Digital line segment coding: A new efficient contour coding scheme

B.B. Chaudhuri, M.Tech., Ph.D., Mem.I.E.E.E., and M.K. Kundu, M.Tech.

Indexing terms: *Codes and decoding, Digital communication systems, Image processing*

Abstract: A new coding scheme is proposed for two-tone image contours. The basic idea is to detect digital line segments on the contour and code them using fixed- or variable-length codewords; the present work deals mainly with fixed-length codewords. It is demonstrated that the conventional contour run length coding by Freeman's chain code is a special case of this scheme. The data compressibility of the scheme is studied and the test results on several contours are presented. The results show that the present scheme is superior to the conventional scheme.

1 Introduction

Efficient coding for the digital transmission or storage and retrieval of two-tone images, such as printed and handwritten script, engineering drawing, fingerprints etc. is useful in image-processing problems. The coding schemes can be grouped into two classes, namely exact and approximate coding. Exact coding schemes do not introduce any distortion; i.e. from the coded data one can reconstruct the digitised original image exactly. Approximate coding methods, on the other hand, introduce distortion to the reconstructed image. The work reported here falls under the class of exact coding of two-tone images. It is assumed that the images are already digitised in a rectangular grid structure.

Huang [1] discussed three basic heuristic concepts for exact coding: skipping white, coding-only boundary points (or contour) and pattern recognition. Of these, the standard technique of coding-only boundary points is the contour run length coding (CRLC). The literature on run length codes is quite rich [2–5], and some run length coding techniques can be used for contour coding. However, Freeman's direction code [6] is almost always used for the purpose. Both four-direction and eight-direction Freeman's codes have been found to be popular. Recently Saghri and Freeman proposed a generalised direction code [7].

The concept of coding-only boundary points is used in the new coding scheme proposed in this paper. However, a pattern recognition concept is also incorporated in the scheme. The pattern recognition concept concerns the recognition of digital straight line segments on the boundary. It will be shown that the conventional CRLC is a special case of the new coding scheme, and for a contour with many digital line segments of fine angular resolution the data compression by the present scheme is better than by CRLC. The scheme is discussed in Section 2, and the code and coding algorithm are given in Section 3. The results of using the present code on some two-tone pictures are discussed in Section 4.

2 Contour run length and line segment coding

A closed digital contour may be defined as a string of pels such that if a pel is in the contour, then two, and only two, of its eight neighbouring pels are also in the contour. In the open contour the rule is violated at the end pels where only one of the eight neighbours is in the contour. Clearly, a corner sharper than 90° in the contour is not allowed by this definition.

In CRLC a pel on the contour is chosen as the initial point. From the initial point the first run begins along the direction of the next pel and continues so long as the subsequent contour pels can be traversed along the same direction; then a new run is started along a new direction. In each run the direction and the number of pels, called the run length, are noted. The process continues until the contour is traversed. Encoding may be done either by fixed-length or by variable-length codewords. All the contours of the object can be treated in a similar manner.

Each codeword consists of two subwords, one for the direction and the other for the run length. The binary representation of the direction and run length is a simple way of generating the codewords. If there are d possible directions and if l is the maximum of the run length in the object, a fixed-length code has length $b_d + b_l$, where b_d and b_l are the smallest integers greater than or equal to $\log_2 d$ and $\log_2 l$, respectively. Better compression is possible with variable-length codes, but the coding and decoding methods are expensive.

Usually, $d = 4$ or $d = 8$ is chosen in CRLC. For $d = 4$ the neighbouring directions are 90° apart, while for $d = 8$ the neighbouring directions are 45° apart. The directions and corresponding runs for a closed contour are shown in Fig. 1, where Fig. 1*c* is a representation for the crack code [12]. In Figs. 1*c* and *d* the solid line with the arrows shows how the contour may be traversed for encoding. Each arrow on the line denotes the end of a run.

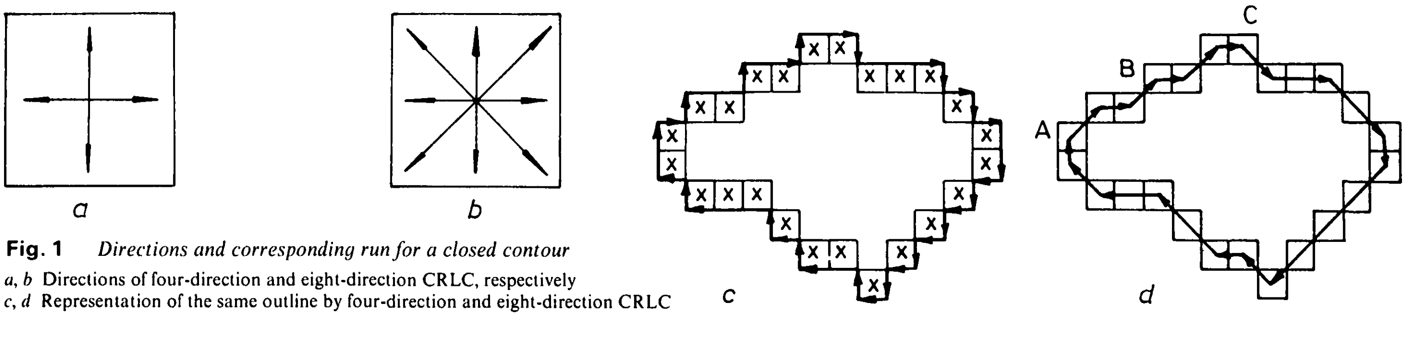

Fig. 1 *Directions and corresponding run for a closed contour*
a, b Directions of four-direction and eight-direction CRLC, respectively
c, d Representation of the same outline by four-direction and eight-direction CRLC

Paper 3206E (C2), first received 29th April 1983 and in revised form 23rd February 1984

The authors are with the Electronics & Communication Sciences Unit, Indian Statistical Institute, Calcutta 700035, India

The present scheme, called digital line segment code (DLSC), is a generalisation of CRLC. In this scheme, the digital line segments of the contour are found and coded. The concatenation of pels representing a string may be a valid digital straight line under certain conditions. The conditions stated by Freeman are [8]:

(*a*) at most, two basic directions are present in the string and these differ only by unity, modulo eight

(*b*) if there are two directions, one of them always occurs singly

(*c*) successive occurrences of the principal direction occurring singly are as uniformly spaced as possible.

It was pointed out by Pavlidis [9] that condition (*c*) is somewhat 'fuzzy'. Rosenfeld [10] defined a chord property and showed that the necessary and sufficient condition for a chain code being the chain code of a line is the chord property. Recently, Wu [11] presented an algorithm that recognises whether or not a string is the string of a digital line.

It is easy to see from Figs. 1*c* and *d* that each run denotes a straight line satisfying the above properties. Also, the number of runs in Fig. 1*d* is less than in Fig. 1*c*. This number would reduce further if we could find some 'basic shape' of string in addition to the single pels repeating along a certain direction on the contour. This task is equivalent to searching digital straight lines of resolution finer than 45° if the 'basic shape' satisfies the above conditions. The basic shape may be called the line segment (LS) unit. In Fig. 1*d* it is seen that the basic shape or LS unit of the string AB is repeated thrice between A and C of the outline. Thus, the portion AC could be represented by a single run instead of six runs as in Fig. 1*d*, and the number of bits required to code them would become less. In fact, the portion AC is a digital straight line.

It is possible to find an infinite number of LS units that satisfy the chord property. However, to code an object boundary a finite number of them can be chosen for several practical reasons. With a larger number of different strings, with a finer direction resolution, it requires a bigger subword to code the direction. Also, it may be a waste of data bits if very fine resolution is chosen when the line segments of the objects to be encoded have a coarse direction resolution. Lastly, the detection and subsequent encoding of successive line segments may be more costly and somewhat ambiguous if the direction resolution is high. We restrict ourselves to digital line segments with slopes $\pm 1/(m+1)$, $\pm(m+1)$ or $\pm m/(m+1)$, $\pm(m+1)/m$, $m \in 1^+$ with respect to the horizontal or vertical directions shown in Fig. 1*b*.

Let P and Q be the two basic directions satisfying condition (*a*) of which Q occurs singly as in condition (*b*) given previously. P, or Q, is one of the eight directions of Fig. 1*b* satisfying the two conditions. For a positive integer m, let mPQ denote m successive occurrences of pels in the direction of P followed by the occurrence of a pel along Q in a string. For the kind of line segments being considered, mPQ is the representation of an LS unit. In fact, mPQ may represent digital line segments with any of the slopes $\pm 1/(m+1)$, $\pm(m+1)$ or $\pm m/(m+1)$, $\pm(m+1)/m$ described previously. If $m = 0$, digital lines along the eight directions of Fig. 1*b* are only accounted resulting in the conventional Freeman's scheme.

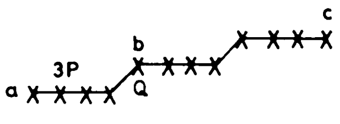

Fig. 2 *A discrete line segment*

As an example, consider Fig. 2, of which the portion *ab* constitutes an LS unit repeating thrice over *ac*. The LS unit may be represented by $3PQ$, where P denotes the horizontal direction and Q denotes a direction at $+45°$ with respect to P. Similarly, the LS unit for the digital line AC of Fig. 1*b* is PQ. The slope of the line in Fig. 2 is $+1/4$ with respect to the horizontal and that of the line of portion AC of Fig. 1*d* is 1/2 with respect to the horizontal direction.

3 DLSC and coding algorithm

For efficient coding of the digital outline the following properties of a digital line should be considered.

First, both mPQ and QmP are the representation of an LS unit of a line of the same slope, as shown in Fig. 3A.

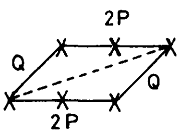

Fig. 3A *Representation of the same direction by direction units mPQ and QmP*

Hence, in a digitised contour a digital straight line may begin with a singly occurring, i.e. Q-type, or multiply occurring, P-type, direction; however, the same codeword cannot be used for coding both mPQ or QmP because in that case the contour string cannot be reproduced exactly after decoding.

Secondly, as shown in Fig. 3B, the angle $a(i)$ between the line joining the last two pels of the previous line segment and the line joining the last pel of the previous line segment with the first pel of the current line segment

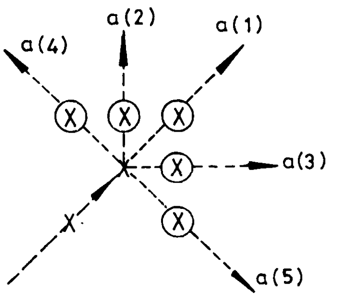

Fig. 3B *Five possible directions of starting a new line segment*

may be 0°, ±45°, ±90°. Let these angles be denoted as $a(1)$, $a(2)$, $a(3)$, $a(4)$ and $a(5)$, respectively.

Thirdly, there are two possible orientations of Q at ±45° with respect to P and vice versa, for which mPQ (or QmP) is a valid LS unit. The orientations may be denoted as $b(1)$ and $b(2)$, respectively.

Each DLSC codeword also consists of two code subwords. The first subword S_1 denotes the LS unit of the present line segment and its relative orientation with respect to the past line segment, while the second subword denotes the number of times the LS unit is repeated in the current line segment, i.e. run. Taking careful consideration of the three properties given, it can be shown that for digital lines denoted by LS unit mPQ (or QmP) $m = 0, 1, \ldots, n$ the minimum number of different subwords possible for S_1 is $[(n-1)4 + 1 + 2]5 = (4n-1)5$. For $n = 3$ this number is 55 and six bits are required to encode S_1. In general, it requires q bits for S_1, where q is the smallest integer greater than or equal to $\log_2 (4n-1)5$. A typical codebook for $n = 3$ is given in Table 1.

Table 1: Typical codebook for fixed-length DLSC; $n = 3$

Angle $a(i)$	Angle $b(k)$	Direction unit P, mPQ or QmP	Code subword
		P	0 0 0 0 0 0
$a(1)$	$b(1)$	PQ or QP	0 0 0 1 0 1
		$2PQ$	0 0 1 1 1 1
		$Q2P$	0 1 1 0 0 1
		$3PQ$	1 0 0 0 1 1
		$Q3P$	1 0 1 1 0 1
	$b(2)$	PQ or QP	0 0 1 0 1 0
		$2PQ$	0 1 0 0 0 0
		$Q2P$	0 1 1 0 1 0
		$3PQ$	1 0 0 1 0 0
		$Q3P$	1 0 1 1 1 0
		P	0 0 0 0 0 1
$a(2)$	$b(1)$	PQ or QP	0 0 0 1 1 0
		$2PQ$	0 1 0 0 0 1
		$Q2P$	0 1 1 0 1 1
		$3PQ$	1 0 0 1 0 1
		$Q3P$	1 0 1 1 1 1
	$b(2)$	PQ or QP	0 0 1 0 1 1
		$2PQ$	0 1 0 0 1 0
		$Q2P$	0 1 1 1 0 0
		$3PQ$	1 0 0 1 1 0
		$Q3P$	1 1 0 0 0 0
		P	0 0 0 0 1 0
$a(3)$	$b(1)$	PQ or QP	0 0 0 1 1 1
		$2PQ$	0 1 0 0 1 1
		$Q2P$	0 1 1 1 0 1
		$3PQ$	1 0 0 1 1 1
		$Q3P$	1 1 0 0 0 1
	$b(2)$	PQ or QP	0 0 1 1 0 0
		$2PQ$	0 1 0 1 0 0
		$Q2P$	0 1 1 1 1 0
		$3PQ$	1 0 1 0 0 0
		$Q3P$	1 1 0 0 1 0
		P	0 0 0 0 1 1
$a(4)$	$b(1)$	PQ or QP	0 0 1 0 0 0
		$2PQ$	0 1 0 1 0 1
		$Q2P$	0 1 1 1 1 1
		$3PQ$	1 0 1 0 0 1
		$Q3P$	1 1 0 0 1 1
	$b(2)$	PQ or QP	0 0 1 1 0 1
		$2PQ$	0 1 0 1 1 0
		$Q2P$	1 0 0 0 0 0
		$3PQ$	1 0 1 0 1 0
		$Q3P$	1 1 0 1 0 0
		P	0 0 0 1 0 0
$a(5)$	$b(1)$	PQ or QP	0 0 1 0 0 1
		$2PQ$	0 1 0 1 1 1
		$Q2P$	1 0 0 0 0 1
		$3PQ$	1 0 1 0 1 1
		$Q3P$	1 1 0 1 0 1
	$b(2)$	PQ or QP	0 0 1 1 1 0
		$2PQ$	0 1 1 0 0 0
		$Q2P$	1 0 0 0 1 0
		$3PQ$	1 0 1 1 0 0
		$Q3P$	1 1 0 1 1 0

The size of the second code subword S_2 depends on the size of the run, R. If the maximum value is R_{max}, a fixed length code requires r bits to represent it, where r is the smallest integer greater than or equal to $\log_2 R_{max}$.

Therefore it requires $q + r$ bits to represent a line segment whose direction unit takes a value from mPQ, $m = 0, n$. However, the average bit requirement can be reduced appreciably if a variable-length code such as Huffman code or B-code [12] is used.

Let us examine the compressibility of the present code. We define compression ratio δ_{DLSC} of a two-tone picture as

$$\delta_{DLSC} = \frac{\text{total number of pels in the picture}}{\text{number of bits required to represent the picture by DLSC}}$$

For a picture of size $M \times M$ with N different contours, let there be L_i, $i = 1, N$, line segments with direction units P or mPQ, $m = 1, n$, present on the contours. It requires a total of $\sum_{i=1}^{N} (q + r)L_i$ bits to represent them. Also, for each of the N contours, a starting pel is to be coded. If it requires K bits to code a starting pel, we have, for DLSC, the compression ration δ_{DLSC} as:

$$\delta_{DLSC} = \frac{M^2}{NK + \sum_{i=1}^{N} (q + r)L_i} \tag{1}$$

A picture is compressible if $0 < 1/\delta_{DLSC} < 1$.

To compare the compression ratio of DLSC with that of the conventional CRLC with eight directions as in Fig. 1*b*, let L_i' be the number of runs present in the ith contour. For the kind of contour defined in Section 2 it requires a minimum of 2 bits to represent the direction of each run. With everything else remaining constant, we have

$$\delta_{CRLC} = \frac{M^2}{NK + \sum_{i=1}^{N} (2 + r)L_i'} \tag{2}$$

The relative compressibility, δ_{rel}, is given by:

$$\delta_{rel} = \delta_{DLSC}/\delta_{CRLC} = \frac{NK + \sum_{i=1}^{N} (2 + r)L_i'}{NK + \sum_{i=1}^{N} (q + r)L_i} \tag{3}$$

DLSC can compress better than CRLC when

$$\sum_{i=1}^{N} L_i' \Big/ \sum_{i=1}^{N} L_i > \frac{q + r}{2 + r} \tag{4}$$

For $q = 6$ and $r = 8$ the right-hand side of eqn. 4 is 1.4, which means that DLSC compresses better than CRLC if the total number of runs for CRLC is more than 1.4 times the total number of runs for DLSC.

The coding strategy is to start with $n = 0$ and increase n by one at each iteration of the algorithm DLSC given in the following and choose the value of n for which eqn. 2 is maximum. The value of n, the codebook and coded picture may then be used for transmission, storage and retrieval.

The coding algorithm begins with choosing the initial pel, x_{ii}. For the open contour the choice is simply one of the two end pels of the contour. For the closed contour a pel is to be searched so that it denotes the beginning of an mPQ, $m = 0, n$, direction line segment. A simple way of choosing the initial point is to search for a 90° bend in the pel string. If a 90° bend is unavailable, a 45° bend may be used. Also, at the end of each run, the strategy is to search a line segment with unit mPQ (or QmP). Unit Q, i.e. $m = 0$, is allowed only when mPQ, $m > 1$, is unavailable at the current position.

Let the successive pels be denoted by x_i, x_{i+1}, x_{i+2}, etc. Then the coding algorithm for DLSC with a given $n > 1$ can be described as follows.

Algorithm DLSC

Step 1: Choose x_{ii} and code it. Denote $x_i = x_{ii}$. Define a direction d_i as the direction of x_i from x_{i-1}, where x_{i-1} is the contour pel previous to x_i in clockwise or anti-clockwise sense. (The sense is maintained throughout the execution of the algorithm).

Step 2: Find the angle $a(i)$ between d_i and the line from x_i to x_{i+1}. Check if x_i, x_{i+1} and x_{i+2} are colinear. If yes, the LS unit is either mPQ or Q. Go to step 3. Otherwise the LS unit is QmP. Go to step 5.

Step 3: If there exist contour pels x_{i+j}, $j = 0$, $n + 1$, along the same direction or if there exist pels x_{i+j}, $j < n$, along the same direction, but the line joining x_{i+j} and x_{i+j+1} is at an angle $\pm 90°$ with respect to the line joining x_{i+j-1} and x_{i+j}, the LS unit is Q. Find the run R along the direction and code the line segment using $a(i)$, P and R. Go to step 6. Otherwise go to step 4.

Step 4: Find j for which the pels x_{i+j}, $j < n + 1$, are along the same direction. Replace m by the maximum value of j. Find the angle $b(k)$ made by the line joining x_{i+j} with the direction of x_{i+j} with respect to x_{i+j-1}. Count the run R of mPQ and code the line segment with the current values of $a(i)$, $b(i)$, mPQ and R. Go to step 6.

Step 5: Find j for which the pels $x_{i=j+1}$, $j < n + 1$, are along the same direction. Replace m by the maximum value of j. Find $b(k)$ as the angle between the line joining x_{i+j} and x_{i+j+1} and the line joining x_i and x_{i+1}. Count R of QmP and code the line segment with the current values of $a(i)$, $b(k)$, QmP and R.

Step 6: If the last pel x_l of the line segment is the last pel of the contour, go to step 7. Otherwise let $x_i = x_l$. Replace d_i by the direction of line joining x_i and x_{l-1}. Go to step 2.

Step 7: Stop

Step 8: End

The case $n = 0$ is not considered in the algorithm since it denotes ordinary eight direction code and simpler algorithm can be used for the coding. The algorithm can be used repeatedly if more than one contour is present in a picture. The number of comparisons of direction required to find a direction unit mPQ or QmP is $m + 1$. If there are Rl_i direction units in the lth discrete line segment of the ith contour, the maximum number of comparisons C required by this algorithm is

$$C = \sum_{i=1}^{N} L_i Rl_i(m + 1)$$

In addition to C, some comparisons are required to find the initial point of each contour. The number of such comparisons vary from contour to contour. If the average is I_{av} then, for N contours, $I = NI_{av}$ comparisons are required. The complexity of the algorithm is therefore of the order of $O(C) + O(I)$.

4 Results and discussion

The algorithm has been tested on the digitised contours of a butterfly, a chromosome, a handwritten numeral '8' and a contour map given, respectively, in Figs. 4–7. The results have been compared with the conventional CRLC with eight directions. In CRLC two bits have been used to represent the relative direction. Two instead of three bits are used because there are only four relative directions possible in the type of contour defined in Section 2. Thus, we tried to minimise the bit requirement in CRLC and then test the efficiency of DLSC with respect to CRLC. The bit requirement to represent the run in CRLC has been kept the same as that in DLSC because R in the case of CRLC

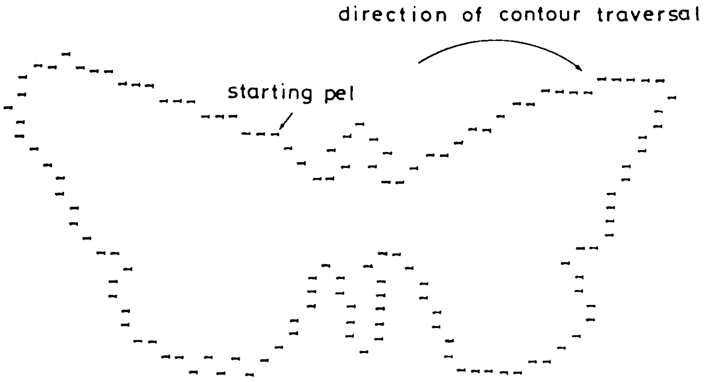

Fig. 4 *Digitised outline of a butterfly*

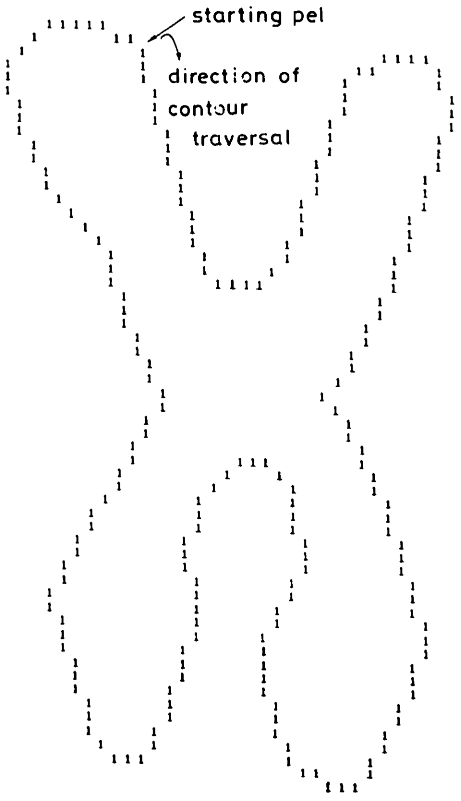

Fig. 5 *Digitised outline of a chromosome*

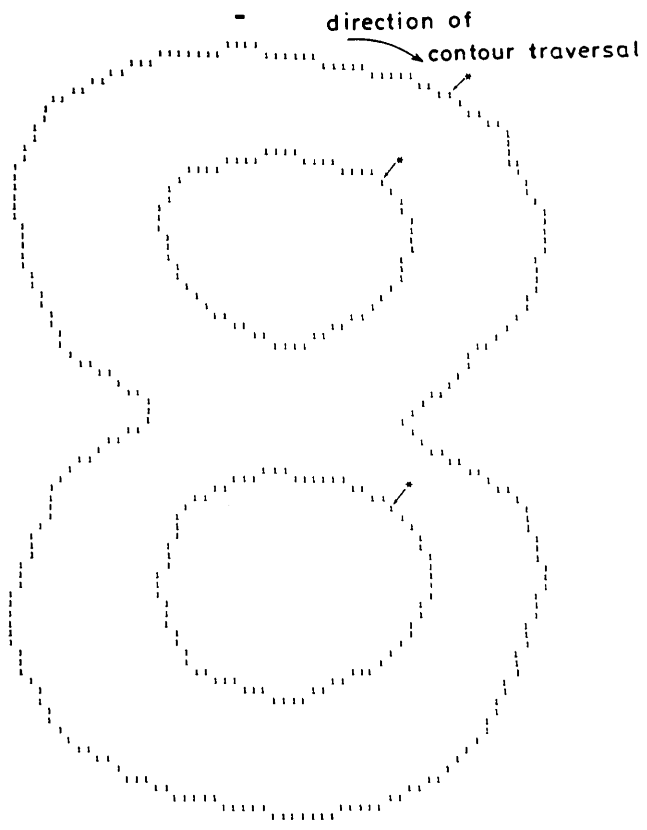

Fig. 6 *Digitised outline of a handwritten numeral '8'*
* Starting pel

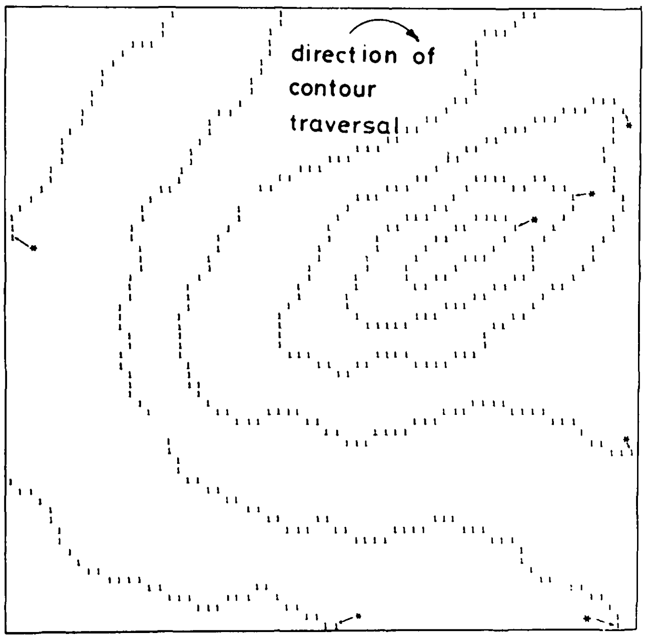

Fig. 7 *Digitised outline of a contour map*
* Starting pel

Table 2: Comparison of data compression performance for two methods

	Bits required to represent		Total data bits required in		
Figure	x_{ij}	S_2	CRLC	DLSC*	δ_{rel}, %
4	15	6	799	531	150.47
5	15	6	1175	603	194.85
6	17	7	2085	1169	179.12
7	15	6	2537	1509	168.12

* Bits required to represent S_1 in DLSC is 6 bits for all Figures

is greater than or equal to R in case of DLSC. The comparative results are given in Table 2. It is seen that DLSC with $n = 3$ performs better than CRLC in all the four Figures. In fact, the best compression is achieved for $n = 3$.

In actual implementation it is necessary to code the beginning or ending of a contour. For the codes given in Table 1, 55 out of 64 possibilities in 6-bit representation have been utilised. The bit representation of any of the rest $(64 - 55 = 9)$ of the possibilities can be used to represent the beginning of a contour. The code used in the present case is 111111.

The present scheme can be applied to shape description and recognition problems. To describe the shape it is useful to get the polygonal approximation of the contour of the object. The polygonal approximation is inherent in DLSC and hence the stored codewords can be used directly for description.

In a general contour map, two contours may cross at a certain position which has not been considered in the definition of contours given in Section 2. To code such pictures, Morrin's scheme [5] may be included in DLSC described here.

As stated earlier, further data compression is possible by using variable-length codes. To find an efficient variable-length code it is necessary to know the probability of occurrences of direction units *mPQ* or *QmP* and of the number of times a unit is repeated in a run of different contours of a set of pictures. An optimum Huffman code or a nearly optimum B_n-code can be constructed with sacrifices in coding and decoding complexity. The details of DLSC with variable-length codes will be discussed in a forthcoming paper.

The extension of the work to three-dimensional image is an interesting problem. The problem is being studied currently in the present laboratory and the useful results will be communicated in future.

5 References

1 HUANG, T.S.: 'Coding of two-tone images', *IEEE Trans.*, 1977, **COM-25,** pp. 1406–1424
2 SCHREIBER, W.F., HUANG, T.S., and TRETIAK, O.J.: 'Contour coding of images' *in* HUANG, T.S., and TRETIAK, O.J. (Eds.): 'Picture bandwidth compression' (Gordon and Beach, New York, 1972)
3 HUANG, T.S.: 'Run length coding and its extensions' *in* HUANG, T.S., and TRETIAK, O.J. (Eds.): 'Picture bandwidth compression' (Gordon and Beach, New York, 1972)
4 STERN, P.A., and HEINLEIN, W.E.: 'Facsimile coding using two dimensional run-length prediction'. Proc. Seminar on Digital communication, Zurich, 1974
5 MORRIN, T.H.: 'Chain link compression of arbitrary black white images', *Comput. Graphics & Image Process.*, 1976, **5,** pp. 172–189
6 FREEMAN, H.: 'On the encoding of arbitrary geometric configurations', *IRE Trans.*, 1961, **EC-10,** pp. 260–268
7 FREEMAN, H., and SAGHRI, J.A.: 'Generalised chain codes for planar curves'. Fourth International Joint Conference on Pattern recognition, Kyoto, Japan, 1978
8 FREEMAN, H.: 'Boundary encoding and processing' *in* LIPKIN, B.S., and ROSENFELD, A. (Eds.): 'Picture processing and psychopictories' (Academic Press, New York, 1970), pp. 241–263
9 PAVLIDIS, T.: 'Structural pattern recognition' (Springer-Verlag, New York, 1977)
10 ROSENFELD, A.: 'Digital straight line segment', *IEEE Trans.*, 1974, **C-23,** pp. 1264–1269
11 WU, L.D.: 'On the chain code of a line', *ibid.*, 1982, **PAMI-4,** pp. 347–353
12 ROSENFELD, A., and KAK, A.C.: 'Digital picture processing' (Academic Press, New York, 1982)

Dr. Chaudhuri received his B.Sc., B.Tech. and M.Tech. degrees from Calcutta University in 1969, 1972 and 1974, respectively, and his Ph.D. degree from the Indian Institute of Technology, Kanpur, in 1980. In 1978 he joined the Indian Statistical Institute as a faculty member where he is currently working as an Associate Professor. He was awarded the Leverhulme Visiting Fellowship in 1981–82 to work at Queen's University, Belfast. His area of research interest includes pattern recognition, optical communication, image processing and artificial intelligence. He has published nearly 45 research papers and coauthored in four edited books. A fellow of the Optical Society of India and a member of the IEEE, he is a referee to some international journals.

M.K. Kundu received his B.Sc., B.Tech. and M.Tech. degrees from the University of Calcutta in 1969, 1972 and 1974, respectively. He worked as a UGC research fellow in the department of Radio Physics and Electronics during 1975–76. In 1976 he joined the research and development division of Tata Iron and Steel Company as an assistant engineer (research). He worked in the field of Instrumentation and process automation of iron and steel making. In 1982 he joined the Indian Statistical Institute as a computer engineer. His current research interests include image processing, computer vision, pattern recognition and microprocessor-based instrumentation. He is a member of the Indian Unit of International Association for Pattern Recognition.