Real-Time Cartoon Rendering of Smoke

Morgan McGuire*, Andi Fein, Colin Hartnett — Brown University

Overview

We render amorphous shapes like smoke, clouds, and fluids with two-tone shading, self-shadowing, and silhouettes in the style of a cartoon. Our method operates directly on a particle system without ever forming a surface mesh. It renders only five polygons per particle and executes primarily on the GPU. It can be extended to isosurfaces and volume data sets, and has applications in games, animation, and visualization. Our implementation renders thousands of particles at 30 FPS and never reads the frame buffer.

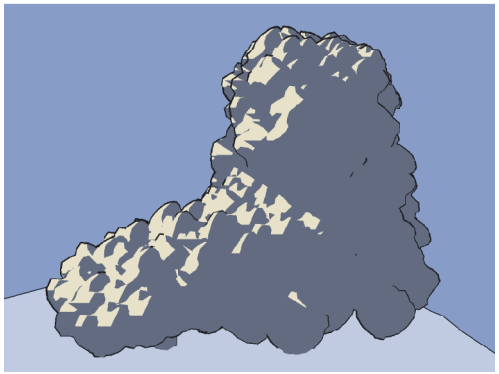

FIGURE 1: *Shading reveals the billowing shape of cartoon smoke.*

Preprocessing

We prerender a normal map and depth map for a single smoke puff using a bundle of random spheres as our geometric model. For variation, we repeat this process four times and then pack the results into a single texture map. The 3D geometry is then discarded; only billboards generated using the normal and depth maps are used for real-time rendering.

Each particle in the smoke simulation is permanently assigned one of the four variants of the smoke puff, providing diversity and frame coherence.

FIGURE 2: *Normal map, depth map, and final toon-shaded puffs.*

Two-tone shading

The body of each particle is rendered as a billboard parallel to the image plane. The pixel processor performs diffuse shading using the normal map and then quantizes the result using a dependent read against a 1D texture (i.e., a palette).

Outlines

To produce black silhouettes and some internal contour lines for shape we add a second billboard behind each particle. The second billboard is slightly larger, shaded entirely black, and offset along the view vector by a small distance, *d*. The thickness of the outline increases with the relative size of the black billboard. The number of internal contours drawn decreases as *d* increases; a reasonable value for *d* is the world-space width of the billboard.

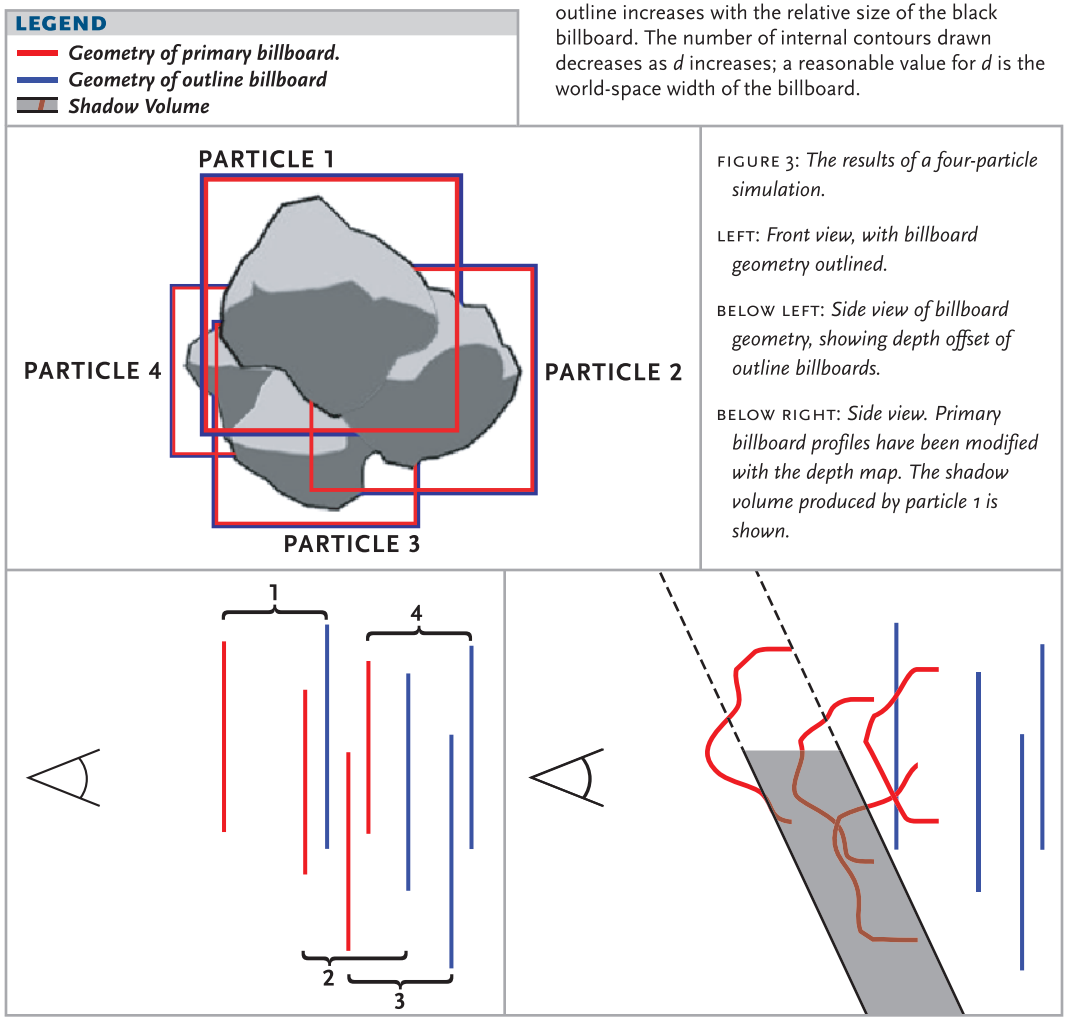

FIGURE 3: *The results of a four-particle simulation.*

LEFT: *Front view, with billboard geometry outlined.*

BELOW LEFT: *Side view of billboard geometry, showing depth offset of outline billboards.*

BELOW RIGHT: *Side view. Primary billboard profiles have been modified with the depth map. The shadow volume produced by particle 1 is shown.*

Self-shadowing

Thick smoke exhibits self-shadowing, a particularly striking effect when the smoke is between the viewer and the light source so that only the fringes of the smoke receive illumination. We use a special case of Crow's shadow volume method and extend it with nailboards. For each smoke puff, we create the shadow volume that would be cast by a viewer-facing cube. This shadow volume contains only one front- and one back-facing polygon, aligned so that they exactly overlap in image space. These shadow volumes meet the billboards at straight boundaries, revealing the billboard shape. To create a wavy intersection, we modify the depth value of each billboard and shadow-volume pixel using the prerendered smoke puff depth map.

FIGURE 4: *In the image on the left, the 2D billboard geometry is revealed where the puffs intersect the ground. Nailboards restore depth information, causing the smoke puffs to appear 3D (right).*

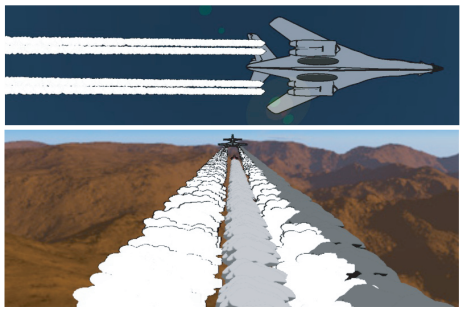

FIGURE 5: *Vapor trails rendered using the fixed-function variant of our method.*

Fixed-function variant

As an alternative for graphics cards that do not support pixel programs, we implemented a fixed-function variant of the algorithm. Rather than prerendering normal and depth maps, this variant renders four 2D sprite textures each frame directly from the puff model. This approach has two major drawbacks: it only considers directional lights, and self-shadowing is not possible.

* morgan@cs.brown.edu

BROWN

SIGGRAPH2004