

A CONTEXT ADAPTATION MODEL FOR THE COMPRESSION OF IMAGES WITH A REDUCED NUMBER OF COLORS

Armando J. Pinho and António J. R. Neves

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

ABSTRACT

Recently, Chen *et al.* proposed a method for compressing color-quantized images that is based on a binary tree representation of colors and on context-based arithmetic coding with variable size templates. In this paper, we address the problem of context adaptation and we propose a model that provides improvements in the lossless compression capabilities of Chen's method.

1. INTRODUCTION

The compression of color-quantized images (or color-indexed images) can be addressed using two different approaches. One of these approaches relies on the use of standard lossless image coding techniques combined with appropriate preprocessing. Frequently, this preprocessing consists of colormap reordering in order to produce images of indexes that are smoother and, therefore, easier to compress by general purpose image coding techniques, such as JPEG-LS [1, 2] or lossless JPEG2000 [3, 4] (for a recent review and comparison of palette reordering techniques, see [5]).

The second approach for color-quantized image compression calls for the development of specialized coding techniques. Among the most successful methods we can point out, for example, PWC (piecewise-constant image model) [6], EIDAC (embedded image-domain adaptive compression) [7] or RAPP (runs of adaptive pixel patterns) [8]. More recently, a specialized technique has been proposed by Chen *et al.* [9]. This technique, besides its high compression efficiency, is characterized by a lossy-to-lossless quality-progressive decoding, allowing stopping the reconstruction at any point and getting a full-resolution image with a reduced number of colors.

The coding method proposed by Chen *et al.* was inspired by the work of Orchard *et al.* in color quantization [10] and is based on the construction of a binary tree, where the root node represents all image colors and the leaf nodes

represent each individual color. During encoding, the tree is traversed in a specific order, with the aim of minimizing the reconstruction error. For each node, the encoder sends the weighted average color of each of its two children nodes and the location of the pixels having a color change. The encoding of these pixel locations is performed by means of a context-based arithmetic encoder with variable size contexts.

In this paper, we address the model for context adaptation proposed by Chen *et al.* and used by the encoding method introduced in [9]. We propose a modification of this model such that the choice of the context template size is made more effectively, providing increased compression performance.

In the following Section we provide a brief description of Chen's method. Then, we focus our attention in the part concerning the encoding of pixel locations and, particularly, in the context adaptation model. Finally, we propose the new model, we present experimental results and we draw some conclusions.

2. CHEN'S CODING METHOD

In this Section, we provide a brief description of Chen's method (more details can be obtained in [9]). We start by describing how the binary tree is constructed and then we explain how the required information is efficiently encoded.

2.1. Constructing the tree

We denote by M the number of different colors in the image, by $C = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_M\}$ the set of colors (RGB triplets, in our case), and by $S_0 = \{1, 2, \dots, M\}$ the set of color indexes used for identifying the colors, where index i represents color \mathbf{c}_i . The number of pixels associated with color \mathbf{c}_i is given by p_i .

Each node, j , of the binary tree represents a certain subset of the colors of the image. We refer to this node using the corresponding set of color indexes, S_j . Moreover, we

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

associate to each node a representative color, \mathbf{q}_j , which is given by the weighted average color of the node, i.e.,

$$\mathbf{q}_j = \frac{\sum_{i \in S_j} p_i \mathbf{c}_i}{\sum_{i \in S_j} p_i}.$$

Whenever $|S_j| > 1$, the node is split in two, in such way that the two new subsets of colors are separated by a plane which is perpendicular to the principal direction of the data and passes through the average color value, \mathbf{q}_j . The principal direction of the data corresponds to the eigenvector associated with the largest eigenvalue of the covariance matrix of the weighted colors in S_j , \mathbf{C}_j , where

$$\mathbf{C}_j = \sum_{i \in S_j} p_i (\mathbf{c}_i \mathbf{c}_i^t - \mathbf{q}_j \mathbf{q}_j^t).$$

2.2. Traversing the tree

Although, in our explanation, we separated the construction of the tree from the part of traversing the tree and encoding the associated information, in the case of Chen's method these two operations can be done simultaneously. The first node to be encoded is the root node, for which the value of \mathbf{q}_0 is transmitted. Then, until the whole tree is traversed, the following procedure is iterated:

1. From the nodes available for processing, the node with the largest associated eigenvalue of \mathbf{C}_j is chosen. Let us call it node m .
2. Encode the identification of this node, i.e., the value of m (the numbering of nodes is such that it can be reproduced by the decoder without additional side information) and the representative colors of its two children, \mathbf{q}_m^l and \mathbf{q}_m^r , corresponding to subsets S_m^l and S_m^r ($S_m = S_m^l \cup S_m^r$).
3. Encode the location of the pixels with colors belonging to S_m^l . Notice that since the location of the pixels with colors belonging to S_m is known by the decoder, then we only need to encode, for each of those pixels, if its color belongs now to S_m^l or to S_m^r .

3. ENCODING PIXEL LOCATIONS

For encoding the location of the pixels that have their colors modified due to a node splitting, Chen *et al.* proposed context-based arithmetic coding [9]. Contexts are constructed based on the template shown in Fig. 1(a), where the context pixels are numbered according to their distance to the encoding pixel. The context is constructed using a sequence of bits, $b_1 b_2 \dots b_k$, where

$$b_i = \begin{cases} 0, & \text{if } \|\mathbf{q}^i - \mathbf{q}_m^l\| \leq \|\mathbf{q}^i - \mathbf{q}_m^r\|, \\ 1, & \text{otherwise} \end{cases},$$

and where \mathbf{q}^i denotes the current color of the pixel in the reconstructed image corresponding to position i of the context template.

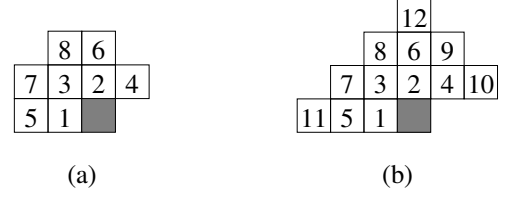


Fig. 1. (a) Context template used by Chen *et al.* (b) Context template used in this work.

Chen *et al.* noticed that better results could be obtained if the size of the context template, i.e., the value of k , was progressively reduced during encoding [9]. Moreover, they proposed the relation

$$k(n) = 9 - \lfloor \log_2 n \rfloor \quad (1)$$

for doing this adaptation, where $n - 1$ indicates the number of colors already encoded. Since the information used for calculating the contexts is binary, then the total number of contexts is given by

$$C(n) = 2^{k(n)}.$$

4. A NEW CONTEXT ADAPTATION MODEL

One of the drawbacks of this context adaptation model is that it does not take into account the size of the image being encoded. In fact, it seems reasonable to admit that larger images might allow larger contexts, because they provide more data for adapting the underlying probability models.

Having this motivation in mind, we investigated the appropriateness of a model that could make use of this information. First, we note that (1) can be rewritten as

$$k(n) = 9 - \lfloor \log_2 n \rfloor = \lceil 9 - \log_2 n \rceil = \left\lceil \log_2 \frac{512}{n} \right\rceil. \quad (2)$$

We intend to study the following generalization of (2),

$$k(n) = \lceil \alpha(N) - \log_2 n \rceil, \quad (3)$$

where N denotes the number of pixels of the image and

$$\alpha(N) = m \log_2 N + b, \quad (4)$$

i.e.,

$$k(n) = \left\lceil \log_2 \frac{N^m 2^b}{n} \right\rceil. \quad (5)$$

Moreover, and in order to accommodate wider variations of $k(n)$ that are due to the $\alpha(N)$ term, we introduced four additional pixel locations in the context template (see Fig. 1(b)).

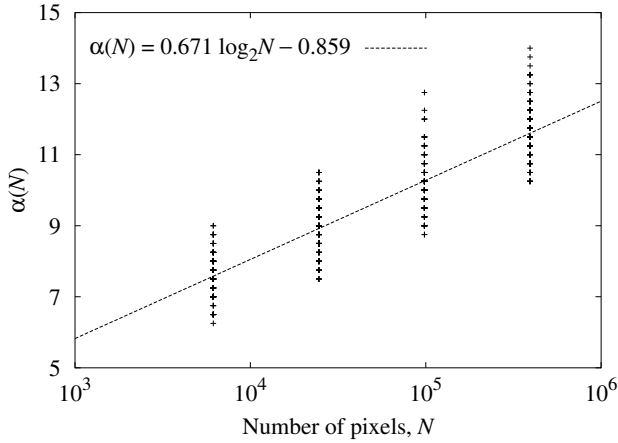


Fig. 2. Plot showing the fitting of the model to the training data. Note that each ‘+’ mark in the graphic generally indicates several superimposed data points.

The function $\alpha(N)$ was adjusted using values taken from the kodak set of 23 images quantized to 256, 128 and 64 colors, and for sizes of 768×512 (the original size), 384×256 , 192×128 and 96×64 .¹ For each of these images, the best value of $\alpha(N)$ was found. Figure 2 shows the fitting of the model to the data obtained from the training set, resulting in $m = 0.671$ and $b = -0.859$.

To evaluate the appropriateness of the proposed context adaptation model, we performed tests using a collection of color-indexed images that have been already used in [11]. These are images from both synthetic and natural origins and of various sizes and number of colors.² Table 1 presents the results obtained using both the context adaptation model originally proposed in Chen’s work, and the model proposed in this paper.

As can be observed from the Table, from the 30 test images, the proposed context adaptation model provides worse results for only three of them. However, in overall terms, the impact of these negative results are negligible, since these images represent a very small fraction (about 0.3%) of the total number of bytes required to encode the whole set of images. The average lossless compression gain attained was 4%.

5. CONCLUSION

In this paper, we proposed a new context adaptation model for the color-quantized image coding method of Chen *et*

al. [9]. This model provided an average lossless compression gain of 4% on a set of images including images from natural and synthetic sources, having a wide range of sizes and number of colors.

6. 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, and G. Sapiro, “The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS”, *IEEE Trans. on Image Processing*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.
- [3] ISO/IEC, *Information technology - JPEG 2000 image coding system*, ISO/IEC International Standard 15444-1, ITU-T Recommendation T.800, 2000.
- [4] D. S. Taubman and M. W. Marcellin, *JPEG 2000: image compression fundamentals, standards and practice*, Kluwer Academic Publishers, 2002.
- [5] A. J. Pinho and A. J. R. Neves, “A survey on palette reordering methods for improving the compression of color-indexed images”, *IEEE Trans. on Image Processing*, vol. 13, no. 11, pp. 1411–1418, Nov. 2004.
- [6] P. J. Ausbeck Jr., “The piecewise-constant image model”, *Proceedings of the IEEE*, vol. 88, no. 11, pp. 1779–1789, Nov. 2000.
- [7] Y. Yoo, Y. G. Kwon, and A. Ortega, “Embedded image-domain compression using context models”, in *Proc. of the 6th IEEE Int. Conf. on Image Processing, ICIP-99*, Kobe, Japan, Oct. 1999, vol. I, pp. 477–481.
- [8] V. Ratnakar, “RAPP: Lossless image compression with runs of adaptive pixel patterns”, in *Proc. of the 32nd Asilomar Conf. on Signals, Systems, and Computers*, 1998, 1998, vol. 2, pp. 1251–1255.
- [9] X. Chen, S. Kwong, and J.-F. Feng, “A new compression scheme for color-quantized images”, *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 904–908, Oct. 2002.
- [10] M. T. Orchard and C. A. Bouman, “Color quantization of images”, *IEEE Trans. on Signal Processing*, vol. 39, no. 12, pp. 2677–2690, Dec. 1991.
- [11] A. J. Pinho and A. J. R. Neves, “A note on Zeng’s technique for color reindexing of palette-based images”, *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 232–234, Feb. 2004.

¹These images can be obtained from <ftp://www.ieeta.pt/~ap/images/kodak>.

²These images can be obtained from <ftp://www.ieeta.pt/~ap/images/synthetic> and <ftp://www.ieeta.pt/~ap/images/natural2/256>.

Image	Colors	Rows \times Cols	Chen contexts		Proposed contexts		
			Size	Bpp	Size	Bpp	%
pc	6	2185 \times 1575	135,051	0.313	117,794	0.273	12.8
books	7	318 \times 179	8,102	1.138	8,047	1.130	0.7
music	8	111 \times 111	830	0.538	829	0.538	0.1
winaw	10	465 \times 633	13,046	0.354	12,980	0.352	0.5
party8	12	286 \times 526	3,341	0.177	3,442	0.183	-3.0
netscape	32	100 \times 612	11,176	1.460	11,176	1.460	0.0
sea_dusk	46	325 \times 484	1,497	0.076	1,595	0.081	-6.5
benjerry	48	60 \times 466	2,847	0.814	2,848	0.814	0.0
gate	84	108 \times 564	15,329	2.013	15,302	2.009	0.2
descent	122	200 \times 320	17,911	2.238	17,853	2.231	0.3
sunset	204	480 \times 640	59,806	1.557	59,489	1.549	0.5
yahoo	229	59 \times 460	5,764	1.699	5,771	1.701	-0.1
airplane	256	512 \times 512	121,424	3.705	121,121	3.696	0.2
anemone	256	471 \times 722	164,523	3.870	161,073	3.789	2.1
arial	256	493 \times 735	238,231	5.259	235,740	5.204	1.0
baboon	256	512 \times 512	178,131	5.436	177,768	5.425	0.2
bike3	256	919 \times 781	287,320	3.202	278,711	3.106	3.0
boat	256	512 \times 512	150,633	4.596	148,546	4.533	1.4
clegg	256	880 \times 814	344,181	3.843	326,807	3.649	5.0
cwheel	256	600 \times 800	122,990	2.049	108,225	1.803	12.0
fractal	256	508 \times 763	241,345	4.981	238,026	4.912	1.4
frymire	256	1105 \times 1118	323,584	2.095	286,396	1.854	11.4
ghouse	256	600 \times 800	214,900	3.581	208,669	3.477	2.9
girl	256	512 \times 512	134,186	4.095	132,364	4.039	1.4
house	256	256 \times 256	32,913	4.017	32,853	4.010	0.2
lena	256	512 \times 512	126,064	3.847	124,018	3.784	1.6
monarch	256	512 \times 768	139,317	2.834	135,679	2.760	2.6
peppers	256	512 \times 512	117,683	3.591	113,865	3.474	3.2
serrano	256	794 \times 629	129,348	2.071	119,057	1.907	8.0
tulips	256	512 \times 768	145,345	2.957	141,568	2.880	2.6
Average	–	–	–	2.267	–	2.176	4.0

Table 1. Lossless compression results, in number of bytes and in bits per pixel (bpp), for a number of synthetic and natural color-indexed images. The “Chen contexts” refers to the compression results obtained with the context adaptation model originally proposed by Chen *et al.*, whereas the “Proposed contexts” columns show compression results for the context adaptation model proposed in this paper. The percentages represent compression gains obtained by the new model.