performed using JPEG-LS after palette reordering using mZeng's technique. "Global" indicates global palette reordering. "Block Packing" refers to block histogram packing after global palette reordering using mZeng's method, whereas "Block Reordering" and "Block Reordering (Best)" indicate, respectively, the results of applying the proposed approach on square blocks of 128×128 pixels and of using square blocks with size indicated in column "bSize".

		Block Packing		Block Reordering		Block Reordering (Best)		
Image	Global	bpp	%	bpp	%	bpp	%	bSize
01	5.506	5.500	0.1	5.388	2.1	5.353	2.8	224
02	4.883	4.892	-0.1	4.786	1.9	4.786	2.0	128
03	2.618	2.586	1.2	2.283	12.7	2.283	12.8	128
04	4.175	3.879	7.0	3.641	12.8	3.632	13.0	96
05	4.956	4.841	2.3	4.699	5.1	4.694	5.3	112
06	4.824	4.676	3.0	4.341	9.9	4.296	10.9	160
07	3.493	3.475	0.4	3.323	4.8	3.270	6.4	192
08	5.659	5.407	4.4	5.163	8.7	5.101	9.9	176
09	3.915	3.894	0.5	3.719	5.0	3.645	6.9	176
10	4.703	4.262	9.3	3.969	15.6	3.954	15.9	112
11	4.479	4.358	2.6	4.110	8.2	4.083	8.8	144
12	3.847	3.639	5.4	3.269	15.0	3.262	15.2	112
13	6.318	6.079	3.7	5.648	10.5	5.648	10.6	128
14	4.586	4.492	2.0	4.206	8.2	4.206	8.3	128
15	3.538	3.420	3.3	3.128	11.5	3.107	12.2	144
16	4.567	4.385	3.9	4.021	11.9	4.021	11.9	128
17	4.668	4.535	2.8	4.261	8.7	4.250	9.0	112
18	5.228	5.044	3.5	4.764	8.8	4.764	8.7	128
19	4.819	4.620	4.1	4.164	13.5	4.164	13.6	128
20	3.436	3.426	0.2	3.178	7.4	3.167	7.8	192
21	4.684	4.610	1.5	4.289	8.4	4.269	8.9	112
22	4.791	4.524	5.5	4.393	8.3	4.365	8.9	112
23	2.991	2.767	7.5	2.565	14.2	2.560	14.4	144
Average	**4.465**	**4.318**	**3.2**	**4.057**	**9.1**	**4.038**	**9.5**	**–**

7. Zeng, W., Li, J., Lei, S.: An efficient color re-indexing scheme for palette-based compression. In: Proc. of the 7th IEEE Int. Conf. on Image Processing, ICIP-2000. Volume III., Vancouver, Canada (2000) 476–479
8. Battiato, S., Gallo, G., Impoco, G., Stanco, F.: A color reindexing algorithm for lossless compression of digital images. In: Proc. of the IEEE Spring Conf. on Computer Graphics, Budmerice, Slovakia (2001) 104–108
9. Spira, A., Malah, D.: Improved lossless compression of color-mapped images by an approximate solution of the traveling salesman problem. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP-2001. Volume III., Salt Lake City, UT (2001) 1797–1800
10. Pinho, A.J., Neves, A.J.R.: A note on Zeng's technique for color reindexing of palette-based images. IEEE Signal Processing Letters **11** (2004) 232–234
11. Pinho, A.J., Neves, A.J.R.: A survey on palette reordering methods for improving the compression of color-indexed images. IEEE Trans. on Image Processing (2004 (in press))

Table 3. Compression results, in bits per pixel (bpp), of the color-indexed "Kodak" images. The lossless compression of the index images was performed using JPEG-LS after palette reordering using Memon's technique. "Global" indicates global palette reordering. "Block Packing" refers to block histogram packing after global palette reordering using Memon's method, whereas "Block Reordering" and "Block Reordering (Best)" indicate, respectively, the results of applying the proposed approach on square blocks of 128×128 pixels and of using square blocks with size indicated in column "bSize".

		Block Packing		Block Reordering		Block Reordering (Best)		
Image	Global	bpp	%	bpp	%	bpp	%	bSize
01	5.368	5.412	-0.8	5.170	3.6	5.153	4.0	144
02	4.737	4.781	-0.9	4.666	1.5	4.634	2.2	160
03	2.433	2.445	-0.4	2.166	10.9	2.149	11.7	224
04	3.812	3.632	4.7	3.419	10.2	3.410	10.5	112
05	4.861	4.705	3.2	4.530	6.8	4.530	6.8	128
06	4.659	4.395	5.6	4.220	9.4	4.161	10.7	160
07	3.266	3.261	0.1	3.103	4.9	3.072	5.9	176
08	5.278	4.981	5.6	4.979	5.6	4.955	6.1	176
09	3.747	3.810	-1.6	3.606	3.7	3.571	4.7	320
10	4.387	4.076	7.0	3.764	14.1	3.764	14.2	128
11	4.206	4.148	1.3	3.960	5.8	3.960	5.8	128
12	3.629	3.444	5.0	3.144	13.3	3.144	13.4	128
13	6.013	5.782	3.8	5.569	7.3	5.569	7.4	128
14	4.177	4.136	1.0	4.032	3.4	4.013	3.9	176
15	3.265	3.177	2.7	2.983	8.6	2.960	9.4	176
16	4.148	4.022	3.0	3.847	7.2	3.847	7.3	128
17	4.128	4.088	0.9	3.989	3.3	3.976	3.7	240
18	5.046	4.893	3.0	4.522	10.3	4.522	10.4	128
19	4.271	4.245	0.6	3.974	6.9	3.973	7.0	144
20	3.131	3.219	-2.8	3.063	2.1	3.011	3.8	192
21	4.399	4.373	0.5	4.092	6.9	4.092	7.0	128
22	4.726	4.417	6.5	4.254	9.9	4.254	10.0	128
23	2.747	2.591	5.6	2.442	11.1	2.424	11.8	144
Average	**4.193**	**4.088**	**2.5**	**3.891**	**7.2**	**3.876**	**7.5**	**–**

12. Pinho, A.J., Neves, A.J.R.: Block-based histogram packing of color-quantized images. In: Proc. of the IEEE Int. Conf. on Multimedia and Expo, ICME-2003. Volume 1., Baltimore, MD (2003) 341–344
13. Pinho, A.J., Neves, A.J.R.: JPEG 2000 coding of color-quantized images. In: Proc. of the IEEE Int. Conf. on Image Processing, ICIP-2003. Volume 2., Barcelona, Spain (2003) 181–184
14. Pinho, A.J.: An online preprocessing technique for improving the lossless compression of images with sparse histograms. IEEE Signal Processing Letters **9** (2002) 5–7