Growing Up with Fluid Simulation on "The Day After Tomorrow"

Jason Iversen
Digital Domain

Ryo Sakaguchi
Digital Domain

Introduction

Fluid Simulation is a popular topic in the CG industry and there are many considerations, pitfalls and caveats to properly reap the benefits of physical simulation. This sketch attempts to discuss many of these in the context of the production "The Day After Tomorrow". The production required the photo-realistic destruction of New York City by a massive storm surge and the need to develop a fluid simulation solution seemed necessary to achieve that high a level of realism.

Technical Considerations: What can we create?

Using experience with previous incarnations of our in-house fluid simulator, we quickly decided that with new technology published on particle level sets and computational fluid dynamics algorithms that a new product should be developed to produce the quality of simulation required by this project.

Some of the output requirements from the simulator were obvious: *level sets* gave us an accurate water/air interface, and *vector fields* described the water velocity. Experience with the level sets showed that we would need to develop further techniques to address the aesthetic needs. Whilst they provide accurate and smooth surfaces, the detail on the surface was not great enough for the sheer scale of the water in the shots. Here we found a solution to parameterize the water surface so that more detail could be added to the surface of the resultant simulations. By flowing particles along the velocity fields, we could use their proximity to the water surface to define a local texture space per particle. We relied on this texture space to adequately handle the water branching and merging and so these particles were read directly into the renderer to create additional displacement and colouration. These particles were also used to float debris on the water surface.

The output resolution of the water surface was on average in the range of about half a million polygons and we still could glean more resolution from the level set, and so we looked into rendering the level set directly. Firstly, we investigated rendering the level set directly by ray-tracing against the zero contour defined in the level set, and secondly by generating the polygons at a high resolution at render time by means of Procedural DSO's in the renderer. Each of these had significant drawbacks for the benefit of the increased resolution: ray-tracing against the level set yielded a smooth surface but proved difficult to motion blur, and the procedural render-time method did not yield significantly better detail. Neither method accounted for the post-processing required to achieve the shot treatment.

Creative Considerations: Beating it into shape.

Choreographing simulations has proved to be a notoriously difficult task and we found it necessary to split this task into balancing the inputs and the initial conditions to the simulator, affecting the simulation itself with strategic modifications to the fluid solver, manipulating the copious simulation output data, and finally by processing the final rendered images. Techniques used involved the addition of water/boundary friction, the capability to override vector-fields during the simulation, variable viscosity across the water volume, warping the output geometry and accessing the vector-fields and level sets in the original rest space, temporal filtering of surface features to reduce noise and artifacts, re-timing of the output data using both interpolation of the geometry across the velocity-field and the final processing of the output images by performing optical flow analysis for re-timing and motion blur purposes.

To achieve the director's wish of a swell passing through the city, we had to tame the initial energetic results by flowing low-energy simulations in fairly featureless environments which produced predictable data we could post-process effectively, while taking care not to affect the life and apparent viscosity of the water. Much of the destructive message of the storm surge was left to the how we treated the whitewater.

Whitewater: The final ingredient.

The whitewater splashes were rendered as a volumetric solution with their motion driven mostly from the collision of the simulated water surface with the CG New York City buildings and other debris geometry. An artist-tweakable splash velocity vector was generated from the average vector of the water velocity and collided surface normal, giving a reliable good first pass at secondary splash particle motion. Hero splashes required more care and hand animation, and so we built a splash library of these in both 3D and 2D. Particles were also generated from areas of high level set curvature. Populating the splash voxel-buffer based on these particles by drawing random splashes directly into the buffer proved far more effective than trying to calculate droplet connectivity using metaballs. We also achieved good looking specular reflection "glints" on the droplets by boosting the lighting on a random subset of droplets- with no time coherence.

Figure 1. A frame from a key sequence utilizing fluid simulation

Conclusion

We discovered that a project of this nature requires a blend between realistic physical simulation and artistic direction, which in turn requires not only a reliable and accurate fluid simulation, but a large reliance on a well thought-out work-flow and a complete tool set for manipulating the simulation before, during and after the simulation phase.