WEB BASED CHINESE CALLIGRAPHY LEARNING WITH 3-D VISUALIZATION METHOD

Yingfei Wu[1], Yueting Zhuang[2], Yunhe Pan[3], Jiangqin Wu[4]

College of Computer Science, Zhejiang University, Hangzhou 310027, China
[1]hardwyf@hotmail.com, [2, 3, 4]{yzhuang, panyh, wujq}@zju.edu.cn

ABSTRACT

Chinese calligraphy is pictographic and each calligraphist has his own writing style. People often feel difficult in writing a demanded beautiful calligraphy style. In order to help people enjoy the art of calligraphy and learn how it is written step-by-step we present a new approach to animate its writing process by 3-D visualization method. In this paper some novel algorithms used in the approach are presented to solve the following problems: 1) estimate varied stroke's thickness 2) extract strokes order from an offline Chinese calligraphic writing. Through this approach we implement a system. Experimental result is given to demonstrate the application finally.

1. INTRODUCTION

Chinese calligraphy is a famous type of Chinese art. Some calligraphic writings were written long times ago. Modern people can't see how they were written at that time. The key issue in learning to write such calligraphy is to figure out its writing process. Our aim is to extract strokes order and stroke's thicknesses from offline images obtained by scanner and finally animate the writing process with 3-D visualization method.

There is some previous research about strokes order extraction. Yoshihar Kato estimated drawing order of single-strokes handwritten images [1]. Jäger retrieved temporal information of single-strokes using the minimum curvature by solving a travelling salesman problem from a skeleton image [2]. However Chinese calligraphic writing is composed of units and each unit is composed of strokes. Unit can not be treated as a single-strokes handwritten image. So the scheme [1, 2] is difficult to extract strokes order from Chinese calligraphic writings. Another important issue is stroke's thickness extraction. Ip and Wong proposed a model to simulate the relationship between the maximum width of stroke and radius of brush stem. The model can estimate the radius after determining the maximum width of the stroke. [3]. The model needs a training process. Here we propose a new strategy to extract strokes order and stroke's thickness from Chinese calligraphic writings. With these methods we implement a web based 3-D Chinese calligraphy recur system.

This paper is organized as follows: in Section 2, the algorithm of stroke's thickness and strokes order extraction is described. In Section 3, experimental result is presented. The conclusion is described in Section 4.

2. EXTRACTION OF STROKE'S THICKNESS AND ORDER

To extract stroke's thickness we begin by applying the thinning algorithm to extract stroke skeleton. Thinning is a process of obtaining the skeleton of a calligraphy character image. Skeleton is invariant to the stroke's thickness and keeps the main stroke structure of a calligraphy character in features. According Rutoviz's definition of crossing number for a pixel, the degree ρ of a skeletal pixel is defined as the number of pixels existed in its 8-neighborhood. Skeletal pixel can be categorized into three types according to ρ: vertex pixel, line pixel and fork pixel, defined as following:

a) if $\rho = 1$, then it is vertex pixel;

b) if $\rho = 2$, then it is line pixel;

c) if $\rho > 2$, then it is fork pixel;

Vertex pixel is the start point or end point of a stroke and it can be categorized into two categories: start vertex and end vertex. Fork pixel can be further classified due to intersection mode of two strokes: X cross, T cross. The another type of fork pixel is corner of stroke. Under T cross circumstance, fork pixel maybe is start vertex of a stroke. We denote this kind of fork pixel as hidden start vertex.

2.1. Stroke's Thickness Extraction

Stroke's thickness is an important property to rebuild writing process. It consists of thickness of each skeletal pixel. [3] describes a method using ellipse to obtain stroke's thickness. We find that it is complicated to use ellipse to describe thickness of a skeletal pixel as contour pixels around a skeletal pixel is discrete after extraction of skeleton and contour of a calligraphic character. So we use a discrete method to approximate the thickness w. w can be

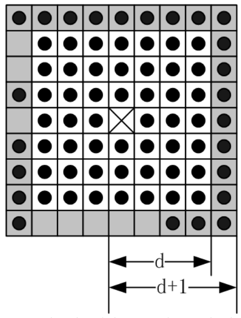


Fig. 1. Central square marked with "X" is a skeletal pixel. (d+1) is the distance from the skeletal pixel to the first ring of which not all the contour pixels are black.

defined as: $w = d + \frac{T_b}{T_{d+1}}$, (1)

where d represents the distance from skeletal pixel to the outermost ring of which all the contour pixels are black, T_b is the total number of black pixels of the Gray Ring. Gray Ring represent the first square ring of which not all the contour pixels are black. T_{d+1} is the total number of the pixels of Gray Ring. We can get:

$$T_{d+1} = (2(d+1)-1)^2 - (2d-1)^2 = 8d \quad (2)$$

From (1) and (2), the thickness of skeletal pixel can be calculated using the following formula:

$$w = d + \frac{T_b}{8d} \quad (3)$$

2.2. Removal of Spurious Vertex and Spurious Cluster Point

Spurious vertex occurs at the corner of strokes such as tick, pressing down and etc. It is due to the variation of the thickness during the calligraphic writing process. Spurious vertex is not a real vertex and it will create unwanted stroke segment.

Spurious vertex usually connects to a cluster which is an intersection of some strokes. To remove spurious vertex we define a vertex set $V = \{v_1, v_2, ..., v_n\}$ for each cluster. Each element of V is a vertex pixel and has a direct path to the cluster. $\mathrm{O} = \{dist(v_c, v_i), (v_i \in V)\}$ is a distance set, where $dist(v_c, v_i)$ is the distance from cluster point v_c to the vertex pixel v_i. Let $dist_{\max}$ be the maximum value of O. The probability of v_i to be a spurious vertex is defined as: $\theta = \frac{dist(v_c, v_i)}{dist_{\max}}$. α is a constant value used as a threshold to find spurious vertexes according to the following rule:

If $\theta < \alpha$, then v_i is a spurious vertex; otherwise, v_i is not a spurious vertex.

Spurious cluster point is caused by the thin ink connection between two adjacent strokes which belong to two different units. It makes two different units like one unit. Spurious cluster point must be removed from skeleton to get the right units information for further character structure analysis. The ratio of the cluster point's thickness and the average stroke's thickness of a character is used to estimate the probability. A cluster point has more probability to be a spurious cluster point if the ratio is smaller.

2.3. Strokes Order Tracing

After removing spurious vertex and spurious cluster points, strokes order can be extracted. We use $G = \{g_1, g_2, ..., g_n\}$ to define the skeleton graph, where each sub-graph of G represents a unit consisting of strokes which have path connected with each other.

2.3.1. Character Structure Analysis

Writing rules are important when people learn how to write a Chinese calligraphy character in the right way [4]. The general principle of these rules is writing from left to right and from top to bottom. To use these rules, character structure should be extracted first. Chinese characters have many structure types, such as left-right structure, top-bottom structure, surrounding structure, and semi-surrounding structure and so on. For example, if a character belongs to left – right structure, all the strokes of the left unit should be traced first and then the right unit.

The algorithm of orientation relationship calculation between each pair of sub-graphs of G is used to extract the character structure. Supposing we want to get the orientation relationship between two sub-graphs g_k and g_l . g_k consists of a strokes $\{s_1, s_2, ..., s_a\}$ and g_l consists of b strokes $\{s_1, s_2, ..., s_b\}$. The orientation relationship can be represented by a two-tuple $R = \{R_h, R_v\}$,where R_h represents the horizontal orientation relationship, and R_v represents the vertical orientation relationship. The orientation relationship has four types: at the left side, at the right side, at the top side, at the bottom side. TABLE 1 shows the probable orientation relationship between g_k and g_l.

We use the following formula to get R_h :

TABLE 1
Orientation relationship between two sub-graph corresponding to the value of R_h and R_v

	Orientation Relationship
$R_h = 1$	g_k is at the left side of g_l
$R_h = 0$	g_k is neither at the left side nor at the right side of g_l
$R_h = -1$	g_k is at the right side of g_l
$R_v = 1$	g_k is at the top side of g_l
$R_v = 0$	g_k is neither at the top side nor at the bottom side of g_l
$R_v = -1$	g_k is at the bottom side of g_l

$R_h = T(h_l, h_r)$, where $T = \begin{Bmatrix} 1, if\,(h_l > h_r) \\ 0, if\,(h_l = h_r) \\ -1, if\,(h_l < h_r) \end{Bmatrix}$ and h_l is the probability of g_k at the left side of g_l, and h_r is the probability of g_k at the right side of g_l. R_v, h_t and h_b are defined in the similar way as R_h, h_l and h_r.

Formula (4) is the definition of h to calculate h_l, h_r, h_t, or h_b according to the orientation relationship needed to be compared between g_k and g_l:

$$h = \frac{\sum_{i=1,j=1}^{i=a,j=b} p(f_1(s_i), f_2(s_j))}{a * b}, s_i \in g_k, s_j \in g_l \quad (4)$$

where $p = \begin{Bmatrix} 0, if\,(f_1(s_i) \geq f_2(s_j)) \\ 1, if\,(f_1(s_i) < f_2(s_j)) \end{Bmatrix}$. With different orientation relationship to be compared, function f_1 and f_2 have different definitions. TABLE 2 shows the detail definitions of f_1 and f_2. After analyzing the character structure, we extract the traversing sequence of all the sub-graphs.

2.3.2. *Marking Double Traced Path*

Double traced path (DTP) is a segment of a stroke which needs to be travelled twice. During real writing process, calligraphist often uses DTP in local area to make the whole character more beautiful. The directions of traversing DTP first time and second time are just reversed with the change of start point and end point. The following is the definition of DTP:

TABLE 2
List of function definition of f_1 and f_2 corresponding to orientation relationship to compare. v_s and v_e represent the start vertex and the end vertex of a stroke of g_k. v_s' and v_e' represent the start vertex and the end vertex of a stroke of g_l.

Orientation Relationship	f_1	f_2
h_l	$Max(v_s.x, v_e.x)$	$Min(v_s'.x, v_e'.x)$
h_r	$Min(v_s.x, v_e.x)$	$Max(v_s'.x, v_e'.x)$
h_t	$Max(v_s.y, v_e.y)$	$Min(v_s'.y, v_e'.y)$
h_b	$Min(v_s.y, v_e.y)$	$Max(v_s'.y, v_e'.y)$

$D_p = \{(v_i, v_c), (v_c, v_i)\}$, where v_c is a member of clusters and v_i can be start vertex, end vertex or branch vertex. There is a direct path between v_i and v_c. D_p traverses to v_c from v_i and backs to v_i from v_c. DTP is difficult to detect so DTP is marked first before we start extracting strokes order. Path (v_c, v_i) will be added to the stroke set as a new stroke after marking (v_i, v_c) as a DTP.

2.3.3. *Minimum Energy Tracing*

Now we can extract strokes order from sub-graphs one by one according the traversing sequence of sub-graphs. The vertex closer to the origin of the calligraphic image will have more priority to be selected as a start vertex in a sub-graph.

After a start vertex is selected the Minimum Energy Tracing (MET) algorithm is used to trace the path. The main idea of MET is that when an intersection point is encountered, the traversal proceeds along the smoothest path for it costs minimum energy. This method is especially useful to extract strokes order from cursive script calligraphic writings.

Tracing process of a stroke will be finished when the algorithm reaches to end vertex. All the cluster points passed during stroke tracing process will be marked as a hidden start vertex. These hidden start vertexes will be added to the start vertex set.

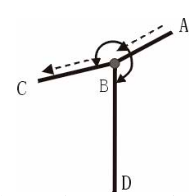


Fig. 2. Traversal starts from point A to B. Then it encounters a crossing point B. As the angle between AB and BC is closer to 180 degree than the angle between AB and BD, the path BC is chosen to be traced next.

The algorithm will continue selecting start vertex and tracing other strokes of this sub-graph until all the start vertexes have been traced.

3. EXPERIMENTAL RESULTS

Based on the above approach we implement a web based 3-D Chinese calligraphy recur system. This system consists of two modules. One is the Calligraphic Image Preprocessing Module which is responsible for extracting strokes order and stroke's thickness and storing them into database. In this module the first step is to split the original image into small images according to the border of each character. The second step is to regularize the split image: 1) Removing noise; 2) Convert image into gray image. A noisy hole in image will result in wrong skeleton extraction. The third step is to extract stroke's thickness and strokes order. It includes these sub-steps: extraction of skeleton and stroke's thickness, removal of spurious vertex and spurious cluster points, character structure analysis and marking DTP. The last step is strokes order tracing.

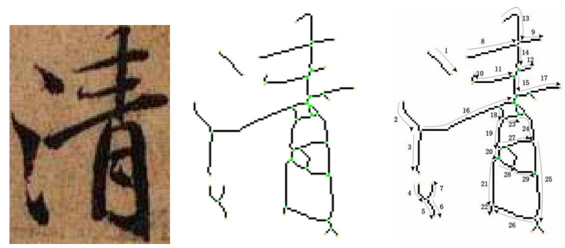
The other module is the Client Module which is responsible for animating the writing process. It is an applet coded in Java with the support of the Java 3D API. Users can view the animation of calligraphic writing process by their browser.

The experiment result of using α as threshold to remove spurious cluster points is shown in TABLE 3. Fig. 3a is a character written by a famous Chinese calligraphist called Xizhi Wang. The skeleton after applying the thinning algorithm is shown in Fig. 3b. Fig. 3c is the result of strokes order extraction. Some screen shots of the writing process is shown in Fig. 4.

TABLE 3

Matching and missing rate of removing spurious cluster points using our 300 sample characters database with different threshold.

α	Matching Rate (%)	Missing Rate (%)
0.20	63.3	32.1
0.24	67.5	28.3
0.28	69.8	20.7
0.32	65.4	25.2
0.35	62.3	29.9



(a) (b) (c)

Fig. 3. (a) Original calligraphy character and (b) its skeleton after thinning, (c) the result of strokes order extraction: this character has 29 strokes according to our proposed stroke extraction rule.

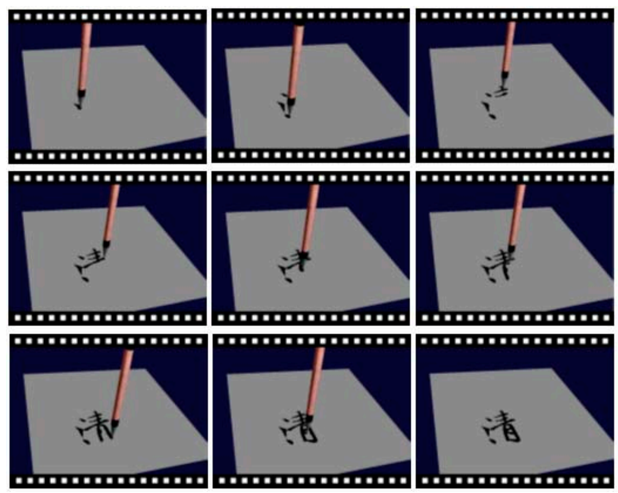


Fig. 4. Screen shots of the writing process: one shot every 3 seconds

4. CONCLUSION

In this paper we discuss some novel methods to extract stroke's thickness and strokes order from Chinese Calligraphic writing. Based on these ideas, we implement a web based 3-D Chinese calligraphy recur system that can help people enjoy the writing beauty of calligraphy character and set good writing examples for users to follow.

5. ACKNOWLEDGEMENT

This project is supported by NSFC and CADAL (China America Digital Academic Library) project.

6. REFERENCES

[1] Yoshiharu Kato, Makoto Yasuhara, "Recovery of drawing order from single-strokes handwriting images", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL.22, NO.9, pp. 938-949, SEPTEMBER 2000

[2] Stefan Jäger, Daimler-Benz Research and Technology, "Recovering Writing Traces in Off-Line Handwriting Recognition: Using a Global Optimization Technique", Proc. 13th Int'l conf. Pattern Recognition, pp. 152-154, 1996

[3] Sam T.S. Wong, Howard Leung and Horace H.S. Ip, "Model-based Analysis of Chinese Calligraphy Images", 9th Intl. Conf. on Information Visualization (IV05), London, England, July 2005.

[4] Zen Chen, Chi-wei Lee, and Rei-Heng Cheng, "Handwritten Chinese Character Analysis and Preclassification Using Strokes Structural Sequence", Proceedings of ICPR'96, pp. 89-93, 1996