PROCEDURAL MODELLING AND ANIMATION OF BREAKING WAVES

MASTERS THESIS

FINELLA FAN

N.C.C.A BOURNEMOUTH UNIVERSITY

21 August, 2009

List of Contents

List of Figures

Chapter 1

Introduction

Breaking waves, especially plunging breakers, are some of the most spectacular phenomena that nature possesses and have long been one of the most fascinating subjects for painters, photographers, surfers and film makers.

The modelling and animation of natural phenomena such as breaking waves has always been a challenge in computer graphics as they appear in a variety of different forms and are difficult to control. This thesis discusses the process of generating a specific shot for breaking waves, plunging waves in particular, and at the same time building a Houdini Digital Asset for creating such shots.

The aim of this project was to develop a tool that could focus on creating shots for breaking waves. Side Effects Houdini was used throughout the pipeline from modelling to rendering. Nuke was used for the final compositing of the shot. The tool was designed to provide the user with enough control over the wave's general look which includes the shape of the wave, shading of the wave surface and whitewater. Lighting, rendering and compositing were not included in the tool but are discussed in the thesis as a reference on how to create a complete shot of breaking waves.

Considering that there is usually very limited time during production, the option of using fluid simulation for the waves was ruled out from the beginning as this project planned to use as less simulation as possible. The techniques used in this project were mostly based on the techniques used in the film Surf's Up (see the Previous Work section for more detail) with the idea of using blendshapes and wave rigs to provide the user with a controllable tool for generating breaking waves.

Chapter 2

Previous Work

2.1 Ocean Waves

In 1986, Fournier and Reeves [1] presented a model for the ocean surface based on sinusoids and described the water surface as orbital motions of individual points on the water surface as a wave passes by. The model was able to generate waves coming up a shoreline and could also detect the position, direction, and speed of spilling breakers in order to generate spray and foam using particle systems. In the same year, Darwyn Peachey [2] presented a similar approach that used basic shapes other than sinusoids.

Jerry Tessendorf [3] presented a method using height fields for simulating ocean water based on statistical models and the latest work was given in the SIGGRAPH 2004 course notes. The statistical wave models consider the wave height as a random variable of horizontal position and time. Statistical models are also based on the ability to decompose the wave height field as a sum of sine and cosine waves using Fast Fourier Transforms (ffts), which are a rapid method of evaluating the sums.

Since the year 2005, the Houdini Ocean Toolkit (HOT) was provided by Drew Whitehouse [4] as an OTL for rendering deep ocean waves based on the algorithms of Jerry Tessendorf's SIGGRAPH 2004 course notes. The tool uses Houdini's SOPs, VEX functions and VOPs for displacing and shading the ocean surface geometry.

Although the above models were capable of producing fairly realistic-looking ocean waves for relatively clam ocean surfaces which are usually referred to as deep water waves, plunging breaking waves could not be modelled correctly with a parametric surface based on sinusoidal assumption or using height fields to represent ocean surfaces. This project did not apply the above techniques but they provided an insight into generating ocean waves in computer graphics and served as a useful starting point for the project.

2.2 Breaking Waves

Jeschke, Birkholz and Schmann [5] presented a procedural model for generating breaking ocean waves that could be used for interactive visualisation. The procedural model described every point in the ocean surface using a mixture of cosine, exponential, rotation and scaling functions in dependence of time and space. Every point in the scene, even foam generated by the breaking waves could always be calculated without using information from previous time steps. The animation of the ocean waves is restricted to a continuously changing parametrisation of the functions.

Mihalef, Metaxas and Sussman [6] presented a fluid animation control method to control breaking waves. In their Slice Method framework, a library of 2D breaking waves is provided for the user to define the vertical 2D shape of a breaking wave at a desired moment in its evolution, the system then computes the subsequent dynamics with the aid of a 3D Navier-Stokes solver to integrate the 2D simulations into a 3D shape.

In 2007, the same year as its film *Surf's UP* [16] was released, the team at Sony Pictures Imageworks presented a course at SIGGRAPH on how the film was made in general and how they created the breaking ocean waves which were also treated as main characters in the film. Blendshapes of several wave profile curves were used for the creation of different phases in a breaking wave and ring-like wave rigs were provided so that the animators could easily control the overall shape and movement of the waves. Particle systems were used to generate whitewater. The tools used in production were Alias Maya, Side Effects Houdini and Pixar's Renderman. [7]

In 2008, a three week project was carried out at Side Effects to produce a scene where waves crashing onto a lighthouse. For the modelling of the wave surface, the scene used shelf tools such as Waveform and Sculpted Particle Fluid tools. The Whitecaps tool was used to add whitewater and the overall level of detail to the waves. The lighthouse team also developed a custom material for the appearance of the water surface and a custom CHOP network to automate the splashes. [8]

A commercial called *Milk* by Psyop [9] came out in 2009. There were a few shots of breaking waves in the commercial. One of the 3D artists at Psyop, Florian Witzel, provided the making of *Milk* on his website [10] and also gave a talk regarding the same topic [11]. Though different in the overall style and tools / user interface provided, the concepts behind generating breaking waves in Milk were similar to that of *Surf's Up* which used wave rigs.

Most of the techniques used in this project were based on the concepts presented in the *Surf's Up* course notes.

Chapter 3

Technical Background

3.1 Fundamentals of Breaking Waves

3.1.1 How Breaking Waves Work

Water particles on the ocean surface travel in an orbital motion. When a wave crest approaches, the water particles rise and move forward. At the position below the crest, the water particles stop rising and are moving forward at the speed of the crest. As the trough advances, the particles slow their falling rate and start moving backward, until at the bottom of the trough they move only backward. As a deep-water wave moves into shallow water, the water particle orbits become flattened circles or ellipses. The orbits are compressed and the forward speed of the wave is reduced due to the interaction with the bottom.

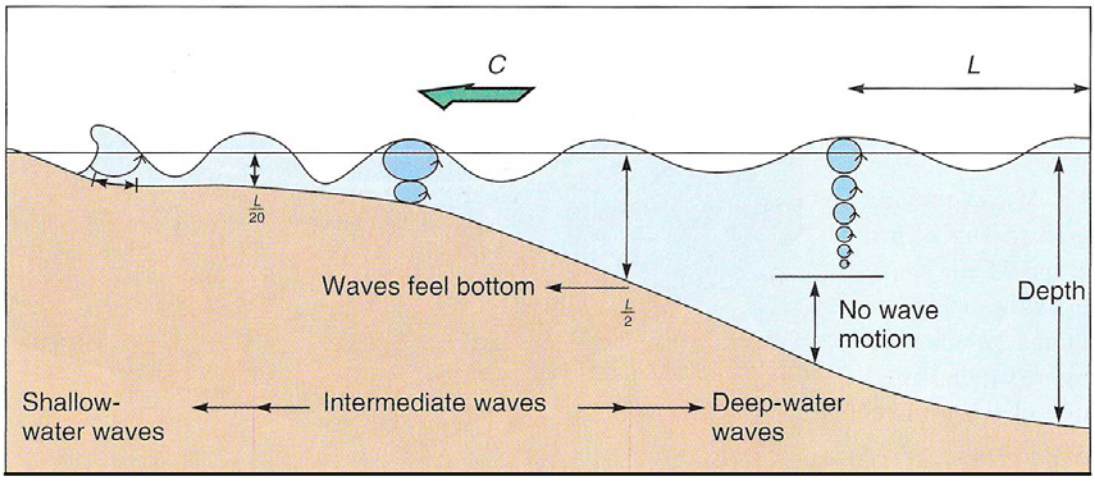


Figure 3.1: Deep-water waves become intermediate waves and then shallow-water waves as depth decreases and wave motion interacts with the sea floor. [12]

As a series of waves approach shore, trailing waves catch up on leading waves, because the trailing waves are in deeper water and traveling faster, while the leading waves are in the shallowest water and going the slowest. Wavelengths decrease and wave heights increase. Taller waves mean larger orbitals and faster orbital speeds for the water. Orbital speeds increase and wave speeds decrease. Eventually the orbital speed of the water is greater than that of the wave itself. The crest gets out ahead of its support, and the wave breaks.

For any given wavelength, there is a maximum possible wave height and it is measured by the steepness of the wave. If the ratio of the height over the length approaches 1:7, the wave's crest angle approaches 120^o and the wave becomes unstable and breaks. Then it is replaced by another wave produced by the rising wind. [12] [13] [14]

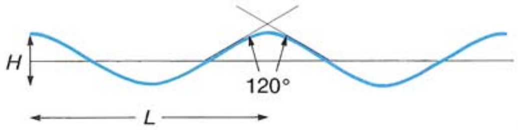


Figure 3.2: Wave steepness. When wave height over wavelength approaches 1:7, the wave's crest angle approaches 120 and the wave breaks. [12]

3.1.2 Types of Breaking Waves

- **Spilling Breakers**
  Spilling breakers usually occur where there is only a gently sloping shoreline. Spilling breakers start breaking slowly some distance from shore and are caused when a layer of water at the crest moves forward faster than the wave as a whole and only the top part of the wave curls over. Foam forms at the crest and eventually covers the leading face of the breaker.

- **Plunging Breakers**
  When the crest travels faster than the base of the breaker, it curls over and forms a tunnel, with a convex back and a concave front, and then falls downward violently toward the wave trough, the energy is dissipated over a short distance. Plunging breakers appear when there is greater slope to the bottom and are usually associated with the long swells generated by distant storms. The energy dissipated by plunging breakers is concentrated at the plunge point (i.e., where the water hits the bed) and can have great erosive effect. Plunging breaking waves are the most spectacular type of breakers and are much beloved by surfers.

- **Collapsing Breakers**
  Collapsing breakers are similar to plunging breakers, except that the front face of the waves may be less steep and instead of the crest curling over, the wave

collapses in the middle or near the bottom of the wave instead of at the top. Such breakers represent a transition form from plunging to surging breakers and usually happen on beaches with rather steep slopes, and under moderate wind conditions.

- **Surging Breakers**
  Surging breakers are formed on extremely steep shores. Surging breakers are typically formed from long, low waves. Thus, the front faces and crests remain relatively unbroken as the waves roll onto the beach in a brief surge of water. [12] [13] [14]

This project focused on the generation of plunging breakers only.

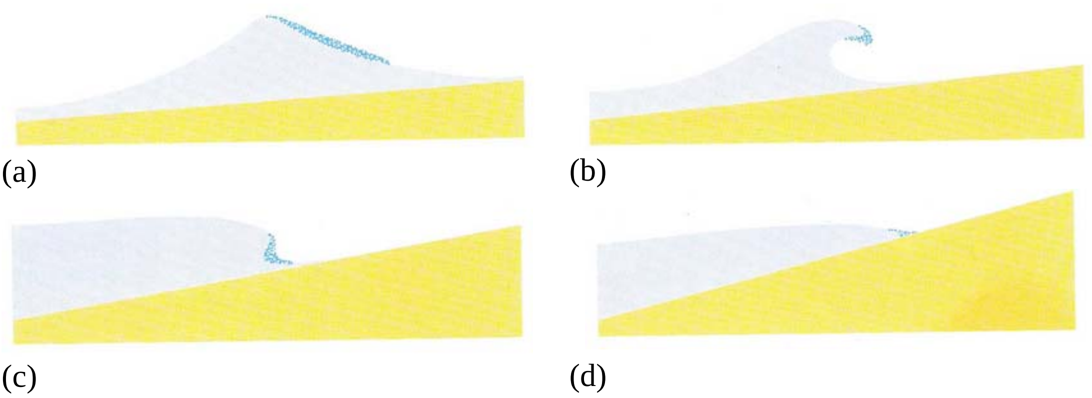

(a) (b)
(c) (d)

Figure 3.3: Four types of breaking waves. (a) spilling breaker. (b) plunging breaker. (c) collapsing breaker. (d) surging breaker. [14]

3.2 Side Effects Houdini

Side Effects Houdini was the chosen tool for this project because of its nature of proceduralism. The node based network treats each operation as a black box of utility that provides the user with a wide range of control and flexibility over the production pipeline. The user does not need to know what is contained inside each node as long as the user knows what is happening to the node's inputs and outputs. Furthermore, Houdini is widely used in the visual effects film industry for creating effects shots.

For this project, the most used contexts in Houdini are briefly explained as follow:

- **SOP (Geometry Surface Operators)**
  SOPs are always contained within an Object level node. SOPs are responsible for creating and modifying the geometry. This project used SOPs for creating the surface geometry of the breaking waves.

- **SHOP (Shader Operators)**
  The Shaders context is where the nodes that control the surface appearance of the geometry objects are defined. There is a Material Palette in Houdini that provides several pre-defined materials. Each material contains one or more SHOPs. The SHOPs used in this project were based on the materials in the Material Palette with some modifications.

- **POP (Particle Operators)**
  POPs are used to generate particles. Particles are useful for modelling realistic natural phenomena such as smoke, water, fire and dust. This project used POPs for the whitewater and SOPs were used as birthing sources and for passing attributes to the particles.

- **ROP (Outputs Rendering Operators)**
  The Outputs context is where to control how images are rendered. This project applied Houdini's production quality renderer called Mantra for the rendering of the breaking wave shot.

- **HDA (Houdini Digital Asset)**

  Houdini Digital Assets encapsulate a series of operations into a single node that is portable between different files or networks. The assets usually provide the users with easy-to-use interfaces so that the user does not need to worry about what kind of complex process is contained within that node. Some of the benefits of using a digital asset are:

  - An HDA is easy to use and easy to create.
  - An HDA is reusable in different files or networks in the same file.
  - An HDA can help new users in learning how Houdini works.
  - An HDA can facilitate production changes and the user can also change the interface and content to meet his / her to needs.

In addition to creating a shot for breaking waves, this project also wrapped up the nodes and networks used in the pipeline and created a digital asset that could be reused or extend for similar shots of breaking waves in the future. [15] [21]

3.3 Nuke

The Foundry’s Nuke is a powerful compositing tool with a variety of element resolutions and colour spaces, provided all within a floating-point 2D and 3D environment. Nuke was used for compositing the final shot of the breaking wave.

Chapter 4

Implementation

The aim of this project was to develop a Houdini tool that could generate shots specifically for breaking waves. The project started out by creating a breaking wave shot and as the system developed, the tools used for the shot was then turned into a digital asset. Rather than being physically-based, the breaking wave geometry in this project was modeled by hand with procedural animation controls provided.

4.1 Pre-production

In addition to the research that was mentioned in the Previous Work section, collecting reference for breaking waves was also carried out in the pre-production process. Most of the reference images and videos came from surfing as those reference photos and videos capture the wave perfectly in almost every angle.

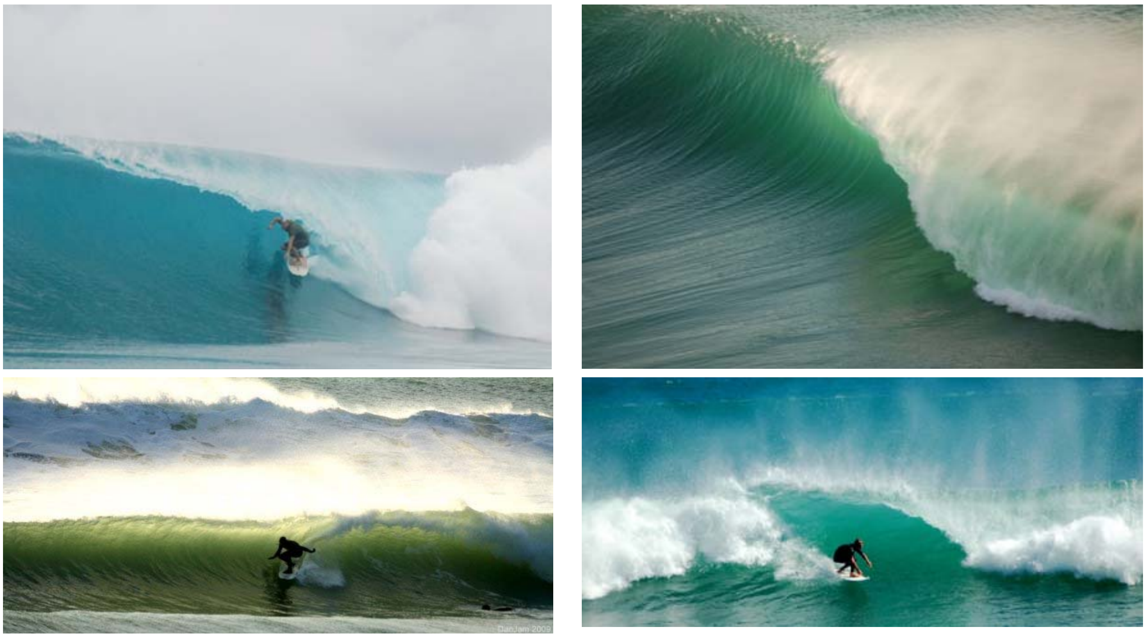


Figure 4.1: Reference breaking wave images from [19]

Surfing videos on the internet as well as in films such as *Surf's Up* [16] , *Riding Giants* [17] and *Blue Crash* [18] were closely watched and studied to learn more about the general appearance and movement of breaking waves.

A test render at a very early stage of the production was performed and the rendered image was then modified in Photoshop with colour correction and added mists and splashes. This image was used as a reference throughout the project.

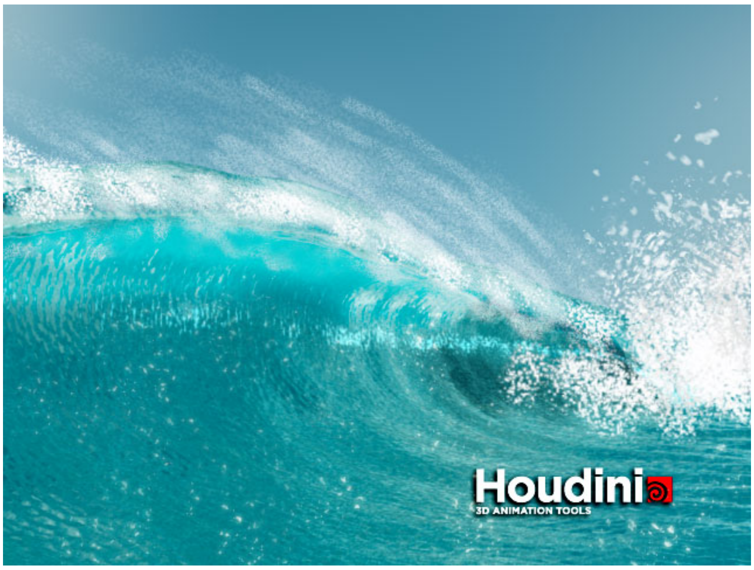


Figure 4.2: Early wave surface render with colour correction and added whitewater in Photoshop

4.2 Modelling

4.2.1 Wave Geometry

The modelling of the breaking waves started out as a single NURBS curve. Everything that came later in the production pipeline was based on this single curve. This curve was then modelled by hand into different shapes representing different phases in a breaking wave life-cycle. Profile curves provided by the Surf's Up course note and real world breaking wave pictures were used as reference for modelling the different shapes in a breaking wave life-cycle. The benefit of using a simple NURBS curve for modelling the breaking wave was that only a few CV points needed to be tweaked to achieve the desired look without the need to modify a complex geometry at this stage.

One of the most challenging part of this project was getting the shape of this profile curve looked correct and mimicked a breaking wave in reality. Not only was the shape of the breaking wave important, the position of each CV point and how they move along the geometry also played a major role in the whole system. This single profile curve affects the animation, shading and whitewater system that comes later in the pipeline.

A total of eleven profile curves were created and acted as blendshape targets for a breaking wave's full life-cycle. A Sequence Blend SOP node was used for blending the shapes between those eleven profile curves. Once the blendshape profile curve was generated, it was copied ten times and placed in a row along the positive x direction. Those blendshape profile curves were then lofted using a Skin SOP node to form the wave surface. Before any animation control was added, the wave surface was only a flat grid.

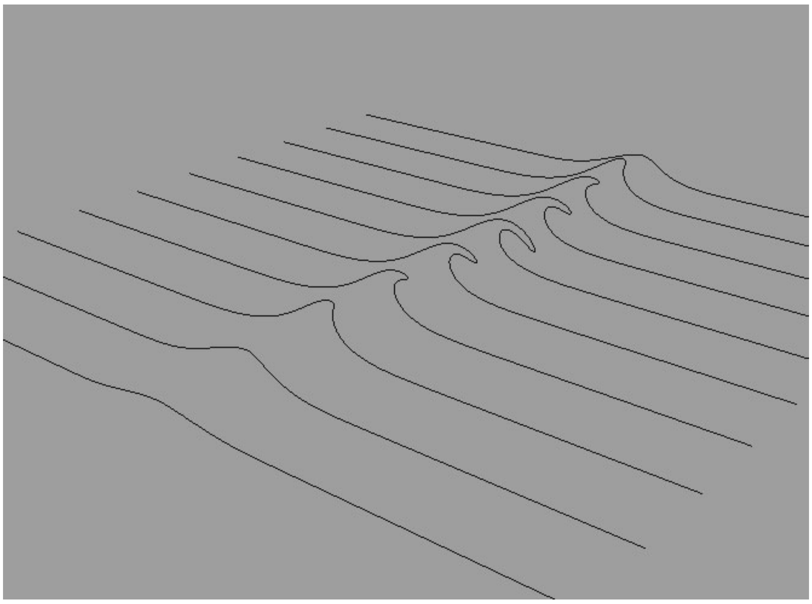


Figure 4.3: Profile curves that were used as blendshape targets.

4.2.2 Wave Trains

Wave trains are packets of similar waves that have approximately the same pattern and speed. Wave trains in [3] generated wave height fields with noise patterns. This project employed that concept. A Mountain SOP node was used to displace the wave surface. The mountain SOP node can generate a variety of noise patterns and is useful for creation of fractal surfaces such as mountains, desserts and water surfaces. It provided more details and worked well with the wave surface in this project.

4.3 Animation

To provide the user with a more flexible control over the movement of the breaking wave, two types of animation controls were designed. The user could choose from either an automatic animation control or a custom animation control. Both of those controls provided parameters for the user to adjust the breaking and forward motion of the breaking wave and set the keyframes.

4.3.1 Custom Animation Control

The animation for the breaking wave shot in this project initially started out with the custom animation control which could be used to build a custom breaking wave by controlling how each cross-section profile curves move individually. The custom animation control allowed the user to customise the wave movement by controlling and setting keyframes on parameters that were associated with each blendshape profile curves as well as the transformation for forward motion. Any kind of arbitrary breaking wave movement could be achieved by tweaking those parameters

4.3.2 Auto Animation Control

An animation control that could automate the breaking of the wave was designed later in the production. By setting keyframes for only five parameters, the user could get a fairly realistic movement of a breaking wave without the need to set keyframe for each and every cross-section profile curves (ten curves in this case). The breaking wave shot that were produced in this project started out with the custom control and ended up using the auto control for the final shot.

The five parameters that contributed to the auto animation controls are explained as follow:

- Center: lets the user specify at which point along the cross-sections that the wave starts to break. This breaking point can be exactly on each cross-section profile curve or somewhere between those curves.
- Span: controls how wide the wave spans.
- Falloff: adds non-linear variation to the wave shape. When the Falloff is greater than 1, it has an ease out effect on the breaking wave. If the Falloff is less than 1, the life-cycle of the breaking wave tends to be shorter.

- Time: this parameter controls the evolution of a breaking wave. 0 means at the beginning of the breaking wave's life-cycle. As the value increases, the wave shape evolves and deforms out of the centre line of the surface. Time = 1 means that the wave reaches the very end of its life-cycle.
- Speed: determines how fast the wave moves forward.

4.4 Whitewater

Houdini's particle systems were used to create the whitewater effects which include mist that comes off the lip of the wave and splashes that happen when the tip of the wave crest crashes into the trough. Both the mist and splashes were important and indispensable elements in creating the shot for breaking waves. A general pipeline used for the whitewater in this project was using SOPs as the source for the birth of the particles, some attributes from SOPs might be passed to some of the POPs for adjusting the behaviour of the particles. If the simulation of the particles was acceptable and satisfying, the particles were then cached out to disk as bgeo files using a ROP Output Driver node. Those bgeo files could then be read back into the scene without having to simulate the whole sequence again and facilitated the rendering process that came later.

4.4.1 Mist

The curvature of the surface around the lip area was measured using a Measure SOP node. The curvature attribute information gathered was then used to find out when the breaking waves would start emitting mist particles and only when the curvature of the surface was steep enough that the mist would be emitted from the wave surface.

The normal of each point on the wave's back face surface was adjusted so that the direction of each normal would follow the tangent of the wave surface. This tangent direction determines in which way the mist would travel. In the POP Network for generating the mist particles, an Attractor POP node was used for this purpose which was to mimic the vortices / turbulence cased by the wave in the real world. The normal information was fed into the Attractor POP node and the direction of the normal acted as a magnetic force for the direction on which the particles would travel.

The number of particles emitted for the mists depends on the number of points that was scattered using a Scatter SOP node on the wave surface around the lip area. Those scattered points were also the source for birthing particles. The mist particles used inherited velocity with a bit of variation on all of the x, y, z direction so that the mist would not look too uniform. Wind turbulence was added using a Wind POP node to give the mists more diversity with some randomised noise pattern.

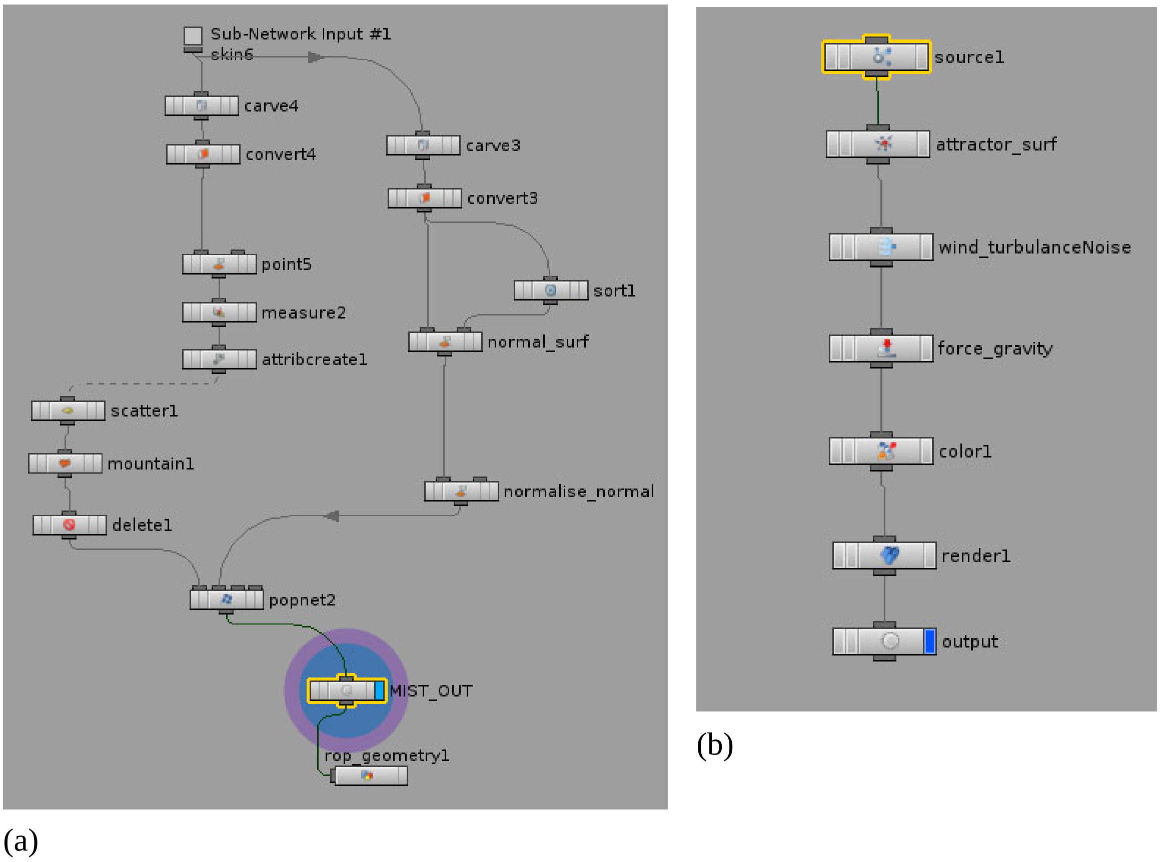
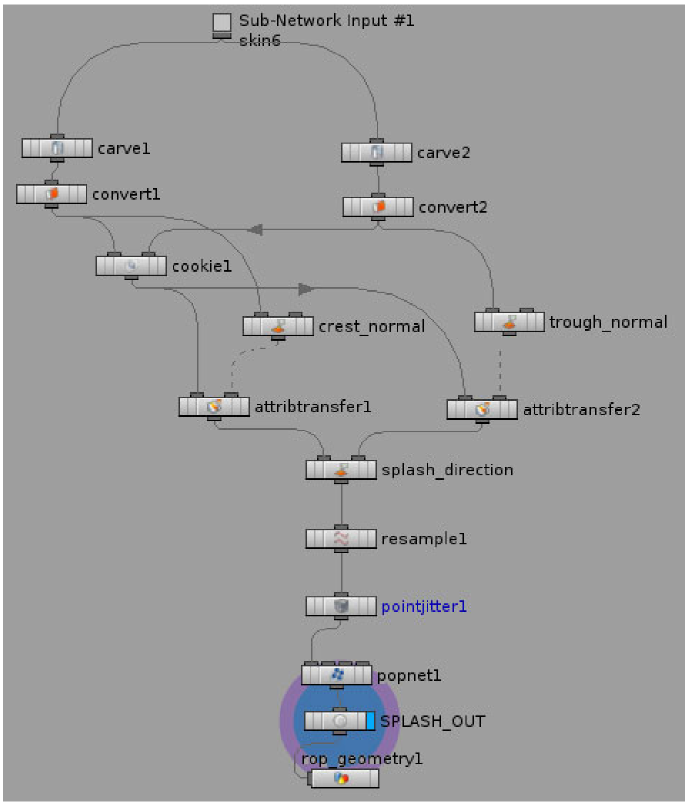


Figure 4.4: (a) SOPs for generating mist. (b) POP Network for mist.

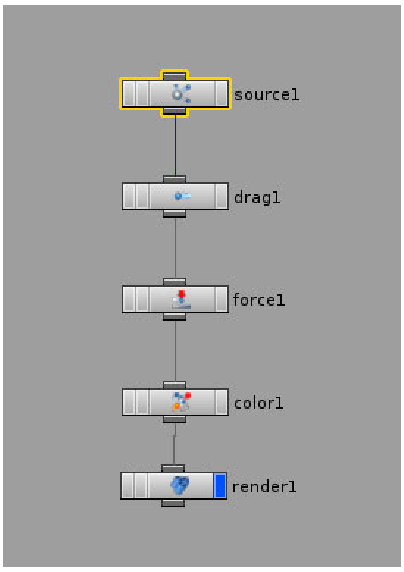
4.4.2 Splashes

In order to find out where and when the splashes were to occur, the intersection area of the crest and the trough was first determined using a Cookie SOP node. The Cookie SOP node was able to compute and extract a polygonal contour line around the intersection area. Then the surface normal of the crest surface and the trough surface were transferred to the contour line separately. Both of those normals were used to determine the direction of the splashes. If more weight was assigned to the crest's normal, then the splashes would travel more toward the vertical direction. If more weight was assigned to the trough's normal, the splashes tend to travel more along the horizontal direction. The points on the contour line were resampled with a Resample SOP node to increase the level of detail and the position of each point was randomised using a Pointjitter SOP node. Those points were used as the source for emitting the splash particles.

A POP Network was fed into the network for creating the actual splashes. The particles emission rate per frame were clamped according to the total points on the contour line since the number of points on the contour line would change as the intersection area of the crest and trough changed every frame as long as there was animation involved. The Cookie SOP had to recalculate the contour line whenever the intersection area changed. The more points on the contour line, the more particles were emitted and vice versa. The splash particles used inherited velocity from the points on the contour line with a bit of variance on all of the x, y, z direction. The y direction had the most variation when creating the braking wave shot as the splashes usually spread dramatically along the vertical direction for breaking waves.



(a)



(b)

Figure 4.5: (a) SOPs for generating splashes. (b) POP Network for splashes.

4.5 Shading

The shading process for the wave surface started out by using default materials contained in Houdini's Material Palette such as Ice Cube and Frosted Glass that theoretically possess similar characteristics as the ocean wave surface. Several other materials such as Basic Liquid and Water Surface that existed within the VOP VEX Surface SHOP network were also tested. The final shader ended up as a modification of the Frosted Glass shader and the Noise Displacement shader.

4.5.1 Surface Shader

The Frosted Glass shader not only worked well with glass, it also comprised the necessary attributes such as reflection and refraction for creating a shader for the ocean surface. Modifications were made to make the ocean water surface look more realistic. The index of refraction was changed to 1.333 to match that of water / liquids at a temperature of around 20 degrees. A photograph of the sky with white clouds was taken and was used as the environment map for the shader. The wave surface would reflect the environment map of the sky and add more plausibility to the wave surface.

Inside the VOP / VEX Network for the Frosted Glass shader, a Simple Subsurface Scattering VEX node with a light green colour was added to the output colour. Subsurface scattering (SSS) is a characteristic of translucent objects that light is scattered by interacting with the object's material when it attempts to penetrate the surface of those objects. Materials such as ocean waves, skin, milk, wax, ice…etc., all have this SSS feature. [20]

An additional subnetwork was added in the Frosted Glass shader in order to render out grayscale images of the normal information. The black and white images of the normal were used later in the project for compositing purposes. The grayscale normal information was used to enhance the effect of subsurface scattering inside the tube of the wave and at the lip of the breaking wave.

The result created with this modified surface shader was fairly satisfying.

4.5.2 Displacement Shader

Besides the wave trains that were created during the modelling stage, the wave surface also needed to have smaller displacements that would affect reflection, refraction and the movement of the wave surface. Those displacements made a big difference to the overall look of the surface.

A Vex Choppy Water SHOP node was first used for testing the displacement. Later, a Noise Displacement shader from the Material Palette provided by Houdini was modified and used for the final shot. The wave surface was divided into three different parts according to its colour attribute that were painted by hand in SOPs. Red represents the area around the lip. Blue was used to illustrate the front face of the wave and green represented the back face of the wave. The red area used a Veins noise pattern for displacing the surface around the lib. The wave surface which included the blue and green areas both used the Worley Noise pattern for displacement. When a breaking wave evolves, the pattern on the front and back faces of the wave move toward the same direction which is toward the lip of the wave, but the movements are in different direction. The movement of the noise pattern for this displacement shader was done by keyframing the offset value in the direction where the pattern for the front face would travel. The movement for the back face displacement pattern was set in this system to generate automatically in the opposite direction of that of the front face. This displacement shader and the movement of the wave surface patterns added to the realism that it was not just the deformation of the geometry but the wave was actually moving.

4.6 Digital Asset

Once the shot was created, the nodes and networks used for the generation of the breaking wave were collapsed and turned into a digital asset. A user interface was designed by referencing to the parameters used most often while creating the breaking wave shot.

4.7 Lighting and Rendering

This section illustrates how lighting and rendering were done for the breaking wave shot but were not included in the digital asset. Three lights were used in the scene. Two distant lights coming from both sides with approximately an angle of 45 degrees facing the wave. The one to the left of the wave served as the key light. A point light was also used as an ambient light for the scene.

Mantra's Micropolygon Rendering was used for rendering the whole scene. Whereas for the wave surface, Mantra would switch the Ray Tracing rendering engine automatically as the surface needed to be ray traced. For the mist and splashes, particles were rendered out as points with motion blur and depth map shadows applied. Three Mantra nodes were created in the Outputs level for the wave surface, mist and splashes respectively.

The general rendering process for all these elements was to render out an ifd file that was saved on disk first. An ifd file is a description file of the scene and is equivalent to a RIB file for Renderman. Once the ifd files for each frame were rendered out, they could be rendered using the command line and were distributed to different machines to speed up the rendering process.

4.8 Compositing

The compositing was done in Nuke. The passes used to create the breaking wave shot included a diffuse pass, a specular pass, a reflection pass, a refraction pass, a translucence (SSS) pass, an extra translucence (SSS) pass, a mist pass and a splashes pass.

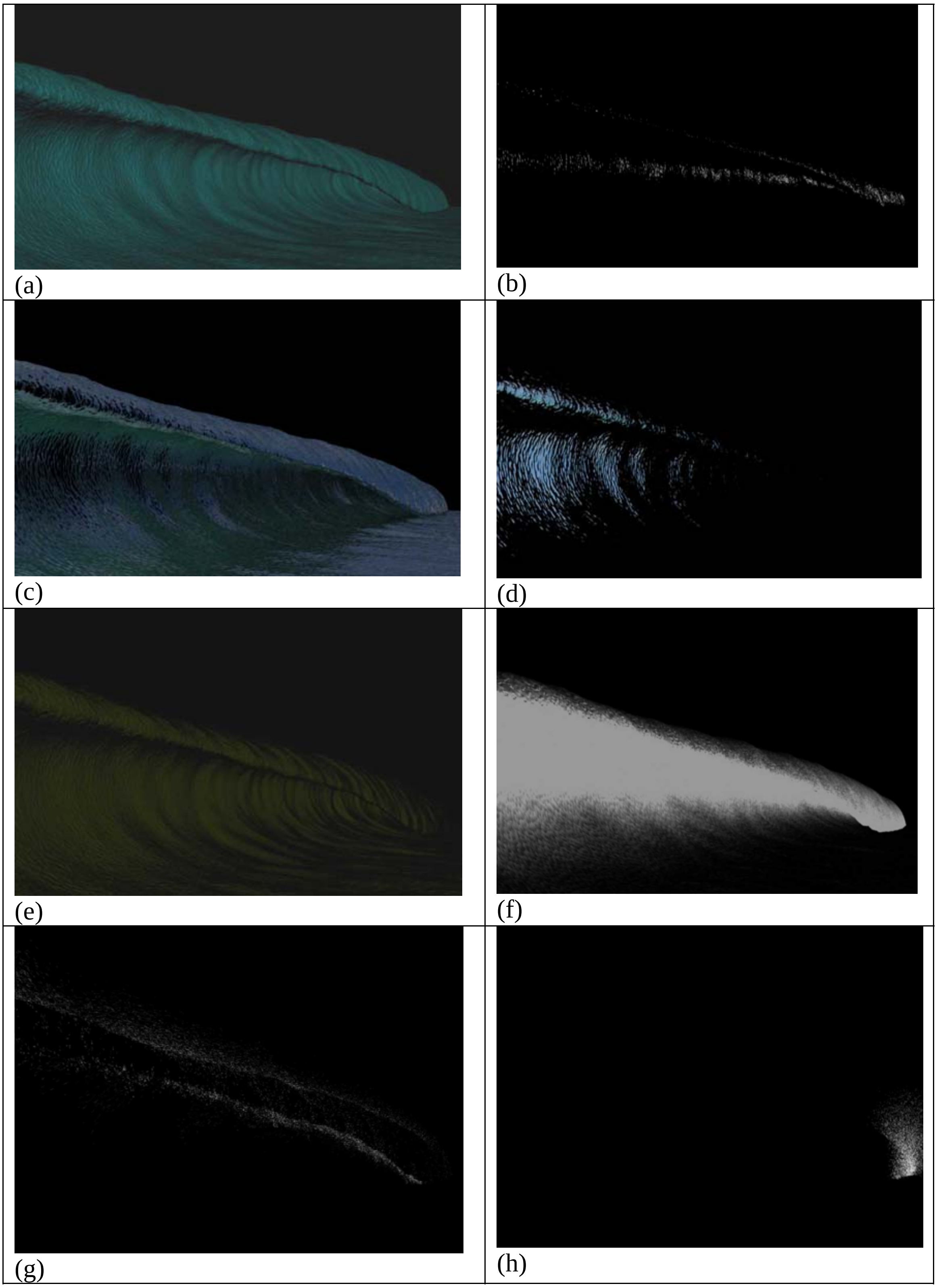


Figure 4.6: Passes used for compositing (a) diffuse pass. (b) specular pass. (c) reflection pass. (d) refraction pass. (e) translucence pass. (f) extra translucence pass. (g) mist. (h) splashes.

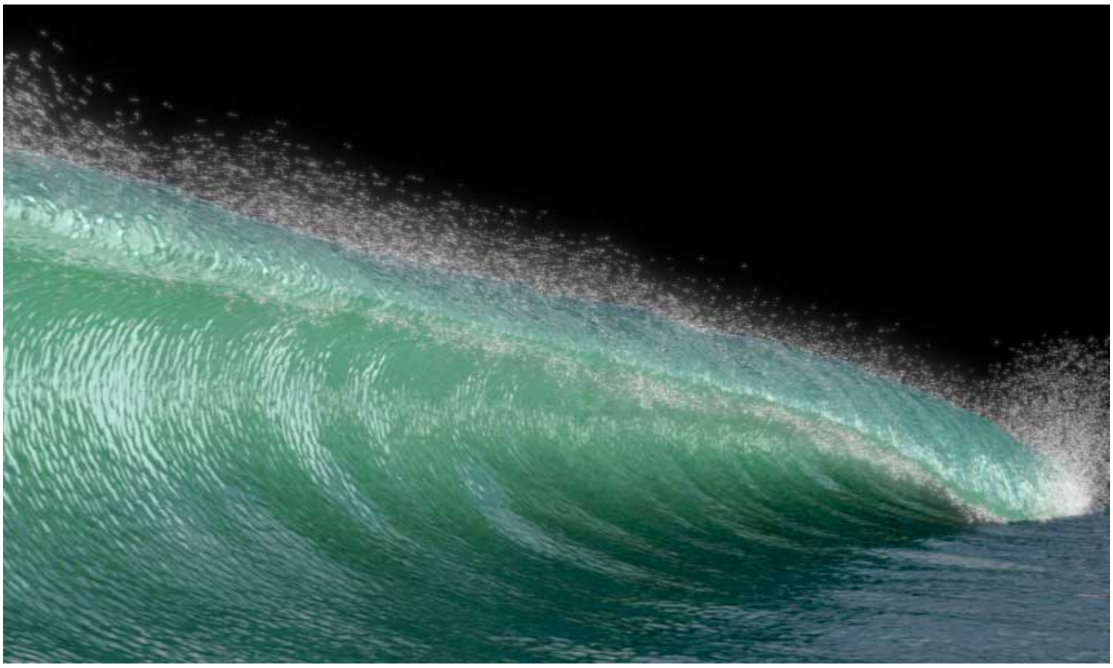


Figure 4.7: Still image for the final shot.

Chapter 5

Problems

The biggest challenge faced in the progress of this project was to facilitate smooth control and interpolation of the basic wave profile curve. The generation of the NURBS profile curve and modelling the shape into different phases of a breaking wave was fairly straightforward. However, problems started to arise once animation was involved. The first problem faced was that the points on the profile curve could not move smoothly because of the interpolation used in blendshape and the surface would produce strange shapes unexpectedly. The second problem was that when the points were not distributed evenly, the shader appeared to be stretched and changed the appearance dramatically as the wave evolved. The curve was then modified several times and more points were added in the centre area of the curve to enable more control over the part where the wave was breaking / deforming. The fixed profile curve produced an acceptable result in the shot but more work could be carried out to make it look even more realistic.

Chapter 6

Conclusion and Future Work

The aim of this project was to develop a usable tool that could be used specifically for creating breaking wave shots and actually implement the shot using this tool. The tool is now an easy-to-use digital asset that can be imported into different files or networks according to the user's needs. It is useful for generating shots for breaking waves and the outcome of the shot produced for this project was fairly pleasing. Although there is still a lot of room and potential for extending the breaking wave tool, the objective for this project has been fulfilled.

Additional features could be added to the tool to enhance its capability. As mentioned in the previous section, the wave profile curve could be modified into more desirable shape or even add more curves to the blendshape for more variation of the breaking wave. Different types of breaking waves such as the spilling breaker, collapsing breakers and surfing breakers could be included as well.

The whitewater system in the tool, especially the mist that travels along the surface tangent, still does not look realistic enough and needs adjustment. The particle system for mist could be tweaked more to make the particle behaviours resemble how mist travels on a real breaking wave surface which seems to follow a curvy line instead of a straight line.

There is also an important element missing in this tool. It is the foam on the water surface. Foam could be added into the tool using either particle systems or shaders. Ideally, foam could be integrated using noise patterns for both surface and displacement shaders as one of the initial development plans was to use as less simulation as possible.

References

[1] Fournier, A. and Reeves W. T., 1986. **A simple model of ocean waves.** *In: SIGGRAPH: Proceedings of the 13th annual conference on Computer graphics and interactive technique,* 1986 New York, NY, USA: ACM, 75-84.

[2] Peachey, D. R., 1986. Modeling waves and surf. *In: SIGGRAPH: Proceedings of the 13th annual conference on Computer graphics and interactive technique,* 1986 New York, NY, USA: ACM, 65-74.

[3] Tessendorf, J., 2004. Simulating ocean water. *In: ACM SIGGRAPH 2004 Course Notes*, 2004 New York, NY, USA: ACM.

[4] Whitehouse, D., 2008. *The Houdini Ocean Toolkit*. Available from : http://anusf.anu.edu.au/~drw900/houdini/ocean/docs/

[5] Jeschke, S., Birkholz, H., and Schmann, H., 2003. A Procedural Model for Interactive Animation of Breaking Ocean Waves. *In: WSCG POSTERS Proceedings*, February 3-7, 2003, Plzen, Czech Republic: WSCG'2003.

[6] Mihalef, V., Metaxas, D., and Sussman M., 2004. Animation and control of breaking waves. *In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2004 Aire-la-Ville, Switzerland, Switzerland: Eurographics Association , 315-324.

[7] Bredow, R., Schaub, D., Kramer, D., Hausman, M., Dimian, D., Duguid, R. S., 2007. Surf's up: the making of an animated documentary. *In: ACM SIGGRAPH 2007 courses*, 2007 New York, NY, USA: ACM, 1-123.

[8] Hurd, R., Fullerton, J., Lait, J., Leung, A., Talens, P., Tucker, S., Wagner, J., 2008. *Lighthouse*. Side Effects. Available from :http://www.sidefx.com/index.php?option=com_content&task=view&id=1349&Itemid=304

[9] Psyop, 2009. *Milk*. New York, USA: Psyop.
Available from: http://www.psyop.tv/

[10] Witzel, Z., 2009. *Rigging the waves – Behind the Scenes of Milk*, *Lighting the .waves – Behind the Scenes of Milk*. New York, USA: Psyop. Available from: http://www.florianwitzel.com/com/milk/?refresh=1

[11] Witzel, F., *Siggraph 2009 - Psyop/Mass Market*. Area Editor. Available from: http://area.autodesk.com/inhouse/videos/siggraph_2009_psyop_mass_market

[12] Sverdrup, K., 2006. *A. Fundamentals of oceanography*. 5th ed. Boston : McGraw-Hill Higher Education.

[13] Stowe, K., 1996. *Exploring ocean science*. 2nd ed. New York : John Wiley.

[14] Open University Co, 1999. *Waves, tides and shallow-water processes*. 2nd ed . Oxford : Butterworth Heinemann.

[15] Cunningham, W. and Bowmar, P., 2006 *The magic of Houdini*. Boston, MA : Thomson Course Technology.

[16] Surf's Up, 2007. Film. Directed by Ash Brannon and Chris Buck. USA: Columbia Picures and Sony Pictures Entertainment.

[17] Riding Giants, 2004. Film. Directed by Stacy Peralta. USA: Sony Pictures Classics.

[18] Blue Crush, 2002. Film. Directed by John Stockwell. USA: Universal Pictures.

[19] 2009. *Atlantic Wedge and Raglan April 09*.
Available from: http://photos.surfline.com/

[20] Jensen, H. W., 2006. *Subsurface Scattering*.
Available from: http://graphics.ucsd.edu/~henrik/images/subsurf.html

[21] 2009. *Houdini Documentation*. Available from:http://www.sidefx.com/

Appendix

User Guide

To install the digital asset for the breaking wave, first go to **File > Install Digital Asset Library** in a Houdini scene. Then specify the path where the wave.otl is contained, use the **Install Library To** option to choose where the type library would be available, then press **Accept**. [21]

The breaking wave digital asset exists within the **Geometry Level**, so lay down a **Geometry Object** in the **Scene Level**, then dive inside it, delete the default **File SOP**. Press the **Tab** key and type **Breaking Waves,** select and press **Enter** to put down the asset.

Breaking Waves is used for generating breaking ocean waves. This asset allows the user the create the geometry, animation, shading and whitewater for the wave.

Parameters

Wave Geometry

Wave Height	This is the height of the wave.
Wavelength	This is the length of a wave from one crest/or trough to another.
Translate	The location to put down the breaking wave.
Smooth	This is the overall smoothness of the surface. It controls the amount the wave surface is subdivided.
Fractal Depth	This is the number of iterations of the fractal Brownian motion noise on the wave surface.
Roughness	Fractal roughness.
Height	Displacement amount of the height of the fractal noise on the wave surface.
Frequency	Noise frequency.
Offset	Noise offset.

Animation

Control Type	Choose whether the animation is using an Auto or a Custom control.

Auto

Center	This is the point along the center of the cross-section wave profile curves that the wave starts to break.
Span	How wide the breaking wave is.
Falloff	Adds non-linear variation to the wave shape. When the Falloff is greater than 1, it has an ease out effect on the breaking wave. If the Falloff is less than 1, the life-cycle of the breaking wave tends to be shorter.
Time	Controls the evolution of a breaking wave. 0 means at the beginning of the breaking wave's life-cycle. As the value increases, the wave shape evolves and deforms out of the centre line of the surface. Time = 1 means that the wave reaches the very end of its life-cycle.
Speed	How fast the wave is moving forward.

Custom

Wave Profile (s)	Controls the evolution of a single breaking wave profile curve.
Forward (s)	This is the forward motion of a single breaking wave profile curve.

Whitewater

Mist

Display	This is whether to display the particles or not.
Birth Rate	Number of particles to emit.
Life Expectancy	How long the particle will live (in seconds).
Life Variance	Particles will live the number of seconds in Life expectancy, plus or minus this number of seconds. Use 0 for no variance.
Initial Velocity	Set or add to velocity attribute.
Variance	Variance to the initial velocity set above
Attractor Scale	This is how strong the attractor (moving along the surface) affects the particles moving direction.
Turbulence	Number of iterations of fractal noise to add.
Noise Frequency	Spatial frequency of noise field in X, Y, and Z.
Noise Amplitude	Maximum value of noise field.
Gravity	This is how strong the gravity affects the particles.

Splash

Display	This is whether to display the particles or not.
Birth Rate	Number of particles to emit.
Life Expectancy	How long the particle will live (in seconds).
Life Variance	Particles will live the number of seconds in Life expectancy, plus or minus this number of seconds. Use 0 for no variance.
Initial Velocity	Set or add to velocity attribute.
Variance	Variance to the initial velocity set above
Splash Up	How high the particles will travel.
Splash Spread	How wide the particles will travel.
Jitter	Overall scale of the jitter effect.
Gravity	This is how strong the gravity affects the particles.

Shading

Surface

Diffuse

Diffuse Intensity	The contribution of diffuse to the material.
Diffuse Color	This is the diffuse color.
Diffuse Roughness	This controls the size or spread of the diffuse contribution.

Lighting Effects

Shadows

Shadow Color	This value is used to tint the color of the shadow.

SSS

Subsurface Intensity	This is used to control the intensity of the returned color.
Subsurface Color	The color used to tint the color returned from the subsurface scatter calculation.
Index of Refraction	This is the Index of Refraction used in the subsurface scattering calculation to determine how muchlight bends as it pass through the surface.
Depth	This specifies how far the light travels after it pass through the surface.
Ensure Faces Point Forward	This toggle will change the normal to face forward towards the camera. It is recommended this toggle be on.

Specular

Specular Intensity	A multiplier for the specular contribution to the material.
Specular Color	The specular highlight color.
Specular U Roughness	This value controls the size or spread of the specular highlight. If Anisotropic is selected, this value controls the specular highlight in only the U direction.
Specular V Roughness	If Anisotropic is selected, this value controls the specular highlight in only the V direction. It is ignored with any other specular function.
Specular Sharpness	This value controls the sharpness of the edge of the specular highlight. It is used only if Glossy specular is selected.
Specular Type	A selector for the specular function.

Reflections

Reflection Intensity	A multiplier for the reflection contribution to the material.
Reflection Tint	This is a tint color for reflections.

Ray Trace

Refraction Intensity	A multiplier for the refraction contribution to the material.
Refraction Tint	This is a tint color for refraction.
Index of Refraction	This determines how much the ray bends as it passes from one material into another. Usually for water this is 1.333.
Jitter Amount	This controls how randomly the samples are distributed.
Ray Bias	This is typically a small number used to help solve self-intersection issues.
Area Samples	The number samples to calculate. The more samples, the better the quality, the slower the render.

Environment Map

Environment Map	If there is the path to a texture here and ray trace is turned off, then this map is used to calculate the reflection contribution. If ray trace is on and there is a texture, then the texture color is returned, when the ray does not hit anything.
Transform Space	he reflection ray used to do the look up in the map is transformed into this space.
Background Color	This color is returned when the ray does not hit anything and there is no environment map specified.

Displacement

Frequency	This is the scale of the noise pattern. Larger values give smaller, but not more detailed, patterns. Smaller values will benefit from higher Octaves values.
Offset	This vector positions the noise on the object in x, y and z.
Noise Bias	This value shifts the median value of the noise by passing it through a power function. Values greater than 1, will dampen the noise down, whereas values less than 1 will sharpen the contrast between the noise values.
Lower Noise Range	Control the size of the space between the rough areas.
Upper Noise Range	Control the height of the rough areas.