行政院國家科學委員會專題研究計畫成果報告

數位圖書館及博物館之自動化資訊處理(1/3)－子計畫二：

中國碑帖文字之擷取、辨識與儲存

Shape Description of Chinese Characters in Calligraphy Documents

計畫編號：NSC 89-2213-E-009-091
執行期限：88 年 8 月 1 日至 89 年 7 月 31 日
主持人：李錫堅　　國立交通大學資訊工程系

一、中文摘要

這個系統包括兩個主要的模組: 字形輪廓切割之擷取與字形輪廓切割之描述。在字形輪廓切割之擷取部分，我們先將字形輪廓變平滑以防止字形輪廓有凹凸不平的現象並去除字中的白色小雜點，接著找出字形輪廓，在輪廓上找出曲率變化較大的點當作輪廓線的斷點，這些斷點將每一個輪廓線切成數個輪廓線段。在字形輪廓切割之描述部分，首先我們先決定每一個輪廓線段的描述形態，看是用直線描述或用貝茲曲線來描述，若是使用貝茲曲線來描述，首先估計每一個曲線控制點的起始位置，接著定義一個判斷函數，看曲線與輪廓線段的密合度，若由此判斷函數得知曲線與輪廓線段的差異大，則移動曲線的控制點讓曲線更接近原輪廓線段，詳細的說，我們先找出通過控制點且與輪廓線段兩端點相垂直的直線，還有此直線與輪廓線段和曲線的各自交點，利用交點的距離來移動控制點往左往右或增加斷點，藉此讓曲線更接近輪廓線段，當每一個輪廓線段和曲線都很接近的時候，這表示我們可以用這些曲線來描述原來的字形輪廓而且可以描述得很好。在實驗中，對於一個含有 4500 個輪廓線段的 A4 影像，曲線描述的處理時間約為 5 秒，成功描述輪廓線段的平均移動次數為 1.32 次。

Abstract

The purpose of this proposed system is to develop an automatic image vectorization tool. It consists of two main phases: contour segment extraction and contour segment description. In contour segment extraction phase, we first perform image smoothing to avoid the appearances of rough contours and remove white pixels inside characters. Then we find contours of the characters using contour tracing. In other words, we remove the inner points of the characters. Therefore, we must fill inner points of the characters latter. After contours are obtained, we detect local high curvature points on contours to be corner points. The corner points divide each contour into several contour segments. That is, a contour segment is defined as the contour points from a corner point to the next one. In the following procedures, we use a cubic Bezier curve or a straight line to describe a contour segment. In contour segment description phase, we first determine whether a contour segment is described by a straight line or a Bezier curve. If it is described by a straight line, we connect its two endpoints. Otherwise we initialize and adjust control points according to a cost function which is designed to evaluate the curve fitness. If all of the Bezier curves for the contour segments have the least cost, we have the best descriptions.

Keywords: contour segment extraction,

contour segment description,
Bezier curves.

1. Introduction

Describing the shape of an image region is a very important processing underlying various digital image processing, especially, pattern recognition and computer vision. There are some methods proposed for shape description in the past. Cinque, Levialdi and Malizia [1] presented a method for shape description consisting of an approximation of a shape by a variable number of Bezier curve segments. In their method, they can control the accuracy of the Bezier approximation by a parameter thus controlling the complexity and resolution of the approximation process. The technique they proposed is suited to a variety of shape-based image retrieval applications and matching processes.

Cinque and Lombardi [2] proposed a multiresolution approach that uses a diffusion process to describe the shape of a 2D object. Their method relies on the concept of a structural coding of an object at varying levels of resolution. A tree structure represents the evolution of the contour at increasing levels of detail, where each tree node represents a contour segment via a set of attributes to provide a richer description of the image shape.

Chang and Yan [3] presented a curve-fitting algorithm for vectorizing hand-drawn key frames in a computer-aided cartooning system. The vectorization in their system can be divided into three processes; skeletonization, tracing and approximation. They use piecewise cubic Bezier curves to approximate the given line drawings.

Kimoto and Yasuda [4] proposed a shape description scheme by decomposition of regions using ellipsoids as primitive elements are discussed from the viewpoint of both amount of data and degree of approximation.Wall and Danielsson presented a fast sequential method for polygonal approximation of digitized curves. It uses a scan-along technique where the approximation depends on the area deviation for each line segment. The algorithm outputs a new line segment when the area deviation per length unit of the current segment exceeds a prespecified value.

2. Contour Segment Extraction

2.1 Image Smoothing

To reduce the rough and uneven situations on the contours of characters, we check eight neighbors for each pixel in source images (See Fig.1). In the following cases, the center black pixel of the window is removed.

Z1	*Z2*	*Z3*
Z4	õ	*Z5*
Z6	*Z7*	*Z8*

Fig. 1. Labels of a black pixel's neighbors.

Case1: All of eight neighbors are white. It is a isolated point.

Case2: Only one of eight neighbors is black. It is an end point.

Case3: One of the following conditions is satisfied. (1) Only ,Z_7, Z_6 and Z_5 are black , (2) Only ,Z_5, Z_3 and Z_0 are black, (3) Only ,Z_9, Z_1 and Z_2 are black, (4) Only ,Z_2, Z_4 and Z_7 are black.

Case4: One of the following conditions is satisfied. (1) Only ,Z_7, Z_6 and Z_5 are black ,

(2) Only ,Z_5, Z_3 and Z_0 are black, (3) Only ,Z_9, Z_1 and Z_2 are black, (4) Only ,Z_2, Z_4 and Z_7 are black.

2.2 Contour tracing

The contours of the characters are the set of points of the characters that are adjacent to points of the complement of the characters. The coordinate locations of points on contours are very important information in pattern recognition and other applications. They can be obtained by using a contour tracing algorithm.

There are two kinds of contours we used in contour tracing: "black contour" and "white contour". The black contour represents the outer contour of a black region and the white contour represents the outer contour of an inner white region. An illustration is given in Fig.2.

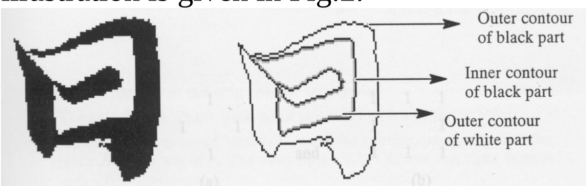

Fig.2 The illustration of Contours

2.3 Corner detection

The purpose of corner detection is to detect some potential feature points that can sufficiently describe the shape of an object. In our system, the corner points are defined as those points with local maximum curvature values on the contours of calligraphy characters. After corner points are detected, a contour will be divided into several contour segments. We define a contour segment as the contour points from a corner point to the next one. We take the branch points on the contours as corner points first to avoid the fitting error in latter procedures.

There are two main steps in corner detection method: (1) to estimate the curvature of each point on the contour by a bending value, and (2) to locate those points with local maximum bending values as corner points.

2.3.1 The bending value

After contour tracing, an order sequence of points , where n is the total number of points on the contour, is obtained. Two points, and , can be considered as the forward and backward neighbors of point , respectively. Furthermore, the two points, and , are the kth forward and backward neighbors point , respectively.

When the included angle between the forward and backward vectors is small, the sum vector of them is large. Otherwise, the sum vector is small. The smaller the included angle is, the higher the curvature is. An example is shown in Figure 2.3. Depending on this concept, we define the bending value as the length of the sum vector to estimate the degree of the curvature. The larger the bending value is, the higher the degree of the curvature is. Since the length of the sum vector of and depends on the length of and , the two vectors and should be normalized first. We use the unit vectors of and here. On the basis of the same length of and for each point , we can take the length of the sum vector as the bending value to estimate the degree of the curvature.

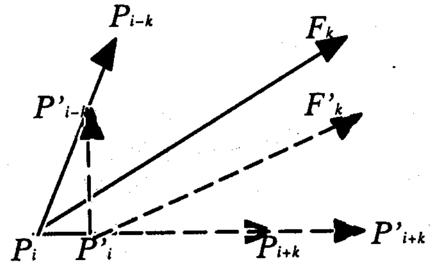

Fig.3 Relations between the included angle and the sum vector.

The sum vector $\vec{P_{ifk}}$ is denoted as follows.

$\vec{P_{ifk}} = \vec{u_b} + \vec{u_f}$, where $\vec{u_b}$ is the unit vector of $\vec{P_iP_{i+k}}$ and $\vec{u_f}$ is the unit vector of $\vec{P_iP_{i-k}}$.

The bending value $B_v(i)$ is defined as follows.

$$B_v(i) = \frac{1}{2}\left|\vec{u_b} + \vec{u_f}\right|, 0 \le B_v \le 1$$

The denominator of the above equation normalizes the bending value to lie between 0 and 1.

2.3.2 Corner points

If $B_v(i)$ is greater than a threshold and for , point is considered as a corner point. In other words, we locate those points with local maximum bending values as corner points. If r is too large and two corner points are located in the same range of 2r, one of them will not be detected. This may cause the curve fitting error in the succeeding procedures. In our experiment, the value of r is set smaller than k.

2.3.3 Corner Insertion

Two conditions are considered to add a corner point in the middle of the contour segment. (1) If a segment is very long, its description may have high error.(2) The difference of the Freeman codes of two segment ends is too large.

2.3.4 Corner deletion

If a point and its two neighboring points, and , are on a straight line, the point will not be considered as a corner point. If such a point is detected to be a corner point, we delete it.

3. Contour Segment Description

Two types are used to describe the shape of characters: "straight line" and "cubic Bezier curve".

If the maximum distance from the contour segment to the line connecting two endpoints and is small, we describe the contour segment by a straight line. Since this condition represents that each point on the contour segment is very closed to the line . It is suitable to use a straight line to approximate it. Otherwise, we describe it by a cubic Bezier curve.

We initialize the control points of two sub-segments denoted as . The direction and length of the control points and are selected. The direction of the control point and are changed to the direction. The purpose to change the direction of initialing the control point and is to satisfy the property 4 (geometric continuity) of Bezier curves.

4. Experimental results

We will evaluate the feasibility of our shape description method and compare the shape description results with Adobe Streamline 4.0 standard version. We will also present and analyze our experimental results.

There are eight pages including 2731 contours selected from different Chinese ancient calligraphy copies. After corner detection, 35907 contour segments are extracted

Table 4.1 shows the segment count and the segment rate. We can find the segment rate of the straight lines is lower that that of the Bezier curves.

	Straight Line	Bezier curve
Segment count	9546	26361
Segment rate	26.6%	73.4%

Table 4.2 show that we focus on dealing with the Type 1 (general case), since most situations belong this type.

	Type 1	Type 2	Type 3	Others
Type count	26252	16	4	89
Type rate	99.6%	0.04%	0.02%	0.34%

Table 4.3 show the number of fitting times and successful rate in our system.

Iteratoin num	Number of successful fitting	Number of re-fitting contour	Fitting successful rate(%)	Accumulated number of successful	Accumulated fitting successful

ber	contour segments	segments		fitting contour segments	rate(%)
1	5576	20785	78.85	20785	78.85
2	2150	3426	13.00	24211	91.84
3	594	1556	5.90	25767	97.75
4	142	453	1.71	26219	99.45
5	34	108	0.40	26327	99.87
6	13	21	0.08	26348	99.95
7	2	11	0.04	26359	99.99
8	1	1	0.01	26360	99.99
9	0	1	0.01	26361	100
10	0	0	0	26361	100

5. References

[1] L. Cinque, S. Levialdi and A. Malizia, "Shape desciption using cubic polynomial Bezier curves," Pattern Recognition Letters, Vol. 19,No. 9, pp.821-828,1998.

[2] L. Cinque and L. Lombardi, "Shape description and recognition by a multiresolution approach," Image and Vision Computing, Vol. 13, No. 8, 1995.

[3] H. H. Chang and H. Yan, "Vectorization of hand-drawn image using piecewise cubic Bezier curves fitting," Pattern Recognition, Vol 31, No. 11, pp. 1747-1755, 1998.

[4] T. Kimoto and Y. Yasuda, "Shape description and representation by ellipsoids," Signal Processing: Image Communication, Vol 9, No. 3, pp. 275-290, 1997.

[5] M. J. Wang, W. Y. Wu, L. K. Huang and D. M. Wang, "Corner detection using bending value," Pattern Recognition Letters, Vol. 16, No. 6, pp. 575-583, 1995.

[6] K. Wall and P. E. Danielsson, "A fast sequential method for polygonal approximation of digitized curves," Computer Vision, Graphics, and Image Processing, Vol. 28, pp. 220-227, 1984.

[7] P. J. Schneider, "An algorithm automatically fitting digitized curves," Graphics Gems, A. S. Glassner, ed., pp. 612-626. Academic Press, San Diego, CA, 1990.

[8] H. N. Phien and N. Dejdumrong, "Efficient algorithms for Bezier cureves," Computer Aided Geometric Design, Vol 17, pp. 247-250, 2000.

[9] T. M. Cronin, "A boundary concavity code to support dominant point detection," Pattern Recognition Letters, Vol 20, pp. 617-634, 1999.

[10] J. Hu, D Yu and H. Yan, "A multiple point boundary smoothing algorithm," Pattern Recognition Letters, Vol 19, pp. 657-668, 1998.

[11] S. Lejun and Z. Hao, "A new contour fill algorithm for outlined character image generation," Comput. & Graphics, Vol. 19, No. 4, pp. 551-556, 1995.