TASK QUARTERLY 11 No 1–2, 129–136

GENERATING NEW STYLES OF CHINESE STROKES BASED ON STATISTICAL MODEL

MIAO XU[1] AND JUN DONG[2]

[1]*School of Information Science and Technology,*
East China Normal University,
Shanghai, 200 062, P. R. China
xumiao_1@yahoo.com.cn

[2]*Software Engineering Institute,*
East China Normal University,
Shanghai, 200 062, P. R. China
jdong@sei.ecnu.edu.cn

(Received 24 December 2006; revised manuscript received 17 January 2007)

Abstract: Chinese calligraphy is one of the most important Chinese arts: a form of entertainment as well as an embodiment of figurative thinking. In this paper, a statistical model-based approach to generating new styles of Chinese character strokes is proposed. Original calligraphy samples are aligned in a common co-ordinate frame and a training set consisting of landmarks is generated semi-automatically. The most significant features of the training set are extracted and a statistical model is built in order to generate strokes in new styles. The Bezier curve is used to fit the discrete contour data.

Keywords: figurative thinking, Principal Components Analysis, Bezier curve

1. Introduction

In calligraphy, Chinese characters can have many different fonts, and even the same font often appears in different styles. Figure 1 shows various "Horizontal" stroke styles of the "Li" font, one of the most widely used Chinese calligraphy font.

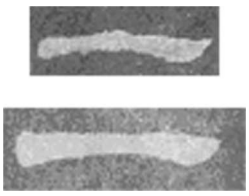

Figure 1. Different "Horizontal" stroke styles of the "Li" font

Because of their complicated structure and appearance, being able to master some styles does not mean being able to manage other styles satisfactorily. Moreover,

some written characters are not aesthetical enough. At the same time, simulating calligraphy creation has its importance in cognitive science, where cognitive processes, patterns of figurative thinking and intelligence models are key issues [1]. In this paper, a statistical model-based approach to generating new styles of Chinese strokes is proposed to simulate the basic process of calligraphy creation.

The paper is organized as follows. Selected related works are reviewed in Section 2. An overview of the proposed approach is given in Section 3. A statistical model of generating new styles of Chinese characters is presented in Section 4. Results are discussed in Section 5, while potential applications and remaining questions are discussed in Section 6.

2. Related works

In most cases, the automatic generation of Chinese calligraphy can be divided into several phases: (1) decomposition of characters into strokes, (2) modeling from a training set and (3) generation of new artwork. An algorithmic framework has been proposed for an advanced virtual brush to be used in interactive digital painting [2]. Compared with other virtual brushes, this system is designed to present a realistic brush in that it simulates the complex painting functionality of a running brush in accurate and stable fashion. Moreover, an intelligent system using a constraint-based analogous-reasoning process has been devised, combining knowledge from multiple sources to support a restricted form of reasoning [3]. The result is visually pleasing since it can automatically generate Chinese calligraphy that meets aesthetic requirements. However, the transformation on each level depends on the weight of different samples. Its efficiency is a bit low, as a lot of experience is required to adjust the weight. It would be better if the primary features of a character could be extracted first. The main purpose of our work is to devise a way to extract critical features from large numbers of samples and then build a model with those extracted features.

3. Overview of the new approach

What we want to obtain is a statistical model with a parameter. This statement can be represented as:

$$\boldsymbol{X} = M(\boldsymbol{b}), \tag{1}$$

where $\boldsymbol{X}$ is the target stroke to be generated with a new style and parameter $\boldsymbol{X}$ is a vector that determines the stroke's features (*e.g.* its length or width). Its dimensions represent the number of modes found. When the parameter is replaced with specific values, a new style will be generated. It is better to find correspondence between the parameter and the stroke style.

Therefore, samples are selected to form a training set. Each of them is marked with several points, including the points on the contour and control points, which will be used later in generating the consecutive contour curve. After that, the training set is aligned using Generalized Procrustes Analysis (GPA) and the samples' primary features of these are extracted by Principal Components Analysis (PCA) [4]. A few characteristic features of the training set are extracted and represented in the form of several eigenvectors. With the extracted features, we can generate new styles of strokes in steps shown in Figure 2.

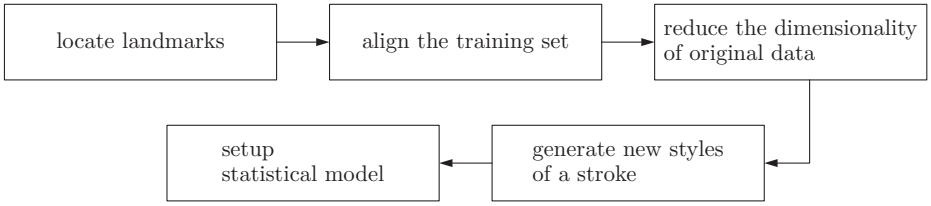

Figure 2. Five steps of generating new styles of a stroke

4. Construction of the model

4.1. Locating landmarks

Suitable landmarks (*i.e.* selected points on each boundary of a stroke) should be located in order to form a training set. Good choices for landmarks are points which can be consistently located from one stroke to another, for example, the corners of stroke boundaries [5]. In our approach, a semi-automatic method is used for this purpose.

(1) *Calculate the angles of all the points on the contour of a stroke.* The l-angle of a point, P_k, is defined as the angle between two lines, $P_{k-l}P_k$ and P_kP_{k+l}. Usually, calligraphy is not perfectly clear and error would appear in the contour data. So, we compute ten angles, from 1-angle to 10-angle, and take their mean as the final angle of point P_k.

(2) *Search the points located at sharp turns of the boundary.* A threshold is set in order to find these turning points. If the angle of point P_k is below the threshold, the point will be inserted in the training set, $\boldsymbol{T}$. Some points along the boundary between two sharp turning points are also inserted into the training set, $\boldsymbol{T}$, so that the shape of a stroke can be described clearly.

(3) *Insert the control points of the Bezier curve in the training set*, $\boldsymbol{T}$. Control points C_k are calculated with the following equation:

$$C_k = 2T_k + T_{k-1}/2 + T_{k+1}/2, \tag{2}$$

where T_{k-1}, T_k and T_{k+1} are three continuous turning points found in step 2.

The following process requires various training strokes to have the same number of landmarks, so that inserting or deleting turning points by hand is allowed and new control points are calculated according to formula (2). Figure 3 is an example of automatically generated landmarks. The real contour of the stroke is dashed, while the solid line is a contour generated by the marked points. Please note that the rectangle points are those on the contour while the circular points are the calculated control points.

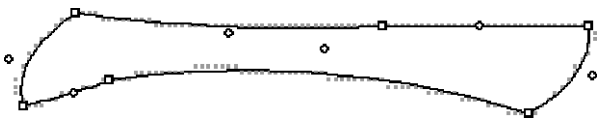

Figure 3. Automatically generated landmarks: points on the contour are marked rectangular, control points are circular; the original stroke contour is dashed, the solid line is a contour generated by the marked points

A stroke can be represented by landmark points. For example, if there are n landmark points for a stroke, $\{(x_i, y_i)\}$ being the co-ordinates of the i^{th} point, then a vector of $2n$ elements, $\boldsymbol{X} = (x_1, \ldots, x_n, y_1, \ldots, y_n^T)$, is created to describe the stroke.

4.2. Aligning the training set

Before any further operations are performed on the training set, the variation that might affect further analysis should be removed and one should make sure that all the samples are in the same co-ordinate frame.

One of the most popular algorithms among them is GPA, used for comparing shapes of objects. The main idea of this method is moving each shape to minimize the sum of the distances of each shape to the mean.

Each shape undergoes three kinds of transformations during the alignment: translation, scaling and rotation. Because all the above three transformations are similarity transformations, the geometrical information remains unaltered by the process.

We center all samples on the origin, with a unit scale and some fixed but arbitrary orientation. The last of three approaches to alignment introduced in [5] is applied in this paper. This approach is to transform each shape into space tangent to the mean. The space tangent to a vector $\boldsymbol{x}$ is a hyperplane normal to $\boldsymbol{x}$, passing through $\boldsymbol{x}$. The advantage of the tangent space approach is that it keeps the distribution compact and minimizes non-linearity [5].

4.3. Statistical model

If the distribution of parameter $\boldsymbol{b}$ can be learnt from the training set, we are able to generate new styles, which are related with but different from the original samples.

Let us suppose to have s samples, each of n points in a 2-dimension plane. In our cases, n tends to be a large number (determined by the number of landmarks) and it is not easy to analyze such a large body of data. We have to reduce its dimensionality first.

Reducing dimensionality. PCA is a useful method of reducing dimensions to a manageable range. The method's basic idea is to describe the variation of a set of multivariate data in terms of uncorrelated (linearly independent) variables, each of which being a linear combination of the original variables. The new variables are derived in the decreasing order of importance so that, for example, the first principal component accounts for as much variation in the original data as possible. The objective of this analysis is usually to judge whether the first few components account for most of the variation in the data. If this is the case, it is argued that they can be used to summarize the data with little loss of information, reducing the data dimensionality.

PCA is often performed by Singular Value Decomposition (SVD) on a covariance matrix of samples. This process produces two orthogonal matrixes and a diagonal matrix, *i.e.* $\boldsymbol{X} = \boldsymbol{USV}^T$. Here, $\boldsymbol{X}$ is the samples' covariance matrix, $\boldsymbol{S}$ is a diagonal matrix, and $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrixes. The elements of $\boldsymbol{S}$ (eigenvalue of $\boldsymbol{X}$) are arrayed in the descending order. Assuming that Φ_i is the corresponding eigenvector

of λ_i, we can create a model with $\boldsymbol{\Phi}$ containing t eigenvectors corresponding to the t largest eigenvalues.

Setting up the model. After all the eigenvectors have been computed, t most significant eigenvectors are selected to form a matrix. Here, the t eigenvectors could represent the main feature of the stroke. Therefore, new styles of a stroke can be generated by using the following equation:

$$\boldsymbol{X} = \bar{\boldsymbol{X}} + \boldsymbol{\Phi} \bullet \boldsymbol{b}, \tag{3}$$

where $\bar{\boldsymbol{X}}$ is the mean after alignment of the original sample, while $\boldsymbol{\Phi}$ is the matrix composed of the t eigenvectors mentioned above.

Regarding the number of modes, t, the following rule is applied:

Since the first t components account for a proportion $P = \sum_{i=1}^{t} \lambda_i / V_T$ (V_T being the total variance in the original data equal to the sum of all eigenvalues), given a rate of say 98%, we can choose t largest eigenvalues.

Generating new styles. A new style closely related with the original training set can be generated with a specific $\boldsymbol{b}$. The range of parameter $\boldsymbol{b}$ should be specified, *e.g.* if b_i, a component of vector $\boldsymbol{b}$, is independent and the distribution is Gaussian, the range of b_i is $|b_i| \leq 3\sqrt{\lambda_i}$. However, it has proved not to be an easy task to specify the range in our cases.

4.4. Curve fitting of discrete contour data

The data generated by the model are discrete points. In order to obtain a consecutive representation of a stroke, curve fitting is necessary, of which there are several alternative methods. The most widely used one is to represent the contour of a stroke with a few sets of lines and curves. Here, the curves may be quadratic spline, Beta-spline, B-spline or Bezier curves not greater than cubic [6]. We have used a quadratic Bezier curve, having several advantages in our model. It requires only a few points to generate a consecutive contour of a stroke (see Figure 3). This significantly reduces the amount of computation and removes error from the training set. It can retain the original style of the font when a stroke is zoomed in or out, so it has been accepted as an industry standard.

When fitting a contour, we first judge whether the points are in a line or not. If positive, the Bresenham algorithm is used; if negative, a quadratic Bezier curve is applied [7]. The control points which computed at the landmark stage are used in the process.

The obtained results have proved that a quadratic Bezier curve is sufficient to generate a Chinese stroke with just a few points.

5. Results

We have used Visual C++ and Matlab 6.5 to implement experiment with the model proposed in this paper.

In our first experiment, the training set originated from a single sample, *viz.* the Chinese "Horizontal" character stroke in the "Li" font (see Figure 3), stored in Microsoft Word 2000. The results are shown in Figure 4. The places marked with circles were modified significantly with a different parameter.

In the second experiment, our training set was also based on the Chinese "Horizontal" character in the "Li" font, but the original samples came from multiple sources. Some of them are shown in Figure 5 and the results are presented in Figures 6–7. The results of the second experiment indicate that the dimension of $\boldsymbol{b}$ actually represents different modes.

These modes explain global variation due to different styles. The results also demonstrate that less significant modes cause smaller, more local changes (see Figure 7). For example, the mode corresponding to b_1 was more significant than others, corresponding to b_2 and b_3.

The results obtained from our system using four training samples as the input are shown in Figure 8. They demonstrate that this approach can yield strokes of different styles that can be reconstructed as a single character. Please note that the characters shown in Figure 8 are decomposed into several primitive strokes and reconstructed by hand[1].

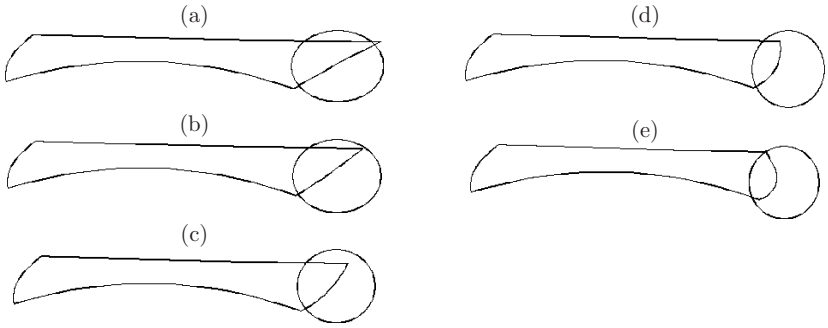

Figure 4. The training set comes from only one sample. The parameter $\boldsymbol{b}$ has only one dimension, $\boldsymbol{b} = [b_1]$, which is adjusted from -0.10 to 0.10: (a) $b_1 = -0.10$, (b) $b_1 = -0.05$, (c) $b_1 = 0$, (d) $b_1 = 0.05$, (e) $b_1 = 0.10$

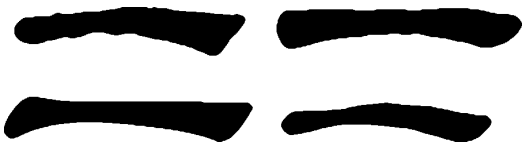

Figure 5. Samples of the second experiment

6. Applications and discussion

The approach proposed here has several potential applications. First of all, it can be applied in the publishing industry, especially in publications of ancient artwork. As some calligraphy works have suffered severe abrasion during their long history, it is impossible to maintain the original artwork. This approach could be used to generate similar characters to be selected for publishing.

1. We selected some generated stokes and constructed a single character by hands. An automatic method is being studied and developed to aid this work. However, this work is beyond the paper here.

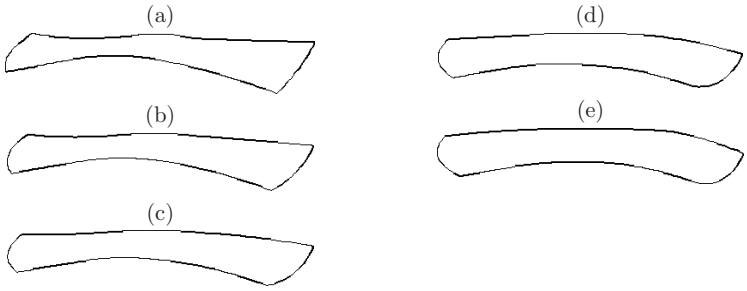

Figure 6. The training set comes from multiple samples. The parameter $\boldsymbol{b}$ computed by PCA has three dimensions: $\boldsymbol{b} = [b_1, b_2, b_3]$. The first element b_1 in parameter is adjusted from -0.10 to 0.10: (a) $b_1 = -0.10$, (b) $b_1 = -0.05$, (c) $b_1 = 0$, (d) $b_1 = 0.05$, (e) $b_1 = 0.10$. The other two elements, b_2 and b_3, remain 0 in this experiment. Note that all of the new generated strokes are similar to the original samples, but each of them has different style. In conclusion, they are derived directly from the statistics of a training set

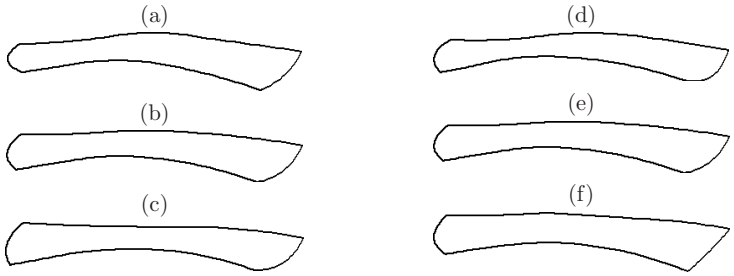

Figure 7. The training set the same as in the second experiment, but other elements of parameter $\boldsymbol{b}$ adjusted: b_2 in the left column and b_3 in the right column. In the left column, the change occurred mainly at the stroke's ends (the left end thickened and the right one became thinner with increasing b_2). In the right column, the change occurred mainly in the middle part, its upper boundary more protruding with increased b_3

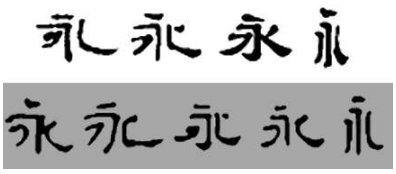

Figure 8. A single character ("forever" in Chinese) in different styles: training samples in the top row, the bottom row reconstructed with strokes automatically generated by our system

Another application is to generate personalized fonts according to users' preferences. For example, users may provide the system with their handwriting, from which primary features will be extracted and plausible styles will be generated. Users can adjust and choose one or more styles according to their preference.

The approach has some advantages in generating new styles of Chinese character strokes. It does not require much computation once the PCA algorithm is adopted and reduces the dimensionality of the original samples into a manageable range. Since a quadratic Bezier curve is used to fit the discrete contour data, only a few points are required to generate new styles of a stroke.

However, extensive further work is necessary.

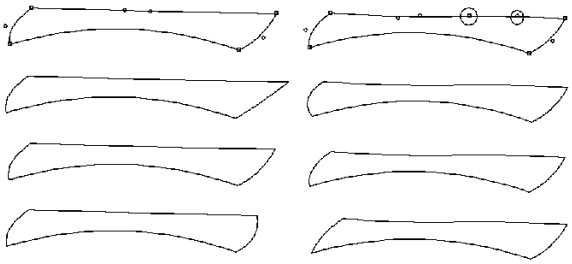

Figure 9. The training set used in the first experiment (one sample only). The between the two columns differ in the number and position of landmarks (the two points marked with circles in the second column). The landmarks in left column are as per the first experiment, while those in the right column were marked by hand. The area changed by the model is related with the number and positions of landmarks when samples are insufficient

First of all, when a training set is too small, the effect is uncertain, as shown in Figure 9. This defect can be eliminated by supplying the model with more samples.

Secondly, we have not managed so far to obtain one-to-one correspondence between elements of parameter $\boldsymbol{b}$ and specific features of style (long or short, *etc.*). If we can establish such relationships, this approach will be more efficient. For example, if element b_1 represented the length and element b_2 represented the width, we could easily adjust the length or width of characters without affecting their other features simply by adjusting the relevant element.

Thirdly, our model focuses on transforming a stroke instead of a whole character, so the construction of characters with strokes is not discussed here.

Acknowledgements

The authors would like to acknowledge Dr Lao Zhiqiang of Department of Radiology, University of Pennsylvania, for providing useful material and discussion.

Supported by Shanghai Basic Research Key Project (06JC14058), National Basic Research Program of China (2005CB321904).

References

[1] Dong J 2006 *World Sci.-Technol. Res. Dev.* **28** (3) 7 (in Chinese)
[2] Xu S H, Lau F C M, Tang F and Pan Y H 2003 *Computer Graphics Forum* **22** 533
[3] Xu S H, Lau F C M, Cheung W K and Pan Y H 2005 *IEEE Intell. Syst.* **20** (3) 32
[4] Dryden I L and Mardia K V 1998 *Statistical Shape Analysis*, New York John Wiley & Sons, London
[5] Cootes T F and Taylor C J 2000 *Statistical Models of Appearance for Computer Vision, Technical Report*, University of Manchester
http://www.isbe.man.ac.uk/∼bim/Models/app_models.pdf
[6] Ma X H, Pan Z G and Zhang F Y 1996 *Chinese J. Computers* **19** (2) 81 (in Chinese)
[7] Hearn D and Baker M P 1997 *Computer Graphics*, 2nd Edition, Prentice Hall, New Jersey