Magic Brush – A Novel Digital Painting Environment

Songhua XU

Department of Computer Science, Zhejiang University, Hangzhou, P.R.China, Zip Code: 310027

E-mails: songhua@csis.hku.hk, xsonghua@hotmail.com

***Keywords*—Magic Brush, Paint Systems, Interaction Techniques, Physically Based Modeling, Real-time Simulation.**

I. INTRODUCTION

A long cherished dream of both scientists and artists since the birth of digital computers is to be able to use the computer to produce beautiful art. With the ever increasing power of modern computers, the computer as a serious artistic tool is now within reach for many. One major research area in computer art is emulating traditional art media for interactive painting or calligraphy. A brush is many times more complex than a pen, and hence the problem presents a huge challenge for the interested computer scientists. We describe our solution to the problem in this article.

Smith has written a good survey on the early painting systems [2]. More recently, several researchers have successfully implemented a virtual brush (or "e-brush") that can mimic a real 3D brush for painting and calligraphy through physically-based modeling [6], [3], [8]. Compared with many existing 2D virtual brushes which often require the user to specify the contour's control points and the texture of painted strokes (e.g., [11], [5], [1]), these 3D brushes are more natural to use, especially for non-computer specialists. A guiding principle for the design of an e-brush is that it must present itself as a familiar tool to the traditional human artist, because presenting a system that requires the designer to adapt, distract, or place attention outside of the actual task will hinder the creative process to which the designer can come up with ideas. With the e-brush we have implemented, the user will not be required to specify any control point or to adjust any parameter for the texture, but can operate the brush in more or less the same way as a physical brush.

In a good virtual brush design, each step of the painting process should be simulated in high realism. Here simulation refers to both the production of strokes on the virtual canvass and the continuous visual display of the running brush. The latter is necessary if the user is to be able to feel the presence of a brush in order to have full control over it and to manipulate the brush as in real life. An even more perfect virtual brush should also provide a haptic interface, like what is done in [3].

In this article, we describe a new e-painting system [13], [12], [15], with focus on the modeling aspects—the geometry and the dynamics of the paintbrush. Our modeling of the paintbrush stands out among existing approaches because of its unique ability to capture the highly complex geometry of a physical brush as well as its dynamic behavior during painting with great accuracy. Figure 1 shows the overall architecture of our magic brush. Our modeling approach pays special attention to many very fine details that are called for by the creation of high quality digital paintings. Because of these very fine details, the system presents a "realistic" simulation of the physical brush to the user who would operate like a real brush and expect the output to be as good as real. As can be seen through the sample calligraphy artwork at Figure 10 and the samples of painting at Figure 11, the artwork created by our realistic magic brush can rival the real artwork.

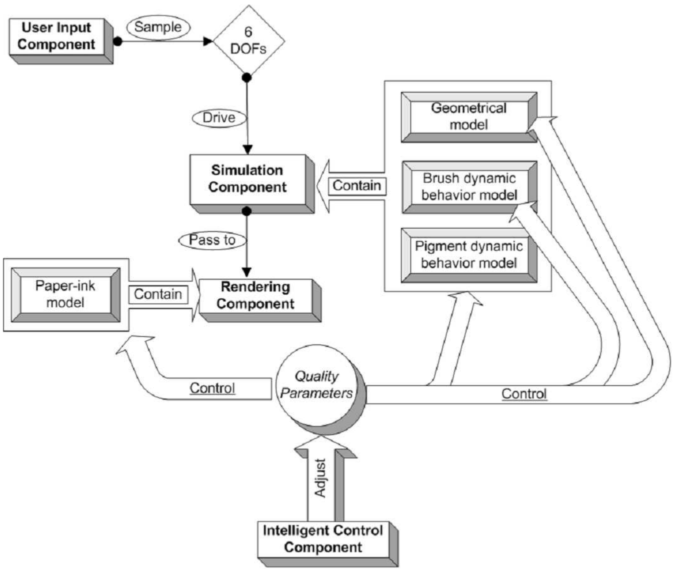


Fig. 1. Architecture of our magic brush.

The article is organized as follows. Sections II and III present our realistic modeling of the geometry and the dynamics of the magic brush respectively. Section IV gives an overview of the new e-painting system we developed based on our modeling. Section V concludes the paper and discusses possible future tasks.

II. MODELING THE PAINTBRUSH'S GEOMETRY

One of the earliest paintbrush models was by Strassmann [11], in which brush strokes are created by sweeping a 1D brush bristle over a skeleton, and the color, width and wetness of the brush can be varied. Wong and Ip's virtual brush [6] is modeled as an inverse cone which can produce an elliptic drawing mark. In the DAB project [3], a subdivision surface is wrapped around a spring-mass particle system skeleton to emulate the brush surface. Most recently, Chu and Tai [8] use a geometrical based approach to model an un-split brush tip and an alpha map to model a split brush tip. None of these models however is powerful enough to model a physical brush to a high degree of likeness to the physical brush's real geometry. In particular, the common situation that a brush splits into a large number of hair bundles appears to be out of the reach of all these existing models. According to many practising artists, a feature to model such a high degree of splitting is very much desirable. The novel e-brush modeling method we propose here can effectively deal with this and other hard problems using little memory and CPU resource.

A. Three-layer Hierarchical Modeling

Our realistic modeling of the paintbrush geometry is organized as a three layered hierarchy: the lowest layer consists of *painting primitives*, which are clusters of hair threads; the intermediate layer consists of clusters of painting primitives; the highest layer is the whole brush tip bundle.

Hair threads whose positions in the 3D space are close to each other and their geometries are similar are modeled together as one *painting primitive*, which is a single, aggregative representation of both the geometry and dynamics of these hair threads. A painting primitive is the smallest granularity in our modeling. Hence, the overall modeling capability of our magic brush derives from the modeling power of a painting primitive. We construct the model of a painting primitive **H** through the general sweeping operation in CAD by moving a variable ellipse $E(t)$ along a 3D trajectory, $K(t)$. We call the trajectory "the skeleton of **H**", which is represented as a 3D B-spline. The generated skinning surface is the surface of **H** while the swept volume is the interior volume of **H**. During the sweeping operation, we ensure that the sweeping ellipse $E(t)$ always lies on the normal plane of the sweeping trajectory $K(t)$, i.e. $E(t)|_{t=t_0} \perp K(t)|_{t=t_0}$. See Figure 2 for the illustration of a painting primitive in its initial geometry, and a deformed painting primitive.

$E(t)$ is a variable ellipse in that the lengths of its major axis $L(t)$, minor axis $S(t)$ and its orientation $\theta(t)$ (on the normal plane of its sweeping trajectory) can all be varied during sweeping. Thus, given three B-splines, $S(t)$, $L(t)$ and $\theta(t)$, we can uniquely determine an $E(t)$ $(0 \leqslant t \leqslant 1)$. This means by four B-splines, $\big(K(t), S(t), L(t), \theta(t)\big)$, the geometry of a painting primitive, **H**, can be presented parametrically. This representation makes it easy and intuitive for end users to tailor the geometry of a painting primitive and further customize their magic brush if the users so choose. Figure 9.(a) shows the graphical user interface to be used for this purpose.

The general sweeping operation we adopted is a suitably powerful modeling metaphor, which meets the demand of modeling long furry objects. Our representation is simple to generate and capable of capturing all kinds of geometry that a paintbrush may have, as is demonstrated in Figure 4.(a). In comparison, although the subdivision surface mesh employed in [3] is a more powerful modeling metaphor in general, unlike our general sweeping operation, it is not specifically customized for the modeling of brush hair. Therefor, in spite of its being more sophisticated than our method, it cannot deal with some of the more extreme cases, such as heavily deformed brush hair.

General sweeping, however, is a time-consuming operation. Moreover, the internal solid models of all the painting primitives together could take up a substantial amount of space, especially in situations when the brush is split into many small hair bundles. Consideration on system response time and memory resources calls for a strategy to eliminate as much as possible the redundancy in representing and simulating the brush hair. We introduce the idea of *painting primitive cluster* to group disjoint painting primitives whose geometries are similar into one single modeling unit. This is reasonable move because it can be easily observed in real life when a brush is split into numerous disjoint hair threads or clusters, there would only be a limited number of sharply distinctive geometries among the clusters. Note that the grouping considers only the geometries but not the physical positions of the painting primitives in the 3D space. Given a painting primitive cluster and the geometry of any individual painting primitive in the cluster, the geometries of all the remaining painting primitives of the cluster can be easily derived via simple affine transformations. This design gives rise to a brush geometry model that is compact in memory, and enables fast simulation of the brush's actions.

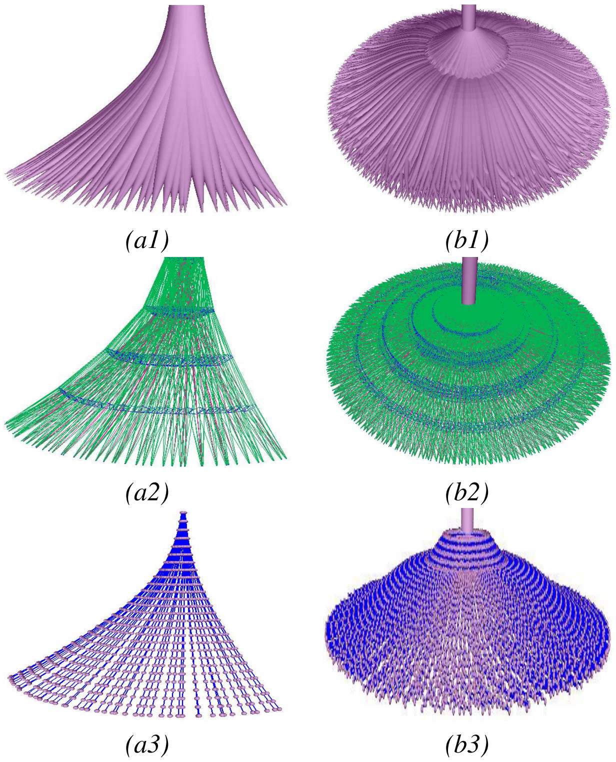

(a1) (b1)
(a2) (b2)
(a3) (b3)

Fig. 3. (a1 to a3) A split magic brush when shaded, in wireframes, and in terms of profiles of the sweeping ellipses, respectively; (b1 to b3) a brush that is heavily splitting.

Figures 3 shows two examples of complex brush geometry modeled using our three-layer hierarchy. Figure 4 shows the brush modeling when one, two, or three layers in the hierarchy are in effect.

B. Real-time Visual Display of the Brush

Visual feedback is important for any interactive system. In a painting session, the user needs to feel the physical presence of the brush in order to manipulate it at will. To provide a good visual feedback could require a huge amount of computation because of the highly complex geometry of a realistic virtual

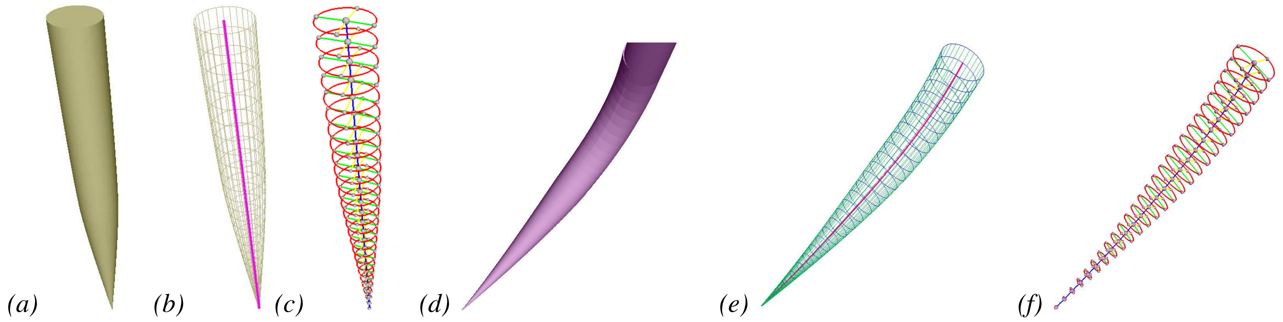


Fig. 2. (a) – (c), an initial painting primitive: (a) when shaded; (b) as a wireframe with the sweeping trajectory highlighted; (c) as a series of profiles of the sweeping ellipse, which are in red. (d) – (f), a deformed painting primitive: (d) when shaded; (e) wireframe; (f) profiles of the sweeping ellipse.

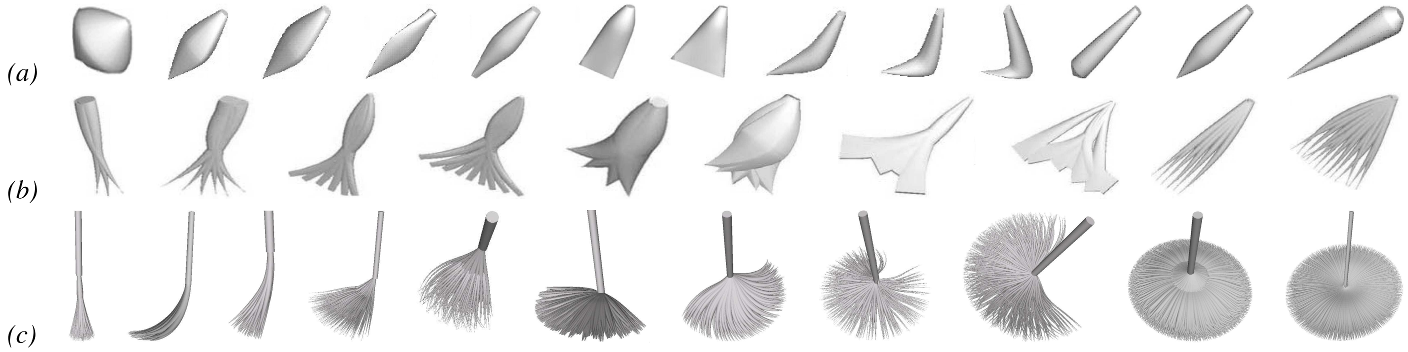


Fig. 4. Brush geometry modeling in three levels: (a) using only the top level—the whole brush tip bundle; (b) using two levels—the top level and the "painting primitive" level; (c) using all three levels.

brush. Our hierarchical modeling provides a solution to efficiently tackle the problem. As discussed previously, we only need to explicitly model the geometry of one painting primitive for each painting primitive cluster, from which all other geometries belonging to the same cluster can be derived. Thus all the painting primitives in a cluster share the same data structure, and one tessellation process being applied to the geometry of one painting primitive is sufficient for tessellating all the painting primitives in the cluster (by applying the same affine transformation as mentioned before).

We also make sure that modeling satisfies the preconditions necessary for taking advantage of the hardware acceleration facility through the "Display Lists" feature in OpenGL. The time-consuming commands for rendering the geometries of the painting primitives would therefore be optimized by the driver as the affine transformations can be pre-computed. With the hierarchical modeling approach and hardware acceleration, our magic brush can achieve real-time visual feedback of the complex geometry of the brush using a reasonably small amount of memory and CPU time.

III. MODELING THE PAINTBRUSH'S DYNAMIC BEHAVIOR

Simulating the dynamics of a real paintbrush, which includes the brush's deformation due to outer force, recovery from deformation when the force vanishes, splitting due to inner stress, etc., is a non-trivial problem because of the complexity of the brush's geometry and the physical principles that underlie the brush's behavior. In the DAB project [3], the motion of the brush geometry is simulated through a pair of first-order differential equations. The dynamics of Chu and Tai's brush [8] are modeled as springs whose deformation is via constrained energy minimization. In spite of all these efforts, what would be a good model of brush dynamics that supports realistic, efficient, and stable brush motion simulation using a reasonable amount of system resources remains not completely answered.

Our design of the magic brush system offers a highly detailed dynamic modeling of the behavior of a physical paintbrush. The modeling is divided into two phases. The first phase consists of on-line computation of the computationally inexpensive and input-sensitive physical processes, such as brush deformation due to brush pressing. We adopt the approach of using a "phenomenal model" which simulates the dynamics of a changing object based on observations instead of by the highly complex underlying physical laws that govern the changes. It proves to be a highly economical approach when we have to model a large number of brush features. The result is fast simulation of a sufficiently detailed model of the brush. With this observational modeling approach, however, we compromise some degree of modeling accuracy. And so in the second phase, off-line data are used to calibrate and refine the on-line simulated results. These data come from a simulation error calibration database constructed from off-line acquired ground truth about simulation errors. Our design represents a balance between a complete on-line based approach and one that is at the other extreme. The former demands a huge amount of runtime resources in order to achieve real-time response; the latter could result in a database which is too large to manage. This "observation model plus calibration database" approach achieves high realism for the brush dynamics being simulated as well as interactivity with little incurred computational cost. See Figure 5 for some simulated

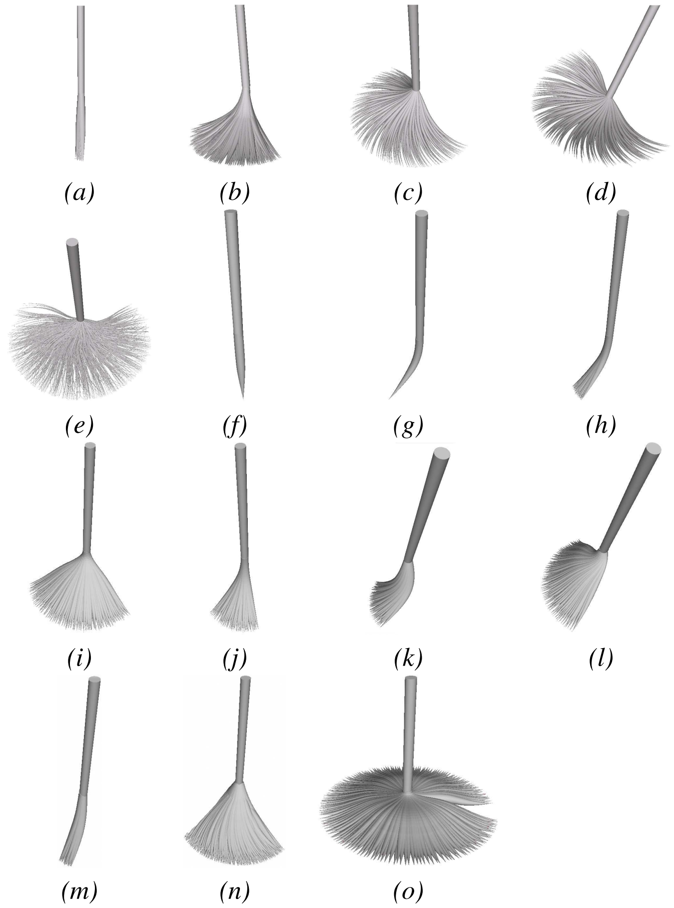


Fig. 5. (a)–(e) Simple dynamic deformation of the magic brush: (a) the initial geometry, (b) pressed down, (c) rotated, (d) tilted, (e) further pressed down and rotated. (f)–(o) A more sophisticated magic brush deformation process: (f) the initial brush; (g) pressed down before splitting is simulated; (h) after splitting is simulated; (i) rotated and further pressed down; (j) lifted a bit with some split brush threads merging; (k) titled; (l) further rotated and pressed down; (m) further lifted to the initial brush position; (n) pressed down again; (o) completely pressed down to the virtual paper with some rotation.

dynamic deformation of our magic brush.

During painting, the geometry of the paintbrush is deformed due to the outer force arising from friction between the brush and the paper. Brush deformation is modelled in the first phase of the two-phase modeling, in two different parts: deforming of painting primitive due to brush-paper collision and deforming of painting primitive due to the brush's inner stress.

A. Deformation Due to Brush-Paper Collision

A.1 Deformation of skeleton

The main constraint to observe here is that the brush's skeleton cannot penetrate the virtual paper. During brush painting or writing, if some part of a skeleton $K(t)$ penetrates the virtual paper ψ at an intersecting point $K_o(t_p)$ between the original skeleton $K_o(t)$ and the virtual paper ψ, we would deform $K_o(t)$ by replacing it with a new curve $K_n(t)$ to satisfy the above constraint. The new curve is actually an interpolated result of the projected skeleton and the rotated skeleton. The deformation scheme is illustrated by Figure 6.

We also model the kinking up of a painting primitive due to large friction induced by high stress which is according to classic Newton force. The larger the friction, the slower the painting primitive will move. This non-uniform displacement will cause local prolongation and compression of the painting primitive's skeleton, which when become too severe will lead to kinking up of the skeleton, as shown in Figure 7. To simulate this, we introduce additional curvature to the skeleton so that the arc length of the skeleton is the skeleton's original length and the chordal length is its squeezed length.

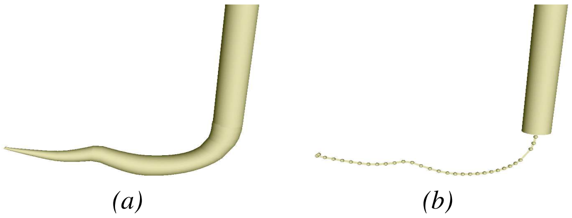


Fig. 7. (a) A kinked-up painting primitive; (b) its corresponding skeleton.

A.2 Deformation of sweeping ellipse

Recall that the geometry model of a painting primitive is constructed by sweeping a variable ellipse along a trajectory. When the brush is deformed, the profiles of the variable ellipse touching the paper will also be deformed. We apply minimization to the areas of the parts of the deformed profiles that are under the virtual paper plane, with the hard constraint that the profiles' areas cannot change. These areas could change at a later stage in the modeling when we take into account the inner stress. We minimize the areas concerned for the simple reason that the physical brush cannot go under the canvass; minimizing gives us the best approximation to what would happen in reality.

B. Deformation Due to Inner Stress

Once the painting primitive is deformed due to collision between the brush and the paper, inner stress will develop inside the painting primitive. Because we model the geometry of a painting primitive explicitly, its volume is computable. Based on the volumes of the initial and the deformed geometry of a painting primitive, we can estimate the inner stress of the painting primitive. This inner stress will then give rise to further deformation of the geometry of the painting primitive, in the form of distension and splitting of the painting primitive.

B.1 Estimating the Inner Stress

We notice any point in a certain profile $E(t)|_{t=t_0}$ of the variable ellipse $E(t)$ can be determined by the pair (ν, w) under a given $L(t_0)$, $S(t_0)$ and $\theta(t_0)$. Accordingly, we set up a local planar elliptical polar coordinate system for $E(t_0)$ by taking the major and minor axes of $E(t_0)$ as two axes of the coordinate system, and the center of $E(t_0)$ as its original. We can then establish a point to point correspondence between the undeformed profile of $E(t_0)$ and its deformed counterpart in the new local coordinate system. With this correspondence, we can estimate the local inner stress of a point within the brush head volume. By integrating this point-wise stress, we can further evaluate the average inner stress $\rho(E(t))$ of the ellipse $E(t)$ as well as the average inner stress $\rho(\mathbf{H})$ of the deformed painting primitive $\mathbf{H}$.

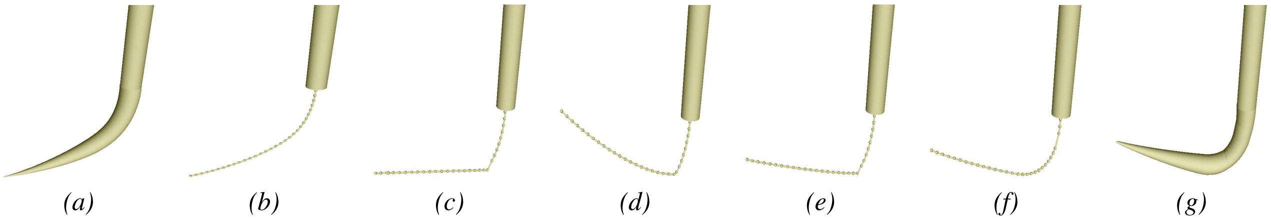


Fig. 6. Skeleton deformation of the magic brush: (a) a painting primitive penetrating the virtual paper; (b) its corresponding skeleton; (c) the projected skeleton, (d) the rotated skeleton; (e) the interpolated version of the skeleton; (f) the new skeleton after smoothing, which is completely above the virtual paper; (g) the deformed painting primitive with the new skeleton.

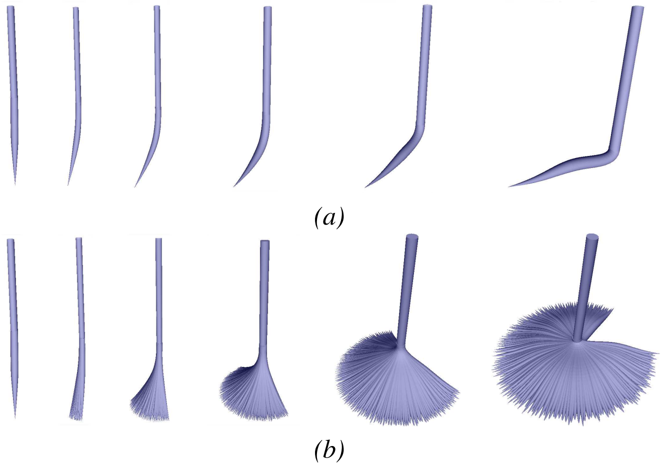


Fig. 8. A brush deforming when being pressed down continuously: (a) with no splitting of hair; (b) with splitting.

B.2 Splitting of painting primitive

If there is a part inside the volume of a painting primitive, whose stress is above the threshold of a maximum tolerable inner stress, the painting primitive will split. Splitting of the painting primitive takes place in a stress-descending order if multiple parts of the magic brush meet the splitting criterion simultaneously. Split painting primitives could merge again if their inner stress starts to come down. In the current version of our modeling, merging however is controlled through user interaction, mimicking the way the user caresses a physical brush to merge some split hair. Figure 8 shows some brush motion simulation results with and without brush splitting.

B.3 Recovery from Deformation

To simulate recovery from deformation, we model the material of the painting primitive as a kind of hybrid elastic-rigid material. According to study in material science [17], the deformed painting primitive will recover by a certain degree, which ranges from full recovery in the case that the material is purely flexible to no recovery at all if the material is completely rigid. In our computation model, we determine the recovery extent based on the amount of inner stress of the painting primitive and the painting primitive's inherent threshold of elasticity and threshold of rigidity. Using the estimated recovery extent as the weight, we can then simulate the brush's recovery by interpolating its original un-deformed geometry and its current deformed geometry.

C. Calibrating the On-line Results

To ensure the accuracy of our simulation of the brush dynamics with respect to a physical brush, we calibrate the fast on-line simulation results in the second phase of the modeling. The procedure relies on data from a "simulation error database", whose records came from sampling using a real brush. Since the deformation of our magic brush's geometry is according to the six degrees of freedom (DOFs) of the input, the database is indexed by the differential of the magic brush's six DOFs between two consecutive simulation time slices and the current brush geometry. The content of each record in the database is the corresponding transformation on magic brush's current geometry. Our brush deformation database operates at the level of painting primitive in our three-level hierarchy. This reduces substantially the number of different cases that need to be separately sampled and stored in the database. To further reduce the size of the calibration database, we assume the effects that all the six DOFs have on the brush geometry are independent. As a result, the whole calibration database is divided into many small sections, each of which being responsible for calibrating the part of the deformation caused by variation of only one specific DOF. It turned out that a modest number of records (40) is enough for performing a satisfactory calibration to improve the on-line simulation results in our system.

Compared with traditional physically-based approaches to model the brush dynamics through numerical computation, there are two important advantages of our on-line plus off-line modeling approach. First, we do not have to model those extremely complicated physical processes of a brush's dynamics. Second, we can avoid the very time-consuming and probably unstable numerical computation for solving differential equations on the fly if some truly powerful but complex equations can be established to model all the underlying detailed dynamics governing the brush's behavior.

IV. E-PAINTING SYSTEM BASED ON REALISTIC MAGIC BRUSH MODELING

We have built a complete working e-brush system based on the modeling strategies just described. Compared with other virtual brushes, this new system is designed to present a realistic brush in the sense that the system accurately and stably simulates the complex painting functionality of a running brush, and therefore is capable of creating high-quality digital paintings with minute aesthetic details that can rival the real artwork.

A. Additional Components of Our New Painting System

Other than modeling on the brush's geometry and dynamics, the system has also incorporated a novel pigment model and a user manipulability improvement component. For the paper's length limitation, we will only give an overview in the following. In-depth technical details are available at [13], [12], [15].

A.1 A Novel Pigment Model

During painting, the contour of the current ink mark left by our magic brush on virtual paper can be computed by intersecting the geometrical models of these two virtual objects. To produce a good texture for the ink mark, a pigment model is required. A number of pigment models for digital painting have been proposed, but they are either not well embedded into a 3D virtual paintbrush, or developed in early years and therefore could only produce some very coarse painting results. A recent pigment model is contributed by [4]. It is however too slow to be used in interactive painting because of the need to solve heavy differential equations on the fly. Other pigment models that can be found in existent virtual brush systems are too simplistic: they either simply transfer ink values from the penetrating brush tip onto the ink mark area [8], or apply alpha blending in the above ink transformation to simulate glazing effects [3].

For our system, we devised a new pigment model that is best for expressive oriental painting. The prominent feature of this new pigment model is that it is completely and seamlessly integrated into our realistic magic brush model: we store both the local ink color and the wetness in each control point of the geometry model of our magic brush. When generating the ink mark, each point in the ink mark is painted using a color which is a linearly interpolated result between the ink mark's original color on the paper and the color of the point on the brush surface contacting this ink mark. The interpolation weight is a random number, whose distribution is controlled by the current inner stress of the painting primitive, the local wetness around the painted pixel, as well as the quality parameters of the magic brush. This probability based pigment model allows us to simulate the dry brush effect, the running style effect, and the ink saturation effect. These are important aesthetic effects that can contribute significantly to the expressiveness of the painting system.

A.2 User Manipulability Improvement Component

As with any real brush, it can happen that end users would feel unsatisfied with their creations using the magic brush. Instead of training the users, our magic brush has the unique feature of training the brush to cater to the personal painting habits of different users. In our realistic magic brush model, a number of quality parameters are embedded which are specifically introduced to simulate different kinds of brushes. This set of "quality parameters" can control the modeled magic brush dynamics as well as the ink depositing process, which can be customized for different users through a machine training module. More sophisticated AI techniques can be employed to further increase the usability of our painting system by making it more intelligent at automatic system self-tuning and adaptation [16], [14].

B. The Running System

Figures 9.(a)–Figures 9.(b) show two screenshots of the GUI of the running system. (a) was taken when the user was customizing the geometry of the painting primitive—the skeleton as well as the profiles of the sweeping ellipse. (b) shows a painting in the making, with some closeup views as (c)–(d).

Figure 10 shows sample calligraphy artwork by our magic brush. And Figure 11 is a collection of examples of e-paintings created using the magic brush, which are imitations of real paintings. To give an idea on the effort required for digital painting using our magic brush, the horse (Figure 11.(d)) took about 1.5 hours to complete. The more complex pictures, such as Figure 11.(g), took a bit longer. Although the simulation attends to many fine details, our magic brush running on a PC with 256M memory and an AMD Duron 1.2 GHz processor can respond interactively to user commands. Currently, we use a WACOM pen on a tablet to get the position, the pressure (used as the brush's vertical displacement), and the tilt of the magic brush; and keyboard input to get the remaining two DOFs. A better input device in the future should provide some degree of haptic feedback.

V. CONCLUSION AND FUTURE WORK

We have presented the design of a powerful painting system based on realistic modeling of the paintbrush. The amount of details being modelled necessitates the many time or space optimizations that we have introduced into the design. The result is a high degree of realism in every simulation step. Here is a summary of the unique features of our modeling approach. (1) Clustered and hierarchical modeling is used to minimize redundant representations and computations, and together with hardware acceleration, the model is easily renderable in real-time. (2) Division of the modeling into on-line tasks and off-line calibration makes possible an accurate and stable simulation of the brush's motion using little computational resources. (3) Our magic brush can automatically determine both its geometric contour and the texture of its ink mark on the virtual paper without any human intervention. (4) Other features such as the pigment model and the user manipulability adaptation component make the magic brush system a powerful one and natural to use for creating high-quality e-artwork.

Although we have gone after a detailed modeling of the paintbrush, a real brush operates in a fashion that is orders of magnitude more complex than what is currently simulated. Features that can be added in the future versions include repeated dipping effects, and more user's control of the brush during painting. Other features requiring longer-term effort include support for 3D painting (such as oil painting), and vectorization of painting results. The latter could lead to many interesting applications such as animation of e-paintings.

VI. ACKNOWLEDGEMENT

This project has won the 1^{st} placement and the "Edison Cup" in the *GE Fund "Edison Cup" Technology Innovation Competition*, a competition organized by Institute of International Education (US). It is also awarded the outstanding prize and the "Challenge Cup" in *the* 8^{th} *Chinese National "Challenge*

(a) *(b)* *(c)*

(d) *(e)* *(f)*

(g)

Fig. 11. Sample paintings by magic brush. (a) "Spring garden", (b) song birds, (c) mandarin ducks, (d) a running horse, (e) parrots, (f) orchid, (g) bamboos.

Cup" College Student Research Project Competition, a competition organized by Chinese Association of Scientists and Chinese Ministry of Education.

REFERENCES

[1] Adobe Photoshop (software), Adobe Systems Incorporated.

[2] A.R. Smith. Digital paint systems: an anecdotal and historical overview. *IEEE Annals of the History of Computing*, 2001.

[3] B. Baxter, V. Scheib, M.C. Lin and D. Manocha. DAB: interactive haptic painting with 3D virtual brushes. *Proc. of SIGGRAPH '01*:461–468, 2001.

[4] C.J. Curtis, S.E. Anderson, J.E. Seims, K.W. Fleischer and D.H. Salesin. Computer-generated watercolor. *Proc. of SIGGRAPH '97*:421–430, 1997.

[5] Expression (software), *Creature House Ltd., Hong Kong.*, first released in October 1999. http://www.creaturehouse.com/

[6] H.T.F. Wong and H.H.S. Ip. Virtual brush: a model-based synthesis of Chinese calligraphy. *Computers and Graphics* **24**(1):99–113, 2000.

[7] M.P. Salisbury, S.E. Anderson, R. Barzel and D.H. Salesin. Interactive pen-and-ink illustration. *Proc. of SIGGRAPH '94*:101–108, 1994.

[8] N.S.H. Chu and C.L. Tai. An efficient brush model for physical-based 3D painting. *Proc. of Pacific Graphics '02*.

[9] S. Saito and M. Nakajima. 3D physics-based model for painting. *SIGGRAPH '99 Sketches, Conference Abstracts and Applications*:226–226, 1999.

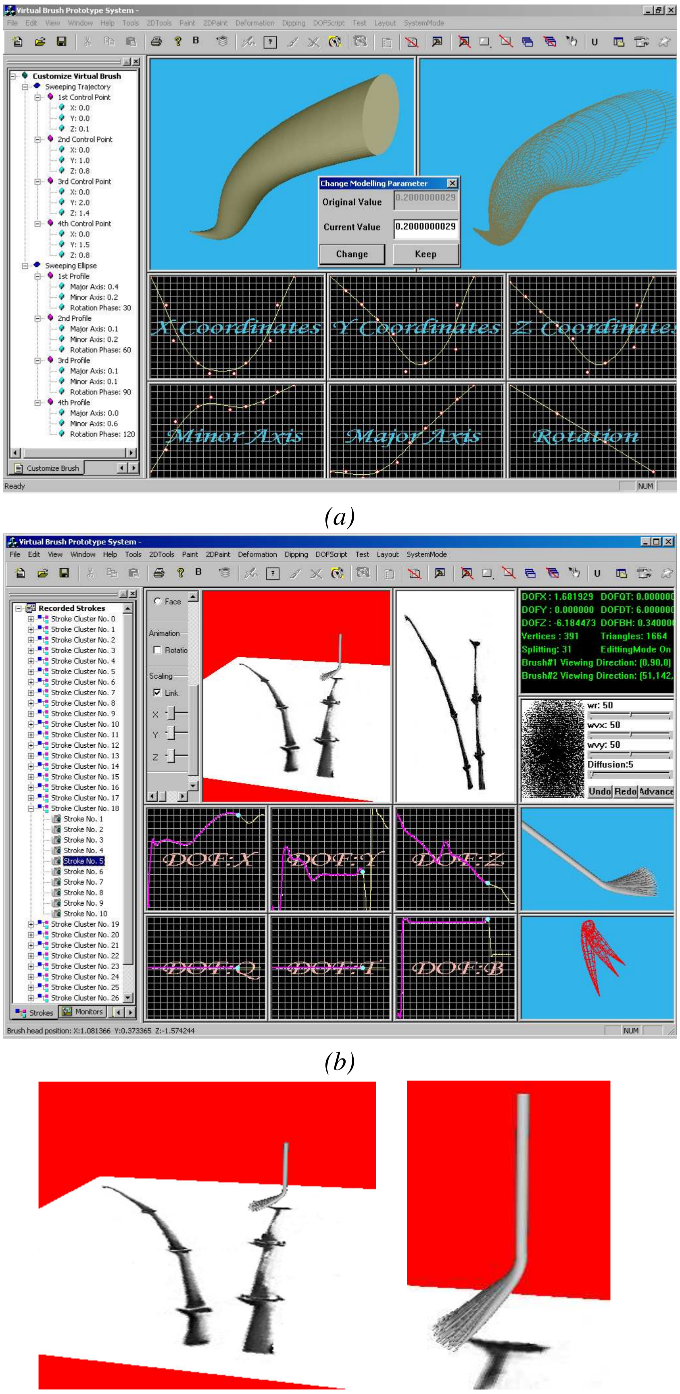
[10] S.C. Hsu and I.H.H. Lee. Drawing and animation using skeletal strokes. *Proc. of SIGGRAPH '94*:109–118, 1994.

[11] S. Strassmann. Hairy brushes, *Proc. of SIGGRAPH '86*:225–232, 1986.

[12] S. Xu, F.C.M. Lau, F. Tang and Y. Pan. Advanced design for a realistic virtual brush. *Proc. of the 24th Annual Conference of the European Association for Computer Graphics (Eurographics '03). Also in Computer Graphics Forum,* ***22****(3):533–542, 2003.*

[13] S. Xu, M. Tang, F.C.M. Lau and Y. Pan. A solid model based virtual hairy brush. *Proc. of the 23rd Annual Conference of the European Association for Computer Graphics (Eurographics '02). Also in Computer Graphics Forum,* ***21****(3):299–308, 2002.*

[14] S. Xu, F.C.M. Lau, K.W. Cheung and Y. Pan. Automatic generation of artistic Chinese calligraphy. *Proc. of the Nineteenth National Conference on Artificial Intelligence and the Sixteenth Annual Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI '04). Extended version available as Techinical Report HKU-CSIS-TR-2003-02.*



Customize Virtual Brush
Sweeping Trajectory
1st Control Point
X: 0.0
Y: 0.0
Z: 0.1
2nd Control Point
X: 0.0
Y: 1.0
Z: 0.8
3rd Control Point
X: 0.0
Y: 2.0
Z: 1.4
4th Control Point
X: 0.0
Y: 1.5
Z: 0.8
Sweeping Ellipse
1st Profile
Major Axis: 0.4
Minor Axis: 0.2
Rotation Phase: 30
2nd Profile
Major Axis: 0.1
Minor Axis: 0.2
Rotation Phase: 60
3rd Profile
Major Axis: 0.1
Minor Axis: 0.1
Rotation Phase: 90
4th Profile
Major Axis: 0.0
Minor Axis: 0.6
Rotation Phase: 120

Original Value 0.2000000029
Current Value 0.2000000029

(a)

Recorded Strokes
Stroke Cluster No. 0
Stroke Cluster No. 1
Stroke Cluster No. 2
Stroke Cluster No. 3
Stroke Cluster No. 4
Stroke Cluster No. 5
Stroke Cluster No. 6
Stroke Cluster No. 7
Stroke Cluster No. 8
Stroke Cluster No. 9
Stroke Cluster No. 10
Stroke Cluster No. 11
Stroke Cluster No. 12
Stroke Cluster No. 13
Stroke Cluster No. 14
Stroke Cluster No. 15
Stroke Cluster No. 16
Stroke Cluster No. 17
Stroke Cluster No. 18
Stroke No. 1
Stroke No. 2
Stroke No. 3
Stroke No. 4
Stroke No. 5
Stroke No. 6
Stroke No. 7
Stroke No. 8
Stroke No. 9
Stroke No. 10
Stroke Cluster No. 19
Stroke Cluster No. 20
Stroke Cluster No. 21
Stroke Cluster No. 22
Stroke Cluster No. 23
Stroke Cluster No. 24
Stroke Cluster No. 25
Stroke Cluster No. 26

DOFX : 1.681929 DOFQT: 0.000000
DOFY : 0.000000 DOFDT: 6.000000
DOFZ : -6.184473 DOFBH: 0.340000
Vertices : 391 Triangles: 1664
Splitting: 31 EdittingMode On
Brush#1 Viewing Direction: (0,90,0)
Brush#2 Viewing Direction: (51,142,

wr: 50
wvx: 50
wvy: 50
Diffusion:5

Brush head position: X:1.081366 Y:0.373365 Z:-1.574244

(b)

(c) (d)

Fig. 9. GUI of the running system: (a) the running system—user customizing the magic brush; (b) the running system—bamboos being painted; (c & d) closeup views of (b).

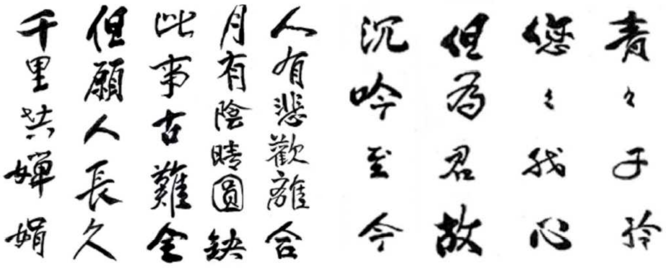


Fig. 10. Two Chinese love poems by magic brush.

[15] S. Xu, M. Tang, F.C.M. Lau and Y. Pan. Virtual hairy brush for painterly rendering. *Accepted for publication at Graphical Models. Also available as Techinical Report HKU-CSIS-TR-2002-04.*

[16] Y. Gu, S. Xu, M. Tang and J. Dong. AI supported computer-generated pen-and-ink illustration. *Proc. of IEEE Computer Supported Cooperative Work in Design (DSCWD '01): 227–231, 2001.*

[17] W.D. Calister and W.D. Callister Jr. *Materials Science and Engineering: An Introduction*, Wiley, John & Sons, Incorporated, 1999.

CURRICULUM VITAE

Songhua Xu is currently a 3rd year graduate student at the Department of Computer Science, Zhejiang University, Hangzhou, Zhejiang, P.R. China. He got his B.E. from Zhejiang University in 2001 with *Summa Cum Laude*. From July 2002 to March 2004, he visited Microsoft Research Asia as a research assistant. His research interests are on computer graphics, especially computer art, and artificial intelligence. He is currently a student member of Eurographics.