Ocean Mission on *Cars 2*

Alexis Angelidis * Josh Anon † Gary Bruins‡ Jon Reisch § Esdras Varagnolo ¶
Pixar Animation Studios

On a mission to Dr. Z's lair in the middle of the ocean, secret agent Finn McMissile must face the harsh environment of a stormy ocean.

Early in the development of the film there was no interaction with ocean water. As production progressed the ocean became an important element of the fast paced action sequences, and the night lighting illustrated in Figure 1 demanded an expansion of our tools.

We discuss our challenges and solutions across multiple disciplines in addressing the technical challenges.

Figure 1: A shot from *Cars 2*.

1 Water Surface and Interaction

To represent the stormy water surface geometry we implemented J. Tessendorf's wave system [Tessendorf 2004], with a mechanism to split ranges of wave frequency. This allowed layout artists to provide a base ocean made of low frequency waves that would later be detailed in a seamless looking way. We implemented a procedural self-intersection healing mechanism that allows waves to look as cuspy as visually required. To avoid the memory cost of a finely tessellated mesh capturing the tiny detail of the waves, the geometry that was not represented by polygons was rendered as displacements. In most shots we used three to four wave trains to span the range of visible waves and avoid repetition.

To layout the shots, we established categories of shooting situations and built equivalent rigs on the wave system. In some shots, we needed to put the camera on a second boat, defined as a virtual boat with its own size and position. In other shots we needed a camera operator on the deck of the main boat. To account for the operator's delayed reaction to the waves, instead of constraining a camera to the main boat, we created a second virtual boat with identical size and motion parameters as the main boat but with offsets to the wave reaction parameters. We also discovered that we needed to modify the camera constraint mechanism to provide control over side to side motion. Feeling seasick was a subjective indicator that the parameters had been pushed too far and we needed to back off.

Simulating both the ocean water and the splash together would have been prohibitive. We decoupled the ocean from the splash with a frequency filter applied locally to remove the high and medium frequencies that would then be simulated. For ocean integration such as boat splashes and spray, we used Houdini's fluid flip solver.

*silex@pixar.com
†josha@pixar.com
‡bruins@pixar.com
§jreisch@pixar.com
¶esdras@pixar.com

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

2 Volumetric Lighting Approach

One of the challenges in lighting the ocean came when we decided to have refractions of the water below the surface. The primary goal was to describe depth, thickness, murkiness and subtle color shifts. We later expanded on that idea by including underwater explosions, props and sets with strong directional light sources. This enhanced the sense of scale, mystery and action the story was calling for. There were three main benefits of integrating the volumetric lighting approach with our surface lighting tools. First, we were able to reuse existing color and volume light shapers. Second, the crew was familiar with them and the learning curve was not steep. Third, lighters were able to customize depth, color, intensity and coverage on a per shot basis, with the utmost level of control, achieving in the end what would become the lighting vision for the sequence.

To reduce the overwhelming computation of illuminating a participating medium on either side of the water surface, several accelerations needed to be implemented, as shown in Figure 2. To illuminate the thin volumetric ambient fog, we polygonized at render time an implicit surface that defined a region as small as possible for rendering the volume (1). To attenuate light as depth increased, we filled the ocean below the surface with a volumetric shader to model the build up of opacity (4). To accelerate the distortion of reflections and refractions of volumetric lights, we reduced the number of calls to the volume shader by storing the volume data in a volume cache (2),(3). The fog density was modeled with a programmable volume shader [Scheepers and Angelidis 2008]. For fast moving headlights, we used the volume primitive of PRMan 15.0 that provides motion blur derived from a space constrained to the light source.

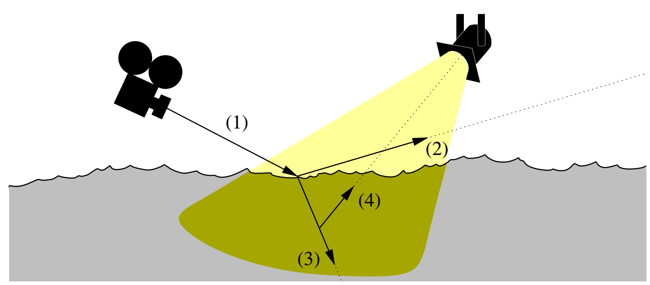

Figure 2: (1) Direct light computed by raymarching in an implicit surface. (2)(3) Volume refraction/reflection calculated in a shader cache. (4) Depth attenuation calculated with ocean volume stored in a deep shadow map.

References

SCHEEPERS, F., AND ANGELIDIS, A. 2008. Atmos: a system for building volume shaders. In *ACM SIGGRAPH 2008 talks*, ACM, SIGGRAPH '08, 64:1–64:1.

TESSENDORF, J., 2004. Simulating ocean surface. ACM SIGGRAPH 2004 Course, January.