# Research and Development of Brush Calligraphy System Based on Mobile Terminal

XIA Yuxiao[1,a], LI Fang[1,a], WU Zhongcheng[1,a], SHEN Fei[1,a], WEN Guohua[1,a]

[1]High Magnetic Field Laboratory, Chinese Academy of Science, Hefei, Anhui, 230031, China

[a]xyx1202@126.com

**Keywords:** mobile devices; brush calligraphy; handwriting generation; ink diffusion; practical applicability

**Abstract.** In this study, a brush calligraphy system was designed, in order to bring change to the monotonous writing trajectory demonstration on mobile devices and offer better user experience during writing. Handwriting position and velocity were captured by information acquisition module. Based on the captured information, writing trajectory was drawn in real time by handwriting generation module. An eight neighborhood ink diffusion algorithm was proposed and ink diffusion module was implemented to achieve ink diffusion effect. Experimental results demonstrated that the simulation effect is similar to the real effect of brush writing. This system achieved a realistic of calligraphy on mobile devices as well as improved the usability of brush calligraphy system by leaving out the requirement for specialized input equipment and the complex process of parameter setting.

## Introduction

As the rapid development and popularization of mobile terminals such as smart phones and tablet devices in recent years, handwriting interaction has become one of the most common ways of interaction in people’s daily life. But the monotonous writing trajectory in mobile devices has gradually made people forget how to write beautifully and ignore the artistic value contained in handwriting. Many studies have been carried out in order to achieve the art effects of brush calligraphy. From the aspect of design methods, these studies can be divided into three categories, including virtual brush modeling, calligraphy generation and trajectory analysis based handwriting beautification.

In terms of virtual brush modeling, Strassmann[1] proposed a two-dimensional model in which the stroke was defined by a set of pressure and position parameters. Wong et al.[2] synthesized realistically Chinese calligraphic handwritings by using the oval brush model as well as the ink deposit model. An elastic 3D virtual brush was introduced by Lee[3] to improve the writing effect of Strassmann’s model in 2004. Mi et al.[4] simplified the calculation by using an empirical-based raindrop model to stimulate the contact area between the brush and paper. Xu et al.[5] developed a model based on the calligraphy generation technology to produce different fonts with calligraphic style using SARP transformation. Other attempts were also made to improve the writing effects in electronic devices through handwriting beautification. Jungpil et al. [6] developed a generation method for a handwritten-style font that reflects an individual's handwriting by using vector quantization. Jakubiak et al.[7] tried to use SSF(i.e. Stylized Stroke Fronts) method to beautify the writing stuff. However, existing methods are large calculating, time-consuming and inapplicable to non-pressure-sensitive mobile terminates.

In this paper, a brush calligraphy system was designed. Writing trajectories were generated on the basis of the handwriting position and velocity captured by information acquisition module. An eight neighborhood ink diffusion algorithm was proposed and ink diffusion module was implemented to achieve ink diffusion and realistic brush calligraphy effects. Experimental results demonstrated that a realistic of calligraphy on mobile devices has achieved. Furthermore, the writing experience of this brush calligraphy system is similar to that of traditional writing on paper.

## System Design

**System Overview.** In order to achieve a real-time realistic calligraphy effect, the whole system is divided into three modules: information acquisition module, handwriting generation module and ink diffusion module (see Fig. 1). Information acquisition module is used to collect user's position information and speed information during writing. Handwriting generation module utilizes the position information and speed information acquired by handwriting generation module to rotate and scale the brush. And the brush filling algorithm is used to solve the problem of low acquisition frequency and generate smooth handwriting. In the ink diffusion module, an eight neighborhood ink diffusion algorithm is adopted for real-time ink diffusion. The degree of ink diffusion is controlled by the user selected paper type and ink type, and finally different ink diffusion effects are achieved.

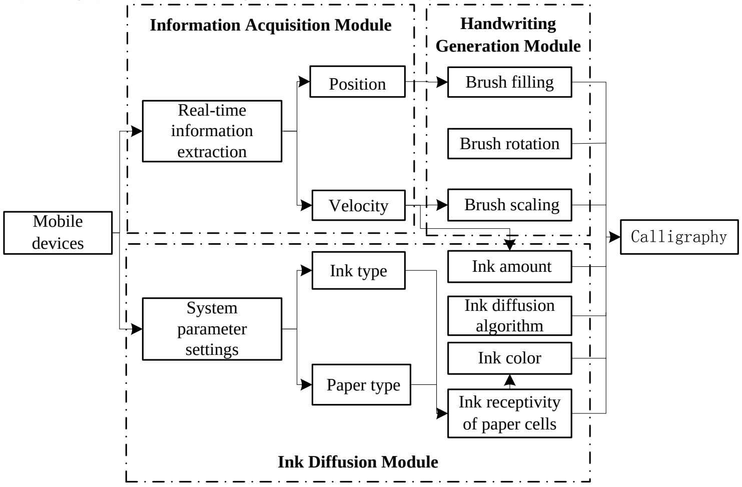


Fig. 1 System design

**System Module Design.** Information Acquisition Module. The main function of information acquisition module is to monitor user's handwritten touch events, including the collection of pen position information and velocity information. In the process of writing, a series of coordinate points can be acquired and denoted by $(p_1, p_2, p_3\ldots\ldots p_{n-1}, p_n)$ ， $p_i(x_i, y_i, t_i)$ $(0<i<n+1)$. Where $t_i$ is the timestamp, $x_i$ represents the position of the horizontal axis at time $t_i$, $y_i$ represents the position of vertical axis at time $t_i$. Then, the velocity $V_i$ of time $t_i$ can be defined as:

$$v_i = \frac{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}{t_i - t_{i-1}} \tag{1}$$

Handwriting Generation Module. At present, most of the brush simulation algorithms require a pressure-sensitive device. However, mobile devices generally are not equipped with a pressure sensor. Therefore, in this section we work to achieve the generation of the handwriting based on writing speed information and position information from a practical point of view.

(1) Brush Rotation. According to the experience of a virtual brush model[4], only the case of the contact portion of the brush and paper is considered. The real brush model and the corresponding "raindrop" brush model are shown in Fig. 2. Raindrop brush model is composed of two circumscribed circle and the area surrounded by two common tangent components of the two circles.

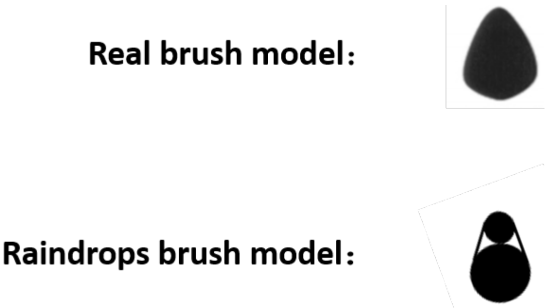
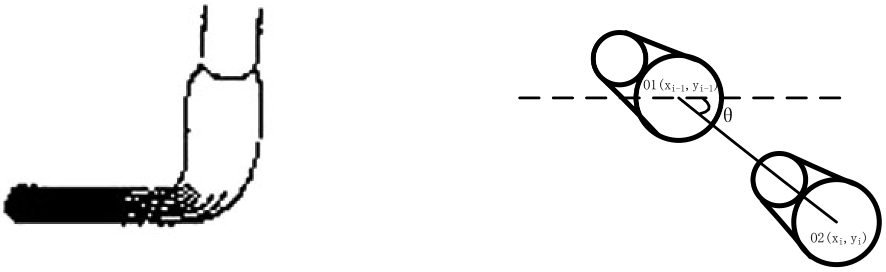


Fig. 2 Real brush model and raindrop brush model

The process of brush writing is shown in Fig. 3a. during writing the direction of the brush is always the same with the movement direction, and therefore, in this paper we determine the rotation angle of the brush by the slope of current point and the previous point, to achieve real-time rotation of the brush. As shown in Fig. 3b, the rotation angle is calculated as follows:

$$\theta = \operatorname{atan}\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right) \tag{2}$$

Where $x_{i-1}$ and $y_{i-1}$are the previous point coordinates, $x_i$ and $y_i$ are the coordinates of the current point, $\theta$ is the angle of rotation.



a The process of brush writing b The direction of brush rotation

Fig. 3 The process of brush writing and the direction of brush rotation

(2) Brush Scaling. In this paper, we propose a brush scaling method based on the speed without the utilizing of pen pressure. The thickness of the stroke is adjusted by the handwriting speed. Provided that $p_{i-1}(x_{i-1}, y_{i-1}, t_{i-1}, w_{i-1})$ and $p_i(x_i, y_i, t_i, w_i)$ are two adjacent points, where in $w_{i-1}$、 $w_i$ are stroke width of each point. The writing state is judged by the speed difference $\triangle V$, the speed difference is calculated as follows:

$$\Delta v = (v_i - v_{i-1}) * a \tag{3}$$

Where, $v_i$ is velocity of the current point, $v_{i-1}$ is velocity of the previous adjacent point, $a$ is a proportionality constant. If $\Delta v \in [0.75, 0.9]$, corresponding to uniform movement, no significant change is shown in stroke thickness; if $\Delta v \in (0, 0.75)$, corresponding to deceleration of pen movement, stroke gradually becomes thicker; if $\Delta v \in (0.9, +\infty)$, corresponding to acceleration, thinner stroke is drawn.

Stroke width changes with three parameters: width $w_{i-1}$ of the previous adjacent point $p_{i-1}$, the reference width $w$ and velocity difference $\triangle V$. The strategy is: the greater velocity difference is, more dramatic the change of the speed is; the wider reference width is, the greater the change in stroke width is. Therefore, stroke width of $p_i$ can be defined as:

$$w_i = \begin{cases} w_{i-1} + w & \Delta v \in (0, 0.75) \\ w_{i-1} & \Delta v \in [0.75, 0.9] \\ w_{i-1} - w & \Delta v \in (0.9, +\infty) \end{cases} \tag{4}$$

Where, $w_i$ is stroke width of the current point, $w_{i-1}$ is stroke width of the previous adjacent point by iterative calculation, $\triangle V$ is the speed difference calculated by Eq. 3, $w$ is the reference width calculated by the speed difference, when the speed difference is larger, the reference width changes larger, while the reference width reaches a certain value then the change width of the stroke tends to a constant. Therefore, the basic relationship between the width and the speed difference can use power function fitting, as the following formula:

$$w = w_{\max} - b(w_{\max} - w_{\min})\Delta v^{\lambda} \tag{5}$$

Where, $\lambda$ is the harmonic factor, $b$ is the scaling parameter, $w_{\max}$ is the maximum reference width, $w_{\min}$ is the minimum reference width.

(3) Brush Filling. Since the sampling frequency of the mobile terminal device is usually too low

to cause a sparse brush which cannot form smooth handwriting, so the brush need to be filled. In the course of the response of the touch event, will take advantage of handwriting filling algorithm adaptive padding every two adjacent points, ultimately generate a continuous smooth curve. The algorithm is as follows:

**Step 1.** Determining the starting point ($x_{i-1}$, $y_{i-1}$) and the end point ($x_i$, $y_i$) of two adjacent points; the starting point of rotation angle $\theta_{i-1}$ and stroke width $w_{i-1}$; the end point of rotation angle $\theta_i$ and stroke width $w_i$.

**Step 2.** Calculating the slope of the starting and the end point slope calculation wired $k$:

$$k = (y_i - y_{i-1})/(x_i - x_{i-1}) \tag{6}$$

Calculate increments $\Delta x$ in the x-direction cause a change in y to generate a straight line:

$$\begin{cases} x_{j+1} = x_j + \Delta x \\ y_{j+1} = y_j + \Delta y = y_j + \Delta x * k \end{cases} \tag{7}$$

Linear generation completed, go to Step 5.

**Step 3.** Calculating the number of the filling points as n from the starting to the end point:

$$n = (x_i - x_{i-1})/\Delta x \tag{8}$$

Calculate the rotation angle tolerance $\Delta\theta$:

$$\Delta\theta = (\theta_i - \theta_{i-1})/(n+1) \tag{9}$$

Calculate the stroke width tolerance $\Delta D$:

$$\Delta w = (w_i - w_{i-1})/(n+1) \tag{10}$$

**Step 4.** According to the rotation angle tolerance and the scaling factor tolerance to calculate the rotation angle and stroke width of each filling point, and draw a rotating scaled “raindrops” on the straight line by every equidistant step.

Ink Diffusion Module. calligraphy have varieties of effects, this is closely related with the complexity in composition of paper and ink, Lee[3,8] has developed a cyclical sine model to simulate the distribution of fiber in paper through adjusting parameters of sine curve. Paper cells is connected by fiber, through which the ink is transfered.

Young et al.[9] simplifies the model of fiber, and makes a new model that divide paper into three layers, simulating ink transfer with changing the connectivity between paper cells.

While these interactive model have obtained good simulating results, but they are all highly complex in computing, which is not applicable on portable device with low computing capacity.

Therefore, this article focus on further improvement on existing paper-ink model, proposing an ink diffusion algorithm in 8 neighborhood displaying calligraphy through setting fiber structure in paper and viscosity of ink.

(1) Paper type. This article is based on improvement of Guo et al.[10], which divides paper into many cells. In every cells, there is specific number of fiber crossing points, and assuming it is *N(x,y)*. We can simulate different types of paper through defining the range of *N(x,y)*.

(2) Ink type. The diffusion of ink depends on not only paper structure but also ink attributes. Ink diffusion is a process involves the quantity of ink, the ink receptivity of paper and ink color.

a) Ink amount. Normally the ink amount is determined by the writing speed. The faster you write, the more the amount of ink exists, the greater the range of ink diffuses. Ink amount $W_0$:

$$w_0 = \alpha * v_i \tag{11}$$

Where $\alpha$ is the scale factor and $V_i$ is the writing speed.

b) Ink receptivity of paper cells

The absorbency of paper is related with the paper itself and ink viscosity. The ink viscosity μ shows the value of intermolecular friction when the ink is flowing. The greater the ink viscosity is, the slower the diffusion speed. Conversely, the feebler the ink viscosity is, the faster the diffusion speed. So the ink receptivity of paper is proportional to not only the numbers of fiber crossing points, but also the ink viscosity. Based on the analysis above, the ink receptivity $M_{xy}$ of the ($x$, $y$) can be represented as:

$$M_{xy} = \beta * N(x, y) * \mu \tag{12}$$

$\beta$ is the scale factor.

c) Ink color

In the process of ink diffusion, people will have a visual effect of halo, because the deposition of ink particles will create a filtration effect. The variable ink color effect can be expressed in flexible combination of R, G, B and Alpha. In this paper, we propose the effect of ink diffusion can be presented by adjusting the Alpha transparency, which means the less ink receptivity, the smaller Alpha, the stronger transparency, and the lighter ink color.

(3) Ink Diffusion Algorithm

This paper uses ink diffusion algorithm in eight neighborhood to obtain the ink information of paper cells in chirography edge, and achieve the ink diffusion effect by exchanging the ink amount of a point with its surrounding eight points. Fig.4 presents the process of ink diffusion.

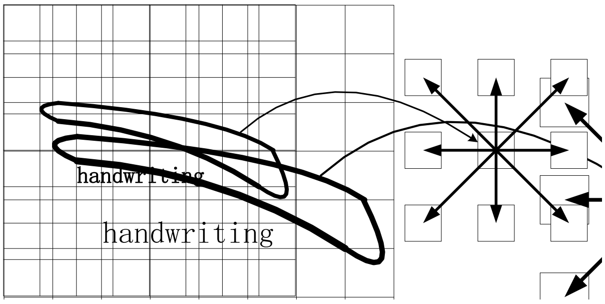


Fig 4. The process of ink diffusion

The specific algorithm is as follows:

**Step1.** After an action of writing, all the edges of brush model will be joined the “diffusion queue”.

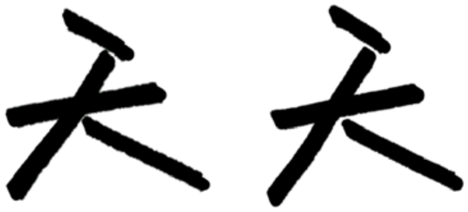
**Step2.** The first element in the “diffusion queue” which is not empty will be the current point $p$, otherwise the diffusion will be stopped.

**Step3.** If the ink amount of $p$ is more than the surrounding points $\{p_i, i=1,2\ldots8.\}$, the ink will be diffused to the surrounding eight points, or $p$ will stop diffusion and go step 2.

**Step4.** If the ink amount of $p$ is greater than the rest ink amount of others $\{p_i, i=1,2\ldots8.\}$ (the difference of the ink receptivity from the current ink amount), then coloring $p$, adding $p$ to the diffusion queue, and the ink amount of the new point is the amount of $p$ less the rest amount. Otherwise, the ink will be totally absorbed, the diffusion of the point will be stopped and go step 2.

## Implementation and Results

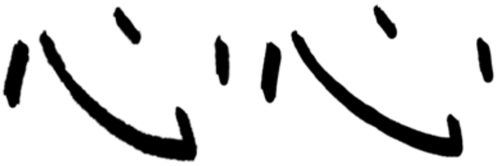
**Parameter Analysis.** The writing effects can be different through different parameter settings. Figure 5 shows the effects of different ink viscosity on the same paper structure. The range of the fiber crossing points is [1, 10], and the ink viscosities when writing the two characters are 0.4 and 0.8 respectively. As expected, the ink diffusion range is smaller with the increase of ink viscosity. This result is in line with the actual ink diffusion effect.



a $\mu$=0.4 b $\mu$=0.8

Fig. 5 Effect of different ink viscosity

The results of the same ink viscosity on different paper structures are shown in Fig. 6. The ink viscosity is 0.6, and the ranges of the fiber crossing points are [1, 10] and [1, 20] respectively. In line with the actual diffusion method, the fiber crossing points increases with smaller ink diffusion range.



a $N$=[1, 10] b $N$=[1, 20]

Fig. 6 Effect of different paper structures

**Experiment of Comparison.** The comparison of the experimental results between this study and the literature [4] is revealed in Fig. 7. The simulation of calligraphy in literature [4] was based on the force information which captured by a force sensitive handwriting device. The stroke thickness was determined by the force value as well as the center of gravity information. Nevertheless, writing force is unavailable through a common mobile device. The stroke thickness was stimulated according to the writing speed in our study. According to the comparison results, this system not only reached the realistic effect of calligraphy, but also improved the universality by eliminating the requirement of specific written equipments.

a the experimental results of literature [4] b the experimental results in this paper

Fig 7 the experimental results in this paper compared with that of literature [4].

**Time Consuming.** Under the condition of $N$=[1, 10] and $\mu$=0.6, 200 handwriting tasks were perform on Onda V975m and Teclast P11HD tablets respectively to measure the consuming time of this brush calligraphy system. The results are shown in table 1. The results show that the system is less time consuming, and thus satisfies the requirement of real-time interaction.

**Table 1 the test results of time consuming**

| name | parameter 1 | parameter 2 |
|---|---|---|
| model | ondaV975m | Teclast P11HD |
| CPU | 2GHz | 1.6GHz |
| DPI | 2048*1536 | 1920*1200 |
| p_num | 213 | 213 |
| a_num | 1762 | 1033 |
| Tr | 57ms | 21ms |
| Tc | 72ms | 94ms |
| Tt | 129ms | 115ms |

Note: p_num is the average number of collected points in the original trajectory, a_num is the points number after the interpolation, Tc is the average time of handwriting generation, Tr is the average time of ink diffusion, Tt is the average of the total time.

## Summary

Motivated by the previous studies of calligraphy simulation system, we designed a brush calligraphy system on mobile devices, which took practicability into account by generating writing trajectories through handwriting position and velocity information. After improving the paper and ink model of previous studies, an eight neighborhood ink diffusion algorithm was proposed to achieve calligraphy effects. The stimulation results indicated that this brush calligraphy system can reflect individual handwriting information of the writer. By leaving out the requirement for expensively specialized input equipment and the complex process of parameter setting, this system provided good user experience while writing. Our research work improved the universality of the brush calligraphy system and it will be valuable for similar research of brush calligraphy simulation on mobile devices.

## References

[1] S. Strassmann: Hairy Brushes. SIGGRAPH ’86 Proceedings. USA：ACM Press, Vol. 20(4): (1986), p. 225-232

[2] HTF. Wong, H.H.S. Ip: Virtual brush:a model-based synthesis of Chinese calligraphy. Computers & Graphics(S0097-8493).( 2000)

[3] J. Lee: Simulating oriental black-ink painting. IEEE Computer Graphics and Applications(S0272-1716), Vol. 19(3): (1999), p. 74-81

[4] X.F. Mi, M. Tang and J.Z. Lin, et al.: An experience based virtual brush model. journal of computer research and development, Vol. 40(8): (2003),p. 1244-1251

[5] S.H. Xu, M. Tang and F. Lau, et al.: A Solid Model Based Virtual Hairy Brush. Computer Graphics Forum(S0167-7055), Vol. 21(3): (2002),p. 299-308

[6] S. Jungpil, K. Suzuki: Interactive system for handwritten-style font generation. (2004)

[7] E.J. Jakubiak, R.N. Perry and S.F. Frisken: An improved representation for stroke-based fonts. Boston, Massachusetts : ACM.(2006)

[8] J Lee: Diffusion rendering of black ink paintings using new paper and ink models. Computers & Graphics (S0097-8493), Vol. 25(2): (2001) p.295-308

[9] J.Y. Young, D.H. Lee: Interactive rendering technique for realistic oriental painting. Journal of WSCG, Vol.11(1): (2003) p.538-545

[10] Q.L. Guo, L. Tosiyasu and Kunii: Modeling the diffuse paintings of ‘sumie’. Modeling in computer graphics(Proceedings of the IFIP WG5.10). Berlin:Springer, (1991), p.329-338