Eurographics/ACM SIGGRAPH Symposium on Computer Animation (2009)
E. Grinspun and J. Hodgins (Editors)

Posters and Demos

Animating Smoke as a Surface

Tyson Brochu[†] and Robert Bridson[‡]

University of British Columbia, Vancouver, Canada

Abstract

We present a method for animating highly-detailed smoke by advecting a deformable surface through a procedurally-generated velocity field, avoiding costly volumetric simulation. We present three applications: a plume of thick, nearly opaque smoke; thinner, curly smoke; and a heavy, fog-like layer of smoke.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Animation

1. Introduction

In visual effects, fluids such as smoke are often simulated with a grid-based fluid solver or particle system. The density of soot is usually tracked using either a set of marker particles, or a volumetric density function on an Eulerian grid. If a particle system is used, tens of millions of particles may be required to prevent a grainy, bumpy, or blobby look, which can occupy considerable storage and network resources. To adequately capture fine, wispy detail, a volumetric grid would have to be of similarly high resolution. If the smoke effect is particularly dense, only the set of particles or grid nodes near the surface may be of interest, while the interior is almost completely occluded.

Our approach models smoke by defining a surface mesh which encloses a volume of soot. Conceptually, we consider the surface to be embedded in a fluid (the air). However, we do not model the entire fluid domain — instead we compute and apply procedural forces only at surface vertices. This allows us to achieve detail at the resolution of the surface mesh, rather than at some grid resolution.

2. Procedural Fluids

Procedural methods for animating fluids have been widely used for many years, starting with "flow primitives" for particle systems, introduced by Sims [Sim90] and Wejchert and Haumann [WH91]. Our method uses the linear superposition of several fluid flow primitives to generate plausible smoke without an Eulerian fluid simulator. We use *vortex rings* to generate upward motion, *divergence sources* to simulate smoke dispersion, and *curl noise* [BHN07] to add extra small-scale detail. (Note that any procedural or simulation-based flow could be used.)

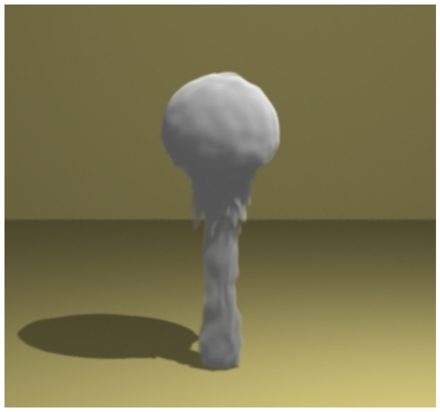

Figure 1: *We procedurally animate a dense smoke plume using only an adaptive triangle mesh advected through a procedural velocity field. No volumetric calculation is used.*

3. Surface Tracking

In order to use a triangulated surface to define a volume of fluid, we require a surface tracking system with a certain set of features. Obviously an adaptive surface is required to generate highly detailed results. We would also like to maintain good mesh quality in terms of aspect ratios and surface

[†] tbrochu@cs.ubc.ca
[‡] rbridson@cs.ubc.ca

smoothness. To meet these requirements, we use *El Topo*, a C++ implementation of Brochu and Bridson's method for explicit surface tracking [BB09]. This implementation also allows the option of maintaining an intersection-free mesh and handling topological changes in the surface if volumes of smoke overlap or separate.

4. Examples

We demonstrate our approach with three example scenarios. We first demonstrate a dense, "mushroom" plume dominated by several source primitives and a strong vortex ring (figure 1). Second, we generate a thinner, curly plume dominated by curl noise (figure 2). Finally, we apply only curl noise to a flat, open surface. In figure 3, only the top surface of the volume is simulated, with the sides added at render time.

We render each example with our own ray tracer, using a single-scattering volumetric shader. When rendering dense smoke, our volume shader is optimized to gather only a few samples near the surface, treating contributions from the deep interior as occluded. For *very* dense smoke, our volumetric renderer could potentially be replaced with a sophisticated surface shader or a combination of surface and volume shading to obtain more detail.

Using our approach, the integration of surface vertices takes a few seconds per frame, up to a minute per frame if collision detection and resolution are used to prevent self-intersections. Although adaptively refining and improving the mesh and handling surface collisions introduces some computational cost, the complexity of our simulation method scales linearly with the resolution of the 2-dimensional surface mesh rather than with the resolution of an underlying 3-dimensional grid or particle system. Even in thin, wispy examples, this represents significant savings since far fewer mesh vertices than separate particles are required to produce a smooth, non-grainy render without excessive blur.

5. Conclusion

We have presented a technique for procedurally modeling dense smoke as a surface. We define the region of dense soot with a closed surface, rather than with a particle system or volumetric density grid. We compute and apply velocities only at the vertices of this surface, effectively focusing computational effort on the visible regions.

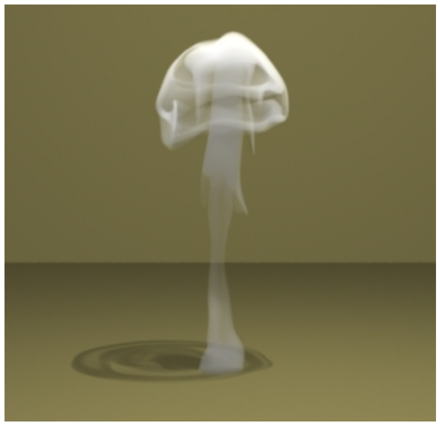
Figure 2: *We animate thin, curly smoke using a combination of vortex rings and curl noise*

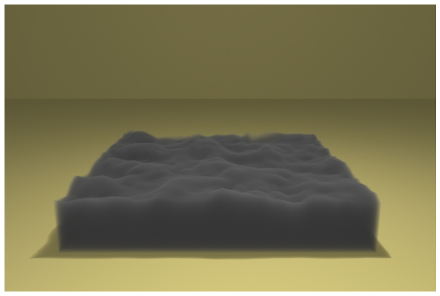
Figure 3: *Applying our method to an open surface allows us to model larger volumes*

References

[BB09] BROCHU T., BRIDSON R.: Robust topological operations for dynamic explicit surfaces. *SIAM Journal on Scientific Computing 31*, 4 (2009), 2472–2493.

[BHN07] BRIDSON R., HOURIHAM J., NORDENSTAM M.: Curl-noise for procedural fluid flow. *ACM Trans. Graph. (Proc. SIGGRAPH) 26*, 3 (2007), 46.

[Sim90] SIMS K.: Particle animation and rendering using data parallel computation. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1990), ACM, pp. 405–413.

[WH91] WEJCHERT J., HAUMANN D.: Animation aerodynamics. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1991), ACM, pp. 19–22.