A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves [1)]

Dmitry Chetverikov and Zsolt Szabó
Image and Pattern Analysis Group,
Computer and Automation Research Institute
Budapest, Kende u.13-17, H-1111 HUNGARY
`http://visual.ipan.sztaki.hu`

Abstract:
A new algorithm is proposed for detection of corners and other high curvature points in planar curves. A corner is defined as a location where a triangle with specified opening angle and size can be inscribed in the curve. The tests compare the new algorithm to four alternative algorithms for corner detection.

1 Introduction

This paper deals with detection of high curvature points in planar curves. It is well known [1] that these points play a dominant role in shape perception by humans. Locations of significant changes in curve slope are, in that respect, similar to intensity edges. If these characteristic contour points are identified properly, a shape can be represented in an efficient and compact way with accuracy sufficient in many shape analysis problems.

Corner detection in planar curves is related to corner detection in grayscale images which is not addressed here. Characteristic contour points have traditionally been in the focus of the scale space theory [5] that allows for a 'natural', although sophisticated and computationally demanding, definition of such points at varying scale. However, in many online applications, especially in industry, processing time is a crucial issue. Computational load is to be minimized without significant loss of robustness.

Various less complicated corner detection algorithms have been developed. A number of frequently cited approaches are discussed in the survey by Liu and Srinath [4], where comparative

---

[1)] To appear in: Proc. 23^{rd} Workshop of the Austrian Pattern Recognition Group, 1999. This work was supported in part by OTKA grant T026592. The authors acknowledge contribution of Beáta Puxler to implementing the alternative corner detection algorithms.

experimental results are also given. Four of the algorithms tested in [4] are used in our tests as well, namely those by Rosenfeld and Johnston [6], Rosenfeld and Weszka[7], Freeman and Davis [3], and Beus and Tiu [2]. In this paper, these algorithms are referred to as RJ73, RW75, FD77, and BT87, respectively. RW75 is a modification of RJ73, while BT87 is a modification of FD77. A summary of the four algorithms is given in section 2.

Although the notion of corner seems to be intuitively clear, no generally accepted mathematical definition exists, at least for digital curves. In a sense, different approaches give different — but conceptually related — computational definitions to a visual phenomenon. The lack of unequivocal ground truth makes comparative performance evaluation tests less significant than they could, and should, be.

In this paper we present a new, fast and efficient algorithm for detection of high curvature points. (For simplicity, they will be called 'corner points'.) The parameters of the algorithm are easy to understand and tune to particular sharpness and scale. The new algorithm, referred to as IPAN99, is described in section 3. (IPAN stands for Image and Pattern Analysis group.) Experimental results shown in section 4 compare the new method to the alternative ones mentioned above.

2 Summary of four corner detectors

To make the paper self-contained, in this section we give a brief summary of the four alternative corner detection algorithms used in our comparative tests. The summary is based on the survey [4]. Each algorithm inputs a chain-coded curve that is converted into a connected sequence of grid points $\mathbf{p}_i = (x_i, y_i)$, $i = 1, 2 \ldots, N$. A measure of corner strength ('cornerity') is assigned to each point, then corner points are selected based on this measure. For each approach, we summarize these two main steps and list the parameters of the algorithm and their default ('best') values. Setting of the parameters is discussed in more detail in section 4.

When processing a point $\mathbf{p}_i$, the algorithms consider a number of subsequent and previous points in the sequence, as candidates for the arms of a potential corner in $\mathbf{p}_i$. For a positive integer k, the forward and the backward *k-vectors* at point $\mathbf{p}_i$ are defined as

$$\mathbf{a}_{ik} = (x_i - x_{i+k}, y_i - y_{i+k}) = (X_{ik}^+, Y_{ik}^+) \tag{1}$$

$$\mathbf{b}_{ik} = (x_i - x_{i-k}, y_i - y_{i-k}) = (X_{ik}^-, Y_{ik}^-) \tag{2}$$

where X_{ik}^+, Y_{ik}^+ and X_{ik}^-, Y_{ik}^- are the components of $\mathbf{a}_{ik}$ and $\mathbf{b}_{ik}$, respectively.

2.1 Rosenfeld and Johnston RJ73

Corner strength. k-cosine of the angle between the k-vectors is used, which is defined as

$$c_{ik} = \frac{(\mathbf{a}_{ik} \cdot \mathbf{b}_{ik})}{|\mathbf{a}_{ik}||\mathbf{b}_{ik}|} \tag{3}$$

Selection procedure. Starting from $m = \kappa N$, k is decremented until c_{ik} stops to increase: $c_{im} < c_{i,m-1} < \ldots < c_{in} \not< c_{i,n-1}$. $k = n$ is then selected as the best value for the point i. A corner is indicated in i if $c_{in} > c_{jp}$ for all j such that $|i - j| \leq n/2$, where p is the best value of k for the point j.
Parameter. The single parameter κ specifies the maximum considered value of k as a fraction of the total number of curve points N. This limits the length of an arm at κN. The default value is $\kappa = 0.05$.

2.2 Rosenfeld and Weszka RW75

Corner strength. Averaged k-cosine of the angle between the k-vectors is used, which is defined as

$$\overline{c}_{ik} = \begin{cases} \frac{2}{k+2} \sum\limits_{t=k/2}^{k} c_{it} & \text{if } k \text{ is even,} \\ \frac{2}{k+3} \sum\limits_{t=(k-1)/2}^{k} c_{it} & \text{if } k \text{ is odd,} \end{cases}$$

where c_{it} are given by (3).
Selection procedure. Same as in RJ73, but for $\overline{c}_{ik}$.
Parameter. Same as in RJ73, with the same default $\kappa = 0.05$.

2.3 Freeman and Davis FD77

Corner strength. In a point i, the angle between the x-axis and the backward k-vector defined in (2) is given as

$$\theta_{ik} = \begin{cases} \tan^{-1}\left(Y_{ik}^{-}/X_{ik}^{-}\right) & \text{if } |X_{ik}^{-}| \geq |Y_{ik}^{-}|, \\ \cot^{-1}\left(X_{ik}^{-}/Y_{ik}^{-}\right) & \text{otherwise,} \end{cases}$$

The incremental curvature is then defined as

$$\delta_{ik} = \theta_{i+1,k} - \theta_{i-1,k} \tag{4}$$

Finally, the k-strength in i is computed as

$$S_{ik} = \ln t_1 \cdot \ln t_2 \cdot \sum_{j=i}^{i+k} \delta_{jk} \tag{5}$$

where

$$t_1 = \max\{t : \delta_{i-v,k} \in (-\Delta, \Delta), \forall\ 1 \leq v \leq t\} \text{ and}$$
$$t_2 = \max\{t : \delta_{i+k+v,k} \in (-\Delta, \Delta), \forall\ 1 \leq v \leq t\}$$

account for the effect of the forward and backward arms as the maximum spacings (numbers of steps from i) that still keep the incremental curvature δ_{ik} within the limit $\pm\Delta$. The latter is set as

$$\Delta = \arctan \frac{1}{k-1} \tag{6}$$

Selection procedure. A point i is selected as a corner if S_{ik} exceeds a given threshold S and individual corners are separated by a spacing of at least $k+1$ steps.
Parameters. The two parameters are the spacing k and the corner strength threshold S. The default values are $k = 5$ and $S = 1500$.

2.4 Beus and Tiu BT87

Corner strength. Similar to FD77, with the following modifications. The arm cutoff parameter τ is introduced to specify the upper limit for t_1 and t_2 as a fraction of N:

$$t_1 = \max\{t : \delta_{i-v,k} \in (-\Delta, \Delta), \forall\ 1 \leq v \leq t, \text{ and } t \leq \tau N\}$$
$$t_2 = \max\{t : \delta_{i+k+v,k} \in (-\Delta, \Delta), \forall\ 1 \leq v \leq t, \text{ and } t \leq \tau N\},$$

where δ_{ik} and Δ are given by (4) and (6), respectively. The corner strength is obtained by averaging (5) between two values k_1 and k_2:

$$S_i = \frac{1}{k_2 - k_1 + 1} \sum_{k=k_1}^{k_2} S_{ik}$$

Selection procedure. Same as in FD77.
Parameters. The four parameters are the averaging limits k_1 and k_2, the arm cutoff parameter τ, and the corner strength threshold S. The default values are $k_1 = 4$, $k_2 = 7$, $\tau = 0.05$, and $S = 1500$.

3 The new algorithm

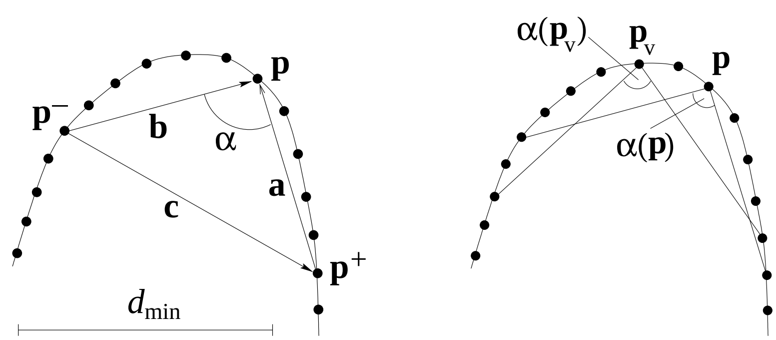
The proposed two-pass algorithm defines a corner in a simple and intuitively appealing way, as a location where a triangle of specified size and opening angle can be inscribed in a curve. A curve is represented by a sequence of points $\mathbf{p}_i$ in the image plane. The ordered points are densely sampled along the curve, but, contrary to the other four algorithms, no regular spacing between them is assumed. A chain-coded curve can also be handled if converted to a sequence of grid points. In the first pass the algorithm scans the sequence and selects candidate corner points. The second pass is post-processing to remove superfluous candidates.

First pass. In each curve point $\mathbf{p}$ the detector tries to inscribe in the curve a variable triangle $(\mathbf{p}^-, \mathbf{p}, \mathbf{p}^+)$ constrained by a set of simple rules:

$$\begin{aligned} d_{min}^2 \le |\mathbf{p} - \mathbf{p}^+|^2 \le d_{max}^2 \\ d_{min}^2 \le |\mathbf{p} - \mathbf{p}^-|^2 \le d_{max}^2 \\ \alpha \le \alpha_{max}, \end{aligned} \tag{7}$$

where $|\mathbf{p} - \mathbf{p}^+| = |\mathbf{a}| = a$ is the distance between $\mathbf{p}$ and $\mathbf{p}^+$, $|\mathbf{p} - \mathbf{p}^-| = |\mathbf{b}| = b$ the distance between $\mathbf{p}$ and $\mathbf{p}^-$, and $\alpha \in [-\pi, \pi]$ the opening angle of the triangle. (See figure 1a.) The latter is computed as

$$\alpha = \arccos \frac{a^2 + b^2 - c^2}{2ab}$$



(a) (b)

Figure 1: Detecting high curvature points. (a) Determining if p is a candidate point. (b) Testing p for sharpness non-maxima suppression.

Variations of the triangle that satisfy the conditions (7) are called admissible. Search for the admissible variations starts from $\mathbf{p}$ outwards and stops if any of the conditions (7) is violated. (That is, a limited number of neighboring points is only considered.) Among the admissible variations, the least opening angle $\alpha(\mathbf{p})$ is selected. $\pi - |\alpha(\mathbf{p})|$ is assigned to $\mathbf{p}$ as the *sharpness* of the candidate. If no admissible triangle can be inscribed, $\mathbf{p}$ is rejected and no sharpness is assigned.

Considering the vector product of $\mathbf{b} = (b_x, b_y)$ and $\mathbf{c} = (c_x, c_y)$, it is easy to see that the corner is convex if $b_x c_y - b_y c_x \ge 0$, otherwise it is concave.

Second pass. The sharpness based non-maxima suppression procedure is illustrated in figure 1b. A corner detector can respond to the same corner in a few consecutive points. Similarly to edge detection, a post-processing step is needed to select the strongest response by discarding the non-maxima points.

A candidate point $\mathbf{p}$ is discarded if it has a sharper valid neighbor $\mathbf{p}_v$: $\alpha(\mathbf{p}) > \alpha(\mathbf{p}_v)$. In the current implementation, a candidate point $\mathbf{p}_v$ is a valid neighbor of $\mathbf{p}$ if $|\mathbf{p} - \mathbf{p}_v|^2 \leq d_{max}^2$. As alternative definitions, one can use $|\mathbf{p} - \mathbf{p}_v|^2 \leq d_{min}^2$ or the points adjacent to $\mathbf{p}$.

Parameters. d_{min}, d_{max} and α_{max} are the parameters of the algorithm. d_{min} sets the scale (resolution), with small values responding to fine corners. The upper limit d_{max} is necessary to avoid false sharp triangles formed by distant points in highly varying curves. α_{max} is the angle limit that determines the minimum sharpness accepted as high curvature. In practice, we often set $d_{max} = d_{min} + 2$ and tune the remaining two parameters, d_{min} and α_{max}. The default values are $d_{min} = 7$ and $\alpha_{max} = 150°$.

4 Tests

Five corner detection algorithms have been implemented and tested. One of them is the proposed algorithm, IPAN99, while the other four are those listed in section 2. The test shapes were taken from the paper [4]. The printed images of [4] were scanned, which introduced some noise into the original noise-free pictures. In addition, limited random noise was added to the scanned images to better test the robustness of the algorithms. The eight resulting, noisy test shapes are shown in figure 2.

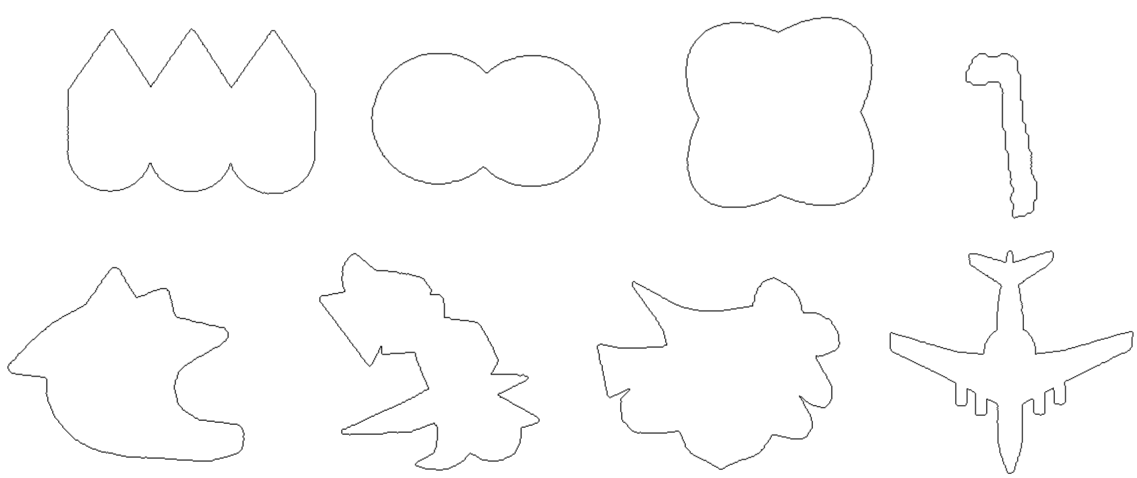


Figure 2: Shapes used in the tests.

From the practical point of view, the following performance evaluation criteria are applicable to corner detectors. (1) *Selectivity*: The rate of the correct detections should be high, the rate of the wrong ones should be low. (2) *Single response*: Each corner should be detected only once. (3) *Precision*: The positions of the detected corners should be precise. (4) *Robustness* *to noise.* (5) *Easy setting of parameters*: Given a new task, it should be easy to tune the parameters to this task. Ideally, different categories of shapes should be correctly processed with no need to significantly modify the parameters. (6) *Robustness to parameters*: Minor changes in parameters should not cause drastic changes in performance. (7) *Speed.*

Similar criteria are often used to characterize edge detectors. To empirically evaluate most of the criteria, one needs a large reference database with ground truth. Such a database is available for edge detection. Unfortunately, to our best knowledge, no ground-truthed database has been created for corner detection. Partially, this is because the notion of a corner seems to be more task-dependent and subjective than the notion of an intensity edge. Traditionally, certain 'standard' collections of shapes have been used to compare corner detectors. The one we use has also appeared in several studies, including that by Liu and Srinath [4].

However, Liu and Srinath have less noisy images and tune the parameters of an algorithm to each of the shapes, so as to obtain the best possible result for each particular shape. In practice, there is usually no way to manually tune the parameters to each particular input. For this reason, we decided to concentrate on the default parameter values, as those values that provide the best overall performance for the whole set of input shapes. These default values are given in sections 2 and 3. Only when the defaults yielded a poor result they were modified until an acceptable result was obtained. Because of the lack of ground truth, a subjective judgment of the detection quality was applied.

The main pictorial results are shown in figures 3–7. IPAN99 distinguishes convex and concave corners which are marked differently. The parameter values for which the results were obtained are summarized in table 1. (Deviations from the default values D are only shown.) The proposed algorithm yields reasonable results at the default values for all the 8 shapes. For 3 of them, somewhat better results can be obtained when the parameters are slightly modified, as shown in the bottom row of figure 7. For stable performance, FD77 and BT87 need more frequent modification of their parameters. (In case of BT87, only S needed to be varied.) Because of the higher level of noise and due to different settings of parameters, the overall performance of the four alternative algorithms is worth than that reported in [4].

Table 1: Parameter values for the 8 test images.

Algorithm	im1	im2	im3	im4	im5	im6	im7	im8
RJ73	D	7	D	D	D	D	D	D
RW75	D	10	D	D	D	D	D	D
FD77	D	$7, 2500$	$7, 2500$	$5, 500$	D	$7, 1000$	D	D
BT87	D	D	D	500	1000	1300	D	1000
IPAN99	D	D	D	D	D	D	D	D

Table 2 shows average CPU times per shape, in milliseconds, used by the algorithms for 4 typical shapes of different complexity. The algorithms were run 100 times on a 333 MHz PC using the default parameters. Clearly, execution time depends on implementation and parameter values. Some general conclusions can still be drawn. Contour 4 is much shorter than the other seven, so the processing times are the smallest in this case. RJ73, RW75 and IPAN99 depend on the number of corners more strongly than FD77 and BT87. The difference in processing times is significant for complex shapes.

Table 2: Typical processing times.

Algorithm	im2	im4	im7	im8
RJ73	24.3	14.5	52.7	87.9
RW75	109.7	54.2	246.1	422.6
FD77	25.3	5.9	20.4	24.7
BT87	50.8	19.3	66.2	83.1
IPAN99	10.6	10.1	17.2	27.4

5 Conclusion

The corner detection algorithms described in this paper are available for online testing over the Internet at the following web site: `http://visual.ipan.sztaki.hu`. Two testing options are provided. One can either select one of the standard shape images provided, or submit his own image and obtain the results. This gives a potential user a possibility to select an algorithm that fits his application.

References

[1] H. Asada and M. Brady. The curvature primal sketch. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:2–14, 1986.

[2] H.L. Beus and S.S.H. Tiu. An improved corner detection algorithm based on chain-coded plane curves. *Pattern Recognition*, 20:291–296, 1987.

[3] H. Freeman and L.S. Davis. A corner finding algorithm for chain-coded curves. *IEEE Trans. Computers*, 26:297–303, 1977.

[4] H.-C. Liu and M.D. Srinath. Corner detection from chain-code. *Pattern Recognition*, 23:51–68, 1990.

[5] F. Mokhtarian and A.K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14:789–805, 1992.

[6] A. Rosenfeld and E. Johnston. Angle detection on digital curves. *IEEE Trans. Computers*, 22:875–878, 1973.

[7] A. Rosenfeld and J.S. Weszka. An improved method of angle detection on digital curves. *IEEE Trans. Computers*, 24:940–941, 1975.

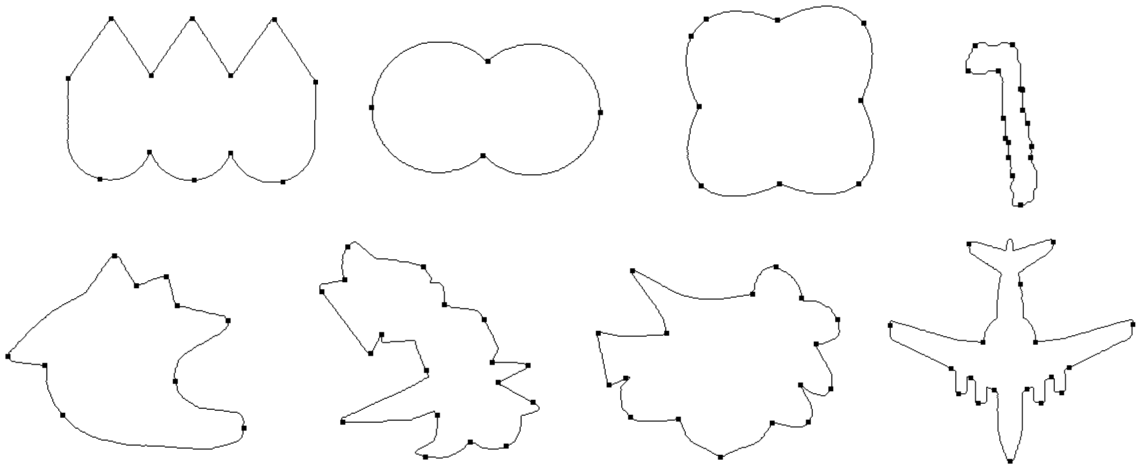


Figure 3: Results of RJ73.

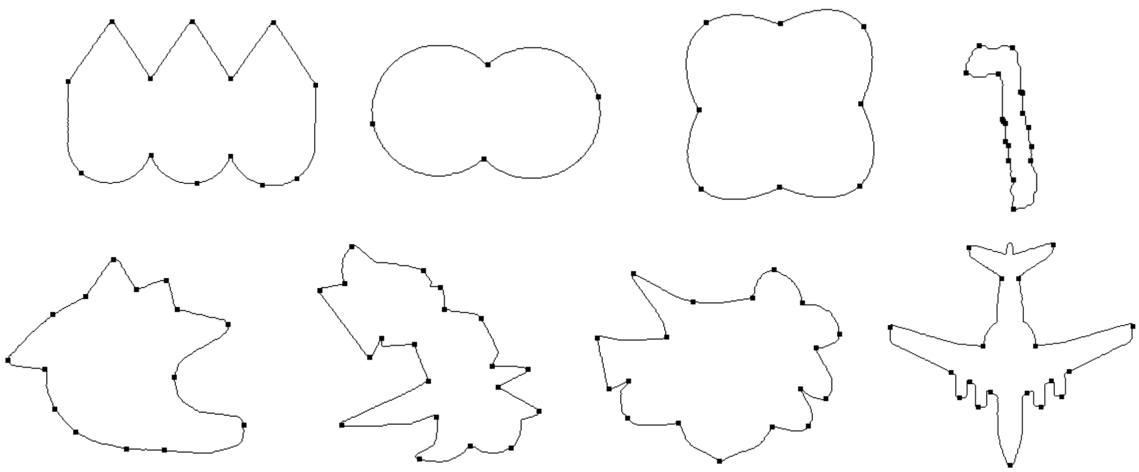


Figure 4: Results of RW75.

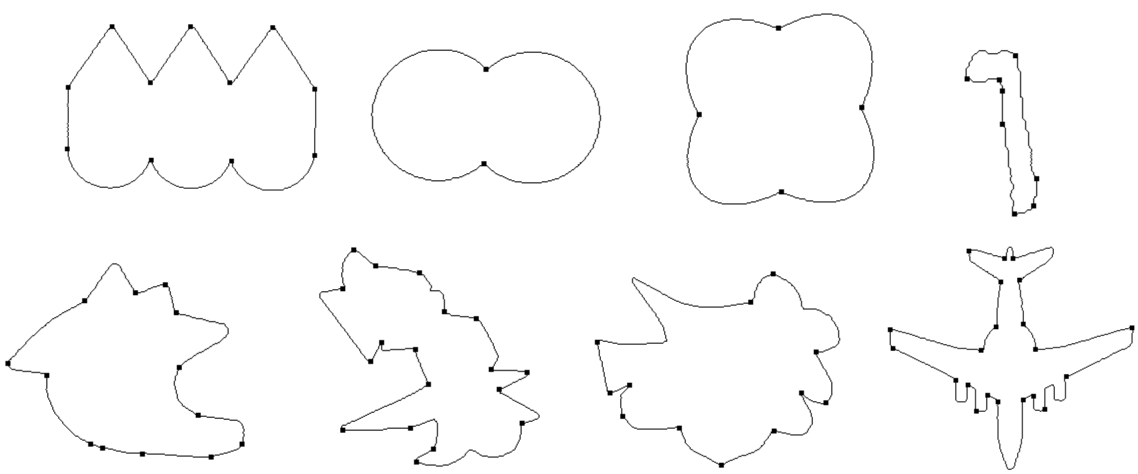


Figure 5: Results of FD77.

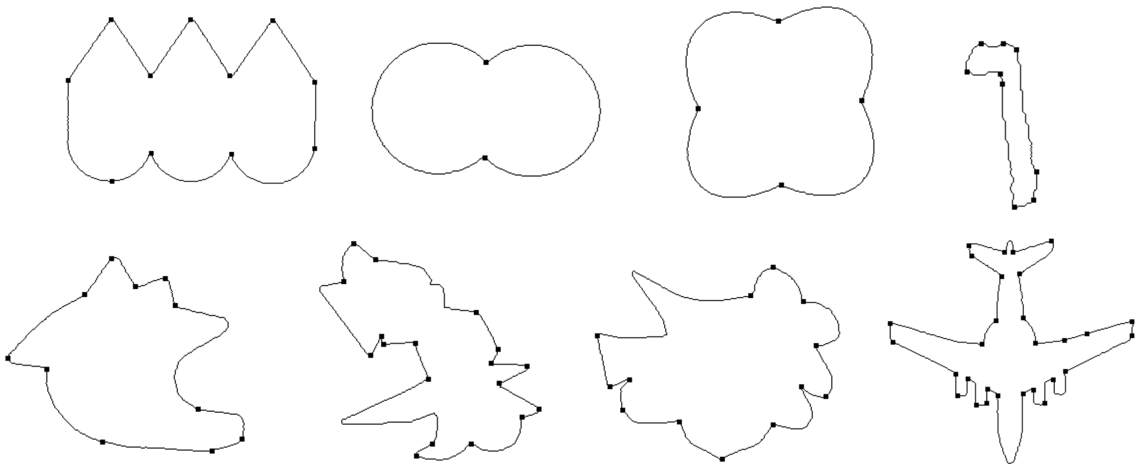


Figure 6: Results of BT87.

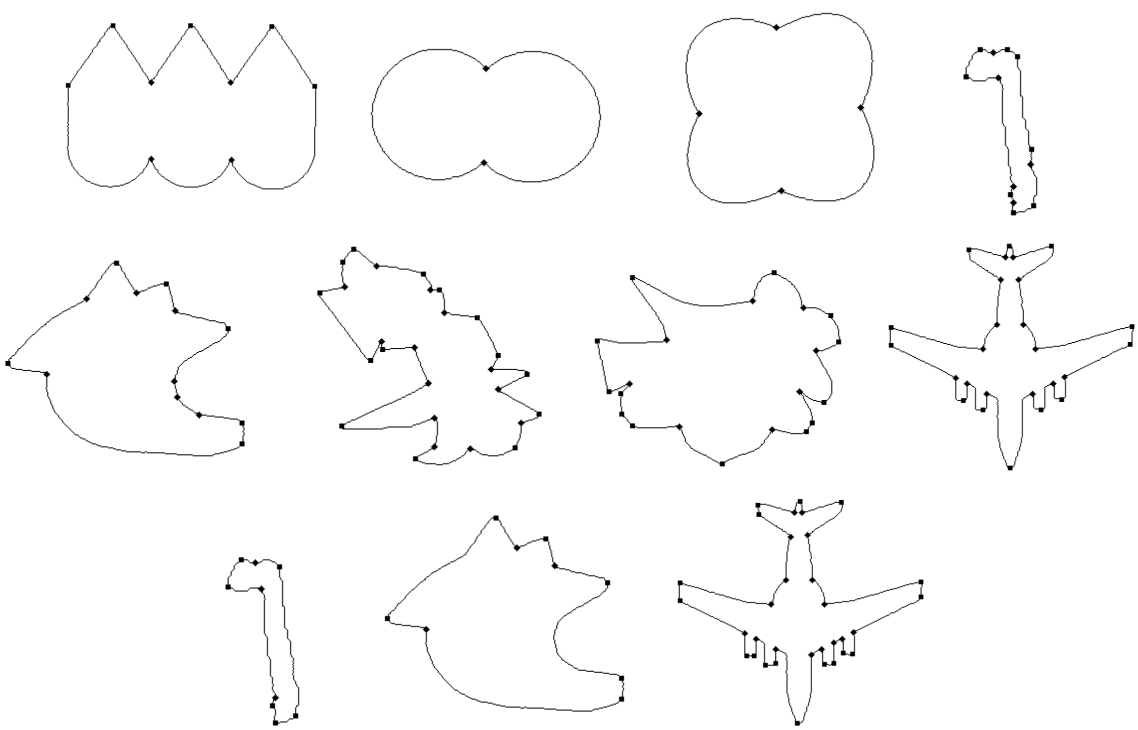


Figure 7: Results of IPAN99. For the bottom row, the parameter values differ from the default ones: left and center $8, 140°$**, right** $5, 140°$**.**