EUROGRAPHICS 2003 / P. Brunet and D. Fellner
(Guest Editors)

Volume 22 (*2003*), *Number 3*

Advanced Design for a Realistic Virtual Brush

Songhua Xu[†,‡], Francis C.M. Lau[‡], Feng Tang[†] and Yunhe Pan[†]

† CAD & CG State Key Lab of China, Zhejiang University, P.R. China
‡ Department of Computer Science and Information Systems, The University of Hong Kong, Hong Kong

Abstract

This paper proposes a novel algorithmic framework for an advanced virtual brush to be used in interactive digital painting. The framework comprises the following components: a geometry model of the brush using a hierarchical representation that leads to substantial savings in every step of the painting process; fast online brush motion simulation assisted by offline calibration that guarantees an accurate and stable simulation of the brush's dynamic behavior; a new pigment model based on a diffusion process of random molecules that considers delicate and complex pigment behavior at dipping time as well as during painting; and a user-adaptation component that enables the system to cater for the personal painting habits of different users. A prototype system has been implemented based on this framework. 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. The advanced features also give rise to a high degree of expressiveness of the virtual brush that the user can comfortably manipulate. http://www.csis.hku.hk/∼songhua/e-brush/ provides supplementary materials for this paper.

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

Virtual brush is an important tool for interactive painting [23, 17, 20, 21, 13, 12, 16, 22, 8, 4, 19, 24, 15]. Smith has written a good survey on the early painting systems [2]. These early systems offered a 2D brush for painting. The newer painting systems are more sophisticated, which can for example automatically generate painterly rendering results based on some input reference images [3]. To provide a virtual brush that can mimic the real 3D brush for painting and calligraphy writing has become a popular area for research in recent years [8, 4, 24, 15]. The researchers are attracted by the unique ability of a 3D virtual brush to generate expressive and realistic results through techniques such as physically-based modeling. Common to most existing 3D virtual brushes are three core components that implement the most essential functionalities of a 3D virtual brush: a brush geometry model, a brush motion simulator, and a pigment behavior model. The following subsections give an overview of existing work with regard to these three core components.

1.1. Previous Work

1.1.1. Virtual brush geometry models

One of the earliest models was by Strassmann [23], where brush strokes are created by sweeping a 1D brush bristle over a skeleton. It is effectively a 2D heuristic approach which is not natural to use for non-computer specialists. Wong and Ip's virtual brush is modeled as an inverse cone and can produce an elliptic drawing mark [8]. This represents a substantial improvement over Strassmann's in terms of usability. However, because of the need to manually specify an intricate set of parameters controlling the shape, density and opacity of the current brush drawing mark, the interactivity of Wong and Ip's system is limited. In the DAB project [4], a subdivision surface is wrapped around a spring-mass particle system skeleton to represent the brush geometry. One disadvantage of using a subdivision surface is that the splitting of the brush is very hard to model. To generate the subdivision surface to model the brush head, either interpolation or some approximating scheme is used. An interpolated brush

head, however, often cannot deform smoothly because of the frequent occurrence of high curvature in the brush head surface, while approximation has the problem of properly placing control points to yield the desired surface. On top of all this, how to suitably anchor the control points of the surface to the mass particles is a non-trivial problem. In the work by Xu et al. [24], general sweeping is employed to establish the solid geometry model of each hair cluster. The problem however is that general sweeping is a time-consuming operation. At different stages of painting, much computation is needed to apply general sweeping operations to update the model. Also, the solid model requires a fair amount of memory for its internal representation, and after the brush is split many times, the demand for memory could become a bottleneck. In the work by Chu and Tai [15], a single hair bundle is modeled by a geometry model that is mathematically equivalent to that by Xu et al. [24]'s. Unlike the latter which simulates the spreading and splitting of the brush tip by a geometry approach, Chu and Tai use an alpha map to implement cluster modeling for the split brush. This results in an over simplification which limits the expressiveness and the amount of fine details that the brush is able to produce.

It appears that none of the above is sufficient to model a physical brush with a high degree of likeness to the brush's real behavior. One frequently occurring situation in reality, for example, is that of the brush splitting into a large number of hair clusters. To model this rather "chaotic" situation is out of reach for all the existing brush geometry models. According to many practising artists, such a high degree of splitting is very much desirable. In this paper, we propose a hierarchical geometry model for a "realistic" brush; together with a fast rendering procedure, our model enables efficient simulation of the most complex effects of a real brush, such as brush splitting.

1.1.2. Virtual brush motion simulators

Simulating the motion and dynamic deformation of a real paintbrush is a tough problem because of the complex static geometry and dynamics of the modeled brush.

In DAB [4], the motion of the brush geometry is simulated by Newtonian dynamics using a pair of first-order differential equations and Aristotelian dynamics through a single first-order differential equation. Solving differential equations involves some tradeoff between system stability, computation efficiency and simulation accuracy. In fact, the brush is a heavily damped system. When the penholder is motionless, the brush geometry stops deforming. Unfortunately, such a large damping force in Newtonian dynamics is hard to estimate. The authors of DAB therefore apply constraint satisfaction at the end of each step of motion simulation to ensure that the resultant brush geometry is a plausible configuration. This gives rise to the issue of simulation preciseness versus system stability. Unlike other classical motion simulation approaches which care a lot about simulation accuracy, the approach by Xu et al. [24] aims at providing a user with a customized virtual hairy brush. Their brush motion and brush geometry deformation are controlled by a set of "quality parameters" which, through a machine training module, can be customized for different users. These quality parameters affect the deformation process of a virtual hairy brush, and are not meant to be directly set or adjusted by the user. Unless the user can provide enough samples to train the system, he/she could only accept the default values for these quality parameters. The performance of the machine training process depends on the number of samples the user would input into the system. For satisfactory customization result, a large number of samples is often needed. The dynamics of Chu and Tai's brush [15] are modeled as springs, which are deformed through constrained energy minimization. This method of modeling can simulate small-scale deformation of the brush geometry but not large-scale bending or stretching due to the restriction of constrained energy minimization.

In this paper, the simulation of the motion and dynamic deformation of our hierarchical virtual brush geometry model is performed through a collaboration between online and offline computations. We minimize on the amount of online computation, and the result is calibrated by an error calibration mechanism using a simulation error database that was constructed offline. Compared with other virtual brush systems, we can simulate the brush motion with high precision using just a small amount of computation on the fly.

1.1.3. Pigment behavior models

Pigment behavior simulation presents yet another challenge to virtual brush system designers. One approach to simulating the ink's behavior is through a cellular automaton [18]. A classic work on pigment model is that by Curtis et al. [6], which studies the complex interaction behind pigment mixing. Their approach can synthesize very realistic water-color effects. During digital painting, the main loop of their algorithm iterates through several sub-processes simulating pigment behavior in a sequential manner. One sub-process will promote or restrain another sub-process. In a real painting process, however, these four sub-processes have subtle interaction, and therefore simulating the sub-processes separately would not necessarily add up to accurately simulating the pigment behavior. Another problem with their approach is that solving all the complex differential equations is too time-consuming for an interactive painting system. Other existing pigment behavior models include Lee's [11] for black-white painting and Kunii et al.'s [26] for diffused ink painting.

In Strassmann's hairy brushes [23], the color of brush strokes can be varied along the stroke's skeleton direction. In Wong's virtual brush [8], only monochrome calligraphic writing is supported. In DAB [4], a bidirectional, two-layer paint model is employed. They use essentially an additive compositive formula to mix the paint. In their paint representation, they assume that the paint surface contains two

layers. One layer is completely wet where paint would transfer out and another is completely dry. This over simplification of paint behavior makes the system incapable of achieving exquisite artistic effects demanded by the most seasoned painters. In Xu's virtual brush [24], a simple alpha blending strategy is employed to perform the pigment mixing.

Pigment mixing is a very important factor that contributes to the final results of painterly rendering and therefore deserves a very careful treatment. An ideal pigment behavior model should lead to fast response time in system implementation such as those described in [23, 8, 4, 24], and highly realistic results as demonstrated in [6]. In this paper, we liken the complex pigment's behavior in a painting process to that of heat or moisture, and model the pigment's behavior as a diffusion process. We introduce a strategy to divide the overall computation into an offline and an online part, therefore solving the complex differential equations modeling the pigment's behavior accurately and stably in real time.

There are many other fancy features that previous virtual brush systems are capable of, such as the ability to do 3D painting [17, 12, 16]. The DAB system [4] has haptic feedback. The system by Wei et al. [28] uses an intelligent model to beautify the output. Chan et al. [5] create 3D Chinese painting animation using existing commercial software packages.

1.2. Our Virtual Brush

The distinct features of our design for a realistic virtual brush include: (1) hierarchical modeling of the complex geometry of real paintbrushes, which leads to substantial savings in computation and representation in many steps of the simulation; (2) division of overall brush motion simulation into small online tasks and offline calibration, which makes possible an accurate and stable simulation of the brush's motion in real-time; (3) modeling of pigment behavior at dipping time and during painting as a diffusion process of random molecules subject to a certain physical condition, which gives rise to a new expressive pigment model that can simulate the most delicate and complex pigment behavior; and a strategy of solving the diffusion equations through a look-up table, which makes the numerical computation fast and stable; (4) a user manipulability adaptation component, which lets our system cater to the personal painting habits of different users.

Section 2 presents the hierarchical geometry model. Section 3 discusses the brush motion simulator and the division into online and offline tasks. Section 4 explains the pigment behavior model. Section 5 briefly introduces the intelligent self-adaptive component for improving user manipulability.

2. Geometry Model

To model a paintbrush with the granularity of a single hair thread which is done in Wong and Ip's system [8] is inefficient because a typical real brush may consist of thousands or even tens of thousands of individual hair threads. To overcome this inefficiency, Xu et al. [24] model a brush as clusters of brush hair. Their approach however can only handle the case of a brush splitting into a small number of hair clusters but not one with heavy splitting, for reasons explained in Section 1.1.1.

Figure 1: *Some complex brush geometries of our virtual brush.*

In this paper, we propose a two-level hierarchical brush geometry model. At the lower level of the hierarchy, hair threads whose position and geometry in 3D space are close to each other are gathered together and modeled as one *hair macro*. At the upper level, disjoint hair macros whose geometries are similar are classified into the same *cluster of hair macros*. The motivation behind having two levels is to eliminate as much as possible the redundancy in representing and simulating the brush hair. In real actions, a brush can easily split into thousands of disjoint clusters of hair threads. Cluster-based modeling alone as in Xu et al.'s approach [24] is not sufficient to deal with the highly chaotic geometry of the brush. It can be easily observed that even in such a situation, there exist only a few sharply distinctive geometries among all the geometries of hair clusters. We call these distinct geometries *primitive geometries*. With these primitive geometries, the geometries of all the hair clusters can be approximately derived via simple affine transformations. Figure 1 shows some complex brush geometries of our virtual brush. Based on this two-level hierarchical representation, our virtual brush model can efficiently represent the complex geometry as well as simulate the dynamic behavior of real paintbrushes having thousands of disjoint hair clusters.

2.1. Geometry Model of a Single Hair Macro

2.1.1. Hair macro

A hair macro represents the smallest granularity in the modeling. Hence, the overall modeling capability of the system hinges upon the modeling power of a hair macro.

Strassmann swept a 1D brush bristle over a skeleton to create the geometry model for brush strokes [23]. We go one step further by sweeping an ellipse along a skeleton in 3D space to create the model of a hair macro. During sweeping, we make sure that the ellipse always lies on the normal plane

with respect to its skeleton. The shape of the moving ellipse can be varied during sweeping.

Denote the skeleton of one hair macro as $C(t)$ $(0 \le t \le 1)$, which is a B-spline in 3D space, and the ellipse $E(t_0)$ lying on the normal plane of $C(t)|_{t=t_0}$ as

$$E(t_0) \triangleq \{(x,y)|x = A(t_0)\nu\cos\theta,\ y = B(t_0)\nu\sin\theta \quad (0 \le \theta < 2\pi, 0 \le \nu \le 1)\}, \tag{1}$$

in which $A(t_0)$ and $B(t_0)$ are half of the lengths of the major and minor axes of the ellipse $E(t_0)$ respectively. Here $A(t)$ and $B(t)$ $(0 \le t \le 1)$ are two one-dimensional B-splines. By controlling the three B-splines, $A(t), B(t), C(t)$, the user can tailor the geometry of a hair macro. The definition for the geometry of hair macro can therefore be written as

$$H \triangleq (A(t), B(t), C(t)) \ (0 \leqslant t \leqslant 1). \tag{2}$$

2.1.2. Cluster of hair macros

We use clustering to achieve efficiency in modeling the highly complex brush geometry, whereby hair macros whose geometries are similar are grouped into the same cluster. The internal representation of a cluster consists of two parts: the primitive geometry of the cluster, and some affine transformations each of which to be used to derive the geometry of a constituent hair macro based on the primitive geometry. This design makes the brush geometry model quite compact in memory, and facilitates fast simulation of the brush's actions at runtime.

2.2. Fast Rendering of Geometry

Visual feedback is important for any interactive system. But providing a good perceptual feedback may cost too much computation, especially for a realistic virtual brush whose static and dynamic geometry could be highly complex. Our hierarchical modeling helps tackle the situation.

All the hair macros of the same cluster not only share the same data structure but also a unique multiresolution tessellation process. As discussed in Section 2.1.2, each hair macro belonging to the same cluster can derive its specific geometry by applying the corresponding affine transformation to the cluster's primitive geometry. We need therefore only to tessellate the primitive geometry once for all the hair macros in the cluster. Our tessellation process is multiresolutional—that is, the density of the collection of triangles generated is made proportional to the displayed screen size. The granularity of the tessellation is set to be the smallest one needed to render the surface of a hair macro in the cluster. We also take advantage of the hardware accelerated OpenGL command "Display Lists" when computing the affine transformation and for the rendering. The result of all this is that our virtual brush can give very realistic visual feedback on the complex geometry of the virtual brush in action with a reasonably small amount of computation.

To sum up, we can derive all the specific geometries of all the hair macros of one cluster based on one primitive geometry. By this method, with less than say 10 primitive geometries, we can model efficiently the complex geometry of a realistic split brush even when the number of separate hair clusters goes beyond 10000, as confirmed by our experiments. Compared with the solid model representation and single-level approach by Xu et al. [24], our hierarchic modeling leads to much reduced memory consumption and avoids redundant computations.

3. Motion Simulator

Figure 2: *Dynamic deformation of a primitive geometry during painting.*

Figure 3: *Dynamic deformation of the virtual brush. (a) The initial geometry, (b) pressed down, (c) rotated, (d) tilted, (e) further pressed down.*

Our system performs a highly detailed simulation of the motion of the virtual brush. Most other physically-based systems sacrifice simulation precision for system response time; as a result, some of the important physical factors that are part of the real physical process are neglected from the very beginning. Our design tackles the problem using a two-phase approach to simulate the dynamic deformation of the brush. The two phases are online computation and calibration using offline data. The approach achieves high realism and interactivity using relatively little computation. In the online simulation, only those computationally inexpensive and input-sensitive physical processes are simulated. In the second phase, the online simulated results are calibrated using offline data.

In the online simulation, based on the current geometry of the virtual brush, the brush's inner stress is estimated, which is used to determine the state of the brush for deformation and recovery purposes.

For calibrating the online simulation results, we constructed in advance a database of system simulation errors against a collection of samples of real brush motion. These offline-acquired data in the database are then used to calibrate the online simulation results. The calibration proce-

dure first looks for one or more situations in the database that are closest (i.e., within a certain predefined threshold) in terms of "system condition" to the current brush's state. Then by a simple linear interpolation, the procedure finds the correct calibration to be applied to the current simulation result. Here, "system condition" includes the current geometry of the hair macro and the differential of the system's six degrees of freedom (DOFs), to be explained in Section 3.5. Figures 2 and 3 show the dynamic deformation of a primitive geometry and our virtual brush respectively.

3.1. Estimating Stress in Hair Macro

We adopt a simple strategy to estimate the inner stress of a hair macro. By Equation 1, the geometry of a hair macro is modeled as the volume of sweeping a variable ellipse $E(t)$ along a B-spline. We denote the ellipse that lies on the normal plane at the point $C(t_0)$ in the steady geometry of the hair macro as $E_s(t_0) = \{(x_s, y_s) | x_s = A_s(t_0)\nu_s \cos\theta_s,\ y_s = B_s(t_0)\nu_s \sin\theta_s\ (0 \le \theta_s < 2\pi;\ 0 \le \nu_s \le 1)\}$. Here the subscript s indicates that the brush is in its steady state. Suppose after a certain deformation of the virtual brush, the ellipse becomes $E_d(t_0) \triangleq \{(x_d, y_d) | x_d = A_d(t_0)\nu_d \cos(\theta_d),\ y_d = B_d(t_0)\nu_d \sin(\theta_d)\ (0 \le \theta_d < 2\pi;\ 0 \le \nu_d \le 1)\}$. The subscript d indicates that the brush model is deformed. Between the ellipses $E_s(t_0)$ and $E_d(t_0)$, we can establish a point to point correspondence according to their coordinates in the (ν, θ) space. Two points (ν_s, θ_s) and (ν_d, θ_d) are correlated if and only if $\nu_s = \nu_d$ and $\theta_s = \theta_d + \tau$, where τ is a parameter to be determined later. Correlated points are supposed to represent the same hair thread inside the hair macro during painting. The thread has a displacement from its position (x_s, y_s) in the steady geometry to its position (x_d, y_d) in the deformed geometry, where the origins of the local coordinate systems are the centers of the ellipses $E_s(t_0)$ and $E_d(t_0)$ respectively. We can then compute the stress $\rho(x_d, y_d)$ at point (x_d, y_d) in the ellipse $E_d(t_0)$ as

$$\rho(x_d, y_d) \triangleq \begin{cases} \tan(\max\{k_1\varphi, -\frac{\pi}{2}\}) & \text{when } \varphi \le 0 \\ \tan(\min\{k_2\varphi, \frac{\pi}{2}\}) & \text{when } \varphi > 0 \end{cases}. \quad (3)$$

Here, $\varphi \triangleq 1 - \sqrt{\frac{x_s^2 + y_s^2}{x_d^2 + y_d^2}}$, and k_1, k_2 are two positive real numbers that control the force response property of the virtual brush. The assumption we use in this estimation is that the stress in any steady geometry of a hair macro is always the same, which is a simplified conclusion from material science [27].

Based on this point-wise stress estimation, we can derive the formulae to evaluate the average inner stress $\rho(E_d(t))$ of the ellipse $E_d(t)$ and the average stress $\rho(H_d)$ of the deformed hair macro H_d by integrating the point-wise stress over the ellipse's inner area and throughout the volume of the hair macro respectively. The positive ρ suggests that the stress is inward, which will compress the geometry of hair macro; whereas the negative ρ suggests that the stress is outward and will dilate the hair macro; zero ρ means the hair macro is in its steady geometry. In the above computation, the value of τ will affect the computation result. By our design, $\tau(t)$ is supposed to be a number between $[0, 2\pi)$ to make the absolute value of $\rho(E_d(t))$ minimum.

3.2. Deforming Geometry of Hair Macro

The brush's geometry deforms under the outer force due to the friction between the brush and the paper. According to our geometry model of hair macro ($H \triangleq (A(t), B(t), C(t))$, Equation 2), the online simulation of the brush's motion is carried out in two parts: deforming the skeleton of hair macro $C(t)$; and deforming the sweeping ellipses of the hair macro, namely $A(t)$ and $B(t)$.

3.2.1. Deforming of skeleton

The main constraint in deforming the skeleton $C(t)$ of a hair macro is that $C(t)$ cannot penetrate the virtual paper. If at some moment the skeleton does penetrate the virtual paper at the point $C(t_0)$, we will project the part of skeleton below the virtual paper, denoted as $C_b \triangleq C(t)\ (0 \le t < t_0)$, onto the virtual paper to get C_p. We will also rotate C_b around the point $C(t_0)$ by a minimum angle to get C_r on which no point penetrates the virtual paper. And then we perform a simple linear interpolation between C_p and C_r to get an intermediate curve C_i which is used to replace the part of C_b in $C(t)$. The reason behind this replacement can be found in the study of material science [27]. If the perceptual material performance of the hair macro in our virtual brush is purely rigid, C_p should be used to replace C_b because the hair macro will actually break at $C(t_0)$; if its perceptual material performance is purely flexible, C_r is the correct choice for this replacement. Since we model the material of hair macro as a kind of hybrid rigid-flexible material, C_i is a reasonable choice covering the two possible cases. The weight used in the linear interpolation to compute C_i is a coefficient representing the degree of rigidity-flexibility in terms of the material property of the hair macro. After the replacement, we also ensure that the curvature of the skeleton would not change too abruptly over a small neighbouring area $\delta(t_0) = (t_0 - \varepsilon, t_0 + \varepsilon)$ around $C(t_0)$ in $C(t)$. This curve smoothing is done by uniformly sampling $C(t)$ in the range of $[0, t_0 - \varepsilon]$ and $[t_0 + \varepsilon, 1]$ and then applying a B-spline curve fitting to those sampled points. Finally the resultant B-spline curve is used as the new skeleton for this hair macro, which ensures a smooth transition of the changing skeleton around point $C(t_0)$.

We also simulate the kinking up of the hair macro due to the friction between the paper and brush head. According to classic Newton force, high stress would induce larger friction. The larger the friction is, the slower the hair macro will move. This non-uniform displacement can cause local prolongation and condensement on the skeleton of the hair macro. Denote the points on the skeleton of the hair

macro touching the paper as $p_1, p_2, \cdots, p_n$. According to Equation 3, we can compute the stress at these points as $\rho(p_1), \rho(p_2), \cdots, \rho(p_n)$. Let the mean stress of those points be S_m. Suppose the displacement of the hair macro is **D**; then the displacement of point p_i on the skeleton would be $\frac{\rho(p_i)}{S_m} \times \mathbf{D}$. Thus the distances separating $p_1, p_2, \cdots, p_n$ will be changed. Extended distances will pull the skeleton while shortened ones will squeeze the skeleton. If squeezed excessively, the skeleton will kink up. Suppose the distances between points p_i and p_j in the original and the squeezed skeletons are l_o and l_s respectively. We introduce a circular arc R to replace the distance to simulate the effect of kinking up with R's arc length between p_i and p_j being l_o, and R's chordal length being l_s. Here the chord of R refers to the line segment that connects p_i and p_j and which lies on the virtual paper.

3.2.2. Deforming of ellipse

During dynamic deformation, those ellipses touching the virtual paper, $E(t)$'s, will be deformed. Possible deformations include: (1) rotation of the ellipse; (2) prolongation or condensement of the ellipse' major or minor axis, and (3) displacement of the centroid of the ellipse, ι. (1) and (2) will yield an inner stress $\rho(E(t))$ for the ellipse $E(t)$, estimated by the method discussed in Section 3.1. Our design picks the strategy of minimizing $\rho(E(t)) + k_3||\iota||$, with the hard constraint that the deformed ellipse cannot penetrate the virtual paper and the area of the ellipse cannot change, where k_3 is a positive real coefficient.

3.2.3. The degree of distensibility

The area of each sweeping ellipse in a hair macro represents the number of hair threads in the macro being modeled. There is another parameter, which we call "degree of distensibility", which affects this area. A high degree of distensibility suggests that the hair threads are not very densely packed inside the hair macro. For simplicity, we model the degree of distensibility β as a function of the stress in the hair macro H as follows.

$$\beta(H) \triangleq \frac{1}{2} - \frac{1}{\pi}\arctan(k_4 \times \rho(H)) \tag{4}$$

Since $\rho(H)$'s range is the whole real number axis, the range of $\beta(H)$ is $(0,1)$. Here k_4 is a positive coefficient controlling the sensitivity of the distensibility to the inner stress. For each hair macro H, if its previous and current degrees of distensibility are β_1 and β_2, all of H's major and minor axes will have a length which is $\frac{\beta_2}{\beta_1}$ times their original values.

3.3. Hair Macro Splitting

If there exists a point in a hair macro, whose stress is above the threshold of a maximum tolerable inner stress, the hair macro H will split. According to the way we compute the stress in H, discussed in Section 3.1, the distribution of the stress within H is continuous. This means if there exists one such point in H whose stress is greater than the splitting threshold, it is likely that the neighbouring points are also above the threshold. Therefore, our splitting procedure works on groups of points. For one specific sweeping ellipse, we need first of all to separate those points whose stress are above the threshold from the other points. If the number of connected points whose stress are above the threshold is very small, there will be no splitting. This is the effect of physical attraction between neighbouring hair threads. After the small group elimination, real splitting operation comes into the picture. The separated point groups will become two ellipses, the centers of which are the centroids of each group. To determine the length of major and minor axes of the ellipse, we impose two constraints: the area of the ellipse should be equal to the area of the group before splitting, and the ratio of the major axis to minor axis of the ellipse is equal to the ratio of the length to the width of the bounding box of the point group. The orientation of the major axis of the new ellipse is determined by the principle axis of the group. Splitting the hair macro takes place in a stress-descending order if multiple point groups meet the splitting criterion. After splitting H at the largest average stress point group, the stress inside H is recomputed. Splitting continues as long as the splitting condition holds.

3.4. Recovery from Deformation

To simulate recovery from deformation, we model the material of the hair macro as a kind of hybrid elastic-rigid material. According to material science [27], there exists a steady geometry for such kind of hybrid elastic-rigid material. If the inner stress of the material does not exceed its threshold of elasticity, the material remains in an elastic state, in which all the deformation can be recovered. In the elastic state, once the outer force exerted on the material is released, the brush geometry will return to its original steady geometry. If the stress keeps increasing so that it exceeds the elasticity threshold, the steady geometry will be "shifted". Under this circumstance, the brush geometry upon releasing will only recover partially. We implement this shifting by interpolating between the original steady geometry and the current deformed geometry. Beyond the threshold of elasticity, there is the threshold of rigidity. The deformed hair macro will not have any recovery at all if the stress exceeds this threshold.

3.5. Calibrating Simulation Results

As can be seen so far, all the dynamic brush geometry deformation is simulated by fast online operations. To ensure accuracy of the simulation with respect to a realistic brush, we introduce a procedure to calibrate the fast online simulation results. The procedure relies on data from a "simulation error database". These data were sampled previously using a real brush. There are two important advantages to this offline

database approach. First, we do not have to model the extremely complicated physical process of a brush's dynamic deformation/recovery with high accuracy. Second, we can avoid the very time-consuming and probably unstable numerical computation to solve differential equations on the fly if some truly powerful but complex equations can be established to simulate all the underlying detailed dynamics governing the brush's deformation. The brush deformation database operates at the level of hair macro in our two-level hierarchy. This reduces substantially the number of different cases that need to be separately sampled and stored in the database.

Since the deforming of a hair macro's geometry is according to the six DOFs of the input, the database is indexed by: (1) the differential of the virtual brush system's six DOFs ($\partial\mathbf{D}$) between the current and the previous simulated time slice; and (2) the current geometry of the hair macro $H \triangleq \{A(t), B(t), C(t)\}$ (Equation 2). The content of each record in the database is the corresponding transformation $\mathbf{T}$ on H's current geometry, namely $A(t), B(t), C(t)$. When collecting records to establish the calibration database, we choose the samples in such a way that no two similar brush dynamic deformation processes are sampled and stored.

The above keywords to index the database have many dimensions. We construct a manageable database by taking advantage of the inter-relationships among the fields of the keywords. The product of $A(t_0)$ and $B(t_0)$ reflects the area of the ellipse $E(t_0)$, and hence the number of hair threads in the hair macro under a certain distensibility. If the hair macro will not split, $B(t_0)$ can be computed from $A(t_0)$. Thus, we can use four control points to represent the geometry deformation on $A(t)$ and four control points for $C(t)$. We also assume there is no inter-relationship between the dynamic deformations on $A(t)$ and $C(t)$—i.e., changing either one of the two splines will not have any direct effect on the other spline. This assumption comes from the observation that there is only a weak relationship between deformations on $A(t)$ and $C(t)$ during a real brush's deformation. As a result, the whole database can be divided into two sub-databases: **DA** for calibrating the simulated dynamic deformation on $A(t)$, and **DC** for calibrating the simulated deformation on $C(t)$. For simplicity, we ignore the possible but minute deformation on the brush geometry introduced by the x,y displacement of the virtual brush. So only four of the six DOFs will affect the brush's geometry deformation process. We also assume that the effects these four DOFs have on the brush geometry are separable. Thus **DA** and **DC** are further divided into four sections, each of which is used to calibrate the error in the simulation of the geometry's dynamic deformation under the change of one DOF. For each of the sections in **DA** and **DC**, the index words are of low dimensions: one dimension for the varied DOF and the remaining dimensions for the control points of $A(t)$ in **DA** or $C(t)$ in **DC**.

During database retrieval, we find the several records in the database whose distances to the input index are within a certain threshold. The contents of the retrieved records are interpolated using the distances between the indices of the retrieved records and the input index as the weights. For database retrieval, the distance between two splines is defined to be the sum of the euclidian distances of their corresponding four control points.

With the way we organize the database as described above, the database in our prototype implementation contains a modest number of records (40). This number of records is already sufficient for performing a satisfactory calibration to correct/improve the online simulation results.

4. Virtual Brush with Pigment

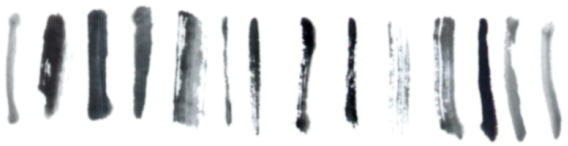

Figure 4: *Single strokes by the virtual brush with black pigment.*

4.1. General Pigment Diffusion Equations

Based on Fick's [1] diffusion analysis on a quantitative basis, Fourier's [9] research on heat conduction, and knowledge from the textile industry [14], we adopt Equation 5 to describe the diffusion of pigment in the virtual brush and using the nomenclature of vector analysis:

$$\frac{\partial \mathbf{C}}{\partial t} = \text{div}\ (D\ \text{grad}\ \mathbf{C}) \tag{5}$$

where $\mathbf{C}$ is the concentration of pigment at a certain location and time inside the hair macro, and D is the diffusion coefficient. In our design, D is supposed to be proportional to the stress of points where the virtual brush and paper touch and where diffusion takes place, namely $D \triangleq k_5\rho$, where k_5 is a coefficient, and stress ρ is evaluated by the method introduced in Section 3.1. After simulating the pigment diffusion process, a point on the virtual paper is colored by taking the color value of the pigment as its color value and $\min\{\frac{\mathbf{C}}{\mathbf{C}_{max}}, 1\}$ as its alpha value. Here, $\mathbf{C}_{max}$ works as a threshold. If the concentration of pigment at a point exceeds the threshold, the displayed color is the pigment's original color. Otherwise, some translucency is introduced to reflect this unsaturation of pigment's concentration. If multiple pigments are deposited at the same pixel, we interpolate the color values of all the pigments in the L,U,V color space to compute the color value. The interpolation weights are the percentages of each pigment's concentration against the overall pigment concentration at the point. $\min\{\frac{\mathbf{C}_a}{\mathbf{C}_{max}}, 1\}$ is used as the alpha value for the color in this multiple pigment

situation, where $\mathbf{C}_a$ is the overall pigment concentration at the point. In our virtual brush, diffusion is applied and simulated only when the concentration of pigment is not uniformly distributed.

4.2. Simulating Pigment Diffusion at Dipping Time

We model the physical process of brush dipping like the two-phase dying process in the textile industry. In the first phase, we only simulate the radial diffusion inside the hair macro. In the second phase, we compute the axial diffusion. In both phases of the diffusion process, the effects due to fluid dynamics are considered. Section 4.2.1 discusses the generic equation describing pigment diffusion inside hair macro; Section 4.2.2 and Section 4.2.3 discuss how to simulate the radial and axial pigment diffusion respectively inside the hair macro during brush dipping.

4.2.1. Equations for pigment diffusion

We view the geometry of the hair macro defined by Equation 2 as a deformed cylinder. Given the pigment's concentration in a normal non-deformed cylinder, we can use a simple affine transformation to map each voxel in the hair macro into the non-deformed cylinder. The curves of $A(t)$, $B(t)$ are mapped to the lines of $A(t) \equiv R$, $B(t) \equiv R$, and $C(t)$ is mapped as $C(t) = (0,0,t)$, where R is the radius of the non-deformed cylinder. By this simple mapping strategy, we only need to simulate pigment diffusion inside an ideal cylinder, to be discussed in the following.

Through transformation of coordinates by replacing (x,y,z) with $(r\ \cos\theta, r\ \sin\theta, z)$, the equation for diffusion in a cylinder can be derived from Equation 5 in terms of the cylindrical coordinates (r,θ,z) as follows.

$$\frac{\partial \mathbf{C}}{\partial t} = \frac{1}{r}\{\frac{\partial}{\partial r}(rD\ \frac{\partial \mathbf{C}}{\partial r}) + \frac{\partial}{\partial \theta}(\frac{D}{r}\ \frac{\partial \mathbf{C}}{\partial \theta}) + \frac{\partial}{\partial z}(rD\ \frac{\partial \mathbf{C}}{\partial z})\} \quad (6)$$

4.2.2. Radial diffusion of pigment

For this case, pigment concentration is simplified to be a function of radius r and time t only. Suppose R is the radius of the cylinder, $\mathbf{C}_0$ is the concentration of pigment in the ink bottle, and $f(r)$ is the initial concentration of pigment before the current diffusion, which is useful when multiple brush dippings are simulated. Crank [10] gives the analytic solution to Equation 6 under this condition as follows.

$$\mathbf{C}(r,t) = \mathbf{C}_0\{1 - \frac{2}{R}\sum_{n=1}^{\infty}\frac{1}{\alpha_n}\frac{\mathbf{J}_0(r\alpha_n)}{\mathbf{J}_1(R\alpha_n)}\exp(-D\alpha_n^2 t)\} + \frac{2}{R^2}\sum_{n=1}^{\infty}\exp(-D\alpha_n^2 t)\frac{\mathbf{J}_0(r\alpha_n)}{\mathbf{J}_1^2(R\alpha_n)}\int r\ f(r)\mathbf{J}_0(r\alpha_n)dr \quad (7)$$

where the α_ns are the positive roots of $\mathbf{J}_0(R\alpha_n) = 0$, in which $\mathbf{J}_0(x)$ is the Bessel function of the first kind of order zero, $\mathbf{J}_1(x)$ is the Bessel function of the first order, and D is the diffusion coefficient.

4.2.3. Axial diffusion of pigment

Solutions of the diffusion equation in this case can be derived as:

$$\mathbf{C}(z,t) = \frac{M}{2\sqrt{\pi Dt}}\exp(-z^2/4Dt) \quad (8)$$

where z is the axial distance for diffusion, and D is the diffusion coefficient with M being the overall pigment concentration in the diffusion source area.

4.3. Simulating Pigment Diffusion During Painting

We model the physical process of painting using the virtual brush as a dying process in which the pigment is continuously diffusing from the brush head onto the virtual paper at the points of touching. We assume the virtual paper to be a flat plane. In the simulation, we also take into account the effect due to fluid dynamics. For the ideal physical process, Crank [10] gives the solution to the the diffusion Equation 6 for the case of a continuous point source diffusing process—Equation 9. To realistically render the current footprint of the brush on the paper, we simulate the pigment's transformation at the points of touching between the brush and the paper, once for each point. Section 4.4 discusses how this part of the simulation can be accelerated. After the diffusion, the source point on the virtual brush has a decrease in concentration by the amount of the sum of the increase of pigment concentration on the virtual paper.

$$\mathbf{C}(r,t) = \frac{q}{4\pi Dr}erfc(\frac{r}{2\sqrt{Dt}}) \quad \text{where } erfc(x) = 1 - \frac{2}{\sqrt{\pi}}\int_0^x \exp(-\eta^2)d\eta \quad (9)$$

In Equation 9, q is the ink transfer coefficient which is a constant, D is the diffusion coefficient, and $\mathbf{C}(r,t)$ is the amount of pigment concentration change introduced by the diffusion process for a point at time t and distance r from the source point.

4.4. Fast Solutions to the Diffusion Equations

Although the analytic solutions to the general diffusion equation (Equation 6) can be derived as Equations 7, 8, and 9, evaluating the numeric values (for Equations 7 and 9) to satisfy any real-time requirement is a non-trivial task. Fortunately, both solutions, $\mathbf{C}(r,t)$, in Equations 7 and 9 are functions of two variables r and t. Thus we can generate a look-up table which enumerates the discrete combination of these two variables (r_i,t_j) under a certain diffusion coefficient D and their corresponding solution $\mathbf{C}(r_i,t_j)$. This look-up table is indeed an $R^3 \rightarrow R^1$ mapping. During online simulation, rather than computing the numeric values of Equations 7 and 9, we search the table, leading to much saving of CPU time. Suppose the current input to be evaluated is $\mathbf{C}(r_0,t_0)$, we retrieve four values from the table: $\mathbf{C}(r_0^+,t_0^+)$, $\mathbf{C}(r_0^+,t_0^-)$, $\mathbf{C}(r_0^-,t_0^+)$, $\mathbf{C}(r_0^-,t_0^-)$. Here r_0^- is the maximum value in the table which is smaller than r_0 and r_0^+ is the minimum value

in the table which is bigger than r_0. Similarly for t_0^- and t_0^+. We then use an interpolation to derive $\mathbf{C}(r_0,t_0)$ based on the values of $\mathbf{C}(r_0^+,t_0^+)$, $\mathbf{C}(r_0^+,t_0^-)$, $\mathbf{C}(r_0^-,t_0^+)$ and $\mathbf{C}(r_0^-,t_0^-)$.

5. Improving User Manipulability

As with any real brush, it can happen that the user would feel unsatisfied with their creations using the virtual brush. Instead of training the user, our virtual brush system has the unique feature of training the brush. This is done through an intelligent *user-manipulability improving component*. This component applies an additional transformation to the user's input before the system commits on the final painting result.

The idea behind the design of this component is as follows. Our virtual brush carries with it a collection of beautiful strokes and the input for creating these strokes. The user can choose among these "known" brush strokes not for his/her own strokes, but use them as training samples. For each selected sample, the user would then use our virtual brush to produce a stroke as close to the sample as possible. The system then applies a numerical analysis to compute a transformation from the user's input to that of the sample stroke. The derived transformation is the "personal habitual bias" of the user. Later, when the user paints using our virtual brush, his/her input will firstly go through the transformation of his personal habitual bias. Thus, it is the transformed input rather than his original input that drives the virtual brush to paint. The details are omitted due to the length of this paper. Please refer to the accompanying website for more technical details on this feature of our virtual brush.

6. Experiment Results

Figure 4 shows some single strokes by our virtual brush with black pigment. Figure 5 to Figure 7 show some real results from using our virtual brush. Most of them took hours to complete, but at any moment during the painting process, our virtual brush running on a PC with 256M memory and an AMD Duron 1200MHz Processor can respond interactively to user commands. Currently, we use a WACOM pen on a tablet to get the position, the pressure (used as vertical displacement), and the tilt of the virtual brush; and keyboard input to get the remaining two DOFs. This input strategy is very much the same as the one used in Xu et al.'s brush system [24]. A better input device in the future should make the painting process much smoother and more efficient.

7. Conclusion

We have presented the design of a powerful painting virtual brush system. We targeted at a realistic modeling by covering as much minute details of the brush's geometry and motion as well as the delicate pigment behavior taking place during the painting process as possible. Such an approach has made necessary the many time or space optimizations

Figure 5: *"Summer water lily" by the virtual brush.*

Figure 6: *"Quiet village" by the virtual brush.*

Figure 7: *"Spring garden" by the virtual brush.*

that we have introduced into the design. The result is a high degree of realism as can be seen in the generated digital artwork.

There are still many interesting problems to be addressed in the future. One of them is that of choosing the appropriate samples for the brush motion calibration database that can enumerate all the possible motions a painting brush could experience without repetitive sampling. For the currently employed pigment model, only water-soluble pigment

is simulated. An obvious future task would be to extend the model to cover oil painting and maybe other kinds of painting by following Small's [7] and Cockshott's [25] pioneering approaches. For the user manipulability improving component, finding the features that distinguish good input leading to visually pleasing brush strokes from bad input could help establish a mechanism to perform auto-beautification.

References

1. A. Fick. *Annln Physics* **170**(59), 1855. 7
2. A.R. Smith. Digital Paint Systems: An Anecdotal and Historical Overview. *IEEE Annals of the History of Computing*, 2001. 1
3. A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. *Proc. of SIGGRAPH '98*:453–460, 1998. 1
4. 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. 1, 2, 3
5. C. Chan, E. Akleman and J. Chen. Two Methods for Creating Chinese Painting. *Proc. of Pacific Graphics '02*: 403–412, 2002. 3
6. 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. 2, 3
7. D. Small. Simulating Watercolor by Modeling Diffusion, Pigment, and Paper Fibers. *Proc. of SPIE '91*, February 1991. 10
8. 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. 1, 2, 3
9. J.B. Fourier. *Theorie analytique de la chaleur*, 1822. English translation by A. Freeman, Dover Publ., New York, 1955. 7
10. J. Crank. *The Mathematics of Diffusion*, Oxford Press, 1975. 8
11. J. Lee. Simulating oriental black-ink painting. *IEEE Computer Graphics and Applications* **19**(3):74–81, 1999. 2
12. M. Agrawala, A.C. Beers and M. Levoy. 3D painting on scanned surfaces. *Proc. of the 1995 symposium on Interactive 3D graphics*:145–150,215, 1995. 1, 3
13. M.P. Salisbury, S.E. Anderson, R. Barzel and D.H. Salesin. Interactive pen-and-ink illustration. *Proc. of SIGGRAPH '94*:101–108, 1994. 1
14. M.R. Gregor. *Diffusion and Sorption in Fibre and Films, Vol. 1. Introduction with particular reference to dyes*. Academic Press, London, New York. 1974. 7
15. N.S.H. Chu and C.L. Tai. An efficient brush model for physical-based 3D painting. *Proc. of Pacific Graphics '02*. 1, 2
16. Pixologic. Z-brush, http://pixologic.com, 2000. 1, 3
17. P. Hanrahan and P. Haeberli. Direct WYSIWYG painting and texturing on 3D shapes. *Proc. of SIGGRAPH '90*:215–223, 1990. 1, 3
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 rendering of trees. *Journal of Visualization and Computer Animation*, 1999. 2
19. R.D. Kalnins, L. Markosian, B.J. Meier, M.A. Kowalski, J.C. Lee, P.L. Davidson, M. Webb, J.F. Hughes and A. Finkelstein. WYSIWYG NPR: drawing strokes directly on 3D models. *Proc. of SIGGRAPH '02*:755–762, 2002. 1
20. S.C. Hsu and I.H.H. Lee and N.E. Wiseman. Skeletal strokes. *Proc. of SIGGRAPH '93*:197–206, 1993. 1
21. S.C. Hsu and I.H.H. Lee. Drawing and animation using skeletal strokes. *Proc. of SIGGRAPH '94*:109–118, 1994. 1
22. S. Saito and M. Nakajima. 3D physics-based model for painting. *SIGGRAPH '99 Sketches, Conference Abstracts and Applications*:226–226, 1999. 1
23. S. Strassmann. Hairy brushes, *Proc. of SIGGRAPH '86*:225–232, 1986. 1, 2, 3
24. S. Xu, M. Tang, F.C.M. Lau and Y. Pan. A solid model based virtual hairy brush. *Computer Graphics Forum (Proc. of Eurographics '02)*, **21**(3):299–308, 2002. 1, 2, 3, 4, 9
25. T. Cockshott. Wet and Sticky: A novel model for computer based painting. PhD. Thesis, Computing Science Department (Research Report 91/R20), University of Glasgow, 1991. 10
26. T.L. Kunii, G.V. Nosovskij and T. Hayashi. A diffusion model for computer animation of diffuse ink painting. *Computer Animation*, 1995. 2
27. W.D. Calister and W.D. Callister Jr. *Materials Science and Engineering: An Introduction*, Wiley, John & Sons, Incorporated, 1999. 5, 6
28. X. Wei, S. Lu, M. Song and B. Luo. Computer pattern design and painting technique based on aesthetics knowledge. *Computer Aided Drafting, Design and Manufacturing*, **2**(2):32–40, 1992. 3