COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 30, 269–278 (1985)

A Generalised Digital Contour Coding Scheme

M. K. KUNDU, B. B. CHAUDHURI, AND D. DUTTA MAJUMDER

Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta 700 035, India

Received August 31, 1983; revised May 15, 1984; accepted January 22, 1985

A generalised 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 length or variable length codewords. It can be shown that the conventional contour run length coding using 8 or 4 direction Freeman's chain code is a special case of the present scheme. The data compressibility is tested on several closed contours and their noisy degraded versions. The effect of gradual increases of noise level on the relative compressibility with respect to the conventional method is also studied. Extension of the work has also been proposed. © 1985 Academic Press, Inc.

I. INTRODUCTION

Contour run length coding (CRLC) is one of the popular error-free coding techniques for two-tone image data compression [1]. To this end it is convenient to represent the image contour pels in the form of a string using Freeman's direction codes [2]. Either 4-direction or 8-direction Freeman chain codes can be used, but the 8-direction codes are more popular. A full-fledged scheme for contour tracing and CRLC scheme has been reported by Morrin [3].

The present work is a generalisation of CRLC. It is easy to see that the algorithm for conventional CRLC with 4 or 8-direction codes looks for straight line segments of the contour with resolution 90° or 45° and code them using fixed length or more efficient variable length codewords. It is shown that the algorithm may be generalised to include digital straight line segments with finer resolution as well. The different digital straight line segments can be identified either from the properties given by Freeman [4] and Rosenfeld [5, 6] or from the algorithm given by Brons [7] and Wu [8]. Although the coding technique is more expensive, better data compression is possible by this scheme. The scheme is equally applicable to raw digital outline or the polygonal approximation of a contour in least mean squared error sense.

The scheme, named as digital line segment coding (DLSC) is explained in Section II. In Section III the results of using DLSC to some two-tone image contours are presented. A few of the contours are subject to random noise and the results of using DLSC to them are given.

II. DLSC SCHEME

If represented in the 8-direction Freeman chain code an arc or a string of such code is a valid digital straight line segment under certain conditions. The first two of the three conditions stated by Freeman [4] are

(1) At most two basic directions are present in the string and these differ only by unity, modulo eight.

(2) If there are two directions, one of them always occurs singly.

Rosenfeld [5] defined a chord property and proved that the necessary and sufficient condition for a chain code being the chain code of a line is the chord property.

Our aim is to detect the different line segments on a contour and code them unambiguously. However, it is not possible to look for lines with any degree of angular resolution because of its unlimited searching expenses and code size. We shall restrict ourselves to digital line segments with slopes $\pm 1/(m+1)$, $\pm(m+1)$, $\pm m/(m+1)$, $\pm(m+1)/m$ in digital grid where m is a positive integer.

Let P and Q be the two basic directions satisfying condition (1) of which Q occurs singly as in condition (2) given above. For a positive integer m, let mPQ denote m successive occurences of P followed by the occurence Q in a string. For the kind of line segments being considered, mPQ represents a smallest string that repeats along the line segment. We call mPQ as the representation of a line segment (LS) unit. Clearly, QmP is also the representation of the same LS unit. Also, when $m = 0$, the distinction between P and Q is meaningless and either P or Q represents an LS unit. No other form of LS unit is allowed in the present case although other valid LS units are possible. Furthermore, let us assume for convenience that each valid line segment consists of integer multiples of a LS unit only.

DEFINITION 1. A closed contour is a finite 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. On the other hand, any open contour is a finite string of pels satisfying the above condition except only at two pels, each of which has only one neighbour pel in the contour.

Clearly, a corner sharper than 90° in the conventional 8-direction line segment sense is not allowed by the definition. Also, the contours having nodes and branches are not accounted for simplicity. Relaxation of these constraints will be discussed in Section III.

Each codeword consists of three code subwords namely, subword for absolute or relative direction of (i.e., the direction of the present line segment with respect to the immediate past) line segment, subword for the LS unit spanning the line segment, and subword for the number of times the LS unit is repeated along the line segment (i.e., run length). Let these subwords be denoted by A, B, and C, respectively.

The number of relative directions for subword A can be limited to five. The matter is explained in Fig. 1a, where u and v denote the last two pels of the previous line segment. The current line segment may start from v along one of the five possible directions $a(1)$, $a(2)$, $a(3)$, $a(4)$, and $a(5)$ making angles 0°, +45°, +90°, −45°, and −90°, respectively, to the uv direction. For $m = 0$, $a(1)$ cannot occur. For subword B, the LS unit of the current line segment is to be coded. If there are mPQ, $m = 0, 1, \ldots, n$ possible LS units of which each unit mPQ, $m = 2, 3, \ldots, n$ has two possible variations mPQ and QmP as shown in Fig. 1b, then the total number of possibilities to be encoded is $2n$. In addition, the direction of Q with respect to P or vice versa is to be encoded. Since the directions of P and Q differ by unity, modulo eight, there are two possibilities—one in +45° and the other in −45° as shown in Fig. 1c. It is to be noted that the subword B is not necessary for CRLC (when $m = 0$), since only LS unit P is used to denote digital lines. For subword C the number of possibilities to be encoded is equal to the length of the longest run in the contour.

The coding strategy is as follows. Start with $n = 0$ and code the outline. Next, increase n by 1 in stages and at each stage code the outline. Finally, choose the value

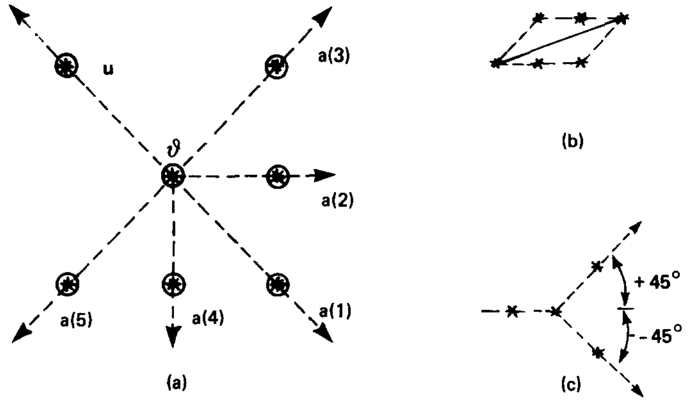

FIG. 1. (a) Possible directions of starting a new line segment unit. (b) Representation of a line segment by mPQ or QmP unit (here $m = 2$). (c) Possible relative directions of Q with respect to P.

of n for which the total bit requirement is minimum and store the code book as well as the coded outline.

To proceed further one should specify the kind of codebook to be used. At first, the simple fixed length code is discussed below.

For better compression using fixed length codewords let us consider subwords A and B together. For A there are 5 possibilities to be encoded. For B it should be noted that $mPQ = QmP, m = 0, 1$. Also, for $m = 0$, the direction of Q with respect to P is meaningless. Considering the redundancy, we can easily see that the total number of possibilities to be encoded is $[(n-1)4 + 1 + 2]5 = (4n-1)5$ for $n \geq 1$. Simple binary code requires q_n bits, where q_n is the smallest integer greater than or equal to $\log_2(4n-1)5$. The length of code subword C is r_n, where r_n is the smallest integer greater than or equal to $\log_2 R_{\max}$, where $R_{\max}$ denote the maximum possible run length in a contour.

If there are L_{ni} possible line segments in the ith outline and if K bits are required to represent the beginning of each outline then the total number of bits required is

$$B_n = \sum_{i=1}^{N} \left[K + (q_n + r_n) L_{ni} \right] \cdots, \tag{1}$$

where N is the number of outlines in a picture frame. The strategy is to choose n for which B_n is minimum. Clearly, the scheme is at least as compressible as the CRLC scheme.

We define the compression ratio of DLSC scheme as

$$\delta_{\text{DLSC}} = \frac{\text{Total no. of pels in the picture}}{\text{No. of bits required to represent picture in DLSC}}$$

If the two tone picture matix is of size $M \times M$ then

$$\delta_{\text{DLSC}} = \frac{M^2}{\sum_{i=1}^{N} \left[K + (q_n + r_n) L_i \right]} \cdots. \tag{2}$$

Similarly, for conventional CRLC scheme

$$\delta_{\text{CRLC}} = \frac{M^2}{\sum_{i=1}^{N}\left[K + (p_n + r_n)L'_i\right]} \cdots, \tag{3}$$

where L'_i is the number of runs present in the ith contour and p_n is the number of bits required to represent the different directions of the Freeman chain code. The relative compression ratio of DLSC in comparison to CRLC is defined as

$$\delta_{\text{rel}} = \delta_{\text{DLSC}}/\delta_{\text{CRLC}} = \frac{NK + \sum_{i=1}^{N}(p_n + r_n)L'_i}{NK + \sum_{i=1}^{N}(q_n + r_n)L_i} \cdots. \tag{4}$$

There are different types of variable length codes. The basic idea of all these codes is to assign smaller codewords for more frequent events. The Huffman code [9] is one of the most efficient among the variable length codes where the average codeword size is found from the entropy. Let $P(a(i), m(j), b(k), l)$ denote the probability of a line segment starting at $a(i)$ relative direction having $m(j)$th among the $2n$ possible direction units with $b(k)$th relative direction of Q with respect to P (or vice versa) and run length l. Then the entropy H_n of line segments in the contour is

$$H_n = -\Sigma p(a(i), m(j), b(k), l)\log_2 p(a(i), m(j), b(k), l) \cdots, \tag{5}$$

where the summation extends over all the possibilities of $a(i)$, $m(j)$, $b(k)$, and l. The average codeword D_n of a line segment using Huffman code is limited by

$$H_n \leq D_n \leq H_n + 1 \cdots. \tag{6}$$

Here again the coding strategy is to choose n for which $D_n L_n$ is minimum, where L_n the number of line segments in the contour using LS units admissible by $m = 0, 1, \ldots, n$.

In general, the compressibility by Huffman code is better than that by ordinary fixed length binary code. But if the plot of $p(a(i), m(j), b(k), l)$ is flat over i, j, k, and l, the fixed length code is nearly as efficient as Huffman code. In such case, it is useful to construct the codebook with those variables for which the probability plot is not flat. The rest of the variables may be coded using fixed length coding scheme. For example, here the Huffman code may be used only for the run length l while $a(i)$, $m(j)$, and $b(k)$ may be fixed-length coded. Such hybrid techniques has the advantage of smaller codebook size without appreciable sacrifice in the data compression.

III. RESULTS AND DISCUSSION

The present scheme is tested on the digitised contours of a butterfly, a chromosome a contour map, a handwritten numeral 8, and the outlines of three lakes given, respectively, in Figs. (2–5) and Figs. (6–8). Both fixed length and variable

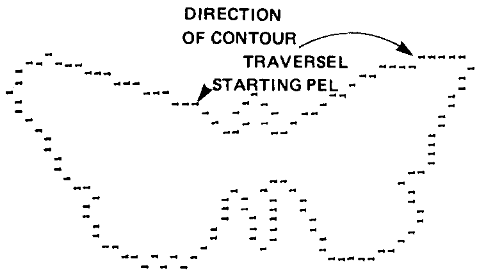

FIG. 2. Digitised outline of a butterfly.

length binary coding scheme as used to compare the data bits required in each case. In the fixed length scheme, the subword for run length is such that maximum run along any direction may be represented. For contours of Figs. (2–8) it is found that maximum run length is 9. Hence 4 bits are allotted for subword C. For $n = 0$, two bits are used to represent the relative direction of the runs. This is so because according to Definition 1 four relative directions at $\pm 45°$ and $\pm 90°$ are only possible for the runs. For variable length scheme it is found that the probability plot of $p(a(i), m(j), b(k), l)$ is quite flat. On the other hand, the probability plot of $p(l)$ is nearly exponentially decaying with l. Hence the run length l has been coded using Huffman's scheme while $a(i), m(j), b(k)$ use fixed length subwords. Each figure is treated separately for coding l. However, it is found that the Huffman subword is same for all the figures.

The total number of bits required to represent the contours are plotted against n, for fixed length and variable length coding scheme. The plots are given in Figs. 9 and 10, respectively. It is seen that the minimum occurs at $n = 3$ for all the contours in case of fixed length coding scheme. In the variable length coding scheme, however, the minimum occurs at $n = 1$ for the butterfly contour and at $n = 0$ for lake I. For

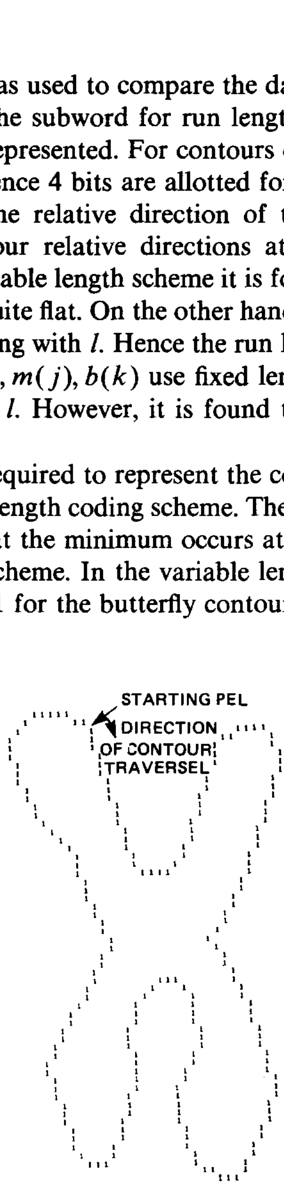

FIG. 3. Digitised outline of a chromosome.

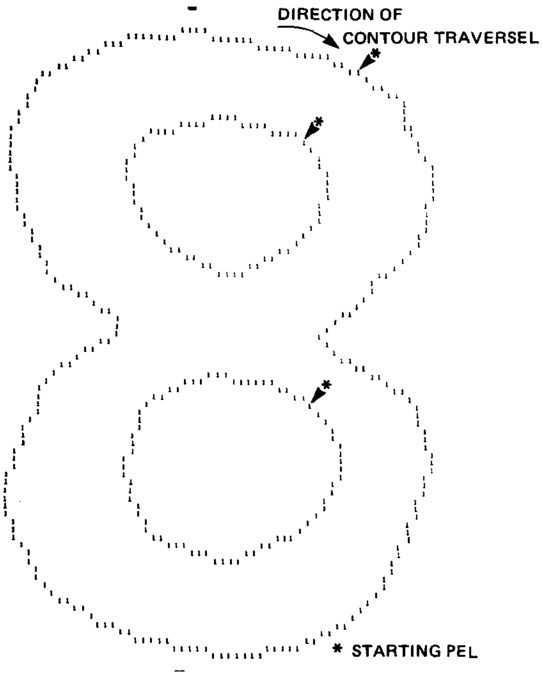

FIG. 4. Digitised outline of a hand-written numeral 8.

the other contours the minimum is at $n = 3$. Each minimum of Fig. 9 is less than the corresponding minimum of Fig. 10.

The efficiency of the generalised scheme has been tested on noisy data as well. The procedure of generating noisy outline is similar to that given in [11]. The procedure is briefly described below.

Let X_i be any arbitrary point on the boundary whose adjacent vertices are X_{i-1} and X_{i+1}. Let the centroid of the triangle $X_{i-1} X_i X_{i+1}$ by Y_i. Then due to noise

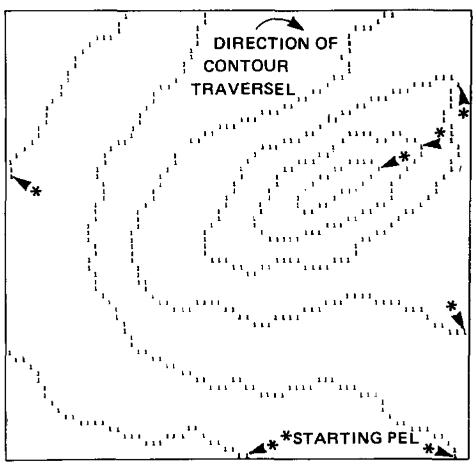

FIG. 5. Digitised outline of a contour map.

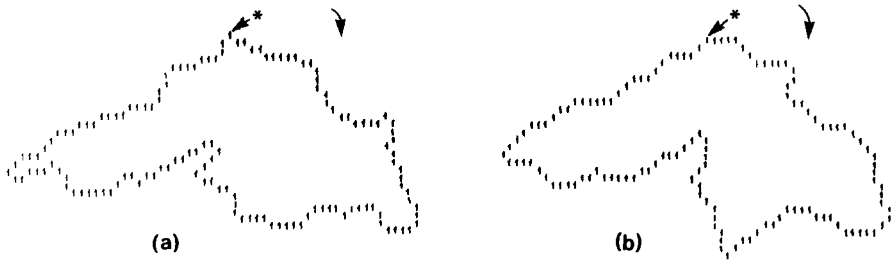

FIG. 6. (a) Digitised outline of the lake (I). (b) The same outline when noisily degraded ($\sigma^2 = 6.25$).

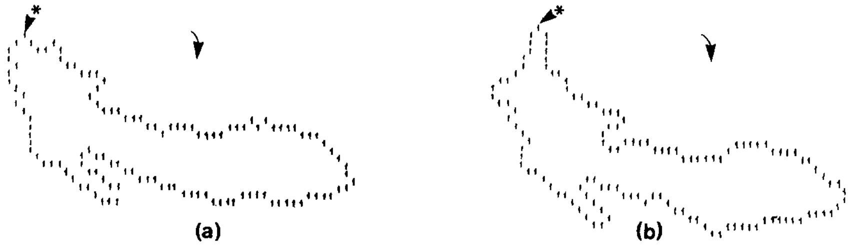

FIG. 7. (a) Digitised outline of the lake (II). (b) The same outline when noisily degraded ($\sigma^2 = 6.25$).

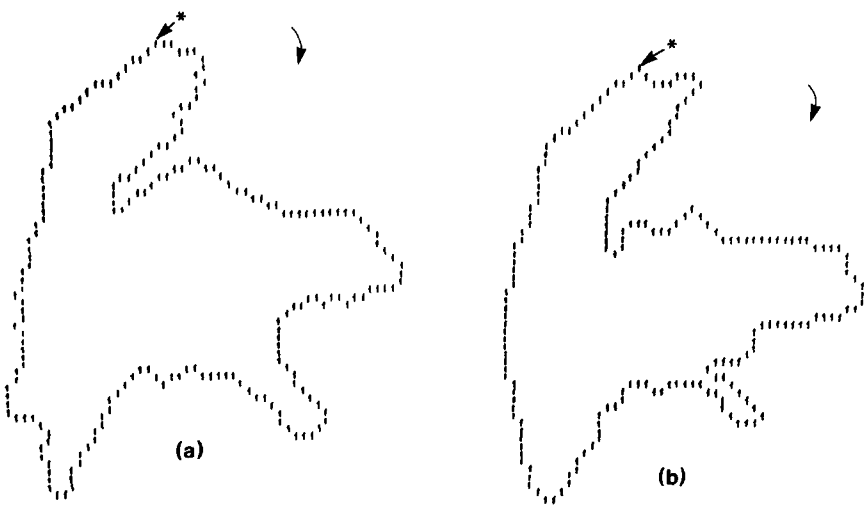

FIG. 8. (a) Digitised outline of the lake (III). (b) The same outline when noisily degraded ($\sigma^2 = 6.25$).

perturbation the point X_i is displaced to a new position Z_i, which is on the line joining X_i and Y_i, where

$$Z_i = X_i + \lambda(Y_i - X_i)$$

and λ is a guassian random variable of mean zero and variance σ^2. When $0 < \lambda < 1$

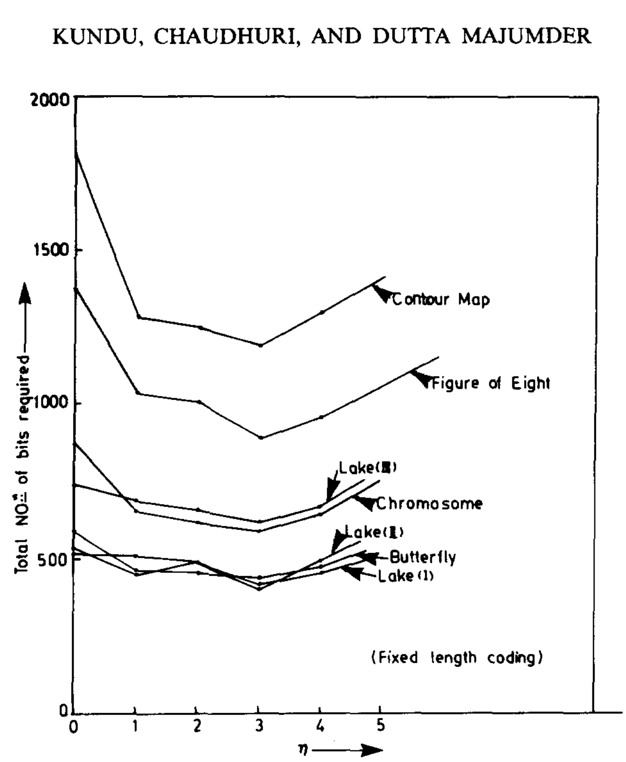

FIG. 9. Bit requirement for representing Figs. (2–8) using fixed length code.

the point Z_i lies on line between X_i and Y_i. Whenever λ is negative or greater than unity, the point lies on the infinite line joining X_i and Y_i but not between them.

For a given σ^2 the random purturbation λ was made on different evenly spaced points of the contour. The whole procedure was repeated for different values of σ^2 to generate noisy boundaries at different noise levels. This procedure ensures a global and local deformation of the boundary for moderate to fairly large noise level. Some ideal outlines and their noisy versions are shown in Figs. 6a–8a and Figs. 6b–8b, respectively. The outlines and their noisy versions for different σ^2 were subject to CRLC and DLSC. The results are plotted in Fig. 11 as δ_{rel} against σ. It is seen for all the outlines, that at first δ_{rel} decreases rapidly with σ. The rate of decrease is less for moderate σ, even it may be negative. This is apparently because excessive noise changes the outline and its line segment statistics appreciably resulting, sometimes, in the improvement also.

A discussion of contour branching is in order. The roots of the branching may be found as in Morrin's scheme [3]. Taking each root as starting point, all contours may be traversed and coded so that no contour is encountered twice.

For digital contour with pels having more than two neighbours, an extra bit may be used to indicate them. Such extra bit may be necessary for branched contours, especially for coding the portion of the contour near the root.

The direct extension of the work to 3-dimensional contour can be done in a tomographic fashion, i.e., taking 2-dimensional slices of the 3-dimensional contour. For better compression, however, it may be useful to encode digital planer surfaces. The problem is being studied currently in the present laboratory and the useful results will be communicated in future.

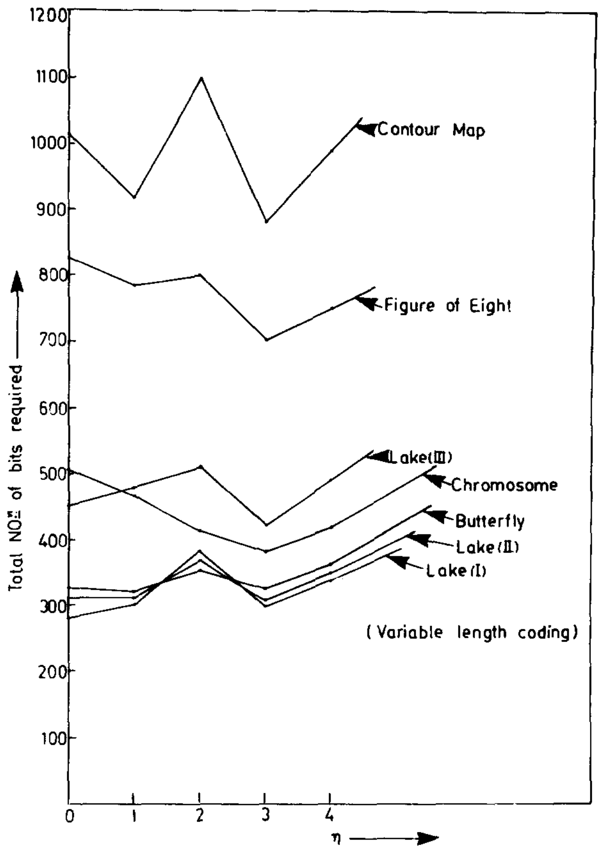

FIG. 10. Bit requirement for representing Figs. (2–8) using variable length code.

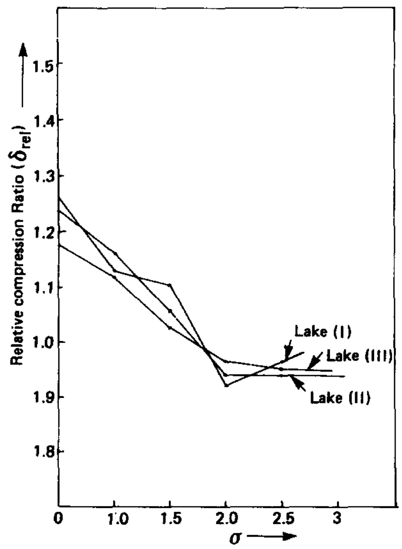

FIG. 11. Variation of δ_{rel} with σ for Figs. (6–8).

ACKNOWLEDGMENTS

The authors wish to thank Mr. S. Chakraborty and J. Gupta for preparing the manuscript.

REFERENCES

1. T. S. Huang, Coding of two tone images, *IEEE Trans. Commun.* **COM-25**, 1977, 1406–1424.
2. H. Freeman, On the encoding of arbitrary geometric configuration, *IRE Trans. Electron. Comput.* **EC-10**, 1961, 260–268.
3. T. H. Morrin, Chain link compression of arbitrary black-white images, *Comput. Graphics Image Process.* **5**, 1976, 172–189.
4. H. Freeman, Boundary encoding and processing, in *Picture Processing and Psychopictories* (B. S. Lipkin and A. Rosenfeld, Eds.), pp. 241–263, Academic Press, New York, 1970.
5. A. Rosenfeld, Digital straight line segment, *IEEE Trans. Comput.* **C-23**, 1974, 1264–1269.
6. A. Rosenfeld and C. E. Kim, How digital computer can tell whether a line is straight, *Amer. Math. Monthly* **89**, 1982, 230–235.
7. R. Brons, Linguistic methods for description of a straight line on a grid, *Comput. Graphics Image Process.* **2**, 1974, 48–62.
8. L. D. Wu, On the chain code of a line, *IEEE Trans. Pattern Anal. Mach. Intell.* **4**, 1982, 347–353.
9. D. A. Huffman, A method for the construction of minimum redundancy codes, *Proc. IEEE* **40**, 1952, 1098–1101.
10. R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, London, 1977.
11. R. L. Kashyap and B. J. Oommen, *A Geometrical Approach to Polygonal Dissimilarity and Classification of Closed Boundaries*, School of Computer Science SCS-TR-18, Jan. 1983, Carlton University, Ottawa, Canada.