Available online at www.sciencedirect.com

SCIENCE DIRECT®

Image and Vision Computing 24 (2006) 534–540

www.elsevier.com/locate/imavis

On the relation between Memon's and the modified Zeng's palette reordering methods

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

Signal Processing Lab, DET/IEETA, University of Aveiro, Campus Universitario de Santiago, 3810-193 Aveiro, Portugal

Received 20 September 2004; received in revised form 3 February 2006; accepted 16 February 2006

Abstract

Palette reordering has been shown to be a very effective approach for improving the compression of color-indexed images by general purpose continuous-tone image coding techniques. In this paper, we provide a comparison, both theoretical and experimental, of two of these methods: the pairwise merging heuristic proposed by Memon et al. and the recently proposed modification of Zeng's method. This analysis shows how several parts of the algorithms relate and how their performance is affected by some modifications. Moreover, we show that Memon's method can be viewed as an extension of the modified version of Zeng's technique and, therefore, that the modified Zeng's method can be obtained through some simplifications of Memon's method.

Keywords: Color-indexed images; Palette recording; Lossless image compression; JPEG-LS; JPEG2000

1. Introduction

A color-indexed image is represented by a matrix of indexes (the index image) and by a color-map or palette. Each index points to a color-map entry, establishing the corresponding color of the pixel. For a particular image, the mapping between index values and colors is not unique—it can be arbitrarily permuted, under the condition that the corresponding index image is changed accordingly. However, although equivalent in terms of representation, for most continuous-tone image coding techniques, such as JPEG-LS [1,2] or lossless JPEG2000 [3–5], different mappings may imply dramatic variations in the compression performance. Moreover, despite the existence of specialized approaches for coding color-indexed images (see, for example [6–9]), it remains an important issue to ensure that standard techniques are able to produce acceptable results within this class of images.

Palette reordering is a class of preprocessing methods aiming at finding a permutation of the color palette such that the resulting image of indexes is more amenable for compression (for a survey, see [10]). These preprocessing techniques have the advantage of not requiring post-processing and of being cost-less in terms of side information. However, if the optimal configuration is sought, then the computational complexity involved can be high. In fact, the number of possible configurations for a table of M colors corresponds to the number of permutations of M objects, which equals $M!$ Clearly, exhaustive search is impractical for most of the interesting cases, which motivated several sub-optimal, lower complexity, proposals.

In this paper, we address two of the most effective palette reordering methods: Memon's method [11] and the modified Zeng's method [12] (for short, we will refer to the modified Zeng's method as mZeng's method). A comparison of the performance and computational complexity of these two methods is presented in [10]. The main objective of this paper is to provide a detailed analysis and comparison of both techniques, showing how they relate, which are their similarities and fundamental differences. A particularly interesting and potentially useful finding is that mZeng's method can be viewed as a simplified version of Memon's method.

The remainder of the paper is organized as follows. In Section 2, both reordering methods are described, using an unifying notation, with the aim of exposing similarities and differences between them. In Section 3, a detailed comparison is performed. Section 4 proceeds with the comparison using experimental data. Finally, in Section 5, some conclusions are drawn.

* Corresponding author. Fax: +351 23 4370545.
E-mail addresses: ap@det.ua.pt (A.J. Pinho), an@ieeta.pt (A.J.R. Neves).

0262-8856/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2006.02.005

2. Palette reordering

Palette reordering methods can be classified into two main classes. To one of those classes, to which we refer as 'color-based methods', belong methods that are characterized by relying only on the information provided by the color palette. The other class characterizes techniques relying only on the statistical information conveyed by the index image, independently of its meaning in terms of color representation. We refer to the techniques in this class as 'index-based methods'.

The main idea behind index-based methods for palette reordering is that colors that occur frequently close to each other should have close indexes. Therefore, based on this principle, the assignment of the indexes is usually guided by some function, $w(i, j)$, measuring the number of occurrences corresponding to pixels with index i that are spatially adjacent to pixels with index j, according to some predefined neighborhood.

In the remainder of this section we describe two techniques that fall under the class of index-based methods, and which are among the best known palette reordering methods [10]: Memon's method [11] and the modified Zeng's method [12].

2.1. Memon's method

Memon et al. formulated the problem of palette reordering within the framework of linear predictive coding [11]. In that context, the objective is to minimize 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

$$\sum_{i=1}^{M}\sum_{j=1}^{M} c_{ij}|i-j|, \tag{1}$$

where c_{ij} 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 (1) 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 [11]. In fact, if we consider a complete non-directed weighted graph $G(V, E, w)$, where each vertex in $V=\{v_1, v_2, \ldots, v_M\}$, corresponds to a palette color, and $w(v_i, v_j)=c_{ij}+c_{ji}$, $(v_i, v_j)\in E$, corresponds to the weight associated to the edge defined between vertices v_i and v_j, then the goal is to find a one-to-one mapping (permutation) σ: $V\to\{1, 2, \ldots, M\}$, among all possible permutations σ_n, satisfying:

$$\sigma = \arg\min_{\sigma_n} \sum_{(v_i,v_j)\in E} w(v_i,v_j)|\sigma_n(v_i)-\sigma_n(v_j)|. \tag{2}$$

With the aim of seeking approximate solutions for (2), Memon et al. proposed two heuristics: one based on simulated annealing, the other, faster to compute, based on a technique called 'pairwise merging'.

Essentially, the pairwise merging heuristic consists on repeatedly merging ordered sets of colors until obtaining a single set. Initially, each color (graph vertex) is assigned to a different set. Then, each iteration consists of two steps:

Step 1: From all possible pairs of sets, choose the one satisfying:

$$(\mathcal{S}_u,\mathcal{S}_v) = \arg\max_{(\mathcal{S}_a,\mathcal{S}_b)} \sum_{a\in\mathcal{S}_a}\sum_{b\in\mathcal{S}_b} w(a,b). \tag{3}$$

Step 2: Among all possible merging combinations of $\mathcal{S}_u$ and $\mathcal{S}_v$, choose the one minimizing

$$\sum_{i=1}^{|\mathcal{S}_m|}\sum_{j>i}^{|\mathcal{S}_m|}(j-i)w(m_i,m_j), \tag{4}$$

where $\mathcal{S}_m=\{m_1,m_2,\ldots,m_{|\mathcal{S}_m|}\}$, $|\mathcal{S}_m|=|\mathcal{S}_u|+|\mathcal{S}_v|$, is the set under evaluation, obtained from a particular merging of $\mathcal{S}_u$ and $\mathcal{S}_v$. The m_i represent, therefore, the elements of sets $\mathcal{S}_u$ and $\mathcal{S}_v$.

To alleviate the computational burden involved in selecting the best way of merging the two ordered sets, Memon et al. proposed a reduced number of configurations [11]. If $\mathcal{S}_u=\{u_1,u_2,\ldots,u_{|\mathcal{S}_u|}\}$ and $\mathcal{S}_v=\{v_1,v_2,\ldots,v_{|\mathcal{S}_v|}\}$ are the two sets under evaluation, and if $|\mathcal{S}_u|,|\mathcal{S}_v|>1$, then the following configurations are considered:

$$\begin{aligned}&\{u_1,u_2,\ldots,u_{|\mathcal{S}_u|},v_1,v_2,\ldots,v_{|\mathcal{S}_v|}\}\\&\{u_{|\mathcal{S}_u|},\ldots,u_2,u_1,v_1,v_2,\ldots,v_{|\mathcal{S}_v|}\}\\&\{v_1,v_2,\ldots,v_{|\mathcal{S}_v|},u_1,u_2,\ldots,u_{|\mathcal{S}_u|}\}\\&\{v_1,v_2,\ldots,v_{|\mathcal{S}_v|},u_{|\mathcal{S}_u|},\ldots,u_2,u_1\}.\end{aligned} \tag{5}$$

Alternatively, if one of the sets has size one, then the following configurations are tested (without loss of generality, we consider $|\mathcal{S}_u|=1$) :

$$\begin{aligned}&\{u_1,v_1,v_2,\ldots,v_{|\mathcal{S}_v|}\}\\&\{v_1,u_1,v_2,\ldots,v_{|\mathcal{S}_v|}\}\\&\{v_1,v_2,u_1,\ldots,v_{|\mathcal{S}_v|}\}\\&\qquad\vdots\\&\{v_1,v_2,\ldots,v_{|\mathcal{S}_v|},u_1\}.\end{aligned} \tag{6}$$

2.2. Modified Zeng's method

The palette reindexing method proposed by Zeng et al. [13] is based on an one-step look-ahead greedy approach, which aims at increasing the lossless compression efficiency of color-indexed images. In [12], a modification of Zeng's algorithm was proposed, relying on a Laplacian model for the distribution of first order prediction residuals, and on the assumption that the entropy of the absolute differences between neighboring pixels is a good indicator of the degree of compressibility of an image.

The algorithm starts by finding the index that is most frequently located adjacent to other (different) indexes, and the index that is most frequently found adjacent to it. This pair of indexes is the starting base for an ordered set $\mathcal{S}$ that will be constructed, one index at a time, during the operation of the reindexing algorithm. We denote by v_i the indexes already assigned to the ordered set (i indicates the position of the index in the ordered set and, therefore, its distance to the left side of the set) and by u those still unassigned. Therefore, just before starting the iterations, $\mathcal{S}=\{v_1, v_2\}$, where

$$v_1 = \arg\max_{u_j} \sum_{u_i \neq u_j} w(u_i,u_j) \tag{7}$$

and

$$v_2 = \arg\max_{u} w(v_1,u). \tag{8}$$

The function $w(i, j)=w(j, i)$ denotes the number of occurrences (measured on the initial index image) corresponding to pixels with index i that are spatially adjacent to pixels with index j. New indexes can only be attached to the left or to the right extremity of the ordered set.

It is well-known that, for a memoryless source, the number of bits required to represent the occurrence of a given symbol s is given by $-\log_2 P(s)$, where $P(s)$ denotes the probability of occurrence of s. Therefore, we start by defining the estimated code length implied by placing index u on the left side of $\mathcal{S}$

$$l_\mathrm{L}(u) = -\sum_{v_i \in \mathcal{S}} w(u,v_i)\log_2 P(i), \tag{9}$$

and by placing it one position farther away

$$l^+_\mathrm{L}(u) = -\sum_{v_i \in \mathcal{S}} w(u,v_i)\log_2 P(i+1). \tag{10}$$

We also calculate similar estimates for the right side

$$l_\mathrm{R}(u) = -\sum_{v_i \in \mathcal{S}} w(u,v_i)\log_2 P(|\mathcal{S}|-i+1), \tag{11}$$

and

$$l^+_\mathrm{R}(u) = -\sum_{v_i \in \mathcal{S}} w(u,v_i)\log_2 P(|\mathcal{S}|-i+2). \tag{12}$$

According to the greedy strategy of Zeng's algorithm, the next index, $\bar{u}$, that should integrate $\mathcal{S}$ is the one that implies the largest increase in code length if its choice is postponed to the next iteration. Then, the new index, $\bar{u}$, should satisfy

$$\bar{u} = \arg\max_{u \notin \mathcal{S}} \Delta l(u), \tag{13}$$

with

$$\Delta l(u) = \begin{cases} l^+_\mathrm{L}(u) - l_\mathrm{L}(u), & \text{if } l_\mathrm{L}(u) < l_\mathrm{R}(u) \\ l^+_\mathrm{R}(u) - l_\mathrm{R}(u), & \text{otherwise.} \end{cases} \tag{14}$$

In words, for each candidate index, u, its best position (left or right) is chosen, i.e. the one that minimizes the code length. Then, among all those indexes, we pick the one producing the largest increase in code length if its choice is postponed to the next iteration.

Using (9) and (10), we can write

$$l^+_\mathrm{L}(u) - l_\mathrm{L}(u) = \sum_{v_i \in \mathcal{S}} w(u,v_i)\log_2 \frac{P(i)}{P(i+1)} = \sum_{v_i \in \mathcal{S}} \alpha_i w(u,v_i), \tag{15}$$

if the best position for index u is the left side, or, using (11) and (12)

$$l^+_\mathrm{R}(u) - l_\mathrm{R}(u) = \sum_{v_i \in \mathcal{S}} w(u,v_i)\log_2 \frac{P(|\mathcal{S}|-i+1)}{P(|\mathcal{S}|-i+2)} = \sum_{v_i \in \mathcal{S}} \alpha_{|\mathcal{S}|-i+1} w(u,v_i), \tag{16}$$

if the best position is the right side, where

$$\alpha_k = \log_2 \frac{P(k)}{P(k+1)}, \tag{17}$$

and where $P(k)$ denotes the probability of occurrence of a difference of k units between two neighboring pixels.

Moreover, we can also write

$$l_\mathrm{R}(u) - l_\mathrm{L}(u) = \sum_{v_i \in \mathcal{S}} w(u,v_i)(\log_2 P(i) - \log_2 P(|\mathcal{S}|-i+1)) = \sum_{v_i \in \mathcal{S}} \beta_i w(u,v_i), \tag{18}$$

where

$$\beta_k = \log_2 \frac{P(k)}{P(|\mathcal{S}|-k+1)}. \tag{19}$$

The new ordered set will be $\{\bar{u},v_1,\ldots,v_{|\mathcal{S}|}\}$, if $l_\mathrm{L}(\bar{u}) < l_\mathrm{R}(\bar{u})$, or $\{v_1,\ldots,v_{|\mathcal{S}|},\bar{u}\}$, otherwise. This iterative process should continue until assigning all indexes. Finally, the re-indexed image is constructed by applying the mapping $v_i \mapsto (i-1)$ to all image pixels, and changing the color-map accordingly.

For exponentially distributed residuals, i.e. considering

$$P(k) = A\theta^k, \quad 0<\theta<1, 0 \leq k < M, \tag{20}$$

Eq. (17) reduces to

$$\alpha_k = \log_2 \frac{A\theta^k}{A\theta^{k+1}} = -\log_2\theta, \tag{21}$$

and (19) reduces to

$$\beta_k = \log_2 \frac{A\theta^k}{A\theta^{|\mathcal{S}|-k+1}} = (2k-|\mathcal{S}|-1)\log_2\theta, \tag{22}$$

i.e. the parameter β_k decreases linearly with k (note that $\log_2\theta<0$), and α_k can be set to an arbitrary positive value, e.g.

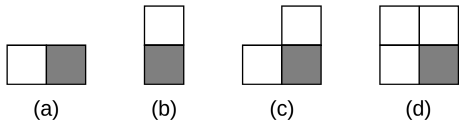

Fig. 1. Configurations of neighboring pixels, in relation to the pixel in gray, for constructing function $w(i, j)$.

one. Moreover, note that, in this case:

$$\Delta l(u) = l_L^+(u) - l_L(u) = l_R^+(u) - l_R(u) = \sum_{v_i \in \mathcal{S}} w(a, v_i). \quad (23)$$

Summarizing, the mZeng's algorithm is composed of an initialization phase, and two iterative steps:

Initialization: Construct the initial set $\mathcal{S} = \{v_1, v_2\}$, as given by (7) and (8).

Step 1: Compute the next index, $\bar{u}$, using

$$\bar{u} = \arg\max_{a \notin \mathcal{S}} \sum_{v_i \in \mathcal{S}} w(a, v_i). \quad (24)$$

Step 2: Compute

$$l_R(\bar{u}) - l_L(\bar{u}) = \sum_{v_i \in \mathcal{S}} (|\mathcal{S}| - 2i + 1) w(\bar{u}, v_i) \quad (25)$$

and choose the left side of $\mathcal{S}$ to attach $\bar{u}$ if $l_R(\bar{u}) - l_L(\bar{u}) > 0$.

3. Analysis of the algorithms

In Section 2, we presented Memon's and mZeng's techniques for palette reordering. We now proceed with a comparative discussion of both algorithms, with the aim of identifying similarities and differences among them. To do so, we will focus our analysis mainly in two points, corresponding to steps 1 and 2 of the algorithms, and to which we refer as the selection and merging phases, respectively. However, before we address these two points, we will discuss some issues regarding the construction of the $w(i, j)$ function and also regarding the initialization phase of the algorithms.

The construction of the $w(i, j)$ function, responsible for conveying the information of how frequently the pairs of neighboring pixels occur in the image, has typically been left somewhat vague by the authors. Memon referred that this function should reflect the number of times a given index i is used as the predicted value of a pixel having index j [11]. However, he did not mention if that hypothetical first-order predictor uses the left pixel, the upper pixel or even the upper-left pixel. Zeng referred that function $w(i, j)$ should represent the number of occurrences of pixels with index i that are spatially adjacent to pixels with index j [13]. However, the type of neighborhood, 4-connected or 8-connected, was not specified.

To better understand how a particular choice of this function affects the performance of the reordering algorithms and, therefore, the compression efficiency of the coding methods applied after reordering, we performed some experiments using the four configurations depicted in Fig. 1. Detailed results are reported in Table 1, and show that, according to the experiments performed, the best configuration for Memon's method is that presented in Fig. 1(a), whereas for mZeng's method is the one depicted in Fig. 1(c).

Regarding the initialization phase, while it is explicit for mZeng's method, defined by (7) and (8), for Memon's technique it coincides with step 1 of the algorithm. In general, the first pair of indexes generated by the two techniques differs, because the most frequent pair, i.e. the one chosen by Memon's method, does not necessarily contain the most frequent index. Once more, it is not evident of how a particular choice of the first pair of indexes might influence the final outcome of the algorithms. To get some indication on this matter, we ran a

Table 1
Comparison of the coding performance using JPEG-LS and lossless JPEG2000 for different constructions of the $w(i, j)$ function

Dither	Colors	Memon				mZeng			
		(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)
JPEG-LS									
No	64	**2.641**	2.660	2.647	2.663	**2.709**	2.719	**2.709**	2.727
	128	**3.441**	3.461	3.461	3.469	3.559	3.569	**3.552**	3.565
	256	**4.204**	4.253	4.225	4.247	**4.473**	4.482	4.475	4.505
Yes	64	3.341	3.329	**3.307**	3.327	3.436	3.445	**3.420**	3.441
	128	**4.112**	4.150	4.122	4.125	4.239	**4.226**	**4.226**	4.238
	256	4.870	4.890	**4.865**	4.887	**5.107**	5.115	5.120	5.138
Global average		**3.768**	3.791	3.771	3.786	3.921	3.926	**3.917**	3.936
JPEG2000									
No	64	**2.981**	2.999	2.985	3.002	3.052	3.061	**3.046**	3.064
	128	**3.793**	3.816	3.812	3.827	3.925	3.930	**3.914**	3.925
	256	**4.576**	4.617	4.595	4.611	**4.868**	4.872	4.870	4.899
Yes	64	3.590	3.568	**3.553**	3.565	3.672	3.677	**3.653**	3.671
	128	**4.354**	4.386	4.359	4.355	4.481	4.461	**4.458**	4.470
	256	5.149	5.153	**5.138**	5.146	5.381	**5.378**	5.388	5.406
Global average		**4.074**	4.090	**4.074**	4.084	4.230	4.230	**4.222**	4.239

The configurations (a)–(d) are depicted in Fig. 1.

Table 2
Comparison of the coding performance using JPEG-LS and lossless JPEG2000 for different initializations of Memon's and mZeng's reordering methods

Dither	Colors	Memon	MemonZi		mZeng	mZengMi	
		bpp	bpp	%	bpp	bpp	%
JPEG-LS							
No	64	2.641	2.642	0.0	2.709	2.744	−1.3
	128	3.441	3.443	−0.1	3.552	3.570	−0.5
	256	4.204	4.206	−0.1	4.475	4.494	−0.4
Yes	64	3.341	3.341	0.0	3.420	3.438	−0.5
	128	4.112	4.117	−0.1	4.226	4.258	−0.8
	256	4.870	4.864	0.1	5.120	5.140	−0.4
JPEG2000							
No	64	2.981	2.982	0.0	3.046	3.085	−1.3
	128	3.793	3.791	0.1	3.914	3.932	−0.5
	256	4.576	4.577	0.0	4.870	4.885	−0.3
Yes	64	3.590	3.590	0.0	3.653	3.671	−0.5
	128	4.354	4.358	−0.1	4.458	4.499	−0.9
	256	5.149	5.144	0.1	5.388	5.417	−0.5

'MemonZi' refers to Memon's method using the initialization of mZeng's method, whereas 'mZengMi' refers to mZeng's method using the initialization of Memon's method.

simple test consisting on permuting the two initialization approaches between the two algorithms (detailed results are given in Table 2).

The outcome of this experiment was that for Memon's method the effect was virtually null. However, plugging the initialization approach of Memon's method into mZeng's technique produced an average decrease in lossless compression of around 0.6%. The main conclusion that can be drawn from this simple experiment is that Memon's method seems to be more tolerant to the initialization than mZeng's method. In fact, this is not surprising, since whereas the former has the possibility of splitting apart the first pair at a later time, therefore correcting for poor initializations, the latter does not have this flexibility.

Step 1, the selection phase, shows some apparently significant differences. As can be observed from the comparison of (3) and (24), whereas (3) seeks the best pair of sets, $(\mathcal{S}_u,\mathcal{S}_v)$, among all possible pairs, (24) seeks the best pair, $(\{\bar{u}\},\mathcal{S})$, but only among the pairs formed by one particular set (which is always the largest) and all the other sets (which are always of size one). Therefore, Memon's method has a greater freedom for picking up the next couple of sets to be merged. In practice, and based on the experimental data set described in Section 4, we found that, on average, the best pair obtained with (3) contains the largest set together with a set of size one for about 80% of the time. In other words, we found that in four out of five iterations, step 1 of Memon's method provides the same output as step 1 of mZeng's method.

Step 2, the merging phase, differs considerably in both algorithms. In fact, whereas for mZeng's technique only two possibilities are considered, i.e.

$$\{\bar{u},v_1,\ldots,v_{|\mathcal{S}|}\} \qquad (26)$$

and

$$\{v_1,\ldots,v_{|\mathcal{S}|},\bar{u}\}, \qquad (27)$$

Memon's method considers, most often, $\max(|\mathcal{S}_u|,|\mathcal{S}_v|)$ merging configurations. This claim is related to the observation already pointed out when step 1 was discussed, i.e. that frequently $\min(|\mathcal{S}_u|,|\mathcal{S}_v|)=1$, which forces the algorithm to use the combinations described in (6). Moreover, if those combinations are removed and only those described in (5) are allowed, then the merging phase of both algorithms becomes equivalent in terms of combinations tested (note that, when $|\mathcal{S}_u|=1$ or $|\mathcal{S}_v|=1$, then only two of the four configurations described in (5) are different, and those coincide with (26) and (27)).

Still regarding step 2 of the algorithms, it remains to be shown that (4) and (25) are, in fact, equivalent. We start by noting that, for configuration (26), Eq. (4) can be written as

$$\sum_{i=1}^{|\mathcal{S}|}\sum_{j>i}^{|\mathcal{S}|}(j-i)w(v_i,v_j)+\sum_{i=1}^{|\mathcal{S}|} iw(u,v_i) \qquad (28)$$

and, for configuration (27), it results in

$$\sum_{i=1}^{|\mathcal{S}|}\sum_{j>i}^{|\mathcal{S}|}(j-i)w(v_i,v_j)+\sum_{i=1}^{|\mathcal{S}|}(|\mathcal{S}|-i+1)w(u,v_i). \qquad (29)$$

Subtracting (28) from (29) we obtain (25), which shows that, for the two configurations represented in (26) and (27), step 2 of both algorithms are equivalent.

The main objective of this section was to show that, although developed using different approaches, the algorithms underlying Memon's and mZeng's techniques share some common points. Effectively, despite the fact of both being based on a Laplacian model of the differences among neighboring pixels, they proposed different heuristic algorithms to address the optimization problem. Moreover, the analysis that we performed showed that Memon's method can be viewed as an extension of mZeng's method.

In the remainder of this paper we proceed with the comparison of both methods, but now with a more

Table 3
Comparison of the coding performance using JPEG-LS and lossless JPEG2000 for several modifications of Memon's and mZeng's methods: '-Grp" indicates that the selection phase always comprises the largest set and one set of size one; '-INS' indicates that only the merging options defined in (5) are allowed, whereas '+INS' indicates the inclusion of the merging configurations defined in (6)

Dither	Colors	Memon							mZeng			
			-GRP		-INS		-GRP -INS				+INS	
		bpp	bpp	%	bpp	%	bpp	%	bpp	%	bpp	%
JPEG-LS												
No	64	2.641	2.655	−0.5	2.748	−4.1	2.722	−3.1	2.709	−2.6	2.660	−0.7
	128	3.441	3.450	−0.3	3.616	−5.1	3.574	−3.9	3.552	−3.2	3.463	−0.6
	256	4.204	4.256	−1.2	4.519	−7.5	4.481	−6.6	4.475	−6.4	4.322	−2.8
Yes	64	3.341	3.338	0.1	3.489	−4.4	3.460	−3.6	3.420	−2.4	3.321	0.6
	128	4.112	4.122	−0.2	4.298	−4.5	4.263	−3.7	4.226	−2.8	4.112	0.0
	256	4.870	4.951	−1.7	5.138	−5.5	5.130	−5.3	5.120	−5.1	4.926	−1.2
JPEG2000												
No	64	2.981	3.000	−0.6	3.096	−3.9	3.068	−2.9	3.046	−2.2	2.996	−0.5
	128	3.793	3.808	−0.4	3.995	−5.3	3.945	−4.0	3.914	−3.2	3.817	−0.6
	256	4.576	4.643	−1.5	4.923	−7.6	4.872	−6.5	4.870	−6.4	4.711	−2.9
Yes	64	3.590	3.589	0.0	3.734	−4.0	3.700	−3.1	3.653	−1.8	3.565	0.7
	128	4.354	4.364	−0.2	4.549	−4.5	4.511	−3.6	4.458	−2.4	4.352	0.0
	256	5.149	5.225	−1.5	5.425	−5.4	5.404	−5.0	5.388	−4.6	5.192	−0.8

Percentages are relative to Memon's method.

experimental character. We will provide detailed experimental results regarding the already addressed aspects of the construction of the $w(i, j)$ function and the initialization phase. Also, we will present and discuss results obtained after changing some parts of both algorithms, in order to better understand, in practice, their similarities and differences.

4. Experimental results

In this section, we present experimental results based on the set of the 23 'kodak' 768×512 true color images.[1] Color quantization was applied, both with and without Floyd–Steinberg color dithering, creating images with 256, 128 and 64 colors. Image manipulations have been performed using version 1.2.3 of the 'Gimp' program.[2] The color-quantized images can be obtained from ftp://www.ieeta.pt/~ap/images/kodak. Compression results are given for JPEG-LS[3] and for lossless JPEG2000.[4] The compression figures presented in Tables 1–3, in bits per pixel (bpp), represent average results over the 23 images, and include the sizes of the uncompressed color-maps.

The first set of experiments, presented in Table 1, evaluates the impact of each of the four configurations depicted in Fig. 1, used to build the function $w(i, j)=w(j, i)$, on the compressibility of the reordered images. As can be seen, configuration (a) is the one that globally shows the best results for Memon's technique. For mZeng's method, the experimental results indicate configuration (c) as the best. However, the results also show that the gain provided by these configurations is only marginal.

Table 2 presents detailed results regarding the initialization phase of both reordering methods. As can be seen, the impact of using mZeng's initialization on Memon's method (denoted by 'MemonZi' in the Table 2) is minimal. On the contrary, the effect of using the initialization procedure of Memon's method in mZeng's technique deserves to be considered. In fact, in this case, the average loss in compression performance is around 0.6%, showing a higher sensibility of mZeng's method to the appropriateness of the first pair of indexes that is created. As mentioned in Section 3, this is due to the reduced flexibility of mZeng's method when compared to Memon's method in terms of the merging capabilities.

In Table 3, we show how the methods behave when some modifications are performed in what we denoted as steps 1 and 2 of the algorithms. The first column of results in Table 3 is devoted to Memon's method (using its own initialization procedure and the configuration of Fig. 1(a) for the construction of $w(i, j)$). The percentages included in Table 3 have been calculated in relation to the compression values displayed in the first column.

The results presented under label '-GRP' have been generated by suppressing the capability of Memon's method to select an arbitrary pair of sets, limiting the choice only to the largest set and to a size-one set. In other words, this corresponds of using step 1 of mZeng's method. As can been observed, in general, there is a decrease in compression performance, somewhat more accentuated for images with more colors.

The results in the column labeled '-INS' have been obtained by suppressing the ability of Memon's method to insert size-one sets into the other set, during the merging phase. This

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

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

[3] Using V2.2 of the SPMG JPEG-LS codec with default parameters (http://spmg.ece.ubc.ca).

[4] Using the JasPer 1.700.2 JPEG2000 codec with default parameters (http://www.ece.uvic.ca/~mdadams/jasper).

corresponds to restricting the merging combinations to only those in (5), i.e. corresponds to using step 2 of mZeng's method. As can be seen, in this case the decrease in compression performance is much larger than in the previous case, showing that the possibility of interleaving indexes during the merging phase is a major difference between the two methods.

The results of a final modification of Memon's method are reported in the column with label '-GRP -INS', where the two modifications just described have been combined. Surprisingly, in this case the results are better than for the '-INS' case. Our conjecture regarding this behavior is that allowing the selection of arbitrary pairs of sets, as defined by step 1 of Memon's method, without allowing breaking some sets in the future, prevents the correction of some poor selections made in previous iterations. Therefore, from the experimental results obtained, it seems that for step 1 of Memon's method to be effective, then step 2 should include the merging configurations of (6).

It is worthwhile to note that if the initialization procedure and the $w(i, j)$ function of mZeng's method are used in Memon's method with the '-GRP -INS' modifications then both algorithms become equivalent. Although, we do not have included those results in Table 3, we verified that, in fact, they are equal. Instead, we include in Table 3 the results of modifying mZeng's method through the inclusion of the capability of putting new indexes not only in the extremities of the set, but also inside it (see column '+INS'). This corresponds to using step 2 of Memon's method in mZeng's method. As can be seen, this modification provides a considerable improvement. However, this is also the part of the algorithm that imposes a complexity of $o(M^4)$, instead of the $o(M^3)$ complexity of the original mZeng's method [10].

5. Conclusions

In this paper, we provided a detailed description and analysis of two of the most effective palette reordering methods, used for improving the compression of color-indexed images by general purpose continuous-tone lossless image coding techniques. Our objective was to show how these two methods relate and how different parts of their corresponding algorithms contribute to their performance. The main conclusion of this study was that Memon's method can be viewed as an extension of mZeng's method, the latter being included into the former.

To achieve this objective, we performed a step by step comparison of both methods using an unifying notation. Moreover, we provided a discussion on issues related to the construction of the function that conveys the neighboring information and also on the effect of the initialization phase. With this work, we believe having contributed to a better understanding of how state-of-the-art techniques for palette reordering work, easing the task of those seeking further improvements.

Acknowledgements

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

References

[1] ISO/IEC, Information technology—Lossless and near-lossless compression of continuous-tone still images, ISO/IEC 14495-1 and ITU Recommendation T.87, 1999.
[2] M.J. Weinberger, G. Seroussi, G. Sapiro, The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS, IEEE Transactions on Image Processing 9 (8) (2000) 1309–1324.
[3] ISO/IEC, Information technology—JPEG 2000 image coding system, ISO/IEC International Standard 15444-1, ITU-T Recommendation T.800, 2000.
[4] A. Skodras, C. Christopoulos, T. Ebrahimi, The JPEG 2000 still image compression standard, IEEE Signal Processing Magazine 18 (5) (2001) 36–58.
[5] D.S. Taubman, M.W. Marcellin, JPEG 2000: Image Compression Fundamentals, Standards and Practice, Kluwer, Dordecht MA, 2002.
[6] P.J. Ausbeck Jr., The piecewise constant image model, Proceedings of the IEEE 88 (11) (2000) 1779–1789.
[7] Y. Yoo, Y.G. Kwon, A. Ortega, Embedded image-domain compression using context models, in: Proceedings of the Sixth IEEE International Conference on Image Processing, ICIP-99, vol. I, Kobe, Japan, October 1999, pp. 477–481.
[8] V. Ratnakar, RAPP: Lossless image compression with runs of adaptive pixel patterns, in: Proceedings of the 32nd Asilomar Conference on Signals, Systems, and Computers, 1998, vol. 2, 1998, pp. 1251–1255.
[9] X. Chen, S. Kwong, J.-F. Feng, A new compression scheme for color-quantized images, IEEE Transactions on Circuits and Systems for Video Technology 12 (10) (2002) 904–908.
[10] A.J. Pinho, A.J.R. Neves, A survey on palette reordering methods for improving the compression of color-indexed images, IEEE Transactions on Image Processing 13 (11) (2004) 1411–1418.
[11] N.D. Memon, A. Venkateswaran, On ordering color maps for lossless predictive coding, IEEE Transactions on Image Processing 5 (5) (1996) 1522–1527.
[12] A.J. Pinho, A.J.R. Neves, A note on Zeng's technique for color reindexing of palette-based images, IEEE Signal Processing Letters 11 (2) (2004) 232–234.
[13] W. Zeng, J. Li, S. Lei, An efficient color re-indexing scheme for palette-based compression, in: Proceedings of the Seveth IEEE International Conference on Image Processing, ICIP-2000, vol. III, Vancouver, Canada, September 2000, pp. 476–479.