A novel superellipse segmentation of planar curves

Wu-Chih Hu
Department of Computer Science and Information Engineering, National Penghu University,
#300, Liu-Ho Road, Makung City, Penghu, Taiwan, ROC
wchu@npu.edu.tw

Abstract

Superellipse is a flexible primitive and it can represent a large variety of shapes. This paper presents a novel superellipse segmentation of planar curves based on the types of breakpoints. In the proposed method, the breakpoints are divided into corners and smooth joints, and the types of the segments on both sides of a breakpoint are identified. Using these breakpoint types can effectively reduce the computational cost, and reserve the important features of planar curves. Tests show that the proposed method has a number of interesting properties including being scale invariant, threshold-free, and efficient.

***Keywords*:** superellipse, segmentation, breakpoint.

1. Introduction

A key problem area in computer vision is the extraction of meaningful features from images. Curve segmentation is one of the most important jobs since a segmented contour can be used to describe an object in a meaningful form for higher level image processing, such as shape analysis and pattern recognition. The most popular approach is based on edge data which is better represented in a more manageable form. Many techniques have been proposed for this purpose in the past two decades. Polygonal approximation is the simplest approach. The line segments are almost extracted from curves based on the corner detection [1-3] or dominant point detection [4-6]. But polygonal approximation is rarely used for further shape analysis. To go from curve segmentation to shape analysis, one could include higher order primitives such as circular arcs [7, 8], conic arcs [9, 10], and splines [11, 12] in curve segmentation.

Curve segmentation using the conic arcs or splines would obtain a flexible description, and these primitives are important in our daily life and in industry. But, a more flexible primitive is always sought and studied. Superellipse provides an interesting form for representing a range of objects whose shapes may be deformed by altering various squareness parameters. With the squareness parameter, superellipse is able to represent a wide variety of shapes such as rounded rectangles, ellipses, diamonds, pinched diamonds, etc. Hence, the superellipse is a flexible primitive in computer vision, and curve segmentation using superellipses is proposed in the last decade [13-17].

Superellipses were first formulated by Gardiner [18], and the three dimensional version--superquadrics--were popularized in computer graphics by Barr [19] and in computer vision by Pentland [20]. A superellipse centered on the origin with its axes aligned with the coordinate system can be represented by the following implicit equation.

$$(x/a)^{2/\varepsilon} + (y/b)^{2/\varepsilon} = 1 \qquad (1)$$

Its parametric form at angle ϕ is given by

$$\begin{cases} x(\phi) = a\cos^{\varepsilon}\phi \\ y(\phi) = b\sin^{\varepsilon}\phi \end{cases}, 0 < \phi \le 2\pi \qquad (2)$$

where a and b are the lengths of the major and the minor axes, respectively, and ε is called the squareness. A superellipse with different values of the squareness ε, it can represent a wide variety of shapes, as shown in Fig. 1, where superellipses are shown with $a = 2b$ and ε values of 0.1, 0.5, 1.0, 2.0, 5.0 and 10.0 (from the exterior to the interior contours, respectively).

Detection of a superellipse involves estimating the six associated parameters, that is, the center (x_0, y_0), the lengths of the major and the minor axes (a, b), the orientation θ and the squareness ε. Traditional approaches are computationally expensive for estimating these parameters since the cost function is nonlinear and nonlinear programming is usually employed. For instance, Rosin et al. [13-16] use Powell's conjugate direction technique [21] to adjust the six associated parameters with initial values. The squareness ε is set to 1.0 and the other parameters are selected from an approximating ellipse.

In the procedure proposed by Rosin et al., the six parameters $(x_0, y_0, \theta, a, b, \varepsilon)$ of a superellipse all are repeatedly estimated by Powell's conjugate direction technique based on edge data or line segments from polygonal approximation, hence the computational cost is high. Besides, the important feature—corner--is not reserved. In this paper, superellipse segmentation based on types of breakpoints [7] is proposed to effectively reduce the computational cost, and reserve the important feature (corner) of planar curves. The advantages of breakpoint types are that it is:

- Threshold-free—No threshold within the algorithm,
- Stable—Invariant to transformations of the data (rotation, translation, and scale), and
- Extendible—Types of breakpoints can be extended to other primitives.

In the proposed scheme, the breakpoints are categorized as five types: *c-ll*, *c-la*, *c-aa*, *s-la*, and *s-aa* by using AKC function and PHF [7], where *c* indicates a corner and *s* is a smooth joint. These types of breakpoints are very useful for superellipse segmentation to reduce the amount of segments employed in the merging iteration of superellipse fitting, that is, it can effectively reduce the computational cost. This concept using breakpoint-types proposed in this paper is not referred in the existent methods for superellipse segmentation.

In the remainder of this paper, the breakpoint classification is presented in Section 2. In Section 3, the superellipse segmentation is proposed. Section 4 presents experimental examples and evaluations of the results. Finally, conclusion is made in Section 5.

2. Breakpoint classification

In this section, the associated breakpoints are first detected by using the methods of dominant point or corner detection [1-6], and then the breakpoints are categorized as five types: corner-*ll*, corner-*la*, corner-*aa*, smooth joint-*la* or smooth joint-*aa* by using AKC (adaptive *k*-curvature) function and PHF (projective height function) [7]. The type *ll* means that the segments on both sides of the breakpoint are line segments; *la* stands for a joint of a line segment and an arc; *aa* represents a joint of two arcs; and corner is a discontinuous tangent and smooth joint is associated with continuous tangent, but discontinuous curvature [22].

The AKC function is briefly described as follows. Consider three consecutive breakpoints P_{i-1}, P_i and P_{i+1}, and two consecutive segments S_i and S_{i+1} starting at P_{i-1}, joining at P_i and ending at P_{i+1}. Let lengths of S_i and S_{i+1} be l_i and l_{i+1}, respectively, and $k = \min(l_i, l_{i+1})$. And, let the region-of-support for P_i be $[P_i - \tilde{k}, P_i + \tilde{k}]$, where $\tilde{k} = k/2$ and define the *k*-vector at a point $P_j \in [P_i - \tilde{k}, P_i + \tilde{k}]$ as

$$\vec{a}_{jk} = (P_{j^+}(x) - P_j(x), P_{j^+}(y) - P_j(y)) \quad (3a)$$

$$\vec{b}_{jk} = (P_{j^-}(x) - P_j(x), P_{j^-}(y) - P_j(y)) \quad (3b)$$

where $P_{j^+} = P_j + \tilde{k}$ and $P_{j^-} = P_j - \tilde{k}$, and then the *k*-cosine between $\vec{a}_{jk}$ and $\vec{b}_{jk}$ is

$$c_{jk} = \frac{\vec{a}_{jk} \cdot \vec{b}_{jk}}{\|\vec{a}_{jk}\|\|\vec{b}_{jk}\|} \quad (4)$$

The AKC function of P_i is defined as the values c_{jk} $\forall P_j \in [P_i - \tilde{k}, P_i + \tilde{k}]$.

And, the PHF is described as follows. Let the intervals for evaluating the projective heights (PHs) of the segments on both sides of the breakpoint P_i be $[P_i - 3\tilde{k}/2, P_i - \tilde{k}/2]$ for the segment S_i, and $[P_i + \tilde{k}/2, P_i + 3\tilde{k}/2]$ for the segment S_{i+1}, which insure that the PH evaluation does not exceed either P_{i-1} and P_i or P_i and P_{i+1}. Consider a point $P_j \in [P_i - 3\tilde{k}/2, P_i - \tilde{k}/2]$, and the associated endpoints $\hat{P}_{j^-} = P_j - \tilde{k}/2$ and $\hat{P}_{j^+} = P_j + \tilde{k}/2$. Then, the parametric form for the line segment between the endpoints is

$$\begin{cases} x = \hat{P}_{j^-}(x) + \lambda_x t \\ y = \hat{P}_{j^-}(y) + \lambda_y t \end{cases}, t \in [0,1] \quad (5)$$

where $\lambda_x = \hat{P}_{j^+}(x) - \hat{P}_{j^-}(x)$ and $\lambda_y = \hat{P}_{j^+}(y) - \hat{P}_{j^-}(y)$, and the projective height of P_j is

$$h(P_j) = \sqrt{\frac{[\lambda_x(P_j(y) - \hat{P}_{j^-}(y)) - \lambda_y(P_j(x) - \hat{P}_{j^-}(x))]^2}{\lambda_x^2 + \lambda_y^2}} \quad (6)$$

The PHF is defined as the projective heights $h(P_j)$ over the interval of $[P_i - 3\tilde{k}/2, P_i - \tilde{k}/2]$ or $[P_i + \tilde{k}/2, P_i + 3\tilde{k}/2]$. Define a line accumulator and an arc accumulator on the region of support of PHF and use 0.5 to discriminate a line from an arc, since it is the maximum possible deviation of the PH of a digitized line segment can make. If $h(P_j) < 0.5$, then the line accumulator is incremented by one; otherwise add one to the arc accumulator. The type of segment is determined by comparing the values in the two accumulators. If the value in the line accumulator is greater, then the segment is identified as a line; otherwise it is an arc.

The procedure of breakpoint classification is described as follows. The breakpoints can be categorized as corners and smooth joints by testing a global maximum in the AKC function. Clearly, if there is a global maximum at the breakpoint in the AKC function, then it is a corner; otherwise it is a smooth joint. And, using the PHF scheme, the types of segments on both sides of a breakpoint can be efficiently detected and a breakpoint can be further divided into types *ll*, *la* (or *al*), and *aa*. Combining the results of AKC and PHF schemes, the breakpoints are categorized as five types: corner-*ll* (*c*-*ll*), corner-*la* (*c*-*la*), corner-*aa* (*c*-*aa*), smooth joint-*la* (*s*-*la*) and smooth joint-*aa* (*s*-*aa*). For understanding the geometry of corner and smooth joint, the breakpoint types can be illustrated clearly from Fig. 2. Fig. 2(a) is *c*-*ll*, Fig. 2(b) and Fig. 2(c) are *c*-*la*, Fig. 2(d)~2(f) are *c*-*aa*, Fig. 2(g) is *s*-*la*, and Fig. 2(h) and 2(i) are *s*-*aa* breakpoints.

Practically, however, due to the errors in breakpoint-detection, two situations may be brought about. Hence, the breakpoint compensation is necessary. The first case is that there is a type *c*-*ll* followed by a type *c*-*aa* (or *s*-*aa*) immediately (Fig. 3(a)), and there must be a type *s*-*la* between these two consecutive breakpoints. To determine the location of such a type *s*-*la*, form a straight line by connecting the type *c*-*ll* and *c*-*aa* (or *s*-*aa*). The point in the original segment that is farthest from this connected line segment is then considered as such a type *s*-*la*, following the idea by Duda and Hart [23]. The result of the first case is shown in Fig. 3(b).

The second case is that there is a type *c*-*la* surrounded by two types *c*-*aa* (or *s*-*aa*) immediately (Fig. 4(a)), then there must be a type *s*-*la* preceding or following the type *c*-*la*. The location of such a type *s*-*la* can be obtained by using the PHF of the type *c*-*la*. That is, if there is a line segment on the forward side of *c*-*la*, then the new *s*-*la* should be on the forward segment, and vice versa. The exact location of the new *s*-*la* is also determined at the point where the distance from the line connecting the *c*-*la* and *c*-*aa* (or *s*-*aa*) is the maximum. The result of the second case is shown in Fig. 4(b).

3. Superellipse segmentation

From the geometric property (see Fig. 1), it is obvious that $\varepsilon = 2$ produces a diamond; $0 < \varepsilon < 2$ generates convex superellipses (where $\varepsilon = 1$ is an ellipse); and $\varepsilon > 2$ results in concave superellipses (pinched superellipses). And, the contours are symmetric to the center of the superellipse.

For concave superellipses, corners appear in the contours, and they are important features in image processing. Besides, the major and the minor axes of a superellipse become shortened as a result of truncation. This happens in the contour tracing to form a closed contour for a superellipse detection when there is another trace that branches out. In this case, the 1-pixel wide branch is considered as noise and hence is truncated.

Hence, the proposed superellipse segmentation based on breakpoint types is considered to deal with the convex superllipses ($0 < \varepsilon < 2$).

The cost function of superellipse segmentation is defined as the Euclidean distance d_p along the line passing through the point (x_p, y_p) and the center of the superellipse [14].

$$d_p = \sqrt{(x_p - x_e)^2 + (y_p - y_e)^2} \tag{7}$$

For a superellipse centered on the origin with axes aligned with the coordinate system, superellipse point (x_e, y_e) are formulated as

$$x_e = \left| \frac{1}{(1/a)^{2/\varepsilon} + (y_p / x_p b)^{2/\varepsilon}} \right|^{\varepsilon/2} \tag{8a}$$

$$y_e = x_e (y_p / x_p) \tag{8b}$$

Superellipse segmentation based on breakpoint types is done as described below. The first fitted segment is the first segment for the open contour, or the segment following corners for the closed contour. But, if the segment between consecutive corners is line, then line segment is reserved and not fitted. The Powell's conjugate direction technique [21] is used to adjust the six associated parameters with initial values. The squareness ε is set to 1.0 and the (x_0, y_0, a, b) parameters are selected from an approximating ellipse [24]. The orientation θ is estimated by finding the principal axis of the data,

$$\theta = \frac{1}{2} \tan^{-1} \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \tag{9}$$

where μ_{pq} is the (p, q)th central moment calculated by

$$\mu_{pq} = \sum_i (x_i - \bar{x})^p (y_i - \bar{y})^q \tag{10}$$

and $(\bar{x}, \bar{y})$ is the centroid for the i data points.

Powell's algorithm performs a gradient descent as follows. (i) Set the parameters $(x_0, y_0, \theta, a, b, \varepsilon)$ obtained above as initial values. (ii) Do a gradient descent by varying the first parameter. Once the minimum of the cost function has been found, repeat for the second parameter and so on for all parameters. (iii) Combine the change in the parameters into a vector and minimize the cost function by gradient descent along this vector. (iv) Repeat from stage (ii) until the change of parameters is sufficiently small.

When the superellipse fitting to the first segment is finished, the region-of-support would be extended by using the merging procedure to obtain larger superelliptic arc, and avoid redundancy.

Two conditions must be obeyed in the merging procedure for two consecutive segments: (i) If the breakpoint is a corner, then these segments can't be merged. (ii) Two consecutive segments must be convex.

If consecutive segments satisfy the above two conditions, then the merging procedure is performed as follows. Calculate the measurement μ of new superelliptic arc for the two consecutive segments, and then compare the original measurement. If the former is smaller than latter, then the consecutive segments are successfully merged to a new superelliptic arc. The next consecutive segment is added in merging procedure, and then the above process is iterated until the merging procedure is failed. Then, the new segment is fitted, and the merging procedure is performed again.

$$\mu = \frac{\text{maximum deviation}}{\text{segment length}} \tag{11}$$

Using the measurement μ, the lower the measurement is, the more significant the superelliptic arc is. That is, the longer the superelliptic arc is, the greater the deviation that will be tolerated. Thus, the same shape of curve at different scales will have the same significance.

4. Experimental results

Two experiments have been done to demonstrate the performance of the proposed superellipse segmentation. The original data is shown in Fig. 5(a). Fig. 5(b) is the result of superellipse fitting, and the superellipse segmentation is indicated in Fig. 5(c). In Fig. 5(c), the corners (indicated as circles) and line segments are reserved. If using Rosin's method, the corners and line segments are fitted as $\varepsilon = 2$ superellipses (diamonds). In the proposed method, corners are reversed to obtain the important features in image processing, and line segments are not further treated to effectively reduce the computational cost in the iteration of superellipse fitting. Hence, the proposed superellipse segmentation based on breakpoint types is more efficient than Rosin's method.

Fig. 6(a) is the overlapping curves, and the result of superellipse fitting is shown in Fig. 6(b). Sueprellipse segmentation is indicated as Fig. 6(c). For the overlapping curves, the triple joint type is used in the superellipse segmentation. The triple joint is defined as the joint between the overlapping objects. In superellipse fitting, the property of the triple joint is like as the corner, and the associated segments of the triple joint are not employed in the merging procedure.

5. Conclusion

The superellipse segmentation based on breakpoint types is proposed successfully. In the result of superellipse segmentation, superelliptic arcs, line segments, and corners are obtained to describe planar curves more meaningfully. In the proposed method, the breakpoints are categorized as five types: *c-ll*, *c-la*, *c-aa*, *s-la*, and *s-aa* by using AKC function and PHF, where *c* indicates corner and *s* is smooth joint.

Using the breakpoint types, the segments employed in merging iteration of superellipse fitting can be effectively reduced, that is, the computational cost can be effectively reduced.

In the proposed superellipse segmentation, since no threshold is required, the result is not influenced by the selection of thresholds. That is, the performance of the proposed scheme is threshold-free. Further, using the significance measurement μ (the ratio of the maximum deviation divided by the segment length) can obtain the performance that the same shape of curve at different scales will have the same significance. Besides, the concept using breakpoint-types proposed in this paper is not referred in the existent methods for superellipse segmentation, and the performance is better than Rosin's method [13-16].

Acknowledgements

This paper has been supported by the National Science Council, Taiwan, ROC under Grant No. NSC 93-2213-E-346-001.

References

[1] J. Sporring, O.F. Olsen, M. Nielsen, and J. Weickert, "Smoothing images creates corners", Image and Vision Computing, Vol. 18, pp. 261-266, 2000.

[2] H.T. Sheu, and W.C. Hu, "A rotationally invariant two-phase scheme for corner detection", Pattern Recognition, Vol. 29, pp. 819-828, 1996.

[3] A. Rattarangsi, and R.T. Chin, "Scale-based detection of corner of planar curves", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, pp. 430-449, 1992.

[4] W.Y. Wu, "Dominant point detection using adaptive bending value", Image and Vision Computing, Vol. 21, pp. 517-525, 2003.

[5] B. Sarkar, S. Roy, and D. Sarkar, "Hierarchical representation of digitized curves through dominant point detection", Pattern Recognition Letters, Vol. 24, pp. 2869-2882, 2003.

[6] C.H. Teh, and R.T. Chin, "On the detection of dominant points on digital curves", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 11, pp. 859-872, 1989.

[7] H.T. Sheu, and W.C. Hu, "Multiprimitive segmentation of planar curves---A two-level breakpoint classification and tuning approach", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, pp. 791-797, 1999.

[8] J.H. Horng, "An adaptive smoothing approach for fitting digital planar curves with line segments and circular arcs", Pattern Recognition Letters, Vol. 24, pp. 565-577, 2003.

[9] W.C. Hu, "Multiprimitive segmentation based on meaningful breakpoints for fitting digital planar curves with line segments and conic arcs", Image and Vision Computing, Vol. 23, pp. 783-789, 2005.

[10] Z. Zhang, “Parameter estimation techniques: a tutorial with application to conic fitting”, Image and Vision Computing, Vol. 15, pp. 59-76, 1997.

[11] W.C. Hu, and H.T. Sheu, “Quadratic B-spline for curve fitting", Proceedings of the National Science Council, R.O.C., Part A: Physical Science and Engineering, Vol. 24, pp. 373-381, 2000

[12] G. Medioni, and Y. Yasumoto, “Corner detection and curve representation using cubic B-splines”, Computer Vision, Graphics, and Image Processing, pp. 267-278, 1987.

[13] P.L. Rosin, and G.A.W. West, “Nonparametric segmentation of curves into various representations”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, pp. 1140-1153, 1995.

[14] P.L. Rosin, and G.A.W. West, “Curve segmentation and representation by superellipse”, IEE Proceeding--Vision, Image and Signal Proceeding, Vol. 142, pp. 280-288, 1995.

[15] P.L. Rosin, “Fitting superellipses”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, pp. 726-732, 2000.

[16] X. Zhang, and P.L. Rosin, “Superellipse fitting to partial data”, Pattern Recognition, Vol. 36, pp. 743-752, 2003.

[17] W.C. Hu, and H.T. Sheu, “An efficient and consistent method for superellipse detection”, IEE Proceeding--Vision, Image and Signal Processing, Vol. 148, pp. 227-233, 2001.

[18] M. Gardiner, “The superellipse: a curve that lies between the ellipse and the rectangle”, Scientific American, Vol. 21, pp. 222-234, 1965.

[19] A.H. Barr, “Superquadrics and angle-preserving transformations”, IEEE Computer Graphics Applications, Vol. 1, pp. 11-23, 1981.

[20] A. Pentland, “Perceptual organisation and the representation of natural form”, Artificial Intelligence, Vol. 28, pp. 293-331, 1986.

[21] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, “Numerical Recipes in C”, Cambridge University Press, Cambridge, UK, 1992.

[22] H. Asada, and M. Brady, “The curvature primal sketch”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, pp. 2-14, 1986.

[23] R.O. Duda, and P.E. Hart, “Pattern Classification and Scene Analysis”, Wiley, New York, 1973.

[24] A. Albano, “Representation of digitised contours in terms of conic arcs and straight line segments”, Computer Vision, Graphics and Image Processing, Vol. 3, pp. 23-33, 1974.

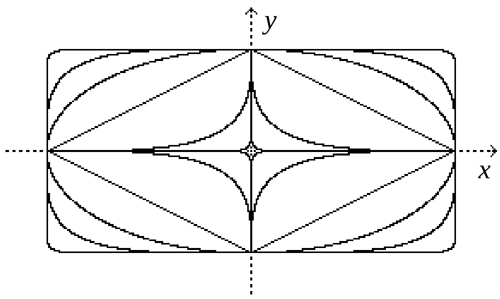


Fig. 1. Superellipses with different squareness.

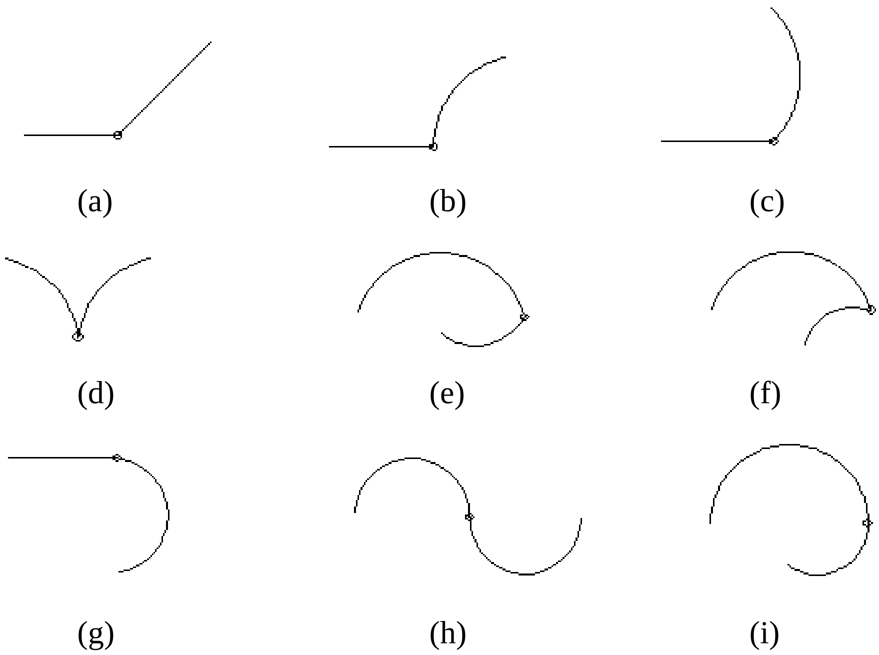


Fig. 2. Breakpoint types of corners and smooth joints.

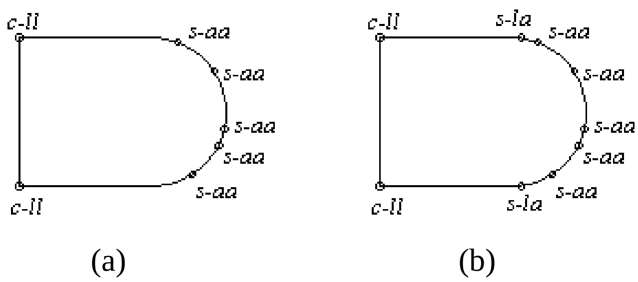


Fig. 3. Recovery of type *s-la* joints.

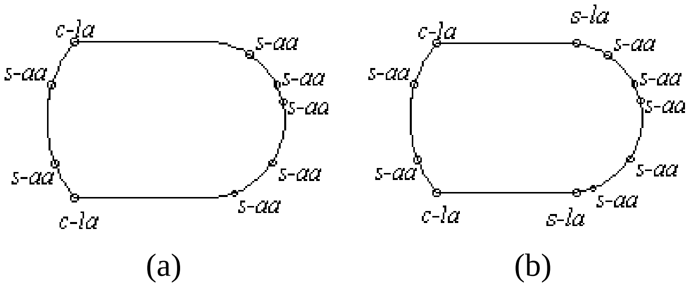
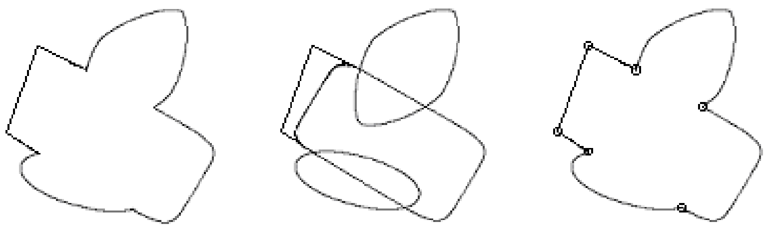
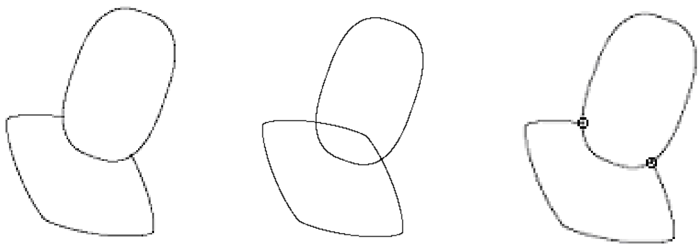


Fig. 4. Recovery of type *s-la* joints.



(a) (b) (c)

Fig. 5. Superellipse segmentation of curves.



(a) (b) (c)

Fig. 6. Superellipse segmentation of overlapping curves.