Design of Perimeter Estimators for Digitized Planar Shapes

JACK KOPLOWITZ, SENIOR MEMBER, IEEE, AND ALFRED M. BRUCKSTEIN

***Abstract*—Measurement of perimeters of planar shapes from their digitized images is an important task of computer vision systems. A general methodology for the design of simple and accurate perimeter estimation algorithms is described. It is based on minimizing the maximum estimation error for digitized straight edges over all orientations. Two new perimeter estimators are derived and their performance is tested on digitized circles using computer simulations. The experimental results may be used to predict the performance of the algorithms on shapes with arbitrary contours of continuous curvature. The simulations also show that fast and accurate perimeter estimation is possible, even for objects that are small relative to pixel size.**

***Index Terms*—Binary images, digitized planar shapes, estimation of perimeter, line drawings, perimeter measurement of quantized objects.**

I. Introduction

Measurement of areas, perimeters, centroids, directional diameters, and other shape-related parameters is an important task of industrial computerized vision systems. Such systems perform the shape analysis on digitized images resulting from sampling of planar shapes of interest on a rectangular or square grid of picture elements, pixels [1]-[4]. The sampling grid is determined by the particular image sensing array and its resolution is generally beyond the control of the designer. If the digitization resolution is high, i.e., the pixel size is small compared to the details of objects of interest, the digitized representation of their shapes is accurate and so will be the measurements of various shape parameters. However, it is also important to have measurement procedures that yield accurate and consistent estimates of shape parameters for objects that are small compared to the pixel size as well.

For simplicity, we assume that objects whose shapes are to be measured are placed on a contrasting background and their outlines are smooth curves. The usual point sampling image-digitization procedure is assumed, i.e., the objects are projected on the square grid of pixels (corresponding to the sensor array), generating a binary image according to whether the pixel center belongs to the object or not [2], [4], [5]. The resulting binary image is the input for various shape measurement algorithms. In this paper, we discuss a method for the design of algorithms to evaluate the perimeter of the original 2-D shape from its binary image. Note that the contour of objects in their digitized representations is a stepwise boundary comprising the horizontal and vertical links corresponding to pixel outlines (Fig. 1). This contour may be represented using a four-directional chain code [6], [7], yielding what is usually called the "crack code" of the digitized object boundary [1]. The length of this boundary is considerably longer than the perimeter of the original 2-D shape in most cases. The problem of finding efficient and accurate perimeter estimation algorithms therefore arises naturally and has been addressed by several researchers.

The approaches to perimeter estimation can be based either on some contour reconstruction method and subsequent measurement of the length of the reconstructed boundary or on counting the links of the "crack code" which all have the same length (equal to pixel size) and performing adjustments (say, based on the number of corners) that have the equivalent effect of smoothing the stepwise boundary. A contour reconstruction approach first produces an approximation of the true object boundary, for example, by determining straight edges of maximal length that are consistent with the digitized image. The perimeter estimate is then defined as the length of the resulting, in this case polygonal, approximation of the boundary; see, e.g., [9], [5]. The second type of perimeter estimation algorithms, considerably simpler and faster, are based on counting the links or the grid points encountered on traversing the perimeter. If the perimeter estimate is chosen to simply equal the number of links in the crack code times the pixel size, large errors would be incurred since, for example a convex 2-D shape will always yield a boundary length equal to the perimeter of its extent rectangle [see Fig. 1(b)]. To alleviate this problem, one can multiply the length of the digitized boundary by a factor that would yield zero expected error between the true and estimated length of straight boundaries. The average is taken over straight edge boundaries with uniformly distributed orientation. The resulting perimeter estimator, although unbiased, has a large variability in the induced estimation error. To reduce this variability, Proffitt and Rosen [10] proposed a perimeter estimation procedure which uses two parameters corresponding to the

Manuscript received July 6, 1987; revised March 17, 1988. Recommended for acceptance by O. Faugeras. The work of J. Koplowitz was supported in part by a Lady Davis Fellowship while he was on leave at the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa, Israel.

J. Koplowitz is with the Department of Electrical and Computer Engineering, Clarkson University, Potsdam, NY 13676.

A. M. Bruckstein is with the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel.

IEEE Log Number 8926693.

0162-8828/89/0600-0611$01.00 © 1989 IEEE

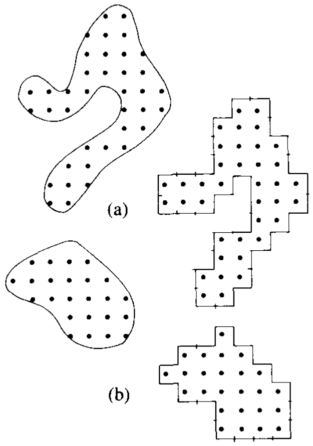


Fig. 1. Digitization by point sampling.

quantities immediately in evidence, the number of links and of corner points in the crack code. Clearly, corners contribute to overestimating the length of the boundary; therefore, the contribution to the perimeter of links meeting at corner points should be made less than a pixel length. This suggested that the perimeter should be measured using a formula of the form

$$\hat{L} = S_l(\#\text{links}) - S_c(\#\text{corners}) = \psi_n N_n + \psi_c N_c \quad (1.1)$$

where N_n is the number of noncorner points and N_c is the number of corner points in the chain code of the boundary. The weighting factors S_l, S_c or, equivalently, ψ_n and ψ_c, should be determined to meet some criteria, say, unbiasedness and minimal errors for straight lines of uniformly distributed directions.

Proffitt and Rosen show that for a straight line at arbitrary angle θ, the counts N_n and N_c can easily be determined. Neglecting end effects, a line of length L has digitized length $L \cos \theta + L \sin \theta$, the sum of its vertical and horizontal increments. For an angle $0 \leq \theta \leq \pi/4$, the number of corner points is just twice the number of vertical links. Thus, per unit length of line, we have

$$\begin{cases} \eta_n(\theta) + \eta_c(\theta) = \cos \theta + \sin \theta \\ \eta_c(\theta) = 2 \sin \theta \end{cases} \quad (1.2)$$

where $\eta_n(\theta)$ and $\eta_c(\theta)$ denote the density of noncorner and corner points per unit length of a straight edge of orientation θ, assuming a unit grid size. The coefficients ψ_n and ψ_c (or $S_l = \psi_n$, $S_c = \psi_n - \psi_c$) were then determined so as to give zero average error and minimal mean-square error (MSE) for straight lines uniformly distributed over all orientations. The optimal coefficients (in the MSE sense) are $\psi_n = \pi(\sqrt{2} + 1)/8$ and $\psi_c = \pi(\sqrt{2} + 2)/8$, yielding an error standard deviation of 2.3 percent. The same performance can also be obtained by cutting corners, i.e., replacing adjacent perpendicular links with a diagonal link to transform the four-directional crack code into an eight-directional chain code. Multiplying the digitized perimeter by the factor $(\sqrt{2} - 1)8/\pi$ then yields zero expected error [5], [11]-[13]. Note that the corner cutting implied by passing to the eight-directional chain code performs an effective smoothing of the digitized boundary.

In [14], and independently in [15], yet another simple perimeter estimation procedure was proposed that introduces correction terms for runs of links having identical direction, provided they correspond to simple concavities in the digitized boundary as depicted in Fig. 2. The correction terms implicitly define the perimeter as the length of a polygonal approximation of the discrete outline. Note that this approach too smoothes out concavities of the digitized object outline. The paper of Wechsler [15] reports experimental tests on the performance of the proposed algorithm on a set of examples, obtaining estimation errors between -1 and $+3$ percent on circles with diameters of at least 20 times the pixel size and on rectangles and triangles of comparable sizes.

In this paper, we discuss a general methodology for designing accurate and simple perimeter estimation procedures. The systematic way of deriving a perimeter estimation algorithm is based on analyzing digitizations of straight edges with arbitrary orientations. As in the procedure of [10], we wish to obtain simple length estimators which are uniformly good on lines of all orientations. The perimeter estimators should not be based on good orientation estimators for long straight lines. Rather, the estimators should be local in nature in order to also accommodate possible high curvature portions of the object boundaries.

Although it may not be evident which additional local properties may be helpful, the classification of points into two classes, corner and noncorner, can be extended to many classes. Consider classification procedures which depend only on a local neighborhood containing the point. The ergodic theorem [16] states that the frequency of occurrence of a particular class on a single arbitrarily long line equals the probability of that class for a random line segment with the same slope and uniformly distributed intercept. If the boundary curve can be assumed to be straight over the classification distance, i.e., the span of the local neighborhood, then the perimeter estimator performance is determined solely by the distribution of the boundary tangent. Thus, by the ergodic theorem, the expected performance is the same as that of a set of arbitrarily long lines having the same tangent distribution.

After presenting the general methodology for perimeter estimator design, we extend the Proffitt-Rosen procedure to derive an estimator based on a more refined classification of the points in the chain code of the boundary with only a slight increase in the classification distance. This method achieves a maximal error of 0.6 percent for straight edges, and its performance on randomly thrown circles of even very small radii is excellent. Subsequently, we propose and analyze a perimeter estimation procedure which may be regarded as a generalized and improved version of the method presented in [15]. We show that the two methods discussed are, in fact, very

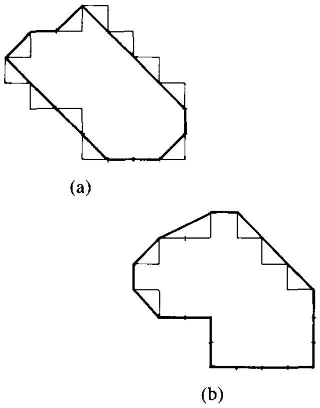


Fig. 2. Smoothing edges by (a) passing to an eight-directional chain code and (b) closing simple concavities (see [15]).

similar despite the differences at a superficial consideration. They both may be regarded as facets of the same general perimeter estimation process.

The paper is organized as follows. The next section introduces a general methodology for deriving accurate perimeter estimators. Various factors affecting the performance of the estimations are discussed. It is argued that for the estimator to be insensitive to orientation, the best criterion to minimize is the maximal percentage error for straight edges of all orientations. Using this evaluation criterion, we derive, in Section III, two new, simple, and accurate perimeter estimation methods and test their performance on randomly thrown circles of varying radii.

The last section discusses the results of Section III, showing that they can be used to predict the estimation performance on general shapes with continuous curvature. Also, several possible extensions are discussed.

II. A Design Methodology for Perimeter Estimators

To any point P of the chain code, we can associate a list of properties $\{\text{prop } P\}$. We define a general length estimator as

$$\hat{L} = \sum_{\text{all points} P_i} \psi\{\text{prop } P_i\}. \tag{2.1}$$

Therefore, the length estimate of a chain coded contour will be the sum of some function of the local properties of the points P_i. Both the trivial perimeter evaluation procedure of link (or point) counting and the Proffitt-Rosen estimator immediately fit this general description. In the first case, ψ is a constant, and in the second,

$$\psi\{\text{prop } P\} = \begin{cases} L_c, & \text{if } P \text{ is a corner point} \\ L_n, & \text{if } P \text{ is a noncorner point.} \end{cases} \tag{2.2}$$

Suppose the properties of a point P are used to classify it to one of D disjoint classes $C_1, C_2, \cdots, C_D$ and $\psi\{\text{prop } [P]\} = \psi\{C_i\}$, i.e., the weight of point P in the estimator is a function of its class. Then, the estimator (2.1) may be rewritten as

$$\hat{L} = \sum_{i=1}^{D} N(C_i)\,\psi(C_i) \tag{2.3}$$

where we denoted by $N(C_i)$ the number of points of class C_i in the chain code. The relative estimation error is defined as

$$\epsilon = (L - \hat{L})/L = 1 - \hat{L}/L. \tag{2.4}$$

Suppose a straight edge of orientation θ is digitized. Let $\eta(C_i, \theta)$ be the density of points of class C_i per unit length. Then

$$\frac{\hat{L}(\theta)}{L} = \sum_{i=1}^{D} \eta(C_i, \theta)\,\psi(C_i) \tag{2.5}$$

and $\psi(C_i)$ should be chosen so as to make (2.5) as close to 1 as possible, uniformly over all θ. The Appendix summarizes some basic results concerning the properties of digitized straight edges, results that may be used to obtain the densities $\eta(C_i, \theta)$ for various point classifications.

The expression

$$\frac{4}{\pi} \int_{\theta=0}^{\pi/4} \left(1 - \frac{\hat{L}}{L}(\theta)\right) d\theta = B \tag{2.6}$$

measures the bias of the length estimator. For long boundaries with a fairly uniform distribution of tangent angle, one wishes to have unbiased length estimators, i.e., $B = 0$, and this immediately leads to

$$1 - \frac{4}{\pi} \sum_{i=1}^{D} \psi(C_i) \int_{\theta=0}^{\pi/4} \eta(C_i, \theta)\, d\theta = 0, \tag{2.7}$$

a linear relation between the $\psi(C_i)$'s. If unbiasedness is a requirement, this relation should be considered as a constraint on the design parameters $\psi(C_i)$.

Good estimators should have relative error as small as possible over the range of all orientations. Several global error measures can be considered for digitizations of straight edges, under the assumption that θ is uniform over $[0, \pi/4]$. The mean-square error (MSE) of an estimator is defined as

$$\frac{4}{\pi} \int_{\theta=0}^{\pi/4} \left(1 - \frac{\hat{L}(\theta)}{L}(\theta)\right)^2 d\theta = \text{MSE} \tag{2.8}$$

and the mean absolute deviation (mean AD) is given by

$$\frac{4}{\pi} \int_{\theta=0}^{\theta/4} \left|1 - \frac{\hat{L}(\theta)}{L}\right| d\theta = \text{mean AD}. \tag{2.9}$$

One can choose the coefficients $\psi(C_i)$ to minimize the MSE or the mean AD for either *biased* or *unbiased* estimators. The resulting performance is dictated by the choice of classes of points C_i and their relative frequency functions $\eta(C_i, \theta)$. Therefore, good estimators are obtained for point classifications C_i that enable the above measures to be small. For many object boundaries, particularly polygonal shapes such as triangles or rectangles, certain tangent angles may be dominant. In this case, it is important for the estimator to be insensitive to the orientation of the object relative to the grid. This leads to the *maximum absolute deviation* (max AD) as an alternative measure of performance, the minimization of which is, in our opinion, better suited for designing *robust* length estimators for arbitrary boundaries. Therefore, our performance measure will be

$$\text{max AD} = \max_{\theta \in [0, \pi/4)} \left| 1 - \frac{\hat{L}(\theta)}{L} \right|. \tag{2.10}$$

Clearly, if we minimize the maximum absolute deviation over all θ, and denote the achieved minimum by m, we then also have

$$\text{MSE} \leq \left\{ \max_{\theta \in [0(\pi/4)]} \left| 1 - \frac{\hat{L}(\theta)}{L} \right| \right\}^2 \leq m^2 \tag{2.11a}$$

$$\text{mean AD} \leq \max_{\theta \in [0(\pi/4)]} \left| 1 - \frac{\hat{L}(\theta)}{L} \right| \leq m. \tag{2.11b}$$

While it is generally possible to obtain estimators with lower MSE or mean AD than the estimator minimizing the max AD, such estimators will incur higher error at certain angles θ and their performance may not be *uniformly* good. Let us also point out that in some applications such as for mechanical drawings, we might wish to have highly accurate length estimators for boundaries of certain "preferred" orientations $\{\theta_k, k = 1, 2, \cdots, K\}$, which may appear particularly often. We could then design the length estimators so as to *force zero error* at the specified orientations, provided, of course there are enough "free" parameters to do so. Indeed, if in (2.5) we set $\theta = \theta_k$ and require $\hat{L}(\theta_k)/L$ to be equal to one, then a set of linear equations is obtained for $\psi(C_i)$. This set will be solvable provided the set of classes of points is "rich" enough.

The general point classification and *mini-max AD* methodology described above will first be applied to the case when points of the chain code are divided into two classes, corner and noncorner points, as done by Proffitt and Rosen. The complexity of the algorithm will then be increased slightly by also classifying the points according to the properties of their immediate neighbors in the chain code. Subsequently, we discuss perimeter estimation based on points classified according to the length of the runs of horizontal or vertical links in the chain code to which they belong.

III. New, Fast, and Accurate Perimeter Estimators

Max AD Criterion for the Proffitt-Rosen Estimator

We have defined in (2.3) a general perimeter estimator which requires the classification of points of a chain code into disjoint sets according to their local properties. An immediate classification is into corner (c) and noncorner (n) points, and according to (1.2) (also see the Appendix), we have

$$\eta(c, \theta) = 2 \sin \theta \tag{3.1a}$$

$$\eta(n, \theta) = \cos \theta - \sin \theta. \tag{3.1b}$$

The Proffitt and Rosen method assigns to the noncorner and corner points different weights $\psi(c) = \psi_c$ and $\psi(n) = \psi_n$, so as to make the length estimator unbiased and of minimal MSE. In this case, (2.7) and (2.8) together with (3.1) yield

$$\psi_c = \pi(\sqrt{2} + 2)/8 \quad \text{and } \psi_n = \pi(\sqrt{2} + 1)/8 \tag{3.2}$$

leading to an unbiased estimator with standard deviation ($\sqrt{\text{MSE}}$) of 2.3 percent.

The ratio $\hat{L}/L$ as a function of θ in the interval $\theta \in [0, \pi/4]$ can be expressed as

$$\begin{aligned} \frac{\hat{L}(\theta)}{L} &= 2\psi_c \sin \theta + \psi_n(\cos \theta - \sin \theta) \\ &= (-\psi_n + 2\psi_c) \sin \theta + \psi_n \cos \theta \\ &= F \cos (\theta - \zeta). \end{aligned} \tag{3.3}$$

The error function is

$$\epsilon(\theta) = 1 - \frac{\hat{L}(\theta)}{L} = 1 - F \cos (\theta - \zeta). \tag{3.4}$$

Any two constraints on $\epsilon(\theta)$ will determine F and ζ. We can design the estimator to minimize the maximum absolute deviation of $\epsilon(\theta)$. When minimized, the maximum AD occurs at the middle and end points of the interval $0 \leq \theta \leq \pi/4$. This requires that $\zeta = \pi/8$, and F is determined by

$$\begin{aligned} F - 1 &= 1 - F \cos (0 - \pi/8) \\ &= 1 - F \cos (\pi/4 - \pi/8) = \text{max AD} \end{aligned} \tag{3.5a}$$

which yields a maximum absolute deviation of

$$\begin{aligned} F - 1 &= \frac{2}{1 + \cos (\pi/8)} - 1 \\ &= \frac{1 - \cos (\pi/8)}{1 + \cos (\pi/8)} = 0.0396 \end{aligned} \tag{3.5b}$$

compared to 0.05194 for the Proffitt-Rosen estimator. The weights corresponding to (3.5) are

$$\psi_n = \frac{2 \cos (\pi/8)}{1 + \cos (\pi/8)}, \quad \psi_c = \frac{\sqrt{2} \cos (\pi/8)}{1 + \cos (\pi/8)}. \tag{3.6}$$

Finally, we point out that the weights ψ_c and ψ_n could also be chosen to force zero error at two arbitrarily chosen angles θ_1 and θ_2.

Design of a New Accurate Perimeter Estimator

To improve the performance of the perimeter estimator discussed above, let us increase the number of classes of points. To do so, we can, for example, classify the *noncorner* points according to the length of the chain code runs of horizontal or vertical links to which they belong. This, however, leads to a countably infinite number of classes. Let us reduce the number of classes by first classifying the noncorner points according to whether they connect two links of a run of length two or whether they belong to longer runs. Denote the two classes so defined by n and nn, respectively. From the results of the Appendix (also see Fig. 3), it is readily seen that in the interval $\theta \in [0, \tan^{-1} 1/3]$ where all the runs are of length three or more, only corner points (of class c) and points of class nn can occur. For a straight edge of orientation $\theta \in [\tan^{-1} 1/3, \tan^{-1} 1/2]$, which upon digitization produces runs of length two and three, points of class c, nn, and n occur. For $\theta \in [\tan^{-1} 1/2, \pi/4]$, inducing runs of length one and two, only points of class n and c occur.

A very simple analysis yields the densities of points in each class for the three regions defined above. (We could also use the results of the Appendix; however, a straightforward analysis is possible here.) From the general results on corner and noncorner points, (3.1), we have that

$$\eta(c) = 2\sin\theta, \qquad \theta \in [0, \pi/4] \qquad (3.7)$$

$$\eta(nn) = \cos\theta - \sin\theta, \qquad \theta \in [0, \tan^{-1} 1/3] \qquad (3.8)$$

$$\eta(n) = \cos\theta - \sin\theta, \qquad \theta \in [\tan^{-1} 1/2, \pi/4]. \qquad (3.9)$$

For $\theta \in [\tan^{-1} 1/3, \tan^{-1} 1/2]$, it is necessary to determine the proportion of noncorner points of class n versus those of class nn. Since in this interval nn points always belong to runs of 3 (see Fig. 3), we have

$$\eta(nn) + 2\eta(n) = \eta(c) = 2\sin\theta \qquad (3.10)$$

which together with (3.1b), $\eta(nn) + \eta(n) = \cos\theta - \sin\theta$, yields

$$\begin{cases} \eta(nn) = 2\cos\theta - 4\sin\theta \\ \eta(n) = 3\sin\theta - \cos\theta. \end{cases} \qquad (3.11)$$

Using these results, we obtain

$$\frac{\hat{L}(\theta)}{L} = \begin{cases} \psi(c)2\sin\theta + \psi(nn)(\cos\theta - \sin\theta), \\ \quad \theta \in [0, \tan^{-1} 1/3] \\ \psi(c)2\sin\theta + \psi(n)(3\sin\theta - \cos\theta) \\ \quad + \psi(nn)2(\cos\theta - 2\sin\theta), \\ \quad \theta \in [\tan^{-1} 1/3, \tan^{1} 1/2] \\ \psi(c)2\sin\theta + \psi(n)(\cos\theta - \sin\theta), \\ \quad \theta \in [\tan^{-1} 1/2, \pi/4]. \end{cases} \qquad (3.12)$$

From (3.12), it is seen that over each of the three domains, $\hat{L}(\theta)/L$ is of the form $F_i \cos(\theta - \zeta_i)$ where F_i and ζ_i are determined by the weighting coefficients $\psi(c)$, $\psi(n)$, and $\psi(nn)$, respectively. The three sinusoidal curves join together to form a continuous function [see Fig. 4(a)], the values at the breakpoints at $\theta_1 = 0$, $\theta_2 = \tan^{-1} 1/3$, $\theta_3 = \tan^{-1} 1/2$, and $\theta_4 = \pi/4$ also being determined by the parameters $\psi(c)$, $\psi(n)$, and $\psi(nn)$. However, we obviously cannot "tune" the $F_i \cos(\theta - \zeta_i)$ curves independently. If, using $\psi(c)$ and $\psi(n, n)$, we set the curve $F_1 \cos(\theta - \zeta_1)$ in the interval $[0, \tan^{-1} 1/3]$ to an optimal position according to the max AD criterion (or any other criterion), only one parameter $\psi(n)$ is left to optimize the estimator performance over the two remaining domains. Therefore, we do not have enough freedom to optimize the estimator performance over the three domains which are *implicitly* defined by the chosen point classification rule. While better performance can be obtained than with the immediate corner/noncorner point classification, we would like to have more freedom to tune the $\hat{L}(\theta)/L$ curve.

To achieve this, note that we can also refine the classification of the corner points. A corner point will be called a point of class ccc if its two neighbors are also

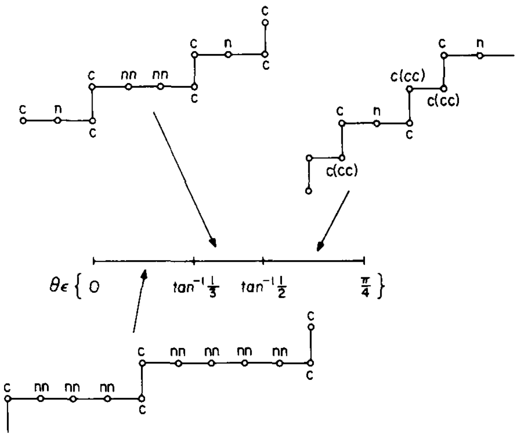
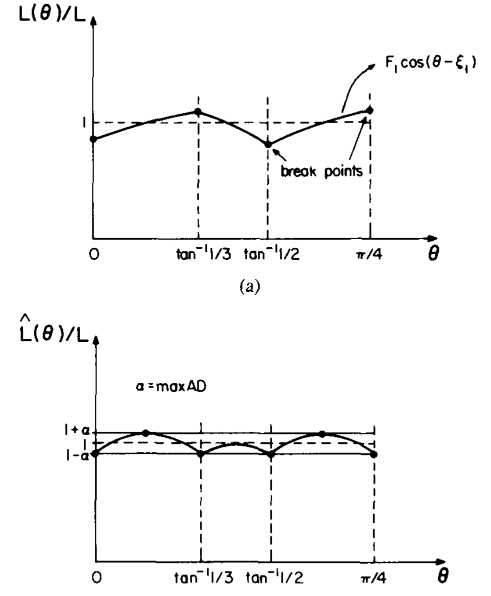


Fig. 3. Types of points for straight edge digitizations (see Appendix).



(a)

(b)

Fig. 4. (a) Performance evaluation curve. (b) The optimal design with c, ccc, n, and nn types of points.

corner points. Otherwise, it will be a *c* point. Clearly, *ccc* points occur only in the domain where runs of length 1 exist, i.e., in the interval $\theta \in [\tan^{-1} 1/2, \pi/2]$. From (3.7), we have that

$$\eta(c) + \eta(ccc) = 2 \sin \theta \qquad (3.13)$$

and from Fig. 3, it is seen that the number of *c* points will be twice the number of *n* points, i.e.,

$$\eta(c) = 2\eta(n). \qquad (3.14)$$

This, together with (3.9), yields

$$\begin{cases} \eta(c) = 2(\cos \theta - \sin \theta) \\ \eta(ccc) = 2(2 \sin \theta - \cos \theta). \end{cases} \qquad (3.15)$$

Thus, by refining the corner point classification, (3.12) is modified to

$$\frac{\hat{L}(\theta)}{L} = \begin{cases} \psi(c)2 \sin \theta + \psi(nn)(\cos \theta - \sin \theta) \\ \qquad \theta \in [0, \tan^{-1} 1/3] \\ \psi(c)2 \sin \theta + \psi(n)(3 \sin \theta - \cos \theta) \\ \quad + \psi(nn)2(\cos \theta - 2 \sin \theta) \\ \qquad \theta \in [\tan^{-1} 1/3, \tan^{-1} 1/2] \\ \psi(c)2(\cos \theta - \sin \theta) + \psi(ccc)2 \\ \quad \cdot (2 \sin \theta - \cos \theta) \\ \quad + \psi(n)(\cos \theta - \sin \theta) \\ \qquad \theta \in [\tan^{-1} 1/2, \pi/4]. \end{cases} \qquad (3.16)$$

Now, using $\psi(c)$ and $\psi(nn)$, we can freely set the $F_1 \cos (\theta - \zeta_1)$ curve over $\theta \in [0, \tan^{-1} 1/3]$. Similarly, using $\psi(ccc)$ and $\psi(n)$, we can independently set $F_3 \cos (\theta - \zeta_3)$ over $\theta \in [\tan^{-1} 1/2, \pi/4]$.

With this classification, we have four parameters implicitly defining three intervals in the range of θ. The parameters may be tuned to set the values of $\hat{L}(\theta)/L$ at the four breakpoints $\theta_1 = 0$, $\theta_2 = \tan^{-1} 1/3$, $\theta_3 = \tan^{-1} 1/2$, $\theta_4 = \pi/4$. This enables us to exploit *all* the freedom implied by the θ-range division since the values at the breakpoints can specify *any* continuous, piecewise sinusoidal function for the $\hat{L}(\theta)/L$ curve.

In each of the three intervals, $\hat{L}(\theta)/L$ consists of a sinusoid segment with period 2π. As with expression (3.4), the max AD, when minimized, occurs at the middle and end points of the largest interval. In this case, the middle interval, of length $\theta_3 - \theta_2$, is the smallest. Its center is at $\pi/8$ since it can be shown that

$$(\tan^{-1} 1/3 + \tan^{-1} 1/2)/2 = \pi/8. \qquad (3.17a)$$

From (3.17a), it follows that the first and third intervals have identical lengths $\phi_{\max} = (\pi/4 - \theta_3) = \theta_2 = \tan^{-1} 1/3$ since

$$(\theta_4 - \theta_3) - (\theta_2 - \theta_1) = \pi/4 - \tan^{-1} 1/2 - \tan^{-1} 1/3 = 0. \qquad (3.17b)$$

Thus, with two largest intervals, the max AD occurs at all the breakpoints θ_i, $i = 1, 2, 3, 4$, and at $\theta_2/2$ and $(\pi/4 - \theta_3)/2$, as shown in Fig. 4(b). With a computation identical to that of (3.4), (3.5), the max AD is given by

$$\max \text{AD} = \frac{1 - \cos (\phi_{\max}/2)}{1 + \cos (\phi_{\max}/2)} \qquad (3.18)$$

where, in this case, $\phi_{\max} = \tan^{-1} 1/3$.

Thus, optimization of $\hat{L}(\theta)/L$ with respect to the max AD criteria, yielding the function of Fig. 4(b), results in a max AD over all θ of 0.650 percent, almost an order of magnitude better than for corner/noncorner point classification. The weights for the optimum $\hat{L}(\theta)/L$ of Fig. 4(b) can be determined by first using (3.16) to obtain the weights corresponding to breakpoint values of $\hat{L}(\theta_i)/L = 1$, $i = 1, 2, 3, 4$. Multiplying these weights by the factor $1 - \alpha$, $\alpha = \max \text{AD}$, determines the optimum weights

$$\begin{aligned} \psi(nn) &= 1 - \alpha = 0.99350 \\ \psi(ccc) &= (\sqrt{2}/2)(1 - \alpha) = 0.70251 \\ \psi(n) &= (2 + \sqrt{5} - \sqrt{10})(1 - \alpha) = 1.06681 \\ \psi(c) &= (\sqrt{10}/2 - 1)(1 - \alpha) \doteq 0.57736. \end{aligned} \qquad (3.19)$$

To determine class *nn* points, it is necessary to check for sequences of three horizontal or vertical links. For class *ccc* points, it is necessary to check for sequences of four alternating links. In either case, the Euclidean distance between the end points of the link sequences is not greater than three pixel units. Thus, if the boundary curve can be assumed to be straight over three pixel units, the estimator performance will be determined by the $\hat{L}(\theta)/L$ curve and the distribution of the tangent angle of the boundary.

This first estimator was implemented and tested on digitized boundaries of randomly placed disks. Two sets of experiments were run. First, 100 circular shapes for each of the integer value radii $R \epsilon \{2, 3, \cdots, 9, 10, 20, 30, 50, 100\}$ were thrown on the grid, with center location uniformly and independently distributed in a square pixel area. The shapes were digitized and their perimeter was estimated by

$$\hat{L} = N_c\psi(c) + N_{ccc}\psi(ccc) + N_n\psi(n) + N_{nn}\psi(nn) \qquad (3.20)$$

where the optimized ψ parameters given above were used. This estimate was then compared to the true perimeter and the percentage error was computed. At each radius, the average error as well as the maximal, minimal error, and average absolute error or deviation (AAD) were determined over the ensemble of 100 runs. To eliminate any possible anomalies due to integer value radii, the experiment was repeated for randomly chosen radii, uniformly distributed over the intervals $(R - 0.5, R + 0.5)$, for $R \in \{2, 3, \cdots, 9, 10, 20, 30, 50, 100\}$. For brevity, only these results are presented in Fig. 5 and displayed in Ta-

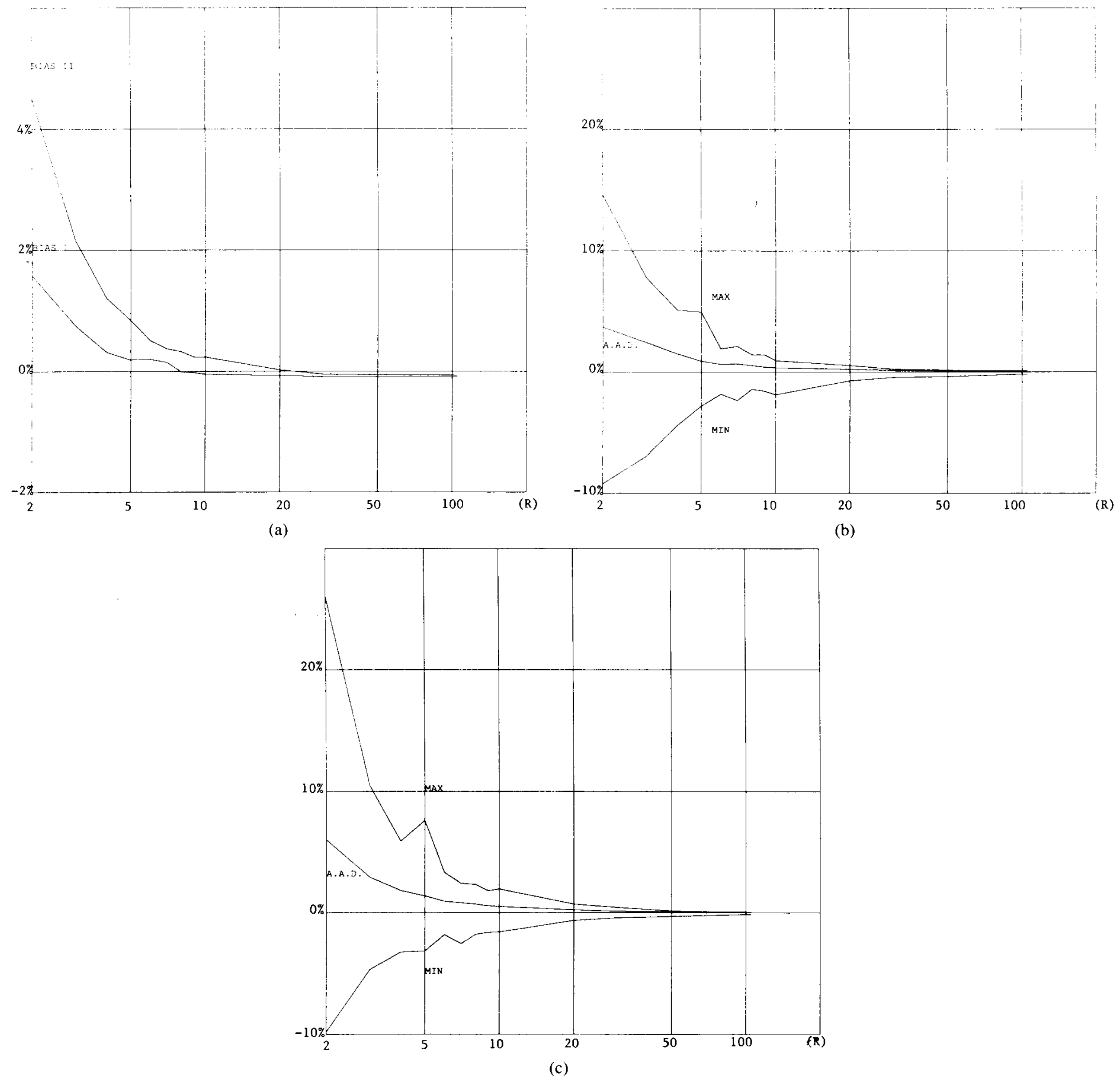


Fig. 5. (a) The average error percentage (bias) for the two algorithms implemented. (b) The minimum (MIN), maximum (MAX), and average absolute deviations (AAD) for method 1 (the four-class method). (c) The minimum (MIN), maximum (MAX), and average absolute deviations (AAD) for method 2 (the corner smoothing method).

ble I. As seen in the lower curve of Fig. 5(a), the average error quickly reaches the bias value of the $\hat{L}(\theta)/L$ curve (2.6), indicating that, for long boundaries, curvature radii of more than five pixel units will have negligible effect on the length estimator. The experimental results displayed in Fig. 5(b) also show that a high degree of accuracy may also be possible for estimating the perimeter of small objects. For circles of radii greater than five pixel units, both the minimal and maximal percentage deviations are already within 2.5 percent and the average absolute error is less than 1 percent.

The performance at integer radii was found to be somewhat better than at radii slightly above or below the integer values, particularly for the maximal and minimal error. This is due to the fact that with an integer value radius, the number of links in the digitized perimeter is constant, and thus insensitive to center point location.

Determination of Breakpoints for $\hat{L}(\theta)/L$

In the above design, the classification of chain code points implicitly defined a division of the $[0, \pi/4]$ interval into three disjoint regions $\{\Theta_i\}$ characterized accord-

TABLE I
SIMULATION RESULTS SUMMARIZING 100 RUNS OF PLACING CIRCLES WITH RANDOM CENTER POINT AND UNIFORMLY DISTRIBUTED RADII BETWEEN $R - 0.5$ AND $R + 0.5$

	ESTIMATOR 1 (c,ccc,n,nn)				ESTIMATOR 2 (edge-smoothing)			
radius ± 0.5	min	max	A.A.D.	A.D.	min	max	A.A.D.	A.D.
2	-9.23	14.67	3.71	1.58	-9.81	26.08	6.03	4.51
3	-6.98	7.78	2.45	0.75	-4.64	10.48	2.92	2.15
4	-4.43	5.13	1.51	0.32	-3.22	5.89	1.82	1.20
5	-2.82	4.95	0.90	0.19	-3.14	7.60	1.38	0.84
6	-1.83	1.93	0.64	0.20	-1.81	3.31	0.92	0.50
7	-2.37	2.13	0.66	0.15	-2.53	2.42	0.82	0.37
8	-1.44	1.44	0.54	-0.01	-1.78	2.33	0.71	0.32
9	-1.58	1.44	0.40	-0.02	-1.61	1.81	0.57	0.23
10	-1.88	0.95	0.36	-0.05	-1.57	1.95	0.51	0.23
20	-0.72	0.54	0.22	-0.07	-0.63	0.70	0.22	0.02
30	-0.45	0.23	0.14	-0.09	-0.41	0.44	0.11	-0.04
50	-0.38	0.15	0.10	-0.09	-0.30	0.14	0.07	-0.06
100	-0.19	0.02	0.09	-0.09	-0.14	0.02	0.06	-0.06

min - minimum of $\hat{L}/L$ - 1 over the 100 runs.
max - maximum of $\hat{L}/L$ - 1 over the 100 runs.
A.A.D. - average absolute deviation (error).
A.D. - average deviation (error).

Note: all errors are relative and measured in percentages.

ing to the types of points that occur in digitized straight edges having orientations $\theta \in \Theta_i$. Clearly, if we refine the point classifications, we expect a corresponding refinement of the division of the θ range.

We saw, however, that there exist situations in which an increase in the number of classes of points does not lead to a refinement of the θ-range division. In our case, the definition of the *ccc* class of points did not alter the θ-range division at all. However, it provided an additional parameter that enabled the further improvement of the estimator by allowing the independent placement of the four breakpoints of the $\hat{L}(\theta)/L$ function. Note that we could have also defined other classes of points, such as the class of corner points that have no corner-point neighbors (denoted, say, by *ncn*) or the class of noncorner points having only noncorner neighbors, denoted by *nnn* (letting *nn* be the class of points with only *one* noncorner neighbor).

The latter class of points obviously appears only for straight edges of orientation $\theta < \tan^{-1} 1/3$. Using $\psi(c)$, $\psi(n)$, $\psi(nn)$, and $\psi(nnn)$ would also enable us to tune the $\hat{L}(\theta)/L$ curve with maximum possible freedom by allowing arbitrary values at the same four breakpoints. There is no advantage, however, to define *five classes* of points such as *c*, *ccc*, *n*, *nn*, and *nnn*. The additional parameter does not further subdivide the θ range, and hence cannot improve the $\hat{L}(\theta)/L$ function since it remains a continuous function composed of three sinusoidal portions. As an aside, we note that the $\hat{L}(\theta)/L$ curve has a piecewise sinusoidal shape only if the classification of chain chode points is such that the density functions $\eta(C_i, \theta)$ are of the form $X_i \cos\theta + Y_i \sin\theta$ for all the θ-range intervals. In our examples, this is clearly the case; however, one might invent classifications based on local chain code properties for which the density functions are not piecewise sinusoidal.

Assume that $\eta(C_i, \theta)$, $i = 1, 2, \cdots, D$ are all either 0 or sinusoidal over the K intervals of the θ range induced by the classification $C_1, C_2, \cdots, C_D$. Then we have $\psi(C_1) \cdots \psi(C_D)$ as D free parameters to design a perimeter estimator with performance curve $\hat{L}(\theta)/L$, a continuous function made of K sinusoidal portions with $K + 1$ breakpoints. If $D = K + 1$ and we can independently set the values of $\hat{L}(\theta)/L$ at the $K + 1$ breakpoints using the $\psi(C_i)$, $i = 1, 2, \cdots, D$ parameters, then we call the design *complete* since all the freedom due to the induced θ-range division can be exploited. If $D < K + 1$, then it may be possible, by adding parameters and without increasing K, to obtain a *complete* design. On the other hand, if $D > K + 1$, we then have parameters that cannot contribute to the shaping of the $\hat{L}(\theta)/L$ curve, and hence the design is *overparameterized* (and consequently, unnecessarily complex).

We have seen that the original Proffitt–Rosen design is complete. Our first step of improvement, which was based on defining three classes of points *c*, *n*, and *nn* and yielded (3.12), was recognized to be underparameterized, but adding one more class of points, the design leading to (3.16) became complete. The example given above, with five classes of points, is an overparametrized design. In general, the value of refining the classification of the chain points should be judged according to its potential of yielding an improvement in performance. Thus, a classification refinement that increases the number of induced intervals of the θ range is always useful, but we should always seek to use the θ-range division maximally via complete designs.

Design of Another Accurate Perimeter Estimator

The foregoing discussion showed that we can improve the accuracy of a perimeter estimation algorithm by refining the division of the orientation range $[0, \pi/4]$ by increasing the number of classes of chain points.

In analyzing straight edge digitizations (see the Appendix), we realize that within the orientation intervals $\theta \in [\tan^{-1} 1/(j + 1), \tan^{-1} 1/j]$, there are only horizontal runs of length j and $j + 1$ alternating with vertical runs of length 1. This fact provides the idea for the following perimeter estimator. Classify the points of the boundary chain code according to the length of the *longest* run they belong to. Thus, noncorner points will be classified according to the run they belong to, and the corner points, which by definition belong to two runs, will be classified according to the longer of the two runs. Therefore, let C_i be the class of points that belong to (longest) runs of length i. If a straight line of orientation $\theta \in \in [1/(j + 1), 1/j]$ is digitized, we only have runs of length j and $j + 1$, and thus points of class C_j and C_{j+1}. Their density was found to be (see (A.8) in the Appendix)

$$\eta(C_j) = (j + 1)[(j + 1)\sin\theta - \cos\theta] \quad (3.21a)$$

$$\eta(C_{j+1}) = (j + 2)(\cos\theta - j\sin\theta). \quad (3.21b)$$

Thus, the performance function of the general perimeter estimator (2.5) over $\theta \in [\tan^{-1} 1/(j+1), \tan^{-1} 1/j]$ is given by

$$\begin{aligned}\frac{\hat{L}(\theta)}{L} &= \psi(C_j)(j+1)[(j+1)\sin\theta - \cos\theta] \\ &\quad + \psi(C_j+1)(j+2)[\cos\theta - j\sin\theta] \\ &= F_j \cos(\theta - \zeta_i). \qquad (3.22)\end{aligned}$$

In particular, for $\theta \in [\tan^{-1} 1/2, \pi/4]$, we obtain

$$\begin{aligned}\frac{\hat{L}(\theta)}{L} &= = \psi(C_1)2[2\sin\theta - \cos\theta] \\ &\quad + \psi(C_2)3[\cos\theta - \sin\theta] \qquad (3.23)\end{aligned}$$

which shows that over this interval, we can do *only* as well as in the previous estimator discussed. We can make the performance curves coincide over this interval by choosing identical breakpoint values at the end points of the interval $[\tan^{-1} 1/2, \pi/4]$. However, over the interval $\theta \in [0, \tan^{-1} 1/3]$, we have here a very fine division of the straight-edge orientation into disjoint regions, having breakpoints at all $\theta_j = \tan^{-1} 1/j$. The performance of this estimator for edge orientations in this interval can therefore be made much better than the performance of the estimator presented in the previous section. If all orientations are uniformly present in the contour of the digitized object, the expected performance will always be better with the second design. This estimator is also important because it generalizes the following ad hoc estimator, based on a natural corner smoothing idea.

Consider the following perimeter estimation procedure. Choose a starting point P_0 on the chain of the digitized object boundary and a direction of traversal. From the starting point P_0, follow the links of the chain code to a point P_1 to which two runs of links are formed with at least one of them being of length one. [It may happen that the immediate neighbor of the starting point in the traversal direction is a corner point; then the first run is of unit length and P_1 will be the next corner point, or it may happen that first a longer run occurs and then P_1 will be the point following the first corner point encountered; see Fig. 6(a).] Increment the perimeter estimate by the Euclidean distance $d(P_0 P_1)$ and set P_1 to be the next starting point. Proceed with this until from some point, the initial starting point P_0 is encountered. Note that the perimeter estimate obtained by this procedure is simply the length of a polygonal approximation of the chain coded contour, and this polygonal approximation is dependent on the initial point P_0 that is chosen. This feature of the above procedure will clearly add some variability to the length estimates, especially for small objects. However, we expect its performance to be quite good since a rather clever corner smoothing is implicit in the algorithm. Let us see how we can analyze the performance of this method.

Referring to Fig. 6(b), we see that, for straight edge digitizations, a run of length j will contribute $\sqrt{1+j^2}$ to the perimeter estimate. Therefore, if we attempt to rephrase the above algorithm in the general framework of classifying chain code points, the assigned weight to the various classes of points must be such that the $j+1$ points of a j run should have a total weight of $\sqrt{1+j^2}$; hence, $\psi(C_j) = (j+1)^{-1}\sqrt{1+j^2}$. It is evident that with such a weight allocation, the point classification algorithm discussed before and the perimeter estimation procedure considered above on a heuristic basis have the same performance on straight edge digitizations. Furthermore, it can be readily checked that with the latter algorithm, we have $\hat{L}(\theta)/L \geq 1$, with equality at all the breakpoints, $\theta_j = \tan^{-1} 1/j$, $j \in N$. This implies that for $\theta \in [\tan^{-1} \frac{1}{2}, \pi/4]$, the corner smoothing algorithm has a worse max AD performance than the one proposed in the previous section. Indeed, the plot of $\hat{L}(\theta)/L$ shows that $\hat{L}(\theta)/L|_{\max \mathrm{AD}} = \hat{L}[\frac{1}{2}(\tan^{-1} 1/2 + \pi/4)]/L = 1.01308$, i.e., an error percentage of 1.3 percent. The performance of the corner smoothing algorithm can, however, easily be made as good as in the previous case by appropriately lowering the $\hat{L}(\theta)/L$ curve in the interval $[\tan^{-1} 1/2, \pi/4]$. This is accomplished by modifying $\psi(C_1)$ and $\psi(C_2)$ to

$$\begin{aligned}\psi(C_1) &= \sqrt{2}(1 - 0.00650)/2, \\ \psi(C_2) &= \sqrt{5}(1 - 0.00650)/3. \qquad (3.24)\end{aligned}$$

Such a choice for the weights reduces the max AD over this interval to 0.65 percent, thereby making the corner

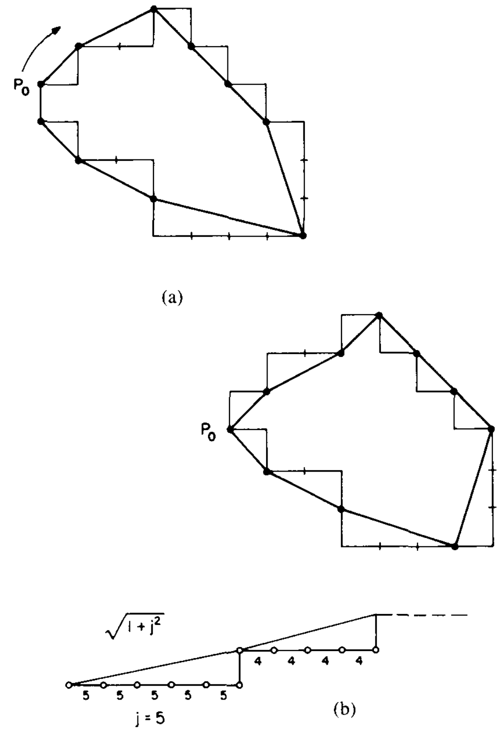


Fig. 6. The corner smoothing perimeter estimation process. Note (a) the initial point dependence. (b) shows the weight of points of class j in the point classification equivalent of the corner smoothing method.

smoothing algorithm identical in performance (with the max AD measure) to the one proposed in the previous section.

We have shown that the run-length classification algorithm may be made identical to an ad hoc algorithm based on measuring Euclidean distances of straight segments smoothing out the corners of the chain code. This smoothing algorithm is similar in spirit to the one proposed in [15] and in [14]. Note, however, that they are not identical, except in the case of straight edge digitizations. Therefore, the foregoing performance analysis also holds for the algorithm of Wechsler and of Grant and Reid on straight edges.

As pointed out previously, both the corner smoothing or the run-length classification procedures cannot improve the theoretical max AD performance. Such algorithms are, however, expected to yield better results for continuous boundaries in which all directions appear since they provide a very fine division of the interval $[0, \tan^{-1} 1/3]$. To test the performance of such an algorithm and compare it to the four-class algorithm introduced previously, we have also implemented the corner smoothing procedure, and tested it on digitized boundaries of randomly placed disks. The algorithm chose an arbitrary starting point and weighted the runs according to $\sqrt{(1 + j^2)}$, except in the cases of runs of length one and two, for which the weights were chosen according to (3.24). It was expected that this algorithm will have very similar performance to the previously tested one, being somewhat better for circular shapes of large radii.

The same set of experiments was run as with the previous algorithm. The results with randomly chosen radii, uniformly distributed over the intervals $(R - 0.5, R + 0.5)$ for $R \in \{2, 3, \cdots, 9, 10, 20, 30, 50, 100\}$, are presented in Fig. 5(a) and (c) and displayed in Table I together with the previous experimental results.

Comparing the results of the corner smoothing algorithm to the four-class algorithm tested before, we realize that for low values of the radii, i.e., high curvatures, the four-case algorithm is better, whereas for large radii, as predicted, the corner smoothing algorithm has an advantage. However, both algorithms are seen to give low errors, even for circles of very small radii compared to the pixel size. This experimental result is quite surprising, and indicates that good performance should be expected, even for small perimeters.

In conclusion, we have seen that classifying the points according to the (longest) run length to which they belong leads to an improved estimator, but only in the sense that the $\hat{L}(\theta)/L$ ratio is better for $\theta < \tan^{-1} 1/3$. The range $[\tan^{-1} 1/2, \pi/4]$ is not subdivided, and this implies that such designs can only be as accurate, in the max AD sense, as the previously discussed one. The design we presented has an infinite subdivision of the θ range and is *complete* in the sense discussed previously. Note that we could also define a finite number of classes of points as those belonging to runs of length $1, 2, \cdots, D - 1$, and a class $C_{\geq D}$ of those belonging to runs of length D or longer. This would be equivalent to assigning identical weight to points of class $C_D, C_{D+1}, \cdots$. Over the interval $\theta \in [0, \tan^{-1} 1/D]$, we have only points of class $C_{\geq D}$, the density of points being $\sin \theta + \cos \theta$. This does not give full freedom to arbitrarily set the breakpoints at that interval. We would consequently have a finite θ-range subdivision but an *incomplete* design (D parameters and D regions).

Finally, we point out again that several different algorithms may have the same straight edge performance, and it is only on the basis of simulations or other additional performance measures that one can decide which algorithm is better suited for particular purposes.

IV. Discussion

We have presented a methodology for designing perimeter (or length) estimators for digitized boundaries. The discussion was in terms of four-directional chain coded boundaries; however, the identical methodology can be used for eight- or six-directional codes as well. There is an immediate improvement in performance with higher directional codes due to their greater precision in representing the boundary. In fact, in the literature, there often appear comparisons for four-directional codes and their eight-directional counterparts showing that the eight-directional codes are better. Vossepoel and Smeulders [11], for example, show that using the Proffitt–Rosen perimeter estimate for eight-directional codes leads to an immediate improvement in performance. A four-directional code can be changed to an eight-directional code by replacing adjacent perpendicular links with a diagonal link. Thus, the methodology presented here makes it easy to recognize that a Proffitt–Rosen type approach on an eight-directional code (in which we differentiate between diagonal links and horizontal or vertical ones) is, in fact, equivalent to defining more classes of points for the four-directional code. Suppose the corner smoothing or run classification algorithm is applied on an eight-directional code. Classify the points according to the length of the run they belong to and whether the run is horizontal, vertical, or diagonal. The effect of this classification is to divide the θ range of $[0, \pi/4]$ into intervals delimited by breakpoints at $\theta_i = \tan^{-1} 1/i$, $i \geq 1$, as well as at $\theta_i^* = \tan^{-1} (i - 1)/i$, $i \geq 2$. This yields a complete design with the $\hat{L}(\theta)/L$ curve in each interval being a sinusoid determined by two ψ parameters which set the breakpoint values. The dominant error here is in the longest θ-range interval $(\tan^{-1} 1/2, \tan^{-1} 2/3)$, and the optimal max AD estimator yields an extremely low error or 0.12 percent over the entire range of straight edge orientations.

We have seen that a relatively simple design methodology yields good and simple length estimators, their performance being optimized over all the orientations of straight edge digitizations. It is interesting to ask how these estimators perform on arbitrary curves. If the curves are long, have moderate curvature everywhere (relative to the pixel size), and the tangent directions are uniformly distributed over $[0, 2\pi]$, we can, of course, expect to have an average error roughly proportional to the bias or average of the $[\hat{L}(\theta)/L - 1]$ curve. If the directions have arbitrary distribution, we can still expect a maximal overall error less than the max AD of the estimator.

Experimental results show that the bias is insensitive to moderate curvature and only slightly affected at high curvatures. From the experimental graph of bias versus curvature, the length estimator performance can be roughly predicted for long boundaries based on either *a priori* information or estimates of the curvature distribution. The experimental results also show that the mean absolute deviation, the maximal, and minimal deviation for digitized disks are surprisingly good at high curvatures. This indicates that accurate perimeter estimation is possible for small objects as well.

Appendix
Digitized Straight Edges and Their Properties

This section briefly derives some facts on crack codes of straight edge digitizations. Suppose the object digitized on a square grid of size one is the half plane defined by

$$y \le f(x) = mx + n. \tag{A.1}$$

Without loss of generality, assume that the line orientation $\theta = \tan^{-1} m$ is between 0 and 45° ($\pi/4$). Recalling the definition of point sampling image digitization, we realize that for $x = k \in N$, the black pixels are those with centers having $y \le [mk + u]$, $y \in N$. This shows that the crack code of the straight boundary digitization has runs of horizontal links separated by single vertical links. The horizontal runs correspond to the pixels whose y coordinate is some $L \in N$ and x coordinates satisfy $x \in [x_L\ x_{L+1})$ where

$$mx_L + n = L \quad \text{and} \quad mx_{L+1} + n = L + 1, \qquad \forall L \in N. \tag{A.2}$$

Since $x_{L+1} - x_L = 1/m = \cos\theta/\sin\theta$, it is immediate that the horizontal runs can only have $[1/m]$ or $[1/m] + 1$ links. Furthermore, the proportion of runs of length $[1/m]$ and $[1/m] + 1$ can readily be determined. Indeed, the runs of the boundary chain code are determined by the number of integer coordinates that fall into consecutive segments of length $1/m$ on the x axis; see Fig. 7. Suppose that one of the segments of length $1/m$ starts at some point $K + \epsilon$. If $1/m$ is rational, i.e., $1/m = U/V$ for U, $V \in N$, U and V being relatively prime $[(U, V) = 1]$, then after placing V successive horizontal segments, we shall be at $(U + K) + \epsilon$, i.e., at the same starting point modulo an integer. Therefore, the pattern of runs of length $[1/m]$ and $[1/m] + 1$ will be repeated periodically with period V. The proportion of runs of length $[1/m]$, $\alpha_{[1/m]}$ can readily be determined from

$$\alpha_{[1/m]} V \left[\frac{1}{m}\right] + (1 - \alpha_{[1/m]}) V \left(\left[\frac{1}{m}\right] + 1\right) = U \tag{A.3}$$

yielding

$$\alpha_{[1/m]} = \left[\frac{1}{m}\right] + 1 - \frac{1}{m}. \tag{A.4}$$

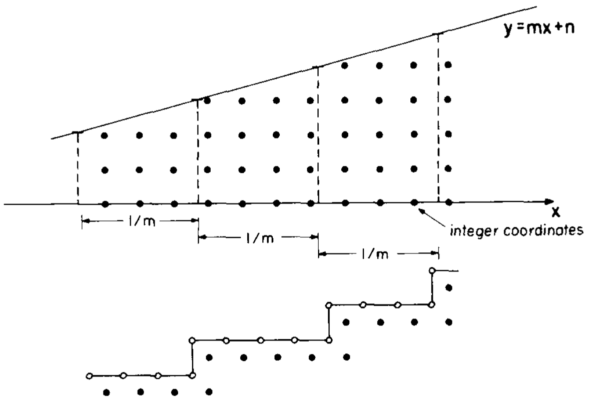


Fig. 7. Geometry of straight edge digitization.

If $1/m$ is not rational, we can find two sequences of rationals that approach $1/m$ from above and below, and thereby prove that, although the patterns of runs are not periodic, the proportion of runs of length $[1/m]$ is still given by (A.4), in general.

With (A.3) and (A.4), we have the statistics of runs for any orientation $\theta = \tan^{-1} m$ if the edge is straight and of infinite extent. It is easy to realize that if ϵ is chosen randomly with uniform distribution over $[0, 1]$ and the starting point of a segment of length $1/m$ is $K + \epsilon$, then (A.4) also gives the probability of having a run of length $[1/m]$, i.e., of having exactly $[1/m]$ integers in the interval $[K + \epsilon, K + \epsilon + 1/m]$.

From the statistics of runs, we can easily obtain the statistics of corner and noncorner points of the boundary chain code. The basic relations are illustrated in Fig. 7. To a run of length p, we have $(p - 1)$ noncorner points and two corner points. Suppose a straight edge of orientation θ is digitized, and the resulting chain code has runs of length $[1/m]$ and $[1/m] + 1$ where $m = \tan\theta$. Consider a very long portion of this edge of length L. Let Q_L denote the total number of runs. Then the horizontal distance $L\cos\theta$ will be covered by $Q_L\alpha_{[1/m]}$ runs of length $[1/m]$ and $Q_L(1 - \alpha_{[1/m]})$ runs of length $[1/m] + 1$. Therefore, we have

$$\begin{aligned} L\cos\theta &\\ &= Q_L\left\{\alpha_{[1/m]}[1/m] + (1 - \alpha_{[1/m]})([1/m] + 1)\right\} \\ &= Q_L\frac{1}{m} = Q_L\frac{\cos\theta}{\sin\theta}. \end{aligned} \tag{A.5}$$

This shows (again) that $Q_L = L\sin\theta$. For each run, we have two corner points, and therefore the number of corner points per unit length is

$$N_c = 2\sin\theta. \tag{A.6}$$

The number of noncorner points/unit length is found similarly, yielding

$$\begin{aligned} N_n &= \sin\theta\left\{\alpha_{[1/m]}([1/m] - 1) + (1 - \alpha_{[1/m]})[1/m]\right\} \\ &= \sin\theta\left\{\frac{1}{m} - 1\right\} = \cos\theta - \sin\theta. \end{aligned} \tag{A.7}$$

Note that we could classify the points (or the links) of the boundary into more refined classes according to the properties of their neighbors too. To any point of the chain code, we can associate a list of properties $\{\text{prop } P\}$ such as whether the point is a corner point or not, if it is a corner point the length of the link runs it joins, if it is not a corner point the length of the run it belongs to, etc.

If we classify the points according to such properties, the above-presented analysis can easily tell us the frequencies of points of each class per unit length as a function of straight edge orientation. For example, classify the points of the chain code according to the length of the *longest* run they belong to. Thus, noncorner points will be classified according to the run they belong to, and the corner points, which by definition belong to two runs, will be classified according to the longer of the two runs. Therefore, let C_j be the class of points that belong to (longest) runs of length j. If a straight line of orientation $\theta = \tan^{-1} m$ is digitized and $m \in [1/(j+1), 1/j]$, then we have runs of length j and $j + 1$. In a run of j, we have $j + 1$ points of class C_j, and in a run of $j + 1$, $j + 2$ points of class C_{j+1}. Over the interval $\theta \in [\tan^{-1} 1/(j + 2), 1/j]$, we have $\sin\theta$ runs per unit length and $\alpha_j = j + 2 - \cos\theta/\sin\theta$ of them are of length j. Therefore, we readily obtain

$$\eta(C_j) = (j+1)((j+1)\sin\theta - \cos\theta) \quad \text{(A.8a)}$$

$$\eta(C_{j+1}) = (j+2)(\cos\theta - j\sin\theta). \quad \text{(A.8b)}$$

References

[1] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. New York: Academic, 1976.

[2] T. Pavlidis, *Algorithms for Graphics and Image Processing*. Rockville, MD: Computer Science, 1982.

[3] C.-S. Ho, "Precision of digital vision systems," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 593-601, 1983.

[4] M. Rink, "A computerized quantitative image analysis procedure for investigating features and an adapted image process," *J. Microscopy*, vol. 107, pp. 267-286, 1976.

[5] Z. Kulpa, "Area and perimeter measurement of blobs in discrete binary pictures," *Comput. Graphics Image Processing*, vol. 6, pp. 434-451, 1977.

[6] H. Freeman, "Computer processing of line drawing images," *Comput. Surveys*, vol. 6, pp. 57-97, 1974.

[7] ——, "Boundary encoding and processing," in *Picture Processing and Physhopictorics*, B. Stipkin and A. Rosenfeld, Eds. New York: Academic, 1970, pp. 241-263.

[8] Z. Kulpa, "More about areas and perimeters of quantized objects," *Comput. Vision, Graphics, Image Processing*, vol. 22, pp. 268-276, 1983.

[9] U. Montanari, "A note on minimal length polygonal approximation to a digitized contour," *Commun. ACM*, vol. 13, pp. 41-46, 1970.

[10] D. Proffitt and D. Rosen, "Metrication errors and coding efficiency of chain coding schemes for the representation of lines and edges," *Comput. Graphics Image Processing*, vol. 10, pp. 318-332, 1979.

[11] A. M. Vossepoel and A. W. M. Smeulders, "Vector code probability and metrication error in the representation of straight lines of finite length," *Comput. Graphics Image Processing*, vol. 20, pp. 347-364, 1982.

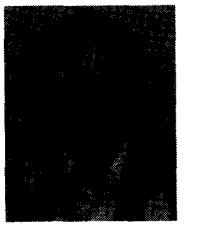
[12] J. Koplowitz and G. T. Toussaint, "A unified theory of coding schemes for the efficient transmission of line drawings," in *Proc. Canadian Conf. Commun. Power*, Oct. 1976.

[13] J. Koplowitz, "On the performance of chain codes for quantization of line drawings," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 180-185, 1981.

[14] G. Grant and A. F. Reid, "A fast and precise boundary tracing algorithm," *Mikroskopie*, vol. 37, pp. 455-457, 1980.

[15] H. Wechsler, "A new and fast algorithm for estimating the perimeter of objects for industrial vision tasks," *Comput. Graphics Image Processing*, vol. 17, pp. 375-385, 1981.

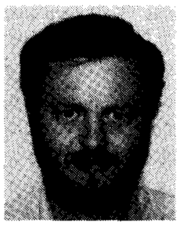
[16] P. Billingsley, *Ergodic Theory and Theory of Information*. New York: Wiley, 1965.



Jack Koplowitz (S'71-SM'79) received the B.E.E. degree from the City College of New York, New York, NY, in 1967, the M.E.E. degree from Stanford University, Stanford, CA, in 1968, and the Ph.D. degree from the University of Colorado, Boulder, in 1973.

From 1967 to 1970 he was a member of the Technical Staff at Bell Laboratories, Holmdel, NJ, working in the area of data communications. Since 1973 he has been with the Department of Electrical and Computer Engineering at Clarkson University, Potsdam, NY. He spent the 1986-1987 academic year as a Lady Davis Fellow at the Department of Electrical Engineering, Technion—Israel Institute of Technology, Haifa. His research interests are in pattern recognition, image processing, and statistical communications.

Dr. Koplowitz served as Secretary of the Board of Governors of the IEEE Information Theory Group from 1981 to 1983. From 1984 to 1987 he served as Associate Editor for Pattern Recognition of the IEEE Transactions on Information Theory.



Alfred M. Bruckstein was born in Sighet, Transylvania, Romania, on January 24, 1954. He received the B.Sc. and M.Sc. degrees in electrical engineering, from the Technion, Israel Institute of Technology, Haifa, in 1977 and 1980, respectively, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1984.

Since October 1984 he has been with the Faculty of Electrical Engineering at the Technion. His research interests are in estimation theory, signal and image processing, computer vision, algorithmic aspects of inverse scattering, point processes, and mathematical models in neurophysiology.

Dr. Bruckstein is a member of SIAM and AMS.