Journal of Southeast University (English Edition) Vol. 20 No. 2 June 2004 ISSN 1003—7985

Contour extraction and curvature calculation for fragment reassembly

Zhu Yanjuan Zhang Liyan Zhou Laishui

(Research Center of CAD/CAM Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: The Canny edge detector inevitably misses some important and obvious edges during contour extraction, which causes gaps in the contour. We propose a geometric method to locate, measure and fill the gaps precisely. With the complete contour information, we present a convolution approach, which utilizes an appropriate linear interpolation to resample the contour to calculate pointwise curvature. This approach distributes discrete points within a convolution window uniformly. It ensures a one-to-one correspondence between every point and its weight, thus the accuracy is guaranteed under this condition. A related parameter selection is also suggested. Experimental results show that the proposed methods are robust and accurate.

Key words: contour extraction; curvature calculation; convolution integral; discrete points

The basic task in computer-aided reconstruction of archaeological finds or other similar objects is to reassemble broken objects automatically. Extracting contours and calculating curvatures are two essential aspects in the assembly process. Since the 2-D broken piece is globally and locally represented by its closed contour, and the 3-D fragments can be described completely by its contour or ridge together with its surface facet. Meanwhile, curvature can uniquely define the 2-D contour and, in connection with torsion, it can also represent the 3-D contour exactly. Curvature and torsion are efficient, highly descriptive features of geometric objects. For a twice-differentiable curve, the curvature at a point can be expressed in terms of the first and the second derivatives of the curve at that point. No such simple definition exists for a digital curve. This paper deals with extracting contours from scanned images and calculating pointwise curvatures of the extracted contour. The main effort prepares for reassembly of the 2-D fragments, which can be used in mural paintings and approximately in flat potsherds.

For contour extraction, previous work of Refs. [1, 2] was to track an intensity level curve between the background and the fracture surfaces. This algorithm is frequently confused in some practical applications because of its ideal assumption, namely, the foreground color $\delta_{\max}$ is a constant. Ref. [3] ignored the process of data acquisition and just assumes a binary representation of each piece is available.

Curvature calculation for discrete contours has been done mainly on two aspects. Medioni, et al. [4,5] proposed to use a parametric cubic B-spline for representing a boundary curve and then calculate the corresponding curvature. Vialard[6] constructed the model of Euclidean paths, which was a semi-continuous representation of the underlying real discrete curve. The correlative curvature is calculated according to this model. These methods are based on constructing a continuous or semi-continuous model to represent the discrete contour. Other categories of methods tried to use some simple methods to approximate digital curvature, e.g. Wolfson[7] adopted the arc-length versus total turning angle graph, namely $\theta(s)$, to describe a curve. Chetverikov[8] implemented a technique to show high curvature points in planar curves via triangles with specified opening angles. Their methods show critical points of the contour without specific curvature values. In addition, these methods are typically sensitive to noise.

Based on the Canny edge detector we developed a geometric method to obtain the contour of a fragment automatically and accurately, and then presented a convolution approach to the problem of curvature calculation for discrete points.

The Canny approach is one of the most widely used in edge detection, but it inevitably misses some important and obvious edges during boundary detection[9]. The missed edges cause gaps in the object contour. They make feature computation about curvature and perimeter almost impossible, to say nothing of contour matching or fragment reassembly. In this paper, we propose a method to find the

Received 2003-11-11.

Foundation items: The National Natural Science Foundation of China (No. 60273097), the Teaching and Research Award Program for Outstanding Young Teachers in Higher Education Institutions of MOE, P. R. C., and Scientific Foundation of Nanjing University of Aeronautics and Astronautics (No. S0272-054).

Biographies: Zhu Yanjuan (1970—), female, graduate; Zhang Liyan (corresponding author), female, doctor, professor, zhangly@nuaa.edu.cn.

locations of missed edges and fill in the gaps according to the orientation of the original contour. After being processed by this method, the discrete contour is convolved with a Gaussian kernel, which makes it feasible to remove the noise of the contour and calculate the curvature simultaneously.

The accuracy of curvature computation is a key issue in the contour description. An appropriate linear interpolation is developed to resample the discrete contour while ensuring the precision of the curvature computation. In addition, parameter selection in the curvature calculation is also proposed. The significance of all these works is that it alleviates the problem of fragment reassembly and makes it feasible.

1 Contour Extraction

The Canny operator can detect an overwhelming majority of the boundaries. However, if the gradient magnitude of edges is the local maximum in the image, while it is not in the gradient direction, it will not be picked up[10]. So these edges are missed, which causes gaps in continuous edges (see Figs. 1 (a) and (b)). A geometric method is presented to solve this problem. The first step is to assign specific coordinates for every detected edge point. The algorithm is as follows:

Step 1 Select an arbitrary point P_i on the contour as starting point.

Step 2 Suppose P_i is a center point, find the point P_{i+1} with the smallest distance to P_i. The coordinates of P_{i+1} can be calculated according to the distance, and then labeled P_{i+1} checked.

Step 3 Select P_{i+1} as the current center point, find the closest point P_{i+2} in its neighborhood. Calculate the coordinates of P_{i+2} and label them P_{i+2}.

Step 4 Repeat step 2 and step 3 until all the edge points have been labeled.

Step 5 Store all of the edge points in an array.

Then we check each pair of adjacent points $P_i(x_i, y_i)$ and $P_{i+1}(x_{i+1}, y_{i+1})$ in the array. If the relationship between the two points accords with any of the following three conditions, we will think there are missed edges. In other words, if the distance between two adjacent points is more than 1 or 2, it means some points among them are not detected, which results in gaps.

- $x_i = x_{i+1}$, $y_i \neq y_{i+1}$ and $|y_{i+1} - y_i| > 1$;
- $y_i = y_{i+1}$, $x_i \neq x_{i+1}$ and $|x_{i+1} - x_i| > 1$;
- $x_i \neq x_{i+1}$, $y_i \neq y_{i+1}$ and $|x_{i+1} - x_i| + |y_{i+1} - y_i| > 2$.

Once the gap is found, it should be filled according to the orientation of the gap. Being processed by this method, the boundary information is clear and complete (see Figs. 2 (a) and (b)). Then we utilize a convolution integral to compute curvature.

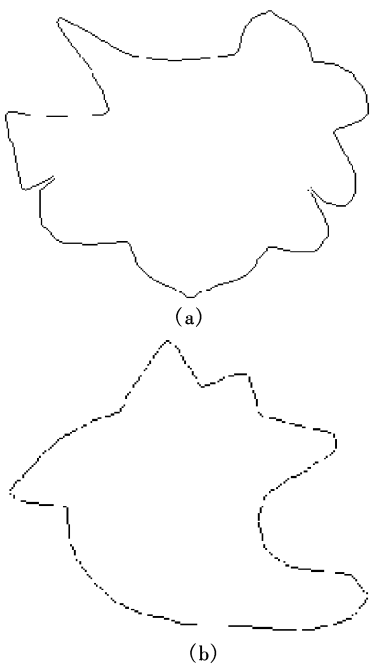

Fig. 1 Two contours extracted from their corresponding digital images by Canny edge detector, some obvious edges are missed. (a) Contour Ⅰ; (b) Contour Ⅱ

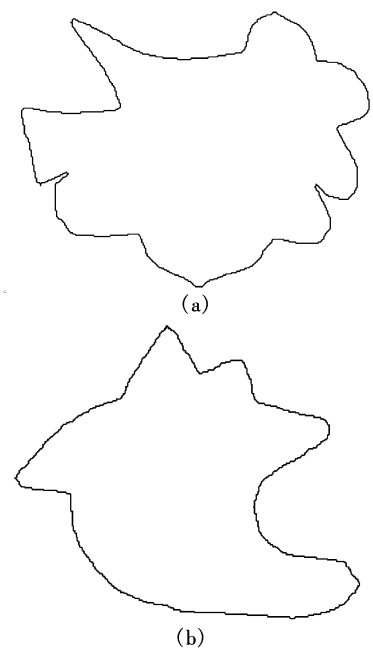

Fig. 2 Two extracted contours processed by the geometric method, all gaps are filled exactly. (a) Contour Ⅰ; (b) Contour Ⅱ

2 Convolution Integral with Gaussian

Given an arbitrary function $f_1(t)$ and another function $f_2(t)$, the convolution integral of $f_1(t)$ and $f_2(t)$ is as follows[11]:

$$g(t) = f_1(t) * f_2(t) = \int_{-\infty}^{+\infty} f_1(\tau) f_2(t-\tau)\,\mathrm{d}\tau$$

The integral range $-\infty$ to $+\infty$ is suitable for any generic function. If the function is definite, the range is decided by the function's domain. A significant property of convolution lies in that the mapping will have the same smoothness, if the kernel has some smoothness. For this reason, we select Gaussian distribution function Eq. (1) as the kernel and convolve with the discrete contour, which may be considered as an arc-length domain signal.

$$G(u, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{u^2}{2\sigma^2}} \tag{1}$$

Accordingly, the convolution of two functions is defined as

$$F(u, \sigma) = \int_{-\infty}^{+\infty} f(v) \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{(u-v)^2}{2\sigma^2}} \mathrm{d}v$$

Fig.3 illustrates the convolution process further.

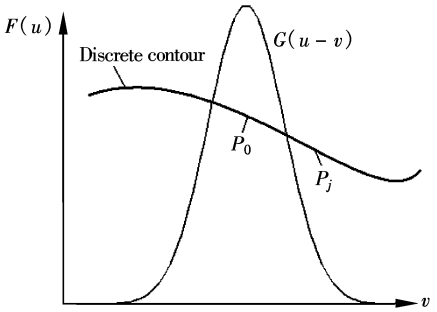

Fig.3 Discrete point contour and convolution of Gaussian kernel

It is shown that, for a set of discrete points, Gaussian distribution assigns relevant weight factors without exception, whereas the weights are quite different. Suppose P_0 is a center point, the corresponding weight of P_0 is the largest. The weights of its neighbor points decrease with the increase of distance between the selected point P_i and the center point P_0. As the distance becomes larger, the weight will gradually reduce to zero. The meaning is obvious, for the point is so far away from the center, that it contributes no effect to the local character of the center point. The convolution process also indicates that its essence is weighted averaging. Therefore, the convolution can be expressed as

$$F(u, \sigma) = \sum_{-\infty}^{+\infty} f(v) \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{(u-v)^2}{2\sigma^2}} \Delta v \tag{2}$$

where $\frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{(u-v)^2}{2\sigma^2}}$ is the weight factor and Δv is the step. If the convolution is done in different scales parameter σ, which makes it feasible to smooth the same outline in different scales, then an evolved version of the curve is formed naturally.

Based on the property of the Gaussian kernel, we use a truncation parameter m to specify the size of the smoothing neighborhood. For the purpose of minimizing the truncation error, in our work, m is set according to Eq.(3) which is larger than 3σ.

$$m = \sqrt{\sigma^2 \ln\left(\frac{2\pi}{\sigma^2 \varepsilon^2}\right)} \tag{3}$$

where ε is the given tolerance. Hence,

$$F(u, \sigma) = \sum_{u-m}^{u+m} f(v) \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{(u-v)^2}{2\sigma^2}} \Delta v$$
$$(u-v) \in [-m, m]$$

We finally introduce a property of the convolution that will be used later.

$$F'(t) = f_1'(t) * f_2(t) = f_1(t) * f_2'(t) \tag{4}$$
$$F''(t) = f_1''(t) * f_2(t) = f_1(t) * f_2''(t) \tag{5}$$

The proof is very simple, so it is not addressed **here.**

3 Curvature Calculation

A planar curve is the track of point when it moves with time. Without loss of generality, the curve can be represented by a vector equation as

$$\boldsymbol{r} = \boldsymbol{r}(t) = \{x(t), y(t)\} \qquad t \in (\alpha, \beta)$$

The curvature of a curve at any point is defined as the rate of change in tangent vector direction with respect to arc length s. The formulization is[12]

$$k = \frac{x'(t)y''(t) - x''(t)y'(t)}{[x'^2(t) + y'^2(t)]^{\frac{3}{2}}}$$

For a discrete contour $\boldsymbol{r} = \{x(u), y(u)\}$, where u is an arbitrary parameter, let $x(u)$, $y(u)$ convolve with the Gaussian kernel respectively.

$$\boldsymbol{r}_\sigma = \{X(u, \sigma), Y(u, \sigma)\}$$

where $X(u, \sigma) = x(u) * G(u, \sigma)$, $Y(u, \sigma) = y(u) * G(u, \sigma)$.

Then curvature of the curve $\boldsymbol{r}_\sigma$ is[13]

$$K(u, \sigma) = \frac{X_u(u, \sigma) Y_{uu}(u, \sigma) - X_{uu}(u, \sigma) Y_u(u, \sigma)}{[X_u(u, \sigma)^2 + Y_u(u, \sigma)^2]^{\frac{3}{2}}}$$

Utilizing the convolution property in Eq.(4) or (5), we can infer that

$$X_u(u, \sigma) = \frac{\partial}{\partial u}(x(u) * G(u, \sigma)) = x(u) * G_u(u, \sigma) = \sum_{u-m}^{u+m} x(v) \left[\frac{-(u-v)}{\sigma^3\sqrt{2\pi}}\right] \mathrm{e}^{-\frac{(u-v)^2}{2\sigma^2}} \Delta v$$

$$X_{uu}(u, \sigma) = \frac{\partial^2}{\partial u^2}(x(u) * G(u, \sigma)) = x(u) * G_{uu}(u, \sigma) = \sum_{u-m}^{u+m} x(v) \left[\frac{(u-v)^2 - \sigma^2}{\sigma^5\sqrt{2\pi}}\right] \mathrm{e}^{-\frac{(u-v)^2}{2\sigma^2}} \Delta v$$

$$Y_u(u, \sigma) = \frac{\partial}{\partial u}(y(u) * G(u, \sigma)) = y(u) * G_u(u, \sigma) = \sum_{u-m}^{u+m} y(v) \left[\frac{-(u-v)}{\sigma^3\sqrt{2\pi}}\right] \mathrm{e}^{-\frac{(u-v)^2}{2\sigma^2}} \Delta v$$

$$Y_{uu}(u,\sigma)=\frac{\partial^2}{\partial u^2}(y(u)*G(u,\sigma))=y(u)*$$

$$G_{uu}(u,\sigma)=\sum_{u-m}^{u+m}y(v)\left[\frac{(u-v)^2-\sigma^2}{\sigma^5\sqrt{2\pi}}\right]e^{-\frac{(u-v)^2}{2\sigma^2}}\Delta v$$

Considering $x(v)$ and $y(v)$, we put forward a linear interpolation method to obtain the corresponding value.

Given two points P_i and P_{i+1}, the inserted point P_t is decided by Eq.(6).

$$P_t=\left(1-\frac{s-\sum_{i=0}^{j}D[i]}{\sum_{i=0}^{j+1}D[i]-\sum_{i=0}^{j}D[i]}\right)P_i+\frac{s-\sum_{i=0}^{j}D[i]}{\sum_{i=0}^{j+1}D[i]-\sum_{i=0}^{j}D[i]}P_{i+1} \quad (6)$$

where s is the interval between the inserted point and the center point, j is the function of s. The value of $D[i]$ is the distance between two adjacent points, such as P_i and P_{i+1}. In other words, the linear interpolation method is a process to resample the outline, which distributes discrete points within convolution window uniformly. With the variance of the step Δv in Eq.(2), the number of resampled points is changed accordingly. This method also ensures every point and its weight is one-to-one correspondence. The precision of curvature calculation is guaranteed under this condition.

4 Experimental Results

Depending on the pointwise curvature of a curve and its arc length, we can draw a curvature graph about the related contour. In order to evaluate the quality of the curvature estimation, we utilize the curvature graph to detect corners; at the same time we pay attention to the effects of different parameter selections. Firstly we give the results when the truncation parameter $m=3\sigma$ and the step $\Delta v=2$ (see Figs.4 and 5). It is obvious that false corners are detected and not all true corners are detected. This result means that curvature under this condition is not good enough to be used.

Then we adjust the size of the smoothing neighborhood and set m according to Eq.(3) (see Figs. 6 and 7).

Results are improved from those in the last experiment, but there are still overlapping corners. We decrease the step Δv to resample the contour again (see Figs.8 and 9).

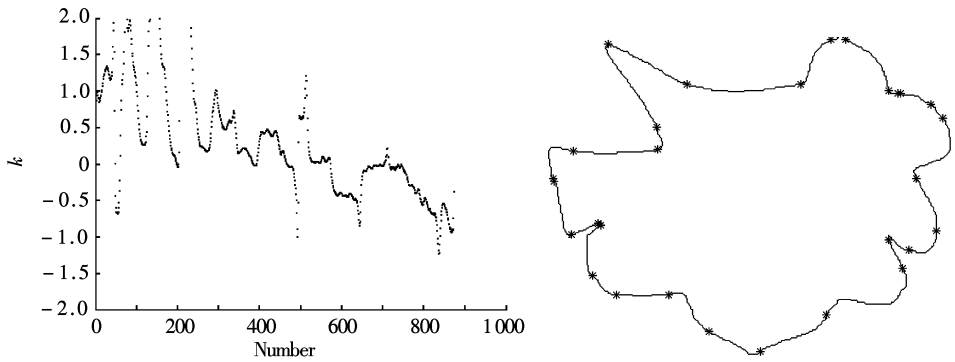

Fig.4 Contour Ⅰ curvature graph and its corner detection ($\sigma=4$, $m=12$, $\Delta v=2$)

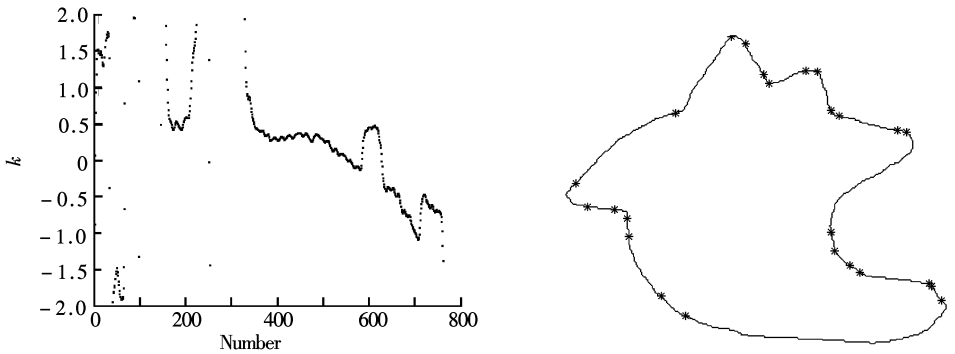

Fig.5 Contour Ⅱ curvature graph and its corner detection ($\sigma=4$, $m=12$, $\Delta v=2$)

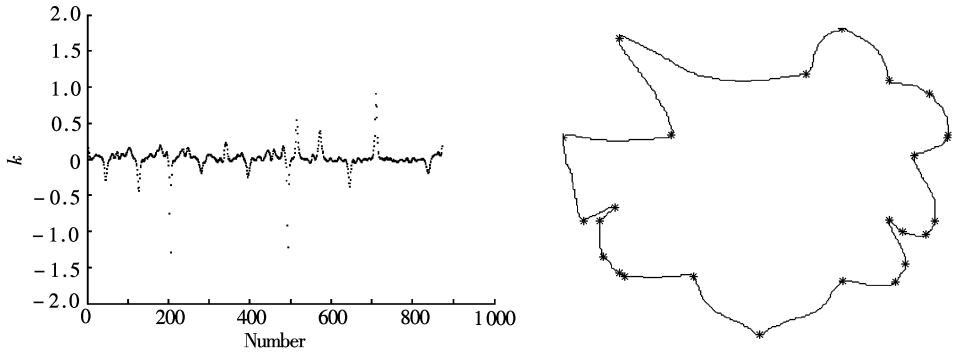

Fig.6 Contour Ⅰ curvature graph and its corner detection ($\sigma=4$, $m=39.995$, $\Delta v=2$)

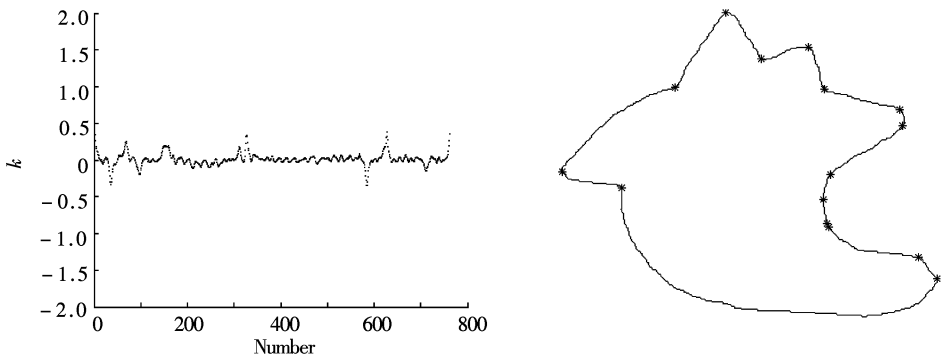

Fig.7 Contour Ⅱ curvature graph and its corner detection ($\sigma=4$, $m=39.995$, $\Delta v=2$)

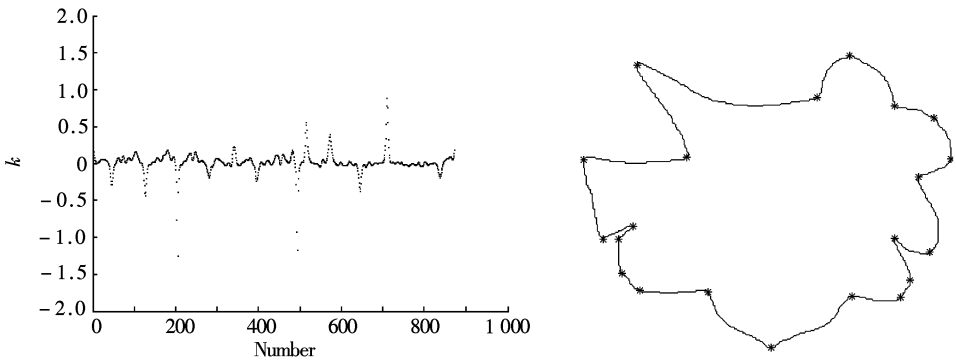

Fig.8 Contour Ⅰ curvature graph and its corner detection ($\sigma=4$, $m=39.995$, $\Delta v=0.8$)

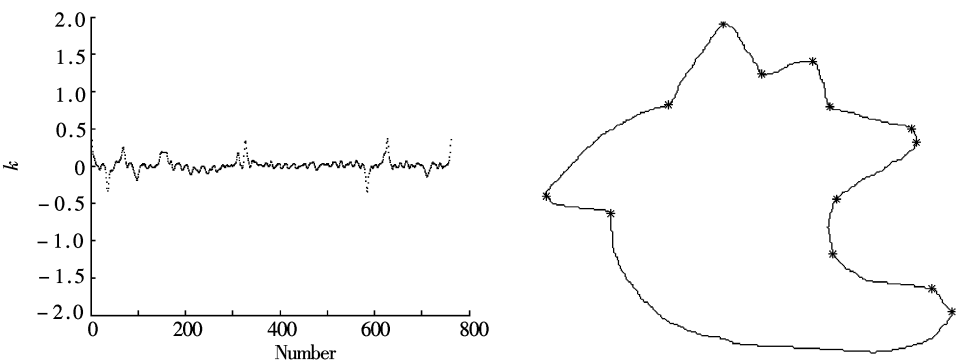

Fig.9 Contour Ⅱ curvature graph and its corner detection ($\sigma=4$, $m=39.995$, $\Delta v=0.8$)

We find that there are no overlapped corners in Fig.8 and Fig.9 and the localization of corner point is also perfect. It illuminates that the parameter selection is good and meanwhile the curvature calculation is precise. Accordingly, the description of contour is clear and compact.

Since the value of parameter σ is changeable, the same contour can be described at different scales. As σ varies from small to large, we can depict the contour from fine to coarse levels. An optimal value

of σ should be able to filter noise from the contour, while at the same time preserve small-scale details. In our experiments, when $\sigma = 4$, the corresponding result is better than all the others.

5 Conclusion

Fragment reassembly depends on the availability of good edge data. The approach of Canny detects the majority information about fragment contour, but it misses some obvious and critical edges which may be potential corners. A geometric method is developed here to locate the missed edges, measure the gap's size and orientation, and then fill the gaps precisely. With the complete boundary information, a method of discrete curvature calculation is presented. This method removes noise from the contour, while at the same time calculates pointwise curvature. It is translationally and rotationally invariant as well as scale-invariant. In addition, an appropriate linear interpolation is proposed to resample the contour and make sure of the result's precision. Related parameter selection is also suggested. The robustness of this approach is demonstrated through examples. Further effort can be made to utilize the curvature information to reassemble fragments.

References

[1] Leitão H C G, Stolfi J. A multi-scale method for the reassembly of fragmented objects [A]. In: *Proc British Machine Vision Conference* [C]. **Bristol, 2000, 2**: 705 - 714.

[2] Leitão H C G, Stolfi J. A multi-scale method for the reassembly of two-dimensional fragmented objects [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2002, 24**(9): 1239 - 1251.

[3] Kong Weixin, Kimia B B. On solving 2-D and 3-D puzzles using curve matching [A]. In: *Proc of the IEEE Conference on Computer Vision and Pattern Recognition* [C]. Hawaii, 2001. 583 - 590.

[4] Medioni G, Yasumoto Y. Corner detection and curve representation using cubic B-splines [J]. *Computer Vision, Graphics and Image Processing*, **1987, 39**(3): 267 - 278.

[5] Liu H C, Srinath M D. Corner detection from chaincode [J]. *Pattern Recognition*, **1990, 23**(1): 51 - 68.

[6] Vialard A. Geometrical parameters extraction from discrete paths [A]. In: Miguet S, Montanvert A, Ub′eda S, eds. 6*th International Conference on Discrete Geometry for Computer Imagery* [C]. **Lyon, 1996, 1176**: 24 - 35.

[7] Wolfson H J. On curve matching [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **1990, 12**(5): 483 - 489.

[8] Chetverikov D, SzabóZ. A simple and efficient algorithm for detection of high curvature points in planar curves [A]. In: *Proceedings of the* 23*rd Workshop of the Austrian Pattern Recognition Group* [C]. Steyr, 1999. 175 - 184.

[9] Boyer K L, Sarkar S. Comments on the localization performance measure and optimal edge detection [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **1994, 16**(1): 106 - 108.

[10] Ding Lijun, Goshtasby A. On the Canny edge detector [J]. *Pattern Recognition*, **2001, 34**(3): 721 - 725.

[11] Pan Wenjie. *Fourier analysis and application* [M]. Beijing: Peking University Press, 2000. (in Chinese)

[12] Shi Fazhong. *Computer aided geometric design & non-uniform rational B-spline* [M]. Beijing: Beijing University of Aeronautics and Astronautics Press, 1994. 17 - 25. (in Chinese)

[13] Mokhtarian F, Mackworth A K. A theory of multiscale, curvature-based shape representation for planar curves [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **1992, 14**(8): 789 - 805.

碎片拼合的轮廓提取和曲率计算

朱延娟　张丽艳　周来水

(南京航空航天大学 CAD/CAM 工程研究中心, 南京 210016)

摘要: Canny 算子在对物体进行轮廓提取时,会不可避免地漏检一些明显的边界,导致轮廓的不连续. 提出了一种几何方法来定位、度量轮廓上的间断点, 然后将其准确地填充起来. 在获得轮廓的完整信息后, 文中提出采用卷积积分的方法, 通过线性插值对轮廓进行重采样来计算各离散点曲率. 该方法保证了卷积窗口内离散点分布得均匀、一致, 并且使得每一离散点与其权重都满足一一对应关系, 从而保证了曲率计算的精确性. 同时给出了相关参数的选择方法. 实验结果表明,算法是准确和稳定可靠的.

关键词: 轮廓提取; 曲率计算; 卷积积分; 离散点

中图分类号: TP391