Bubbles and Foam in *Partysaurus Rex*

Adam Harder
Pixar Canada

Chris Mangnall
Pixar Canada

Pixar's short film *Partysaurus Rex* required simulated soap bubbles – a lot of soap bubbles, including bathtub foamscapes consisting of more than 3 million individual bubbles. Bubbles and foam acted as sets and set dressing, floated on cascades of moving water, coated and interacted with characters, and burst when hit by droplets of water from a shower. All of this was accomplished with a combination of a Houdini bubble model and solver, a set of RenderMan shaders and plugins, and a great deal of painstaking shot work.

Figure 1: ©2012 Disney / Pixar. All rights reserved.

1 Modeling and Simulation

We began by generating bubbles as spheres whose radii were chosen stochastically. When two bubbles come into contact in the real world, their respective centres move closer together as surface tension causes the bubble film to maintain minimum area. In our model, we simulated this adhesive force by adding a spring between each pair of bubbles that were in contact. The rest length of a newly created spring was some user-controlled fraction of the sum of the radii of the two bubbles. Large numbers of bubbles joined together by these springs behaved as a cohesive foam. Springs were deleted when they were stretched past a user controlled *snap length*, typically 110% of the rest length. This allowed a foam (consisting of a large number of bubbles) to tear apart as characters or water currents pushed through it.

The system was solved using a *position-based dyamics* approach. The length constraints for the springs were solved using *successive over-relaxation*, and the system was moved forwards using a Verlet integration scheme. Collisions were resolved by moving colliding bubbles to the surface of the collider and inheriting the velocity from the colliding surface.

Large masses of bubbles were sculpted in Maya as polyhedral volumes, then used as stand-ins for Layout and Animation. In the FX department, these stand-ins were combined and voxelized using Houdini's volume tools. Procedural methods were used to add stochastic detail. The resulting volumes were then filled with bubbles, and passed to the bubble simulator.

To fill a volume with bubbles, we started by selecting a large cube of pre-simulated bubbles with a particular set of bubble parameter values. We had several such cubes containing different species of foam prepared for use in different situations.

A wide variety of foam species could be achieved by varying the bubble spring snap length and the *relaxation* parameter. Low values for the relaxation parameter resulted in more wobbly foam, higher values increased rigidity. Lower snap lengths caused more plasticity, with higher values causing a more brittle foam.

A soap scum layer of very small bubbles was generated using a particle system constrained to the water surface and advected using a fluid sim.

2 Shading and Rendering

One of the key visual characteristics of real life bubbles in contact with one another is the shape of the interface between them. For very simple systems of two bubbles, the interface forms a flatter spherical cap. These areas provide broader specular highlights which causes a characteristic twinkling in the foam when bubbles move and interfaces are created and destroyed. For larger systems, the interfaces are not necessarily spherical; however approximating them as spherical is more efficient and proved to be adequate for our purposes. The bubbles were rendered as displaced spheres, using a RenderMan procedural to convert particle caches from Houdini to RiSpheres at render time. A custom shading dso was written to read the particle caches. The displacement shader iterated through pairs of overlapping bubbles and displaced the surface of the current bubble to coincide with the computed spherical bubble interface surfaces.

Large, dense systems of bubbles are highly scattering, due to refraction through the meniscus along the many edges between interfaces. To avoid costly ray tracing, we approximated this effect using RenderMan point-based scattering, using one point per bubble. Deep shadow maps were generated for diffuse effects using the same shortcut. Each bubble was tagged with its depth from the surface of the bubble model, which allowed us to transition the shading from sharp, glossy reflections on the outside, to the much softer scattering approximation on the inside. A third blobby surface model was used for occlusion.

Masses of foam were often hollow, with an outer shell of bubbles surrounding an empty interior. This technique reduced the number of bubbles to be simulated and also improved rendering efficiency by reducing the number of transparent bubble surfaces. To guarantee that the renderer could not see through to the empty interior, the depth value tagged on each bubble was used to gradually increase opacity in the shader so that full opacity was reached at the innermost layer of bubbles.

3 Acknowledgements

We would like to thank Darwyn Peachey and our colleagues in the Pixar Canada FX team.

SIGGRAPH 2013, July 21 – 25, 2013, Anaheim, California.

ACM 978-1-4503-2261-4/13/07