

Real-Time Non Photorealistic Paint Spreading using Stencil Volumes

Marc ten Bosch*
Digipen Institute of Technology

Abstract

I borrow the concept of a stencil volume from shadow volumes and apply it to the simulation of a fluid spreading on a surface, without resorting to a texture-based approach. I introduce stencil paint volumes that grow in real-time using a physically-based system. Points inside the paint volumes are rendered in color, and those outside in black-and-white. This mimics the known artistic effect of a monochrome image with colored highlights to emphasize certain areas, found in painting, film and other media.

1 Introduction and Related Work

The shadow volume method of shadow determination was introduced by Crow [1977]. He creates volumes containing all points shadowed by an object by extruding the contour edges of that object away from the light source. Heidmann [1991] adapted shadow volumes to hardware acceleration and as a result introduced stencil volumes as a method of determining the shadowed regions of a scene. After rendering the volumes the stencil buffer holds a mask, which separates the pixels that lie outside the volumes from the ones that lie inside.

I extend the stencil volumes method to a different application, namely the rendering of a fluid spreading on an arbitrary surface in real-time. In particular, to simulate the spreading of paint, stencil paint volumes are created and updated using a physically-based model. This technique provides significant advantages over a texture-based approach. It borrows from shadow volumes in that it is not prone to aliasing problems. It provides a more dynamic effect without the cost of storing individual frames of animation, and is independent of the surface's mesh tessellation. Moreover, for an artistic effect reminiscent of paint spreading on a canvas I render the inside of the painted region in color and the outside in monochrome.

2 Construction and Update

The paint volume is initialized to a spherical mesh. The vertices on that mesh form a particle system. The simulation of particle movement is done by integrating the equations of motion. For maximum performance, Euler integration is used. A pressure force maintains the sphere's radius around a desired value, while an elastic force maintains the particles close to their original positions. Undesirable transient oscillations are damped by a friction force. The particles are initially at rest, but are then randomly perturbed to simulate the

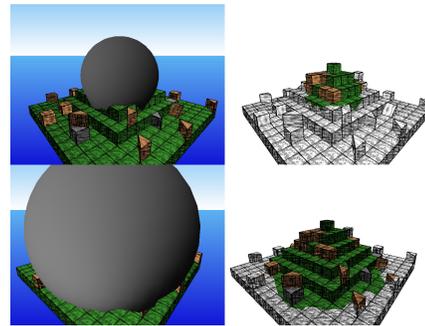


Figure 1: Visualization of paint volumes (left), final result (right).

material microstructure of the surface they lie on. The paint volume grows and shrinks by varying the desired resting radius. To simulate additional paint being added to the volume, causing it to spread, I slowly increase the desired radius. Additional forces can be applied to take into account the surface's shape, and in particular to keep the volume in a certain region of space. For greater control at minimal performance cost one can split a single volume into multiple smaller, less detailed volumes. My implementation arranges the volumes in grid cells that can spill from one to the other in a manner similar to classical fluid simulations.

3 Rendering

For a non photorealistic look reminiscent of a drawing I use a simplified form of real-time hatching [Praun et al. 2001] on top of texture and color to convey lighting. To increase efficiency I render both the grayscale and colored versions of the scene in the same pass. To do this, I use programmable hardware to set the alpha component of every fragment to the grayscale value of the computed color. After the stencil buffer is set by rendering the paint volumes, I set the stencil test to keep only the fragments that lie outside the paint volumes (with stencil value equal to zero). I then render a screen aligned quad using the alpha saturate blend mode of OpenGL to copy the alpha component over the red, green and blue components. Multiple passes can be accumulated to define more than two regions. For shadow volumes, soft shadows can be created by accumulating the effects of multiple clustered light sources [Heidmann 1991]. For paint, the effects of multiple concentric paint volumes can be accumulated to render washed out or tinted paint.

References

- CROW, F. C. 1977. Shadow algorithms for computer graphics. In *Proceedings of ACM SIGGRAPH 77*, ACM Press, 242–248.
- HEIDMANN, T. 1991. Real shadows, real time. *Iris Universe 18*, 28–31.
- PRAUN, E., HOPPE, H., WEBB, M., AND FINKELSTEIN, A. 2001. Real-time hatching. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press, New York, NY, USA, 581.

*e-mail: mbosch@digipen.edu