A Procedural Ocean Toolkit

Mårten Larsson* Digital Domain
Jens Zalzala† Digital Domain
Greg Duda‡ Digital Domain

1 Introduction

Several projects at Digital Domain have involved creating computer generated images of an ocean surface. The most successful approach used is a well tested technique based on statistical measurements of the ocean surface and an FFT based algorithm described in [Tessendorf2002].

For a recent large project the challenge was not to render just an ocean but to render an ocean with thousands of boats all creating foam trails and wakes that interact with each other as well as the ocean waves. On top of that, many shots needed a shoreline with waves crashing onto the beach. This sketch will describe the procedural system created for generating an ocean with wakes, foam trails and shoreline blending seamlessly with the FFT based ocean technique.

2 Wakes and foam trails

To generate the ocean surface a grid and two shaders are used (one for displacement and one for surface shading). The ocean is rendered as one continuous surface.

The entire wake system is driven by the position of the boats. Trailing splines are created automatically from the boat animations. The attributes for describing a whole spline are then stored, along with attributes for what type of boat it is and how its foam trail is shaped, in a point for each boat. These points are then fed to the shaders responsible for creating the ocean surface. At render time the spline is reconstructed in the renderer and a UV space is dynamically created that runs along the spline. Recreating the spline at render time solves two problems. The UV spaces can have an almost infinite precision even when the ocean is rendered with a very coarse grid. Since the ocean often extends all the way out to the horizon, it is important to try and keep the resolution of the grid down. The second problem this solves is that it allows the shader to handle an arbitrary number of overlapping UV spaces and because of that it can handle interaction of an arbitrary number of wakes. This is also a very important feature since the trails often overlaps when many boats are close to each other.

Simple maps with a wake pattern are then mapped on to the UV space and used to displace the surface before the ocean wave displacement is added. An attribute on the point decides what texture to use depending on what type of boat is making the trail. The foam trails are generated by applying procedural noise patterns on the same UV space. These maps and UV spaces are dragged around automatically by the boats.

To reduce render times a LOD system is used for the UV spaces. The curvature of the spline is used in the shader to automatically choose between a spline, multiple line segments or just a straight line as the basis for the UV space. This reduces render time since the spline can be expensive to evaluate at every shader sample.

*e-mail: mlarsson@d2.com
†e-mail: jzalzala@d2.com
‡e-mail: gduda@d2.com

3 Shoreline

The fully procedural shoreline setup relies on simple geometry since crashing waves are difficult to create with only displacement shaders and it is convenient for volumetric renders of foam and spray.

Figure 1: Fully CG ocean.

The shoreline waves are driven by a set of points distributed around each island or land-mass in the ocean. The points are placed using a Poisson distribution to ensure a certain distance between each wave. All points have an automatic animation moving them from their original position to a position on the beach. Wave geometry is instanced on each point. The waves automatically orient themselves towards the shore by using a projected normal from the beach geometry. Each wave geometry has a procedural animation of a crashing wave timed by using the distance to shore. This animation includes the wave flattening out on the beach and moving back into the ocean under the newer waves. Each geometry has a UV space controlling the foam it leaves behind. These wave geometries are then rendered with the regular ocean displacement and surface shader. This way the ocean blends seamlessly with the shoreline waves and even wakes from boats can mix with the crashing geometry waves.

4 Conclusion

Because it is procedural, this system has a very short setup time. Once the points for the wakes are generated with the click of a button and a Poisson distribution is generated automatically using the land geometry, the ocean system is ready to render the final image. It creates an all inclusive ocean render with all masks and complex displacement of the ocean scape in the same render.

References

TESSENDORF, J. 2002. Simulating Ocean Water. In *Course Notes #9 (Simulating Nature: Realistic and Interactive Techniques), SIGGRAPH 2002*, ACM Press / ACM SIGGRAPH