Surf's Up Beach Break

Tom Kluyskens
Sony Imageworks
tkluyskens@imageworks.com

1. Introduction

When joining the *Surf's Up* team, we were asked do develop a solution for shore break – or **beach break** – for the 2 beaches featured in the movie. It soon became clear that the beach break would be visible in many shots (about 70). For that reason we envisioned a highly procedural setup, ideally allowing a TD to generate beach break for a shot in less than an hour. And this without overloading the Imageworks renderfarm in the process.

This sketch focuses on the 4 main parts of the *Surf's Up* beach break effect:

1. waves
2. 'licks' - the part of the waves that laps on shore
3. surface foam
4. whitewater

1. Waves

Wave formation can be boiled down to a few relatively simple mathematical equations based on the ocean floor profile. We implemented those rules in a *Houdini* setup.

A series of particle lines is being emitted out in the ocean, towards the beach - representing incoming waves. Every particle controls a curve profile that can take on any wave profile shape, depending on its position above the ocean floor, and the accumulated depth D and dD information. A series of profiles is skinned, resulting in a wave surface.

2. Licks

At some point in the wave's life cycle, a second, denser line of particles is emitted. These particles sense the beach slope and have specific rules based on simple fluid dynamics and animation curves to regulate the velocity up and back down the beach. In the same way as with the waves, each particle is associated with a curve profile, and these curve profiles are skinned to form the lick surface. From these surfaces, displacement maps are derived for in-shader displacement compositing.

3. Foam

In a separate setup, a foam pattern (convection) life cycle is simulated and rendered as a repeatable texture. The result is an image sequence representing a 3D shader: X and Z being U and V, and the sequence being the Y axis.

A hybrid *RenderMan* splatting shader takes "bites" out of this structure. The shader calls on attribute-carrying particles and rendered attribute maps to decide about placement, life cycle stage, deformation and fading of the original foam texture tile.

Particles are used for conveying attributes to the shader, but this technique is limited in its spatial precision. At leading edges of licks and waves, where more precise delineation is required, the shader switches to attribute texture map lookups.

Using such a shader results in relatively fast and light renders compared to using and rendering an actual particle/fluid sim. It uses a pre-rendered texture foam cycle, which delivers results that are beyond any procedural texturing method that tries to depict foam using noise functions etc.

4. Whitewater

All whitewater in the movie is done through a *RenderMan* clustering DSO, where many *RenderMan* points are added at rendertime based on an input of 'seed points'. The seed point simulation itself is a fairly conventional *Houdini* particle setup.

We wanted a fixed dataset for each beach, and so all whitewater was pre-simulated and stored on disk. To get the required detail from all angles, this resulted in about 200GB of particle data for each beach.

When a beach break shot comes up, an FX TD can set off a simple procedure that takes a chunk out of the full-beach whitewater sim, based on the camera's position. This involves camera-culling invisible parts away, and doing a smooth distance-based LOD cull of the master sim.

Conclusion

The result of the beach break effect setup is a fixed, loopable dataset of particles, texture maps and surfaces on disk, of about 300GB for each beach. This allows a beach break shot to be set up in about 10 minutes, and rendered very efficiently.

References

M. TOMCZAK 2005. Oceanography Lecture Notes. http://gyre.umeoce.maine.edu/physicalocean/Tomczak/IntroOc/lecture09.html