A Kai Style Contour Beautification Method for Chinese Handwriting Characters

Lijuan Liu[1], Weiping Xia[1], Lianwen Jin[1*], Huiyun Mao[1], Feng Tian[2]
[1]School of Electronic and Information Engineering
South China University of Technology, China
[2]Institute of Software, Chinese Academy of Sciences, China
[*]lianwen.jin@gmail.com

Abstract—In this paper, we propose a novel contour-based method to beautify online handwritten Chinese character to the Kai style calligraphy. According to the feature and structure of Kai style calligraphy, Bezier curve is used to sketch the user-input stroke segment contour and the corner contour. We get the whole contour path of beautified character by connecting stroke segments' contour and corners' contour end-to-end. Anti-aliasing technology is used to make the edge of the contour fine and smooth. Finally, the path-fill algorithm is adopted to fill the inner of the contour. Our system is proved to be effective and efficient. Meanwhile, users can choose any color for the contour and the inner, which is more convenient for users' design.

Keywords—handwriting beautification, contour-based method, cubic Bezier curve, path-fill algorithm

I. INTRODUCTION

Chinese calligraphy is among the finest and most important of all Chinese art forms and an inseparable part of Chinese history. Its delicate aesthetic appearance is generally considered to assemble the essence of Chinese culture [1]. So through the study of Chinese calligraphy, we can have a deeper understanding of Chinese characters' historical and cultural development.

The Kai style calligraphy has tidy strokes and well-designed structure, besides, it is used most widely, which makes it very suitable and necessary for beginners to learn. Ming dynasty calligraphy theorist Feng Fang [1] said:" The order to learn calligraphy is to learn the Kai style calligraphy first." Nowadays, however, few people practice handwriting in their daily life, let alone calligraphy. In this environment, we propose an online beautification method for the Kai style calligraphy, with the hope that, on one hand, it may be a tool to beautify the user-input in the digital age, on the other hand, it can be used as a computer-aided calligraphy learning system for calligraphy beginners.

The beautification of online handwriting character is a new topic, only few studies remotely relate to this area. Zhu and Jin [2] proposed a method to fuse the standard calligraphic style with the user's writing style by transfiguring the handwriting with a target template. However, this method needs to store many sets of templates which include the contour and skeleton information of all characters. Meanwhile, character recognition is adopted, which makes it more complicated and less reliable. Junji Man et al [3] firstly calculate the contour lines of strokes using information of writing speed, position and the pre-defined stroke model formula, and then fill the inner of the contour. It can beautify the Japanese Hiragana. However, this method needs heavy computational cost and the resulting style is somewhat monotonous. The ougishi software [4] uses trajectory to simulate calligraphy, which deforms user-input trajectories and results into Caoshu style calligraphy work. Yet the resulting character sometimes is hard to recognize, and the style is somewhat monotonous. Our former work [5] has implemented a new method by using user-input trajectory and a set of rendering rules to beautify the online handwriting character into Kai style calligraphy. Yet this method still has many deficiencies, which makes it hard to be used in real world settings.

In this paper, we introduce a novel contour-based method to beautify the handwriting character to the Kai style calligraphy. Firstly, we briefly introduce our former work and describe its three deficiencies in detail. Secondly, we develop our new approach step by step. Exprimental results show that our method is more effective and efficient than our former work. Meanwhile, our system provids for users to choose any color for the beautified character contour and the inner, which outperform our former work and other handwriting character beautification systems.

II. DEFICIENCIES OF OUR FORMER WORK

Our former method is based on the features of Chinese characters and the Kai style calligraphy [5]. These features are also used in our method. Before introducing our former method and its deficiencies, we will review these features briefly and show our former work's three deficiencies at the end of this section.

A. Features of Chinese characters

According to our former method, Chinese character strokes can be classified into two categories on the basis of their skeleton: (1) basic stroke, which contains only one segment that is overall flat; (2) compound stroke, which contains more than one segment that are joined at the corner position. Compound strokes can be formed by basic strokes.

The Kai font character's stroke is considered to be composed of several segments and corners. The representing form of a stroke is given by Stroke=SC+S+{MC+S}+(EC), where SC stands for the corner at the starting point of a stroke, MC the corner within the stroke, EC the corner at the ending

978-1-4244-6588-0/10/$25.00 ©2010 IEEE

point of the stroke and S the stroke segment.{x} means that x may not appear or repeat several times, while (x) means x appears zero or one time. Some examples can be found in Fig.1.

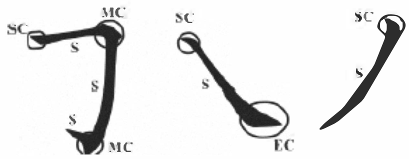


Figure 1. Stroke components

Stroke segments are classified into three categories by their width variation: Type I (normal at the endpoint, while thinner in the middle), Type II (normal at the starting point and gradually thinner) and Type III (gradually become thick from zero width). The stroke segments are illustrated in Fig.2. Nine patterns of corner are summarized, including one point like corner, six kinds of single interface corners and two kinds of double- interface corners. These corners are given in Table. 1.

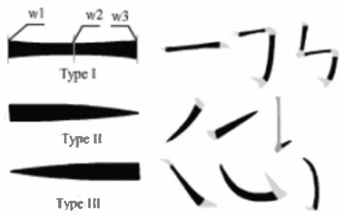


Figure 2. Fundamental stroke segments

TABLE I. FUNDAMENTAL CORNERS

Point	C1
Single interface	C2 C3 C4 C5 C6 C7
Double-interface	C8 C9

B. Three deficiencies of previous work

According to the features introduced in section A, the previous method [5] firstly segments the user's input trajectory into segments, then the segments are fitted into one type of the three segment types. The proper corner shape modeled by numbers of cubic Bezier curves is generated at the corner position. Finally, the beautified segments and corners are organized by a set of rendering rules to generate the beautified character.

However, there are three deficiencies in our former work.

(1) *Rendering of stroke segments*: In our former work, each segment was fitted into cubic Bezier curve using Least Square method [9]. Wu method [6] was used to draw anti-aliased circles to render the segment. The fitted stroke rendered by anti-aliased circles is shown in Fig. 3. When anti-aliasing, there is a large overlap between the previous circle and the next one, which resulted in heavy computational cost, making the method less efficient. Meanwhile, due to anti-aliased circles used for rendering, we couldn't get sharp shape at the end of type II segment, the circles made the sharp end obtuse. A comparision of the end shape of type II segment cropped from characters generated by Flash 8.0 with that obtained by our former method is shown in Fig. 4.

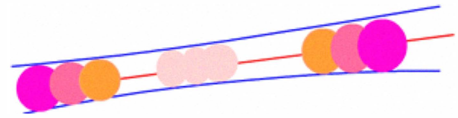
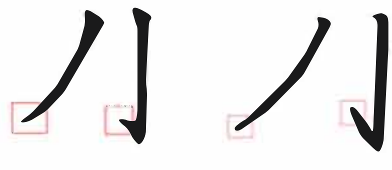


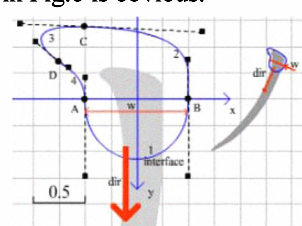
Figure 3. Segment rendering with circles of Wu method



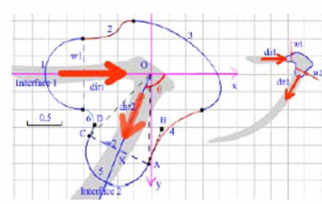
(a) (b)

Figure 4. (a) End shape of type II segment generated by Flash 8.0 ; (b) End shape of type II segment obtained by our previous method

(2) *Smoothing between segments and corners* : In order to get smooth contour in the joint between the segment and the corner model, our former work used extra cubic Bezier curve pieces in the start or the end of the corner model when modeling the corner shape. In Fig.5(a), curve AB, and in Fig.5(b), curve 1 and 5 are these extra curve pieces. This method can somewhat work. However, on one hand, it needs storing redundant information about the extra curves and filling the inner of these curves by flood-fill method[7], which makes it complicated and less efficient to model the corner shape. On the other hand, the anti-aliased circles used for rendering the segment are designed to meet the extra curve in the joint to make the contour of the joint seem soomth and continuous, which is not a real geometric continuity of contour curves and it is hard to cover the discontinuity in the joint. For example, when the beautification width is 3, the discontinuous phenomenon in the joint shown in Fig.6 is obvious.



(a)



(b)

Figure 5. (a)Model of corner C4 ; (b)Model of corner C8

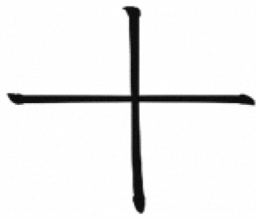


Figure 6. The unsmooth phenomenon in the joint between segments and corners

(3) *Rendering rules*: Rules of rendering in our former work are obtained by repeated observation, which didn't have good mathematical model. So it is hard to say that these rules of rendering are faultless and the system of our former work is robust.

III. CONTOUR-BASED METHOD FOR CHINESE HANDWRITING CHARACTERS

Our goal is to get smooth beautification contour of a character, while the inner part can be filled. So in this paper, we still use the feature of the Kai style character illustrated in Section II. However, the method of beautifying is improved. We use a number of cubic Bezier curve pieces conenected end-to-end to predict the beautified contour of a given character written by users, which means that we could firstly get the contour path of the beautified character, then path-fill algorithm is applied to fill the inner part of the contour and finally obtain the beautification character. We will introduce our method in followings.

A. System architecture

The main flow chart of our proposed system is given in Fig. 7. According to the features of the Kai font character, user's input trajectory, which is captured by mouse or touch screen, is firstly segmented into stroke segments by corner points obtained in a corner detection procedure. Secondly, the proper beautified contour of segments are predicted by cubic Bezier curves and the corners contour are shaped with a number of cubic bezier curves at the proper corner position according to the modified rending rule. Thirdly, the contour path can be obtained by connecting the stroke segments contour and corners contour together in the written order. Finally, we fill the inner with path-fill algorithm and get the beautified character.

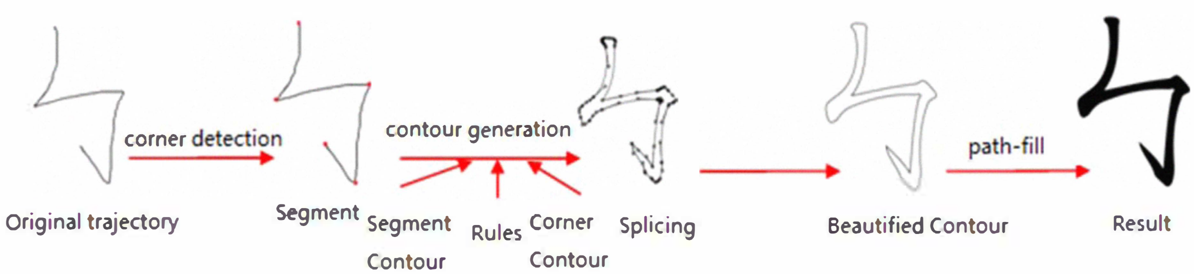


Figure 7. System chart

B. Segments contour prediction

A parametric Bezier curve piece of degree n is defined as (1) [7].

Where P_j are the control points and the $B_{j,n}(t)$ are the Berstein polynomial [7], shown in (2).

$$P(t)=\sum_{j=0}^{j=n} P_j B_{j,n}(t), \qquad 0 \le t \le 1 \tag{1}$$

$$B_{j,n}(t)=\binom{n}{i} t^i (1-t)^{n-i}, \quad i=0,\ldots,n \tag{2}$$

Where $\binom{n}{i}$ is the binomial coefficient. For cubic Bezier curves, $n=3$.

In our method, the beautified segment contour will be predicted and fitted to one of the three kinds of segments contour illustrated in Fig.2. The procedure of finding the outline of segments is described as follows.

Step1. The input trajectory is interpolated using Digital Differential Analyzer method [7].

Step2. Corner detection is carried out on the interpolated curve with the Adaptive Bending Value method [8]. The curve will be divided into one or more segments by the corner points which include both endpoints.

Step3. Each segment is fitted into Bezier curve using Least Square method [9], so we can get four control points P_0, P_1, P_2, P_3 of the fitted Bezier curve.

Step4. As shown in Fig. 8 (takes segment type I as an example), we choose four points in the fitted Bezier curve where $t=0, t=0.25, t=0.75$ and $t=1$. Their value can be computed with (1). Cubic Bezier curve has first-order derivative everywhere property [7]. So we could calculate the tangent value of the above four points, then obtain points in left-side contour and right-side contour respectively at $t=0, t=0.25, t=0.75$ and $t=1$ with the perpendicular bisector method, the width of each point is the same as our former work [7]. By this way we get eight points from both left-side contour and right-side contour. From (1) and (2), for left-side contour and right-side contour, the control points can be derived by(3):

$$\begin{cases} P(0)=P_0 \\ P(0.25)=(1-0.25)^3P_0+3\times0.25\times(1-0.25)^2P_1 \\ \qquad +3\times0.25^2\times(1-0.25)P_2+0.25^3P_3 \\ P(0.75)=(1-0.75)^3P_0+3\times0.75\times(1-0.75)^2P_1 \\ \qquad +3\times0.75^2\times(1-0.75)P_2+0.75^3P_3 \\ P(1)=P_3 \end{cases} \tag{3}$$

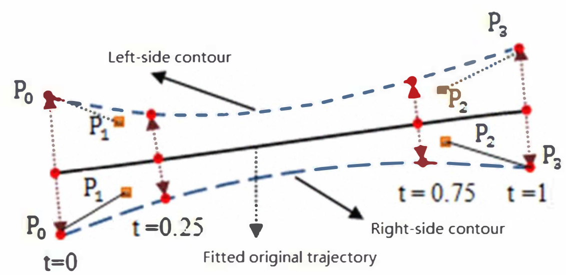


Figure 8. The left-side and right-side contour generation of type I segment

Step5. Store the obtained control points of two sides in two arrays. When we get the whole stroke outlines' control points, we connect them together in accordance with the written order to derive the whole contour path.

C. Corners contour generation

The corner contour is shaped by a number of cubic Bezier curves connected end-to-end, shown in Fig.5. In order to make the corner contour closer to the Kai style calligraphy, in this paper, we remodel the corner contour. The modeling method is similar as our former work [5], except that curves used to smooth the joint between segments and corners are removed, for example, curve AB in Fig.5(a), curve 1 and curve 5 in Fig.5(b) are removed in the remodeling procedure. The number of corners is expanded from nine to eleven. The models of the single interface and one point like corner are not changed, yet the double-interface corners are divided into C8(left heng-zhe), C9(right heng-zhe), C10(left shu-ti) and C11(right shu-ti) (see Fig.9). As shown in Fig. 9, along the written direction, the

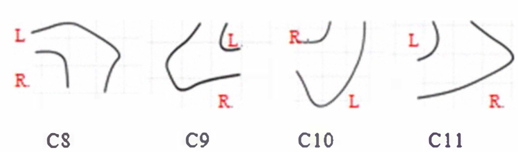


Figure 9. The modeling of double-interface corners

brush's left side is the beautified left contour, and the other side is the beautified right contour. The directions of left and right contour of C8 and C9, C10 and C11 are opposite.

D. Stroke contour path generation

1) Rules for stroke contour generation

We supplement and modify the rule based on statistical information of handwritten Chinese characters. An online database for the "863 program of China" is used to test our former work's rendering rule, from which we saw that the rule for double-interface corners is not so good. Double-interface corners play an important role in Kai style calligraphy, so it is very necessary to improve the rule for it. The way what we do is as follows :

Step1. Choose 100 characters including C8, C9, C10 and C11 corner respectively from different sets of the online handwritten database.

Step2. Use input direction dir1 and output direction dir2 as features, the two features are illustated in Fig.5(b), label the chosen characters manually.

Step3. We finally obtain the rule for the four coners through mathematical statistics with the labeled features,as shown in Table II.

TABLE II. RENDERING RULE FOR DOUBLE-INTERFACE CORNERS

Corners	Rule
C8	$dir1 \in [0,45] \cup [315,360)$ *and* $dir2 \in [70,180]$
C9	$dir1 \in [60,170]$ *and* $dir2 \in [0,60] \cup [290,360)$
C10	$dir1 \in [60,170]$ *and* $dir2 \in [170,290]$
C11	$dir1 \in [0,60] \cup [240,360)$ *and* $dir2 \in [225,300]$

2) Rules for stroke segment cropping

With the modified rendering rule presented in previous section, when we connect segments contour and corners contour together, the cross phenomenon in the joint shown in Fig. 10 will be found. In this case, the cross section won't be filled and the beautification result will be destroyed when filling the inner of the contour with path-fill method. So the user couldn't get beautified contour. In order to get non- cross contour, we design a new rule to achieve this goal.

As is shown in Fig. 11, the yellow line indicates half width of the corner point, which length is $wd/2$, θ is half of the angle between dir1 and dir2, d is the cropped length while wd is the widest value among all the stroke. We use $wd/2$ in order to ensure that there is non-cross between the joints. The cropped length d can be calculated by (4).

$$d = (wd/2)\cot\theta \tag{4}$$

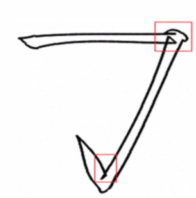


Figure 10. The cross phenomenon when connecting segments contour and corners contour

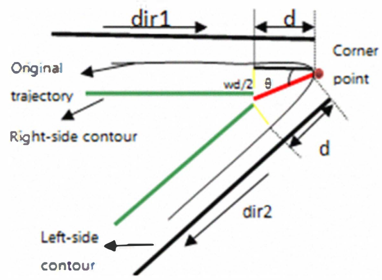


Figure 11. Segments cropping method

3) *Rules for contour smoothing*

The whole contour is composed by segments contour and corners contour by a number of cubic Bezier curves. If we want to get continuous and smooth contour, cubic Bezier curves' first-order geometric continuity should be achieved [7].

Given two cubic Bezier curves $P_i(t)$ and $Q_j(t)$, control points of each are respectively P_i ($i = 0,1,2,3$) and Q_j ($j = 0,1,2,3$) [7], illustrated in Fig. 12.

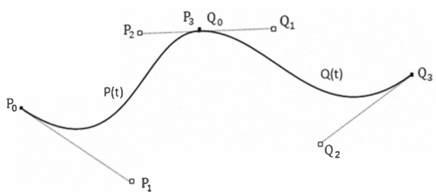


Figure 12. Cubic Bezier curve connecting

The necessary and sufficient condition to enable $P_i(t)$ and $Q_j(t)$ to achieve the first-order geometric continuity is:

$$P_2, P_3 = Q_0, Q_1 \text{ In a line}$$

The first-order geometric continuity property can ensure the joint contour of two cubic Bezier curves natural and smooth [7]. In this paper, we move the shorter cubic Bezier curve's first point Q_0 to the longer one's last point P_3, let $Q_0 = P_3$. Then adjust Q_1 and P_2, make sure that the three points are in one line. One example of the whole contour connected by cubic Bezier curves is shown in Fig.13.

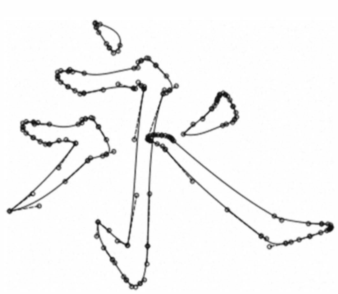


Figure 13. Smooth contour generated by connecting piecewise cubic Bezier curves

E. *Path-fill*

After obtaining the whole stroke contour path, we fill the inner of the contour with efficient path-fill algorithm and do anti-aliasing offered by GUI. Finally, we get the beautified character of the Kai style calligraphy, as illustrated in Fig. 14.

Figure 14. One example for contour-based method

IV. EXPERIMENTAL RESULTS

We applied our method on both handwritings acquired with mouse and samples from the 863 online handwriting database. The new method is compared with our former method [5] in effectiveness and efficiency.

A. *Comparision in effectiveness*

As shown in Figure 15, characters in the first colum are acquired by writing with mouse. The second colum are results by our method, the third colum are results by our former method. The fourth are characters from the 863 online handwriting database. The fifth and the sixth are results of our method and our former method respectively. From Fig. 15, we can see that results of our method are much closer to the Kai style calligraphy. Besides, our method let users output the beautified contour and change the color of contour and the inner, which can't be achieved by our former work or other handwriting character beautification systems. Examples are given in Fig. 16.

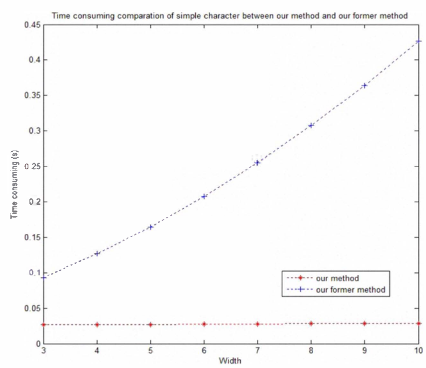
B. *Comparision in efficiency*

Recalling the entire process, our method just needs once fitting and once transformation, and GUI offers efficient anti-liasing and path-fill method, which makes the system highly efficient. However, our former method used repeated circles of Wu method to fill the segments, there is a large overlap between the previous circle and the next one, which makes the system less efficient. Fig. 17 shows the time consumption of beautifying one simple character example and one complex character example respectively.

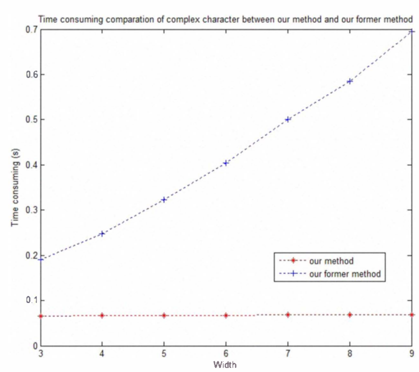
Figure 15. Experimental results of new method and the previous method

Figure 16. Experimental results of the new method to change color of stroke contour and the inner

From Fig.17, it can be seen that for both the simple character and the complex one, when the width grows large, the time used by our method has little change, but the time used by our former method has become very large. For the same width, the time used by our former method is at least more than three times as that of our method. When the width grows, the multiple becomes even greater.



(a)



(b)

Figure 17. (a)Time consuming comparison of simple character '人' between our method and our former method; (b)Time consuming comparison of complex character '到' between our method and our former method

V. CONCLUSION

Based on the features of Chinese characters and Kai style font characters, we propose a new contour-based method to beautify Chinese Handwriting characters. After improving the rendering rule, we make full use of various properties of cubic Bezier curves, dynamically generate the beautified contour of Chinese characters, and use path-fill method to fill the contour path and get the beautified character of the Kai style calligraphy. Experimental results show that our method can reach the expected goal. Compared with the former work, our method is more effective and efficient.

Our appoach has many potential applications. First of all, it may serve as a tool for beautifying online characters in PC or tablet PC software. Compared with other character beautification system, our system can offer some other choices for user, such as changing color of the contour or the inner, even with gradient color, outputing the beautified character or just the contour. Secondly, according to features of Kai style calligraphy, it is very suitable for calligraphy beginners to practise, so it can be used as a computer-aided calligraphy learning system for calligraphy beginners. Thirdly, for its high efficiency, the proposed method can be implemented into mobile devices, which is generally configured with pressure-senseless touchscreens. Under circumstances where signatures, drawings on a greeting card, or other handwritings are needed, all of these characters can be greatly beautified.

As a matter of fact, our system still has much space to be improved. The proposed method is not adapted to all kinds of calligraphy styles, since the method is based on rules of rendering, for some calligraphy styles, such as Cao shu, the rules of rendering could be very complex.Therefore, further study should be carried out to address this issue.

REFERENCES

[1] "Feng Fang · Tong Xue Shu Cheng", http://www.yingbishufa.com/zgyb/lilun/gudai_fengfang002.htm

[2] Xinghua Zhu, Lianwen Jin, "Calligraphic Beatification of Handwritten Chinese Character: A Patternized Approach to handwriting Tranfiguration", *Proceedings of ICFHR2008,* Aug. 2008, pp.135-140

[3] Junji Mano, et al, "A Method to Generate Writing-Brush-Style Japanese Hiragana Character Calligraphy" , icmcs, p.9787, 1999 *IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)* - vol. 1, 1999, pp.9787-

[4] Mikihiro OHBA. Ougishi, *http://www.ne.jp/asahi/o/o/ougishi/*

[5] Weiping Xia, Lianwen Jin, "A Kai Style Calligraphic Beautification Method for Handwriting Chinese Character", *Proceedings of ICDAR2009, July*. 2009, pp.789-802

[6] Xiaolin Wu, Jon G. Rokne, "Double-step incremental generation of lines and circles", *Computer Vision, Graphics, and Image Processing*, pp.331-344, Academic Press Professional, Inc. San Diego, CA, USA, 1987

[7] Donald Hearn , M. Pauline Baker, *"Computer Graphics, C version, 2nd Ed"*, Prentice Hall, 1997

[8] Wang M.J, Wu W.Y, Huang L.K, et al. Corner Detection Using Bending Value[J]. *Pattern Recognition Letters* 199516(6):575-583

[9] Xiaohu Ma, Zhigeng Pan, "The Automatic Generation Algorithm of High Quality Outline Front Using Bezier Curve", *Acta Automatica Sinica*, Vol.20, No.1 Jan. 1994, pp.121-125