

# Detection of significant points and polygonal approximation of digitized curves

Bimal Kr. Ray and Kumar S. Ray

*Electronics & Communication Sciences Unit, Indian Statistical Institute, 203 Barrackpore Trunk Road, Calcutta 700 035, India*

## *Abstract*

Ray, B.Kr. and K.S. Ray, Detection of significant points and polygonal approximation of digitized curves, Pattern Recognition Letters 13 (1992) 443-452.

A new method, based on  $k$ -cosine, is proposed for the determination of region of support and detection of significant points of digitized curves. A new measure of significance, based on  $k$ -cosine, is introduced. The procedure is parallel and requires no input parameter. A polygonal approximation is suggested by joining the successive significant points.

*Keywords.* Significant points, region of support, digitized curve, shape analysis, curvature maxima point, curvature minima point.

## 1. Introduction

Significant points of digitized curves can be used as a compact and effective representation of the curve for shape analysis and pattern recognition [20].

Significant points are of two types, namely, (i) curvature maxima points and (ii) curvature minima points. The technique of detecting local curvature maxima originates from Attneave's famous observation [1] that informations about a curve are concentrated at the curvature maxima points. Freeman [2] suggested that the points of inflexion carry information about a curve and hence they can be used as significant points. The curvature minima points are the points of inflexion.

*Correspondence to:* K.S. Ray, Electronics & Communication Sciences Unit, Indian Statistical Institute, 203 Barrackpore Trunk Road, Calcutta 700 035, India.

Many algorithms have been built up for detecting significant points of digitized curves. The oldest of these algorithms was given by Rosenfeld and Johnston [3]. They, in an attempt to determine whether a procedure [4] designed to detect discontinuities in the average grey level would also detect discontinuities in the average slope, detected significant points as curvature extrema points of digitized curves. The procedure is parallel and needs an input parameter  $m$ . The value of  $m$  was taken to be  $1/10$  or  $1/15$  of the perimeter of the curve. The input parameter was introduced to determine the region of support and the  $k$ -cosine of the boundary points.

An improved version of this procedure was given by Rosenfeld and Weszka [5]. They used smoothed  $k$ -cosine to determine the region of support and to detect the significant points. The procedure is parallel and needs a single parameter  $m$  as in [3].

Freeman and Davis [6] designed a corner-finding scheme which detects local curvature maxima points as significant points. The algorithm consists of scanning the chain code of the curve with a moving 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. The procedure is parallel and needs two parameters  $s$  and  $m$  [10]. Both are smoothing parameters and their assigned values will determine the degree of smoothing. The greater the  $s$ , the heavier is the smoothing. The parameter  $m$  is used to allow some stray noise. For a well quantized chain  $s$  will always be a relatively small number ranging from 5 to 13. And the parameter  $m$  will take value either 1 or 2.

Anderson and Bezdek [7] devised a vertex detection algorithm which, instead of approximating discrete curvature, defined tangential deflection and curvature of discrete curves on the basis of the geometrical and statistical properties associated with the eigenvalue-eigenvector structure of sample covariance matrices. The vertices are the significant points in the sense that they carry information about the curve. The procedure is sequential and needs more than one parameter.

Sankar and Sharma [8] designed an iterative procedure for detecting significant points as points of maximum global curvature based on the local curvature of each point with respect to its immediate neighbors. The procedure is parallel. In contrast to the previous algorithms, it does not need any input parameter.

Each of the algorithms [3], [5], [6] and [7] needs one or more input parameters. The choice of these parameters is primarily based on the level of detail of the curves. In general, it is difficult to choose a set of parameters that can successfully be used for detecting significant points of a curve consisting of features of multiple sizes. Too large a parameter will smooth out fine features, and a small parameter will generate a large number of unwanted significant 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 a

particular value of parameter corresponding to a particular feature size [9].

Though Sankar-Sharma's algorithm [8] does not need any input parameter, it does not involve determination of region of support. The procedure is iterative in nature and fails to operate successfully on curves consisting of features of multiple sizes.

Recently, Teh and Chin [10] have designed a procedure for detecting significant points which requires no input parameter and remains reliable even when features of multiple sizes are present. In Section 2 we give an overview of this algorithm.

In Section 3 we have introduced a new method for the determination of region of support and detection of significant points. We have also introduced a new measure of significance based on  $k$ -cosine and subsequently in Section 5 we have used all these ideas for polygonal approximation.

## 2. Teh and Chin algorithm

### 2.0. A new observation

In contrast to the existing belief that the detection of significant points depends heavily on the accurate measures of significance (e.g.,  $k$ -curvature measures,  $k$ -cosine measure, cornerity measure, weighted curvature measure), Teh and Chin [10] made an important observation: the detection of dominant points relies not only on the accuracy of the measures of significance, but primarily on the precise determination of the region of support. Their procedure is motivated by the Rosenfeld-Johnston angle detection algorithm [3], in which both an incorrect region of support and incorrect curvature measure may be assigned to a point if the input smoothing parameter is not chosen correctly, and hence significant points may be suppressed [11]. To overcome this problem they proposed 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. They have further shown that once the region of support of each point is determined, various measures of significance can be computed accurately for the

detection of significant points. They have called the significant points as the dominant points.

2.1. Determination of region of support

Let the sequence of  $n$  integer coordinate points describe a closed curve

$$c_d = \{p_i = (x_i, y_i) \mid i = 1, 2, \dots, n\} \quad (2.1.1)$$

where  $p_{i+1}$  is a neighbour of  $p_i$  (modulo  $n$ ). The Freeman chain code of  $c_d$  consists of  $n$  vectors

$$c_i = \overline{p_i p_{i+1}} \quad (2.1.2)$$

each of which can be represented by an integer

$$f = 0, 1, \dots, 7 \quad (2.1.3)$$

as shown in Figure 1, where  $\frac{1}{4}\pi f$  is the angle between the  $x$ -axis and the vector. The chain of  $c_d$  is defined by  $\{c_i \mid i = 1, 2, \dots, n\}$  and  $c_i = c_{i \pm n}$ . All integers are modulo  $n$ .

For each point  $p_i$  of  $c_d$ , the region of support ( $k_i$ ) is determined by the following procedure.

Procedure: Determination of region of support

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

$$l_{ik} = |\overline{p_{i-k} p_{i+k}}|. \quad (2.1.4)$$

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

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

$$(a) \quad l_{ik} \geq l_{i,k+1} \quad (2.1.5)$$

or

$$(b) \quad \frac{d_{ik}}{l_{ik}} \geq \frac{d_{i,k+1}}{l_{i,k+1}} \quad \text{for } d_{ik} > 0, \quad (2.1.6)$$

$$\frac{d_{ik}}{l_{ik}} \leq \frac{d_{i,k+1}}{l_{i,k+1}} \quad \text{for } d_{ik} < 0. \quad (2.1.7)$$

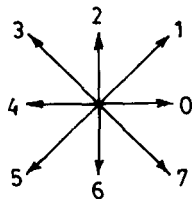


Figure 1. Directions in Freeman chain.

Then the region of support ( $D(p_i)$ ) of  $p_i$  is the set of points which satisfy either condition (a) or condition (b), i.e.

$$D(p_i) = \{p_{i-k}, \dots, p_{i-1}, p_i, p_{i+1}, \dots, p_{i+k} \mid \text{condition (a) or condition (b)}\}. \quad (2.1.8)$$

2.2. Detection of dominant points

To detect the dominant points any one of the three measures of significance ( $S(p_i)$ ) [12], namely,  $k$ -cosine measure,  $k$ -curvature measure and 1-curvature measure, can be used. The measure of significance of each point is determined by using the neighboring points within the extent. The measure of significance and region of support are used to guide the selection of points to be removed. The remaining points (after the removing process) are the dominant points. The procedure leading to the set of dominant points is described below.

Procedure: Detection of dominant points

(1) Determine the region of support of each point by the last procedure described in Section 2.1,

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

(2) Select a measure of significance and calculate its absolute value for each point,  $|S(p_i)|$ .

(3a) 1st Pass. Perform non-maxima suppression as follows: retain only those points  $p_i$  where

$$|S(p_i)| \geq |S(p_j)| \quad (2.1.9)$$

for all  $j$  such that

$$|i - j| \leq k_i/2.$$

(3b) 2nd Pass. Further suppress those points having zero 1-curvature.

(3c) 3rd Pass. For those points surviving after the 2nd Pass, if ( $[k_i \text{ of } D(p_i)] = 1$ ) and ( $p_{i-1}$  or  $p_{i+1}$  still survives) then further suppress  $p_i$  if

$$(|S(p_i)| \leq |S(p_{i-1})|) \text{ or } (|S(p_i)| \leq |S(p_{i+1})|).$$

If 1-curvature is selected as a measure of significance, then go to Step (3d) and do a 4th Pass else those points which survive are the dominant points.

(3d) 4th Pass. For those groups of more than 2

points that still survive, suppress all the points except the two end points of each of the groups.

For those points of exactly 2 points that still survive,

if  $(|S(p_i)| > |S(p_{i+1})|)$  then suppress  $p_{i+1}$

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  $p_i$ .

End of procedure.

### 3. Present algorithm

#### 3.0. A new concept

As stated by Teh and Chin [10], there are two major problems with the detection of significant points of digitized 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 [10] the region of support was determined by using the chord lengths and perpendicular distance. In the present communication we wish to show that we can use  $k$ -cosine itself to determine the region of support for each point. The procedure, as described below, is parallel and does not require any input parameter.

#### 3.1. Procedure: Determination of region of support

(1) Define the  $k$ -vectors at  $p_i$  as

$$a_{ik} = (x_i - x_{i+k}, y_i - y_{i+k}), \quad (3.1.1)$$

$$b_{ik} = (x_i - x_{i-k}, y_i - y_{i-k}) \quad (3.1.2)$$

and the  $k$ -cosine at  $p_i$  as

$$\text{Cos}_{ik} = \frac{a_{ik} \cdot b_{ik}}{|a_{ik}| |b_{ik}|}. \quad (3.1.3)$$

Here  $\text{Cos}_{ik}$  is the angle between the  $k$ -vectors  $a_{ik}$  and  $b_{ik}$ , so that  $-1 \leq \text{Cos}_{ik} \leq 1$ .

(2) Start with  $k = 1$ . Compute  $\text{Cos}_{ik}$  giving increment to  $k$ .

If  $|\text{Cos}_{i,k+1}| > |\text{Cos}_{ik}|$  then  $k$  determines the region of support  $p_i$

else if  $\text{Cos}_{ik} = \text{Cos}_{i,k+1}$  then the greatest  $k$  for

which this relation holds determines the region of support of  $p_i$

else if  $\text{Cos}_{ik}$  and  $\text{Cos}_{i,k+1}$  are of opposite sign then the least value of  $k$  for which it happens gives the region of support of  $p_i$ .

The region of support of  $p_i$ , which is also called the domain [12] of  $p_i$ , is the set of points given by

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

#### 3.2. Measure of significance

The last procedure determines the region of support ( $k_i$ ) for each point  $p_i$ . To detect the significant points we need a measure of significance. Rosenfeld and Johnston [3] used  $\text{Cos}_{ih_i}$  as the measure of significance and  $h_i$  as the region of support of  $p_i$ . Rosenfeld and Weszka [5] used smooth  $k$ -cosine as a measure of significance. In the present communication we propose to introduce a new measure of significance. We denote it by  $\text{Cos}_i$  and define it by

$$\text{Cos}_i = \frac{1}{k_i} \sum_{j=1}^{k_i} \text{Cos}_{ij}. \quad (3.2.1)$$

This measure is a kind of smoothed cosine but it is different from that given by Rosenfeld and Weszka [5].

With the region of support ( $k_i$ ) determined by the procedure given in Section 3.1 and the measure of significance introduced in Section 3.2, we now proceed to detect the significant points.

#### 3.3. Procedure: Detection of significant points

*Comments.* As the procedure runs we remove those points from consideration where  $\text{Cos}_i$ 's are too low ( $\text{Cos}_i \leq -0.800$ ), because in the neighborhood of these points the curves are relatively straight and our ultimate goal is to make a polygonal approximation of the curves.

*1st Pass.* Retain only those points  $p_i$  for which either

$$(a) \text{Cos}_i \geq \text{Cos}_j \quad (3.3.1)$$

for all  $j$  satisfying

$$\begin{aligned} |i-j| \leq k_i/2, \quad k_i > 1, \\ \leq k_i, \quad k_i = 1 \end{aligned} \quad (3.3.2)$$

or

$$(b) \quad \text{Cos}_i \leq \text{Cos}_j \quad (3.3.3)$$

for all  $j$  satisfying

$$\begin{aligned} |i-j| \leq k_i/2, \quad k_i > 1, \\ \leq k_i, \quad k_i = 1. \end{aligned} \quad (3.3.4)$$

In (3.3.1) and (3.3.3) the inequality should hold for at least one  $j$  satisfying (3.3.2) and (3.3.4).

*Comments.* The points detected by (a) are the curvature maxima points and those detected by (b) are the curvature minima points.

*2nd Pass.* If a curvature minima point falls within the region of support of a curvature maxima point, then the curvature minima point is discarded and the maxima point is retained.

If we have two successive points  $p_i$  and  $p_{i+1}$  appearing as curvature maxima points then

if both  $p_i$  and  $p_{i+1}$  have the same cosine and same region support, then retain  $p_i$  and discard  $p_{i+1}$

else if the cosines are the same and the regions of support are different, retain only that point with higher region of support and discard the other.

The points obtained from the 1st and 2nd Pass constitute the set of significant points of the digitized curve.

End of procedure.

**Remark 1.** When two successive points  $p_i$  and  $p_{i+1}$  appear as curvature maxima points and both have the same cosine, then both the points are equally important for being selected as a significant point and thus there is a tie. We have proposed to break the tie by choosing  $p_i$ . On the other hand, when  $p_i$  and  $p_{i+1}$  have the same cosine but different regions of support, then there is no tie and the choice is entirely deterministic.

**Remark 2.** The 1st Pass of the procedure can be carried out in parallel, whereas the 2nd Pass is sequential. We note that the 2nd Pass is carried out only on a small number of points.

#### 4. Computational complexity

The success of a procedure is determined by its computational feasibility. In this section we propose to discuss the computational complexity of the procedure described in Section 3.1.

In this procedure, we determine  $\text{Cos}_{ik}$  by the definition (3.1.1), (3.1.2) and (3.1.3). The computations of the vectors  $\mathbf{a}_{ik}$  and  $\mathbf{b}_{ik}$  need 2 subtractions each.

Let us write  $a_{ik}^{(1)}$ ,  $a_{ik}^{(2)}$  for the  $x$ - and  $y$ -component respectively of the vector  $\mathbf{a}_{ik}$  and  $b_{ik}^{(1)}$ ,  $b_{ik}^{(2)}$  for those of  $\mathbf{b}_{ik}$ . Then

$$\begin{aligned} \text{Cos}_{ik} &= \frac{a_{ik}^{(1)}b_{ik}^{(1)} + a_{ik}^{(2)}b_{ik}^{(2)}}{\sqrt{(a_{ik}^{(1)})^2 + (a_{ik}^{(2)})^2} \sqrt{(b_{ik}^{(1)})^2 + (b_{ik}^{(2)})^2}} \\ &= \frac{a_{ik}^{(1)}b_{ik}^{(1)} + a_{ik}^{(2)}b_{ik}^{(2)}}{\sqrt{\{(a_{ik}^{(1)})^2 + (a_{ik}^{(2)})^2\} \{(b_{ik}^{(1)})^2 + (b_{ik}^{(2)})^2\}}} \end{aligned}$$

The computation of the numerator involves 2 multiplications and 1 addition. The computation  $(a_{ik}^{(1)})^2 + (a_{ik}^{(2)})^2$  involves 2 multiplications and 1 addition, that of  $(b_{ik}^{(1)})^2 + (b_{ik}^{(2)})^2$  involves 2 multiplications and 1 addition. So the computation of the denominator needs 5 multiplications, 2 additions and 1 square root. So for each value of  $k$  the computation of  $\text{Cos}_{ik}$  needs 7 additions, 8 multiplications/divisions and 1 square root, i.e., a total of 16 arithmetic operations. The procedure determines the region of support ( $k_i$ ) for each point  $p_i$ . Let us write

$$k_{\max} = \max_i (k_i).$$

Then the total number of arithmetic operations required to carry out the procedure is  $16n k_{\max}$ . Since  $k_i$  does not depend on data size, hence  $k_{\max}$  too does not. Therefore the computational complexity of the procedure is  $O(n)$ .

The procedure described in Section 3.3 involves only comparisons. The time complexity of the procedure is also  $O(n)$ .

#### 5. Polygonal approximation

A digitized curve can be approximated with arbitrary accuracy by a polygon. For a closed curve

the approximation is exact when the number of segments in the polygon is equal to the number of points in the boundary so that each pair of adjacent points defines a segment in the polygon. In practice the goal of polygonal approximation is to capture the 'essence' of the boundary shape with the fewest possible segments. Although the problem is in general not trivial and can very quickly turn into a time-consuming iterative search, there are a number of polygonal approximation techniques with modest complexity and processing requirements [13].

The earliest of the polygonal approximation techniques involving the fewest possible line segments were given by Ramer [14] and Duda and Hart [15]. Their procedures split the curve into smaller and smaller curve segments until the maximum perpendicular distance of the points of the curve segment from the line segment joining the initial and last point of the curve segment falls below a specified threshold. The curve segments are split at the point most distant from the line segment.

Some of the other works on polygonal approximation were done by Williams [16], Pavlidis and Horowitz [17], Pavlidis [18] and Wall and Danielsson [19]. Williams [16] used a cone intersection method to find the maximal possible line segment. Circles are drawn around each point. Points are merged one after another with the initial point until the intersection of the cones with vertex at the initial point and touching the circles is an empty set. The line segments are obtained by joining the initial point to the last point which passes the test.

Pavlidis and Horowitz [17] used a split and merge technique which fits lines to an initial segmentation of the boundary and computes the least squares error. The procedure then iteratively splits the line if the error is too large and merges two lines if the error is too small.

Pavlidis [18] developed an algorithm which is based upon the concept of a sequence of points being almost collinear. To check whether a sequence of points is collinear/almost collinear the procedure computes an error of fit which is a function of two variables,  $T$  and  $C$ .  $T$  is the maximum of the perpendicular distances of the points (being tested

for collinearity) from the yet-to-be-obtained line segment, and  $C$  is a normalized variable ( $0 \leq C \leq 1$ ) which is determined by the ratio of the number of sign changes the perpendicular distance goes through to the total number of possible sign changes. If  $T - Cw_0 - T_0 < 0$  the line segment is accepted else if  $T - Cw_0 - T_0 > 0$  the line segment is rejected, where  $w_0$  is the weighting factor of  $C$  and  $T_0$  is the acceptable error.

Wall and Danielsson [19] designed a merging technique which is based on the concept of area deviation. The procedure finds the maximal line segment by merging points one after another to the initial point until the area deviation per unit length of the current line segment exceeds a maximum allowed value. The line segments are obtained by joining the initial point to the last point that passed the test.

Polygonal approximation of digitized curves is often a compact and effective representation of the curve for shape analysis and pattern classification [21]. Such representations facilitate the extraction of numerical features for the description of the curve.

In the present communication we use the ordered set of significant points to make a polygonal approximation of digitized curves. The polygon is obtained by joining the successive significant points.

## 6. Approximation errors

The shape of a digitized curve is determined by its significant points. The set of significant points can be regarded as a set of parameters characterizing the digitized curve. So it is very much necessary to locate the significant points accurately so that necessary and sufficient informations about the curve are contained in the location of significant points. In Sections 3.1 and 3.3 we have described a procedure for detecting significant points of digitized curves.

The accuracy of the location of the significant points and the closeness of the polygon to the digitized curve can be determined by the pointwise error between the digital curve and the approximating polygon. We measure the error between

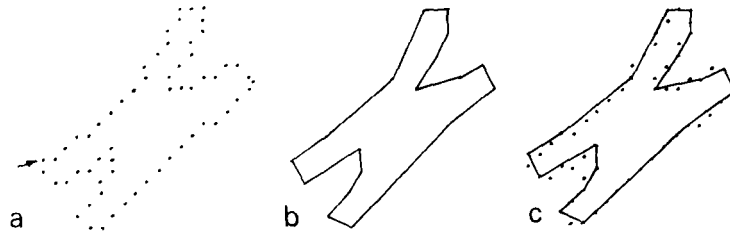


Figure 2. (a) A chromosome shaped curve. (b) Polygonal approximation of (a). (c) Overlap of (b) on (a).

the digitized curve and the approximating polygon by the perpendicular distance of the points  $p_i$  from their approximating line segment. We denote this error by  $e_i$ . Two error norms between the closed digital curve and the approximating polygon can be defined:

(1) *Integral square error*

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

(2) *Maximum error*

$$E_\infty = \max_{1 \leq i \leq n} e_i$$

## 7. Experimental results

We have applied our procedure on four digital curves, namely, a chromosome shaped curve (Figure 2(a)), a figure-8 curve (Figure 3(a)), a leaf-shaped curve (Figure 4(a)) and a curve with four semicircles (Figure 5(a)). The first three have been taken from [3] and the last, which is a special figure exhibiting an image consisting of features of multiple size, has been taken from [10]. The chain codes (input data) of each curve are given in Tables 1-4. The curves have been coded in clockwise direction starting from the point marked with  $\nearrow$  on each curve, using the Freeman chain code defined

in Figure 1. The procedure processes data in clockwise direction.

In an attempt to focus on the efficiency of our procedure as a significant point detector and a

Table 1

Chain code of chromosome shaped curve

01101	11112	11212	00665	65560	01010	76555	45555
55555	43112	12255	45432				

Table 2

Chain code of figure-8 curve

11217	67767	70071	01212	22344	45555	56545	54534
22112							

Table 3

Chain code of leaf shaped curve

33333	32307	00003	32323	07000	03323	22267	77222
12766	61111	16665	66550	00100	56656	55001	10665
65655	55566	67666	66666	64222	22222	22232	24434

Table 4

Chain code of curve in Figure 5 (a)

22222	21221	11111	00100	00000	07007	77777	66766
66666	65767	66564	54434	36666	56554	54444	34332
32222	54544	34232	21213	22			



Figure 3. (a) A figure-8 curve. (b) Polygonal approximation of (a). (c) Overlap of (b) on (a).

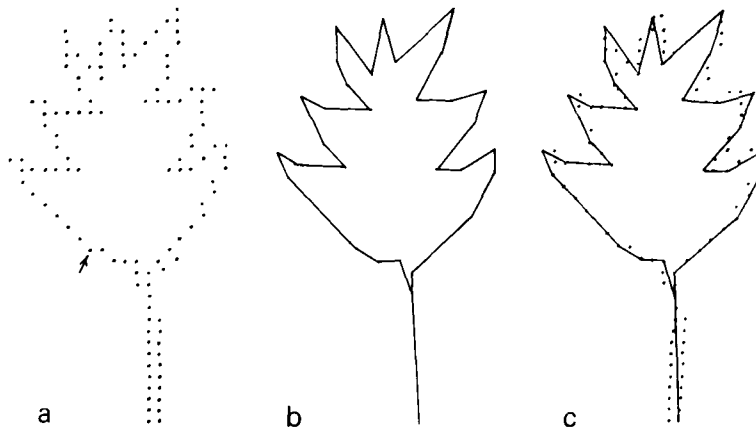


Figure 4. (a) A leaf shaped curve. (b) Polygonal approximation of (a). (c) Overlap of (b) on (a).

polygonal approximation technique we have computed (a) the data compression ratio defined by  $n/n_s$ ,  $n_s$  being the number of significant points, (b) the integral square error and (c) the maximum error. The results are given in Table 5. Table 6 shows the results obtained by the algorithm described in [10].

**8. Discussion**

We have made no attempt of comparing our procedure with the existing ones [3], [5], [6] and [7]. A comparison of these algorithms with that

given in [10] can be found in [10]. But incidently we find the following features of our procedure.

(i) Our procedure, like Sankar-Sharma [8] and Teh-Chin [10], does not require any input parameter.

(ii) Though [8] does not require any input parameter, it does however not determine the region of support. But our procedure, like the procedure of [10], does determine the region of support.

(iii) Our procedure detects more significant points (except for the curve in Figure 3(a)) than any other algorithm and consequently, the approximation errors are lowered.

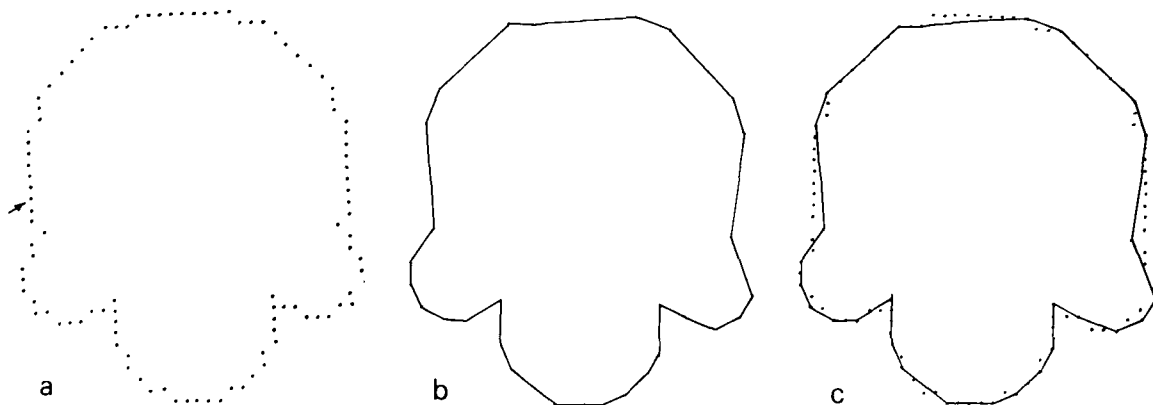


Figure 5. (a) A curve consisting of four semi-circles. (b) Polygonal approximation of (a). (c) Overlap of (b) on (a).



Table 5  
Results of present algorithm

Digital curve of	Number of data points	Number of significant points ( $n_s$ )	Compression ratio ( $n/n_s$ )	Integral square error	Maximum error
Figure 2(a)	60	18	3.33	5.566	0.707
Figure 3(a)	45	12	3.75	5.989	0.894
Figure 4(a)	120	32	3.75	14.718	0.996
Figure 5(a)	102	29	3.50	11.818	0.833

## 9. Conclusion

We have designed a procedure which shows that  $k$ -cosine itself can determine the region of support without using any input parameter. The significant points can be detected with the help of the smoothed  $k$ -cosine defined in (3.2.1). The procedure detects not only the curvature maxima points but also the curvature minima points. The polygon obtained by joining the successive significant points gives a good approximation to the digitized curve. The procedure 'Determination of region of support' implicitly fits a circular arc in the neighborhood of each point. The new concept has been applied on a number of curves and promising results have been obtained.

## References

- [1] Attneave, F. (1954). Some informational aspects of visual perception. *Psychological Review* 61 (3), 183-193.
- [2] Freeman, H. (1967). On the classification of line drawing data. In: W. Wather Dunn, Ed., *Models for the Perception of Speech and Visual Form*. MIT Press, Cambridge, MA, 408-412.
- [3] Rosenfeld, A. and E. Johnston (1973). Angle detection on digital curves. *IEEE Trans. Comput.* 22, 875-878.
- [4] Rosenfeld, A. and M. Thurston (1971). Edge and curve detection for digital scene analysis. *IEEE Trans. Comput.* 20, 562-569.
- [5] Rosenfeld, A. and J.S. Weszka (1975). An improved method of angle detection on digital curves, *IEEE Trans. Comput.* 24, 940-941.
- [6] Freeman, H. and L.S. Davis (1977). A corner-finding algorithm for chain-coded curves. *IEEE Trans. Comput.* 26, 287-303.
- [7] Anderson, I.M. and J.C. Bezdek (1984). Curvature and tangential deflection of discrete arcs: A theory based on the commutator of scatter matrix pair and its application to vertex detection in plane shape data, *IEEE Trans. Pattern Anal. Machine Intell.* 6, 27-40.
- [8] Sankar, P.V. and C.V. Sharma (1978). A parallel procedure for the detection of dominant points on a digital curve, *Computer Graphics and Image Processing* 7, 403-412.
- [9] Witkin, A.P. (1983) Scale-space filtering. *Proc. 8th Int. Joint Conf. on Artificial Intelligence*, Karlsruhe, Germany, 1019-1021.
- [10] Teh, C.H. and R.T. Chin (1989). On the detection of dominant points on digital curves. *IEEE Trans. Pattern Anal. Machine Intell.* 11 (8).
- [11] Davis, L.S. (1977). Understanding shape: angles and sides. *IEEE Trans. Comput.* 26, 236-242.
- [12] Rosenberg, B. (1972). The analysis of convex blobs. *Computer Graphics and Image Processing* 1, 183-192.

Table 6  
Results of Teh and Chin algorithm [taken from (10)]

Digital curve of	Number of data points	Number of dominant points ( $n_d$ )	Compression ratio ( $n/n_d$ )	Integral square error	Maximum error
Figure 2(a)	60	15	4.0	7.20	0.74
Figure 3(a)	45	13	3.5	5.93	1.00
Figure 4(a)	120	29	4.1	14.96	0.99
Figure 5(a)	102	22	4.6	20.61	1.00

- [13] Gonzalez, R.C. and P. Wintz (1987). *Digital Image Processing*. Addison-Wesley, Reading, MA.
- [14] Ramer, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing* 1, 244-256.
- [15] Duda, R.O. and P.E. Hart (1973). *Pattern Classification and Scene Analysis*. Wiley, New York, 328-329.
- [16] Williams, C.M. (1978). An efficient algorithm for the piecewise linear approximation of planar curves. *Computer Graphics and Image Processing* 8, 286-293.
- [17] Pavlidis, T. and S.L. Horowitz (1974). *IEEE Trans. Comput.* 23, 860-870.
- [18] Pavlidis, T. (1982). *Algorithms for Graphics and Image Processing*. Computer Science Press, New York, 281-297.
- [19] Wall, K. and P.E. Danielsson (1984). A fast sequential method for polygonal approximation of digitized curves. *Computer Vision, Graphics, and Image Processing* 28, 220-227.
- [20] Pavlidis, T. (1977). *Structural Pattern Recognition*. Springer, New York.
- [21] Pavlidis, T. (1980). Algorithms for shape analysis and waveforms. *IEEE Trans. Pattern Anal. Machine Intell.* 2, 301-312.