So Real It'll Make You Wet

Willi Geiger* Industrial Light + Magic
Mohen Leo* ILM
Nick Rasmussen* ILM
Frank Losasso* Stanford University
Ron Fedkiw* Stanford University

For "Poseidon", we had to create multiple shots of photorealistic ocean water interacting with computer-generated objects. The challenge was to create water that was seen both on a huge scale and in close-up, and to go beyond any work previously seen.

We solved the following novel problems: rendering a smooth high-resolution tesselation of a levelset and particle set with procedural displacement, surface texture and internal volumetrics; simulating and rendering multiple secondary particle sets driven by a physically based fluid simulation.

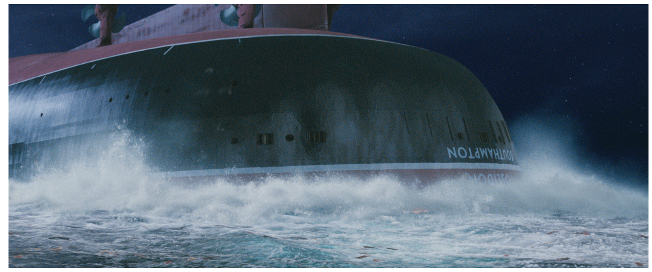

Figure 1: CG ocean and boat at night (Branko Grujcic)

1 Creating the Water Surface

The foundation of our approach is a particle levelset (PLS) fluid simulation engine developed in collaboration with Stanford University. The output is a levelset that represents the main body of fluid, splash particles where the detail of the fluid surface is too fine to be resolved by the levelset and bubble particles where air is trapped inside the fluid.

Rendering the combined PLS output is not straightforward. Ray tracing can efficiently render the levelset, but ray tracing an implicit surface defined by a large particle set is more problematic. Achieving a smooth visual transition between the levelset and the splash particles is difficult, as are additional effects such as displacement to create small ripples and capilliary waves. It also makes it harder to leverage existing toolsets for explicit surface rendering. The dual-contour method is a known and efficient way of tesselating a levelset, but it is prone to visual noise, as it operates at the resolution of the levelset, which is generally much coarser than pixel resolution, and it offers no immediate way to include the splash particles in a single blended surface.

We developed a new method of creating a single mesh from the levelset and splash particles. We start by recursively subdividing the simulation domain at all regions near the fluid surface. Only the current working branch is held in memory and when a leaf node that intersects the surface is found, it is added to a sparse voxel array. We achieve a very high effective voxel resolution because we only need to store a small fraction of the total volume to reconstruct the final surface at the fluid interface.

A quadrilateral mesh is contructed from the exterior voxel faces. Each mesh vertex is moved to the nearest point on the implicit surface, creating a mesh that approximates the implicit surface to a very high degree of accuracy. Finally, a simple mesh relaxation algorithm is run on the mesh to minimize any remaining artifacts, visually approximate surface tension effects and create a smooth blending between the two parts of the implicit surface: the levelset and the particles.

*e-mail: {wgeiger,mohen,nick,frankp,rfedkiw}@ilm.com

The surface is then either sent to the renderer as a quadrilateral subdivision mesh or triangulated, spatially subdivided and sent to the renderer in small pieces. The end result is that an arbitrarily complicated and detailed implicit surface with topology that changes from frame to frame can be rendered with reasonable time and memory requirements. The surface can be rendered with all the known tools for surface texturing, including shader displacement.

2 Rendering the Water Volume

The look of a turbulent ocean is determined by many other effects, including the volume of the water itself. An interior volume shader was used with the surface shaders, which simulated occlusion and scattering effects below the water surface. The bubble particles from the fluid simulation were used to create the characteristic clouds under the surface when the water gets churned up.

Complicated and detailed CG models had to be included in the water renders to hold out both the surface and interior volume. To conserve memory and increase flexibility, holdouts were implemented by rendering deep shadows of these objects from the shot camera, and then using these deep shadow holdouts in the water shaders.

3 Secondary Particles

Multiple secondary particle simulations driven by the output of the fluid simulation were used to create the detailed effects not directly represented by the fluid simulation: spray and mist in the air, and foam on the surface.

The splash particles emit spray particles throughout their lifespan, creating a far higher level of detail than can be resolved by the splash particles alone. Around 1 million spray particles were simulated for most shots, which were rendered as up to 100 million using render-time data amplification techniques. The spray particles have a finite lifespan, at the end of which they emit mist particles that are rendered as an atmospheric volume. In this way, a visual continuum is created such that the water transitions into spray which in turn transitions into mist.

The final secondary simulation was surface foam. When bubble or spray particles hit the water surface, foam particles are emitted. Around 5 million foam particles were usually simulated and advected by the velocity of the fluid to create a floating layer that forms the characteristic patterns of foam seen on an ocean surface. The water surface shader uses the foam particle set to create a layer of displaced, diffusely reflective foam.