DAB: Interactive Haptic Painting with 3D Virtual Brushes

Bill Baxter Vincent Scheib Ming C. Lin Dinesh Manocha
Department of Computer Science
University of North Carolina at Chapel Hill
{baxter,scheib,lin,dm}@cs.unc.edu
http://www.cs.unc.edu/~geom/DAB

Abstract: *We present a novel painting system with an intuitive haptic interface, which serves as an expressive vehicle for interactively creating painterly works. We introduce a deformable, 3D brush model, which gives the user natural control of complex brush strokes. The force feedback enhances the sense of realism and provides tactile cues that enable the user to better manipulate the paint brush. We have also developed a bidirectional, two-layer paint model that, combined with a palette interface, enables easy loading of complex blends onto our 3D virtual brushes to generate interesting paint effects on the canvas. The resulting system, DAB, provides the user with an artistic setting, which is conceptually equivalent to a real-world painting environment. Several users have tested DAB and were able to start creating original art work within minutes.*

Keywords: Haptics, Human Computer Interaction, Painting Systems, Deformable Brush Model

1 Introduction

The *art* of painting refers to the aesthetic aspects of a painterly work. The *craft* of painting deals with the study of materials, including paint medium, tools, supports, and methods, i.e. the manipulation of materials to express an artist's intent and purpose [May70]. The art and craft of painting are closely related: an artist cannot divorce one from the other. Nevertheless, recent technological advances in computer graphics have largely centered around the *art* of painting, with little attention being given to the *craft*.

Commercial painting systems and recent research on the generation of painterly works have mainly emphasized the appearance of the final product. However, the word 'painterly' also describes a fusion of feeling and action, sight and touch, purpose and paint, beyond merely producing an image that gives an artistic impression [May70].

Rather than focus primarily on the rendered appearance, there may be equal merit in recreating the "sight, touch, action and feeling" of the artistic process itself. By designing a setting for artists to freely and creatively express themselves, as they would in a traditional painting environment, computer graphics can serve as a conduit to the craft as well.

1.1 Main Contribution

Our primary goal is to provide an expressive vehicle for interactively creating original painterly works with computer systems. We present a physically-based, deformable 3D brush model, which gives the user control of complex brush strokes intuitively. The haptic feedback enhances the sense of realism and provides tactile cues that enable the user to better manipulate the paint brush. We have also developed a bidirectional, two-layer paint model that, in combination with a palette interface, enables easy loading of complex blends onto our 3D brush model and generates interesting paint effects on the canvas.

Figure 1: *An original work created using DAB. (Rebecca Holmberg, artist)*

We have attempted to provide a minimalistic interface that requires as few arcane buttons, key-presses, and complex controls as possible, yet still offers a great deal of expressive power. With our haptic painting system, *DAB*, most paintings can be created with just the force-feedback device and the space bar on the keyboard. In comparison to the existing computer painting programs, our approach offers the following advantages:

- Natural and expressive mechanisms for manipulating the painting tools, including brushes, palette, paint and canvas;
- Simple and easy loading of complex blends using 3D virtual brushes;
- Physically-based and realistic brush footprints generated automatically by the brush strokes;
- Intuitive and familiar feel of the painting process requiring little or no training.

Our haptic painting system, *DAB*, has been tested by a number of users. A novice user can start painting with just a few (typically less than ten) minutes of simple instruction. Fig. 1 shows a painting created by an amateur artist with *DAB*. Since *DAB* provides a familiar setting, conceptually equivalent to a real-world painting

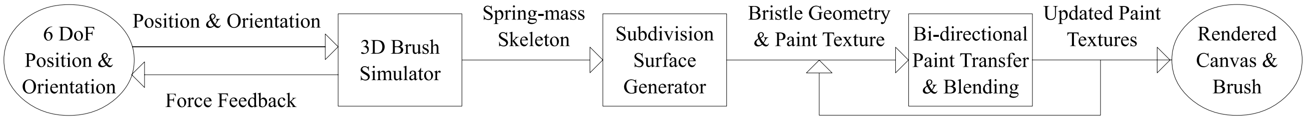


Figure 2: *System Architecture*

environment, an artist need only control the virtual brush as he or she would a real brush. This interface could also be combined with most of the existing interactive painting programs or used as an effective training system for painting.

1.2 Prior Work

Computer-Generated Painting: A number of researchers have developed automatic methods for transforming ordinary images into painterly or otherwise imaginative renderings [Her98, Lit97, Mei96]. Others have developed 2D methods for simulating the look of painting, from Alvy Ray Smith's original "Paint" program [Smi78] to more recent physically-based approaches [CPE92, CAS$^+$97]. Commercial packages such as COREL's Painter [COR00] are able to achieve realistic looking simulations of natural media by clever use of 2D textures and compositing tricks. The amount of training required to proficiently use these commercial painting systems is large, as is the complexity involved in obtaining the precise strokes desired, even for skilled painters.

Modeling of Brushes: Several researchers have endeavored to accurately model the *appearance* of real brush strokes, but most techniques have been 2D heuristics. Strassmann modeled a brush as a one-dimensional array of bristles swept over a trajectory defined by a cubic spline curve [Str86]. This work was able to account for a number of effects achievable with an actual brush, such as varying color, width, and wetness. Wong and Ip [WI00] defined a complex set of interrelated parameters to vary the density, opacity, and shape of a footprint in a way that takes into account the behavior of a three-dimensional round calligraphy brush. The resulting stroke appearances are *informed* by the physical behavior of the brush, but are not actually physically generated. The method as described is only partially interactive.

Our approach for brush modeling shares some similar themes with the work of Saito [SN99] on modeling a physical 3D brush for Japanese calligraphy and *sumie* paintings. However, our technique is more flexible in terms of brush shape, dynamics, and loading, and is able to take advantage of 3D graphics hardware as well.

User Interface: Hanrahan et al. allowed the user to paint directly onto a 3D model by using standard graphics hardware to map the brush from screen space onto the model [HH90]. Commercial systems, such as Z-Brush [Pix00] and Deep Paint [hem00], also allow users to paint directly on surfaces, but this is accomplished with standard 2D brush footprints that are projected onto the surface of the 3D object. The brush itself is not three-dimensional.

Several of the more advanced commercial tools, e.g. Painter, support pen-based input with sophisticated 5-DOF tablet devices, yet most still use only the position and pressure parameters and ignore the tilt. Further discussion on tablet systems is given in Section 6.

The idea of 3D painting has been explored in [ABL95, JTK$^+$99, GEL00] using a simple, rigid 3D brush (tip) controlled by a 6-DOF input device to color 3D surfaces. All these 3D painting systems were restricted to monochrome brushes.

1.3 Organization

The rest of the paper is organized as follows. Section 2 gives an overview of our approach and the user interface. We present the modeling of the paint brushes in Section 3 and force display using haptic devices in Section 4. Section 5 describes our techniques for rendering acrylic or oil-like paint. Next, we briefly highlight the implementation of our prototype painting system with a haptic interface and demonstrate its features via the actual paintings of several volunteers in Section 6.

2 Approach

In this section we give an overview of our approach and the user interface of our haptic painting system.

2.1 Overview

We have developed a novel, physically-based, deformable 3D brush model integrated with a haptic interface. The haptic stylus serves as a physical metaphor for the virtual paint brush. It takes in the position and orientation of the brush and displays the contact force between the brush and the canvas to the user. The bristles of the brush are modeled with a spring-mass particle system skeleton and a subdivision surface. The brush deforms as expected upon colliding with the canvas. This framework allows for a wide selection of brush types to be made available to artists.

Our multi-layered paint model supports important features of paint, such as bidirectional paint transfer, blending, drying, and complex brush loading. The surfaces of the brush, canvas, and palette are coated with paint using this model. A schematic diagram is shown in Fig. 2 to illustrate how various system components are integrated.

2.2 User Interface

We use a SensAble Technologies' PHANToM as a haptic device and a dual-processor Pentium III PC with NVIDIA's GeForce2 graphics card. One processor is dedicated to force display and the other is used to compute the brush dynamics and the paint transfer and blending. Fig. 3 shows the physical setup of our system.

Figure 3: *Haptic Painting System Setup: An artist using a haptic stylus to paint directly on the virtual canvas using DAB.*

Our haptic painting system allows the user to paint directly onto a virtual canvas displayed on the screen. Using the space bar as a toggle, the user can bring up the virtual palette for paint mixing and brush cleaning, or put the palette aside to paint directly onto the canvas. The user is also presented with a wide selection of virtual brushes that mimic different types and shapes of brushes used in traditional painting. A simple menu is presented for saving and loading a clean or previously painted canvas, undoing a brush stroke, quickly drying the canvas partially or completely, etc. Fig. 4 shows a snapshot of our graphical user interface, consisting of the virtual brushes, the palette, and the canvas.

The paint brush deforms in a natural, physical way, as the user moves the brush across the canvas. The user can create strokes with the brush, which behaves much in the way a real brush would. The actual footprints of the brush and resulting strokes are generated based on the user's manipulation of the 3D brush on the virtual canvas.

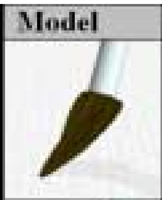
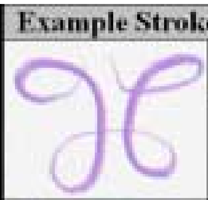
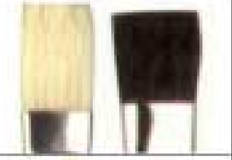
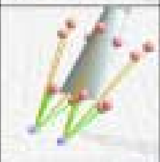
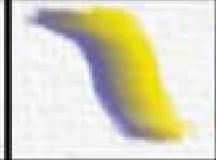
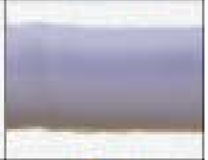
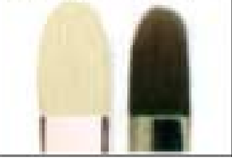
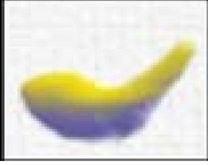
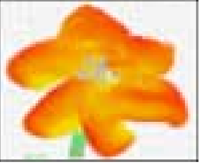
Type	Examples	Model	Structure	Surface	Example Strokes	
Round						
Flat/ Bright						
Filbert						

Table 1: *We show some real brushes, our model for each (skeletal structure and surface mesh), and example strokes generated with each.*

Figure 4: *Graphical User Interface: The virtual canvas with the brush rack and a portion of the palette (LEFT); the brush rack and the palette for color mixing (RIGHT).*

3 Modeling of 3D Brushes

Paint brushes are often regarded as the most important tools at an artist's disposal. A good set of brushes can enable a competent artist to create virtually any effect he or she can imagine, from the intricate detail of cresting waves, wispy billowing clouds, to the subtly blended shifting hues in a sunset. In this section, we describe our techniques for modeling 3D virtual brushes.

3.1 Introduction to Brushes

Figure 5: *Basic Brush Anatomy*

Fig. 5 shows the anatomy of a typical brush. Brush heads are made with a variety of bristles, natural soft animal hair, and synthetic materials. Some of the most common styles for brushes used in oil-like painting [May70] are:

- **Rounds.** Have a simple tubular shape with a semi-blunt point, allowing for a great variety of strokes.
- **Flats.** Thinner and wider than rounds with bristles squared off at the point. Flats are typically longer than they are wide.
- **Brights.** The same shape and construction as flats but typically shorter, with width nearly equal to length.
- **Filberts.** Have a thicker collection of bristles that increase ability to hold paint. Filberts usually have oval-shaped heads.

There are other types of specialty brushes, such as fans and blenders, but the four above are the most versatile and widely used. The second column of Table 1 shows images of each type.

3.2 Overview of Modeling Approach

To model a 3D paint brush requires developing both a geometric representation and a physics-based model for its dynamic behavior. The requirements of an interactive haptic painting system place constraints on the design: the brush dynamics must run at interactive rates and remain stable under all types of user manipulation.

We model the brush head as a subdivision surface mesh wrapped around a spring-mass particle system skeleton. The particle system reproduces the basic motion and behavior of a brush head, while the deformable mesh skinned around this skeleton represents the actual shape of the head. We also derive an approximated implicit integration method based on an existing numerical technique for cloth simulation [DSB99] to take large integration steps while maintaining stability. Although our brush model may appear simplistic at first, it is designed to capture the essential quality of physical brushes to maintain interactivity at minimal computational costs.

Based on our generalized 3D brush model, we are able to adjust some key parameters to generate the different types and shapes of brushes and mimic their physical behavior. In Table 1, we show the geometric structure used to construct each of the brushes described in Section 3.1. We also show the deformation of different brushes as they make contact with the canvas.

3.3 Brush Dynamics

The difficulty in simulating the paint brushes used in acrylic and oil-like painting is that the brushes are numerically stiff dynamical systems, and suffer from numerical instability. Bristles have very little mass. As they bend, energy stored in them can induce large accelerations and velocities when they are abruptly released. The brushes also behave as highly damped systems and we use this property to improve the stability of our solver.

We have considered and evaluated several numerical methods for particle system simulation, but we found the approximated implicit integrator presented in [DSB99] to be most effective for this application, primarily because of its stability. We simulate some brushes with the approximate implicit integrator and others with a variation based on first-order dynamics.

3.3.1 Newtonian Dynamics

The motion of the particle system representing the brush can be described mathematically by Newton's second law, a second order differential equation, decomposed here as a pair of coupled first-order differential equations:

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{v}} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \mathbf{M}^{-1}\mathbf{f} \end{pmatrix} \tag{1}$$

In this equation, **x** is a $3n$ vector containing the spatial coordinates of n particles, $\mathbf{v}$ is a $3n$ vector of particle velocities, and **f** is a $3n$ vector of the forces on those particles. **M** is a $3n \times 3n$ diagonal matrix whose diagonal entries are of the form $M_{ii} = m_{\lceil i/3 \rceil}$, where m_j is the mass of particle j.

The semi-implicit method for simulation of deformable objects [DSB99] approximates a solution to the equations of motion in three stages:

1. Implicit integration of linear force components
2. Approximate post-correction to account for non-linear force components
3. Deformation constraint enforcement to prevent excessive stretch

The resulting solution is much less accurate than other large-step integration techniques, such as that presented by Baraff and Witkin [BW98], but it is both more stable and computationally less demanding. The speed advantage comes from separating out the linear force component, which allows one to solve the equations of motion using just a matrix-vector multiply each step. The integration step has the form:

$$\begin{pmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{v} \end{pmatrix} = \begin{pmatrix} h(\mathbf{v}_0 + \Delta\mathbf{v}) \\ \left(\mathbf{I} - h\frac{\partial \mathbf{f}_l}{\partial \mathbf{x}}\right)^{-1} h\mathbf{M}^{-1}\mathbf{f}_l(\mathbf{x}_0) \end{pmatrix} \tag{2}$$

Since only the linear force components are handled by the integration step, $(\mathbf{I} - h\partial\mathbf{f}_l/\partial\mathbf{x})^{-1}$ becomes a constant matrix.

This method works well for cloth, which generally has weak bending forces, but for brush simulation, approximating the effect of the non-linear force components leads to local errors in angular momentum. We have observed that with some brush skeletons, the solver effectively ignores stiff bend forces, leading to brushes with too much angular motion. Achieving stiff brush behavior is possible, but depends upon the skeletal structure. Section 3.5 discusses our brush construction in more detail.

The final step in the method is the application of deformation constraints. In this step, particles are assumed to have traveled in the right direction, but perhaps too far, inducing excessive stretch in the material. This is corrected by simply altering the positions of particles, iteratively contracting the springs in the material until overstretch is eliminated. The deformation constraints play a major role in the overall stability of the system by ensuring that at the end of every step, every spring is in a physically plausible configuration.

For collision handling and contact response, particles found to be penetrating the canvas are projected up to the nearest point on the surface. We also add a frictional drag force to colliding particles for more realistic results. We model the frictional drag as

$$\mathbf{F}_{\text{friction}} = -\mu \|\mathbf{F}_{\text{normal}}\| \mathbf{v}_{\text{tangential}},$$

where μ is the coefficient of friction.

3.3.2 Aristotelian Dynamics

Real bristles empirically obey the Aristotelian model of physics, which is characterized by the lack of inertia. In this model, objects move only for the duration that forces are applied. Inspired by [WB97], we use this simplified model to simulate most of our brushes. This has advantages for speed, stability, and in some cases usability. With the Aristotelian dynamics model, the motion of the particle system is represented by a single first-order differential equation: $\dot{\mathbf{x}} = \mathbf{M}^{-1}\mathbf{f}$.

Since objects now stop moving instantly in the absence of forces, the result is motion that appears heavily damped, which is precisely how we wish brushes to behave. In the second order model, to simulate this damping requires adding large damping forces to the system to cancel out the large velocities induced by stiff springs. Using a first-order physics model, however, we can circumvent this step entirely.

We modify the approximated implicit integration formula as follows for the first order model:

$$\Delta\mathbf{x} = \left(\mathbf{I} - h\frac{\partial \mathbf{f}_l}{\partial \mathbf{x}}\right)^{-1} h\mathbf{M}^{-1}\mathbf{f}_l(\mathbf{x}_0)$$

Since this equation is still in the same form as Eqn. 2, most of the integration technique remains unchanged. An exception is that we omit the frictional damping force during collisions and just modify velocity, since in the first order model the two have the same effect.

3.4 Brush Surface

We use subdivision surfaces as the geometric representation for the brush head because of their ability to represent arbitrary topology and vertices of arbitrary valence easily. The brush head subdivision surface is defined by control points anchored relative to the mass particles. It is possible to use either interpolating or approximating subdivision schemes for the brush surface.

An interpolating scheme eases the task of choosing reasonable control vertices for the rough mesh, since the limit surface is guaranteed to pass through each of them. In fact, since *all* vertices at *all* subdivision levels are on the limit surface, it also facilitates changing the tessellation level of the mesh. However, due to frequent appearance of high curvature in the resulting surface, often interpolating surfaces do not deform as smoothly as would be expected of a brush.

Approximating schemes generate surfaces that are generally smoother and fairer, but it is more difficult to place control points to achieve the desired surface. The extensions to the Loop approximating scheme presented by [HDD$^+$94] would be useful for accurately modeling sharp features like the finely tapered point of a round brush.

In our implementation we chose to use a triangular base mesh and to subdivide with the interpolating Butterfly rule to make the task of generating the brush control mesh easier. Some example results can be seen in Table 1.

3.5 Brush Generation

Given these dynamical models for simulation, we synthesize a full set of brushes suitable for creating a wide variety of paintings. One type of brush is modeled as a single spine composed of a linear chain of n particles. With our integration method and this structure, we are able to model the softer style of brushes used in Japanese calligraphy, called *fude*. Our *fude* brushes work best with the first order dynamics model, which makes the brush appear more solid by eliminating oscillations.

We model stiffer brushes, like those used in oil and acrylic painting, by using a more complicated skeletal structure. The basic building block for our stiff brushes is five mass particles connected with springs to form a pyramidal truss. The round brush consists of one of these trusses. The four particles that form the base are rigidly fixed to the brush handle and are directly driven by the user's input. The fifth particle serves as the point of the brush.

Table 1 shows a summary of the brush models and gives examples of the strokes that can be generated with each. Wide brushes are formed from two trusses, and filberts are generated from four of them, the outer two being shorter than the inner two. We use each brush structure to define an entire family of brushes of different sizes by parametrically varying the scaling along the three cardinal axes.

4 Haptic Display

An important aspect of our 3D painting system is the ability to provide sufficiently good force feedback to emulate the sensation of applying brush strokes to a canvas. Our 6-DOF armature input device also serves as a 3-DOF force output device. We align the virtual paintbrush with the physical 6-DOF stylus, and position it so that the point of 3-DOF force delivery coincides with the point where the head meets the handle on the virtual brush. In this section, we present our approach for force display.

4.1 Decoupled Haptics

We separate the force computation from the brush deformation computation, since the two have different goals. For instance, the non-dynamical deformation constraints used by the approximated implicit solver are acceptable for approximating the visual aspects of brush behavior, but are not appropriate for force simulation. Furthermore, the force updates for haptic feedback need to be generated at close to 1kHz for smooth jitter-free output, but the deformation calculation only needs to proceed at visual update rates (around 30Hz). Consequently we decouple the force simulation from brush dynamics simulation, and simplify the force computation to run at kHz rates.

4.2 Basic Force Model

The root of our force model is a simple piecewise linear function of the penetration depth of the undeformed brush point. If d_p is the penetration depth , and l_p is the length of the brush head projected onto the canvas normal, $\mathbf{n}$, then the force is modeled as:

$$\mathbf{f}_b(d_p) = \begin{cases} 0 & \text{if } d_p \leq 0 \\ \mathbf{n}(k_1/l_p)d_p & \text{if } 0 < d_p \leq l_p \\ \mathbf{n}(k_1 + (k_2/l_p)(d_p - l_p)) & \text{if } l_p < d_p \end{cases} \tag{3}$$

where k_1 is a small positive constant that models the light spring of bristles and k_2 is a larger positive constant that simulates collision of the actual brush handle with the canvas. The spring constants are normalized by l_p so that the same absolute force is delivered when the handle first hits the canvas, regardless of the brush length or orientation. The value of k_1 can be changed to simulate brushes of varying stiffness.

4.3 Compressive Effects

When a real brush contacts the canvas at close to a right angle, the stiff bristles initially act as strong compressive springs, transmitting an abrupt force to the handle. As more pressure is applied, the bristles buckle and the compressive force reduces as bending forces take over. When the brush makes a contact at an oblique angle, compressive effects play a lesser role in the force felt.

Therefore, we extend the piecewise linear function, Eqn. 3, to a piecewise Hermite curve. This curve is defined by a series of control tuples which contain the penetration depth and corresponding force magnitude, and the linear stiffness of the spring model at that point. We currently use a four-segment piecewise curve, which was derived from the empirical observation of how a brush head behaves under compression.

The initial segment of the piecewise curve models the compressive force. We assign the initial control tuple a fairly strong linear spring constant to simulate the initial strong compressive force. We modulate this compressive force according to the angle of contact, by multiplying the force value of the second control tuple by an angle-dependent coefficient between one and zero. Given θ, the angle between the canvas normal and negated bristle direction vector, the factor we use is

$$\gamma = \begin{cases} \cos^2(2\theta) & \text{if } -\frac{\pi}{4} < \theta < \frac{\pi}{4} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

This results in a compressive force that is strongest when a brush contacts the canvas at a right angle and that tapers off to zero as the brush approaches a 45 degree angle to the canvas.

4.4 Frictional Forces

The final component of the force delivered to the user is a small amount of tangential resistance. Though small in magnitude, frictional forces have a large effect on the user's perceived ability to control the brush by damping small oscillations in the user's hand. We model friction $\mathbf{f}_t$ simply, as a force opposite the current brush velocity, $\mathbf{v}_b$, which is added to the other feedback forces:

$$\mathbf{f}_t = k_t\,(\mathbf{v}_b - \mathbf{n}(\mathbf{n} \cdot \mathbf{v}_b))$$

where k_t is the coefficient of friction.

5 Paint Model

Complementing our expressive brushes and force feedback, we present a paint model capable of capturing complex effects interactively. Our paint model incorporates variable wetness & opacity, conservation of volume, and a hardware-accelerated bi-directional paint transfer algorithm. It supports the following operations and techniques expected from acrylic or oil painting, while maintaining complete interactivity.

- **Blending** – Mixing of multiple pigments to obtain the desired color.
- **Bi-directional transfer** – Transferring paint both from the brush to canvas, and back from the canvas to the brush.
- **Complex brush loading** – Filling the various portions of the brush head with different pigments.
- **Variable dryness** – Controlling the blending of new paint onto previous layers by allowing paint to partially dry.
- **Glazing** – Painting with translucent layers (veils) of colors over other opaque colors (i.e. *underpainting*).
- **Impasto** – Painting with thick volumes of paint without addition of any medium.

Users can also generate similar results using other advanced painting programs. However, with our paint model, they need only manipulate the virtual brushes similar to real ones, in order to automatically generate the intended paint effects.

5.1 Bi-directional Paint Transfer

Paint information is stored on both the canvas and brush in multiple textures (described in Section 5.2). The brush subdivision surface is tessellated to a polygonal surface. When this surface intersects the canvas geometry, the brush is considered to be in contact with the canvas. The bi-directional transfer must correctly modify the paint textures to simulate paint volume being interchanged between the two surfaces. Figure 6 displays a brush stroke possible only with bi-directional paint transfer.

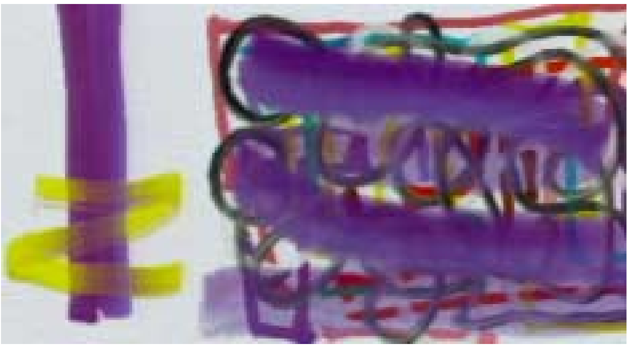


Figure 6: *Bi-directional paint transfer is demonstrated by dragging a yellow paint stroke through wet purple paint (LEFT). A purple glaze of paint has been thinly applied over dry paint (RIGHT).*

The paint transfer problem is first reduced to two dimensions to simplify computation while introducing only slight inaccuracies. In the general case, a projection plane would be chosen that maximizes the area projected by the intersecting curve between the brush and canvas surfaces. Currently we have implemented only a two dimensional canvas, and therefore use the canvas plane for the orthographic projection of the brush. This is achieved with polygon rasterization hardware, for speed and ease of implementation. The projected textures of the brush are used as the brush footprint.

The textures must be updated to simulate paint transfer and mixing. This 2D blending of the footprint with the canvas is discussed in Section 5.3. The simulation of the brush produces discrete instances of the brush surface; to produce a continuous stroke the blending operation is performed over a line connecting the current footprint to the previous one. The centroids of the footprint polygons are used as endpoints. This method provides smooth strokes while the footprint is not changing dramatically.

After 2D blending is complete, the updated textures are reapplied to the surfaces. This is achieved by rendering a variation of the brush subdivision surface mesh. The surface vertices that were projected to the footprint are used as texture coordinates into the now updated footprint textures. The original surface texture coordinates are used as vertex locations to render back into the surface's texture maps.

5.2 Paint Representation

The 3D brush and transfer methods presented here can be combined with many media types such as paint, ink, or watercolor. The *DAB* system currently includes a model that approximates the acrylic and oil families of paint.

Each paint surface contains two color layers. These are referred to as the 'surface' and 'deep' layers. Conceptually, the surface is covered by only a thin surface layer, and more thoroughly by the underlying deep layer. The surface layer is the boundary at which paint transfer between objects occurs. Surface layers are completely *wet*. The canvas's deep layer represents the paint that is completely *dry*. The brush's deep layer represents the reservoir of paint contained within the bristles. The paint transfer between surface layers occurs upon a collision between two objects (i.e. the brush and canvas). Transfer from the brush's reservoir layer to the surface is performed whenever the surface layer is no longer saturated (and paint remains in the reservoir layer). Drying paint from the canvas's surface layer to the dry layer occurs on a timed interval or as requested by the user.

The surface and deep layers are stored in color textures. A representation of the volume of paint in each layer is stored in an attribute texture. The surface layers and brush reservoir layer use fixed point representations, while the dry layer of the canvas is a specialized relative height field, and is described in Section 5.5.

5.3 Paint Mixing

The amount of volume transferred between surface layers is dependent on the volume of paint within each layer. The volume leaving, V_l, is computed from the initial volume, V_i, and transfer rate, R, over the elapsed time, T, by the equation, $V_l = V_i \cdot T \cdot R$. The resulting paint color, C_{new}, is computed by the weighted portions of remaining volume and color, $V_r = V_i - V_l$ and C_i, and incoming volume and color from the other surface, V_l' and C_i':

$$C_{new} = V_r \cdot C_i + V_l' \cdot C_i'$$

This essentially additive compositing formula is easy to work with, and gives predictable results, but does not model the way in which colloidal suspensions of pigment actually mix. The Kubelka-Munk model is the best known model available for accurately compositing pigments, but comes with significantly higher computational cost. See for example [CAS$^+$97].

5.4 Optical Composition

To generate realistic paint effects, the wet and dry layers of the canvas are composited together with an embossing of the paint volume. This allows for glazing effects, as illustrated in Fig. 6. The volume of the wet layer, V_w, is multiplied by the optical thickness, O_t, of the paint, and then used for alpha blending the wet and dry layer colors, C_w and C_d.

$$C_{displayed} = \alpha \cdot C_w + (1-\alpha) \cdot C_d; \quad \alpha = \min(V_w \cdot O_t, 1)$$

5.5 Drying the Canvas

Our paint model also supports variable wetness as shown in Fig. 7. Variable wetness is accomplished by gradually moving paint from the completely wet surface layer of the canvas to the completely dry deep layer.

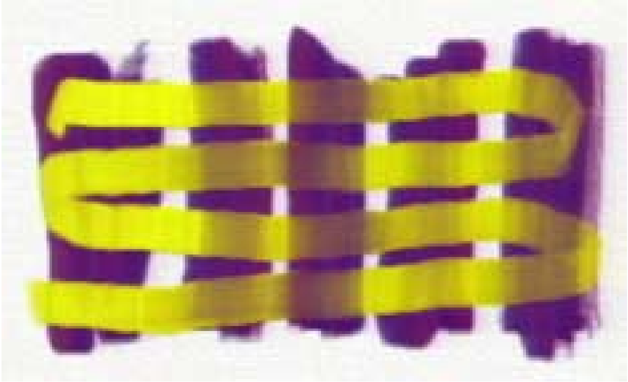


Figure 7: *Variable wetness is displayed as yellow paint has been painted over the purple color stripes of 100%, 50%, 0%, 75%, 25% dryness (left to right).*

The composited color of the paint must not change during drying. The optical blending function is used with this constraint to solve for the new dry layer color, C_d', when some volume, $\delta\alpha$, is removed from the wet layer.

$$C_d' = \frac{\alpha \cdot C_w + (1-\alpha) \cdot C_d - \alpha' C_w}{(1-\alpha')}; \quad \alpha' = \alpha - \delta\alpha.$$

The dry layer of the canvas uses a relative height field to allow for unlimited volume of paint to be added, with a constraint only on the relative change in height between texels. An embossing of the height field is also computed. We use additive blending to combine this embossing and the color buffer to create the final rendered image of the paint.

6 Implementation Results

We have developed a prototype painting system, *DAB*, which combines 3D virtual brushes with a haptic interface and our paint model, as described in this paper. As mentioned in Section 2, the graphical user interface consists of three main elements: the canvas, the palette and the brush rack.

In the absence of a 3D stereo display, we have introduced shadows in our graphical display to enable the users to infer the relative position of the paint brush to the virtual canvas.

6.1 Discussion

A painter's palette not only "lists" available colors, but also allows a painter to mix and create a nearly unlimited number of new ones, and it presents both possibilities simultaneously through a simple, unified interface. Furthermore, creating complex color "gradients" on a painter's palette is just as easy as creating a single color: simply mix the constituent colors less thoroughly. In short, a real palette is a natural interface for color choosing, but one which has not been taken advantage of in previous computer painting systems.

To take best advantage of a painter's palette interface requires a 3D virtual brush like the one presented in this paper. With a 3D virtual brush, loading the complex blends created on the palette is simple, as is creating strokes that use those blends. Combined with an appropriate 3D input device, *DAB* offers a powerful yet simple interface for painting.

We chose to use a Desktop PHANToM for input and haptic feedback because it provides true 6-DOF input with excellent precision and low noise, while offering fully programmable 3DOF force output. Other input devices such as tablets offer at most 5-DOF input (lacking a degree of freedom for twist), and have rather large noise in the tilt measurements.

On a pragmatic level, the force feedback is useful in that it enables the user to detect and maintain contact with the canvas better than if just shadow cues are provided. A tablet gives a physical surface that serves the same purpose, but it always gives the sensation of a rigid pen on a hard surface, rather than a soft, flexible brush on a canvas. Nearly all the users who have used both a tablet system and a haptic device preferred the soft feel of force feedback for brush simulation. Finally, with fully programmable forces, we are also able to change the feel of the brush at will, making it softer or harder for instance.

We are currently planning a detailed user study to thoroughly evaluate and assess the value of force feedback in creating the “right feel” for the artists. Using a programmable force feedback device with a true 3D workspace further enables the possibility to expand our system in a number of exciting directions covered in the next section.

6.2 User Feedback

More than fifteen users have painted with our system. This group of users includes amateurs and art students, both males and females, with ages ranging mostly from early 20’s to late 30’s. Some have prior experience with other computer painting programs and various user interfaces. All the users were able to pick up the haptic stylus and start painting immediately, with little training or detailed instruction. A small selection of their artistic creations is shown in Figs. 8 to 14. Additional images of art works created by our users, and detailed screen shots, are available as supplemental materials on the CD-ROM and on the project website.

Among users who have worked with other painting programs and interfaces, most found our painting system to be more intuitive. For artists with prior painting experience, our painting system was substantially easier to adapt to than other painting programs, while offering similar capabilities, such as undoing brush strokes, drying paint, etc. We attribute this to the fact that *DAB* offers a painting environment that takes advantages of skill transfer. *DAB* also seems to have an appeal for people with an artistic bent, but who would not normally consider painting, as well as for painters who would not normally use a computer painting system.

7 Future Work

Users of all types found *DAB* compelling to work with, however there are many aspects of the system which can be extended.

For improved accuracy in the brush deformation simulation, we continue to investigate the use of other efficient integration and simulation methods such as [BW98]. We are also interested in simulating a greater range of haptic phenomena from the feel of paint textures, to the variation in sensation when using different types of brush fibers, from painting on different backings, or with different mediums. Another natural step would be to go from painting 2D surfaces to painting 3D geometric models.

Our current paint model can be extended to depict more advanced characteristics of oil painting such as: gouging effects from bristle marks, anisotropic BRDFs, multiple wet layers of paint, and lighting-independent rendering of paint bumps. We are also interested in a real-time implementation of the Kubelka-Munk model for compositing. Expanding the set of virtual tools to include more types of brushes and and other artistic tools is also of interest.

Our initial observations taken from a relatively small group of amateur artists, art students, and novices indicate that our approach is effective. We plan to conduct a more thorough and extensive formal user study over a larger group of users to confirm this observation, as well as to evaluate the effectiveness of various contributing factors in our interface design.

8 Acknowledgements

We are grateful to all the people who evaluated our system. Several of their paintings drawn using *DAB* are included in this paper. We would also like to thank the anonymous reviewers for their useful comments. This research was supported in part by ARO DAAG55-98-1-0322, DOE ASCII Grant, NSF NSG-9876914, NSF DMI-9900157, NSF IIS-982167, ONR Young Investigator Award, and Intel.

References

[ABL95] M. Agrawala, A. Beers, and M. Levoy. 3D painting on scanned surfaces. In the Proc. of *1995 Symposium on Interactive 3D Graphics*, pages 145–150. ACM SIGGRAPH, April 1995.

[BW98] D. Baraff and A. Witkin. Large steps in cloth simulation. *Proc. of ACM SIGGRAPH*, pages 43–54, 1998.

[CAS+97] C. Curtis, S. Anderson, J. Seims, K. Fleischer, and D. Salesin. Computer-generated watercolor. *Proc. of SIGGRAPH’97*, pages 421–430, 1997.

[COR00] COREL. Painter. *http://newgraphics.corel.com/products/painter6.html*, 2000.

[CPE92] T. Cockshott, J. Patterson, and D. England. Modelling the texture of paint. *Computer Graphics Forum (Eurographics’92 Proc.)*, 11(3):C217–C226, 1992.

[DSB99] M. Desbrun, P. Schroder, and A. Barr. Interactive animation of structured deformable objects. *Proc. of Graphics Interface’99*, 1999.

[GEL00] A. Gregory, S. Ehmann, and M. C. Lin. *inTouch*: Interactive multiresolution modeling and 3D painting with a haptic interface. *Proc. of IEEE VR Conference*, 2000.

[HDD+94] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *Proceedings of ACM SIGGRAPH*, pages 295–302, 1994.

[hem00] RIGHT hemisphere. Deep paint. *http://www.us.deeppaint.com/*, 2000.

[Her98] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. *Proc. of ACM SIGGRAPH’98*, pages 453–460, 1998.

[HH90] P. Hanrahan and P. Haeberli. Direct WYSIWYG painting and texturing on 3D shapes. In *Computer Graphics (SIGGRAPH ’90 Proceedings)*, volume 24, pages 215–223, August 1990.

[JTK+99] D. Johnson, T. V. Thompson II, M. Kaplan, D. Nelson, and E. Cohen. Painting textures with a haptic interface. *Proceedings of IEEE Virtual Reality Conference*, 1999.

[Lit97] P. Litwinowicz. Processing images and video for an impressionist effect. *Proc. of SIGGRAPH’97*, pages 407–414, 1997.

[May70] R. Mayer. *The Artist’s Handbook of Materials and Techniques*. The Viking Press, 1970.

[Mei96] B. Meier. Painterly rendering for animation. In *SIGGRAPH 96 Conference Proceedings*, pages 477–484. August 1996.

[Pix00] Pixologic. Z-brush. *http://pixologic.com*, 2000.

[Smi78] Alvy Ray Smith. Paint. TM 7, NYIT Computer Graphics Lab, July 1978.

[SN99] S. Saito and M. Nakajima. 3D physically based brush model for painting. *SIGGRAPH99 Conference Abstracts and Applications*, page 226, 1999.

[Str86] S. Strassmann. Hairy brushes. *Computer Graphics (SIGGRAPH’86 Proc.)*, 20:225–232, 1986.

[WB97] A. Witkin and D. Baraff. *Physically Based Modeling: Principles and Practice*. ACM Press, 1997.

[WI00] H. Wong and H. Ip. Virtual brush: A model-based synthesis of chinese calligraphy. *Computers & Graphics*, 24, 2000.

Figure 8: *A painting by Eriko Baxter (LEFT); by Rebecca Holmberg (RIGHT)*

Figure 9: *A painting by Rebecca Holmberg*

Figure 12: *A painting by Andrei State*

Figure 10: *A painting by Rebecca Holmberg*

Figure 13: *A painting by Lauren Adams*

Figure 11: *A painting by Rebecca Holmberg*

Figure 14: *A painting by Sarah Hoff*