Lossless compression of binary image using move-to-front transform and two-role encoder

R. Benzid

An algorithm dedicated to binary image lossless compression is proposed. It uses the move-to-front transform associated to the two-role and the arithmetic encoders. Obtained results, when applying the technique on the benchmark CCITT test images, demonstrate clearly its effectiveness. Therefore, an average bit rate of 0.06 per pixel is achieved.

Introduction: Developed algorithms for data compression can be categorised into two main groups. The first one encloses the lossless techniques which guarantee the perfect preservation of the initial quality of the compressed data. Conversely, the second category includes lossy methods which can attain high compression ratio, compared to lossless strategies. However, they should conserve an adequate level of reconstruction quality appropriate to the application used. As an example, constraints of lossy compression of medical images are more severe than those in audio or natural image compression [1, 2].

One of the open issues that can be considered, is lossless bi-level (binary or two-tone) image compression. Many researches have been conducted to develop algorithms allowing high compression ratio (CR) [3–8]. In [3], the authors presented a non-symmetry and anti-packing (NAM) method for compression permitting both lossless and lossy compression. They reported that the proposed scheme outperforms the linear quadtree and the RLE encoder. In the same optic, Dai and Zakhor [4] suggested a lossless compression technique taking advantage of the two very dissimilar compression techniques: context-based modelling and Lempel–Ziv (LZ) style copying. It has been used for integrated circuit layout compression. The authors of [5–7] used different strategies for efficiently partitioning the whole image into homogeneous rectangular regions that can be easily stored. Another alternative was shown in [8]. Its idea is based on the ordered binary decision diagram (OBDD). Later, an improved version was considered in [9]. Moreover, in [10], the logic functions and spectra were combined successfully to achieve results near to those of [9].

In this context, in this Letter we propose a method for encoding bi-level images. It includes the move-to-front (MTF) transform that reduces the image entropy. Associated to the two-role encoder (TRE), efficient encoding of the resulting data is ensured [11]. The two key steps of the proposed algorithm are: the MTF transform and the TRE encoder. In the following, we present an overview of the two methods.

Move-to-front transform: Its goal is the reduction of the image entropy. As a preprocessing step, it is used, usually, in the case of a data source that presents successive runs of different characters issued from a given alphabet. The result of this transform consists of the replacement of all successive characters in a given run by 0 except the first one. The isolated characters will be replaced with their respective index in the look-up table containing the constituting alphabet of the data source.

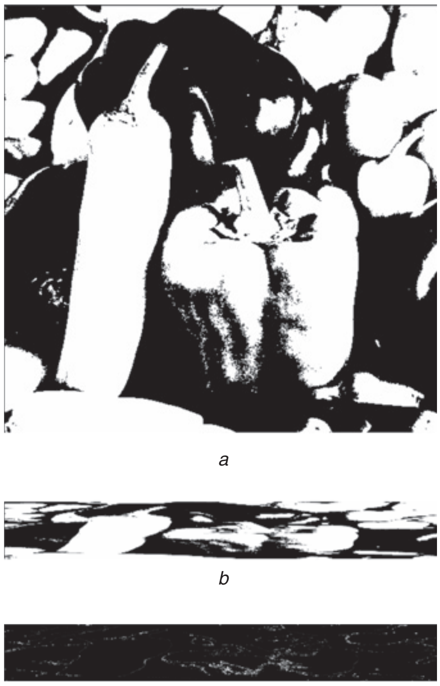
For illustration, we suppose that the source of data is {15, 15, 15, 7, 10, 10, 10, 10, 200}. The alphabet (look-up table) of this source is {7, 10, 15, 200} with respective positions 0, 1, 2 and 3. At the first step, the first 15 of the data source is replaced by its position in the look-up table (position 2) and it is moved to the front (to position 0). The new look-up table will be {15, 7, 10, 200} which gives two consecutive 0s for the two following 15. 7 is replaced by 1 and it is also moved to the front. Consequently, the look-up table is updated as follows: {7, 15, 10, 200}. Conserving the same reasoning, the 10, 10, 10 will be replaced by 2, 0, 0, 0 and again the look-up table is updated to become: {10, 7, 15, 200} then 200 is finally replaced by its position 3. The transformed data source is 2, 0, 0, 1, 2, 0, 0, 0, 3. Note that all the runs of successive characters will be replaced by 0 (except the first character of each run) and the other isolated characters can be of reduced values (depending, evidently, on the length of the look-up table).

Two-role encoder: This encoder is a variant of the conventional RLE algorithm. It is well suited for data sources presenting several successive runs of one most frequent code, which is the case of resulting data obtained from the MTF transform.

Its principle is quite simple. To describe this encoder, first assume that each element of the MTF resulting data is of Q bits. Each nonzero (NZ) element will be shifted by a value of 2^Q. Thus the shifted NZ elements will be in the range $[2^{Q+1} - 1,\ 2^Q + 1]$ and coded with codes of width $(Q + 1)$ bits. With the same $(Q + 1)$ width, any successive runs of zeros can be coded such that the minimum run of zeros that can be coded is 1 and the longer run of zeros can be 2^Q. For illustration, assume that the resulting data after the MTF transform is 1,130, 154, 255, 0, 0, 0, 0, 0, 0. Subsequently, the TRE encoder codes this data as follows: 257, 386, 410, 511, 6, where each code is of width 9 bits (it is remarked that it is slightly different from the algorithm presented in [11]).

Method description: The proposed algorithm can be summarised as follows:
Step 1: Let *I* be the original binary image of size NxM binary pixels. For each column group 8 bits per 8 bits the binary pixel to form a new image of resolution of 8 bits per pixel. The new size is N/8xM.
Step 2: Transform the new image to a vector V by concatenating its lines.
Step 3: Apply the move-to-front transform to the vector V (many runs of zeros appear).
Step 4: Apply the two-role-encoder to the resulting data (each TRE code is of 9 bits width).
Step 5: Apply the well-known arithmetic encoder at the last stage.

For decompression we first apply the arithmetic decoder. Next, the simple TRE decoder is used by comparing each TRE code to 256, if this latter is greater than 256, remove 256 (the resulting value is the value of the already coded NZ), else generate TRE successive zeros. In the same way, apply the inverse MTF using the initial look-up table. To obtain the 8 bits per pixel image, transform the resulting vector to a matrix of N/8xM. At the last step, split each pixel to its constituting 8 bits and the original NxM binary image is restored.



a

b

c

Fig. 1 *Visual illustration of proposed method CR = 5.32:1*

a Original peepers binary image
b Binary to grey level image transformation
c MTF transformed image

Experimental results: The eight benchmark test images used in our experiments were furnished by the French PTT to the CCITT. The documents were scanned with a nominal resolution of 8 pixels/mm giving 1680 pixels per text line. A white margin was added to the right extending each line to 1728 pixels. With columns of 2376 pixels for each, the entire page provides a total of 4105728 binary pixels [12].

For illustration, we present, in Fig. 1, the original peepers binary image, transformed (8 bits/pixel) image and its MTF transformed image. The binary peepers image is obtained by applying the OTSU algorithm on the original 256 grey level image of size 512×512 [13]. We report that the MTF transform can be considered as an edge detector operator. It means that the pixels of regular parts are nullified, and only pixels of contours are nonzero. The use of the TRE encoder is, evidently, justified in this situation. Finally, the application of the arithmetic encoder improves the storage efficiency.

The performance measure is the compression ratio expressed as:

$$CR = \frac{\text{Size of original image in bits}}{\text{Size of compressed image in bits}} \tag{1}$$

Table 1 summarises obtained results by the application of the elaborated method on the eight CCITT images compared to those in [8–10]. From reported results it is clear that the proposed method performs better than the three mentioned algorithms.

Table 1: Comparative results in terms of compression ratio

Image	Method 1 [8]	Method 2 [9]	Method 3 [10]	Proposed method
CCITT 1	6.20	17.41	17.50	20.87
CCITT 2	10.41	29.71	27.40	30.94
CCITT 3	4.4	11.80	11.80	14.32
CCITT 4	1.93	5.10	4.80	5.88
CCITT 5	3.78	10.43	10.60	12.20
CCITT 6	7.29	20.14	19.50	22.27
CCITT 7	1.99	5.35	5.20	6.07
CCITT 8	6.18	17.53	15.70	17.78

Conclusion: We present a simple and efficient algorithm for lossless binary images. After a preprocessing step consisting of the transformation of the binary image to a 256 grey image, the move-to-front reduces considerably the variability of codes contained initially in the original data. Therefore, a dominant code (0) appears and it is found in many successive runs. As a result, the transformed data is suitable to be compressed by the two-role-encoder. The use of the arithmetic encoder, at the last step, improves the compression ratio. Future work will focus on lossless compression of grey and colour images.

25 August 2010
doi: 10.1049/el.2010.2362

R. Benzid (*Department of Electronics, Faculty of Engineering, University of Batna, 05000, Algeria*)

E-mail: rbenzid@lycos.com

References

1 Sayood, K.: 'Introduction to data compression' (Morgan Kuffmann, San Fransisco, USA, 2006)
2 Nelson, M.: 'The data compression book' (M&T Books, New York, USA, 1995)
3 Yunping, Z., Chuanbo, C., and Mudar, S.: 'A NAM representation method for data compression of binary images', *Tsinghua Sci. Technol.*, 2009, **14**, (1), pp. 139–145
4 Dai, V., and Zakhor, A.: 'Lossless compression of VLSI layout image data', *IEEE Trans. Image Process.*, 2006, **15**, (9), pp. 2522–2530
5 Mohamed, S.A., and Fahmy, M.M.: 'Binary image compression using efficient partitioning into rectangular regions', *IEEE Trans. Commun.*, 1995, **43**, (5), pp. 1888–1893
6 Quddus, A., and Fahmy, M.M.: 'Binary text image compression using overlapping rectangular partitioning', *Pattern Recognit. Lett.*, 1999, **20**, pp. 81–88
7 Zahir, S., and Naqvi, M.: 'A new rectangular partitioning based lossless binary image compression scheme'. IEEE Canadian Conf. Electrical and Computer Engineering, Saskatoon, Canada, 2005, pp. 281–285
8 Starkey, M., and Bryant, R.E.: 'Using ordered binary-decision diagrams for compressing images and image sequences', Tech. Rep. (Carnegie Mellon University, CMU-CS-95-105, 1995)
9 Mateu-Villarroya, P., and Prades-Nebot, J.: 'Lossless image compression using ordered binary-decision diagrams', *Electron. Lett.*, 2001, **37**, (3), pp. 162–163
10 Falkowski, B.J.: 'Lossless binary image compression using logic functions and spectra', *Comput. Electr. Eng.*, 2004, **30**, pp. 17–43
11 Benzid, R., Marir, F., and Bouguechal, N.E.: 'Electrocardiogram compression method based on the adaptive wavelet coefficients quantization combined to a modified two-role encoder', *IEEE Signal Process. Lett.*, 2007, **14**, (6), pp. 373–376
12 Langdon, G.G., and Rissanen, J.: 'Compression of black-white images with arithmetic coding', *IEEE Trans. Commun.*, 1981, **COM-29**, (6), pp. 858–867
13 Otsu, N.: 'A threshold selection method from gray-level histograms', *IEEE Trans. Syst. Man Cybern.*, 1979, **SMC-9**, (1), pp. 62–66