On the Detection of Dominant Points on Digital Curves

CHO-HUAK TEH, MEMBER, IEEE, AND ROLAND T. CHIN, MEMBER, IEEE

Abstract—**A parallel algorithm for detecting dominant points on a digital closed curve is presented. The procedure requires no input parameter and remains reliable even when features of multiple sizes are present on the digital curve. The procedure first determines the region of support for each point based on its local properties, then computes measures of relative significance (e.g., curvature) of each point, and finally detects dominant points by a process of nonmaxima suppression. This procedure leads to an important observation that the performance of dominant points detection depends not only on the accuracy of the measure of significance, but mainly precise determination of the region of support. This solves the fundamental problem of scale factor selection encountered in various dominant point detection algorithms. The inherent nature of scale-space filtering in the procedure is addressed and the performance of the procedure is compared to those of several other dominant point detection algorithms, using a number of examples.**

Index Terms—**Angle, chain code, corner, curvature, digital curve, dominant point, image processing, parallel algorithm, pattern classification, polygonal approximation, scale-space filtering, shape analysis, tangential deflection, vertex.**

I. INTRODUCTION

SINCE Attneave's famous observation that information on the shape of a curve is concentrated at dominant points having high curvature [1], a number of algorithms have been suggested for finding the extrema on a digital curve. In general, there are two approaches to the problem. One is to detect the dominant points directly through angle or corner detection schemes [2]–[8]. The other approach is to obtain a piecewise linear polygonal approximation of the digital curve subject to certain constraints on the goodness of fit [9], [10];[1] dominant points then correspond approximately to the actual or extrapolated intersections of adjacent line segments of the polygon. It is pointed out in [15] that piecewise linear polygonal approximation with variable breakpoints will tend to locate vertices as actual dominant points. In this paper, we shall restrict our discussion to the former approach. The latter approach, which is essentially that of performing side detection, can be considered as a dual notion to the former approach.

It is observed that any of the above algorithms, except [5], from either the former or latter approach, requires one or more input parameters. These parameters usually represent the region of support for the measurement of local properties (e.g., curvature) at each point on the curve. They are selected based on the level of detail represented by the digital curve. In general, it is difficult to find a set of parameters suitable for a curve that consists of multiple size features. Too large a region of support will smooth out the fine features of a curve, whereas a small region of support will generate a large number of redundant dominant points. This is a fundamental problem of scale because the features describing the shape of a curve vary enormously in size and extent, and there is seldom a well-defined basis for choosing an appropriate scale (or smoothing) parameter correspond to a particular feature size [16].

In this paper, we present a dominant point detection algorithm which requires no input parameter. Our algorithm is motivated by the Rosenfeld–Johnston angle detection procedure [2], in which both an incorrect region of support and incorrect curvature measures may be assigned to a point if the input smoothing parameter is not chosen correctly, and hence dominant points may be suppressed [17]. To overcome this problem, we propose that the region of support, and hence the corresponding scale or smoothing parameters, of each boundary point should be determined independently, based on its local properties. We further show that once the region of support of each point is determined, various measures of significance (e.g., curvature measures) of each point can be computed accurately for the detection of the dominant points. This leads to an important observation: the detection of dominant points relies not only on the accuracy of measures of significance, but primarily on the precise determination of the region of support. This finding is contrary to the notion of traditional dominant point detection algorithms, in which accurate determination of discrete curvature measures is believed to be the major factor in detecting reliable dominant points.

In Section II, we briefly review various dominant point detection algorithms. These include 1) the angle detection procedure by Rosenfeld and Johnston [2], 2) the improved angle detection procedure by Rosenfeld and Weszka [3], 3) the corner finding algorithm by Freeman and Davis [4],

Manuscript received July 6, 1987; revised February 8, 1988. Recommended for acceptance by O. Faugeras. This work was supported in part by the National Science Foundation under Grant ECS-8352356 and in part by General Motors Foundation, Inc., Dearborn, MI.

C.-H. Teh was with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706. He is now with the Department of Electrical Engineering, National University of Singapore, Singapore 0511, Republic of Singapore.

R. T. Chin is with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706.

IEEE Log Number 8928042.

[1]Most of the polygonal approximation algorithms have been surveyed in [9] and compared in [10]. Others are in [11]–[14].

0162-8828/89/0800-0859$01.00 © 1989 IEEE

4) the dominant point detection procedure by Sankar and Sharma [5], and 5) the vertex detection algorithm by Anderson and Bezdek [6]. In Section III, various measures of significance are defined and discussed; an algorithm to determine the local region of support, providing a basis for choosing the smoothing parameter is described; and the dominant point detection algorithm is presented. In Section IV, we compare our algorithm to the five algorithms mentioned above. We compare the performance of these algorithms using four digital closed curves with respect to 1) the number of detected dominant points, 2) the error introduced in approximating the closed curve by the polygon generated by joining the dominant points, and 3) the computational load. In Section V, the inherent nature of scale-space filtering of our algorithm is addressed, and a discussion and conclusions are presented.

II. Dominant Point Detection Algorithms

Let the sequence of n integer-coordinate points describe a closed curve C,

$$C = \{p_i = (x_i, y_i), i = 1, \cdots, n\} \tag{2.1}$$

where p_{i+1} is a neighbor of p_i (modulo n). The Freeman chain code of C consists of the n vectors

$$\vec{c}_i = \overline{p_{i-1}p_i} \tag{2.2}$$

each of which can be represented by an integer

$$f = 0, \cdots, 7 \tag{2.3}$$

as shown in Fig. 1 where $1/4\pi f$ is the angle between the X-axis and the vector. The chain of C is defined as $\{\vec{c}_i, i = 1, \cdots, n\}$ and $\vec{c}_i = \vec{c}_{i \pm n}$. All integers are modulo n.

Since dominant points on a curve correspond to points of high curvature, the various existing dominant point detection algorithms first compute an estimate of the curvature at each point on the curve. Next, a two-stage procedure is applied to choose dominant points. In the first stage, some input threshold is applied to the curvature estimates to eliminate those points whose curvature is absolutely too low to be considered as dominant points. In the second stage, a process of nonmaxima suppression is applied to the remaining points to eliminate any points whose curvature estimates are not local maxima in a sufficiently large segment of the curve. In this section, we briefly discuss five dominant point detection algorithms [2]–[6]. Table I gives a brief summary of these algorithms and our proposed algorithm. All the algorithms except [6] are parallel in nature, in the sense that the results at each point do not depend on the results obtained at other points.

A. Rosenfeld–Johnston Angle Detection Procedure (1973 [2])

This parallel procedure is analogous to the Rosenfeld–Thurston edge detection algorithm [18], which detects significant maxima in average gray-level gradients by using a variable degree of smoothing.

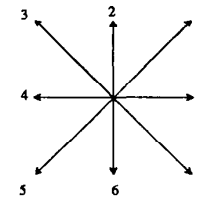

Fig. 1. Freeman code.

TABLE I
Comparison of Various Dominant Point Detection Algorithms

Algorithm	Year	Parallel(P)/ Sequential(S)	Number of Parameters	Measure of Significance	
Rosenfeld–Johnston	1973	P	1: m	cosine value	Eq. (2.5)
Rosenfeld–Weszka	1975	P	1: m	smoothed cosine value	Eq. (2.8)
Freeman–Davis	1977	P	2: s (5–13) m (1,2)	cornerity measure	Eq. (2.12)
Sankar–Sharma	1978	P	None	sum of local curvature	Eq. (2.14)
Anderson–Bezdek	1984	S	4: $\Delta\theta_i$ (5°–25°) m (5–15) ε_{p1} (0.7–0.9) d	tangential deflection	Eqs. (A.8) & (A.9)
Teh–Chin	1987	P	None	(*a*) k–cosine value, (*b*) k–curvature, or (*c*) 1–curvature	Eq. (2.5) Eq. (3.2) Eq. (3.3)

Rosenfeld–Johnston Procedure

1) Define the k vectors at p_i as

$$\vec{a}_{ik} = (x_i - x_{i+k}, y_i - y_{i+k}) \tag{2.4a}$$

$$\vec{b}_{ik} = (x_i - x_{i-k}, y_i - y_{i-k}) \tag{2.4b}$$

and the k cosine at p_i as

$$\cos_{ik} = \frac{\vec{a}_{ik} \cdot \vec{b}_{ik}}{|\vec{a}_{ik}|\,|\vec{b}_{ik}|}. \tag{2.5}$$

Here $\cos_{ik}$ is the cosine of the angle between the k vectors $\vec{a}_{ik}$ and $\vec{b}_{ik}$, so that $-1 \le \cos_{ik} \le 1$, and $\cos_{ik} = +1$ for the sharpest angle (0°), and -1 for a straight line (180°).

2) Select a smoothing factor m (either $n/10$ or $n/15$ [2]) based on the level of detail of C. At each point p_i, compute $\{\cos_{ik}, k = 1, \cdots, m\}$.

3) Assign region of support h_i and curvature value $\cos_{i,h_i}$ to point p_i for the largest h such that

$$\cos_{im} < \cos_{i,m-1} < \cdots < \cos_{i,h_i} \ge \cos_{i,h_i-1}. \tag{2.6}$$

4) Retain those points p_i where $\cos_{i,h_i} \ge \cos_{j,h_j}$ for all j such that

$$|i - j| \le h_i/2, \tag{2.7}$$

as the curvature maxima.

B. Rosenfeld–Weszka Improved Angle Detection Procedure (1975 [3])

The procedure described earlier is analogous to the edge detection algorithm developed in [18]. However, this algorithm can lead to incorrect results when edges occur too close to one another [19]. A modified algorithm, developed in [20], has overcome this difficulty. This improved detection procedure is an analog of the modified edge de-

tection algorithm in [20]. Specifically, after step 2) of the Rosenfeld-Johnston procedure, the k cosines ($k > 1$) at each point are smoothed as follows:

$$\overline{\cos_{ik}} = \frac{2}{k+2} \sum_{j=k/2}^{k} \cos_{ij} \qquad \text{for } k = \text{even} \tag{2.8a}$$

$$= \frac{2}{k+3} \sum_{j=(k-1)/2}^{k} \cos_{ij} \qquad \text{for } k = \text{odd}. \tag{2.8b}$$

The $\{\overline{\cos_{ik}}\}$ are then treated just like the $\{\cos_{ik}\}$ after step 2).

C. Freeman-Davis Corner Finding Algorithm (1977 [4])

This algorithm starts with scanning the chain $\{\vec{c}_i, i = 1, \cdots, n\}$ with a moving straight line segment which connects the end points of a sequence of s links. As the line segment moves from one chain node to the next, the angular differences between successive segment positions are used as a smoothed measure of local curvature along the chain.

Freeman-Davis Algorithm

1) Define L_{is} as the straight line segment spanning s chain links and terminating on the node to which link $\vec{c}_i$ is directed: $L_{is} = \{\vec{c}_j, j = i - s + 1, \cdots, i\}$. The x and y components of L_{is} are given by

$$X_{is} = \sum_{j=i-s+1}^{i} c_{jx} \tag{2.9a}$$

$$Y_{is} = \sum_{j=i-s+1}^{i} c_{jy} \tag{2.9b}$$

where the c_{jx} and c_{jy} are the x and y components, respectively, of the chain link $\vec{c}_j$, and $c_{jx}, c_{jy} \in \{-1, 0, 1\}$. The angle L_{is} makes with the X-axis is given by

$$\theta_{is} = \tan^{-1} \frac{Y_{is}}{X_{is}}, \qquad \text{if } |X_{is}| \geq |Y_{is}| \tag{2.10a}$$

$$= \cot^{-1} \frac{X_{is}}{Y_{is}}, \qquad \text{if } |X_{is}| < |Y_{is}|. \tag{2.10b}$$

2) Define the incremental curvature δ_{is} as twice the mean over two adjacent angular differences

$$\delta_{is} = \theta_{i+1,s} - \theta_{i-1,s}. \tag{2.11}$$

The incremental curvature is a smoothed measure of curvature; the greater the s, the heavier the smoothing. For a well-quantized curve, s will range normally from a minimum of 5 to a maximum of 13.

3) A corner is characterized by three incremental curvature regions—two for which δ_{is} fluctuates within relatively narrow limits due to quantization noise, separated by a third region consisting of precisely $s + 1$ nodes in which $\Sigma\, \delta_{is}$ equals a significant value. Specifically, define the cornerity at node i by[2]

$$K_i = \sqrt{t_{i1}} \times \sum_{j=i}^{i+s} \delta_{js} \times \sqrt{t_{i2}} \tag{2.12a}$$

where

$$t_{i1} = \max\left\{t : \delta_{i-v,s} \in (-\Delta, \Delta), \text{ for all } 1 \leq v \leq t\right\} \tag{2.12b}$$

$$t_{i2} = \max\left\{t : \delta_{i+s+v,s} \in (-\Delta, \Delta), \text{ for all } 1 \leq v \leq t\right\} \tag{2.12c}$$

and

$$\Delta = \tan^{-1}\left(\frac{1}{s-m}\right) \tag{2.12d}$$

with

$$m = 1 \quad \text{or} \quad 2. \tag{2.12e}$$

4) Retain only those points p_i where $|K_i| \geq |K_j|$ for all j such that

$$|i - j| \leq s, \tag{2.13}$$

as the corner points.

D. Sankar-Sharma Dominant-Point Detection Procedure (1978 [5])

In this procedure, the dominant points are computed iteratively as the points of maximum global curvature, based on the local curvature of each point with respect to its immediate neighbors. First it is observed that each point of a closed curve having exactly two neighbors can be classified into three classes based on the local curvature, as shown in Table II. For those points having more than two neighbors, let (i, j) be the point with k immediate neighbors where $k \geq 3$. Find all possible pairs of 2-neighbor configurations of (i, j), viz., a total of $\binom{k}{2}$ combinations. Assign to each such pair the corresponding local curvature, using Table II. From this collection, omit those pairs with zero curvature. Among the rest, if all the pairs have been assigned positive (negative) curvature, then the point (i, j) is assigned the weight $+1$ (-1); on the other hand, if some pairs have positive curvature while the rest have negative curvature, then the point (i, j) is assigned the weight 0.

Sankar-Sharma Procedure

Let $P(i, j)$ refer to the curvature value at the (i, j)th point on the picture plane.

NDOMINATE $\leftarrow n$, the total number of points

DOLD $\leftarrow \varnothing$, *the empty set.*

1) Assign the numerical weight to each point (i, j) as $P(i, j) \leftarrow 0$, or $+1$ or -1 using Table II or an algebraic combination of them.

[2]The square roots in (2.12a) can be replaced by logarithms [4].

TABLE II
LOCAL CURVATURE ASSIGNMENT

Type Of Curvature	Weight Assigned
No curvature	0
Positive curvature	+1
Negative curvature	-1

2) For each point assign a new weight as

$$P_{\text{new}}(i, j) \leftarrow \sum_{s=i-1}^{i+1} \sum_{t=j-1}^{j+1} P_{\text{old}}(s, t). \qquad (2.14)$$

3) Detect the dominant points as the set D defined as

$$D = \{(i, j) \text{ such that } |P(i, j)| \geq |P(s, t)| \text{ for } s = i - 1 \text{ to } i + 1,\ t = j - 1 \text{ to } j + 1\}. \qquad (2.15)$$

If $NDOMINATE > |D|$, the number of currently detected dominant points,
Set $NDOMINATE \leftarrow |D|$; $DOLD \leftarrow D$
redo step 2)
else stop
DOLD is the required set of dominant points.

E. Anderson–Bezdek Vertex Detection Algorithm (1984 [6])

All the methods discussed so far utilize various schemes for approximating discrete curvature. In [6], tangential deflection and curvature of discrete curves are defined based on the geometrical and statistical properties associated with the eigenvalue–eigenvector structure of sample covariance matrices. Specifically, it has been proven in [6] that the nonzero entry of the commutator of a pair of scatter matrices constructed from discrete arcs is related to the angle between their eigenspaces, and the entry (in certain limiting cases) is also proportional to the analytical curvature of the plane curve from which the discrete data are drawn. The sequential algorithm then identifies the location of vertices of the discrete curves based on excessive cumulative tangential deflection between successive vertices, an approach which differs markedly from all the previous approaches of searching for points of relative curvature extrema.

Anderson–Bezdek Algorithm

Inputs: Four input parameters are required.

1) A tangential deflection angle threshold $\Delta\theta_t$, typically in the range, $5° < \Delta\theta_t < 25°$.

2) A smoothing factor m, typically in the range, $5 \leq m \leq 15$.

3) A baseline acceptability parameter ϵ_{b1}, typically in the range, $0.7 < \epsilon_{b1} < 0.9$.

4) A minimum number of data points d between successive baseline vertices.

Outputs: The number of vertices c, their index labels v_i, $i = 1, \cdots, c$, and a designation of each vertex as a baseline threshold or tangential deflection threshold vertex.

Start-Up: Call the first data point a temporary vertex, set $v_0 = 1$, $i = 0$, and $c = 0$.

Steps:

1a) From the ith vertex with label v_i set $l = v_i$.

1b) If $l > n$, stop. Otherwise, define the provisional baseline arc $X(l)$, from which subsequent tangential deflections will be computed as

$$X(l) = \{(x_j, y_j) | j = l, l + 1, \cdots, l + m - 1\},$$

and compute its associated trace-normalized scatter matrix A [see Appendix A].

1c) If $D_X = \sqrt{1 - 4 \det A}$ satisfies $D_X < \epsilon_{b1}$, then in accordance with (A.4) of Appendix A, the arc $X(l)$ is not sufficiently linear in shape, and the provisional baseline $X(l)$ is *unacceptable*; set $l = l + 1$ and return to Step 1b). If $D_X \geq \epsilon_{b1}$, designate $X(l)$ as the *accepted baseline* arc X_{b1}.

1d) If the current value of l exceeds v_i by more than the minimum number d, label the point l as the $(i + 1)$st vertex, designate this vertex as a baseline threshold vertex, put $c = c + 1$ and $i = i + 1$.

2) Starting with the point $k = v_i + 1$, define the subsequent arc $Y(k)$ as

$$Y(k) = \{(x_j, y_j) | j = k, k + 1, \cdots, k + m - 1\},$$

and compute its associated trace-normalized scatter matrix B [see Appendix A].

3) Compute the tangential deflection $\Delta\theta(X_{b1}, Y(k))$ using (A.8) and (A.9) of Appendix A.

4a) If $\cos(2\Delta\theta) > \cos(2\Delta\theta_t)$, set $k = k + 1$. If $k > n + 1$, stop. If $k \leq n + 1$, return to Step 2).

4b) If $\cos(2\Delta\theta) < \cos(2\Delta\theta_t)$, then a vertex should be placed in the vicinity of the kth point. To this end, compute the tangential deflections, $\tau_j = \cos(\Delta\theta(Y(j - m), Y(j)))$, again by (A.8) and (A.9) for $j = k, k + 1, \cdots, k + m - 1$. Locate the $(i + 1)$st vertex v_{i+1} at j where j is the first point $j = k, \cdots, k + m - 1$ where the difference $\tau_{j+1} - \tau_j$ changes sign. If this difference never changes sign, set $v_{i+1} = k + [m/2]$. Designate this vertex as a *tangential deflection* vertex, set $c = c + 1$, $i = i + 1$ and return to Step 1).

III. TEH–CHIN DOMINANT POINT DETECTION ALGORITHM

A. Discrete Curvature Measurement

There are two major problems with dominant point detection on digital curves. One is the precise definition of discrete curvature, the other is the determination of the region of support for the computation of the curvature. In the real Euclidean plane, curvature is defined as the rate of change of slope as a function of arc length. For the curve $y = f(x)$, this can be expressed in terms of deriv-

atives as

$$\frac{\frac{d^2y}{dx^2}}{\left[1+\left(\frac{dy}{dx}\right)^2\right]^{3/2}}. \tag{3.1}$$

For a digital curve, if the discrete curvature is defined by simply replacing the derivatives in (3.1) by first differences, there is a problem that small changes in slope are impossible, since successive slope angles on the digital curve can differ only by a multiple of 45°. This difficulty is overcome in various dominant point detection algorithms discussed in Section II by using $k > 1$ differences, rather than by simply using the first differences ($k = 1$). In other words, a smoothed version of discrete curvature is measured, and k can be viewed as a smoothing parameter. Another way to overcome this problem is to use higher order chain codes [21] where the directions are quantized in more than eight steps.

A number of other authors have also concentrated on techniques involving direct measurements of discrete curvature or functions of discrete curvature [6], [22]–[24]. These measurements are used in various dominant point detection algorithms to detect dominant points in the final steps of nonmaxima suppression. We shall call these functions measures of significance [25], $S(\cdot)$. Hence, the k cosine measures [(2.5) and (2.8)] in Rosenfeld–Johnston and Rosenfeld–Weszka, the cornerity measure [(2.12)] in Freeman–Davis, the weighted curvature measure [(2.14)] in Sankar–Sharma, and the tangential deflection measure [(A.8) and (A.9)] in Anderson–Bezdek algorithms are all different types of measures of significance.

Since an approximate smoothed version of discrete curvature is measured in various algorithms, an appropriate smoothing factor (for example, m in the case of the Rosenfeld–Johnston procedure) has to be selected based on the level of detail represented in the digital curve in order to measure the discrete curvature to a certain degree of accuracy. This smoothing factor is in fact a function of the region of support which is used to compute the measure of significance. In general, the higher the level of detail, the smaller the smoothing factor to be selected. A major difficulty arises when a digital curve has features at various levels of detail. Here, if too large a smoothing factor is selected, some dominant points of fine features will be missed. On the other hand, if too small a smoothing factor is selected, extra nondominant points of coarse features may be detected, which constitutes redundancy. Both cases result in inaccuracy of dominant point detection. It has been remarked in [2] that the user of these procedures has to select a smoothing factor appropriate to the class of curves to be processed. However, the difficulty still remains, because curves usually consist of features of multiple sizes and this can only be overcome by using different smoothing factors for regions having different levels of detail.

In the next section, we show that the reliability and accuracy of dominant point detection depend not only on the accurate determination of discrete curvature, but primarily on the accurate determination of the smoothing factor of each point based on the local properties of that region. Our proposed algorithm requires no input smoothing parameter. The smoothing factor (or the region of support) of each point p_i of the curve, which varies from point to point, is determined based on the local properties of p_i. The determination of regions of support has also been referred to as the determination of domain $D(\cdot)$ [25]. We should point out that although the Sankar–Sharma procedure also requires no input smoothing parameters, it does not involve any determination of regions of support and the results are no better than those of any of the other methods tested (see Section IV).

In order to show and emphasize the implication of the determination of the region of support, we use the following three different measures of significance, which correspond to different degrees of accuracy of discrete curvature measures.

a) k cosine measure in Rosenfeld–Johnston procedure: $\cos_{ik}$, (2.5).

b) k curvature measure: the difference in mean angular direction of k vectors [(2.2)] on the leading and trailing curve segment of the point p_i where the curvature is measured [24], i.e.,

$$CUR_{ik} = \frac{1}{k}\sum_{j=-k}^{-1} f_{i-j} - \frac{1}{k}\sum_{j=0}^{k-1} f_{i-j} \tag{3.2}$$

where f_{i-j} is an integer defined in (2.3) and Fig. 1.

c) 1 curvature measure: ($k = 1$ of b) above)

$$CUR_{i1} = f_{i+1} - f_i. \tag{3.3}$$

B. Determination of Region of Support

The determination of the region of support of a point p_i, $D(p_i)$, constitutes the major problem in various dominant point detection algorithms. $D(p_i)$ in the Rosenfeld–Johnston procedure is determined in Step 3) by (2.6) and $1 \le D(p_i) = h_i \le m$. For the case of the Freeman–Davis algorithm, $D(p_i) = s$. Both m and s are input smoothing parameters. Davis [17] has pointed out that incorrectly chosen regions of support may cause the measures of significance to be computed over inappropriate neighborhoods which may subsequently cause dominant points to be discarded. The Rosenfeld–Weszka procedure deals with this problem by presmoothing the k cosines by averaging [see (2.8)]. However, it does not provide much better results.

The problem of the determination of regions of support has also been addressed by a number of other authors. Langridge [26] pointed out that each boundary point of a closed curve should have its own view of the curve. A dominant point should have a view which constitutes a meaningful region of support of the curve and should block the view from neighboring nondominant points. The author suggested that if we are trying to find whether a

point dominates a section of the boundary, a possible constraint is that the point must be at a maximum height relative to the chord that joins the ends of the boundary fragment. However, while he noted that a criterion limiting how far around the curve this procedure should go is necessary, he did not suggest what criterion should be used. Rosenberg [25] noted that certain points of a convex blob perceptually dominate other points of the blob. He further presented methods for the determination of regions of support with specific reference to convex blobs, and suggested that nonconvex blobs can be analyzed by decomposing them into convex blobs. However, this approach seems unreasonable and has been criticized by Davis [17].

We now present a simple iterative algorithm to determine the region of support of a point p_i of a closed curve.

Determination of Region of Support

1) Define the length of the chord joining the points p_{i-k} and p_{i+k} as

$$l_{ik} = \left| \overline{p_{i-k}p_{i+k}} \right|. \tag{3.4}$$

Let d_{ik} be the perpendicular distance of the point p_i to the chord $\overline{p_{i-k}p_{i+k}}$.

2) Start with $k = 1$. Compute l_{ik} and d_{ik} until[3]

$$\text{(a) } l_{ik} \geq l_{i,k+1} \tag{3.5a}$$

or

$$\text{(b) } \frac{d_{ik}}{l_{ik}} \geq \frac{d_{i,k+1}}{l_{i,k+1}} \quad \text{for } d_{ik} > 0 \tag{3.5b}$$

$$\frac{d_{ik}}{l_{ik}} \leq \frac{d_{i,k+1}}{l_{i,k+1}} \quad \text{for } d_{ik} < 0. \tag{3.5c}$$

Then the region of support of p_i is the set of points which satisfy either condition (a) or condition (b),

$$D(p_i) = \{(p_{i-k}, \cdots, p_{i-1}, p_i, p_{i+1}, \cdots, p_{i+k}) \mid \text{condition (a) or condition (b)}\}. \tag{3.6}$$

As discussed earlier, one of the problems is the accurate detection of dominant points when they occur too close to one another. It can be shown that condition (a) in (3.5a) alone will overcome the problem. In Fig. 2(a) (Fig. 4 of [2]), for example, a region of support of l and a cosine value of -1 will be assigned to p using condition (a) alone; both are correct. Similarly, in Fig. 2(b) (Fig. 2 of [17]), a region of support of l will be assigned to both i and j, while cosine values of cos (I) and cos (J) will be assigned to i and j, respectively, all are correct. Now consider the case of a circle in Fig. 2(c). Every point on the circle will be assigned a region of support of diameter l and a cosine value of 0 using condition (a) alone, resulting in every point being detected as a dominant point.

[3]Since d_{ik} is signed, condition (b) is not used if $d_{ik} = 0$. As soon as $d_{ik} \neq 0$, (3.5b) will be used if $d_{ik} > 0$, and (3.5c) will be used if $d_{ik} < 0$.

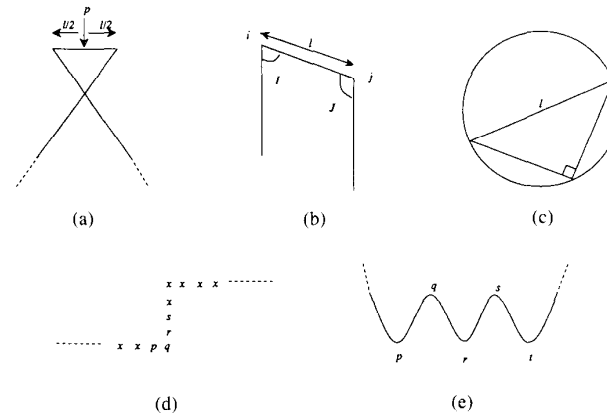

Fig. 2. (a) A region of support of l and a cosine value of -1 is assigned to p using condition (a) in (3.5a). (b) A region of support of l is assigned to both i and j, and cosine values of cos (I) and cos (J) are assigned to i and j, respectively, using condition (a) in (3.5a). (c) A circle of diameter l. (d) Regions of support of 4, 1, and 2 are assigned to p, q, and r, respectively, using condition (b) in (3.5b) and (3.5c). (e) A demonstration of the usage of condition (b) in (3.5b) and (3.5c)

This makes sense for a continuous circle, but for a digital circle (say, of high resolution) there will be too many redundant points.[4] To overcome this, condition (b) is used to limit how far around the curve condition (a) should be applied, a condition which is not made specific by Langridge. In fact in Fig. 2(d) (from [3]), p, q, and r will be assigned regions of support of 4, 1, and 2, respectively, using condition (b) alone, which are argued in [3] to be having "best regions of support" of 4, 3, and 2, respectively. Intuitively, q should be assigned a best region of support of 1, since if small k's give the same cosine value as large ones, the angle must be sharp [27]; therefore the smallest k (1, not 3) should be taken as the best region of support of q. For the type of curve in Fig. 2(e), condition (b) but not condition (a) is necessary to ensure the correct detection of p, q, r, s, and t as dominant points. Hence, in order to ensure correct detection of dominant points for a general curve, both conditions (a) and (b) must be used.

C. Teh–Chin Dominant Point Detection Algorithm

A measure of significance, for example curvature which involves computing derivatives, must be determined over some region of support; however, there is rarely a sound basis for choosing this parameter. The chord length and the perpendicular distance introduced in the previous session provides a basis for choosing the appropriate region of support. The measure of significance of each point is determined by using the neighboring points within the extent. The measure of significance and the region of support of each point are then used to guide the selection of points to be removed. The points remaining after the removing process are the dominant points. We now present our dominant point detection algorithm.

[4]See Figs. 6(a)–6(l) in Section IV for examples on digital circles.

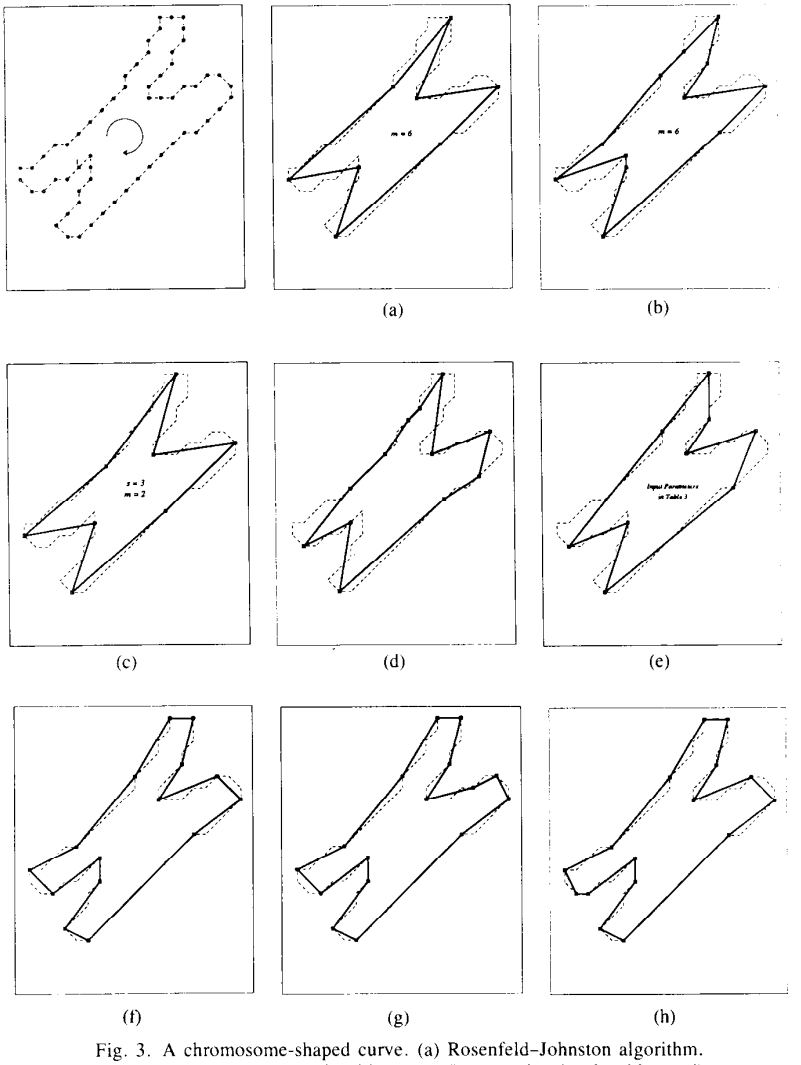

Fig. 3. A chromosome-shaped curve. (a) Rosenfeld-Johnston algorithm. (b) Rosenfeld-Weszka algorithm. (c) Freeman-Davis algorithm. (d) Sankar-Sharma algorithm. (e) Anderson-Bezdek algorithm. (f) Teh-Chin algorithm (k cosine). (g) Teh-Chin algorithm (k curvature). (h) Teh-Chin algorithm (1 curvature).

Teh-Chin Algorithm

1) Determine the region of support of each point by the algorithm described in Section III-B,

$$D(p_i) = \{p_{i-k_i}, \cdots, p_{i-1}, p_i, p_{i+1}, \cdots, p_{i+k_i}\}.$$

2) Select a measure of significance (e.g., from one of the three defined at the end of Section III-A) and calculate its absolute value for each point, $|S(p_i)|$.

3a) 1st pass: Perform nonmaxima suppression as follows; retain only those points p_i where

$$|S(p_i)| \geq |S(p_j)| \tag{3.7a}$$

for all j such that

$$|i - j| \leq k_i/2. \tag{3.7b}$$

3b) 2nd pass: Further suppress those points having zero 1 curvature ($CUR_{i1} = 0$).

3c) 3rd pass: For those points survived after 2nd pass,
if ([k_i of $D(p_i)$] = 1) and (p_{i-1} or p_{i+1} still survived) then
further suppress p_i if ($|S(p_i)| \leq |S(p_{i-1})|$) or ($|S(p_i)| \leq |S(p_{i+1})|$).
if 1 curvature is selected as a measure of significance,
then go to Step 3d) and do a 4th pass
else those points survived are the dominant points.

3d) 4th pass: For those groups of more than 2 points that still survived, suppress all the points except the two end points of each of the groups.
For those groups of exactly 2 points that still survived,
if ($|S(p_i)| > |S(p_{i+1})|$) then
suppress p_{i+1}

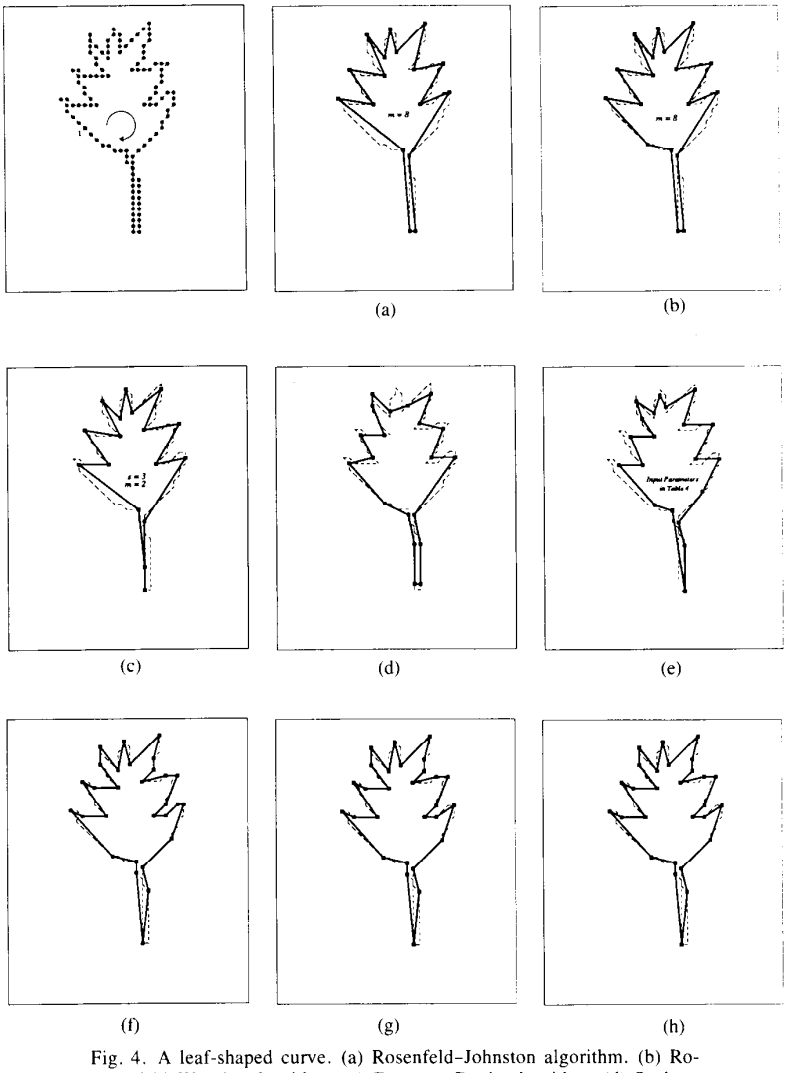

Fig. 4. A leaf-shaped curve. (a) Rosenfeld–Johnston algorithm. (b) Rosenfeld–Weszka algorithm. (c) Freeman–Davis algorithm. (d) Sankar–Sharma algorithm. (e) Anderson–Bezdek algorithm. (f) Teh–Chin algorithm (k cosine). (g) Teh–Chin algorithm (k curvature). (h) Teh–Chin algorithm (1 curvature).

else if ($|S(p_i)| < |S(p_{i+1})|$) then
  suppress p_i
else if ($k_i > k_{i+1}$) then
  suppress p_{i+1}
else suppress p_i.

Remarks:

- The 2nd pass in Step 3b) is equivalent to a sort of "fine tuning" adjustment in the positions of some dominant points. It can be skipped if not desired; a few dominant points may appear one point away from where they should be.
- For the case when 1 curvature is selected as a measure of significance, the 4th pass in Step 3d) can be simplified by simply suppressing p_{i+1} for groups of exactly 2 points that still survived.

Steps 1), 2), and 3a) in the algorithm can be processed in parallel, and Steps 3b), 3c), and 3d) are processed sequentially, starting from the first contour point. The number of points survived after the process of nonmaxima suppression in Step 3a) is usually only a small fraction of the total input contour points. Thus, the sequential processing in Step 3) [i.e., Steps 3b)–3d)] is relatively inexpensive. This can be seen from the result in Section IV that the CPU processing time of our algorithm implemented sequentially is comparable to, and in some cases even much shorter than, those of the rest of the algorithms. The second pass in Step 3b) is essentially used to eliminate points having zero curvature. The third pass in Step 3c) is needed to take care of points having regions of support of 1.

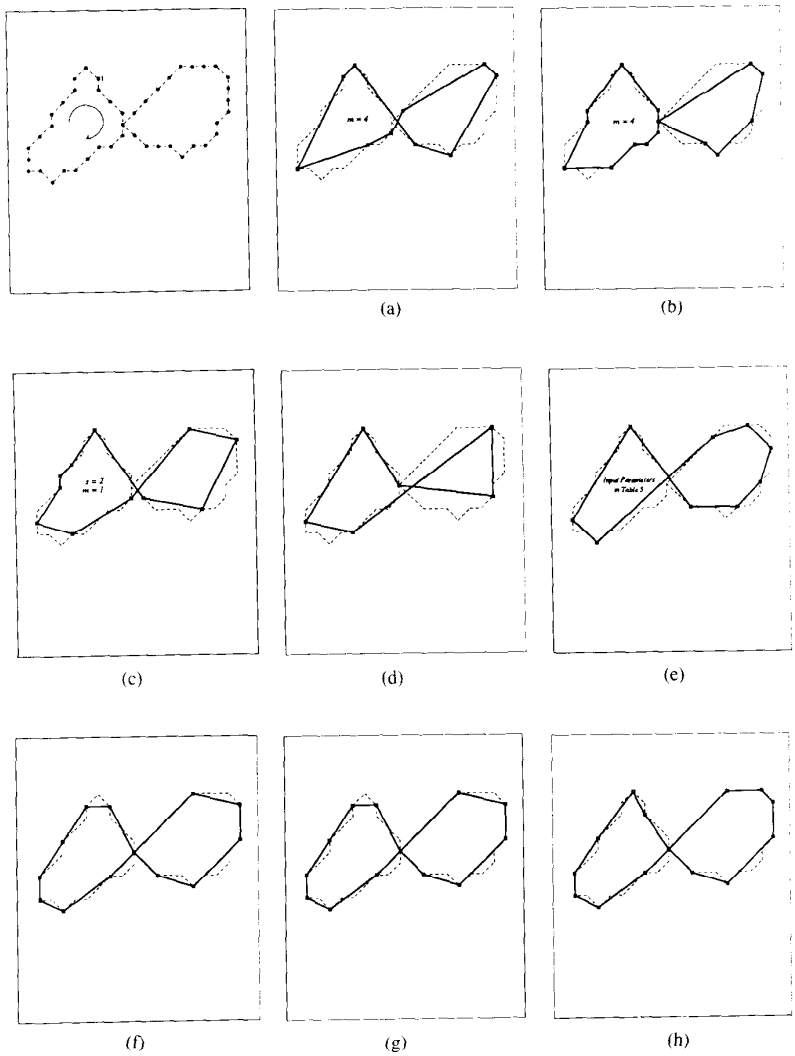

Fig. 5. A figure-8 curve. (a) Rosenfeld-Johnston algorithm. (b) Rosenfeld-Weszka algorithm. (c) Freeman-Davis algorithm. (d) Sankar-Sharma algorithm. (e) Anderson-Bezdek algorithm. (f) Teh-Chin algorithm (k cosine). (g) Teh-Chin algorithm (k curvature). (h) Teh-Chin algorithm (1 curvature).

IV. Experimental Results

Dominant points on a digital curve can be used as a compact and effective representation of the curve for shape analysis and pattern classification [9], [15]. The polygon drawn by joining the adjacent dominant points is used to approximate the shape of the object. A quantitative measure of the quality of the detected dominant points is used in this paper, defined as the pointwise error between the digital curve and the approximating polygon. The error between a point p_i of a digital closed curve C, as defined in (2.1), and the approximating polygon is defined as the perpendicular distance of the point to the approximating line segment. We denote this error by e_i. Two error norms between C and its approximating polygon can be defined

(a) Integral square error,

$$E_2 = \sum_{i=1}^{n} e_i^2 \tag{4.1}$$

(b) Maximum error,

$$E_\infty = \max_{1 \le i \le n} e_i. \tag{4.2}$$

In this section, we compare our algorithm with the five algorithms discussed in Section II. Four closed curves were chosen to compare the algorithms with respect to 1) the number of detected dominant points, 2) the approximation errors E_2 and E_∞, and 3) the total CPU processing time. Fig. 3 is a chromosome-shaped curve with 60 vertices, Fig. 4 is a leaf-shaped curve with 120 vertices, Fig. 5 is a figure-8 curve with 45 vertices, and Fig. 6 shows a

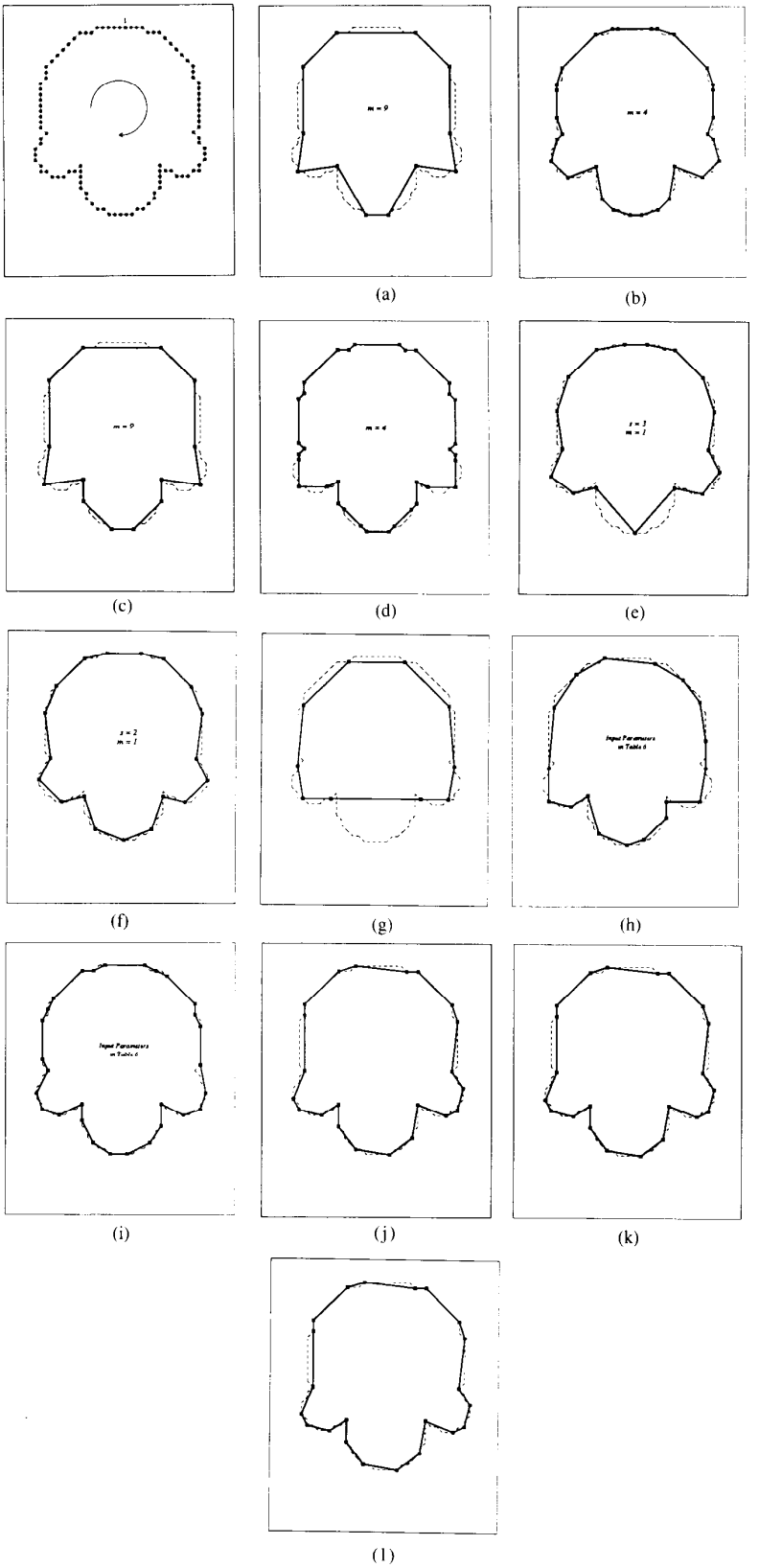

Fig. 6. A curve with four semicircles. (a) Rosenfeld-Johnston algorithm. (b) Rosenfeld-Johnston algorithm. (c) Rosenfeld-Weszka algorithm. (d) Rosenfeld-Weszka algorithm. (e) Freeman-Davis algorithm. (f) Freeman-Davis algorithm. (g) Sankar-Sharma algorithm. (h) Anderson-Bezdek algorithm. (i) Anderson-Bezdek algorithm. (j) Teh-Chin algorithm (k cosine). (k) Teh-Chin algorithm (k curvature). (l) Teh-Chin algorithm (1 curvature).

curve consisting of three semi-circles having different radii, and has 102 vertices. The first three figures are taken from [2] and the last figure represents a typical example of a curve which has features of multiple sizes. The chain code of each curve, which constituted the input to all the algorithms, was coded in a clockwise direction starting from the point marked with □ on each curve, using the Freeman code defined in Fig. 1. The chain codes of all the curves can be found in Appendix B. All algorithms were implemented in Pascal and executed on a VAX 750 computer. All algorithms have the same input and output format,[5] and process data in the clockwise direction.

The results obtained by applying the six algorithms to Fig. 3 are shown in Figs. 3(a)–3(f), which are also summarized in Table III. The processing of Figs. 3–5 requires one set of input parameter each; these are listed in their corresponding tables. The processing of Fig. 6 uses two sets of input parameters to demonstrate the problem of varying feature size. Tables III–VI show the results corresponding to Figs. 3–6. Several observations are drawn from the results.

1) CPU Processing Time: The Sankar–Sharma algorithm is the most expensive one because of the iterative nature of the algorithm. The Rosenfeld–Weszka algorithm is relatively expensive because it takes an additional step of k cosine smoothing. There is little difference in computational loading among the rest of the four algorithms. It should be noted that the processing time of those algorithms requiring input parameters depends on the values of the parameters. In the Teh–Chin algorithm, three different measures of significance are used; of these, the 1 curvature measure is the least expensive.

2) Approximation Errors: Our algorithm consistently outperforms the rest of the algorithms in terms of both approximation error measures. The approximation errors obviously depend on the number of detected dominant points, n_d. The data compression ratios of the total number of input contour points to the number of detected dominant points, n/n_d, are tabulated in the tables.

3) Number of Detected Dominant Points: Of the two algorithms which require no input parameters, our algorithm consistently detects more dominant points than the Sankar–Sharma algorithm does. This is because of the iterative approach of the Sankar–Sharma algorithm, in which the iteration process terminates when the number of detected dominant points converges to a minimum. This final set of minimal number of detected dominant points, however, does not necessary correspond to an accurate representation of the curve, as can be seen in Figs. 4(d) and 6(g). In fact, this is also the cause of its large computational load and large approximation errors. Obviously, the termination criterion is ad hoc and some other criteria, perhaps the approximation errors, could be used.

The number of dominant points detected by the rest of the four algorithms depends on the input parameters. In general, the larger the region of support introduced by the input parameters, the lower the number of dominant points detected. Table VI shows the results of Fig. 6, which depicts a curve consisting of three regions having features of multiple sizes. The upper semicircle represents a coarse feature and requires a large region of support to avoid

[5]For the case of the Sankar–Sharma procedure, an additional set of input data has to be provided to differential the interior and exterior of the closed curve.

TABLE III
RESULTS OF THE CHROMOSOME-SHAPED CURVE IN FIG. 3

Number of Input Contour Points, n = 60							
Algorithm	Input Parameters	Number of Dominant Pts n_d	Compression Ratio n/n_d	Max Error E_∞	Integral Sq Error E_2	CPU Time in Secs t
(1) *Rosenfeld–Johnston*	m = 6	8	7.5	1.54	21.94	4.19
(2) *Rosenfeld–Weszka*	m = 6	12	5.0	1.58	22.61	6.04
(3) *Freeman–Davis*	s = 3, m = 2	8	7.5	1.51	22.56	3.12
(4) *Sankar–Sharma*	None	12	5.0	2.03	28.89	16.46
(5) *Anderson–Bezdek*	$\Delta\theta_t$ = 15°, m = 4, ε_{b1} = 0.75, d = 3	9	6.7	2.03	26.50	5.31
(6) *Teh–Chin* :						
(a) k–cosine	None	15	4.0	0.74	7.20	4.47
(b) k–curvature	None	16	3.8	0.71	5.91	5.09
(c) 1–curvature	None	16	3.8	0.74	6.40	4.03

TABLE IV
RESULTS OF THE LEAF-SHAPED CURVE IN FIG. 4

Number of Input Contour Points, n = 120						
Algorithm	Input Parameters	Number of Dominant Pts n_d	Compression Ratio n/n_d	Max Error E_∞	Integral Sq Error E_2	CPU Time in Secs t
(1) *Rosenfeld–Johnston*	m = 8	17	7.1	1.76	43.42	10.91
(2) *Rosenfeld–Weszka*	m = 8	18	6.7	1.53	30.57	17.40
(3) *Freeman–Davis*	s = 3, m = 2	17	7.1	1.72	45.27	8.16
(4) *Sankar–Sharma*	None	20	6.0	3.48	71.15	44.63
(5) *Anderson–Bezdek*	$\Delta\theta_t$ = 15°, m = 5, ε_{b1} = 0.75, d = 3	20	6.0	1.49	39.18	13.19
(6) *Teh–Chin* :						
(a) k–cosine	None	29	4.1	0.99	14.96	9.70
(b) k–curvature	None	28	4.3	0.99	15.34	12.43
(c) 1–curvature	None	28	4.3	0.99	15.43	9.92

TABLE V
RESULTS OF THE FIGURE-8 CURVE IN FIG. 5

Number of Input Contour Points, n = 45						
Algorithm	Input Parameters	Number of Dominant Pts n_d	Compression Ratio n/n_d	Max Error E_∞	Integral Sq Error E_2	CPU Time in Secs t
(1) *Rosenfeld–Johnston*	m = 4	10	4.5	1.61	22.83	2.97
(2) *Rosenfeld–Weszka*	m = 4	16	2.8	1.59	12.67	3.74
(3) *Freeman–Davis*	s = 2, m = 1	11	4.1	1.34	14.61	2.80
(4) *Sankar–Sharma*	None	6	7.5	2.36	38.57	11.55
(5) *Anderson–Bezdek*	$\Delta\theta_t$ = 15°, m = 4, ε_{b1} = 0.75, d = 3	9	5.0	1.11	8.97	4.49
(6) *Teh–Chin* :						
(a) k–cosine	None	13	3.5	1.00	5.93	3.99
(b) k–curvature	None	13	3.5	1.00	5.93	4.31
(c) 1–curvature	None	14	3.2	0.71	4.15	3.45

TABLE VI
RESULTS OF THE CURVE IN FIG. 6

Number of Input Contour Points, n = 102							
Algorithm		Input Parameters	Number of Dominant Pts n_d	Compression Ratio n/n_d	Max Error E_∞	Integral Sq Error E_2	CPU Time in Secs t
(1) *Rosenfeld–Johnston*	(a)	m = 9	12	8.5	2.04	92.37	10.81
	(b)	m = 4	30	3.4	0.74	8.85	8.09
(2) *Rosenfeld–Weszka*	(a)	m = 9	14	7.3	1.56	59.12	15.29
	(b)	m = 4	34	3.0	1.00	15.40	10.13
(3) *Freeman–Davis*	(a)	s = 3, m = 1	17	6.0	2.54	79.53	7.65
	(b)	s = 2, m = 1	19	5.4	1.41	23.31	9.46
(4) *Sankar–Sharma*		None	10	10.2	8.00	769.53	35.04
(5) *Anderson–Bezdek*	(a)	$\Delta\theta_t$ = 15°, m = 7, ε_{b1} = 0.75, d = 3	18	5.7	1.64	36.14	12.85
	(b)	$\Delta\theta_t$ = 15°, m = 3, ε_{b1} = 0.75, d = 3	29	3.5	1.18	6.43	12.04
(6) *Teh–Chin* :							
(a) k–cosine		None	22	4.6	1.00	20.61	9.66
(b) k–curvature		None	22	4.6	1.00	20.61	11.81
(c) 1–curvature		None	22	4.6	1.00	20.61	9.40

detecting redundant nondominant points. The two semicircles on the left and right represent finer features and hence require a smaller region of support to detect accurate dominant points. Fig. 6(a) shows that for the case of the Rosenfeld–Johnston algorithm, a large region of support performs reasonably well on the upper semicircle, but not on the other three smaller semicircles, where it misses dominant points. On the other hand, Fig. 6(b) shows that by using a smaller region of support, the dominant points of the smaller semicircles are detected accurately, but a large number of redundant nondominant points are detected on the big semicircle. This same phenomenon can also be observed in Figs. 6(c)–6(i).

The above problem is overcome by our algorithm by varying the region of support automatically based on the level of detail in the local region. Figs. 6(j)–6(l) show the results; no dominant points are missed nor are redundant nondominant points detected.

4) The Importance of Region of Support: The results obtained by our algorithm using three different measures of significance (which correspond to different degrees of accuracy of discrete curvature estimates) are very much the same. This shows that dominant point detection relies heavily on the accurate determination of the local region of support, but not on the accuracy of discrete curvature estimations.

V. Discussions and Conclusion

In two-dimensional shape representation, points with high curvature (i.e., dominant points) are an important attribute of shape. The locations of the detected dominant points must be accurate and the number of points must provide a good representation of the shape without redundancy. Moreover, if dominant points are to be used in automatic pattern classification schemes, they must be generated automatically without relying on the knowledge of any special properties of the input pattern. In addition to all of the above discussed issues, another important one is the problem of adaptability. The problem is not so much with the appropriate selection of the input parameters, but with how well the algorithm adapts to changes in the input pattern, such as scale change, translation, and rotation.

Our dominant point detection algorithm does not have the shortcoming of requiring a set of fixed input parameters. Its ability to vary the region of support for the computation of local curvature enables the algorithm to perform reliably even when the object is scaled or consists of multiple size features. This can be seen from the results in Fig. 6(j)–6(l).

The problem of scale has been a common source of difficulty in a number of signal description applications. In general, the description obtained depends heavily on the scale of measurement. At too fine a scale, one is swamped with extraneous detail. At too coarse a scale, important features may be missed, while those that survive will be distorted by the effects of excessing smoothing. In general, there is seldom a sound basis for choosing the scale of measurement *a priori*. One way to overcome the problem is to describe the signal at more than one scale. However, merely computing descriptions at multiple scales does not solve the problem; some means must be found to organize or simplify the description in a compact way. The scale-space filtering approach introduced by Witkin [16] deals effectively with this problem, and manages the ambiguity of descriptions of the signal at multiple scales.

In our algorithm, each region of a curve with an appropriate length is represented by a dominant point based on the local context of the region. This can be viewed as smoothing the curve by varying length windows. Hence, our algorithm is very much similar to the approach in Witkin's scale-space filtering.

It should be noted that our algorithm involves the computation of the perpendicular distance d_{ik} of a point p_i to a chord form by joining the two points p_{i-k} and p_{i+k}, which is computationally heavy. Instead, the perimeter length of p_{i-k} and p_{i+k}, which can be computed easily, can be used to replace d_{ik}.

Appendix A

Let X and Y be a pair of data sets drawn from a closed curve C, $\{\underline{x} = (x_i, y_i),\ i = 1, \cdots, n\}$, then the definition of their associated properties are given in Table VII.

The matrix A in (A.1c) is given explicitly by

$$A = \frac{1}{\sum_i (x_i - \bar{x})^2 + \sum_i (y_i - \bar{y})^2} \times \begin{bmatrix} \sum_i (x_i - \bar{x})^2 & \sum_i (x_i - \bar{x})(y_i - \bar{y}) \\ \sum_i (x_i - \bar{x})(y_i - \bar{y}) & \sum_i (y_i - \bar{y})^2 \end{bmatrix} \tag{A.2}$$

where $(\bar{x}, \bar{y})$ is the mean of X. Define

$$D_X = \sqrt{1 - 4 \det A}, \tag{A.3a}$$

$$D_Y = \sqrt{1 - 4 \det B}, \tag{A.3b}$$

then we have (see [6] for detail)

(a) X is linear if and only if $D_X = 1$, (A.4a)

(a) X is a circle if and only if $D_X = 0$, (A.4b)

(a) X is elliptical if and only if $0 < D_X < 1$. (A.4c)

Define $\Delta\theta(X, Y)$ as the principal angle between the data sets X and Y, then

$$\Delta\theta(X, Y) = \theta_Y - \theta_X. \tag{A.5}$$

The commutator

$$[A, B] \equiv AB - BA \tag{A.6}$$

is a skew-symmetric matrix and hence, has the explicit form

$$[A, B] = \begin{bmatrix} 0 & \delta(X, Y) \\ \delta(X, Y) & 0 \end{bmatrix}. \tag{A.7}$$

TABLE VII
ASSOCIATED PROPERTIES OF TWO DATA SETS X AND Y

Data: X	=	$(\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_n)$	Y =	$(\underline{y}_1, \underline{y}_2, \cdots, \underline{y}_n)$	(A.1a)
Scatter Matrix: S_X			S_Y		(A.1b)
Normalized Scatter Matrix:					
A	=	$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$	B =	$\begin{bmatrix} b_{11} & b_{21} \\ b_{21} & b_{22} \end{bmatrix}$	(A.1c)
	=	$\frac{S_X}{\text{trace } S_X}$	=	$\frac{S_Y}{\text{trace } S_Y}$	
Principal Angles: θ_X			θ_Y		(A.1d)

The principal angle $\Delta\theta(X, Y)$ is related to the commutator $[A, B]$ by

$$\delta(X, Y) = \tfrac{1}{2} D_X D_Y \sin\left[2\Delta\theta(X, Y)\right] \qquad (A.8)$$

and is also given by

$$\cos\left[2\Delta\theta(X, Y)\right] = \frac{(a_{22} - a_{11})(b_{22} - b_{11}) + 4a_{12}b_{12}}{D_X D_Y}. \qquad (A.9)$$

Equations (A.8) and (A.9) enable one to determine $\Delta\theta$ to within an additive factor of $\pm\pi$.

APPENDIX B

Chain Code of the Chromosome-Shaped Curve in Fig. 3

55454	32011	01111	12112	12006	65655
60010	10765	55455	55555	55431	12122

Chain Code of the Leaf-Shaped Curve in Fig. 4

33332	30700	00003	32307	00003	32322
26777	22212	76661	11116	66566	55000
10056	65655	00110	66565	65555	56667
66666	66664	22222	22222	23224	43433

Chain Code of the figure-8 Curve in Fig. 5

76776	77007	10121	22234	44555	55654
55453	42211	21121			

Chain Code of the Curve in Fig. 6

00007	00777	77766	76666	66665	76766
56454	43436	66656	55454	44434	33232
22254	54434	23221	21322	22222	21221
11111	00100	00			

APPENDIX C

C	a digital closed curve.
n	total number of points on C.
p_i	a point on C.
x_i, y_i	x and y coordinates of p_i.
$\vec{c}_i$	Freeman chain vector $\overrightarrow{p_{i-1}p_i}$.
f	8-direction Freeman code as defined in Fig. 1.
$\{\vec{c}_i, i = 1, \cdots, n\}$	chain of C.
m, s	smoothing parameters.
$\vec{a}_{ik}, \vec{b}_{ik}$	k vectors at p_i used in Rosenfeld–Johnston and Rosenfeld–Weszka procedures.
$\cos_{ik}$	k cosine at p_i used in Rosenfeld–Johnston and Rosenfeld–Weszka procedures.
h_i	region of support of p_i used in Rosenfeld–Johnston and Rosenfeld–Weszka procedures.
$\overline{\cos_{ik}}$	smoothed version of $\cos_{ik}$ used in Rosenfeld–Weszka procedure.
L_{is}	straight line segment spanning s chain links used in Freeman–Davis algorithm.
X_{is}, Y_{is}	x and y components of L_{is} used in Freeman–Davis algorithm.
c_{ix}, c_{iy}	x and y components of $\vec{c}_i$ used in Freeman–Davis algorithm.
θ_{is}	the angle L_{is} makes with the X-axis used in Freeman–Davis algorithm.
δ_{is}	incremental curvature used in Freeman–Davis algorithm.
K_i	cornerity measure of p_i used in Freeman–Davis algorithm.
t_{i1}, t_{i2}	length of regions of relatively constant δ_{is} used in Freeman–Davis algorithm.
Δ	threshold for quantization noise used in Freeman–Davis algorithm.
(i, j)	representation of p_i used in Sankar–Sharma procedure.
$P(i, j)$	curvature value of (i, j) used in Sankar–Sharma procedure.
D	set of detected dominant points in Sankar–Sharma procedure.
$\Delta\theta_t$	tangential deflection angle threshold used in Anderson–Bezdek algorithm.
ϵ_{b1}	baseline acceptability parameter used in Anderson–Bezdek algorithm.
d	minimum number of data points between successive baseline vertices in Anderson–Bezdek algorithm.
v_i	label of the point (vertex) p_i used in Anderson–Bezdek algorithm.
l, c	integer counter used in Anderson–Bezdek algorithm.
$X(l), Y(l)$	sets of points $\{(x_j, y_j) \mid j = l, l + 1, \cdots, l + m - 1\}$ used in Anderson–Bezdek algorithm.
A, B	trace-normalized scatter matrix of $X(l)$ and $Y(l)$ used in

	Anderson–Bezdek algorithm.
$\det A$, $\det B$	determinants of A and B used in Anderson–Bezdek algorithm.
X_{b1}	accepted baseline arc used in Anderson–Bezdek algorithm.
τ_i	tangential deflection used in Anderson–Bezdek algorithm.
$S(p_i)$	measure of significance of p_i.
$D(p_i)$	region of support of p_i.
CUR_{ik}	k curvature measure used in Teh–Chin algorithm.
CUR_{i1}	1 curvature measure used in Teh–Chin algorithm.
$\overline{p_{i-k}p_{i+k}}$	the chord joining the points p_{i-k} and p_{i+k}.
l_{ik}	length of the chord $\overline{p_{i-k}p_{i+k}}$.
d_{ik}	perpendicular distance of the point p_i to the chord $\overline{p_{i-k}p_{i+k}}$.
e_i	perpendicular distance of p_i to a line segment.
E_2	integral square error.
E_∞	maximum error.
n_d	total number of detected dominant points.
t	CPU processing time in seconds.

REFERENCES

[1] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev.*, vol. 61, no. 3, pp. 183–193, 1954.

[2] A. Rosenfeld and E. Johnston, "Angle detection on digital curves," *IEEE Trans. Comput.*, vol. C-22, pp. 875–878, Sept. 1973.

[3] A. Rosenfeld and J. S. Weszka, "An improved method of angle detection on digital curves," *IEEE Trans. Comput.*, vol. C-24, pp. 940–941, Sept. 1975.

[4] H. Freeman and L. S. Davis, "A corner-finding algorithm for chain-coded curves," *IEEE Trans. Comput.*, vol. C-26, pp. 297–303, Mar. 1977.

[5] P. V. Sankar and C. V. Sharma, "A parallel procedure for the detection of dominant points on a digital curve," *Comput. Graphics Image Processing*, vol. 7, pp. 403–412, 1978.

[6] I. M. Anderson and J. C. Bezdek, "Curvature and tangential deflection of discrete arcs: A theory based on the commutator of scatter matrix pairs and its application to vertex detection in planar shape data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 27–40, Jan. 1984.

[7] R. L. T. Cederberg, "An iterative algorithm for angle detection on digital curves," in *Proc. 4th Int. Joint Conf. Pattern Recognition*, Kyoto, Japan, Nov. 7–10, 1978, pp. 576–578.

[8] B. Kruse and C. V. K. Rao, "A matched filtering technique for corner detection," in *Proc. 4th Int. Joint Conf. Pattern Recognition*, Kyoto, Japan, Nov. 7–10, 1978, pp. 642–644.

[9] T. Pavlidis, "Algorithms for shape analysis and waveforms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 301–312, July 1980.

[10] J. G. Dunham, "Optimum uniform piecewise linear approximation of planar curves," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 67–75, Jan. 1986.

[11] L.-D. Wu, "A piecewise linear approximation based on a statistical model," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 41–45, Jan. 1984.

[12] K. Wall and P.-E. Danielsson, "A fast sequential method for polygonal approximation of digitized curves," *Comput. Vision, Graphics, Image Processing*, vol. 28, pp. 220–227, 1984.

[13] J. C. Bezdek and I. M. Anderson, "An application of the c-varieties clustering algorithms to polygonal curve fitting," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, pp. 637–641, Sept. 1985.

[14] H. Imai, "Computational-geometric methods for polygonal approximations of a curve," *Comput. Vision, Graphics, Image Processing*, vol. 36, pp. 31–34, 1986.

[15] T. Pavlidis, *Structural Pattern Recognition*. New York: Springer-Verlag, 1977.

[16] A. P. Witkin, "Scale-space filtering," in *Proc. 8th Int. Joint Conf. Artificial Intell.*, Kalsruhe, West Germany, 1983, pp. 1019–1021.

[17] L. S. Davis, "Understanding shape: Angles and sides," *IEEE Trans. Comput.*, vol. C-26, pp. 236–242, Mar. 1977.

[18] A. Rosenfeld and M. Thurston, "Edge and curve detection for digital scene analysis," *IEEE Trans. Comput.*, vol. C-20, pp. 562–569, May 1971.

[19] A. Rosenfeld, M. Thurston, and Y. H. Lee, "Edge and curve detection: Further experiments," *IEEE Trans. Comput.*, vol. C-21, pp. 677–715, July 1972.

[20] L. S. Davis and A. Rosenfeld, "Detection of step edges in noisy one-dimensional data," *IEEE Trans. Comput.*, vol. C-24, pp. 1006–1010, Oct. 1975.

[21] J. A. Saghri and H. Freeman, "Analysis of the precision of generalized chain codes for the representation of planar curves," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 533–539, Sept. 1981.

[22] J. R. Bennett and J. S. MacDonald, "On the measurement of curvature in a quantized environment," *IEEE Trans. Comput.*, vol. C-24, pp. 803–820, Aug. 1975.

[23] M. J. Eccles, P. C. McQueens, and D. Rosen, "Analysis of the digitized boundaries of planar objects," *Pattern Recognition*, vol. 9, pp. 31–41, 1977.

[24] F. C. A. Groan and P. W. Verbeek, "Freeman-code probabilities of object boundary quantized contours," *Comput. Vision, Graphics, Image Processing*, vol. 7, pp. 391–402, 1978.

[25] B. Rosenberg, "The analysis of convex blobs," *Comput. Graphics Image Processing*, vol. 1, pp. 183–192, 1972.

[26] D. Langridge, "On the computation of shape," in *Frontiers of Pattern Recognition*, S. Watanabe, Ed. New York: Academic, 1972, pp. 347–365.

[27] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. New York: Academic, 1982, 2nd ed., p. 261.

Cho-Huak Teh (S'78–M'88) received the B.Eng. degree with honors in 1980 and the M.Eng. degree in 1982, both in electrical engineering from the National University of Singapore, and the Ph.D. degree in electrical and computer engineering in 1988 from the University of Wisconsin, Madison. From 1980 to 1982 he was a Senior Tutor at the National University of Singapore.

He has been on the faculty of the Department of Electrical Engineering at the National University of Singapore since March 1988. His research interests include image processing, signal processing, pattern recognition, computer vision, and related applications.

Dr. Teh is a member of the Pattern Recognition Society and the British Pattern Recognition Association.

Roland T. Chin (S'75–M'79) received the B.S. degree with honors in 1975 and the Ph.D. degree in 1979, all in electrical engineering from the University of Missouri, Columbia.

From 1979 to 1981, he was with Business and Technological Systems, Inc., MD, where he engaged in research in remote sensing data analysis and classification for NASA Goddard Space Flight Center, Greenbelt, MD. Since 1981, he has been on the faculty of the Department of Electrical and Computer Engineering at the University of Wisconsin, Madison, where he is currently Professor and Associate Chair. His current research interests include image restoration, texture analysis, shape descriptions, pattern recognition, visual inspection, object recognition, and related applications.

Dr. Chin is a member of Eta Kappa Nu and Tau Beta Pi, and he is the recipient of the First Presidential Young Investigator Award in 1984.