DRS 2012 Bangkok
Chulalongkorn University
Bangkok, Thailand, 1–4 July 2012

Building a Simulation Platform for Chinese Calligraphy Characters

WANG Chung-Shing[a], CHANG Teng-Ruey[b], LIN Man-Ching[c], and WANG Ya-Hui[a]

[a]Tung-Hai University
[b]Nan-Kai University of Technology
[c]Feng-Chia University

Abstract

This research looked at an existing Chinese calligraphy font, the Kai-font, and carried out geometric modeling and analysis to set up a stroke feature model database. This allowed creation of an independent calligraphy simulation platform for extracting and analyzing fonts. Users can then input through the use of a mouse, digit pad or other systems to draw out the strokes of a Chinese character. The system will then automatically recognize the stroke starting coordinates, angles, area ratio and other parameters for stroke feature analysis and extraction. These data can then be compared with stroke feature information in the database to produce a simulated calligraphy font. This platform also allows the user to output the font simulation result to 3D drawing software such as SolidWorks for further design and applications. The advantage of this research hoped to develop a calligraphy font simulation system with a user friendly interface to allow traditional Chinese beginners having the opportunity to write out a Chinese calligraphy font on the computer. This will in turn serve to stimulate interests in Chinese calligraphy learning. Furthermore, this research enables the user to output the results to a computer aided design field for further application while promoting and spreading the art of Chinese calligraphy.

Keywords: computer geometric modeling, Chinese calligraphy, stroke recognition, feature extraction, stroke database, calligraphy characters platform

1. Introduction

The Chinese calligraphy is a very unique and important asset in Chinese culture and its high artistic value is praised by people all over the world. Although currently there are many different kinds of Chinese fonts available for selection for the computer system, the ability for the user to provide modification to the font itself is very limited. For example: we cannot offer a calligraphy font style that caters to personal handwriting habits to create new or composite words for processing and application. This research uses computer geometry modeling to analyze and extract characteristic strokes to produce a handwritten Chinese calligraphy font in order to build a calligraphy character platform. This offers the users an opportunity to further study and understand the writing structure of Chinese calligraphy.

In this study, we analyzed the Kai-font in Microsoft True Type fonts. Through analysis of the characteristic strokes of the Kai-font and carrying out simultaneous processing of strokes from non-traditional writings, we can remove, process and summarize duplicated strokes to create a stroke characteristic database [Chung & Tseng, 1995]. After that, by using the interface for the system platform, users can input handwritten font stokes with input devices such as the mouse or digit pad. The system will then automatically retrieve the corresponding strokes to carry out “Thinning” and “Stroke feature extraction” process for comparison with the characteristic stroke database to find the corresponding stoke to complete font output. The simulated results are then outputted as a .jpg file. These files are automatically exported into 3D computer aided design software, SolidWorks for further processing and application.

2. Paper Survey

Stroke characteristics are an important feature unique to Chinese calligraphy. Each individual stroke is composed of points, lines and curves. The point corresponds to the black spot; line refers to the linear movement of the tip of the pen and the curve is characterized by curve motion of the pen [Hornby, 1972; Stallings, 1976, 1977]. Strokes can be defined using operation points and parameters. Operation points may include the origin, end point, length, direction… etc. of the stroke. Parameters are then used to define the shape, width, position… etc. for the stroke [Wong & Hsu, 1995]. However, Chinese

calligraphy font strokes have a huge amount of variations which result in an extremely complex way for describing strokes. Even for similar a stroke structure, a small change in the shape, angle, position length or proportion can result in a completely different character.

Calligraphy font recognition, also known as reverse recognition, aims to achieve an effective way to obtain the point cloud data for the fonts. Lin et al., (2001) used an automatic threshold value setting to identify fonts and create a common words database. The search function of the database was then used to identify the fonts in the document. Detection and identification of common characters allows recognition of handwritten or printed fonts. Romero et al., (1997) used artificial neural network algorithm to enhance the identification rate for similar words along with the use of different search directions to modify the neural parameters to improve recognition speed. Lin et al., (1996) used a "Trend-followed" translation technology to find the contour segment and overall characteristic information for the font. Chuang et al., (1995) used heuristic algorithm to identify printed fonts while utilizing contour features to increase recognition rate.

Previous research by the authors used image processing and reverse engineering (RE) in Chinese calligraphy [Wang, et al., 2006]. This research combined processes of RE, grey prediction theory in pattern processing, geometric modeling for constructing Chinese calligraphy characters and rapid prototyping (RP) for model making. First of all, the written Chinese calligraphy was scanned by scanner. Next, the contours of the Chinese characters were detected with image processing. These contours were then converted to point data, which can be easily processed in any CAD software by using B-Spline curves to fit the points. An example was illustrated below using a Chinese compound word "the fortune and treasure are coming" to show the steps of the process. Finally, a RP model was constructed to show various applications for the products in Chinese calligraphy pattern. With this research, a collaboration between the Chinese calligraphy in handcraft and digital virtual design can be realised; but more importantly, the aesthetic aspects in the characters can be preserved (Figure 1).

(a) Image file of compound character

(b) Grey prediction contour detection

(c) Compound character in a paper weight

(d) A hanging decoration

Figure 1 Reverse design process for a Chinese compound character

3. Methodologies

In this research, we developed a simulation platform for Chinese calligraphy so the user can quickly create simulated calligraphy fonts for further application. The research framework consisted of the following three components:

(1) Stroke database construction

(2) Stroke feature recognition and coding.

(3) Creating a simulation platform and user interface for writing stroke input.

3.1 Stroke database construction

In order to create a client based calligraphy simulation system, we first need to generate a corresponding stroke database. We used the existing built-in Kai-font within Microsoft Windows operating system as basis to obtain stroke information. Kai-fonts are TrueType vector fonts with characters constructed from individual strokes and we can extract its font structure data using Microsoft Development Network (MSDN). With TrueType vector fonts, font data structures are arranged as such that one character can be composed of many stroke contours and every contour can be composed of many curves. Statistics show that the Kai-font has 23,230 characters and 178,196 contour data.

Using the character 「永」 as example: The creation of the character with Kai-font is separated into six stroke contours, as shown in Figure 2. The meanings of each original stroke contour data are represented below. The mathematical relationship between ❶❷❸❹❺ is:

❶ = 16 + (❷+ ❸) *4 + (❹ + ❺) * 8

❶ Size of the contour in the font file in bytes; ❷ Number of straight lines; ❸ Number of curves; ❹ Number of points used by the straight line; ❺ Number of points used by the curve; ❻ Angle of xy; ❼ The actual Chinese character ; ❽ What number stroke it is in the Chinese character. Consider the first stroke in the character 「永」, the “point” stroke, the contour has 0 straight lines, 8 curves, 16 points used by the curves (2 points for 1 curve) and a 39.2 degree angle. The total size is 16 + (0+8)*4 + (0+16)*8 = 176 (Bytes).

❶	❷	❸	❹	❺	❻	❼	❽
176	0	8	0	16	392	永	1

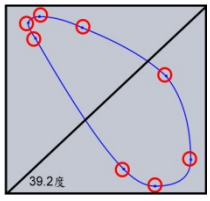


Figure 2 First stroke contour information of the character 「永」

If we input all the contours of the Kai-font into the contour database, a great amount of time is required for the contour recognition search process, which becomes highly inefficient. Therefore, duplicate contours were removed to reduce number of comparisons required in order to increase operational efficiency. The simplest method was to analyse common contours structures between different characters and removing those contour structures that were the same between them.

This research used the GetGlyphOutline function in Win32 API to access contour data in selected characters. The following stroke contour data were then analyzed to determine which contours need to be deleted.

1. Size occupied

Because Kai-font characters were created from individual strokes, the strokes with same structures take up the same amount of data space. For example, the first stroke in 「永」 and「汁」 and the second stroke in「汀」 both have the 「、」 structure and occupied 176 bytes. However, apart from the size, we also need to consider other attributes such as the angle between xy, to decide whether to remove the selected stroke contour as illustrated in Figure 3.

Figure 3 Stroke contours with same data size

2. Number of lines, curves and points

Each individual stroke may contain many lines or curves and we can distinguish between them by the number of lines, curves and points. For example: the first stroke of 「刀」、「刁」、「力」 all contain the same number of lines, curves and points; Therefore, they all they all have the same structure (Figure 4).

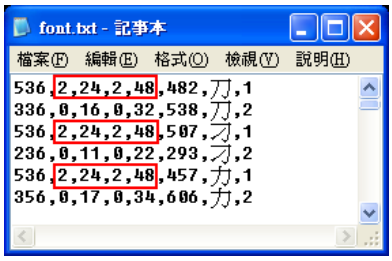


Figure 4 Strokes with same of lines, curves and points

3. Angle of xy

We can determine the structure of the stroke contour by looking at the diagonal angle formed by a box containing the stroke. As shown in Figure 5, the first stroke of the character 「九」 has an angle of 35.2 degrees and the second stroke 62.3 degrees. Calligraphy fonts with same strokes but with different stroke contour height and widths can be distinguished this way.

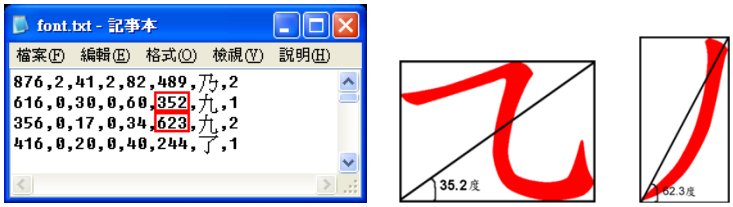
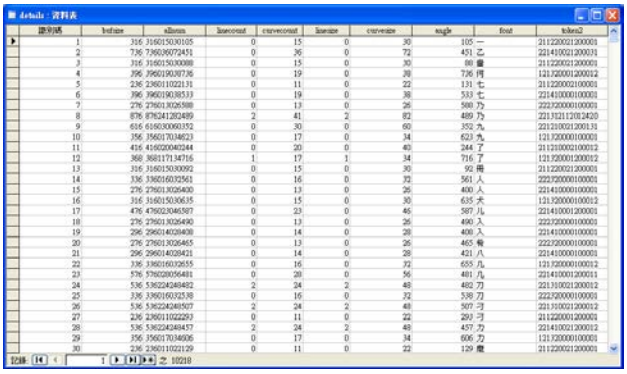


Figure 5 xy angle of the character 「九」

Based on the three criteria mentioned above, same stroke contours can be deleted from the extracted font file. If the three values were the same between different strokes, the one with the largest dimensions was kept because smaller stroke contours were more likely to have errors when carrying out stroke feature analysis. After analysis and deletion, a total of 10,218 stroke contour data remains. This contour characteristic database was then stored in Microsoft Access as shown in Table 1.

Table 1 Contour coding of contour database



3.2 Stroke feature recognition and coding

Stroke feature recognition and coding were the most important aspects of identification and analysis. Normally before feature extraction of writing strokes we needed to carry out a “thinning” (also known as skeletonizing) process. Thinning refers to the process of eliminating points from the outlines of shapes or fonts with different widths until only curves with one pixel width remains. This research used the “Peeling approach” with further corrections and modifications to achieve skinning of font strokes [Zhang & Suen, 1984]. Peeling method, also known as the “Iterative morphological method”, used the relationship between non-zero pixels (which represent the location of line data) and neighboring pixels to determine if it is the edge of the line. If it is, the value is removed and replaced by zero. Using this way similar to peeling method, we can sequentially

reduce width from both sides of the line until the width is only one pixel but still retain the connection of the line.

This research proposed a stroke feature and coding method as basis for stroke feature recognition. Each stroke was encoded as a 15 bit unit to generate the stroke characteristic value. Coding in this research was divided into four parts as shown in Table 2. Font stroke distribution characteristic values, which were mainly used as a point of comparison with written stroke characteristic values, can be separated into four categories (Table 3).

Table 2 Stroke characteristic coding index

items		bits
1. Stroke aspect ratio (length v.s. width)		2
2. Stroke vertex position		2
3. Angle between vertices		1
4.Stroke distribution	(1) The segments in y-axis when scanning in x direction	5
	(2) The segments in x-axis when scanning in y direction	5

Table 3 Stroke distribution characteristic value

Items	Index name
1. The segments area percentage in x-axis when scanning in y direction	bnx0 ~ bnx4
2. The segments area percentage in y-axis when scanning in x direction	bny0 ~ bny4
3.Fitting the partition percentage in x-axis when scanning in y direction	cx0 ~ cx4
4. Fitting the partition percentage in y-axis when scanning in x direction	cy0 ~ cy4

4. A Simulation Platform for Chinese Calligraphy Characters

Current research aims to develop a simulation platform for Chinese calligraphy character recognition. Softwares used for development are shown in Table 4.

Table 4 Software for a simulation platform

Items	Software
Operation system	Windows XP
Programming language	Borland Delphi 7.0
Strokes database	Microsoft Access 2003
3D software	SolidWorks 2010 SP0

The system platform used the “Kai-font” as output for simulated fonts. Figure 6 illustrated the writing simulation process. In addition to outputting as solid fonts, we can also choose to output as font outlines or thin fonts. In addition to providing an output mode on screen, the font simulation platform can utilize the API function in SolidWorks to import simulated fonts as 3D models into software so designers can easily carry out follow-up applications. After importing to SolidWorks, each stroke became an individual unit and can be modified as shown in Figure 7. In Chinese calligraphy, certain auspicious phrases can be written as a compound word. For example: 「招財進寶」、「囍」 …etc. Figure 8 and Figure 9 showed the compound word, 「招財進寶」 and the output screen in SolidWorks.

Figure 6 A process for Chinese character「永」

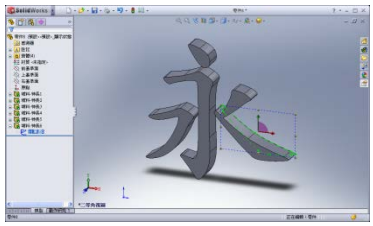


Figure 7 Output「永」to SolidWorks

Figure 8 Compound character writing

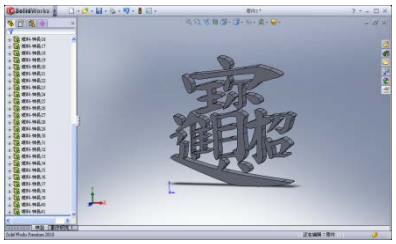


Figure 9 Output to SolidWorks

5. Conclusion

This research aims to develop a platform for Chinese calligraphy font simulation. The results and advantages are as follows:

- A calligraphy font simulation platform was constructed for the Chinese Kai-font. Beginners can easily write out Chinese character strokes with a mouse, digital pen or other input devices and the system platform will construct a simulated calligraphy font in real-time.
- The character stroke database for the Kai-font was reconstructed. Through character stroke feature extraction and analysis we combined strokes that were not practical under normal writing habits in order to delete duplicated strokes. This allowed us to reduce the amount of character stroke data for the Kai-font from 178,196 to 10,218, which increased system processing efficiency.
- A coding system was made for the Kai-font character strokes. Each stroke was encoded as a 15bit unit for comparison with both the index value within the database and the hand-written input strokes.
- With the use of API method within SolidWorks, we can quickly and efficiently export simulated fonts to 3D drawing software such as SolidWorks. This allowed further processing of Chinese compound words, creation of rarely used words, curve fitting and other additional applications.

6. Acknowledgement

This research was partially supported by the National Science Council, Taiwan under the Project (NSC 100-2221-E-029-010).

References

Chuang, C.T., and Tseng, L.Y. (1995). A Heuristic Algorithm for the Recognition of Printed Chinese Characters. *IEEE Transactions on System, Man and Cybernetics,* 25(4), 710-717.

Hornby, A.S. (1972). *The Advanced Learner's Dictionary of Current English*, 2nd Ed.,

Oxford Press.

Lin, J.R., Chen, C.F. (1996). Stroke Extraction for Chinese Characters Using a Trend-Followed Transcribing Technique. *Pattern Recognition,* 29(11), 1789-1805.

Lin, C.F., Fang, Y.F., and Juang, Y.T. (2001). Chinese Text Distinction and Font Identification by Recognizing Most Frequently Used Characters. *Image and Vision Computing,* 19, 329-338.

Romero, R.D., Touretzky, D.S., and Thibadeau, R.H. (1997). Optical Chinese Character Recognition Using Probabilistic Neural Networks. *Pattern Recognition*, 30(8), 1279-1292.

Stallings, W.W. (1976). *Approaches to Chinese Character Recognition, Pattern Recognition*, Pergamon Press, Great Britain, 8, 87-98.

Stallings, W.W. (1977). Chinese Character Recognition, in Fu, K.S., 1977, *Syntactic Pattern Recognition and Application*, Spring-Verlag, New York, 95-123.

Wang, C.S., Hsiao, C.Y., Chang, T.R., and Teng, C.K. (2006). Product Development for Chinese Calligraphy Using Reverse Engineering and Rapid Prototyping, *Virtual and Physical Prototyping,* 1(4), 259-269.

Wong, P.Y.C & Hsu, S.C. (1995). Design Chinese Typeface Using Components, *IEEE Graphics.* 0730-3157/95, 416-421.

Zhang, S., and Fu, K.S. (1984). A Thinning Algorithm for Discrete Binary Images. *Proceedings of the International Conference on Computers and Application*, Beijing, China. 879-886.