Rigging the Oceans of Disney's "Moana"

Jonathan Garcia Sara Drakeley Sean Palmer
Erin Ramos David Hutchins Ralf Habel Alexey Stomakhin

Walt Disney Animation Studios*

Figure 1: *Our ocean rigs allowed our artists to stage and art direct various ocean types including, but not limited to, the open seas (left), shorelines (center), and calm seas (right).* ©*2016 Disney.*

Abstract

Disney's "Moana" was set in an environment inspired by the Pacific Islands, which made the ocean a prominent setting throughout the film. For much of the film, we found it necessary to treat our oceans like we would our hero characters, and so we developed three ocean rigs that formed the basis of all our ocean variants. Using a unified workflow that our artists were already familiar with, these rigs gave them the ability to easily portray and dial in a wide variety of ocean types such as open seas, calm seas, stormy seas, lagoons, shorelines, and wide ocean vistas, some which spanned hundreds of kilometers out to the horizon. This paper describes the techniques we used and some of the challenges we faced in developing these ocean rigs.

Keywords: rigging, animation, oceans, Maya

Concepts: •**Computing methodologies** → **Computer graphics;** *Animation;*

1 Introduction

Our water pipeline in previous films was contained to one or two departments and had limited authoring capabilities. We basically had two types of water: procedural water and simulated water. For our procedurally generated bodies of water, the Environments team would create huge "ground planes" that represented the water surface which were then displaced at render time by our shaders using techniques inspired by Tessendorf Waves [Tessendorf 2004]. For our simulated bodies of water, the Effects team would set up simulation domains, figure out how to blend those back into the main (unsimulated) body of water, and then bake out the final water mesh for downstream consumption. Outside of that, no other department authored or modified bodies of water in our films.

*email:[First Name].[Last Name]@disneyanimation.com

SA '16 Technical Briefs , December 05 - 08, 2016, , Macao
ISBN: 978-1-4503-4541-5/16/12 $15.00
DOI: http://dx.doi.org/10.1145/3005358.3005379

For "Moana", we anticipated that over 1000 shots would contain some variant of the ocean which would include some sort of interaction with it (e.g., boat wakes, splashes, performance water, shorelines, etc.). Because of the sheer volume of such shots, we refined our pipeline to make the oceans first-class citizens so that they could be updated piecemeal throughout our pipeline. Here is a typical example of how our oceans were authored and updated in this new pipeline:

1. Using one of the ocean variants our Environments team created as a starting point, our Layout team refined the choppiness of the waves and the timing of the ocean's swells or the timing of the water lapping up and down the beaches.

2. Our Animation team finessed the timing of the ocean parameters that Layout had set to fit the needs of their character performances.

3. Our Effects team ran boat wake and splash simulations and blended them back into the ocean, as needed.

4. Finally, our Lighting team added in photon caustics, adjusted the ocean's volume shader parameters, and dialed in mesh tessellation and displacement parameters to tune the amount of detail for the oceans.

This new pipeline was developed around the idea that we would represent all of our oceans as level sets. The vast ocean expanse was defined by heightfields and simulation data was represented as signed distance functions. The blending of the simulations into the heightfields was described by a level set compositing graph, which is a set of nodes that read, blend and combine these inputs into a single surface representation in our renderer.

Using an implicit representation for the oceans gave us great flexibility. It allowed us to decouple the ocean mesh from its material shader. So, we did not have to decide a priori the amount of detail we needed to bake into the surface's geometry since the mesh was generated dynamically at render time. As such, our Effects team could dial in their desired level of detail for the ocean surfaces before running their simulations. Any detail that was filtered out of these surfaces could then be added back in by the displacement shader. To get this to work, we had to ensure that the ocean's heightfield expressions used by both the displacement shader and the level set composite graph were always in sync. We did this by storing the heightfield expression as a string attribute on a locator within the ocean element hierarchy so that it could then be referenced by both the displacement shader and the level set composite graph.

We also created a library of ocean variants from which the artist could choose, where each variant defined a different base heightfield expression and an associated base level set composite graph. By varying our heightfield expressions, we could easily represent different ocean surfaces procedurally (e.g., windy seas, calm seas, stormy seas, lagoons, etc.). By varying our level set composite graphs we could get simple oceans that were defined by a single heightfield, or we could get more complex oceans that, for example, blended a heightfield with a simulated shoreline and a few boat wake simulations. Mixing and matching different variations of heightfield expressions with different composite graphs gave us unlimited creative control over our oceans. With this, we now had a foundation for developing ocean rigs that encapsulated this flexibility while presenting a unified and efficient workflow for updating our oceans' characteristics.

2 Our Ocean Rigs

Our ocean rigs were developed leveraging off of two key concepts: representing the ocean as level sets and storing the ocean's heightfield expressions in a single attribute that can be referenced by various parts of our pipeline. Using those ideas as the foundation of our rig designs, we identified three different types of rigs that would allow us to portray the wide variety of ocean types required by our film's art direction: (1) an open sea rig, (2) a shoreline rig, and (3) a distant ocean rig.

Figure 2: *The flexibility of our heightfield expressions allowed us to blend multiple types of oceans into a single asset. The screenshot above shows how we could seamlessly blend the lagoon (the lighter blue water) and the open sea (the darker blue water) regions of the ocean using a single expression. Our ocean rig for this variant exposed controls that allowed each region's parameters to be adjusted independently from the other. ©2016 Disney.*

2.1 Open Sea Rigs

2.1.1 Editing Heightfields

It was undesirable to have artists directly edit the ocean expressions. We wanted to limit the ability to hand edit these expressions to minimize issues in trying to keep the one used by the level set composite graph in sync with the one used by the displacement shader. These expressions were also relatively complex and we wanted to reduce the chance of typos, syntax errors, and deviation from the base expression defined by our Environments team. Furthermore, not all artists were comfortable editing SeExpr, our in-house open source embedded expression language. As a result, we developed a custom Maya node that was effectively an expression builder (Fig. 3).

This custom node was the foundation of our ocean rigs and greatly simplified the workflow for editing the heightfield expressions. Our Environments team created the ocean's base heightfield expression and chose default values for most of the expression's parameters, hitting the broad strokes of the desired art direction. By exposing

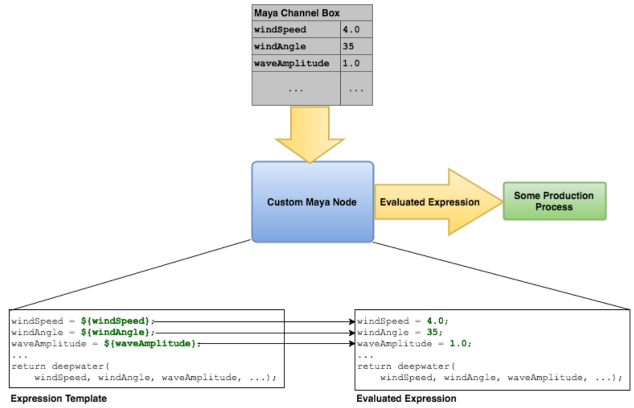

Figure 3: *By templatizing our expressions and using our custom Maya node, artists could adjust the heightfield expression parameters via Maya's channel box. This presented them with a familiar workflow, reducing the learning curve for working with the oceans.*

the rest of the parameters in the rig, our Layout artists could fine-tune the ocean to fit their chosen composition and scene layout. Making it a custom Maya node also let us easily build the ocean rig in such a way that its controls looked and felt like one of our typical character rigs.

This mechanism of editing the expression in conjunction with the way we organized our ocean variants provided us with a couple of benefits:

- It abstracted away a lot of the complexities of our ocean assets, reducing the learning curve to working with our oceans. Updating Maya channels is already a familiar process for our artists, and so this simplified and unified the workflow for updating the expressions used by the level set composite graphs and displacement shaders.
- The rig became the single entry point for editing the heightfield expressions, minimizing issues of syncing up the expression across the various parts of our pipeline.

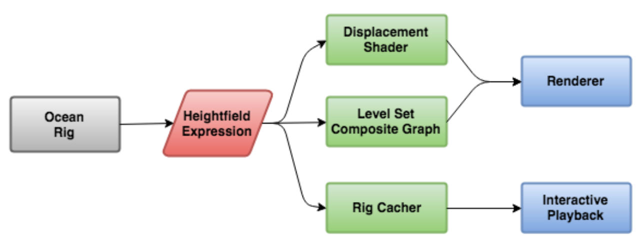

Figure 4: *Our ocean rig was the primary mechanism for updating the heightfield expressions, giving the various parts of our water pipeline a centralized data store from which to pull the heightfield expession.*

2.1.2 Interactive Playback

Even though our open seas were defined by heightfields that extended out to the horizon, our artists typically only needed to preview regions of the ocean that were close to where the action was taking place. So, we provided them with a GL preview of a cropped region of the ocean surface in Maya's viewport. We also added controls in the rig that allowed the artist to move that proxy surface around in space, giving them a "preview window" into the ocean at any point in space.

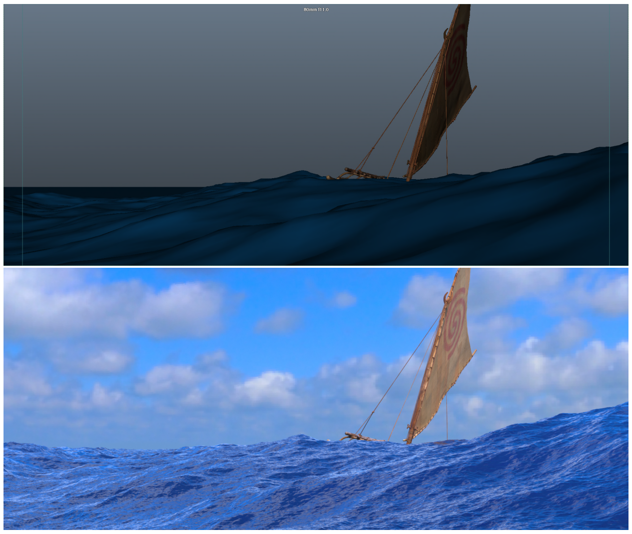

Figure 5: *Not only did our open sea rigs provide real-time playback (top), they were render-ready out of the box (bottom). ©2016 Disney.*

Our initial implementation of this preview surface was too slow for interactive use. Our heightfield expressions were complex enough that we were initially getting between 0.2 and 0.5 fps. To work around this bottleneck we utilized our in-house rig caching technology that uses background Maya processes to cache deformed meshes based on the current state of the rig. Integrating the oceans into this caching technology posed its own set of challenges. The main challenge was that this system is designed for caching meshes that are deformed kinematically. Since the proxy ocean surfaces were deformed implicitly across time, we had to make the ocean's "time" parameter a kinematic control (instead of an implicit one) so that the proxy surfaces would cache properly. After integrating this caching technology, we were able to get faster than real-time playback (24+ fps) after all of the desired playback frames had been cached. In simple scenes with just the ocean and a boat, we could easily get 80+ fps.

2.2 Shoreline Rigs

2.2.1 Editing Level Set Composite Graphs

Because our shorelines were pre-simulated against specific beaches and ocean wave characteristics, it was important that none of the shape properties defined by the heightfield expression changed. So for our shoreline rigs, the only control we wanted to give the artists was over the timing of the water lapping on the shores. With the custom Maya node we built for our open sea rigs, we already had a mechanism for updating the time offsets of the heightfield expression (by having the rig expose the time offset parameter). What was missing was a way to offset the time offsets of the simulated shoreline data within the level set composite graphs. So in the same vein as our custom Maya node for editing heightfield expressions, we created a custom Maya node for editing our level set composite graphs. We stored our composite graphs as JSON, so under the hood, this node was effectively a JSON editor.

2.2.2 Interactive Playback

In shots containing a shoreline, visualizing the water lapping on the shores was more important than previewing the heightfield that

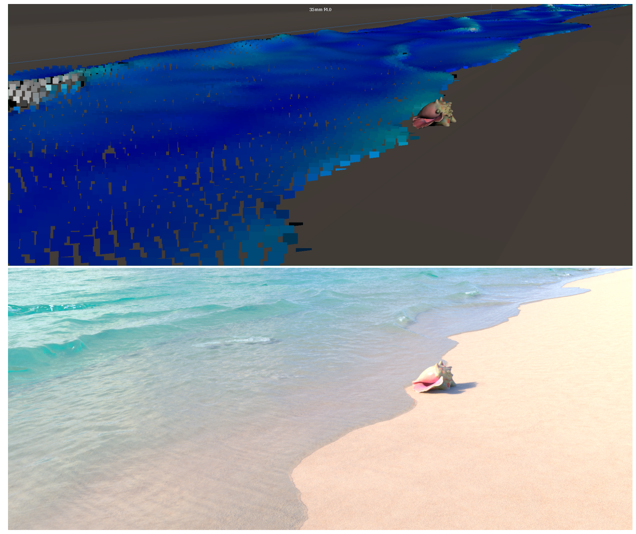

Figure 6: *We used points to visualize our shorelines during interactive Maya sessions (top). This representation yielded real-time playback and was good enough to compose our scenes. Just like the open sea rigs, our shoreline rigs were render-ready out of the box (bottom). ©2016 Disney.*

defined the rest of the ocean. Because of this, we got rid of the proxy surface that the open sea rigs used, and replaced it with a lightweight version of the fully simulated shoreline.

We opted to visualize the shoreline as a relatively sparse set of points, instead of as a low resolution mesh. The main reason we chose this representation was because we already had the infrastructure built for it. Utilizing our proprietary particle instancer to display these points in Maya, we were able to give our artists a complete representation of our shoreline simulations while maintaining faster than real-time playback. To sync the time offsets of these proxy points, we already had a custom Maya node to hook up Maya channels into our particle instancer, and so we just incorporated that into the rig, as well.

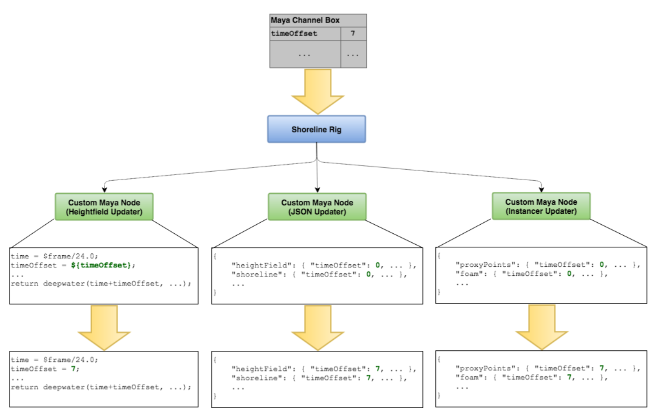

Figure 7: *The rigs simplified how different aspects of the ocean were updated by associating each rig control with multiple systems. For example, the shoreline rigs had a single channel for adjusting time offsets, but that value got injected into 3 different parts of the rig: the heightfield expression, the level set composite graph, and the instancer.*

2.3 Distant Ocean Rigs

Certain moments in "Moana", such as vast high-altitude vista views of the ocean, required so much art direction that it no longer made sense to try to come up with a base heightfield expression from which to start. For these shots, interaction was not necessary and so there was no need to share the heightfield expression between the mesh and the displacement shader. Since it was faster to iterate on displacement for these types of shots, we created a much simpler rig and dubbed these ocean variants as displacement-only rigs. For these rigs, we exposed the heightfield expression in the displacement shader to give artists complete control over it (i.e., it no longer referenced a string attribute stored on the locator, and was directly editable in the shader). Although very similar to the way we did water in our previous films, for consistency, we still represented the ocean surface as a heightfield in these rigs instead of as baked out geometry. We just ended up hardcoding the heightfield expression to yield a zero-height surface (i.e., a flat plane) on which to apply the displacement.

This variant was mainly used in shots where the ocean was in the background (i.e., no characters or boats or other objects were interacting with it) or where we were seeing wide ocean vistas. Even though our open sea and shoreline rig variants got us the majority of our oceans, it was nice to have this variant in our back pocket just in case we needed to get something out quickly.

Figure 8: *An example of where we used our displacement-only oceans to give artists the freedom to art direct the wide ocean vistas (this vantage point is from about 330 meters above sea level).* *©2016 Disney.*

3 Conclusion

Our ocean rigs successfully gave our artists the necessary control to create realistic yet art-directable oceans. They streamlined the process of art directing and finessing the timing of our oceans. With the ocean rigs resembling our character rigs, they also presented our artists with a very familiar workflow. This resemblance lowered the barrier to learning how to use them, allowing artists to iterate faster and focus on their craft rather than learning a new technology.

Even though we successfully used these rigs throughout "Moana", they were not perfect. The following issues give us insight on how to improve our ocean rigs for future projects:

- **Rig Updates:** We did our best to predict the needs of production, but as our ocean rigs moved down our pipeline, feature requests would be called out. Adding these feature requests usually required rebuilds of our rigs. Because we employ a "push" paradigm to distribute assets downstream, these rig updates could be disruptive to shots that were already far along down the pipe.

- **No T-Pose:** The default settings for our character rigs put them into a T-Pose (i.e., a pose where the character is standing at the origin with its arms extended out on each side). This is a clear visual indicator that the character is in its default state. Unfortunately, we had no such indicators with our ocean rigs, sometimes making it difficult to know if the correct ocean settings were being used for a given shot.

  If there was any discrepancy between our boat animation and the oceans, it was very difficult to determine which was wrong: the ocean or the boats. A lot of these discrepancies arose because the oceans somehow got reverted to their default rig settings. Sometimes the defaults were close to what the boats were animated on, but they also varied enough to notice the problem. Having a visual indication of the default state would have helped us more easily identify these situations.

- **Syncing Wet Maps:** Our shoreline rigs only represented the water lapping on the beach. The actual beach geometry was broken out on its own. The way we organize our assets and make them completely separate from each other posed a challenge in ensuring that the time offsets of the shoreline rig matched that of the wet maps of the beaches. We came up with a post-processing step to shore up these time offsets, but it was another thing that our artists had to remember to do.

  We considered giving the artists a workflow for updating a "global time offset" through a single interface which would update the time offsets of any asset listening for it. However, the required changes to our pipeline were too cost-prohibitive. Even though we still think this "global time offset workflow" is a viable solution for future productions, we ended up deciding that we could make do without it for "Moana".

- **Limiting Parameter Exposure:** We initially made a best guess as to which ocean parameters needed to be exposed in our rigs, and hid the rest of them. We did this in an attempt to simplify artist interaction with the ocean rigs. Later, we found out that this just made it more difficult to fine tune certain aspects of the oceans as the artists' need to make more nuanced adjustments grew. To make these updates, we would bypass the rig controls and edit the heightfield expression directly by updating the locator attribute. We had to resort to these brute force updates when we reached a point in production where we had to lock our rigs from future updates to avoid the risk of affecting shots that were already far along in our pipeline.

References

HINSINGER, D., NEYRET, F., AND CANI, M.-P. 2002. Interactive animation of ocean waves. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '02, 161–166.

HORVATH, C. J. 2015. Empirical directional wave spectra for computer graphics. In *Proceedings of the 2015 Symposium on Digital Production*, ACM, New York, NY, USA, DigiPro '15, 29–39.

MILLER, E., BREDOW, R., KRAMER, D., HAUSMAN, M., SHINNERS, P., CARLSON, D., AND CLARK, J. 2007. Making waves for surf's up. In *ACM SIGGRAPH 2007 Sketches*, ACM, New York, NY, USA, SIGGRAPH '07.

TESSENDORF, J. 2004. Simulating ocean surfaces. In *ACM SIGGRAPH 2004 Course Notes*, ACM, New York, NY, USA, SIGGRAPH '04.

WALT DISNEY ANIMATION STUDIOS. SeExpr. http://www.disneyanimation.com/technology/seexpr.html.