

Coding Long Contour Shapes of Binary Objects

Hermilo Sánchez-Cruz and Mario A. Rodríguez-Díaz

Departamento de Sistemas Electrónicos. Centro de Ciencias Básicas
Universidad Autónoma de Aguascalientes. Av. Universidad 940
C.P. 20100. Aguascalientes, Ags. México
hsanchez@correo.uaa.mx

Abstract. This is an extension of the paper appeared in [15]. This time, we compare four methods: Arithmetic coding applied to 3OT chain code (Arith-3OT), Arithmetic coding applied to DFCCE (Arith-DFCCE), Huffman coding applied to DFCCE chain code (Huff-DFCCE), and, to measure the efficiency of the chain codes, we propose to compare the methods with JBIG, which constitutes an international standard. In the aim to look for a suitable and better representation of contour shapes, our probes suggest that a sound method to represent contour shapes is 3OT, because Arithmetic coding applied to it gives the best results regarding JBIG, independently of the perimeter of the contour shapes.

Keywords: Efficiency, Arithmetic coding, Chain code, Contour, Shapes, Binary objects.

1 Introduction

The shape representation of binary objects, is an active research in computer vision, pattern recognition and shape analysis. Binary objects can be seen as bi-level images, because they also are composed of two tones: black and white (B/W). Chain code techniques can be used to represent shape-of-objects in a right discretized fashion. It has been reported interesting applications using chain codes, for example: Mckee and Aggarwal [1] have used chain coding in the process of recognizing objects. Hoque et al. [2] proposed an approach to classify handwritten characters, based on a directional decomposition of the corresponding chain-code representation.

A chain code can be viewed as a connected sequence of straight-line segments with specified lengths and directions [3]. Chain codes can also be used to identify corners in shapes [4]. Salem et al. [5] discuss the capabilities of a chain code in recognizing objects. In [6], Sánchez-Cruz and Rodríguez-Dagnino proposed a code contour shape, called 3OT, and they found better compression properties than Freeman codes.

To compress binary objects, Liu and Zalik proposed the Differential Freeman Chain Code of Eight Directions (DFCCE) by using Huffman algorithm [7]. Liu et al. [8] introduced three new chain codes based on the VCC [9]. The main reason for the popularity of chain codes is their compression capabilities. There are two main categories for image compression algorithms, namely algorithms with loss of information such as

MPEG, JPEG, etc., and lossless compression algorithms, such as JBIG, Huffman, Arithmetic and Lempel-Ziv (LZ). For instance, LZ algorithms are some of the most successful methods for text compression [10] and [11]. Aguinaga et al. [12], compared different entropy coding schemes applied to bi-level images by using run-length codes.

Another evidence to support that is better to codify contours shapes with 3OT codes instead than DFCCE, is by considering next analysis. In [8], three codes were introduced: EVCC, VVCC and C_VCC, in such a paper, C_VCC is found as the best compressor. In the spirit to look for better performance than C_VCC, Sanchez-Cruz, et al. [13], compared C_VCC with 3OT by doing some assignments, when changing appropriately the length of the 3OT chains. The procedure is easy: when contours are coded by 3OT, for every five consecutive 0's, a symbol "3" is introduced; for every five consecutive 1's the symbol "4" is introduced; and for every substring 0110 in the 3OT, a "5" symbol is introduced in a new alphabet: $M_{3OT} = \{0,1,2,3,4,5\}$, where each symbol has been obtained by doing the next assignments from 3OT chains:

$$\begin{aligned} 0 &\longrightarrow 0 \\ 1 &\longrightarrow 1 \\ 2 &\longrightarrow 2 \\ 00000 &\longrightarrow 3 \\ 11111 &\longrightarrow 4 \\ 0110 &\longrightarrow 5 \end{aligned}$$

These new assignments permit us to improve the most recent code known as C_VCC. So, this fact constitutes another evidence that supports to utilize 3OT chain code to represent contour shapes.

An international committee has generated a standard image compressor for bi-level images called Joint Bi-level Image Experts Group (JBIG), which was primarily designed for compression without loss of information, [14]. JBIG has already been improved and the new standard is now called JBIG2 (see [16] and [17]). Sanchez-Cruz et al., [15] compared seven recent chain codes, including the 3OT and, also, JBIG; after they applied Huffman coding to the chains. Their experiments gave better results than the JBIG compressor. They found that the best codes to represent binary objects was DFCCE in comparing with JBIG, if considering a threshold, no more than about 13000 in Perimeter-8. We developed our research by using DFCCE (Differential Freeman Chain Code of Eight Directions) and 3OT (Three Orthogonal Change Directions), because they were the two best codes in [15], and were also compared with JBIG standard. 3OT is composed of three symbols, and we probe here that is suitable to be handled by Arithmetic coding, better than those composed with more symbols, including DFCCE, which has eight symbols. The contribution of our work is to find that there is not a limit in the contour shapes to be represented by 3OT code, whereas in [15] it was found that DFCCE was the best, however, an evident limit in contour perimeters were reported in such a paper.

In this work, we utilized Arithmetic coding to 3OT and DFCCE chains; we found that this method has better performance than JBIG, and compress efficiently irregular

objects in a 100%, of so large contours, even larger than the obtained until now, increasing the limit in perimeter found by [15].

We want to make it clear, we are not proposing replace JBIG by our method, but to use it as a standard to compare the efficiency of chain codes.

In Section 2, we explain the method proposed, whereas in Section 3 we give the results, and in Section 4 we give conclusions and further work.

2 Applying 3OT and DFCCE Chain Codes to Bi-Level Images

With the objective to compare the results with that found in [15], in this work, we calculate the perimeter-8 of a shape, that is given by squares as resolution cells and 8-connected neighborhoods. Fig. 1 shows the 3OT and DFCCE codes to represent the contours using resolution cells, as explained in [15].

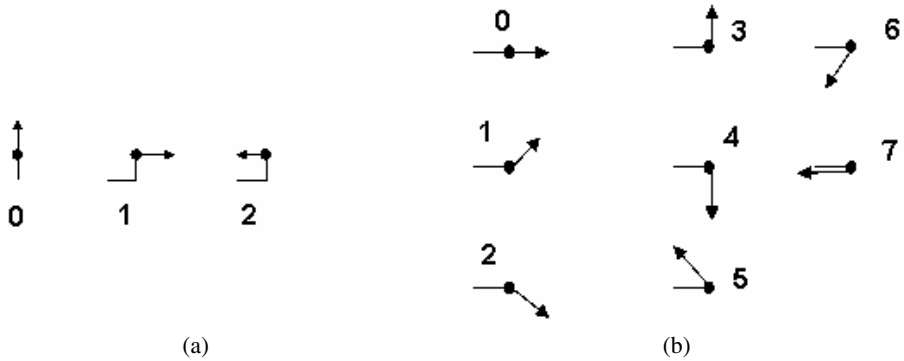


Fig. 1. The utilized codes: a) 3OT and b) DFCCE

Once the 3OT and DFCCE codes are obtained, we apply Arithmetic algorithm to the resulted chain. To compare with the proposed compression based on Huffman algorithm applied to DFCCE, we also computed such a method. Thus, we obtain an amount M_{CODE} , given in bytes. Also, we apply the JBIG compressor and obtain M_{JBIG} , in bytes too.

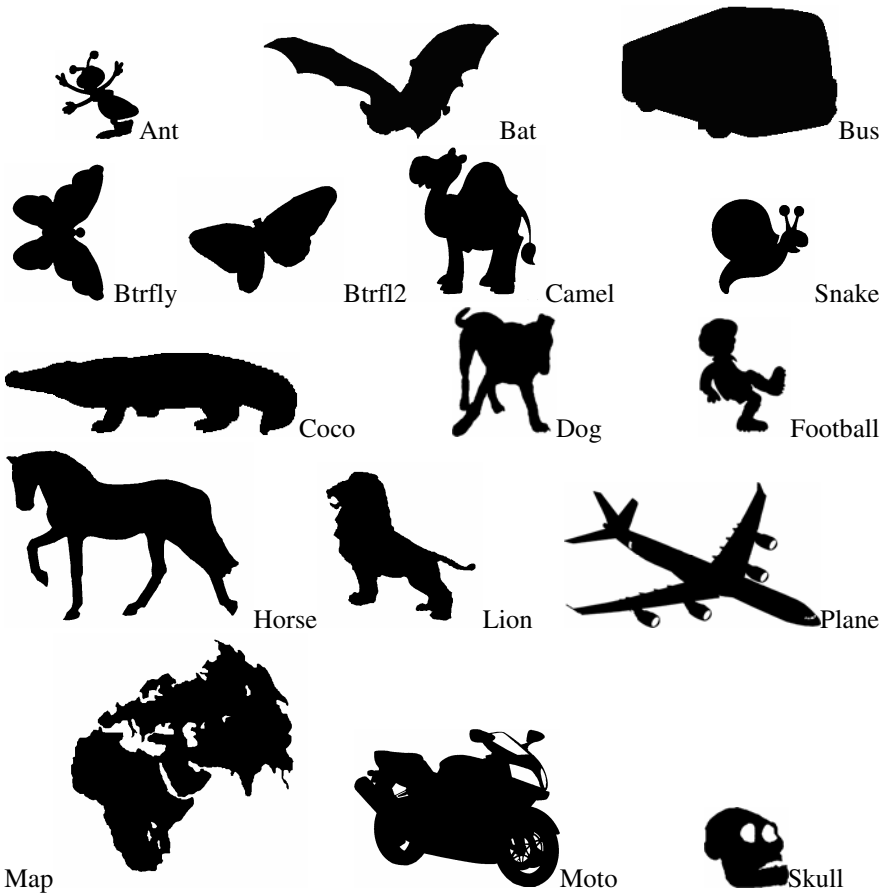
We encode and represent contour shapes for a variety of irregular sample objects, given in Fig. 2. The original size information is given in Table 1.

Table 1. Size of the bi-level images

<i>Object</i>	<i>Size</i>
Ant	235×250
Bat	551×267
Btrfy	667×822
Btrfly2	600×451
Bus	300×250
Camel	640×726

Table 1. (Continued)

<i>Object</i>	<i>Size</i>
Snail	640×633
Coco	302×87
Football	640×850
Dog	694×851
Horse	814×600
Lion	382×380
Plane	1801×1039
Map	648×648
Moto	960×738
Skull	1391×1333

**Fig. 2.** Sample object shapes represented by chain codes. The actual scales appear in Table 1.

Of course, in the bi-level image, shape-of-objects are confined in a minimal rectangle of size $M \times N$. Notice that the smallest image corresponds to the Ant object, whereas Moto's shape almost fill a typical current screen of 14 inches (with a resolution of 1024×768 pixels, for example), and Plane and Skull do not fix into such a screen.

3 Results

To analyze our results, let us define the compression efficiency, regarding JBIG.

Definition. Let $Efficiency = 1 - M_{CODE} / M_{JBIG}$ be the compression efficiency of 3OT code with regard to the JBIG standard.

In Table 2 and 3 are reported the main results of this work. The values of perimeter-8 are given, also the storage memory due to 3OT, DFCCE and JBIG and the relative efficiency of 3OT with regard to JBIG. As can be seen Arithmetic to 3OT and to DFCCE improve compression levels, and are better than both: Huff-DFCCE and JBIG.

Table 2. Length chains, given in perimeter-8, of the coded contour shapes, the storage memory in bytes, and also, Efficiency regarding JBIG of the different codes

<i>Object</i>	<i>P-8</i>	<i>JBIG</i>	M_{DFCCE} (<i>Huffman</i>)	M_{DFCCE} (<i>Arith</i>)	M_{3OT} (<i>Arith</i>)	<i>Efficiency</i> (<i>Arith-3OT</i>)	<i>Efficiency</i> (<i>Aritc-DFCCE</i>)	<i>Efficiency</i> (<i>Huffman-DFCCE</i>)
Ant	1484	398	336	311	309	0.22	0.22	0.16
Bat	1444	392	323	297	284	0.28	0.24	0.18
Btrfly	2682	694	608	532	507	0.27	0.23	0.12
Btrfl2	1473	439	328	306	283	0.36	0.30	0.25
Bus	653	205	133	129	115	0.44	0.37	0.35
Camel	3446	746	715	662	659	0.12	0.11	0.04
Snail	2557	658	548	512	502	0.24	0.22	0.17
Coco	773	230	159	154	145	0.37	0.33	0.31
Football	3482	817	728	674	702	0.14	0.18	0.11
Dog	4634	1101	1001	936	883	0.20	0.15	0.09
Horse	3679	776	783	722	680	0.12	0.07	-0.01
Lion	1577	435	356	338	322	0.26	0.22	0.18
Plane	9591	2211	2190	1957	1789	0.19	0.11	0.01
Map	4140	1031	847	793	848	0.18	0.23	0.18
Moto	5954	1391	1315	1211	1133	0.19	0.13	0.05
Skull	6861	1453	1358	1210	1298	0.11	0.17	0.07

We can see, in Fig. 3, that there exists a linear relationship between Arithmetic coding, applied to 3OT, Huffman and Arithmetic coding to DFCCE and JBIG, vs. Perimeter-8. Whereas Fig. 4 shows an exponential relationship between Efficiency and Perimeter-8. For the case of Huff-DFCCE, similar behavior appears in [15], in which Efficiency of DFCCE and Perimeter-8 were plotted, and Huffman algorithm

was applied. However, the improvement now, is that the graph plotted is farther from the zero efficiency, and is given by Arith-3OT. On the other hand, compression efficiency for 3OT can be approximated by an approximated Gaussian function:

$$\sum_{i=1}^3 a_i e^{-(x-b_i)/c_i)^2} \text{ where } a_1 = 2.2 \times 10^{10}, b_1 = -75.69, c_1 = 140.5; a_2 = 2.42, b_2 = -5693, c_2 = 4226; a_3 = 0.1603, b_3 = 6892, c_3 = 18840.$$

Observe in Fig. 4, that the trend of Huff-DFCCE suggests it will cross with the trend of JBIG, of course further than 9000 units in perimeter-8 (similar behavior was found in Sanchez-Cruz, et al, 2007 for Huff-DFCCE). However, the trend of Arith-3OT and Arith-DFCCE suggest that they never will cross with the trend of JBIG, even more, the slope of the fitted function of Arith-3OT is the smallest. This analysis allow us to say that is better to use 3OT code to represent bi-level images, whenever irregular shapes are into the images. This analysis and the trend of efficiency, suggests that for all perimeter contour coded by 3OT, in which Arithmetic coding is applied has better performance than JBIG, including for larger perimeter contour than the found in (Sanchez-Cruz, et al., 2007) in which Huffman algorithm was more effective to compress binary objects.

Of course, for each hole some extra bits are needed to represent the starting. In case of Moto shape (with the largest amount of holes) 24 holes were coded. Each code will require two starting coordinates, in the "worst case" 960 columns and 738 lines can be coded by 20 bits, multiplied by 24, gives 480bits. So, 60 extra bytes, will be required to codify the Moto shape. Obviously, not the 24 holes have the (960,738) coordinates, this amount is a "worst case", and, even though, this does not change the trends found.

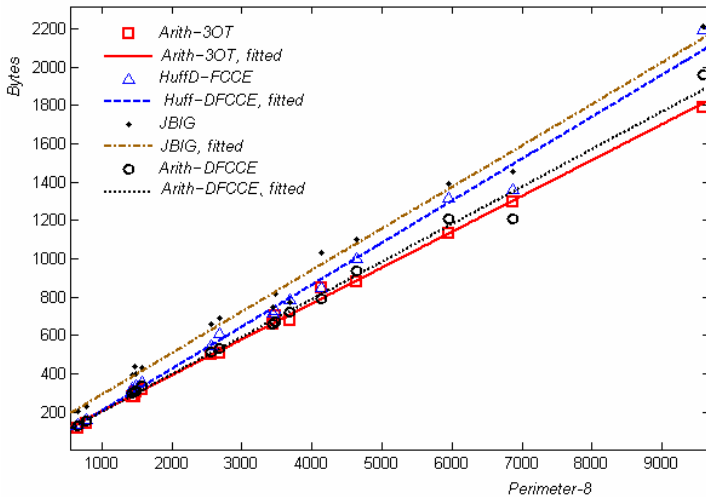


Fig. 3. Linear relationship between Arithmetic coding to 3OT, Huffman algorithm and Arithmetic coding to DFCCE and JBIG, vs. Perimeter-8

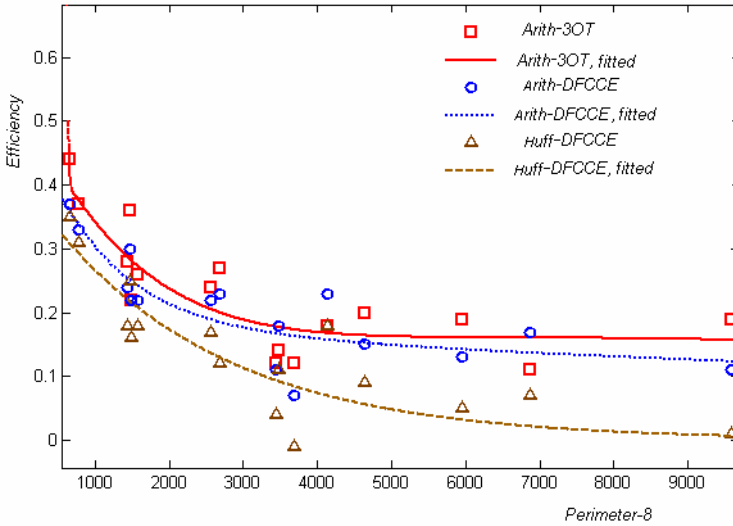


Fig. 4. Approximated functions to the obtained data

4 Conclusions and Further Work

To represent shape of binary objects, we have used the Arithmetic coding applied to different codes, we compared the results with JBIG compressor to measure their efficiency. Undoubtedly, Arithmetic applied to 3OT and DFCCE chain codes, brings better compression levels if comparing with JBIG and with Huffman applied to DFCCE, however, 81% of our sample objects were better compressed for Arithmetic to 3OT than Arithmetic to DFCCE. An interesting detailed study in this differences is suggested to be investigated between this class and the remaining 19%, to see whether some common features are present. It is evident that the whole distribution follows the showed trends, which represents an improvement of a recent work in literature. So, our main contribution is to find that 3OT code constitutes best code to represent binary images with no limitation in contour perimeters. In general, about coding scheme Freeman [18] states: they "must satisfy three objectives: (1) it must faithfully preserve the information of interest; (2) it must permit compact storage and be convenient for display; and (3) it must facilitate any required processing. The three objectives are somewhat in conflict with each other, and any code necessarily involves a compromise among them". So, we consider that 3OT has the three characteristics.

There are several methods to code a 2D object, it would also be important to compare the proposed method with other invariant algorithms in order to assess the suitability of the method in more complex scenes and real world problems.

Considering the superiority of JBIG2 over JBIG, as a future work, comparison of 3OT and DFCCE versus JBIG2 is suggested to be investigated, and also the possible application to maps, trees, text documents, and fractal images.

References

1. Mckee, J.W., Aggarwal, J.K.: Computer recognition of partial views of curved objects. *IEEE Transactions on Computers* C-26, 790–800 (1977)
2. Hoque, S., Sirlantzis, K., Fairhurst, M.C.: A New Chain-code Quantization Approach Enabling High Performance Handwriting Recognition based on Multi-Classifer Schemes, ICDAR. In: Seventh International Conference on Document Analysis and Recognition (ICDAR 2003), vol. 2, p. 834 (2003)
3. Freeman, H.: On the encoding of arbitrary geometric configurations. *IRE Transactions on Electronic Computers* EC-10, 260–268 (1961)
4. Sánchez-Cruz, H.: A Proposal Method for Corner Detection with an Orthogonal Three-direction Chain Code. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) *ACIVS 2006*. LNCS, vol. 4179, pp. 161–172. Springer, Heidelberg (2006)
5. Salem, M., Sewisy, A., Elyan, U.: A Vertex Chain Code Approach for Image Recognition. *ICGST International Journal on Graphics, Vision and Image Processing* 5(3) (2005)
6. Sánchez-Cruz, H., Rodríguez-Dagnino, R.M.: Compressing bi-level images by means of a 3-bit chain code. *Optical Engineering* 44(9), 097004 (2005)
7. Liu, Y.K., Žalik, B.: And efficient chain code with Huffman coding. *Pattern Recognition* 38(4), 553–557 (2005)
8. Liu, Y.K., Wei, W., Wanga, P.J., Žalik, B.: Compressed vertex chain codes. *Pattern Recognition* 40, 2908–2913 (2007)
9. Bribiesca, E.: A new chain code. *Pattern Recognition* 32(2), 235–251 (1999)
10. Rytter, W.: Compressed and fully compressed pattern matching in one and two dimensions. *Proc. IEEE* 88(11), 1769–1778 (2000)
11. Farach, M., Thorup, M.: String matching in Lempel-Ziv compressing strings. In: *Proc. 27th Annu. Symp. Theory Computing*, pp. 703–712 (1995)
12. Aguinaga, L.E., Neri-Calderón, R.A., Rodríguez-Dagnino, R.M.: Compression rates comparison of entropy coding for three-bit chain codes of bilevel images. *Optical Engineering* 46(8), 087007 (2007)
13. Sanchez-Cruz, H., Lopez-Cruces, M., Puga, H.: A Proposal Modification of the 3OT Chain Code. In: Thalmann, D. (ed.) *Computer Graphics and Imaging*, vol. 10, 300 pages, pp. 6–11. Acta Press (2008), A Publication of the International Association of Science and Technology for Development. ISBN: 978-0-88986-719-2
14. Huffman, M.: Lossless bilevel image compression. In: Sayood, K. (ed.) *Lossless Compression Handbook*. Academic Press, New York (2003)
15. Sanchez-Cruz, H., Bribiesca, E., Rodriguez-Dagnino, R.M.: Efficiency of chain codes to represent binary objects. *Pattern Recognition* 40(6), 1660–1674 (2007)
16. Howard, P., Kossentini, F., Martins, B., Forchhammer, S., Rucklidge, W.: The emerging JBIG2 standard. *IEEE Transactions on Circuits and Systems for Video Technology* 8(7), 838–848 (1998)
17. Ono, F., Rucklidge, W., Arps, R., Constantinescu, C.: JBIG2-the ultimate bi-level image coding standard. In: *Proceedings International Conference on Image Processing, Vancouver, BC, Canada*, vol. 1, pp. 140–143 (2000)
18. Freeman, H.: Computer Processing of Line-Drawing Images. *ACM Computing Surveys (CSUR)* 6(1), 57–97 (1974)