Airbrush Simulation for Artwork and Computer Modeling

Jonathan Konieczny*
Gary Meyer†
University of Minnesota
Department of Computer Science and Engineering
Digital Technology Center

Abstract

A computerized airbrush system with a full three dimensional airbrush interface is presented. The position and orientation of an electronic airbrush tool is tracked in space, and, when the trigger is pulled, paint is sprayed onto two and three dimensional objects displayed on a computer monitor. The experimentally derived paint spray model used for the airbrush takes into account factors such as air to paint ratio, viscosity, and distance of the airbrush from the work. Paint mixing between colors applied to the surface is modeled using Kubelka-Munk theory. Computerized stencils, including semi-permeable stencils, can be manually positioned by the artist or projected onto the object's surface. Two and three dimensional examples of traditional airbrush artwork, produced using the system, are presented. The system can also be used as a modeling tool to decorate three dimensional objects.

CR Categories: H.5.1 [Information Interface and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

Keywords: Airbrush, Painting, Virtual Reality, Augmented Reality, Art

1 Introduction

Allowing artists to ply their craft on a computer has always been a major goal within the field of computer graphics. However, while there are programs that permit artists to create virtual artwork and decorate virtual models, far fewer software packages allow artists to do so using skills they learned with real-world tools such as a paint brush or sculpture chisel. For example, most drawing and painting applications available today provide a brush that generates an airbrush spray pattern, but few of these programs work with three dimensional objects and none of them provide a true airbrush interface. In this paper we present an airbrush tool with correct spray paint dynamics and a complete three dimensional interface, and we demonstrate how it can be used to produce both traditional airbrush artwork and texture detail for three dimensional models.

The airbrush has several advantages over a traditional paint brush. First, it leaves no brush stroke marks, permitting the creation of a much smoother and more realistic image. Second, the rate of application can be easily varied, allowing for very smooth paint gradients. Finally, the width and density of a single airbrush stroke is very easily varied mid-stroke, allowing the production of artistic effects that are difficult to achieve with a bristle brush (see the fade away stroke in Figure 2 for an example of such effects). These unique properties make the airbrush a commonly used tool in the artistic and design worlds. While most photo-retouching is now done using programs such as PhotoShop, the airbrush is still employed by commercial artists to produce architectural renderings [Dombek and Porter 2003], medical illustrations [Misstear 1984], advertising signs, and automotive graphics. Fine artists use the airbrush to apply ceramic glazes [Misstear 1984], to decorate textiles [Maurello 1955], and to create wall murals. Model makers add detail to a wide variety of objects including train dioramas [Caiati 1985], costume masks, stuffed animals [Mitchel 2008], wax figures, and fish lures.

While bristle brush painting has been fairly well researched in computer graphics, airbrush has not. Computerized airbrushing has several differences with simulation of bristle brushes. First, no haptics are needed to perform a proper simulation. While the feel of a bristle brush against the surface of the painted object is an important feedback device in bristle painting, an airbrush artist works almost entirely via visual feedback. This significantly lessens the equipment requirement for full airbrush simulation versus brush simulation. Second, most of the potential advantages a real airbrush has over a bristle brush still apply: brush stroke size and rate of application are easily varied, and a smooth final appearance is achieved - brush bristles and resulting brush patterns on the canvas need not be simulated. Airbrushing also stands as a prime example of a true three dimensional real world interface. Studying its success as an interface could provide interesting insights into virtual interface design.

In this paper, an airbrush simulation tool is introduced that closely replicates the process of real airbrushing. This is done not only to give a virtual version of the real world tool, as [Baxter et al. 2004] does for paint brushing, but also to give a new way to decal virtual models and supplement tools such as Photoshop for texture map creation. In order to create as realistic a simulation as possible, many aspects of airbrushing are explored. First, an existing spray particle simulation is optimized for airbrush simulation. Second, a Kubelka-Munk paint simulation is provided for accurate rendering of both wet and dry paint layers as they are sprayed onto the canvas in real time. Third, a realistic airbrush interface is created which closely mimics what a real airbrush looks and feels like. Fourth, interface/artistic tools which airbrush artists use are added into the system, such as true three dimensional stencils and easy paint color selection. Finally, display of the canvas or model, as it is painted, is provided on a computer monitor. The result is a WYSIWYG system which allows existing airbrush artists to create virtual artwork as well as non-photorealistic model texture materials. This system has been evaluated by professional airbrush artists, and some of the resulting artwork created from those trials is presented.

2 Relevant Work

Traditional brush painting and paint mixing have been heavily researched in computer graphics, and both are relevant to the work

*e-mail: jkonie@cs.umn.edu
†e-mail: meyer@cs.umn.edu

presented in this paper. Below is a summary of some of this research.

2.1 Brush and Spray Particle Simulation

Several researchers have simulated the physics of a brush being applied to the surface of a canvas and have used those results to produce painting applications. While the physics of an airbrush are quite different from that of a bristle brush, the end goal is similar: to create a program that realistically simulates the use of that artistic tool by properly modeling the physics of the device.

[Chu and Tai 2002] and [Saito and Nakajima 1999] both gave a deformable brush model for simulating oriental style ink painting. [Baxter et al. 2001 and 2004] gives both a deformable brush model as well as a haptic interface device for simulating oil painting.

[Rudolf et al. 2003] presents both a simulation of wax crayons as well as the physics of the canvas surface to provide a simulation of crayon drawing.

[Konieczny et al. 2008] provides a simulation of spray paint for automotive paint application and training. Our system uses a similar particle based solution for modeling the physics of an airbrush, but enhances it for the finer detail and control required for airbrushing instead of industrial spray painting.

2.2 Paint Simulation

In addition to simulating the physics of the particular paint brush, the paint being used must also be modeled in order to provide a convincing result. Some of these models are intended to be used along with the brush physics models, while others just take existing images and alter them to appear as if they have been painted using some desired brush pattern.

[Curtis et al. 1997] used shallow water diffusion equations to solve watercolor paint mixing. Curtis also presented an implementation for Kubelka-Munk (K-M) paint mixing. The result was a convincing simulation of watercolors, but the results could not be displayed to the user interactively.

The work most similar to our own with respect to its simulation of paint mixing and layering is [Baxter et al. 2004]. This work uses a GPU implementation of K-M mixing. As one paint is added onto another, they are mixed and the result of that mix is presented back to the user interactively. Once a particular layer has dried, that

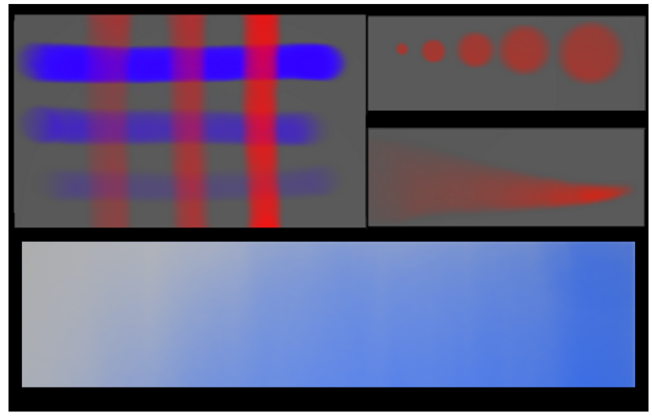

Figure 2: *A range of various brush strokes and beginner exercises. Top-Left: Red lines of differing paint thickness are drawn over blue lines, created by varying how far the trigger is pulled back. This also shows the Kubelka-Munk paint mixing. Top-Right: Dots of differing size, but similar thickness, obtained by pulling back from the canvas, but varying the trigger. Top-Right: A fade away stroke, obtained by pulling away from the canvas while moving the brush. Bottom: A shading exercise creating a smooth transition from dark to light paint thickness.*

layer along with all previous layers is condensed into a new single layer in order to speed the simulation. The solution presented in this paper also provides a real time GPU based implementation of K-M paint mixing. [Rudolf et al. 2003] also uses K-M mixing to simulate wax crayons, but does not do so in real time.

Finally, [Hertzmann 1998 and 2002] uses an algorithmic approach to simulate the appearance of various brush strokes. However, this approach only alters the appearance of existing images.

2.3 Professional Paint Suites

There are a number of professional painting programs that have an airbrush tool, such as Adobe Photoshop. However, these tools appear to simply mimic a random spray pattern on a two dimensional surface, they do not actually simulate spray particles in three dimensions, and they do not give the user a true airbrush interface. Programs such as BodyPaint 3D or Z-Brush do allow a user to paint directly on a three dimensional surface, but they employ a standard two dimensional input device such as a mouse or a tablet PC and they do not provide three dimensional stenciling capabilities. It is our hope that interfaces and physical simulations such as the one presented in our paper can be incorporated into programs such as those above to allow artists to interact with virtual artwork and content as naturally and easily as possible.

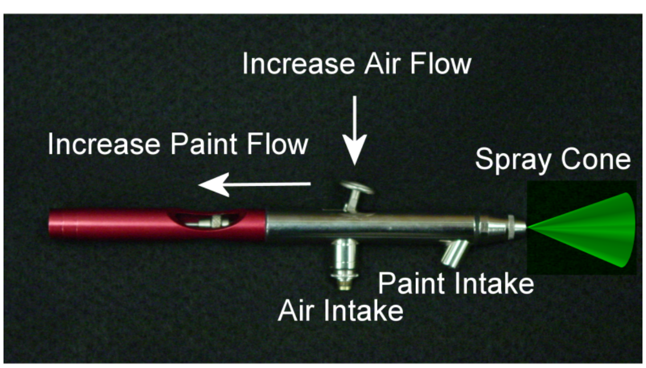

Figure 1: *A typical airbrush. Pulling back on the trigger increases paint flow. Pushing down increases air flow. Both paint spray area and thickness is altered when moving the brush closer or further from the surface*

3 Airbrush Mechanics and Interface

Mechanically, an airbrush is similar to a standard spray paint gun, except that it generates much finer spray particles and provides more precise control over the flow rate. Figure 1 shows a typical airbrush. Pulling back on the airbrush trigger increases flow rate, while pushing down on it increases air flow rate. Typically, air flow and paint flow are used to control the rate at which paint builds up on the surface. The distance of the gun to the target surface is altered to change the radius of the resulting dot of paint that lands on the canvas: closer makes a smaller dot.

Through the manipulation of air flow, paint flow, and distance an airbrush artist is able to create a wide range of brush stroke widths and thicknesses (see Figure 2). There are several advantages over typical brush painting: first, an airbrush can rapidly and widely vary the width of the stroke size, even in a single brush stroke. This is very difficult to achieve with a normal paint brush. Second, no actual brush marks are left: the paint lands on the surface almost perfectly smoothly. This allows for a much smoother and more realistic result from using an airbrush over normal brushes. Finally, the rate of paint flow can be easily varied, allowing both small and large pieces of artwork to be created with a single tool.

The biggest disadvantage of an airbrush is that it is more difficult to create a very thin line with the airbrush: most airbrush artists keep a fine tip paint brush around to paint detail work such as eyelashes on a person's face. Also, it takes more effort to switch paint colors with an airbrush: many airbrush artists use multiple airbrushes so they can rapidly swap between desired colors. It is worth noting, however, that both these disadvantages can be overcome with a computer simulation: color selection is easily made in the program simply by pointing the airbrush at a color palette, and objects can be arbitrarily scaled for varying levels of detail.

As mentioned in Section 1, one of the major differences between simulation of an airbrush versus simulating a normal paintbrush is that a real airbrush provides almost no haptic feedback: therefore a haptic device is not required to simulate the system. However, some kind of six degree of freedom tracker is required. Our current system uses a PCI Bird magnetic tracker, although a cheaper six degree of freedom tracker such as a Wii remote could work as well. In addition to tracking the airbrush, the system described in this paper also tracks both the object being painted as well as the stencil being used so that the user may easily place both into any desired position.

In addition to tracking, the airbrush itself must be simulated with a realistic feeling trigger. Figure 3 displays our electronic airbrush. A real airbrush was taken and modified to have an analogue thumbstick, and a magnetic tracker was added to the back end. One nicety of simulating an airbrush is that the cording required for these two devices is not unexpected by an airbrush artist, as all airbrushes have a cord hooked up to an air compressor. Therefore, the weight and feel of our electronic airbrush is very similar to that of a real airbrush.

The surface to be painted is represented as a model displayed on a computer monitor. This can either be just a flat canvas placed so that it appears the user is painting directly onto the monitor itself, or a full 3D model displayed on the screen, allowing the user to work as if painting the model through a window (see Figure 4).

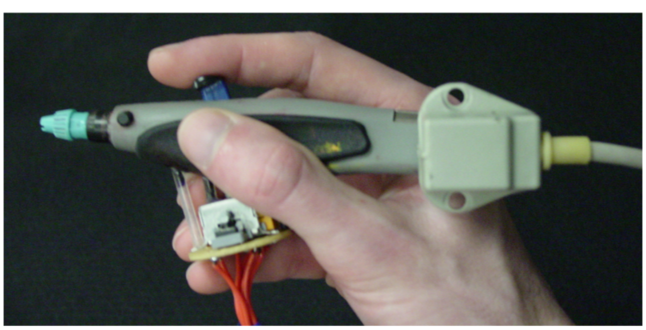

Figure 3: *The electronic airbrush. A real airbrush was hollowed out and the trigger replaced with an analogue trigger. A magnetic tracker sensor is also attached to the right end. This allows a user to hold a real airbrush, and obtain the data from the airbrushes movements on the computer. Note that real airbrushes also have a hose attached to them for airflow: therefore the electronic airbrush actually has very similar bulk to a real airbrush.*

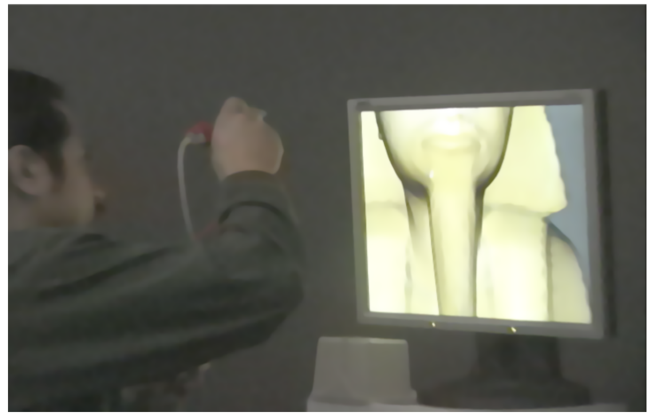

Figure 4: *A picture of the system in use. An airbrush artist (Gustavo Lira) is shown using the airbrush to paint a 3D model. The tracking emitter is positioned below the screen.*

4 Airbrush Simulation

Our airbrush simulation is built on the spray paint particle simulation given in [Konieczny et al. 2008], but includes multiple modifications to allow for the finer control and stencils required by the airbrush. The implementation was optimized and, as described in this section, the following features were added: dynamic splat size determination, automatic frame rate adjustment & optimization, and paint particle & tracking interpolation. The simulation was also integrated with a paint mixing system detailed in Section 5. These new features make a dramatic improvement in the usability of the system.

4.1 Spray Paint Particle Simulation

A ray casting algorithm is used to simulate spray particles as they travel from the spray gun. Each particle's origin is chosen as the center of the spray gun nozzle, and a direction is selected within the gun's spray cone. In all of the images given in this paper, the spray direction is completely random within this cone. However, if desired, a different spray distribution can be chosen according to a specific spray gun's parameters. In order to provide a fast simulation, particles may also be splatted onto the target surface. This allows a wider area to be painted smoothly without having to cast additional rays.

As a pre-computation, the texture density of each triangle is stored in the system, giving a physical area to each texture pixel on the model mesh. When a particle strikes the target mesh, a UV texture coordinate is found using barycentric coordinates. Then, a certain density of paint is stored in the texture pixel that was struck, based on a number of variables: distance of the gun to the surface, viscosity of the paint, air and paint pressure setting of the gun, paint flow rate, and UV texture density. Using the data presented in [Kwok 1991] (see Table 1 for an example), interpolation formulas were developed that allow the simulation to take the above factors into account. Further details can be found in [Konieczny et al. 2008]. By using an equation derived from experimental data to simulate most of these physical effects, the simulation is kept accurate, but with far less computation than a full physical simulation would require.

4.2 Airbrush Modifications

In order to create an airbrush simulation several important modifications were made to the spray paint particle algorithm described in [Konieczny et al. 2008]. First, the frame rate has been increased from 30 frames per second to 60 frames per second. This is because in user testing, it was found that airbrush artists move the spray gun much more rapidly than with normal spray painters, requiring a higher frame rate to keep up with the movements (see Section 7 for an explanation on the user sensitivity to frame rate). The number of particles per frame are dynamically altered to enforce this 60 frames per second. This is done by polling the current frame rate, and decreasing the particles per frame by 0.5% per frame until at least 60 frames per second is achieved. The splat size is then increased to compensate for the decrease in particles, if necessary.

Second, the size of the texture map can have significant effects on the simulation. With a higher resolution texture map, more pixels are altered per frame, forcing more pixels to be altered and uploaded to the graphics card per frame. In order to properly simulate stencils (see Section 6) a high resolution texture map must be used: the current simulation uses a texture map of either 1024x1024 or 2048x2048 to store the paint density. In order to maintain 60fps with textures of this size, an array of altered pixels is kept for each frame, and only those pixels are uploaded to the graphics card per frame. It is also worth noting that adjusting the texture map size used is equivalent to changing the splat size: doubling the texture size is the same as halving the splat size. Therefore, for the most rapid possible computation, it is more desirable to simply decrease the texture size rather than increase splat size, except in cases where the splat size may be changing within a single painting session (see Section 6.1 for a discussion on splat size).

Third, an algorithm to handle tracker latency and frame rate inconsistency has been added. Normally, the tracker is polled at its current position each frame, and a spray cone is generated based on that reading. However, if the user moves the spray gun rapidly, this can result in a splotchy appearance. In order to create a proper, smooth application of paint across the surface, the new system now interpolates the position/orientation of the paint gun between frames and paints everything in between using the following algorithm:

1. Obtain the position and orientation of the tracker of both the last frame and the current frame.

2. Linearly interpolate between the two tracker readings, stepping from the previous frame to the current one.

3. At each step, generate a spray cone and randomly fire paint particles within that cone, as per the usual algorithm.

This generates a much smoother application of the paint on the object than the algorithm given by [Konieczny et al. 2008]. See Figure 5 for an example of using the old per-frame spray algorithm versus the new interpolation method.

Variable	Value	Paint Deposition (gm)	Overspray (%)
A/P Ratio	0.92	4.14	22.32
A/P Ratio	1.49	3.54	31.69
A/P Ratio	2.18	3.19	39.69
Viscosity(cstk)	57	3.54	31.69
Viscosity(cstk)	106	4.32	25.44
Distance(inches)	7.00	4.59	21.93
Distance(inches)	10.00	3.54	31.69
Distance(inches)	14.00	2.75	45.99

Table 1: *Taken from [Kwok 1991], this table shows the effects of A/P ratio, viscosity and distance on spray paint deposition when the paint flow rate was 275cc/min.*

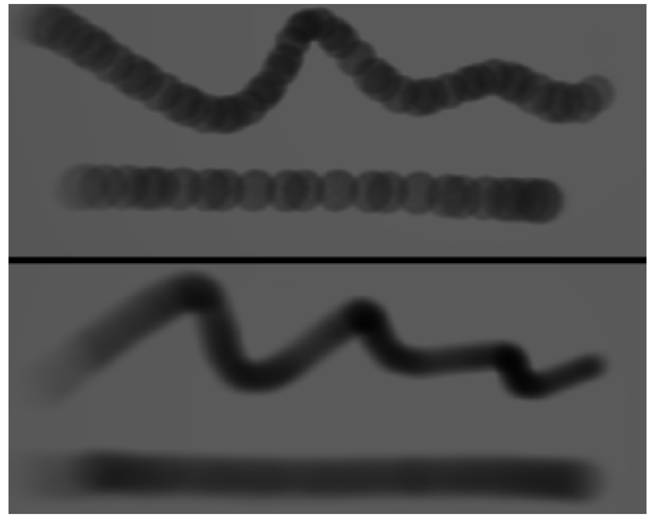

Figure 5: *An example of the previous spray simulation versus the new one. The simulation has been modified to allow rapid hand movements, and automatically interpolates those movements between frames, allowing a much more smooth final paint appearance.*

4.3 Performance

As stated above, the current system is required to run at a 60fps minimum in order to provide a proper interactive experience to an airbrush painter. On a Pentium IV 3.2 Ghz processor with a GeForce quadro 5800 graphics card, the system was able to run with a 2048x2048 texture. The number of particles per frame varied from approximately 20,000+ particles per frame with a 10 polygon model to 4000 particles per frame on a 50,000 polygon model. In each case splat size was adjusted to yield a smooth look to the resulting spray deposition For the 10 polygon model, the optimal splat size was one (a single pixel per particle). For the 50,000 polygon model, nine texels were painted per particle.

5 Paint Simulation

Once a deposition of paint has been built up on the texture map, that paint must be simulated. For airbrushing, the paint most commonly dries before the next layer of paint is applied. However, in some cases artists purposely smear the paint to force the paint pigments to mix rather than dry one over the other. Therefore, in order to properly simulate airbrush paints, both realistic mixing of paint pigments as well as layering of those pigments must be performed.

In addition to simulation of wet and dry paint, paint transparency is extremely important. Many airbrushing effects are achieved by layering multiple layers of transparent paint, sometimes mixing in the occasional opaque paint in areas. Therefore, in order to create a fully functional simulation of airbrushing, a very general paint mixing and appearance simulation must be used.

One solution to paint mixing that is currently regarded as quite accurate is Kubelka-Munk (K-M). K-M provides a solution to an arbitrary number of layers of paint as they are mixed together by finding an absorption (K) and scattering (S) coefficient for each paint being mixed. [Haase and Meyer 1992] note that the K/S ratio and spectral reflectance of a paint can be analytically related in the case of complete hiding: when the material underneath the mix of paints cannot be seen. Unfortunately, this does not hold in the case of

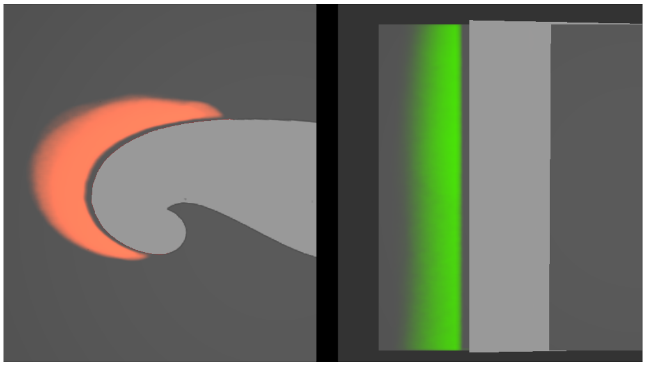

Figure 6: *Examples of free-hand use of the stencil. Left: A french-curve is used to create a curving line across the surface. Right: A straight stencil is held at an upward angle from the canvas to create a hard edge on one side, but a blurred line on the other side. This is useful for creating many artistic effects.*

semi-transparent paints. Since airbrush paints are almost always partially transparent, a more complicated simulation must be used.

Our solution is similar to that given by [Baxter et al. 2004], which gives a real time solution to K-M paint mixing. In order to perform K-M, they equate spectral reflectance and K/S by setting S to be an arbitrary number. Once these have been found a final reflectance for a mix of paints can be calculated (from [Baxter et al. 2004]):

$$b = \sqrt{(K/S)(K/S+2)} \tag{1}$$

$$R = \frac{1}{1 + K/S + b\coth(bSd)} \tag{2}$$

$$T = bR/sinh(bSd). \tag{3}$$

$$R_{tot} = R + \frac{T^2 R_{prev}}{1 - RR_{prev}} \tag{4}$$

where d is the thickness of a layer of paint, R and T are the reflectance and transmittance of a layer, and R_{tot} is the final reflectance for a layer on top of previous layers.

If a user has the specific K and S values, our algorithm allows the user to input them for a given paint. Otherwise, S will default to one and a solution found according to the equations above. As in [Baxter et al. 2004], the user may dry a layer, at which point the system calculates the final reflectance of that layer combined with all previous layers, and places that result into a single layer in order to save processing time. In this manner, many paints can be mixed together and shown, all in real time. Our system filters the full spectral distribution and takes eight samples, as performed in [Baxter et al. 2004], although the implementation could be easily extended to include additional wavelengths in the calculation.

The current system performs this calculation per-pixel in the fragment shader. The total number of textures required to simulate K-M is one plus the number of wet layers. Currently, the system keeps four paint density maps to perform this simulation, or a single floating point RGBA texture. However, if more layers of paint are desired, more textures could be added with minimal computation cost, as the current implementation is CPU bound (from the ray cast simulation). Also, only the currently active layer of paint has altered pixels (the paint color that is being currently sprayed), and therefore the amount of information transferred to the graphics card per frame remains constant regardless of the number of layers of paint being represented.

6 Computerized Stenciling

The use of stencils is extremely important in actual airbrushing. Various cutouts are taken and put on top of the object to define a certain set shape that the user wishes to paint. Not only are solid stencils used, but also semi-permeable stencils such as cotton balls and feathers are employed to create effects such as clouds and fur. Therefore, the inclusion of stencils is critical for proper airbrush simulation.

Computer graphics lends itself well to both the use and creation of stencils. Our algorithm allows arbitrary 3D objects to be used as stencils, and those objects may be semi-permeable (for instance, a model of a feather could be used as a stencil). This is accomplished through the use of models with an alpha texture placed on them. When a particle of paint is cast, the program checks both the stencil object and canvas object for an intersection. If the canvas object is not struck, the particle is immediately discarded without checking the stencil. If it is struck, the stencil is also checked for an intersection. If it is not struck, the paint is applied to the canvas normally. If it is also struck, than a texture lookup is performed on the stencil alpha map. The resulting paint deposition on the canvas is reduced by the value of the alpha map, between 0 and 100%.

In this manner, any arbitrary object can be used to stencil out regions of the canvas object. As with the canvas object, the user is given controls to move the stencil into any desired position and lock it into place. In addition to using stencils free-hand, airbrush artists frequently affix the stencil to the surface in order to allow them to precisely mask out the stencil area. See Figure 6 for examples of the use of stencils.

In order to achieve this functionality, the current implementation allows the user to projectively texture any stencil onto the surface of the object. The canvas object is given an extra alpha texture map with the same UV coordinate system as is used for the paint density map. The user then moves the stencil in space to where they want to attach it to the object. The system projects the stencil onto the canvas object from the user's current viewpoint, copying each stencil

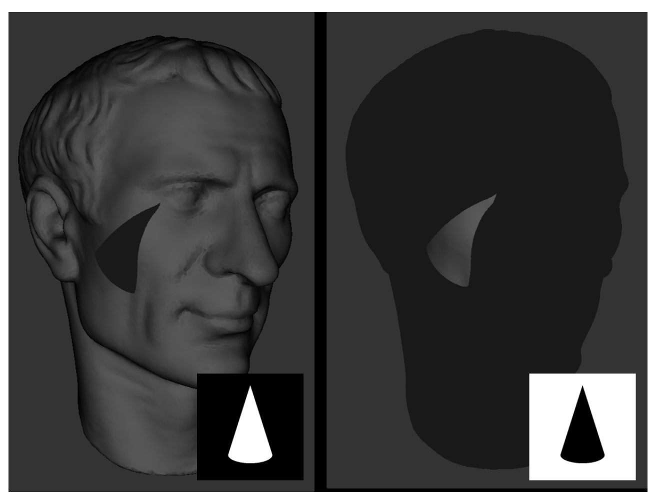

Figure 7: *Projective texturing is used to attach a stencil to the surface of the object when desired. Left: the stencil affixed to the object. Right: the inverse of the left stencil is affixed to the surface, with the rest of the surface automatically also stenciled out. The stencil used in each case is shown in the lower right corner. Such inverted stencils are frequently used so that artists can create differing images inside and outside of a particular region.*

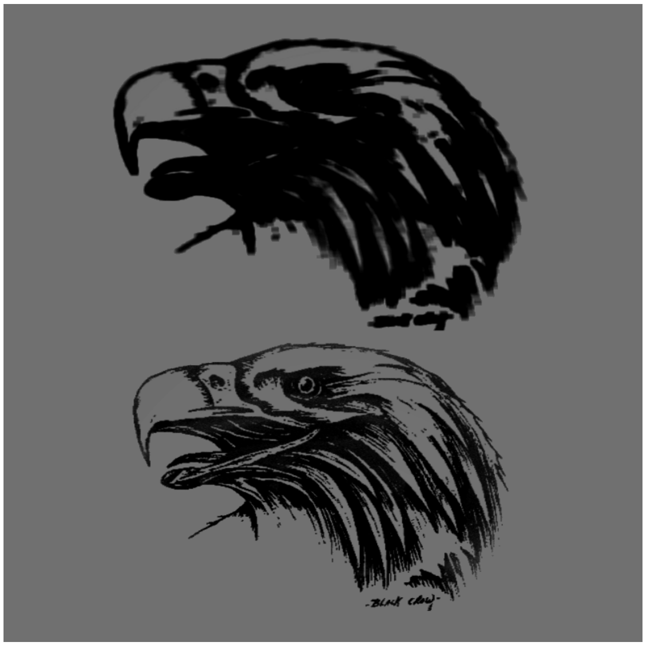

Figure 8: *Top: a stenciled out region is blurred out due to paint splats interpolating between texture pixels that should be blocked out by the stencil. Bottom: The same stenciling is done with the automatic splat size algorithm. The stenciled out region is much sharper.*

alpha value onto the object's alpha map. The region of the canvas outside the projected stencil can also be masked out so the user can prevent excess spray from accidently painting unwanted areas of the object. See Figure 7 for examples of projective stenciling.

The most commonly used stencil is a flat cut out of a desired shape, and is easy to create with a computer: any picture can just be converted into a gray-scale image and used as a stencil. More complicated stencils can be constructed using 3D objects with a custom alpha map laid over them. One interesting possibility is to use the airbrush program itself to create new stencils: the program can be used in conjunction with white paint to create any desired object with an opacity that the user sprays onto it.

To allow the artist to manipulate the stencil in the scene, a tracked, hand held prop is given to the them. The user can then manipulate the prop and see the corresponding movements in the virtual scene. This allows the artist to easily maneuver the stencil to the desired location and then freeze it in space, attach it to the object, or paint freehand. The current system provides no collision detection between the stencil and the canvas object, users appeared to have no difficulty adjusting to this.

6.1 Stencils, Paint Distance, and Splat Size

In order to provide both a real time solution as well as an accurate paint application, splat size must be properly set. As noted in Section 4.1 a larger splat size gives a more smooth look and saves computation time, but using smaller splats with more particles gives a more realistic result. While the necessary splat size is largely based on the computation capability of the computer being used, there are also two important run time variables that have a significant impact on the required splat size: the use of stencils and distance from the gun to the canvas object.

While using a stencil, it is desirable to have as small a splat size as possible. The reason for this is that larger splat sizes can cause fine detail in the stencil to be blurred out of the painted canvas. As the airbrush is pulled further from the canvas, the area which it paints increases rapidly. Therefore, a splat size that previously looked quite smooth when the brush was held close to the canvas may no longer look acceptable when the airbrush is further away.

The solution to these problems is to automatically detect when these situations occur and adjust the splat size accordingly. Therefore, the user currently selects a minimum splat size that looks acceptable to them at a close distance between the brush and the canvas. Then, our algorithm tests the distance each particle traveled between the gun tip and canvas, and increases the splat size as that distance increases. The result is that a smooth look is maintained no matter how close or far away the brush is from the surface. In addition, when a particle intersects the stencil, the splat size is automatically set to the minimum to insure detail is preserved while painting stencils. See Figure 8 for an example of how this fixes the blurry stencil problem mentioned above.

7 User Feedback and Examples

This section details some of the applications that the computer airbrush simulation is capable of performing. Some of them are typical examples that a real airbrush would be used for, such as artistic painting on canvas. Others use the airbrush to decorate 3D models, showing the potential of the system to aid 3D modelers and artists with decorating their work. Most of the illustrations and examples were created by real airbrush artists brought in to use the system and evaluate it.

7.1 User Trials

Several airbrush artists were brought in during the course of development, both to create artwork with the virtual airbrush and to make suggestions for system improvement. The hardware and software were first demonstrated for the artist, and they were then allowed to familiarize themselves with the system. Once an artist felt comfortable using the tools (typically in about 15 to 30 minutes), they were then allowed to take as long as they wanted to paint a piece of artwork. Generally, they were permitted to select their own subject, although for some of the examples they were asked to paint a specific model or piece of art (such as the model shown in Figure 13).

Most of the artwork required less than an hour to make, and none took over two hours. Figures 9, 12, and 13 each required around one hour to create, while Figures 10 and 11 took under a half hour each. Although no formal timing comparisons were made, all of the artists reported that painting similar artwork with a real airbrush would have required about the same amount of time, not counting time spent waiting for paint to dry and swapping colors (both of these maintenance tasks were significantly sped up by using the virtual system). Simple tasks such as adding the eye detail to the model in Figure 13 took only a couple of minutes.

Overall, feedback from artists was quite positive. Most reported that the virtual system felt like a real airbrush after using it for only a few minutes. Some portions of the system that they liked were easy color selection, rapid paint drying, and ease of altering the airbrush gun's parameters. The ability to arbitrarily zoom in on a surface was also noted as being very nice, as detail work not normally possible with an airbrush could be performed quite easily with the virtual system. The mock electronic airbrush (shown in Figure 3) was also remarked as feeling very similar to a real airbrush, which the artists liked.

Figure 9: *Canvas artwork created with the system. The stencils used are shown on the right. Artist: Leah Gall.*

The primary complaint about the system was the tracker latency (as well as the low frame rate in early versions). Airbrush artists were sensitive to any latency in the system, and even in the latest version artists reported that they slowed their brush strokes down slightly in order to compensate for the tracker latency. This accounted for the majority of the reported training time. However, artists stated that, in the end, the slowed strokes did not significantly impact their ability to make the artwork. Their adaptability is perhaps not surprising given the inherent latency (the time it takes the paint to reach the target) in a real airbrush system.

Two other qualitative outcomes from the user tests are also worth noting. First, correct reproduction of feathering effects at the end of an airbrush stroke is critical for successful airbrush simulation. Several of the artists focused on this aspect of the system when providing us with feedback, and the improved interpolation techniques described in Section 4.2 were, in part, the result. Second, artists were interrupted by the calculation time necessary to affix the stencil to an object surface (as shown in Figure 7), which currently takes a few seconds. When creating rapid stencil effects such as that shown in Figure 10, this slowed the artist more than using a real stencil. This calculation could be sped up in future versions for better interactivity.

7.2 Artwork

The primary use of the system is to allow an airbrush artist to paint using a computer system in the same way that they would paint on a real canvas or a real object. This work parallels the research done by [Baxter et al. 2001] for bristle brushes, only for the airbrush.

This is the baseline application for this paper, and it incorporates the simulation of spray particles, K-M mixing, and stencils mentioned above. Paint can be stenciled in, free handed, or both as the artist desires. This offers an artist all the benefits they normally receive from using an airbrush along with the advantages of using a computer: undo, zoom, easy stencil creation/loading, easy color selection, instant paint drying and mixing, and no cleanup. Figures 9, 11, and 12 all give examples of artwork created using the system by a real airbrush artist.

In addition to artwork created on a flat canvas, airbrushes are commonly used on 3D objects such as ceramics and automobiles. Figure 10 shows such a piece of artwork. Thus, the system can be used to prototype and/or practice artwork intended to be painted later with real airbrushes, as well as create pieces of virtual artwork.

7.3 Modeling

The final application for the airbrush simulation is computer model decoration. The airbrush tool allows artists to paint arbitrary 3D models with no training requirement other than their existing airbrushing skills. This opens up the possibility of using airbrush artists to paint models for video games and movies.

Even for artists who know how to use current graphics packages, airbrushing could provide an alternative method of decorating computer models, which has some advantages. First, it is a direct WYSIYG form of texture creation. The artist always sees exactly how the texture they are creating will look on the model as they create it. Second, it is a true 3D artistic input device: both the amount of paint sprayed on the surface and size of the painted area can be easily adjusted in a single stroke. Layered effects can also be created very rapidly. Finally, combining the concept of computer airbrushing with stencils allows artists to quickly generate artistic effects with the program that may be difficult to produce with current programs.

Figure 13 gives an example of using the airbrush to decorate models. An undecorated model is taken and painted by an airbrush artist. Note that detailed effects such as the eyes and facial mole can be very quickly and easily added by artists using the system, allowing pre-existing models to be easily detailed.

8 Future Work/Extensions

There are a few issues and modifications we would like to work on with the airbrush simulation in the future.

First, the use of stereo in the system could be further explored. As a close range 3D application, depth effects can have a strong influence on the user's perception, as noted in Section 7. In preliminary

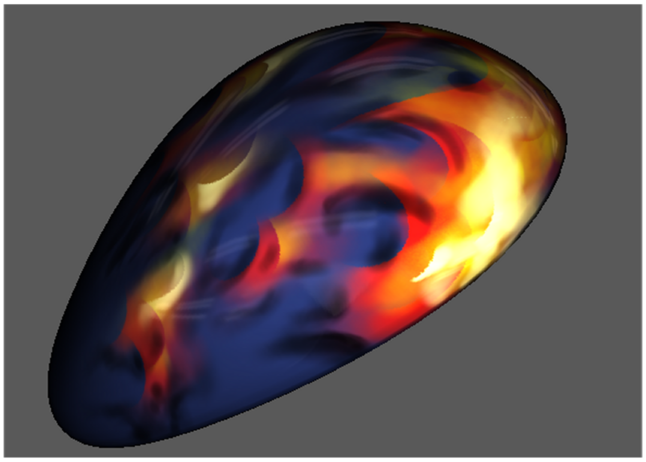

Figure 10: *Flames are drawn on a motorcycle gas tank, and a gloss coat added. Artist: Leah Gall.*

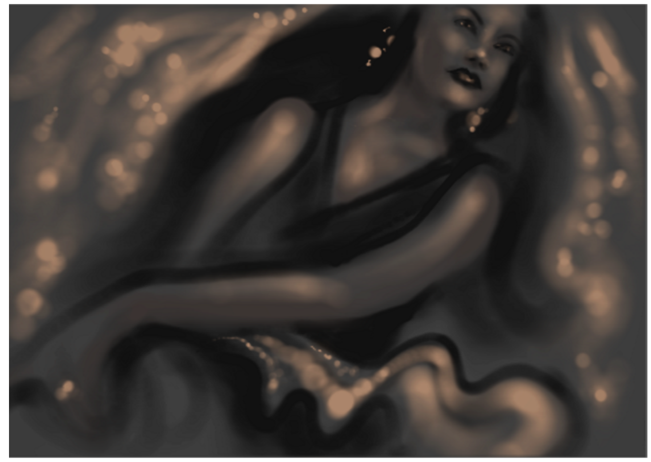

Figure 11: *Canvas artwork created with the system. Artist: Leah Gall.*

Figure 12: *Canvas artwork created with the system. Artist: Marc Dahlin.*

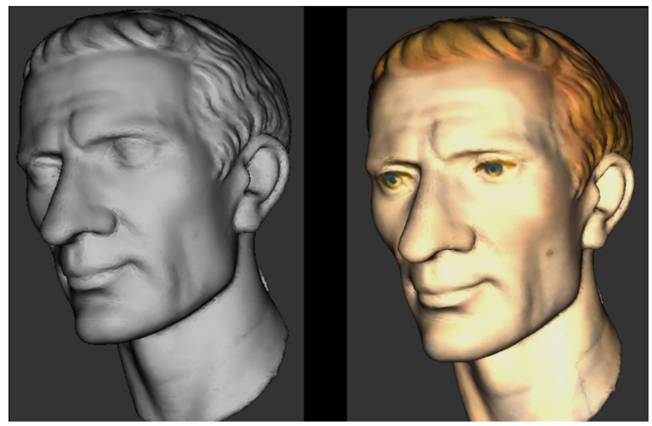

Figure 13: *3D model artwork created with the system. Artist: Leah Gall.*

tests, the system appeared much more compelling when used with 3D head goggles while attempting to paint on 3D objects than just looking at a flat monitor. It would be interesting to perform further testing on artist's ability to paint while using a stereo system versus non-stereo.

The current system also does not utilize audio. Although our test artists did not appear to use audio as a significant feedback cue, it could be added for system completeness. However, care would have to be taken that sound did not become an unnecessary distraction. It could also be interesting to run a study with and without audio feedback to see if the artists do indeed use the audio feedback in any way.

Currently, the stencil is attached to the surface by projectively texturing the stencil. While this allows the user to affix the stencil to the surface in a WYSIWYG manner, it is not a completely accurate representation of how the stencil would wrap around the object. More realism could be achieved with a more precise wrapping method.

Finally, while the current program provides an effective simulation of airbrushing for paints, there is no reason a computer airbrush program needs to be limited to spraying paint. The airbrushing paradigm could be modified to spray on different materials, textures, or even geometry such as fur and feathers onto a model. This could open up many new possibilities in speeding model creation and generating new and interesting artistic effects, all with an easy to use WYSIWYG interface.

9 Conclusion

This paper has introduced a new system for simulating airbrushing. It accurately models both the physical paint particles striking the canvas as well as the interaction between wet/dry paint layers on that canvas. In addition, an interface is provided that closely mimics both the look and feel of a real airbrush. This allows airbrush artists to use the system with no additional training beyond what they already know. The result is an easy to use interface that combines all the advantages of the real airbrush with the features a computer can provide such as easy color selection, undo, and arbitrary model/stencil selection.

The virtual airbrush combined with true three dimensional stenciling yields new and interesting possibilities for WYSIWYG model texture creation as well as possible additions into existing artwork interfaces such as Photoshop, Maya and BodyPaint. Use of this system can therefore be a new way of generating textures that have an artistic touch. This can allow existing airbrush artists to create faux material effects to either supplement or replace traditional rendering techniques.

In the future, it is likely that tracking technology such as the Nintendo Wii will become faster, more accurate, and accessible. This presents the possibility of providing the system to home users. With such commodity tracking hardware, anyone could train themselves how to airbrush and create artwork without the expense of air compressors, airbrushes, and paint.

Acknowledgements

The authors would like to thank artist Leah Gall who gave us excellent feedback on the system throughout its development and created many pieces of artwork for us.

We would also like to thank Gustavo Lira and Marc Dahlin for their helpful comments, and for providing us with several examples of airbrush training exercises and artwork.

This work was partially funded by NSF Grant IIP-0438693.

References

BAXTER, B., SCHEIB, V., LIN, M. C., AND MANOCHA, D. 2001. Dab: interactive haptic painting with 3d virtual brushes. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 461–468.

BAXTER, W., WENDT, J., AND LIN, M. C. 2004. Impasto: a realistic, interactive model for paint. In *NPAR '04: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*, ACM, New York, NY, USA, 45–148.

CAIATI, C. 1985. Advanced airbrushing techniques made simple. Tab Books.

CHU, N. S. H., AND TAI, C.-L. 2002. An efficient brush model for physically-based 3d painting. In *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, Washington, DC, USA, 413.

CURTIS, C. J., ANDERSON, S. E., SEIMS, J. E., FLEISCHER, K. W., AND SALESIN, D. H. 1997. Computer-generated watercolor. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 421–430.

DOMBEK, G., AND PORTER, T. 2003. Airbrush illustration for architecture. W. W. Norton and Company.

HAASE, C. S., AND MEYER, G. W. 1992. Modeling pigmented materials for realistic image synthesis. *ACM Trans. Graph. 11*, 4, 305–335.

HANRAHAN, P., AND HAEBERLI, P. 1990. Direct wysiwyg painting and texturing on 3d shapes. *SIGGRAPH Comput. Graph. 24*, 4, 215–223.

HERTZMANN, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 453–460.

HERTZMANN, A. 2002. Fast paint texture. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, ACM, New York, NY, USA, 91–ff.

KONIECZNY, J., HECKMAN, J., MEYER, G., MANYEN, M., RABENS, M., AND SHIMIZU, C. 2008. Vr automotive spray paint simulation. In *ISVC '08: International Symposium on Visual Computing*, Springer, New York, NY, USA, 998–1007.

KWOK, K. C. 1991. A fundamental study of air spray painting. PhD thesis, University of Minnesota.

MAURELLO, S. 1955. The complete airbrush book. Wm. Penn Publishing.

MISSTEAR, C. 1984. The advanced airbrush book. Van Nostrand Reinhold.

MITCHEL, D. 2008. Airbrushing 101. ArtKulture/Wolfgang Publications.

OWEN, P., AND SUTCLIFFE, J. 1985. The manual of airbrushing. Thames and Hudson.

RUDOLF, D., MOULD, D., AND NEUFELD, E. 2003. Simulating wax crayons. In *Proc. of Pacific Graphics*, 163–172.

SAITO, S., AND NAKAJIMA, M. 1999. 3d physically based brush model for painting. In *SIGGRAPH99 Conference Abstracts and Applications*, 226.