ENCODING OF IMAGE PARTITIONS USING A STANDARD TECHNIQUE FOR LOSSLESS IMAGE COMPRESSION

Armando J. Pinho

Dep. Electrónica e Telecomunicações / INESC
Universidade de Aveiro, 3810 Aveiro, Portugal
(Fax +351–34–370545, Email `ap@inesca.pt`)

ABSTRACT

Recently, a new technique for the lossless encoding of boundary maps was introduced, which is based on the concept of "transition points". In this paper we show that, using a simple representation for the transition points, it is possible to use the JBIG image coding standard for the encoding of image partitions. Moreover, this new approach outperforms, in most cases, differential chain-coding both in efficiency and simplicity of implementation.

1. INTRODUCTION

Chain-coding is a well known method for the encoding of lines superimposed on a regular grid. Several applications have motivated further study and development of this technique, since its introduction by Freeman in 1961 [1]. Among them, we point out the encoding of geographic maps containing terrain contours and coastlines [2], line drawings [3], bi-level images [4], gray-level images [5], object shapes [6–9], and image partitions [10].

One of the most demanding applications has been region-based image coding. The so-called "second-generation" image compression methods [11, 12] boosted the interest of the researchers on finding efficient ways of representing the shape of isolated objects (shape-coding) or sets of neighboring objects (or regions) of a visual scene (partition-coding). Nevertheless, chain-coding seems to maintain a leading position (or, at least, a tie) in what concerns efficiency and popularity in the task of lossless shape-coding [9] and lossless partition-coding [13].

In this paper we address the latter case, i.e., the lossless encoding of image partitions, using a contour-based approach. However, instead of the traditional chain-coding philosophy, we adopt a recently proposed approach, which is based on the concept of "transition point" [14]. These points occur at locations of the boundary map (representing the image partition) which are characterized by changes of state in the columns and / or rows of the edge elements.

The main contribution of this paper is to show that, using the idea of transition point, arbitrary boundary maps representing partitions of images can be transformed quite easily into what we call "the map of transition points". As explained below, a map of transition points can be regarded as an image having two planes, i.e., two bits per pixel. This fact allow us to transform the problem of encoding image partitions into the problem of lossless image coding. To perform the image compression step we chose a standard technique: JBIG. The experimental results presented in this paper show its excellent ability for this task, offering even better results than the ones reported in [14], for which an *ad hoc* encoder was used.

In the remainder of this section we review the concept of transition point, and explain how it can be used for boundary encoding. A more detailed explanation can be found in [14].

Let us consider the set of sites represented in Fig. 1 by small black squares, and let us call them "nodes". An image having $N_{\mathcal{R}}$ rows and $N_{\mathcal{C}}$ columns has $N_{\mathcal{R}}(N_{\mathcal{C}}-1)+N_{\mathcal{C}}(N_{\mathcal{R}}-1)$ edge elements (the white rectangles in Fig. 1) and $N_{\mathcal{R}}N_{\mathcal{C}}-1$ nodes. Moreover, we refer to the nodes marked by arrows as "starting nodes" (there are a total of $N_{\mathcal{R}}+N_{\mathcal{C}}-2$ of them).

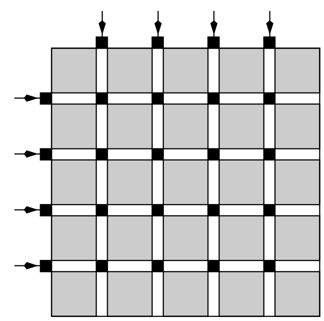

Figure 1: Location of edge elements (white rectangles) and nodes (small black squares). Nodes marked by arrows are called "starting nodes". Shaded squares indicate pixel positions.

If we scan a row of the boundary map from left to right (or a column from top to bottom), beginning at the corresponding starting node, we are able to encode the boundary contents of the given row (or column) by storing the position of the nodes where the sequence of edge elements changes state, i.e., where it goes from a sequence of active edge elements to a sequence of inactive edge elements or vice-versa. We call these nodes "transition points" of the boundary map. At most, a boundary map may have $N_{\mathcal{R}}N_{\mathcal{C}}-1$ transition points, i.e., as much as the number of nodes.

By applying this algorithm to the $N_{\mathcal{R}}-1$ rows of horizontal edge elements and to the $N_{\mathcal{C}}-1$ columns of vertical edge elements of the boundary map, it is possible to represent arbitrary boundaries. Figure 2 shows a small example of a boundary map corresponding to a partition (six regions) of the image. The labeled white squares mark the transition points.

It is not difficult to find out, in this example, that there are transition points due to changes only in the vertical direction (**a** and **h**), only in the horizontal direction (**b** and **d**) and in both directions (**c**, **e**, **f** and **g**). We call them, respectively, vertical tees (**V**), horizontal tees (**H**) and corners (**C**). Figure 3 shows the assignment of these

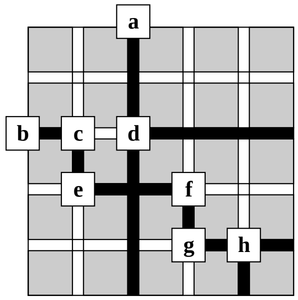

Figure 2: Example of a boundary map and corresponding location of its transition points (labeled white squares).

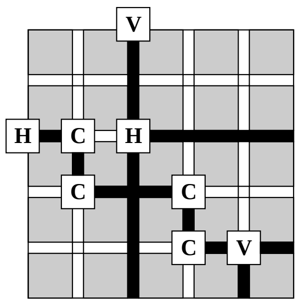

Figure 3: Assignment of the type of transition point: **C** for corner, **H** for horizontal tee and **V** for vertical tee.

three types of transition points to the nodes of the boundary map displayed in Fig. 2.

Having the location and type of all transition points, the reconstruction of the original boundary map is a trivial procedure. In fact, the reconstruction operation can be viewed as a pen moving across each row of nodes (from left to right) and across each column of nodes (from top to bottom) according to the following "drawing rules":

- At the beginning of each row or column the pen is up.
- While moving across a row, the pen changes state (up to down or down to up) if it encounters a corner or a horizontal tee transition point (**C** or **H**).
- While moving across a column, the pen changes state (up to down or down to up) if it encounters a corner or a vertical tee transition point (**C** or **V**).

In order to show the capabilities of this concept when applied to the encoding of boundary maps, it was proposed, in [14], a procedure for representing, efficiently, the set of transition points associated with the boundary map to be encoded. Basically, this procedure relies on the construction of a symbol stream formed by two components: (1) distance, d_i, between transition points; (2) type of transition point, t_i.

The symbol stream, constructed by scanning the nodes as depicted in Fig. 4, has the following structure:

$$d_1, t_1, d_2, t_2, \ldots, d_N, t_N, d_{N+1}$$

where $d_i \in \mathbb{N}_0$, $t_i \in \{\mathbf{C}, \mathbf{H}, \mathbf{V}\}$, and N is the number of transition points.

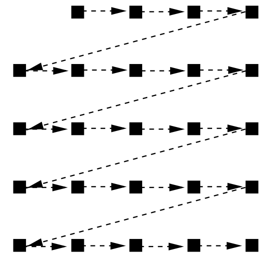

Figure 4: Scanning pattern used in [14] for the construction of the code stream.

After obtaining the symbol stream, the next step is the transformation of symbols into binary codes. To achieve an efficient representation of the boundary maps, variable length codes should be assigned to each of the two data components. In [14] a shift code was adopted for the representation of the d_i component. For the t_i component, the following assignment of binary codes was used: code "0" for symbol **C**, code "10" for symbol **V** and code "11" for symbol **H**. Typically, a shorter codeword can be assigned to symbol **C**, since it is generally the most frequent in a boundary map.

2. THE MAP OF TRANSITION POINTS

As presented above, the idea of using transition points for the encoding of boundary maps is quite simple. Moreover, the implementation of the procedure required for the generation of the two streams of symbols (d_i and t_i) is straightforward. However, the same simplicity does not usually apply to the design and implementation of *ad hoc* binary stream encoders and decoders for variable length codes.

In the remainder of this paper, we present and justify an alternative approach that relies on "of the shelf" standard image encoding algorithms. This is obtained through a suitable transformation of the information conveyed by the two symbol streams mentioned above into an image that can be compressed by standard lossless image compression techniques. In fact, from Figs. 1 and 4 it is not difficult to conclude that, apart from the upper left corner, the set of nodes form an image with the same geometry as the original one. Therefore, the boundary map corresponding to a $N_\mathcal{R} \times N_\mathcal{C}$ image can be represented by another $N_\mathcal{R} \times N_\mathcal{C}$ image whose pixels have only four possible values: three of them for representing the three different types of transition points, and one meaning "no transition point present". Figure 5 shows this image (at symbol level) corresponding to the boundary map of Fig. 3. We will refer to this kind of images as "maps of transition points". Figure 6 shows an example of a boundary map of the upper left quarter of the "lena" image, and the respective image of the map of transition points.

As stated above, the aim of this paper is to show that the encoding of boundary maps describing image partitions can be easily transformed into a lossless image compression problem. Moreover, we show that a standard image coding technique can be used, and also that it offers advantages over the previously proposed encoding method. This new approach requires only a minimal effort of adaptation, which is the construction of the map of transition points.

JBIG (Joint Bi-level Image Experts Group) is a relatively new standard for lossless compression of binary and low-precision gray-

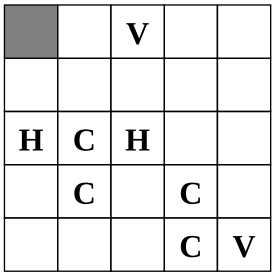

Figure 5: The map of transition points corresponding to the boundary map of Fig. 3. The shaded square indicates that the value of that pixel is arbitrary, since it has no corresponding node.

level images (typically, having less than 6 bits per pixel), known to outperform other existing standards, such as Group 3 and 4 and lossless JPEG [15–17]. Having this in mind, we adopted the JBIG image coding standard for the lossless compression of the maps of transition points.

3. EXPERIMENTAL RESULTS

Figure 7 shows two images (256×256 pixels, 8 bpp), "lena" and "couple", which were used in the experiments reported in this paper. Several boundary maps were generated, using a region growing technique, and lossless encoded using JBIG, according to the approach presented in this paper (denoted as "Tr") and to the differential chain-coding method (denoted as "Ch").

Figure 8 depicts plots of the number of bits required to encode those boundary maps, as a function of the number of active edge elements, for each of the two test images. Each graphic displays results obtained using the two encoding methods ("Tr" and "Ch").

Based on these results, and also on the results presented in [14], we can conclude that the proposed method provides, in general, better compression ratio than differential chain-coding. The improvement clearly increases with the increase in complexity of the boundary maps (larger number of active edge elements). Moreover, JBIG seems to be quite appropriated for encoding the map of transition points. In fact, if we compare the results (for the "lena" image) obtained with JBIG with the results previously obtained [14], we conclude that JBIG is, generally, better.

4. CONCLUSIONS

In this paper we have shown that the problem of encoding image partitions can be easily converted into a problem of lossless image coding. The conversion between the boundary maps that describe the partitions and the images is obtained through the concept of map of transition points. Also, we provided experimental evidence that JBIG can be used successfully in this task. This fact is significant, since it offers a straightforward approach for the implementation of the method, avoiding the step of designing and implementing non-standard *ad-hoc* encoding techniques.

Acknowledgment

The author wishes to thank Markus Kuhn for providing the JBIG-KIT software as a publicly available package (ftp.informatik.uni-erlangen.de/pub/doc/ISO/JBIG). This package was used in the work described in this paper for JBIG coding and decoding of images.

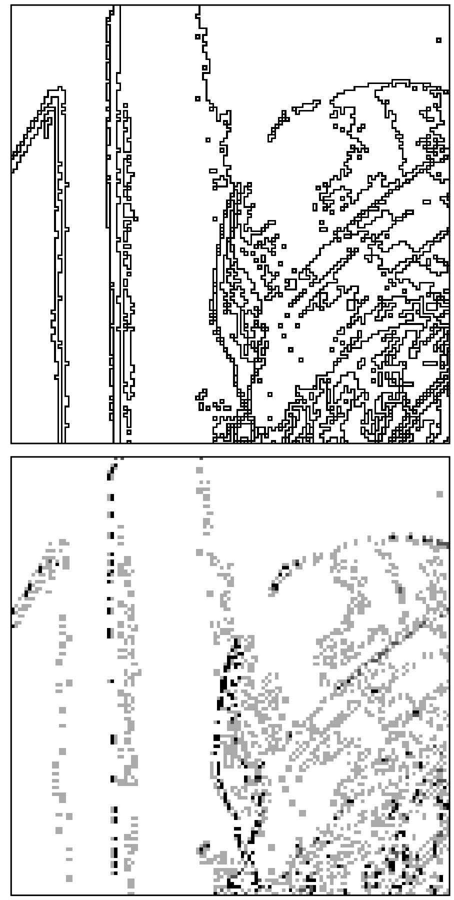

Figure 6: Top, the boundary map (for a given segmentation) of the upper left quarter of the "lena" image. Bottom, the corresponding image of the map of transition points.

5. REFERENCES

[1] H. Freeman. On the encoding of arbitrary geometric configurations. *IRE Trans. on Electronic Computers*, 10:260–268, June 1961.

[2] T. Kaneko and M. Okudaira. Encoding of arbitrary curves based on the chain code representation. *IEEE Trans. on Communications*, 33(7):697–707, July 1985.

[3] T. Minami and K. Shinohara. Encoding of line drawings with a multiple grid chain code. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(2):269–276, March 1986.

[4] R. R. Estes Jr. and V. R. Algazi. Efficient error free chain coding of binary documents. In *Proc. of the Data Compression Conf., DCC'95*, pages 122–131, Snowbird, Utah, March 1995.

[5] J. A. Robinson. Image coding with ridge and valley primitives. *IEEE Trans. on Communications*, 43(6):2095–2102, June 1995.

[6] M. Eden and M. Kocher. On the performance of a contour coding algorithm in the context of image coding. Part I: con-

Figure 7: Images used to obtain the experimental results presented in this paper: “lena” and “couple”.

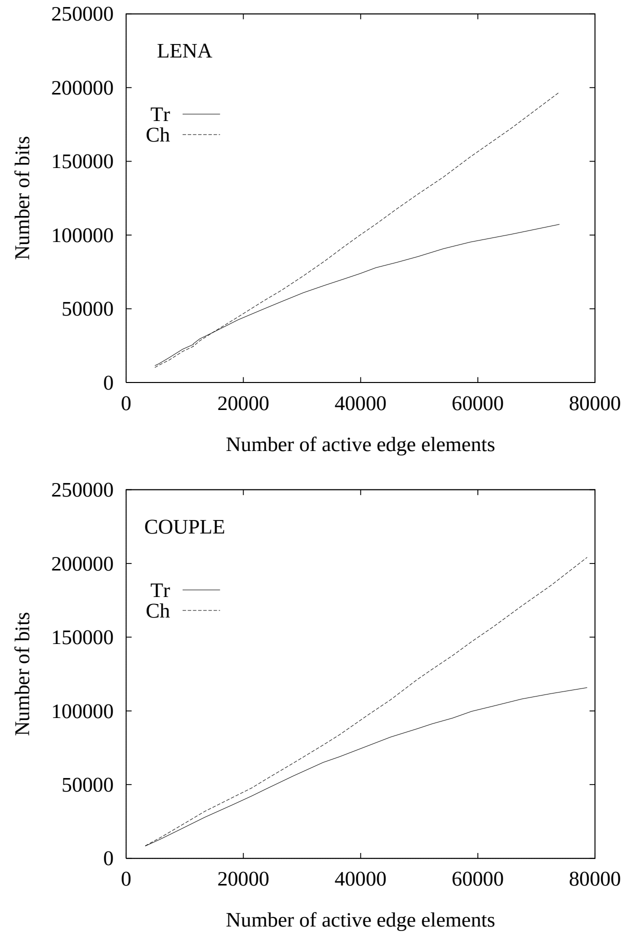

Figure 8: Plots of the number of bits, as a function of the number of active edge elements, required to encode several boundary maps of the “lena” (top) and “couple” (bottom) images. “Tr” refers to the JBIG-based method of transition points, whereas “Ch” denotes the method of differential chain-coding.

tour segment coding. *Signal Processing*, 8(4):381–386, July 1985.

[7] O. Egger, F. Bossen, and T. Ebrahimi. Region based coding scheme with scalability features. In G. Ramponi, G. L. Sicuranza, S. Carrato, and S. Marsi, eds., *Signal Processing VIII — Theories and Applications, Proc. of the 8th European Signal Processing Conf., EUSIPCO’96*, pages 747–750, Edizioni LINT Trieste, Trieste, Italy, 1996.

[8] J.-W. Chung, J.-H. Moon, and J.-K. Kim. Conditional differential chain coding for lossless representation of object contour. *Electronics Letters*, 34(1):55–56, January 1998.

[9] A. K. Katsaggelos, L. P. Kondi, F. W. Meier, J. Ostermann, and G. M. Schuster. MPEG-4 and rate-distortion-based shape-coding techniques. *Proceedings of the IEEE*, 86(6):1126–1154, June 1998.

[10] F. Marqués, J. Sauleda, and A. Gasull. Shape and location coding for contour images. In *Picture Coding Symposium*, pages 18.6.1–18.6.2, Lausanne, Switzerland, March 1993.

[11] M. Kunt, A. Ikonomopoulos, and M. Kocher. Second-generation image-coding techniques. *Proceedings of the IEEE*, 73(4):549–574, April 1985.

[12] M. Kunt, M. Benard, and R. Leonardi. Recent results in high-compression image coding. *IEEE Trans. on Circuits and Systems*, 34(11):1306–1336, November 1987.

[13] P. Salembier, F. Marqués, and A. Gasull. Coding of partition sequences. In L. Torres and M. Kunt, eds., *Video Coding: The second generation approach*, pages 125–170, Kluwer Academic Publishers, 1996.

[14] A. J. Pinho. A method for encoding region boundaries based on transition points. *Image and Vision Computing*, 16(3):213–218, March 1998.

[15] H. Hampel, R. B. Arps, C. Chamzas, D. Dellert, D. L. Duttweiler, T. Endoh, W. Equitz, F. Ono, R. Pasco, I. Sebestyen, C. J. Starkey, S. J. Urban, Y. Yamazaki, and T. Yoshida. Technical features of the JBIG standard for progressive bi-level image compression. *Signal Processing: Image Communication*, 4(2):103–111, April 1992.

[16] *Information technology - Coded representation of picture and audio information - progressive bi-level image compression*. International Standard ISO/IEC 11544:1993 and ITU-T Recommendation T.82(1993), March 1993.

[17] A. N. Netravali and B. G. Haskell. *Digital pictures: representation, compression and standards*. Plenum, New York, 2nd edition, 1995.