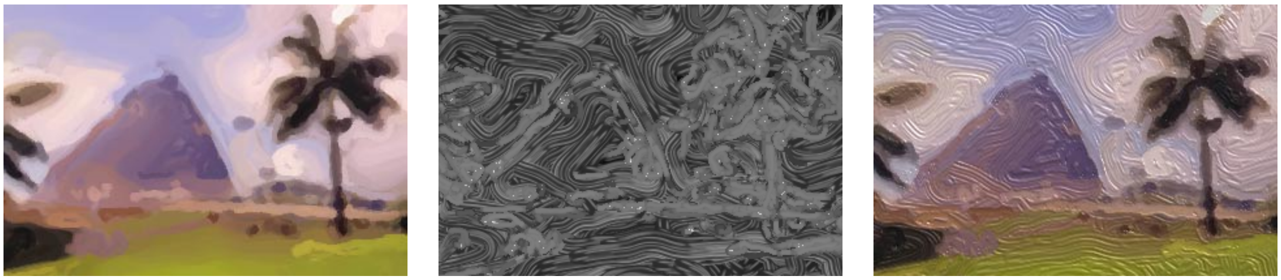
To appear, NPAR 2002: ACM Symposium on Non-Photorealistic Animation and Rendering

Fast Paint Texture

Aaron Hertzmann*
New York University



Original painting | Height field | Painting with lighting

Figure 1: Embossing a painting. A height field is computed by rendering every brush stroke with a height texture. A normal map is computed from the height field, and used for lighting the surface of the painting. The entire process takes only a few seconds with current graphics hardware. (The full painting is shown in Figure 5, top row.)

Abstract

We present a technique for simulating the physical appearance of paint strokes under lighting. This technique is easy-to-implement and very fast, yet produces realistic results. The system processes a painting composed of a list of brush strokes. A height map is assigned to each stroke, and a height field for the painting is produced by rendering the brush strokes textured with the height maps. The final painting is rendered by bump-mapping the painting's colors with the height map. The entire process takes only a few seconds on current hardware.

CR Categories: I.3.3 [Picture/Image Generation]: Computer Graphics—Display Algorithms; J.5 [Computer Applications]: Arts and Humanities—Fine Arts

Keywords: Non-photorealistic rendering, painterly rendering, paint simulation

*Currently at University of Washington CSE Department. Homepage: http://www.cs.washington.edu/homes/hertzman/.

1 Introduction

The physical simulation of media plays a dramatic role in the appearance of non-photorealistic renderings. Using stroke textures can make a simple stroke appear plausible; conversely, sophisticated stroke placements without texture appear "digital." As in other areas of computer graphics, the use of detailed texture can add significantly to the realistic appearance of images.

This paper presents a method for adding realistic paint effects to painterly renderings, by rendering strokes with plausible lighting. The main advantage of this method is that it is very fast and very easy to implement. Our system provides a "back-end" for rendering the brush strokes produced by a painterly rendering system, e.g. paintings produced from images [Hertzmann 1998; Litwinowicz 1997; Synthetik Software 1999] or 3D models [Daniels 1999; Litwinowicz 1999; Meier 1996].

Some researchers have developed systems simulating the flow and buildup of paint and other materials [Cockshott et al. 1992; Curtis et al. 1997]. These methods produce very high quality, sophisticated effects not available from simpler systems. However, the processing time required can be quite severe; moreover, implementing fluid dynamics can be a daunting task.

Some interactive painting tools [Baxter et al. 2001; Fractal Design 1998; Right Hemisphere Ltd 1999; Strassmann 1986] perform procedural paint simulation. However, these systems only require fast rendering of individual strokes. We

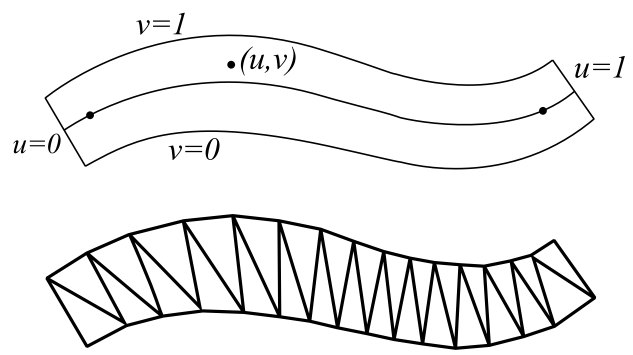


Figure 2: *Top:* Stroke model and texture coordinates. *Bottom:* Stroke tesselation as a triangle strip.

Height map

Opacity map

Figure 3: Brush height and opacity used for large strokes in this paper. Small strokes use a constant-height texture.

focus on processing entire paintings at once, where each painting may contain tens of thousands of strokes.

More commonly, brush strokes are rendered with fixed texture maps [Bronskill 2001; Litwinowicz 1997; Litwinowicz 1999; Meier 1996; Northrup and Markosian 2000] — this seems to be used by most "gallery effects" that use discrete brush strokes. While the strokes appear realistic when viewed individually, together they are less convincing, because the paint does not mix or build up on the surface, and a consistent lighting direction cannot be applied to the individual brush strokes.

Our approach is similar to using fixed texture maps for each stroke, except that we use height maps and then perform an extra lighting step. A global height field is computed for the image by rendering the strokes with the given height maps. This simple change leads to dramatic improvements in the image quality, since the brush strokes are rendered with consistent lighting. Furthermore, because surface normals are computed from the combined height field, boundaries between strokes appear as edges in the lit surface.

2 Algorithm

Our algorithm takes as input an ordered list of brush strokes, a shading model, and a set of stroke height maps. The algorithm renders a "painting" of the strokes. This process has the following steps (Figure 1):

1. A "raw" color image is produced by compositing the brush strokes.
2. A height field is computed that indicates the height of the painted surface at each pixel.
3. The resulting painting is generated by bump-mapping the raw color image, using the Phong shading model.

The first two steps of the process take advantage of graphics hardware for fast rendering.

In this paper, we use brush strokes produced from a source image by an image-processing technique [Hertzmann 1998; Hertzmann and Perlin 2000]. However, the brush strokes could come from any source, such as from a 3D renderer [Daniels 1999; Meier 1996].

Brush stroke model. The shape of a brush stroke is specified by a smooth curve that represents the stroke's spine (Figure 2). Each stroke has a specified brush radius (the screen-space distance from spine to edge), and a solid color. (A color texture map could also be used). Brush strokes are tesselated as triangle strips for rendering with graphics hardware. Texture coordinates (u,v) are defined for the stroke so that texture maps will stretch to fill the stroke without wrapping.

An opacity map and a height map are assigned to each stroke. The opacity data specifies the stroke's transparency, and is used both in the raw color image and the height field. One or more opacity/height map pairs are defined in advance for each stroke size, and assigned randomly to the strokes with that size. In this paper, we use one opacity/height pair for large strokes (shown in Figure 3), and a constant height map for small strokes. MIP-mapping hardware automatically scales the textures to the brush sizes.

Color image computation. We first render a color image that contains the paint color for each pixel, without lighting effects. This is done simply by compositing every brush stroke in back-to-front order, using the opacity map and paint color for each stroke.

Height map computation. Next, we compute a height field that contains the per-pixel height of the painted surface. The height field is computed as a normal grayscale rendering using graphics hardware. The image is initialized to black (zero height), and every stroke is drawn with ordinary alpha-blending. The gray values of the stroke are determined by a

Figure 4: Source images used for the painterly renderings in this paper.

texture map (e.g. Figure 3, top), plus a constant value that is proportional to the number of strokes already drawn. Hence, the first strokes are drawn very dark (shallow in the height field) and the later strokes are drawn light. The constant offset ensures that boundaries between the two appear as height discontinuities in the height field (illustrated in Figure 5, bottom row). The opacity of the stroke is determined by the opacity map (e.g. Figure 3, bottom).

Note that the height map is *not* cumulative. We experimented with adding stroke heights instead of compositing, but found it difficult to prevent hidden strokes from appearing in the resulting height field.

Lighting computation. Once the color image and height fields have been computed, the final painting is rendered. For each pixel, the surface normal is computed from directional derivatives of the height field. The lighting at each pixel is then computed under the Phong illumination model.

3 Results

Figures 1, 5, and 6 show our technique applied to several painting styles and source images. Adjusting parameters of the lighting and height field creates the suggestion of different types of paint.

For most images in this paper, we used the stroke height map shown in Figure 3. This height map was created in a few minutes using a paint program, but yields surprisingly compelling results. A more sophisticated collection of height textures could be created, such as with procedural texture synthesis or by scanning real strokes.

Each painting took under ten seconds to generate on a 1.5GHz Pentium processor with graphics acceleration, and includes 10,000-40,000 brush strokes.

Acknowledgments

We are grateful for input from Aseem Agarwala, Judy Kim, Jon Meyer, and David Salesin.

4 Future Work

This paper demonstrates that plausible lighting effects can be added to painterly rendering systems with relatively little effort or computational cost. There is substantial room to explore in this design space, by creating new types of stroke textures and new lighting functions. Of particular interest is the use of hardware vertex shaders to create new effects.

The paint textures in this paper have the shiny, static appearance of viscous oil paint; more varied styles could probably be achieved simply with suitable stroke textures in addition to height and opacity textures. Fast methods for simulating complex effects with watercolor and paint blending would also be useful.

References

BAXTER, B., SCHEIB, V., LIN, M. C., AND MANOCHA, D. 2001. DAB: Interactive Haptic Painting With 3D Virtual Brushes. *Proceedings of SIGGRAPH 2001* (August), 461–468.

BRONSKILL, J., 2001. System and method for drawing and painting with bitmap brushes, Mar. US Patent no. 6,201,549.

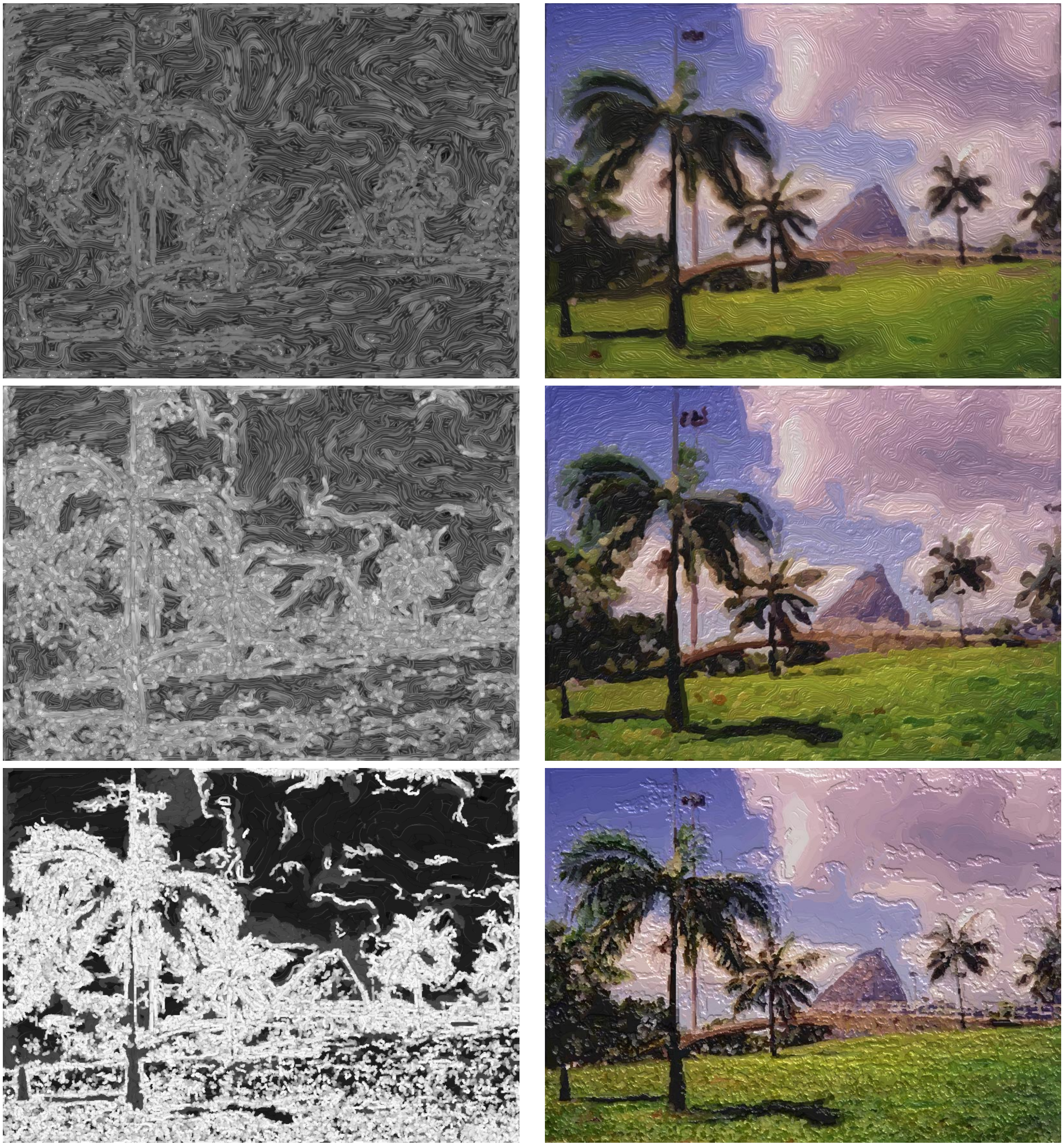
COCKSHOTT, T., PATTERSON, J., AND ENGLAND, D. 1992. Modelling the Texture of Paint. In *Computer Graphics Forum*, A. Kilgour and L. Kjelldahl, Eds., vol. 11, 217–226.

CURTIS, C. J., ANDERSON, S. E., SEIMS, J. E., FLEISCHER, K. W., AND SALESIN, D. H. 1997. Computer-Generated Watercolor. *Proceedings of SIGGRAPH 97* (August), 421–430.

DANIELS, E. 1999. Deep Canvas in Disney's Tarzan. In *SIGGRAPH 99: Conference Abstracts and Applications*, 200.

FRACTAL DESIGN, 1998. Painter 4.0. Software package.

HERTZMANN, A., AND PERLIN, K. 2000. Painterly Rendering for Video and Interaction. In *Proceedings of the*



Height fields Lit paintings

Figure 5: Paintings with different styles of lighting and stroke textures. In the bottom row, the stroke textures are disabled in order to exaggerate the effect of height discontinuities at the boundaries between strokes.

Figure 6: Paintings with different lighting styles. The bottom row demonstrates a more unusual height map. (The middle row is also shown in the color plates, Figure 8.)

First Annual Symposium on Non-Photorealistic Animation and Rendering.

HERTZMANN, A. 1998. Painterly Rendering with Curved Brush Strokes of Multiple Sizes. In *SIGGRAPH 98 Conference Proceedings*, 453–460.

LITWINOWICZ, P. 1997. Processing Images and Video for an Impressionist Effect. In *SIGGRAPH 97 Conference Proceedings*, 407–414.

LITWINOWICZ, P. 1999. Image-Based Rendering and Non-Photorealistic Rendering. In *Non-Photorealistic Rendering*, S. Green, Ed., SIGGRAPH Course Notes.

MEIER, B. J. 1996. Painterly Rendering for Animation. In *SIGGRAPH 96 Conference Proceedings*, 477–484.

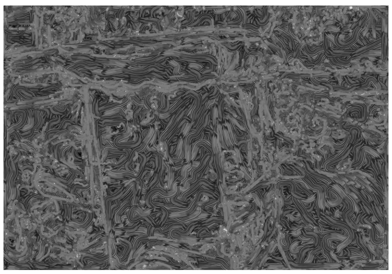
NORTHRUP, J. D., AND MARKOSIAN, L. 2000. Artistic Silhouettes: A Hybrid Approach. *NPAR 2000 : First International Symposium on Non Photorealistic Animation and Rendering* (June), 31–38.

RIGHT HEMISPHERE LTD, 1999. Deep Paint. Software package.

STRASSMANN, S. 1986. Hairy Brushes. In *Computer Graphics (SIGGRAPH '86 Proceedings)*, vol. 20, 225–232.

SYNTHETIK SOFTWARE, 1999. Studio Artist 1.1. Software package.

Original painting



Height field

Painting with lighting

Figure 7: Embossing a painting. A height field is computed by rendering every brush stroke with a height texture. A normal map is computed from the height field, and used for lighting the surface of the painting. The entire process takes only a few seconds with current graphics hardware.

Figure 8: Two lighting and texture styles.