EUROGRAPHICS 2002 / G. Drettakis and H.-P. Seidel
(Guest Editors)

Volume 21 (*2002*), *Number 3*

A Solid Model Based Virtual Hairy Brush

Songhua Xu[1,2] Min Tang[1] Francis Lau[2] Yunhe Pan[1]

[1] CAD & CG State Key Lab of China, Zhejiang University, Hangzhou, China
[2] Department of Computer Science & Information Systems, The University of Hong Kong, Hong Kong

Abstract

We present the detailed modeling of the hairy brush used typically in Chinese calligraphy. The complex model, which includes also a model for the ink and the paper, covers the various stages of the brush going through a calligraphy process. The model relies on the concept of writing primitives, which are the smallest units of hair clusters, to reduce the load on the simulation. Each such primitive is constructed through the general sweeping operation in CAD and described by a NURBS surface. The writing primitives dynamically adjust themselves during the virtual writing process, leaving an imprint on the virtual paper as they move. The behavior of the brush is an aggregation of the behavior of all the writing primitives. A software system based on the model has been built and tested. Samples of imitation artwork from using the system were obtained and found to be nearly indistinguishable from the real artwork.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Methodology and Techniques]: Interaction techniques I.3.5 [Computational Geometry and Object Modeling]: Physically based modeling I.3.4 [Graphics Utilities]: Paint systems

1. Introduction

1.1. Previous Work

The problem of how to simulate Chinese calligraphy using the computer has attracted many researchers. First of all, there needs to be a way to represent characters in the computer, such as using cubic Bezier curves and straight lines[1, 2] or skeletal strokes[3]. Other methods focus on the brush stroke's boundary[4, 5, 6] and its trajectory[7]. Shamir and Rappoport introduced a parametric method to compactly represent existing outline-based oriental fonts[10]. Ip *et al.* discussed a method to encode Chinese calligraphic characters using automatic fractal shape coding[11]. Given the representation of a character, the technique of rasterization can be used to generate the image of the character, useful in applications such as desktop publishing[8].

Apart from representing and rasterizing existent fonts, there is the problem of generating new fonts, which appears to be more challenging and interesting[9]. New Chinese fonts have been generated through shape operations in an algebra of geometric shapes[12]. For centuries the Chinese have used the hairy brush to write characters because of the hairy brush's special aesthetic and expressive power. Thus providing an "e-hairy brush" becomes a meaningful pursuit for computer scientists, which is both theoretically challenging as well as of practical value. The paper by Henmi and Yoshikawa[29] describes one such virtual calligraphy system. A detailed analysis of the effects a virtual hairy brush can produce can be found in the paper by Strassmann[14]. Wong and Ip devised a virtual brush model for synthesizing Chinese calligraphic writings [13], in which the main working units are the cone and some ellipses. There are also hardware approaches to implementing e-hairy brushes, such as the one by Greene[15].

One ultimate goal of research on virtual hairy brushes is to be able to generate beautiful characters and realistic calligraphic artwork. The challenge lies not only in the modeling of the e-brush itself, but also the paper and ink model[16]. Such a model describes the interaction between the ink and the virtual paper. Some elaborate ink diffusion models have been proposed to simulate different ink spreading effects [17, 18]. Classical artificial intelligence, fuzzy logic, and knowledge engineering have been found to be useful in creating beautiful calligraphic artwork with a virtual hairy brush [26, 27, 28].

With a good e-hairy brush model and paper and ink model, beautiful paintings can be generated in the same way

as generating calligraphic artwork. Way and Shih[35] used a simple brush model to synthesize beautiful rock textures in Chinese landscape painting. The contours of the rocks and the areas to which textures are applied need to be manually supplied by the user using some existing image as reference. Many other papers have proposed a similar e-brush approach to tackle the problem of painterly rendering[19, 20, 21, 22, 23].

In general, simulating the hairy brush's different rendering effects falls into the research area of non-photorealistic rendering (NPR), of which a good survey can be found in the paper by Lansdown and Schofield[25]. Creating Chinese calligraphy or paintings by an e-hairy brush in real-time bears close resemblance to generating pen-and-ink illustrations in real-time in terms of both their goals and the problem-solving strategies[24]. However, the aesthetic effects pen and ink can produce can also be created with a fine virtual hairy brush; but not vice versa. The complexity of a virtual hairy brush is obviously way above that of a pen or pencil, which is commensurate with the brush's artistic expressive power. This paper describes a complex virtual hairy brush and its associated paper and ink model, and shows how, despite the complexity, the simulated brush can operate efficiently in real-time.

1.2. Contributions and Special Features

Our method comprises a collection of algorithms to simulate the change of the physical conditions of a hairy brush, including the brush's geometric shape and its ink-related properties during the Chinese calligraphy writing process, where ink-related properties refer to the brush's degree of wetness and color. An interactive software system implementing these algorithms has been constructed which can be used to create calligraphic artwork fully electronically. The ease of use and expressive power of our virtual brush, and the high quality outputs from using the brush are evidences for the effectiveness of our approach.

Pure hardware approaches such as Greene's[15] tend to be expensive and not generally applicable. The method by Pan *et al.*[12] can only generate new fonts from existing fonts, but not user's handwritings into new fonts. The method by Way and Shih[35] requires the user to specify the contour and the parameters of the object to be painted based on a reference image or figure. This painting procedure is very different from the way traditional artists perform their painting tasks. In contrast, the procedure of using our virtual hairy brush to write or paint is exactly the same as the traditional method. Our method therefore should appear to be most natural to those non-computer specialists. In the paper by Baxter *et al.*[20], the brush head is modeled as a subdivision surface mesh wrapping a spring-mass particle system skeleton. The particle system produces the basic motion and behavior of the brush head, with the deformable mesh skin around the skeleton representing the actual shape of the head. This 3D brush model can capture the essential quality of the physical brush for creating oil or watercolor paintings. Such a simple brush model, however, cannot capture all the complex physical properties and conditions of a hairy brush and its ink distribution which need to be there for simulating Chinese calligraphy and painting.

The virtual brush model by Wong and Ip[13], which is closest in the modeling approach and functionalities to our model, offers a feasible means for artists to produce Chinese calligraphic characters electronically. But their method is not all that convenient to use because a complex set of interrelated parameters to control the shape, density and opacity of the current drawing mark need to be specified by user manually, thus making partially interactive at best. More specifically, the user needs to specify the profiles for seven parameters to control the brush motion dynamic model, and another three parameters to control the ink deposition, a total of ten profiles. Only two of these profiles, the stroke's trajectory in X and Y direction, can be sampled by a mouse or a digitizing pen. So there are eight profiles to be input manually. Whereas in our approach, there are many fine devices that can take in the six degrees of freedom of a solid object, and no profile needs to be input by the user. Other than the input parameters, several parameters have been introduced for controlling the virtual brush's quality, whose values can be automatically set via an optimization algorithm. Although our e-brush model is more complex than that of Wong and Ip, all the modeling details and quality parameters of the e-brush can be treated as a black box by the end user. The user can in fact use the e-brush in the same way he/she uses the traditional hairy brush. In this sense, our e-brush is simpler and easier to manipulate than the e-brush based on Wong's and Ip's approach [13]. Most importantly, because of the more complex model we use and the automatically tunable quality parameters we introduce, we can achieve better output results with our e-brush.

Our virtual hairy brush is closer to the real brush than other similar brushes because our e-brush can automatically determine both the geometric contour and the texture of its instant drawing mark on the virtual paper at the same time. This process is real-time and no human intervention is necessary, which is not the case in Wong's and Ip's system[13]. Our virtual hairy brush is genuinely a fully interactive system. Moreover, Wong's and Ip's brush is limited in its artistic expressive capabilities because its current drawing mark is always an ellipse. The drawing mark in our model can be varied and of irregular shape, being generated from any planar parametric curve instead of just an ellipse. This is a vital feature for quality enhancement in an artist's work. The delicate artwork we generated attests to the expressive power of our approach with this feature.

We introduce a useful concept called *writing primitives* which are hair clusters as the basic working units of the virtual hairy brush. This makes real-time simulation of the hairy brush's writing and painting behavior possible. Be-

cause the cross section of each writing primitive intersecting the virtual paper plane, and its ink-related information are computed only once, our method can be much faster than that of Wong and Ip[13] which is based on cones and operates on each hair individually. We have implemented a prototype system, with experimental results demonstrating the correctness and real-time quality of our algorithms.

By embedding ink-related information in the writing primitive's control axis, a single primitive can readily express reasonably complex, interesting, and even mysterious distribution of the ink, including its color and wetness. In our proposed ink model, we use probability to create realistic effects to be used in rendering the current ink-mark, thus enabling our virtual hairy brush to simulate the drying and running effects of calligraphic artwork. Multiple gray levels, full-color paintings, dry brush writing effects, and saturation effects can be produced using this new ink model. All these writing effects are part of the expressive power of the system for computer-aided art creation.

We introduce an inertia predictor to calculate the brush's virtual position based on its sampled position during the writing process. This inertia predictor simulates the acceleration of the virtual hairy brush, which gives the user a feeling that mimics the real brush, and helps to produce output that rivals real artwork. In addition to ways provided by our system to edit various quality parameters of the virtual hairy brush manually by the user, special optimization algorithms are built into the system that can customize these parameters for the user automatically, leading to better maneuverability of the brush and improved quality in the output.

According to the six degrees of freedom of the hairy brush, which are sampled periodically, the computer can simulate the whole Chinese calligraphy process with high accuracy. This can be done in real time, as proven through experimentation. Since many of the geometrical and dynamic parameters of the brush can be automatically determined by the system, it is not necessary to store any bitmap image during the writing process, and thus the storage space required is minimal.

2. Writing Primitives

We rely on the concept of *writing primitive*, which represents a hair cluster (*i.e.*, a small bundle of hair clustering together), to reduce the complexity of the modeling and thus the computational requirement for the implementation. A virtual hairy brush consists of one or more writing primitives. Each writing primitive is described by a NURBS surface, and is constructed through the general sweeping operation in CAD. The behavior of the virtual hairy brush is an aggregation of the behavior of all the writing primitives. This is in sharp contrast with the approach used in by Wong and Ip[13], where every hair is operated on. The use of writing primitives does not diminish in any way the power of the virtual hairy brush in satisfactorily simulating all possible behaviors of a real hairy brush including the branching out behavior. The concept of writing primitives is a natural one in view of the coherence between neighboring hairs and their proximity in ink-related properties. Our experimental results have confirmed this, as well as the expressive power of our model. The artwork created using our system can be seen as better than those in the paper by Wong and Ip.

A writing primitive in the model is defined by its four "characters" (Figure 1). Based on these four characters, the model is constructed through the general sweeping operation in CAD. This operation will construct a NURBS sweeping surface by taking the curve of the writing primitive's middle control axis as its sweeping trajectory and the cross sections defined by the user in advance as the given profiles. The general sweeping operation is implemented according to the minimized rotation frame algorithm by Maurer and Juttler[33]. During the simulation, three of the four characters (not including the bottom control circle) of the writing primitive will be dynamically adjusted according to the input data which describe the brush's current position in the 3D space. The bottom control circle never changes during the writing process.

All the input data are preprocessed to take into account the inertia of the hairy brush, which creates a realistic "feel" of the brush. The cross section of the intersection between a writing primitive and the virtual paper plane is called the writing primitive's current drawing mark. For every time slice of the writing process, ink will be deposited according to the current drawing mark. The union of the current drawing marks of all the writing primitives is the current mark made by the brush on the paper. The final artwork is the accumulation of such marks over all the time slices. Figure 2 shows the state transitions of writing primitives. The virtual hairy brush's working diagram is shown in Figure 4. It will be explained in detail in Sections 4-6.

Section 3 explains the solid model of the virtual hairy brush. Section 4 discusses the sampling and processing of the brush's input. Section 5 discusses how to adjust the parametric models of the e-brush dynamically. Section 6 presents the rendering of the brush's current ink mark at any time instant. Section 7 discusses how the brush and its quality parameters are configured.

3. The Model and the States

3.1. The parametric model

Our model of a virtual hairy brush (**HB**) is in terms of the collection of writing primitives that the brush is composed of. These writing primitives are denoted as $\mathbf{WP}_1, \mathbf{WP}_2, \cdots, \mathbf{WP}_m$. The number of **HB**'s compositive writing primitives will be dynamically adjusted by the system automatically. There are several parameters to control

the brush-generated artwork style, the values of which are within $[0,1]$.

To generate the parametric model of a writing primitive $\mathbf{WP}_i$ $(i = 1,2,\cdots,m)$ through the general sweeping operation, a circle $\mathbf{C}_i$, an ellipse $\mathbf{E}_i$, and a line $\mathbf{L}_i$ are taken as the sweeping profiles, and an axis $\mathbf{A}_i$ is used as the sweeping trajectory. $\mathbf{C}_i$, $\mathbf{E}_i$, $\mathbf{L}_i$, $\mathbf{A}_i$ are called $\mathbf{WP}_i$'s bottom control circle, middle control ellipse, tip control line, and middle control axis, respectively (Figure 1). Among these four characters, $\mathbf{C}_i$ is a static character of $\mathbf{WP}_i$, that is, once $\mathbf{C}_i$ is initialized at the beginning, it remains unchanged during the whole process. In fact, all the writing primitives of a brush share the same bottom control circle, and so this circle is also called the bottom control circle of the brush, denoted as **HB.C**.

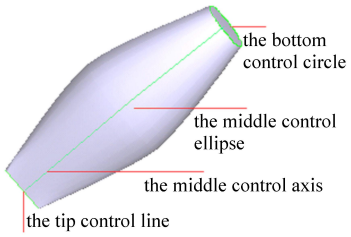

Figure 1: *A writing primitive and its four compositive characters.*

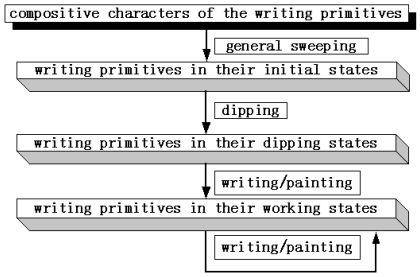

Figure 2: *State transitions of a writing primitive.*

The middle control axis $\mathbf{A}_i$ is a cubic B-spline curve with interpolated key points $\mathbf{P}_{i,1},\mathbf{P}_{i,2},\cdots,\mathbf{P}_{i,n_i}$. Each of these points carries both geometric and ink-related information, including $\mathbf{P}_{i,j}$'s color in RGB and its degree of wetness. For each $\mathbf{A}_i$, its first key point is always set at the center of the bottom control circle, and its last key point at the midpoint of the tip control line. Each key point carries four fields to specify ink-related information: *vector mode* wetness changing vector $\mathbf{wv}_{i,j}$, *vector mode* color changing vector $\mathbf{cv}_{i,j}$, *radiation mode* wetness changing factor $wr_{i,j}$, and *radiation mode* color changing factor $cr_{i,j}$. Figure 3 explains these two modes of ink distribution. Based on these four fields, we can compute the ink-related information of any point $\mathbf{Q}$ on the plane that is perpendicular to the middle control axis $\mathbf{A}_i$ passing through $\mathbf{A}_i$'s key point $\mathbf{P}_{i,j}$ by using the following formulation.

$$
\begin{aligned}
wet_{\mathbf{Q}} &= wet_{\mathbf{P}_{i,j}} \times \\
&\quad (1+\|\mathbf{cc}_{\mathbf{Q}}-\mathbf{cc}_{\mathbf{P}_{i,j}}\| \times wr_{i,j} + (\mathbf{cc}_{\mathbf{Q}}-\mathbf{cc}_{\mathbf{P}_{i,j}}) \bullet \mathbf{wv}_{i,j}) \\
\mathbf{col}_{\mathbf{Q}} &= \mathbf{col}_{\mathbf{P}_{i,j}} \times \\
&\quad (1+\|\mathbf{cc}_{\mathbf{Q}}-\mathbf{cc}_{\mathbf{P}_{i,j}}\| \times cr_{i,j} + (\mathbf{cc}_{\mathbf{Q}}-\mathbf{cc}_{\mathbf{P}_{i,j}}) \bullet \mathbf{cv}_{i,j})
\end{aligned}
$$

where $\mathbf{cc}_{\mathbf{Q}}$, $wet_{\mathbf{Q}}$ and $\mathbf{col}_{\mathbf{Q}}$ are point $\mathbf{Q}$'s current coordinates in the 3D space, its degree of wetness, and its color respectively; and $\mathbf{cc}_{\mathbf{P}_{i,j}}$, $wet_{\mathbf{P}_{i,j}}$ and $\mathbf{col}_{\mathbf{P}_{i,j}}$ are point $\mathbf{P}_{i,j}$'s corresponding values.

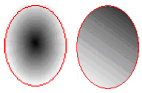

Figure 3: *Radiation (left) and vector (right) ink distribution modes.*

3.2. The three states of a brush

A virtual hairy brush **HB** is assumed to have three possible states in its whole lifecycle: the initial state, the dipping state, and the working state (Figure 2). The initial state of a virtual hairy brush **HB** is when all of its compositive writing primitives are in their free states. The three dynamic compositive characters of a writing primitive are simplest when in this state: the tip control line, the middle control axis and the middle control ellipse are reduced to a point, a straight line, and a circle, respectively. Varying the radius of the circle and its location can result in a series of modeling effects. All the writing primitives of the virtual hairy brush will shift to the dipping state after the brush is dipped into the ink bottle and before touching the paper. By dipping, writing primitives acquire ink-related information, which includes color and degree of wetness according to how the brush is dipped.

If $\mathbf{P}_{i,k},\mathbf{P}_{i,k+1},\cdots,\mathbf{P}_{i,n_i}$ are $\mathbf{WP}_i$'s key points that are soaked in ink, their color is simply set to the ink color. For $\mathbf{P}_{i,l}$, $l = 2,3,\cdots,k-1$, which are the key points not currently immersed in ink, linear interpolations are applied to compute their individual colors $\mathbf{col}_{\mathbf{P}_{i,l}}$ with the assumption that $\mathbf{col}_{\mathbf{P}_{i,1}} \equiv \mathbf{0}$, which is the color code for pure white. Similarly for each key point's degree of wetness. We assume $wet_{\mathbf{P}_{i,1}} \equiv \mathbf{0}$, meaning that $\mathbf{P}_{i,1}$ is all dry.

The working state of a virtual hairy brush is the "deformation" state of the brush. The brush deforms due to touching or pressing against the paper. By varying the eccentricity of the middle control ellipse, the tip control line and the middle control axis, a series of modeling effects can be achieved.

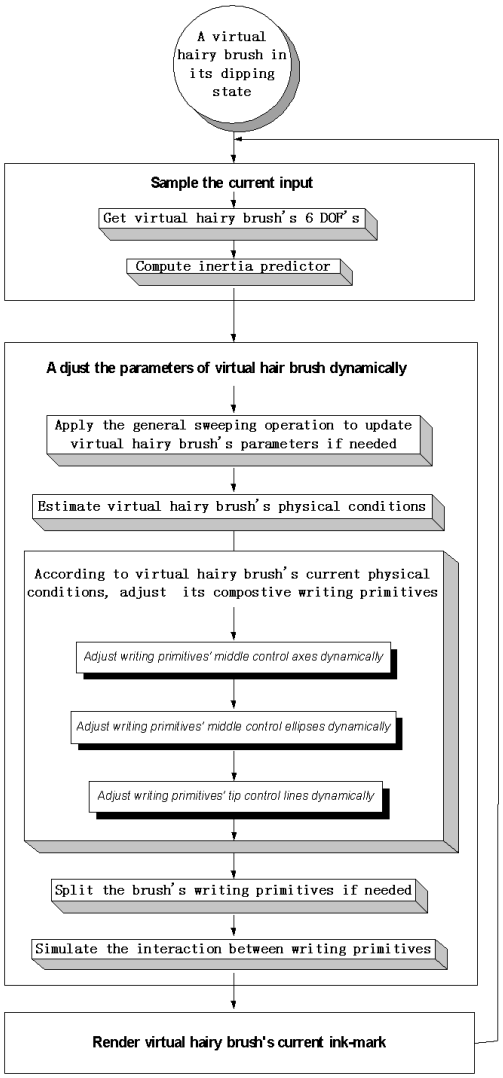

Figure 4: *Virtual hairy brush's working diagram.*

4. Sampling of the Input Data

Sampled input data are used to adjust the brush dynamically, while maintaining the validity (with respect to a real brush) of the parametric model at all times. We need to first obtain the brush's six degrees of freedom. Assume that the sampled datum at time t is $\mathbf{Sam}^t = (x^t, y^t, z^t, d^t, q^t, r^t)$, where (x^t, y^t, z^t) is the center of **HB**'s bottom control circle **HB.C** in the 3D space; d^t the direction to which **HB** tilts; q^t the degree of **HB** tilting, and r^t the degree that **HB** has rotated. There are many ways to input a solid object's six degrees of freedom, such as using some special device [20]. A three dimensional mouse, or a data glove or any other sensor that can take in the six degrees of freedom for a rigid body can also be used as the input hardware for the virtual hairy brush. Some other interesting methods have been introduced [30, 31]. Our current method for input is based on a 3-button mouse. During the writing process, **HB**'s virtual position at time t, $\mathbf{S}^t = (sx^t, sy^t, sz^t, sd^t, sq^t, sr^t)$, is computed from the current sampled datum $\mathbf{Sam}^t$ and an inertia predictor $\mathbf{M}^t = (mx^t, my^t, mz^t, md^t, mq^t, mr^t)$, using the following formula.

$$\mathbf{S}^t = wei_{sam} \times \mathbf{Sam}^t + (1 - wei_{sam}) \times \mathbf{M}^t$$

$\mathbf{M}^t$ comes from **HB**'s displacements in its last few sampling intervals $\mathbf{S}^{t-1}$,$\mathbf{S}^{t-2}$,$\cdots$,$\mathbf{S}^{t-n}$:

$$\mathbf{M}^t = \mathbf{Vel}^t \times dT + \mathbf{S}^{t-1}$$

$$\mathbf{Vel}^t = \frac{\sum_{k=1}^{el}(wei_k \times \frac{\mathbf{S}^{t-k} - \mathbf{S}^{t-k-1}}{dT})}{\sum_{k=1}^{el} wei_k}$$

where $\mathbf{Vel}^t$ is the weighted sum to estimate **HB**'s velocity, dT is the time interval between every two adjacent sampling times, and el the estimation length. This simple estimation method turns out to be reasonable since the speed of the brush in real life rarely changes too abruptly.

By introducing an inertia predictor to influence the sampled position of the brush to yield its virtual position, end users can emulate the effect and enjoy the feeling of brush gliding. For the above formulation, the default parameter configuration is: $el = 4$; $wei_1 = 4$, $wei_2 = 2$, $wei_3 = 1$, $wei_4 = 1$; $wei_{sam} = 0.6$. These default values were obtained by experiments. Changing these values can yield different writing feelings.

5. Dynamic Adjustments of the Brush

5.1. Estimating the physical conditions of the brush

Based on the sampled input data, the physical conditions of the virtual hairy brush are estimated at every time instant, which include the writing primitive's inner stress str_i^t and the pressure on the primitive due to its interaction with the paper. These conditions are needed for dynamically updating the parametric models of the e-brush.

From intuition, the rigidity of $\mathbf{WP}_i$'s hair, $\mathbf{WP}_i$'s historical deformation, the wetness of $\mathbf{WP}_i$, and the size of the part of $\mathbf{WP}_i$ that is against the virtual paper plane ψ all affect the value of str_i^t. We devised accordingly a formula to estimate str_i^t based on these factors. Under most circumstances, we found this formula to be a good approximation of the behavior of a real brush. In comparison with the classical formulation in solid mechanics[34], this approximation formulation is simpler, which is helpful in reducing the computation time.

Similarly, a formulation to estimate $\mathbf{WP}_i$'s intensity of pressure γ_i^t due to the interaction between $\mathbf{WP}_i$ and the virtual paper ψ was derived. It can be easily seen that fast movements of the virtual hairy brush, a high degree of inner stress

of the current writing primitive $\mathbf{WP}_i$ and a coarse virtual paper can all severely deform **HB** (*i.e.*, its compositive writing primitive's middle control axes). Our formulation accounts for this.

5.2. Dynamic adjustment of the middle control axis

5.2.1. The current active point

During the writing process, the crossing of the middle control axis $\mathbf{A}_i$ and the virtual paper ψ is the current active point. The current active point will be inserted into $\mathbf{A}_i$'s series of key points dynamically. The point's ink-related information is computed from the neighboring points in $\mathbf{A}_i$ by linear interpolation.

5.2.2. Deformation of the middle control axis $\mathbf{A}_i$

The middle control axis $\mathbf{A}_i$ of $\mathbf{WP}_i$ will be deformed under outer forces. A local reference frame is set up by taking $\mathbf{WP}_i$'s bottom circle $\mathbf{C}_i$ as the X-Y plane, its center $\mathbf{cen}_i$ as the origin, and the brush shaft's direction as the *Z*-axis (Figure 5). If the current active point $\mathbf{P}_{i,j}$ travels a certain distance in the reference frame during time slice *t*, then all the key points that are underneath the virtual paper ψ will also travel the same distance. Thus, the whole middle control axis is deformed. In the virtual hairy brush, we estimate this distance to be the product of the **HB**'s degree of elasticity and $\mathbf{WP}_i$'s current intensity of pressure.

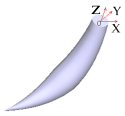

Figure 5: *A deformed writing primitive.*

5.2.3. Recovery of the middle control axis $\mathbf{A}_i$

As an elastomer, a writing primitive $\mathbf{WP}_i$ will recover in some fashion after its deformation once the outer force exerted on it is released. So each time when the virtual hairy brush **HB** is lifted, every key point on each of **HB**'s compositive writing primitives will change its place in the 3D space, that is, all the key points' *z* components will increase by a certain amount. The higher the virtual brush is lifted, the more intense the current writing primitive's inner stress would be, and the recovery would be more substantial.

5.2.4. Dynamic adjustment of the wetness

During the writing process, the color of **HB** is assumed to be constant. The only changing ink-related information is **HB**'s degree of wetness. We assume the part of the middle control axis between the key point $\mathbf{P}_{i,j}$ and its adjacent key point $\mathbf{P}_{i,j+1}$ would intersect with the virtual paper plane ψ. The degree of wetness of these two key points will decrease by a certain amount, which is estimated to be the product of ψ's ink absorbing ability *ab* and $\mathbf{WP}_i$'s current intensity γ_i^t against ψ.

5.3. Dynamic adjustment of the middle control ellipse

We simulate the orientation of a deformed virtual hairy brush by introducing an approximated orientation adjustment formulation. This is based on the assumption that if **HB**'s moving direction is the same as the orientation of writing primitive $\mathbf{WP}_i$'s minor axis, further writing movements will only increase the length of $\mathbf{WP}_i$'s major axis. The increased amount is proportional to the product of $\mathbf{WP}_i$'s current intensity γ_i^t against ψ and the sine of the angle between the direction of the writing movement and the orientation of $\mathbf{WP}_i$'s major axis. If the moving direction does not coincide with the orientation, the ellipse will also be rotated by a certain angle, which is proportional to the product of γ_i^t and the cosine of the above angle. Since writing primitive $\mathbf{WP}_i$'s number of hair, approximated by $\mathbf{E}_i$'s area, is a constant if $\mathbf{WP}_i$ doesn't split during its writing process, $\mathbf{E}_i$'s minor axis' length can be determined given a certain value for its major axis' length.

The middle control ellipse's position relative to the middle control axis also varies during the writing process in order to reflect the deformation of the writing primitive. We assume that the ideal location of the middle control ellipse should be such that it divides all the key points on the middle control axis into two equal groups, because at this location the ellipse's profile has the maximum capacity to control the writing primitive's geometric modeling characters. Since the speed in which the ellipse can move is limited by the mechanical and flowage properties of the virtual hairy brush, the actual position of the middle control axis is a linear interpolation of its ideal position (based on the ellipse's ideal location) and its previous position.

5.4. Dynamic adjustment of the tip control line

The tip control line of a writing primitive $\mathbf{WP}_i$ is but a single point in its initial state. It changes into a real line during writing. The line's elongation and rotation are simulated by employing the same strategy as is applied to the middle control ellipse; that is, the tip control line $\mathbf{L}_i$ will increase in length and be rotated by the same amount as the major axis of the middle control ellipse.

5.5. Splitting of the virtual hairy brush

There is a threshold *tre* which specifies the extent to which $\mathbf{WP}_i$ can be deformed. When this threshold is reached or exceeded, the current writing primitive will split into several smaller writing primitives. This simulates the "branching out" behavior of the virtual hairy brush during the writing process. Specifically, if writing primitive $\mathbf{WP}_i$'s current

inner stress str_i^t becomes greater than *tre*, $\mathbf{WP}_i$ will split into $k = \lfloor \frac{str_i^t}{tre} \rfloor$ new writing primitives $\mathbf{WP}_i^1, \mathbf{WP}_i^2, \cdots, \mathbf{WP}_i^k$.

Each of the new writing primitives $\mathbf{WP}_i^j$ $(j = 1, \cdots, k)$ has a number of hair which is equal to $\frac{1}{k}$ of $\mathbf{WP}_i$'s total number. Note that in the virtual hairy brush, we use the area of the middle control ellipse and the length of the tip control line to compute the number of hair. Therefore the lengths of the middle control ellipse's major and minor axes of each of the new writing primitives from $\mathbf{WP}_i$'s splitting are set to $\frac{1}{\sqrt{k}}$ of $\mathbf{WP}_i$'s original values. And the tip control line's length is set to $\frac{1}{k}$ of the original value.

At the beginning, the virtual hairy brush **HB** contains only one writing primitive. During the writing process, the number of **HB**'s compositive writing primitives may be increased because of the split operation. A brush with one writing primitive is probably enough for official scripts, but for cursive scripts, at least a dozen of primitives would be necessary.

During the split operation, every new writing primitive $\mathbf{WP}_i^j$ $(j = 1, \cdots, k)$ generated has the same number of key points as the original $\mathbf{WP}_i$, with coordinates at a certain distance from $\mathbf{WP}_i$'s. This distance is proportional to the amount of $\mathbf{WP}_i$'s current inner stress exceeding the split threshold *tre*, and the direction of this distance is assumed random.

5.6. Ink flowage between writing primitives

Although each primitive has full control over its behavior during the writing process, due to reciprocity in mechanics and ink's flowage, there could be interaction between writing primitives that are close to each other. To simulate this interaction, we allow each key point's degree of wetness to be affected by its neighboring key points if the distance separating them is within the ink diffusion distance factor η. Linear interpolation is used to compute one key point's current degree of wetness based on its previous value and the average degree of wetness of those neighboring key points.

6. The Writing Process

At time t, each of **HB**'s writing primitives will intersect with the virtual paper plane to yield a cross section. The drawing operations are executed taking into account the writing primitives' ink-related information. The following outlines the algorithm for this process—the virtual hairy brush's real-time writing/painting algorithm.

1. A writing primitive $\mathbf{WP}_u$ constructed by the general sweeping operation is given. The generated sweeping surface is a NURBS surface denoted as $SS_u(s,t)$; s and t are the parameters of this parameterized surface, whose values are within $[0,1]$. The direction of s is the same as that of $\mathbf{WP}_u$'s middle control axis.

   Actions of the algorithm: Intersect $SS_u(s,t)$ with the virtual paper plane ψ to get an intersecting curve cur_u that encloses an area which is the current ink mark (Figure 6). Project each point $\mathbf{V}_{u,v}$ in cur_u onto $\mathbf{WP}_u$'s middle control axis $\mathbf{A}_u$ to obtain the point $\hat{\mathbf{V}}_{u,v}$. Find the two nearest key points on $\mathbf{A}_u$ relative to $\hat{\mathbf{V}}_{u,v}$, and based on the ink-related information of these two points, $\hat{\mathbf{V}}_{u,v}$'s ink-related information is computed by linear interpolation. $\mathbf{V}_{u,v}$'s ink-related information is derived from $\hat{\mathbf{V}}_{u,v}$'s by following $\hat{\mathbf{V}}_{u,v}$'s texture function.

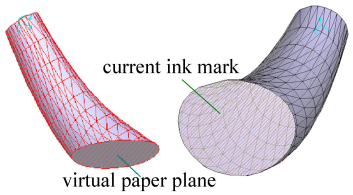

Figure 6: *The virtual paper plane and its drawing mark.*

2. The new degree of wetness of pixel $\lambda_{u,v}$ on ψ which coincides with $\mathbf{V}_{u,v}$ is linear-interpolated from $\lambda_{u,v}$'s previous value with $\mathbf{V}_{u,v}$'s degree of wetness by using ψ's absorbing ability factor as the interpolation weight. A drying factor is introduced to automatically reduce each pixel's degree of wetness periodically until it reaches 0.
   If the degree of wetness exceeds the upper bound of virtual paper ψ's degree of wetness, saturation takes place. We assume that saturation would only affect $\mathbf{V}_{u,v}$'s eight neighboring pixels on ψ. The degree of wetness of a pixel will increase if it is not saturated. The increased amount is proportional to the unsaturated degree. That is, $\mathbf{V}_{i,j}$, which is one of $\mathbf{V}_{u,v}$'s eight neighboring pixels on ψ, will have its degree of wetness increased by the following amount:

$$\frac{(1 - wet_{\mathbf{V}_{i,j}}) \times tr(1 - wet_{\mathbf{V}_{i,j}})}{facw_{u,v}} \times (wet_{\mathbf{V}_{u,v}} - 1) \times tr(wet_{\mathbf{V}_{u,v}} - 1)$$

   where $tr(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$, $facw_{u,v} = \sum_{g=u-1}^{u+1} \sum_{h=v-1}^{v+1} (1 - wet_{\mathbf{V}_{g,h}}) \times tr(1 - wet_{\mathbf{V}_{g,h}})$, and $wet_{\mathbf{V}}$, a real number between 0 and 1, is point $\mathbf{V}$'s degree of wetness.
3. To render the current ink-mark, a new ink model is proposed. The current color of the pixel $\lambda_{u,v}$ is the linear interpolation on $\lambda_{u,v}$'s previous color and $\mathbf{V}_{u,v}$'color. And the interpolation weight is a random number, which has probability of $min(ren \times ab \times \gamma_i^t, 1)$ to be $wet_{\lambda_{u,v}}^t$ and probability of $1 - min(ren \times ab \times \gamma_i^t, 1)$ to be zero; *ren* is **HB**'s color render control factor, *ab* is the virtual paper ψ's absorbing ability, and γ_i^t is the intensity of the interaction between the writing primitive and the virtual paper plane.

We use this formulation to simulate the dry brush drawing effect and the running style effect.

7. Customizing the Brush

7.1. Quality parameters

In real life, brushes having soft hair tend to branch out easily during writing. Some brushes have a good deal of hair and tend to suck in more ink and cause serious saturation during the writing process, while other brushes have rather long hairs and their tip tends to get deformed and rotated to a great extent easily. For our virtual hairy brush, a number of quality parameters can be set to simulate these different kinds of brush. The final electronic artwork generated by the virtual hairy brush can be much affected by the values of these parameters. Similarly, a set of quality parameters can be applied to the virtual paper to simulate a certain kind of paper.

To ease the task of selecting the values to achieve some desired quality, our implemented system offers a set of predefined quality configurations for the user to choose. The system maintains a library of quality configurations, some of which could have been contributed by the users themselves. Of course the end users can configure their favorite virtual hairy brush by assigning values to the quality parameters by themselves. The implemented system provides a window in which the users can adjust these parameters visually. The system can also configure a brush automatically, to be explained next.

7.2. Configuring the brush with machine intelligence

To enable the computer to adjust the quality parameters for the virtual hairy brush as well as those for the virtual paper, users need to carry out a special procedure to train the computer. They would produce several brush strokes, whose boundaries are specified by the machine-training module. This procedure is similar to a beginner starting to learn how to use hairy brushes in real life, referred to as the "Miao-Hong" process in Chinese calligraphy. The number of training samples can be set by users. Of course, the more samples used the better would be the resulting quality of the brush. Simple artwork such as the one in Figure 7 took several minutes for the training.

Suppose there are n training samples, $S^i[len][wid]$, and the corresponding user-created results are $C^i[len][wid] (i = 1, 2, \cdots, n)$, where $S^i[len][wid]$ and $C^i[len][wid]$ are two matrices with each of the elements being the RGB color code of a pixel on the virtual paper ψ. We defined a target function ϑ to indicate the difference between the system's specified training samples and the user-created images with the virtual hairy brush:

$$\vartheta = \sum_{i=1}^{n} \vartheta^i = \sum_{i=1}^{n} \sum_{p=1}^{len} \sum_{q=1}^{wid} (S^i[p][q] \bigotimes C^i[p][q])$$

Here the operator $\bigotimes$ is defined as: $(r_1, g_1, b_1) \bigotimes (r_2, g_2, b_2) = |r_1 - r_2| + |g_1 - g_2| + |b_1 - b_2|$. All the quality parameters of the virtual hairy brush contribute to ϑ. With a certain set of initial values for all the quality parameters, a corresponding ϑ can be computed. Thus the problem to determine a proper set of values to configure all the virtual hairy brush's quality parameters is reduced to the problem of finding a configuration which can minimize or nearly minimize ϑ, which is a typical optimization problem. The Steepest Descent Algorithm[32] in nonlinear programming can give a satisfactory solution.

8. System Implementation and Experiment Results

Please refer to the figure in the color section showing a screen shot of the implemented system, where there is one window responsible for displaying the current sampled input; one for the current control parameters derived from the input; one for parameter values automatically assigned to the writing primitives by an internal algorithm; one for the current drawing mark; one for the current writing primitives in the 3D space; and one last one for the current writing mark imprinted on the paper and the history of all the sampled input. Figures 7–12 show some real artwork digitized from calligraphy samples together with the artwork created by the virtual hairy brush. These samples prove that very realistic-looking calligraphic artwork can be generated by our virtual hairy brush. If carefully tuned, the simulation's result can be nearly indistinguishable from the original one to a human viewer. By fiddling with the quality parameters and the input data for the brush's six degrees of freedom, users can create interesting calligraphy fully electronically. It is well known that to imitate an original calligraphic artwork is a nontrivial task. With our electronic environment, however, such an imitation task becomes relatively easy. The system offers a friendly user interface, which supports adjusting the quality parameters and the input data dynamically to achieve whatever delicate effect the user desires. Figures 13 and 14 and those in the color section show a series of computer artwork created by the first author, a non-artist, using the virtual hairy brush. To give an idea on the effort required for the more complex artwork, the horse picture took about 1.5 hours to complete.

Figure 7: *Real artwork (left) vs. imitation created by the virtual hairy brush (right).*

Figure 8: *On the left is the real artwork; the middle one was created by the virtual hairy brush, and the same on the right with the brush's trajectories highlighted.*

Figure 9: *The same character as in the previous figure; real artwork (left), and imitation by the virtual hairy brush (right).*

9. Future Work

A real brush operates in a fashion that is orders of magnitude more complex than can be modeled by a computer. Features that can be added in the next version to further enhance the modeling includes the merging of writing primitives, repeated dipping effects, more user's control of the brush during writing, etc. Other features requiring longer-term effort include mapping of 2D calligraphic artwork to 3D (to imitate for example chiseling in monuments), and feature extraction of real artwork and applying the extracted features to automatic electronic art creation.

References

1. Y. Chua, Bezier brush strokes, *Comput. Aided Design*, **22**(9):550–555, 1990.
2. T. Nishita, S. Takita, and E. Nakamae, A display algorithm of brush strokes using Bezier functions, *Proc. of CGI '93*, 244–257, 1993.
3. S. Hsu and I. Lee, Drawing and animation using skeletal strokes, *Proc. of SIGGRAPH '94*, 109–118, 1994.
4. J.-W. Ahn, M.-S. Kim, and S.-B. Lim, Approximate general sweep boundary of a 2D curved object, *Comput. Vision, Graph. & Image Proc.*, **55**(2): 98–128, 1993.
5. K.C. Posch and W.D. Fellner, The circle-brush algorithm, *ACM Transactions on Graphics*, **8**(1):1–24, 1989.
6. S.-B. Lim and M.-S. Kim, Oriental character font design by a structured composition of stroke elements, *Comput. Aided Design*, **27**(3):193–207, 1995.
7. J.D. Hobby, Digitized Brush Trajectories. Ph.D. thesis, Stanford University, 1985. (Also Technical Report STAN-CS-85-1070, Stanford University, 1985.)
8. L. Hao and H. Zhou, A new contour fill algorithm for outlined character image generation, *Comput. & Graphics*, **19**(4):551–556, 1995.

Figure 10: *Real artwork (left), and imitation by the virtual hairy brush (right).*

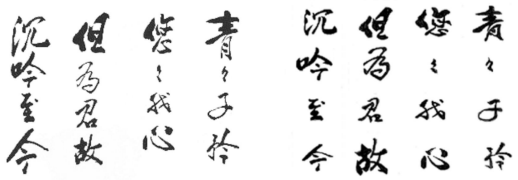

Figure 11: *Real artwork (left) vs. artwork created by the virtual hairy brush (right).*

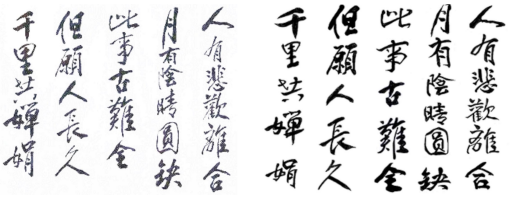

Figure 12: *Real artwork (left) vs. artwork created by the virtual hairy brush (right).*

Figure 13: *Pinasters using the virtual hairy brush.*

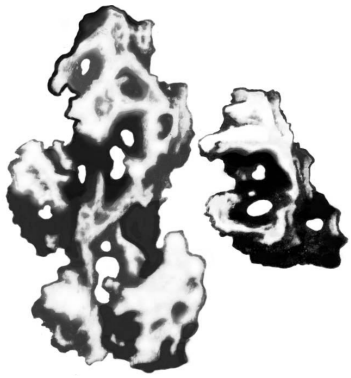

Figure 14: *Stones using the virtual hairy brush.*

9. P. Coueignoux, Character generation by computer, *Comput. Graph. & Image Proc.*, **16**(3):240–269, 1981.
10. R. Shamir and A. Rappoport, Quality enhancements of digital outline fonts, *Comput. & Graphics*, **21**(6):713–725, 1997.
11. H.H.S. Ip, H.T.F. Wong, and F.Y. Mong, Fractal coding of Chinese scalable calligraphic fonts, *Comput. & Graphics*, **18**(3):343–351, 1994.
12. Z. Pan, X. Ma, M. Zhang, and J. Shi, Chinese font composition method based on algebraic system of geometric shapes, *Comput. & Graphics*, **21**(3):321–328, 1997.
13. H.T.F. Wong and H.H.S. Ip, Virtual brush: a model-based synthesis of Chinese calligraphy, *Comput. & Graphics*, **24**(1):99–113, 2000.
14. S. Strassmann, Hairy brush. *Proc. of SIGGRAPH '86*, 225–232, 1986.
15. R. Greene, The drawing prism: a versatile graphic input device. *Proc. of SIGGRAPH '85*, 103–109, 1985.
16. J. Lee, Diffusion rendering of black ink paintings using new paper and ink models. *Comput. & Graphics*, **25**:295–308, 2001.
17. T.L. Kunii, G.V. Nosovskij, and T. Hayashi, A diffusion model for computer animation of diffuse ink painting, *Proc. of Comput. Animation '95*, 98–102, 1995.
18. Q. Zhang, Y. Sato, J. Takahashi, K. Muraoka, and N. Chiba, Simple cellular automaton-based simulation of ink behavior and its application to Suibokuga-like 3D rendering of trees, *J. of Visualization and Comp. Animation*, **10**(1):27–37, 1999.
19. J. Lee, Simulating oriental black-ink painting, *IEEE Comput. Graphics and Applications*, **19**(3):74–81, 1999.
20. B. Baxter, V. Scheib, M.C. Lin, and D. Manocha, DAB: interactive haptic painting with 3D virtual brushes. *Proc. of SIGGRAPH 2001*, 461–468, 2001.
21. C.J. Curtis, Computer-generated watercolor, *Proc. of SIGGRAPH '97*, 421–430, 1997.
22. A. Hertzmann, Painterly rendering with curved brush strokes of multiple sizes, *Proc. of SIGGRAPH '98*, 453–460, 1998.
23. S. Saito and M. Nakajima, 3D physically based brush model for painting, *SIGGRAPH '99 Conference Abstracts and Applications*, 226, 1999.
24. G. Winkenbach and D.H. Salesin, Computer-generated pen and ink illustration, *Proc. of SIGGRAPH '94*, 91–100, 1994.
25. J. Lansdown and S. Schofield, Expressive rendering: a review of nonphotorealistic techniques, *IEEE Comput. Graphics and Applications*, **15**(3):29–37, 1995.
26. T. Nakamura, H. Itoh, H. Seki, and T. Law, A writing system for brush characters using neural recognition and fuzzy interpretation, *Proc. of 1993 Int. Joint Conf. on Neural Networks*, 2901–2904, 1993.
27. T. Yamasaki and T. Hattori, Forming square-styled brush-written Kanji through calligraphic skill knowledge, *IEEE Proc. of Multimedia '96*, 501–504, 1996.
28. X. Wei, S. Lu, M. Song, and B. Luo, Computer pattern design and painting technique based on aesthetics knowledge, *Comput. Aided Drafting, Design and Manufacturing*, **2**(2):32–40, 1992.
29. K. Henmi and T. Yoshikawa, Virtual lesson and its application to virtual calligraphy system, *Proc. of 1998 IEEE Int. Conf. on Robotics & Automation*, 1275–1280, 1998.
30. M. Agrawala, A. Beers, and M. Levoy, 3D painting on scanned surfaces, *Proc. of 1995 Symp. on Interactive 3D Graphics*, 145–150, 1995.
31. D. Hohnson, T.V. Thompson II, M. Kaplan, D. Nelson, and E. Cohen, Painting textures with a haptic interface, *Proc. of IEEE Virtual Reality Conference*, 1999.
32. D.G. Luenberger, *Linear and Nonlinear Programming*, 2nd edition, Addison-Wesley, 1984.
33. C. Maurer and B. Juttler, Rational approximation of rotation minimizing frames using Pythagorean-hodograph cubics, *J. of Geometry and Graphics*, **3**(2):141–159, 1999.
34. L.D. Landau and E.M. Lifshitz, *Theory of Elasticity, Course of Theoretical Physics*, Vol. 7, Pergamon Press, Oxford, 1986.
35. D.-L. Way and Z.-C. Shih, The synthesis of rock textures in Chinese landscape painting, *Proc. of Eurographics 2001*, C123–C131, 2001.