Visual Simulation of Wispy Smoke

Christopher Batty
cbatty@franticfilms.com
Frantic Films

Ben Houston
ben@exocortex.org
Neuralsoft

1 Introduction

Several scenes in the film *Cursed* called for wispy smoke to interact closely with characters. Simple particle systems failed to capture the characteristic motion of wispy smoke, while existing smoke simulators generally produced smoke of a diffuse nature, more appropriate for explosions or large flames than cigarette or incense smoke. This sketch describes our implementation of a flexible, artist-friendly smoke simulator capable of producing realistic wispy smoke for a production environment.

Figure 1: (a) Real and (b) simulated wispy smoke. (c) Simulated smoke in the movie *Cursed*.

2 Capturing Wispy Details

Recent work in smoke simulation [Fedkiw et al. 2001] has used density fields to represent smoke, requiring very high resolutions to capture small-scale smoke details. We therefore elected to track smoke density using individual smoke particles, adopting the fluid and particulate model of [Feldman et al. 2003], but dispensing with the combustion components. In addition to being a physically plausible model, particles can track the crisp, detailed contours that would get lost or blurred in a density field, due to both artificial numerical dissipation and insufficient grid resolution. This allowed us to perform simulations at reduced resolution while maintaining excellent visual quality.

The naïve approach of seeding particles at the start of each timestep resulted in obvious aliasing artifacts. We resolved this by developing an anti-aliasing method taking into account both the fluid velocity and the smoke source's velocity. For each particle emitted, we choose a random time within the current interval, time-interpolate the initial seed position, and finally advect with the current fluid velocities over the remainder of the time step to "catch up" to the correct time. This deceptively simple technique was vital to achieving smooth continuous wisps.

In order to accelerate our simulations, we implemented a moving bounds approach, as in [Rasmussen et al. 2004], which allowed us to simulate a smaller region. We then combined this with a variation on that paper's grid-sourcing method. We used a moving, high-resolution simulation near the smoke source and region of interest, which acted as a source for an encompassing, lower resolution simulation. Open boundary conditions were applied on the small simulation to allow smooth outflow, and the small simulation provided velocities and Neumann boundary conditions for the larger simulation. This approach injects high resolution detail into the smoke particles, which is subsequently retained in the transition to lower grid resolution.

3 Artistic Controls

In order to provide the maximum degree of control and flexibility to artists, we implemented a variety of mechanisms within a 3DS Max plugin. All simulation objects in the scene are tagged with (optionally animatable) information indicating their type and attributes. Smoke particle sources create particles within a region at a given emission rate, as well as specifying the initial temperature there. Conversely, smoke erasers are used to delete particles that enter a particular region. Objects tagged as velocity modifiers either explicitly set or increment the contained velocities by a given vector on each simulation step. To create explosive/implosive forces we implemented pressure sources and sinks as in [Feldman et al. 2003], using their modified Poisson equation to generate divergent velocities. To support interacting objects (critical for our setting), we used the constrained velocity extrapolation approach [Houston et al. 2003] for setting proper object boundary conditions. (For *Cursed*, character meshes were animated to match the movement of the live actors, and then used in simulations.) By augmenting our simulator with this array of tools, we were able to generate production quality simulations with the desired look and behaviour.

4 Rendering

In our simple lighting and rendering model, smoke particles were rendered directly to an initially empty accumulation buffer accounting for camera and fluid motion blur, as well as particle ages. Initially, a physically-based opacity function was used to accumulate the particles' opacity. Later, to give compositors more flexibility, a simple additive accumulation of the particles was used. Camera motion blur was implemented by linearly interpolating between a number of camera samples. Particles were advected forwards and backwards in time using the fluid velocity field of the frame to determine the fluid motion blur. We emulated dissipation using stored particle ages and a parameterized density decay function.

References

FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual simulation of smoke. In *Proceedings of SIGGRAPH 2001*, ACM, 23–30.

FELDMAN, B., O'BRIEN, J., AND ARIKAN, O. 2003. Animating suspended particle explosions. In *Proceedings of SIGGRAPH 2003*, ACM, 708–715.

HOUSTON, B., BOND, C., AND WIEBE, M. 2003. A unified approach for modeling complex occlusions in fluid simulations. In *SIGGRAPH 2003 Sketches and Applications*, ACM.

RASMUSSEN, N., ENRIGHT, D., NGUYEN, D. Q., MARINO, S., SUMNER, N., GEIGER, W., HOON, S., AND FEDKIW, R. P. 2004. Directable photorealistic liquids. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, ACM, 193–202.