Corner Detection and Curve Partitioning Using Arc-Chord Distance

Majed Marji[1], Reinhard Klette[2] and Pepe Siy[1]

[1] Electrical and Computer Engineering, Wayne State University
3100 W. Engineering Bldg., Detroit, MI 48202, USA
[2] The University of Auckland, CITR, Tamaki campus, Bldg. 731
Morrin Road, Glen Innes, Auckland 1005, New Zealand

Dedicated to Professor Azriel Rosenfeld

Abstract. Several authors have proposed algorithms for curve partitioning using the arc-chord distance formulation, where a chord whose associated arc spans k pixels is moved along the curve and the distance from each border pixel to the chord is computed. The scale of the corners detected by these algorithms depends on the choice of integer k. Without a priori knowledge about the curve, it is difficult to choose a k that yields good results. This paper presents a modified method of this type that can tolerate the effects of an improper choice of k to an acceptable degree. The new algorithm seems to yield generally good results.

Keyword: curve partitioning, corner detection, polygonal approximation

1 Introduction

A critical problem in machine vision is how to breakup (partition) the perceived world into coherent or meaningful parts prior to knowing the identity of these parts [9]. In particular, the problem of partitioning digital planar curves has been a subject of intense investigation since the earliest days of machine vision. Some of the more immediate applications include data compression (by using the partitioning points as the basis for regenerating a curve by straight line or spline interpolation), and matching or recognition (by using the partitioning points and/or the partitioned curve segments).

Several authors have studied methods of curve partitioning using the arc-chord distance formulation. A polygonal approximation of the digital curve is then generated by connecting the attained partitioning points - also referred to as corners in the literature- with straight lines.

In the arc-chord distance method, a *chord* whose associated arc spans k pixels (i.e., a k-*point arc* of the curve) is moved along the curve and the distance from each pixel on the k-point arc to the chord is computed. A *significance* (also called *cornerity*) *measure* is formulated using these distances (e.g., maximum, distance-accumulation etc.) and processed in order to define corners of the curve.

Ramer's algorithm [1] recursively partitions an arc at the point whose distance from the chord is a maximum.

Rutkowski [2] computes the maximum distance of each point p on the curve from any chord having a given arc length and having p on its arc, and partitions the curve at local maxima of this distance.

The algorithm of Fischler and Bolls [3] labels each point on a curve as belonging to one of three categories: 1) a point in a smooth interval, 2) a critical point, or 3) a point in a noisy interval. To make this choice, the algorithm analyzes the deviations of the curve from a chord or "stick" that is iteratively advanced along the curve. If the curve stays close to the chord, points in the interval spanned by the chord will be labeled as belonging to a smooth section. If the curve makes a single excursion from the chord, the point in the interval that is farthest from the chord will be labeled a critical point (actually, for each placement of the chord, an accumulator associated with the farthest point will be incremented by the distance between the point and the chord). If the curve makes two or more excursions, points in the interval will be labeled as noise points.

Phillips and Rosenfeld [4] presented a modified version of the algorithm presented in [2]. They also suggested an approach to choosing good values of k in a given part of the curve. To find a good value of k, they determined the best fit straight line for each k-point arc of the curve, and computed the RMS error corresponding to this fit. This process is repeated for a sequence of arc lengths, producing a sequence of fit measures for each border point. In a given part of the border "good" values of k are taken as those which produce local minima in the fit measure.

Han [5] proposed a method similar to that of Fischler and Bolls [3] but used the signed distance to the chord. The algorithm keeps two separate accumulators for the positive and negative arc-chord distances to distinguish between concave and convex corners. For a given chord-length L, a line is drawn from point p_i to point p_{i+L} on the curve. The signed distances from all points $p_1, \ldots, p_{L-1}$ to this line are calculated. The point with positive maximum distance is defined as p_+, and the point with the negative minimum distance is defined as p_- . If the absolute value of the maximum (minimum) distance exceeds a given threshold $D_{\min}$, the counter $(h(p_+)/h(p_-))$, associated with the point that corresponds to the maximum (minimum) is incremented (decremented). The line is advanced by one pixel and the process is repeated until the entire curve is scanned. In other words, this algorithm counts how many times each border point happened to be the farthest point from the line $p_i p_{i+L}$. At the end of the calculation, the points whose accumulator value exceeds a given threshold $H_{\min}$ are marked as concave (convex) points.

Lin, Dou and Wang [6] proposed a new shape description method termed the *arc height function* and used it to detect the corners of the border. A chord that joins border points p_i and p_{i+k} is advanced along the curve, one pixel at a time. A straight-line perpendicular to the chord passing through its center p_c intersects the border at point p_x. The distance between p_c and p_x is the arc height, which, when computed for all positions of the chord gives the arc height function. The corner points of the border correspond to the local maxima of the arc height function.

The algorithm of Aoyama and Kawagoe [7] starts by finding all occurrences of digital straight-line patterns and marking their endpoints as candidate vertices. The best approximating straight-lines are determined by considering the ratio between the height H and the length L of the chord, which is termed the *pseudo curvature* G. Two modifications were made to the calculation of H and G. First, the pseudo curvature calculation was modified in such a way to prevent a long straight-line from being approximated as an inclined line. Second, the distance calculation was modified to take into account cases where the perpendicular line does not intersect the line segment.

Wu and Wang [8] combined corner detection and polygonal approximation. For a given parameter k, a significance measure is assigned to each border point p_i as the ratio $\frac{d_i}{L_i}$, where d_i is the distance between point p_i and the chord $(p_{i-k}, p_{i+k}))$, and L_i is the length of the chord. Local maxima points whose significance is greater than a threshold were taken as potential corners and used as the starting points for polygonal approximation. The border points within each segment (between two corners) are sorted according to their significance (most significant first). The sorted points are tested sequentially by calculating their distance to the chord that joins the end points of the segment; if this distance exceeds a given threshold, the corresponding point is marked as a corner.

The work of Fischler and Wolf [9] extends the technique of [3]. An important contribution of their work over [3] is a major revision of the approach to filtering the critical points, based on comparisons at a given scale as well as across different scales (i.e., different values of the input parameter k). In addition, the sign of the computed saliency measure is taken into consideration.

Han and Poston [10] proposed an enhanced version of the algorithm presented in [5]. Here, instead of incrementing a counter when the distance exceeds a threshold as in [5], the actual signed Euclidean distance is accumulated.

In this work, we propose a new algorithm based on the work of Phillips and Rosenfeld [4] that can tolerate the effect of an improper choice of k to an acceptable degree. In Section 2 we will review the algorithm of [4] and highlight the major observations that led to the development of the new algorithm. The new algorithm will be presented in Section 3 and the experimental results will be given in Section 4.

2 The Method of Phillips and Rosenfeld

The algorithm is illustrated with the aid of Figure 1. Let p be a point on the curve and let k be the chosen arc length. For each chord C whose arc has length k and has p in its interior, let $d(p, C)$ be the perpendicular distance from p to C. Let $M(p, C)$ be the maximum of these distances for all such chords. Point p is called a *partition point* if the value of $M(p, C)$ is a local maximum (for the given k) and also exceeds a threshold $t = k/5 \cong (k/2)\cos(135°/2)$, which is the altitude of an isosceles triangle whose vertex angle is $135°$ and whose equal sides have lengths $k/2$. Point p is considered a local maximum point if the following

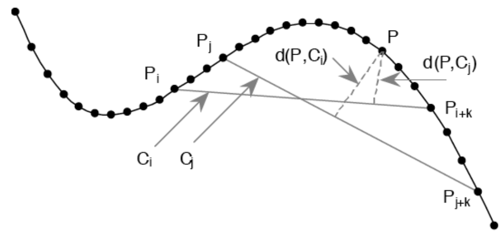

Fig. 1. Illustration of the Phillips and Rosenfeld algorithm.

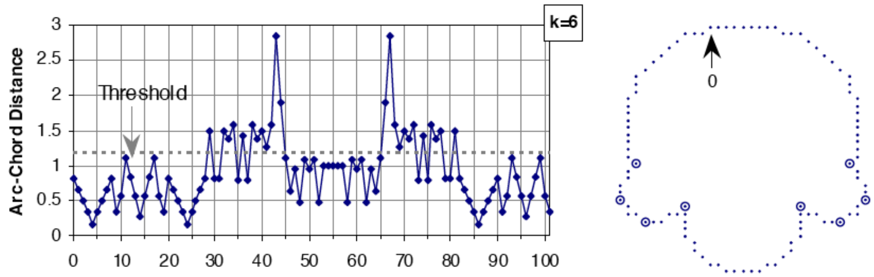

Fig. 2. Arc-chord distance measure and the corners detected by the Phillips-Rosenfeld algorithm using $k = 6$.

condition is satisfied:

$$M(p, C) \geq M(p_x, C), \text{ for all } p_x \in \{p_{i-(k/2)}, \ldots, p_{i+(k/2)}\}$$

To demonstrate the effect of thresholding, the Semicircles shape [11] and its associated arc-chord distance for $k = 6$ are shown in Figure 2. It is clear from this example that we cannot expect the suggested threshold of $k/5$ to work in all cases. Although lowering the threshold value will enable us to detect the missed corners in this example, it may result in many spurious corners for other test shapes.

A potential problem with the local maximum determination scheme is illustrated in Figure 3, which shows an isolated corner model[1] and its arc-chord distance using $k = 6$. In this example, peak A will be suppressed by some points in its neighborhood with higher significance although non-of these points satisfy the local maximum criterion.

Figure 4 demonstrates that the inclusion of the sign information in the definition of the arc-chord distance can prevent some peaks from being masked by

[1] A synthetic curve segment with a single corner. Thus, there are no near by corners that may affect the resulting arc-chord distance measure.

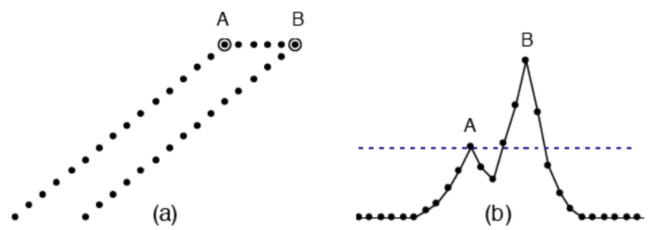

Fig. 3. An isolated corner model (a) and its associated (b) arc-chord distance using $k = 6$.

other neighboring peaks of opposite concavity (convexity). This figure shows an isolated corner model and its associated signed and unsigned arc-chord distance measures using $k = 6$. While peak B in the unsigned measure of Figure 4(a) will be discarded by the non-maximum suppression scheme, it has a better chance of being detected if the signed measure of Figure 4(b) is used instead. In addition, the inclusion of the sign information provides valuable evidence about the concavity and convexity of the curve without introducing any overhead on the subsequent calculations.

Figure 5 demonstrates that the height of a peak is not sufficient by itself to quantify the peak. In this example, although peak A is more visually prominent, it may be suppressed by peak B whose height is larger than that of peak A. This suggests that other criteria should be considered to quantify the strength of the peaks.

The final issue that was not explicitly discussed in [4] is that of plateaus. It is possible to have adjacent curve points with equal arc-chord distances, and trying to resolve these ties arbitrarily may result in detecting false corners. Although this may not cause noticeable problems for real borders, a properly designed algorithm should be able to handle these cases at least systematically.

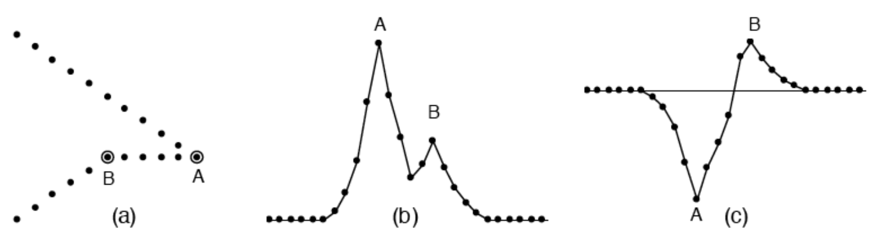

Fig. 4. An isolated corner model (a) and its associated unsigned (b) and signed (c) arc-chord distance measure using $k = 6$.

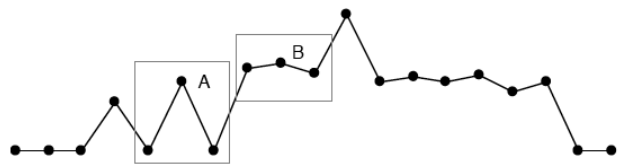

Fig. 5. The height of a peak is not indicative of its prominence.

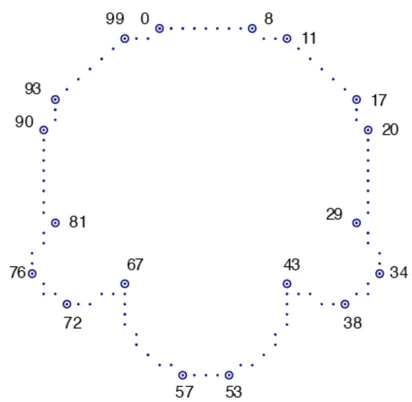

Fig. 6. Semicircles shape used to illustrate the new algorithm. Border points have been numbered for convenience.

3 The new algorithm

The steps of the new algorithm will be described below and will be illustrated with the aid of the Semicircles shape shown in Figure 6.

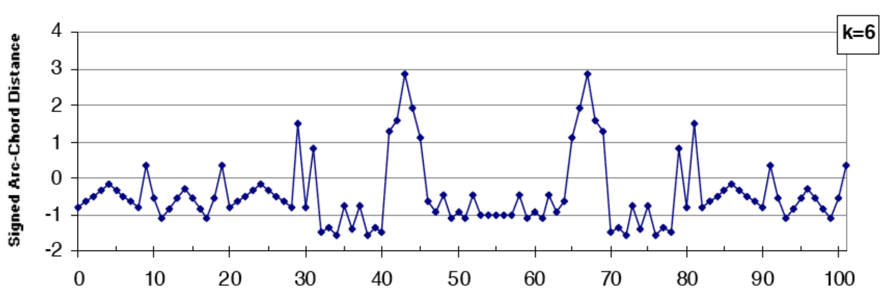

Fig. 7. Signed arc-chord distance for the Semicircles shape using $k = 6$.

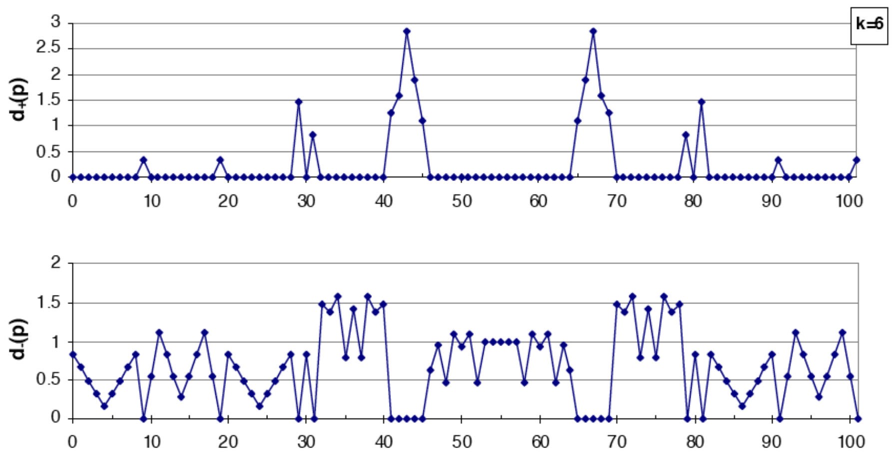

Fig. 8. The functions $d_+(p)$ and $d_-(p)$ for the Semicircles shape using $k = 6$.

1. Compute the arc-chord distance for each border point using the method of Phillips and Rosenfeld. Here however, we use the signed distance from the point to the chord instead of the absolute distance value. For the Semicircles shape, this measure is shown in Figure 7 using $k = 6$.
2. Separate the signed arc-chord distance function $d(p)$ into two functions $d_+(p)$ and $d_-(p)$ as follows

$$d_+(p) = \begin{cases} d(p), \text{ if } d(p) \geq 0 \\ 0 \qquad \text{otherwise} \end{cases} \quad \text{and} \quad d_-(p) = \begin{cases} \mid d(p) \mid, \text{ if } d(p) < 0 \\ 0 \qquad\qquad \text{otherwise} \end{cases}$$

   Figure 8 shows these two functions for the measure of Figure 7.
3. The signals $d_+(p)$ and $d_-(p)$ are processed separately where a search procedure is applied to detect the local maximum points. For each point p_i, we attempt to find the largest possible window that contains p_i such that the significance of all of the points in that window to both the left and right of p_i is strictly decreasing. If such a window exists, then p_i is considered a local maximum point, and the leftmost $P_L(p_i)$ and rightmost $P_R(p_i)$ points of that window are recorded. For example, in Figure 8 ($d_-(p)$), point 28 is a local maximum point with $P_L(p_i) = 24$ and $P_R(p_i) = 29$.
   The two endpoints of *valid* plateaus are handled differently. A plateau whose leftmost and rightmost end points are, respectively, p_x and p_y, is considered to be valid if $d(p_x) > d(p_x - 1)$ and $d(p_y) > d(p_y + 1)$. In this case, we set $P_R(p_x) = P_R(p_y)$ and $P_L(p_y) = P_L(p_x)$. This is illustrated in Figure 9, which represents a segment of the function of Figure 8. In this example, $P_L(p_y) = P_L(p_x) = 52$ and $P_R(p_x) = P_R(p_y) = 58$.
4. The significance of each local maximum point p_i found in the previous step is evaluated as the area of the triangle whose vertices are the points $(p_x, d_\pm(p_x))$

where $p_x = [P_L(p_x), P_R(p_x)]$. This is shown in Figure 10 for the Semicircles shape.

5. The *mean significance* μ is calculated for all the local maximum points. In the above example (for instance) we have 40 local maximum points whose mean significance evaluates to 1.39 (see Figure 10).
6. All local maximum points whose significance value is greater than or equal to the average μ are marked as candidate corners. For the Semicircles test shape, this results in the following 18 points: 11, 17, 32, 34, 36, 38, 40, 43, 53, 57, 67, 70, 72, 74, 76, 78, 93, and 99.
7. The remaining local maxima points are sorted according to their significance in descending order (most significant first) and processed sequentially. For every local maximum point p_i, we consider the two candidate corners that proceed and succeed p_i; denote these two points by p_l and p_r, respectively, as shown in Figure 11.
   Then p_i is considered a candidate corner if

   $$d \geq 1 \quad \text{and} \quad \frac{d}{L} \geq \frac{d_1 d_2}{L^2} \sin\alpha$$

   where α is set to 155°. The first condition is based on the fact that a slanted straight line is quantized into a set of horizontal and vertical line segments separated by one-pixel steps. In addition, we assume that the "border noise" is no more than one pixel, and if the noise level is known a priori, this threshold can be adjusted accordingly. The second condition allows us to detect the vertex of a triangle whose vertex angle is less than α. In order to preserve the symmetry of the shape, all local maxima points with equal significance level are processed in the same iteration.
   For the Semicircles shape, the first iteration examines points 49, 51, 59 and 61 (since all of them have the same significance); all these points satisfy the two conditions and are hence marked as candidate dominant points. The second iteration examines points 8, 20, 28, 82, and 90; none of these points satisfy the two conditions. The process continues until all local maxima points are examined. The output of this step is shown in Figure 12.
8. Because in step 6 we added all the peaks with "above-average" significance without paying attention to their proximity (in terms of border pixels), it

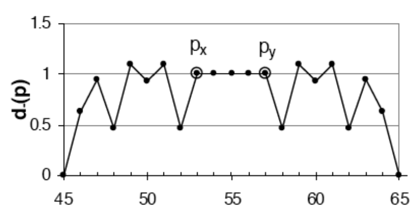

Fig. 9. Handling plateaus.

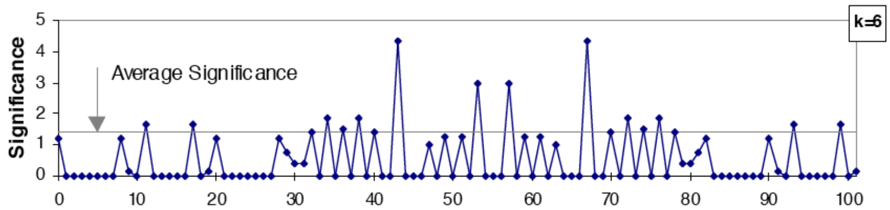

Fig. 10. Significance measure for the Semicircles shape using $k = 6$.

is reasonable to believe that some of the marked candidate corners do not correspond to true corners of the curve. The purpose of the current step is to suppress the false corners (if any). First, we calculate the ratio d/L (see Figure 11) for all candidate corners and sort these corners in ascending order (lowest first). Candidate corners with $d < 1$ are considered insignificant and marked for deletion. Here also, we process all points with identical d/L value in the same iteration. The final result after this step is shown in Figure 13. The figure also shows the corners detected by the Phillips-Rosenfeld algorithm.

4 Experimental results

To see the difference between the new algorithm and the Phillips-Rosenfeld algorithm, the scale space map for the Semicircles shape is shown in Figure 14 using k values in the range $[3, 10(= N/10)]$. Note that the Phillips-Rosenfeld algorithm did not detect any corners for the curve segment $[43, 67]$ for several scales whereas the results of the new algorithm were consistent to within a tolerance of $1 - 2$ pixels. In fact, the polygon generated by the new algorithm did provide a visually pleasing approximation of the shape for all the considered values of k.

The results of the new algorithm for some test shapes are shown in Figure 15. In all cases, we used a value of $k = N/15$, were N is the number of border points.

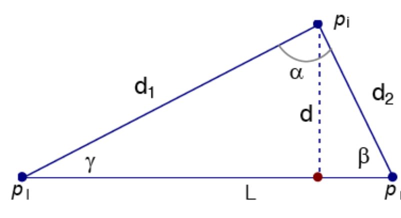

Fig. 11. Conditions for testing local maximum point p_i.

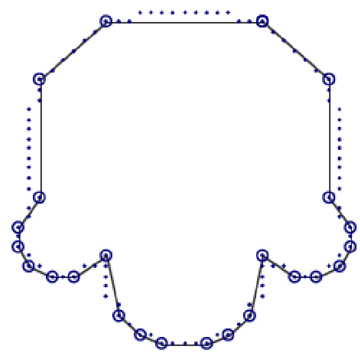

Fig. 12. Candidate corners after processing all local maxima points.

5 Conclusions

We have described a new algorithm for curve partitioning using the arc-chord distance formulation. The algorithm can tolerate the effect of the scale parameter k to an acceptable degree.

References

1. U. Ramer. An iterative procedure for the polygonal approximation of plane closed curves. *Computer Graphics Image Processing*, **1**:244–256, 1972.
2. W.S. Rutkowski and A. Rosenfeld. A comparison of corner-detection techniques for chain-coded curves. Technical Report TR-623, Computer Science Center, University of Maryland, 1978.

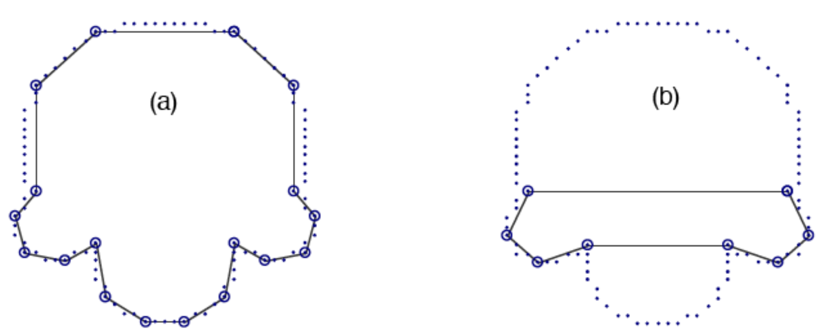

Fig. 13. Corners of the Semicircles shape using $k = 6$ produced by the current algorithm (a) and the Phillips-Rosenfeld algorithm (b).

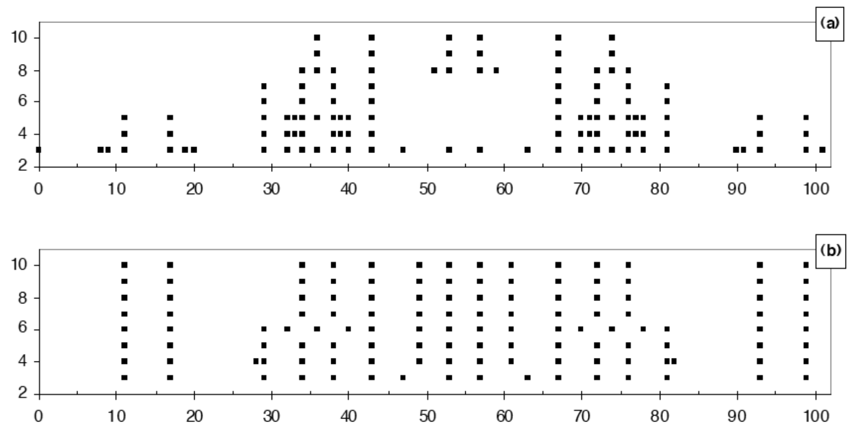

Fig. 14. Scale-space map for the Semicircles shape: (a) Phillips-Rosenfeld algorithm and (b) new algorithm.

3. M. Fischler and R. Bolles. Perceptual organization and curve partitioning, *IEEE Trans. Pattern Analysis Machine Intelligence*, **8**:100–105, 1986.
4. T.Y. Phillips and A. Rosenfeld. A method for curve partitioning using arc-chord distance. *Pattern Recognition Letters*, **5**: 285–288, 1987.
5. J.H. Han. Detection of convex and concave discontinuous points in a plane curve. In *Proc. 3rd Int. Conf. Computer Vision*, December 4-7, Osaka, Japan, pages 71–74, 1990.
6. Y. Lin, J. Dou, and H. Wang. Contour shape description based on an arc height function. *Pattern Recognition*, **25**: 17–23, 1992.
7. H. Aoyama and M. Kawagoe. A piecewise linear approximation method for preserving visual feature points of original figures. *CVGIP: Graphical Models and Image Processing*, **53**: 435–446, 1991.
8. S.-Y. Wu and M.-J. Wang. Detecting the dominant points by the curvature-based polygonal approximation. *CVGIP: Graphical Models and Image Processing*, **5**: 79–88, 1993.
9. M.A. Fischler and H.C. Wolf. Locating perceptually salient points on planar curves. *IEEE Trans. Pattern Analysis Machine Intelligence*, **16**:113–129, 1994.
10. J.H. Han and T. Poston. Chord-to-point distance accumulation and planar curvature: a new approach to discrete curvature. *Pattern Recognition Letters*, **22**:1133–1144, 2001.
11. C.-H. The and R.T. Chin. On the detection of dominant points on digital curve. *IEEE Trans. Pattern Analysis Machine Intelligence*, **11**: 859–872, 1989.

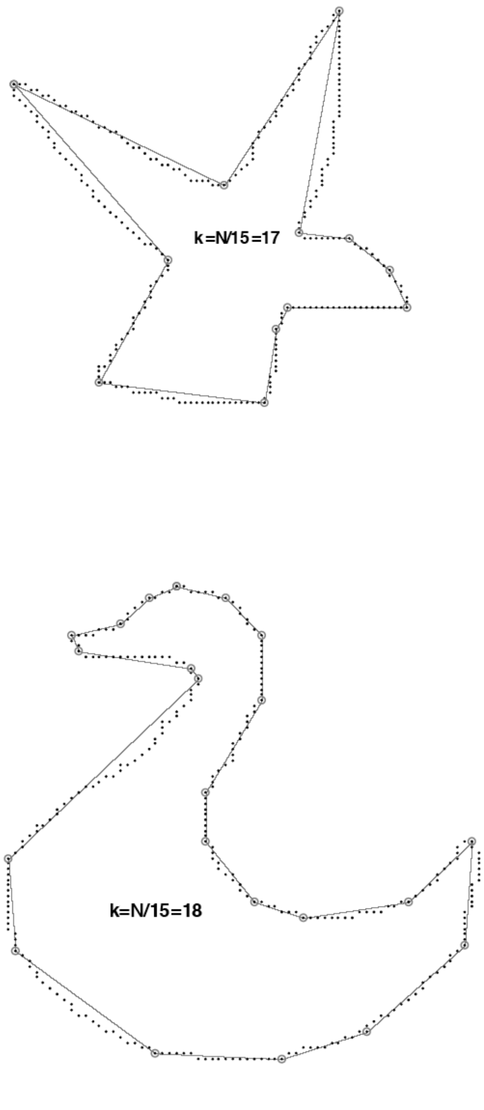

Fig. 15. Results of testing the new algorithm on test shapes.