中華民國八十六年全國計算機會議

平面曲線之多圖元分割法---兩層級分割點之分類及調整法
Multiprimitive segmentation of planar curves --- A two-level breakpoint classification and tuning approach

許新添
Sheu Hsin-Teng
國立台灣科技大學電機系
Department of Electrical Engineering, National Taiwan University of Science and Technology
sheu@mouse.ee.ntust.edu.tw

胡武誌
Hu Wu-Chih
國立台灣科技大學電機系
Department of Electrical Engineering, National Taiwan University of Science and Technology
d8207017@mouse.ee.ntust.edu.tw

摘要

本論文提出一個雙層級式的分割點分類及調整法針對平面曲線從事多圖元分割。本方法除了具有簡單、快速及無需門檻值等優點外，對於量化及前置處理所產生的誤差均具有強健性，在物體比對或識別等應用上有其實用的價值。

關鍵字：多圖元分割，分類，分割點，調整。

Abstract

A two-level breakpoint classification and tuning approach is proposed for the multiprimitive segmentation of planar curves. The advantages of the proposed scheme are that it is simple, fast, threshold-free, and robust to noise and quantization and preprocessing errors, thus allowing it to be employed in a variety of application such as matching and recognition.

Keywords: multiprimitive segmentation, classification, breakpoint, tuning.

1. Introduction

Segmentation of digitized planar curves is one of the most important jobs in early image processing since a segmented outline can be used to describe an object in a meaningful and compact form to facilitate higher level vision processing, such as pattern recognition and shape analysis. Most authors adopt polygonal approximation in which curves are divided into a series of adjoining line segments (see Pavlidis [1]; Teh [2]; Sheu [3]; and their references).

Polygonal approximation is simple but is rarely used for further shape analysis. To go from segmentation to shape analysis, one could include higher order primitives such as circular arcs, ellipses, splines etc. in the segmentation. In most applications, since circular arcs and line segments [4] can be used to approximate an arbitrary curve quite well they will be considered in this research.

In the segmentation of planar curves into line segments and circular arcs, Wu and Rodd [5] have presented a 'seven-criteria' approach by first doing angle detection [6] to locate sharp corners, and then determining whether the segment between a pair of corners is better represented by a line segment, a circular arc or a combination segment by testing against thresholds. If it is a combination segment, the Gaussian convolution is used to insert new joints. The interesting property is that accurate estimation and fast convergence of the parameters of the circular arcs are achieved. However, the results are strongly influenced by the threshold selections.

Rosin and West [7, 8] have described a line and arc detection (LAD) algorithm for multiprimitive segmentation. The first stage in their approach is to find line segments by using Lowe's work [9] and a significant measure. Arc approximation of the curve is then exercised using the same idea as in polygonal approximation until there are less than three line segments. An interesting advantage of the LAD method is that no threshold is required. However, the computation cost for both determining the parameters of the arcs and establishing the tree can be high. Besides, two parts of the supposedly same arc may not be merged if they do not belong to the same branch in the interpretation tree.

Ichoku et. al. [10] have developed a dynamic focusing algorithm (DFA) that employed a fit-and-decrement procedure to detect lines and circular arcs. The interesting property of the DFA is its simplicity. However, the precision of the results are influenced by the thresholds for curve fitting and the step-size chosen. Besides, the computation cost is also high in curve fitting and the results may not be the same if starting from different point.

In this paper, breakpoints are assumed to be available, which can be detected by any method in the literature. In the proposed two-level breakpoint classification and tuning approach, an adaptive *k*-curvature (AKC) function is proposed to identify a breakpoint as either a corner or a smooth joint. Then, a projective height function (PHF) is employed to further identify the type of a breakpoint to be corner-*ll*, corner-*la*, corner-*aa*, smooth joint-*la* or smooth joint-*aa*. Here, *ll* means that the segments on both sides are line segments; *la* stands for a joint of a line segment and an arc; and *aa* represents a joint of two arcs. Subsequently, the joints are tuned to merge consecutive segments or insert new joints and adjust their locations to achieve more accurate and stable segmentation. With the suggested joint tuning, which has not been seen in the literature, recovery of line segments and merging/splitting of circular arcs are accomplished. The advantages of the proposed method include: simplicity, reduced computation cost, and no threshold required. Besides, the proposed scheme is robust to noise and quantization errors, and the set of the identified breakpoints is the same even start working at different point and/or in different direction. Further, the proposed scheme can also correct the wrong results obtained in breakpoint detection.

2. Two-level breakpoint classification

In this section, the two-level breakpoint classification approach is illustrated. In level 1, the AKC function is defined and is used to classify the breakpoints into corners and smooth joints. In level 2, the proposed PHF is employed to further identify the types of segments on both sides of a breakpoint. The reasons of using the AKC function in breakpoint classification and PHF to distinguish the segment types are that the AKC function is sensitive to the type of breakpoints but is susceptible to quantization errors whereas the PHF is robust to quantization errors yet is not sensitive to the breakpoint-type.

2.1 Adaptive *k*-curvature function

Asada and Brady [11] have divided the breakpoints into two types: corner and smooth joint, both indicate isolated curvature changes. The former corresponds to a discontinuous tangent to the contour, as shown in Fig. 1 for which many algorithms have been developed successfully [2, 3, 6, 12]; whereas the latter is associated with continuous tangent but discontinuous curvature, as shown in Fig. 2 and the localization of which usually need more elaboration [5, 11]. Having defined the corner and smooth joint, Asada and Brady [11] then determined these breakpoints by the multiscale convolutions of the curve with the first and second derivatives of Gaussian. Such an approach, although seems complete in the sense of scale and has theoretical values, is computationally intensive. In our method, the AKC function is used to divided into corners and smooth joints the breakpoints using an accurate and efficient rotationally invariant method [3] although many methods in the literature can be used as well.

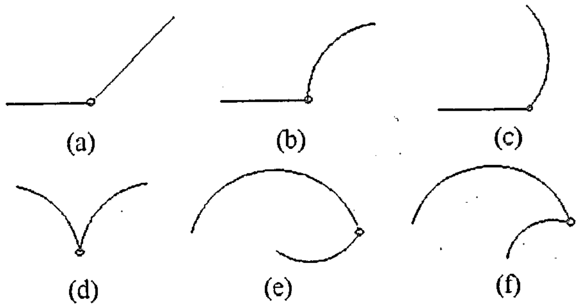


Fig. 1 Corners.

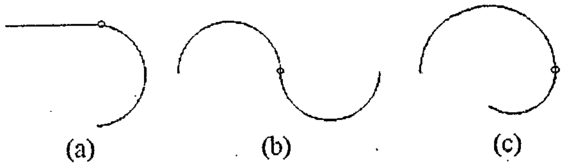


Fig. 2 Smooth joints.

The AKC function is defined based on the *k*-curvature values [13] 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 the lengths of S_i and S_{i+1} be l_1 and l_2, respectively, and $k = \min(l_1, l_2)$. Let the region of support for P_i be $[P_i - \tilde{k}, P_i + \tilde{k}]$, as shown in Fig. 3, 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

$$\bar{a}_{jk} = \left(P_{j^+}(x) - P_j(x),\ P_{j^+}(y) - P_j(y)\right) \tag{1a}$$

$$\bar{b}_{jk} = \left(P_{j^-}(x) - P_j(x),\ P_{j^-}(y) - P_j(y)\right) \tag{1b}$$

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

$$c_{jk} = (\bar{a}_{jk} \cdot \bar{b}_{jk}) / \|\bar{a}_{jk}\| \, \|\bar{b}_{jk}\| \tag{2}$$

Define the AKC function of P_i be the values c_{jk} $\forall P_j \in [P_i - \tilde{k}, P_i + \tilde{k}]$, then the type of a breakpoint P_i can be determined by considering only the associated AKC function, independent of other breakpoints. The independence from other breakpoints is important for precise breakpoint-type discrimination, since the type of both the breakpoint and the segments on both sides are completely determined.

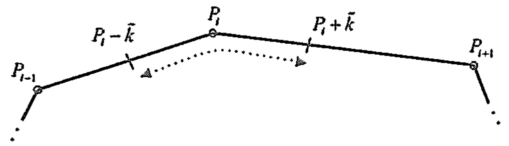
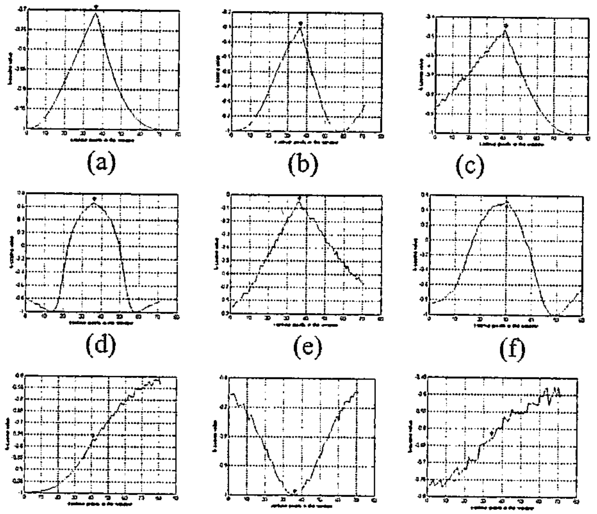


Fig. 3 The region-of-support of AKC function.

The AKC functions for the curves shown in Figs. 1 and 2 are given in Fig. 4 (the locations corresponding to the breakpoints are indicated by arrows). The meaning of the AKC functions shown in Figs. 4(a)~4(f) corresponding to corner-type breakpoints is explained as follows. Since the tangent at the breakpoint is discontinuous, the k-cosines there are greater than those at other points in the region of support. Hence, the AKC function has a global maximum at a corner-type breakpoint. On the other hand, for smooth joints, since the tangents are continuous, the associated k-cosines vary smoothly, and local maximum is not so obvious at the breakpoint, as shown in Figs. 4(g)~4(i). Obviously, if there is a global maximum at the breakpoint, then it is a corner; otherwise it is considered a smooth joint. The advantage of using AKC functions is that the waveform of the AKC function is merely flipped-over horizontally at the breakpoint yet the location of the breakpoint remains the same if the order of the segments is reversed. This overcomes the drawback of DFA in which the results may not be the same if starting from different points or in different directions.


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 4 The AKC functions for the curves of Figs. 1~2. (a)~(f) for Fig. 1(a)~1(f); (g)~(i) for Fig. 2(a)~2(c).

The proposed AKC function can also be used to determine the orientations of the segments joining at a breakpoint. The orientations of consecutive segments are defined as follows:

Definition 2-1

A piece consecutive segments is said to be in a convex direction if there are zero or two valleys on both sides of the breakpoint in the AKC function. It is in a concave direction if there is a single valley.

From the geometry and the definition of the AKC, if the orientation of segments is concave such as Figs. 1(b) and 1(f) then there must be a point at which the vertices of the k-vector, i.e., P_{j^-}, P_j, and P_{j^+}, are co-linear hence a global minimum (a valley) on one side of the breakpoint results, whereas for a convex piece, either no valley (such as Fig. 1(c)) or two valleys (such as Fig. 1(d)) can be obtained.

Although the AKC function can be used to discriminate corners from smooth joints, the type of the segments on both sides are not completely determined. This can be realized from Fig. 4 where a local maximum in the AKC function may arise from either a smooth joint or quantization errors. The scheme in Section 2.2 is developed to deal with this problem.

2.2 Projective height function

Consider three points P_A, P_B and P_C with k-curvature $\cos\theta$ shown in Fig. 5, and define a projective height (PH) h as the shortest distance from P_B to $\overline{P_A P_C}$. The PH is then used to discriminate a line segment from a circular arc since it is less sensitive to quantization errors that may mislead one from taking a line segment as an arc. The robustness of the PH with respect to quantization error is illustrated as follows. Consider a line segment under different orientations from 0^0 to 45^0 with 3^0 increment and the k-vectors with arm lengths varying from 5 to 50 with increment 5. The conventional k-cosines and the PHs of the k-vector are plotted in Fig. 6. Other angles of orientation can be obtained from this plot by symmetric property. The worst situation occurs when the k-cosine of the line segment is -0.981 ($\approx 168.8^0$) at an about 11^0 of orientation. In contrast, by using the projective height of the k-vector, the maximum deviation is reduced to be no more than 0.5 (pixel).

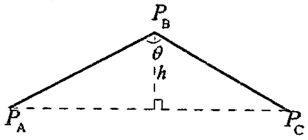
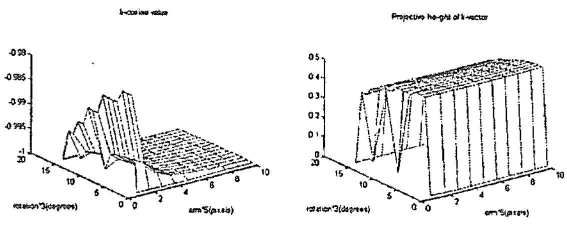


Fig. 5 The model of the projective height.



(a) (b)

Fig. 6 The influence of quantization errors. (a) The k-cosine; (b) The projective height.

Now, if the interval of the AKC function is $[P_i-\tilde{k},\ P_i+\tilde{k}]$, then the intervals for evaluating the PHs of the segments on both sides of the breakpoint P_i are set to be $[P_i-3\tilde{k}/2, P_i-\tilde{k}/2]$ for the segment between P_{i-1} and P_i and $[P_i+\tilde{k}/2, P_i+3\tilde{k}/2]$ for the segment between P_i and P_{i+1}, where the k-vector is evaluated over a range of $\pm\tilde{k}/2$ to insure that the associated boundary for PHs does not exceed the length of the segment between P_{i-1} and P_i or P_i and P_{i+1}.

Consider a point $P_j \in \left[P_i-3\tilde{k}/2, P_i-\tilde{k}/2\right]$, 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 for k-vector evaluation is

$$\begin{cases} x=\hat{P}_{j^-}(x)+\lambda_x\, t \\ y=\hat{P}_{j^-}(y)+\lambda_y\, t \end{cases},\ t\in[0,\ 1] \tag{3}$$

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

$$h(P_j)=\sqrt{\frac{\left[\lambda_x(P_j(y)-\hat{P}_{j^-}(y))-\lambda_y(P_j(x)-\hat{P}_{j^-}(x))\right]^2}{\lambda_x^2+\lambda_y^2}} \tag{4}$$

The PH function (PHF) defined as the projective heights $h(P_j)$ over the interval of $[P_i-3\tilde{k}/2, P_i-\tilde{k}/2]$ or $[P_i+\tilde{k}/2, P_i+3\tilde{k}/2]$ is then used to discriminate a line segment from an arc. Define two accumulators, a line accumulator and an arc accumulator, on the region of support of PHF. From Fig. 6(b), since the maximum projective height of a digitized line segment is no more than 0.5 pixel under any orientations, thus 0.5 is selected to determine every point in the region of support of PHF whether it belongs to a line or an arc segment. If $h(P_j)<0.5$, then the line accumulator is incremented by 1; otherwise add 1 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. By using the PHF 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*. In this paper, since the order in connecting two segments makes no difference, types *la* and *al* are treated in the same way, using type *la* only. Combining the two-levels, the breakpoints are categorized as corner-*ll* (*c-ll*), corner-*la* (*c-la*); corner-*aa* (*c-aa*), smooth joint-*la* (*s-la*) and smooth joint-*aa* (*s-aa*)..

The two-level breakpoint classification scheme can also be used to correct the wrong results obtained in the breakpoint-detection algorithms. For example, if a smooth joint-*ll* is detected, then this breakpoint is illegal hence is removed.

3. Joint tuning

If the breakpoint is a corner, then the segments on both sides can not be merged. However, if it is a smooth joint which may involve line and/or arc segments, then the associated segments may possibly be merged. Line-segment type of smooth joints exist only between type *ll* and *la* joints, whereas arc-type smooth joints will be used to connect only type *la* and *aa* joints. Both will be further exploited in the following.

3.1 Recovery of line segments

Ideally, since a line segment must exist between two consecutive type *c-ll* joints, a type *c-ll* and *c-la* (or *s-la*) joints or two consecutive type *c-la* (or *s-la*) joints, type *c-ll* (or *c-la*) joints can be followed by only type *c-la* (or *s-la*) joints, but not type *c-aa* (or *s-aa*) joints, etc. Practically, however, due to the errors in breakpoint-detection, two situations may be brought about. The first is that a type *c-ll* joint may be followed immediately by a type *c-aa* (or *s-aa*) joint, as shown in Fig. 7(a) and the second is that a type *c-la* joint may be surrounded immediately by two type *c-aa* (or *s-aa*) joints, as shown in Fig. 8(a). Both indicate that a missing line segment has to be recovered by inserting a type *s-la* joint at the point in the original curve that is farthest from the line connecting types *c-ll* and *s-aa* (Fig. 7(b))/types *c-la* and *s-aa* (Fig. 8(b)).

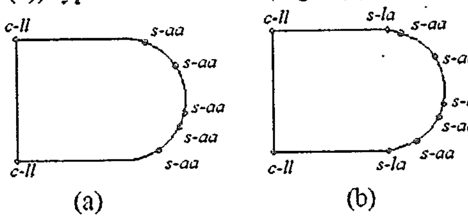


Fig. 7 Recovery of type *s-la* joints in {*c-ll*, *s-aa*} case.

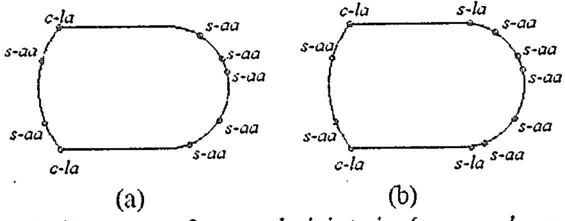


Fig. 8 Recovery of type *s-la* joints in {*s-aa*, *c-la*, *s-aa*} case.

3.2 Merging/splitting of circular arcs

In addition to inserting a new type *s-la* joint between a line segment and a circular arc, one must also detect consecutive circular arcs for possible merging or splitting. To determine a circular arc, it is necessary to estimate both the center and the radius of the arc. The least-mean-square (LMS) approach is popular and simple in which an improved Landau's parameter-estimation algorithm [5], employed in this paper, for fast and accurate estimation of the parameters is developed. Also, it suffices to consider only type *s-la*, *c-la*, *s-aa* and *c-aa* joints.

In the arc-merging/splitting procedure, define the measure of significance of arc, denoted by η, as the ratio of the maximum deviation divided by the associated arc length. Then, using Algorithm 3-1 two smaller arcs are merged into a larger one if the larger arc has smaller measure η whereas the arc is split into two by inserting a new type *s-aa* joint if the measure η of the new arc is larger than the one of the original arc. The merging/splitting procedure is repeated until the entire curve can not be modified.

Algorithm 3-1:

Let the *j*th set of the arc segments be $\{\tilde{Q}_j, \tilde{P}(1), \tilde{P}(2), \ldots, \tilde{P}(N), \tilde{Q}_{j+1}\}$, where $\tilde{Q}_j$ and $\tilde{Q}_{j+1}$ may be type *c-la*, *s-la* or *c-aa* joints and $\tilde{P}(i)$ is the *i*th *s-aa* joint in set *j*.

Step 1: Calculate the center, radius and η for the arc $\{\tilde{Q}_j, \tilde{P}(1)\}$. Let $\eta_{old} = \eta$ and i=2.

Step 2: Extend the region-of-support to $\tilde{P}(i)$. That is, calculate the center, radius and η_{new} for the arc on the interval $\{\tilde{Q}_j, \tilde{P}(1), \ldots, \tilde{P}(i)\}$.

Step 3: If $\eta_{new} \le \eta_{old}$ then merge $\{\tilde{Q}_j, \tilde{P}(1), \ldots, \tilde{P}(i)\}$ into an arc and let $\eta_{old} = \eta_{new}$, i=i+1, and then repeat from Step 2. Otherwise, let $P_1^* = \tilde{P}(i-1)$, $P_2^* = \tilde{P}(i)$.

Step 4: Split the segment $\{P_1^*, P_2^*\}$ by inserting a new joint $Q^* = (P_1^* + P_2^*)/2$ into the interval. Calculate the center, radius and η_{new} on the region-of-support $\{\tilde{Q}_j, \tilde{P}(1), \ldots, \tilde{P}(i-1), Q^*\}$.

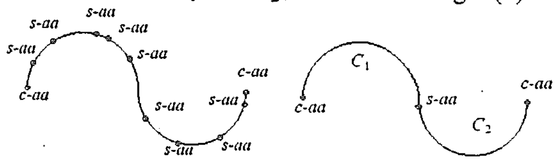
Step 5: If $\left|P_1^* - P_2^*\right| < 3$, then goto Step 7.

Step 6: If $\eta_{new} \le \eta_{old}$, then let $\eta_{old} = \eta_{new}$, $P_1^* = Q^*$; otherwise, let $P_2^* = Q^*$. Repeat from Step 4.

Step 7: Assign point Q^* as a new type *aa* joint and remove the type *aa* joints in the interval $\{\tilde{Q}_j, Q^*\}$. Update the remaining set of type *aa* joints between Q^* and $\tilde{Q}_{j+1}$ and let $Q^* = \tilde{Q}_j$ then repeat from Step 1 until there is no more type *aa* joint in the interval $\{\tilde{Q}_j, \tilde{Q}_{j+1}\}$, and then stop.

In Algorithm 3-1, steps 2 and 3 are for merging whereas steps 4 and 6 are used to split an arc into appropriate pairs of arcs. As a whole, Algorithm 3-1 accomplishes joint tuning by doing both merging and splitting.

Consider the sequence of type *aa* joints in Fig 9(a). Using Algorithm 3-1, the arcs connecting the type *aa* joints are merged and a new type *s-aa* joint is generated that connects arcs C_1 and C_2, as shown in Fig. 9(b).

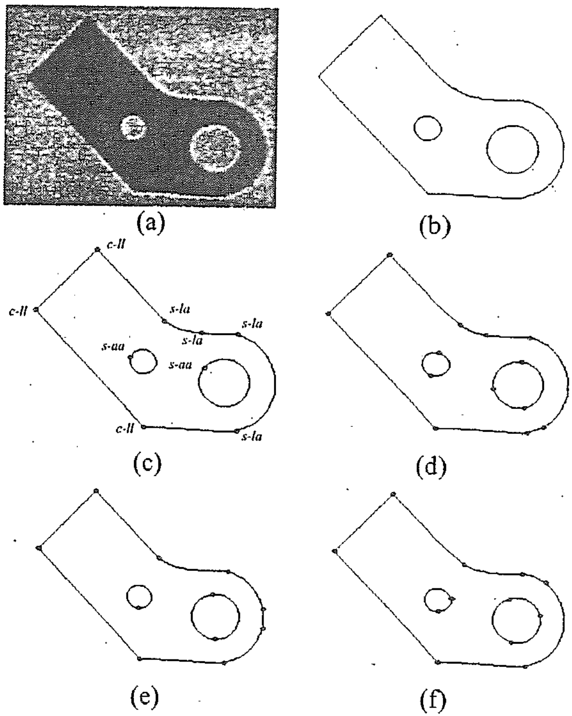


(a) (b)

Fig. 9 Merging/splitting of circular arcs.

4. Experimental results

The experiments have been done on a PC486-33 computer. The image of a cutter is shown in Fig. 10(a), the detected edges [14] are given in Fig. 10(b), and the final results using the proposed scheme, the 'seven-criteria' approach, the LAD and the DFA are shown in Figs. 10(c)~10(f), respectively. It can be seen from Fig. 10(c) that, in the result with the proposed scheme, there are five line segments, two arcs and two circles with each circle represented by a single arc segment, in contrast to the results of the 'seven-criteria', the LAD and the DFA approaches where more than one arc segments are derived, as shown in Figs. 10(d)~10(f).



(a) (b) (c) (d) (e) (f)

Fig. 10 Segmentation of a cutter. (a) Original image; (b) The edge; (c)~(f) for the results using the proposed method, 'seven criteria' approach, LAD and DFA, respectively.

In general, the higher the maximum deviation is, the worse the data fidelity will be. On the other hand, the higher the compression ratio is, the better the performance of the multiprimitive segmentation is. Hence, it is eligible to define the measure of performance of curve segmentation scheme as [3]

$$\mu = \frac{\sum(\text{maximum deviation/segment length})}{\text{compression ratio}} \qquad (5)$$

where compression ratio is defined as the ratio of the total points divided by the parameters of primitives. The execution time, the measure μ and the number of segments are shown in Table 1, indicating that the proposed method is better.

Table 1. The performance for Figs. 10(c)~10(f).

Total points : 896				
Method	Time (sec)	μ	Segment no.	Threshold-free
The proposed method	0.32	7.42×10^{-3}	9	Yes
'seven-criteria'	0.51	1.99×10^{-2}	13	No
LAD	0.67	1.01×10^{-2}	11	Yes
DFA	0.56	1.83×10^{-2}	12	No

5. Conclusions

A two-level approach for both breakpoint classification and joint tuning is proposed for the multiprimitive segmentation of planar curves. In level 1, AKC functions are employed to divide breakpoints into corners and smooth joints. In level 2, PHFs are used to further determine the segments on both sides of the breakpoints hence the types of the breakpoints. Also, with the suggested joint tuning procedure, recovery of line segments and merging/splitting of circular arcs are accomplished. Since joint-type provides useful information for an arbitrary curve, they are included in multiprimitive segmentation in addition to line segments and circular arcs. Besides, the proposed scheme not only can detect and remove the errors made in breakpoint-detection algorithms but also is robust to quantization errors.

In the multiprimitive segmentation using the proposed scheme, since no threshold is required, the result is not influenced by the selection of thresholds that frustrates the 'seven-criteria' approach. Besides, the drawback of LAD, i.e., the expensive computation for establishing complex search trees and the problem that the supposedly same arc may not be merged if they are not in the same branch, are also overcome. Further, the drawback of DFA, i.e., segmentation may not be the same if starting from different points, will not happen using the proposed method. As can be seen from the experimental results, the proposed scheme is fast, accurate, simple, and threshold-free.

References

[1] T. Pavlidis and S.L. Horowitz. Segmentation of plane curves. *IEEE Trans. Comput.*, **23**, 860-870, 1974.

[2] C.H. Teh and R.T. Chin. On the detection of dominant points on digital curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, **11**, 859-872, 1989.

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

[4] F.T. Farago. *Handbook of Dimensional Measurement*. 2nd Ed. Industrial press Inc., New York, 1982.

[5] Q.M. Wu and M.G. Rodd. Boundary segmentation and parameter estimation for industrial inspection. *IEE Proc. E, Computers and Digital Techniques*, **137**, 319-327, 1990.

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

[7] P.L. Rosin and G.A.W. West. Segmentation of edges into lines and arcs. *Image Vision Comput.*, **7**, 109-114, 1989.

[8] G.A.W. West and P.L. Rosin. Techniques for segment image curves into meaningful descriptions. *Pattern Recognition*, **24**, 643-652, 1991.

[9] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artif. Intell.*, **31**, 355-395, 1987.

[10] C. Ichoku, B. Deffontanies and J. Chorowicz. Segmentation of digital plane curves: a dynamic focusing approach. *Pattern Recognition Lett.*, **17**, 741-750, 1996.

[11] H. Asada and M. Brady. The curvature primal sketch. *IEEE Trans. Pattern Anal. Mach. Intell.*, **8**, 2-14, 1986.

[12] H.T. Sheu and H.Z. Yang. Open curve segmentation via a two-phase scheme. *Pattern Recognition*, **26**, 1839-1844, 1993.

[13] A. Rosenfeld and E. Johnston. Angle detection on digital curves. *IEEE Trans. Comput.*, **22**, 875-878, 1973.

[14] S.C. Zhu and A. Yuille. Region competition: unifying snakes, region growing and Bayes/MDL for multiband image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, **18**, 884-900, 1996.