Simulating Massive Dust in *Megamind*

Koen Vroeijenstijn   Ronald D. Henderson

DreamWorks Animation

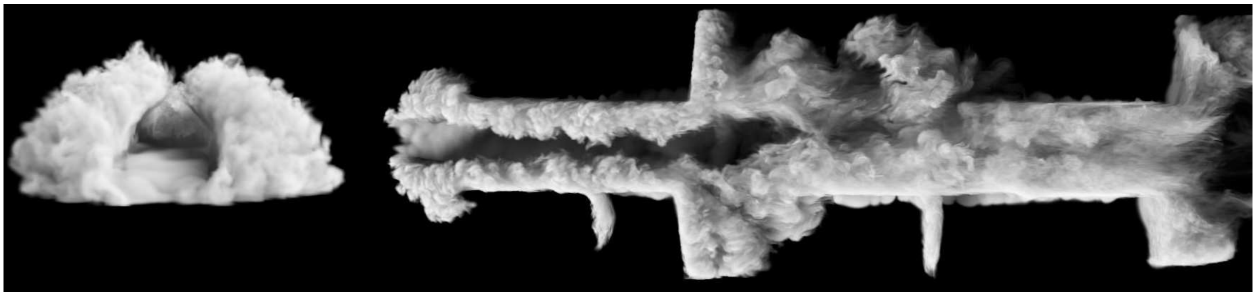

Figure 1: *Massive dust cloud animated using our fluid simulation framework at a resolution of* $1200 \times 195 \times 590$*, which runs in 52 seconds per frame on a single workstation. This is a volume render of the simulated density field (front view and top view).*

In Dreamworks Animation's *Megamind*, the production faced a challenge in creating a shot where the top of a skyscraper is torn off and hurled down a city street, destroying portions of the city and creating a massive dust cloud. Development for this shot inspired a new fluid simulation framework with significant improvements in four areas: speed, visual quality, setup flexibility and artistic control. In this talk we describe the simulation algorithm and the framework that allowed it to be used in our effects pipeline.

Fast, Direct Simulation

Our simulation algorithm is based on integrating the incompressible Navier-Stokes equations, but with several modifications to classic approaches. We represent all solution variables (velocity, pressure, density, etc.) as discrete volumes sampled on a regular, non-staggered grid. In addition to standard buoyancy and external forcing terms, we also include an explicit divergence control field which is incorporated into the pressure solve. We use a MacCormack method with a local limiter for advection [Selle et al. 2008] and treat all other terms implicitly, taking advantage of a Fast Helmholtz Solver for all elliptic systems. The resulting algorithm is unconditionally stable, requires constant solve time per frame, and both memory and computational time scale close to linearly with grid size. All stages of the solver pipeline have a high theoretical parallel efficiency.

There are several advantages to this approach. Unlike first-order upwind methods, the MacCormack method introduces predominantly dispersive rather than diffusive errors, which is far more forgiving when simulating "turbulence." Results generally demonstrate very little numerical diffusion. By including physical diffusion, we explicitly control a small-scale cutoff and achieve excellent correspondence between high- and low-resolution simulations. Simulations can be tweaked at a lower resolution, running at almost interactive speeds, knowing the high resolution version will match the overall motion. Fast solve times allowed us to run direct simulations at nearly one voxel per pixel without resorting to advection techniques to add detail.

Everything is a Volume

All inputs to the simulation framework are represented as discrete volumes, including an arbitrary number of dynamic scalars, forces, and source terms. Collision data is represented as a velocity field and voxel mask. A simulation can be created using existing tools and techniques to populate volume buffers. Since most artists are comfortable generating, manipulating and analyzing volumes, a lot of the guesswork is removed from the simulation setup. Execution can be represented as a node graph where manipulation of the internal solver state is mixed with other volume operations. This encourages artistic exploration since the complete dynamic state can be manipulated to sculpt the simulation in creative ways as it runs.

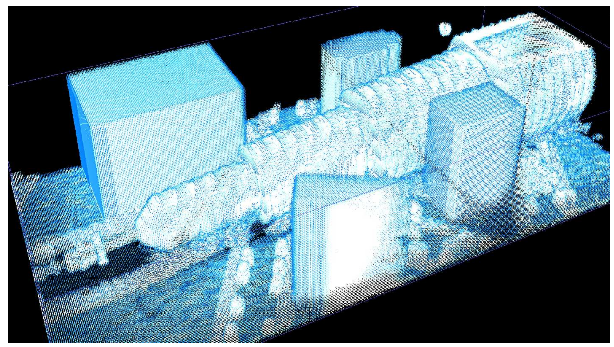

Figure 2: *Visualation of the input collision volume for the simulation environment. Each voxel is repesented as a single point.*

Putting It All Together

In the *Megamind* setup we used only three volume inputs to control the simulations: density, collision, and divergence. Density sources were created from particle simulations starting at the contact line between the tower and city street, and then rasterized into volumes. Deforming and moving geometry was rasterized to create collision volumes. In addition to coupling the solver with the animation environment, we show that manipulating the collision mask can be an effective way to drive additional turbulence that is superior in both look and computational cost to just adding "noise" to the simulation. By setting positive divergence in areas with higher density, we were able to create violent expansion in specific areas of the dust cloud. In addition to a breakdown for the tower destruction, we provide implementation details for how we created an interactive setup and simulation environment and demonstrate scene configurations typical of other production shots.

References

SELLE, A., FEDKIW, R. KIM, B., LIU, Y. AND ROSSIGNAC, J. 2008. An unconditionally stable MacCormack method. J. Sci. Comput. 35, 350-371.

{koen.vroeijenstijn, ron.henderson}@dreamworks.com

SIGGRAPH 2011, Vancouver, British Columbia, Canada, August 7 – 11, 2011.
ISBN 978-1-4503-0921-9/11/0008