Simulating Ocean Water

Jerry Tessendorf

1 Introduction and Goals

These notes are intended to give computer graphics programmers and artists an introduction to methods of simulating, animating, and rendering ocean water environments. CG water has become a common tool in visual effects work at all levels of computer graphics, from print media to feature films. Several commercial products are now available for nearly any computer platform and work environment, in addition to proprietary tools held by a few companies, which generate high quality geometry and images. In order for an artist to exploit the tools to maximum benefit, it is important that he or she become familiar with concepts, terminology, a little oceanography, and the present state of the art.

As demonstrated by the pioneering efforts in the films *Waterworld* and *Titanic*, as well as several other films made since about 1995, images of cg water can be generated with a high degree of realism. However, this level of realism is limited to relatively calm, nice ocean conditions. Conditions with large amounts of spray, breaking waves, foam, splashing, and wakes are approaching the same realistic look.

Broadly, the reader should come away from this material with (1) an understanding of some algorithms that generate/animate water surface height fields suitable for modeling waves as big as storm surges and as small as tiny capillaries; (2) an understanding of the basic optical processes of reflection and refraction from a water surface; (3) an introduction to the color filtering behavior of ocean water; (4) an introduction to complex lighting effects known as caustics and godrays, produced when sunlight passes through the rough surface into the water volume underneath; and (5) some rules of thumb for which choices make nice looking images and what are the tradeoffs of quality versus computational resources. Some example shaders are provided, and example renderings demonstrate the content of the discussion.

Before diving into it, I first want to be more concrete about what aspect of the ocean environment we cover (or not cover) in these notes. Figure 1 is a rendering of an oceanscape produced from models of water, air, and clouds. Light from the clouds is reflected from the surface. On the extreme left, sun glitter is also present. The generally bluish color of the water is due to the reflection of blue skylight, and to light coming out of the water after scattering from the volume. Although these notes do not tackle the modeling and rendering of clouds and air, there is a discussion of how skylight from the clouds and air is reflected from, or refracted through, the water surface. These notes will tell you how to make a height-field displacement-mapped surface for the ocean waves with the detail and quality shown in the figure. The notes also discuss several effects of the underwater environment and how to model/render them. The primary four effects are sunbeams (also called godrays), caustics on underwater surfaces, blurring by the scattering of light, and color filtering.

There are also many other complex and interesting aspects of the ocean environment that will not be covered. These include breaking waves, spray, foam, wakes around objects in the water, splashes from bodies that impact the surface, and global illumination of the entire ocean-atmosphere environment. There is substantial research underway on these topics, and so it is possible that future versions of this or other lecture notes will include them. I have included a brief section on advanced modifications to the basic wave height algorithm that produce choppy waves. The modification could feasibly lead to a complete description of the surface portion of breaking waves, and possibly serve to drive the spray and foam dynamics as well.

There is, of course, a substantial body of literature on ocean surface simulation and animation, both in computer graphics circles and in oceanography. One of the first descriptions of water waves in computer graphics was by Fournier and Reeves[8] , who modeled a shoreline with waves coming up on it using a water surface model called *Gerstner waves*. In that same issue, Darwin Peachey[9] presented a variation on this approach using basis shapes other than sinusoids.

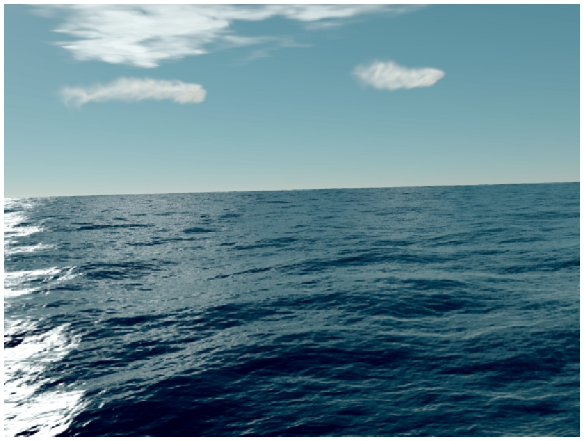

Figure 1: Rendered image of an oceanscape.

In the oceanographic literature, ocean optics became an intensive topic of research in the 1940s. S.Q. Duntley published[13] in 1963 papers containing optical data of relevance to computer graphics. Work continues today. The field of optical oceanography has grown into a mature quantitative science with subdisciplines and many different applications. One excellent review of the state of the science was written by Curtis Mobley[14].

In these lectures the approach we take to creating surface waves is close to the one outlined by Masten, Watterberg, and Mareda[7], although the technique had been in use for many years prior to their paper in the optical oceanography community. This approach synthesizes a patch of ocean waves from a Fast Fourier Transform (FFT) prescription, with user-controllable size and resolution, and which can be tiled seamlessly over a larger domain. The patch contains many octaves of sinusoidal waves that all add up at each point to produce the synthesized height. The mixture of sinusoidal amplitudes and phases however, comes from statistical, emperically-based models of the ocean. What makes these sinusoids look like waves and not just a bunch of sine waves is the large collection of sinusoids that are used, the relative amplitudes of the sinusoids, and their animation using the dispersion relation. We examine the impact of the number of sinusoids and resolution on the quality of the rendered image.

In the next section we begin the discussion of the ocean environment with a broad introduction to the global illumination problem. The radiosity equations for this environment look much like those of any other radiosity problem, although the volumetric character of some of the environmental components complicate a general implementation considerably. However, we simplify the issues by ignoring some interactions and replacing others with models generated by remote sensing data.

Practical methods are presented in section 3 for creating realizations of ocean surfaces. We present two methods, one based on a simple model of water structure and movement, and one based on summing up large numbers of sine waves with amplitudes that are related to each other based on experimental evidence. This second method carries out the sum using the technique of Fast Fourier Transformation (fft), and has been used effectively in projects for commercials, television, and motion pictures.

After the discussion of the structure and animation of the water surface, we focus on the optical properties of water relevant to the graphics problem. First, we discuss the interaction at the air-water interface: reflection and refraction. This leaves us with a simple but effective Renderman-style shader suitable for rendering water surfaces in BMRT, for example. Next, the optical characteristics of the underwater environment are explored.

2 Radiosity of the Ocean Environment

The ocean environment, for our purposes, consists of only four components: The water surface, the air, the sun, and the water below the surface. In this section we trace the flow of light through the environment, both mathematically and schematically, from the light source to the camera. In general, the radiosity equations here are as coupled as any other radiosity problem. To a reasonable degree, however, the coupling can be truncated and the simplified radiosity problem has a relatively fast solution .

The light seen by a camera is dependent on the flow of light energy from the source(s) (i.e. the sun and sky) to the surface and into the camera. In addition to specular reflection of direct sunlight and skylight from the surface, some fraction of the incident light is transmitted through the surface. Ultimately, some fraction of the transmitted light is scattered by the water volume back up into the air. Some of the light that is reflected or refracted at the surface may strike the surface a second time, producing more reflection and refraction events. Under some viewing conditions, multiple reflections and refractions can have a noticeable impact on images. For our part however, we will ignore more than one reflection or refraction from the surface at a time. This not only makes the algorithms and computation easier and faster, but also is reasonably accurate most of the time.

At any point in the environment above the surface, including at the camera, the total light intensity (radiance) coming from any direction has three contributions:

$$L_{ABOVE} = rL_S + rL_A + t_U L_U \ , \tag{1}$$

with the following definitions of the terms:

r is the Fresnel reflectivity for reflection from a spot on the surface of the ocean to the camera.

t_U is the transmission coefficient for the light L_U coming up from the ocean volume, refracted at the surface into the camera.

L_S is the amount of light coming directly from the sun, through the atmosphere, to the spot on the ocean surface where it is reflected by the surface to the camera.

L_A is the (diffuse) atmospheric skylight

L_U is the light just below the surface that is transmitted through the surface into the air.

Equation 1 has intentionally been written in a shorthand way that hides the dependences on position in space and the direction the light is traveling.

While equation 1 appears to have a relatively simple structure, the terms L_S, L_A, and L_U can in principle have complex dependencies on each other, as well on the reflectivity and transmissivity. There is a large body of research literature investigating these dependencies in detail [15], but we will not at this point pursue these quantitative methods. But we can elaborate further on the coupling while continuing with the same shorthand notation. The direct light from the sun L_S is

$$L_S = L_{TOA} \exp\{-\tau\} \ , \tag{2}$$

where L_{TOA} is the intensity of the direct sunlight at the top of the atmosphere, and τ is the "optical thickness" of the atmosphere for the direction of the sunlight and the point on the earth. Both the diffuse atmospheric skylight L_A and the upwelling light L_U can be written as the sum of two terms:

$$\begin{aligned} L_A &= L_A^0(L_S) + L_A^1(L_U) & (3) \\ L_U &= L_U^0(L_S) + L_U^1(L_A) & (4) \end{aligned}$$

These equations reveal the potential complexity of the problem. While both L_A and L_U depend on the direct sunlight, they also depend on each other. For example, the total amount of light penetrating into the ocean comes from the direct sunlight and from the atmospheric sunlight. Some of the light coming into the ocean is scattered by particulates and molecules in the ocean, back up into the atmosphere. Some of that upwelling light in turn is scattered in the atmosphere and becomes a part of the skylight shining on the surface, and on and on. This is a classic problem in radiosity. It is not particularly special for this case, as opposed to other radiosity problems, except perhaps for the fact that the upwelling light is difficult to compute because it comes from volumetric multiple scattering.

Our approach, for the purposes of these notes, to solving this radiosity problem is straightforward: take the skylight to depend only on the light from the sun, since the upwelling contribution represents a "tertiary" dependence on the sunlight; and completely replace the equation for L_U with an empirical formula, based on scientific observations of the oceans, that depends only on the direct sunlight and a few other parameters that dictate water type and clarity.

Under the water surface, the radiosity equation has the schematic form

$$L_{BELOW} = tL_D + tL_I + L_{SS} + L_M \ , \tag{5}$$

with the meaning

t is the Fresnel transmissivity for transmission through the water surface at each point and angle on the surface.

L_D The "direct" light from the sun that penetrates into the water.

L_I The "indirect" light from the atmosphere that penetrates into the water.

L_{SS} The single-scattered light, from both the sun and the atmosphere, that is scattered once in the water volume before arriving at any point.

L_M The multiply-scattered light. This is the single-scattered light that undergoes more scattering events in the volume.

Just as for the above water case, these terms are all related to each other is relative complex ways. For example, the single scattered light depends on the direct and indirect light:

$$L_{SS} = P(tL_I) + P(tL_D) \tag{6}$$

with the quantity P being a linear functional operator of its argument, containing information about the single scattering event and the attenuation of the scattered light as it passes from the scatter point to the camera. Similarly, the multiply-scattered light is dependent on the single scattered:

$$L_M = G(tL_I) + G(tL_D) \ . \tag{7}$$

The functional schematic quantities P and G are related, since multiple scattering is just a series of single scatters. Formally, the two have an operator dependence that has the form

$$\begin{aligned} G &\sim P \otimes P \otimes \left\{ 1 + P + \frac{1}{2!} P \otimes P + \frac{1}{3!} P \otimes P \otimes P + \ldots \right\} \\ &\sim P \otimes P \otimes \exp(P) \ . & (8) \end{aligned}$$

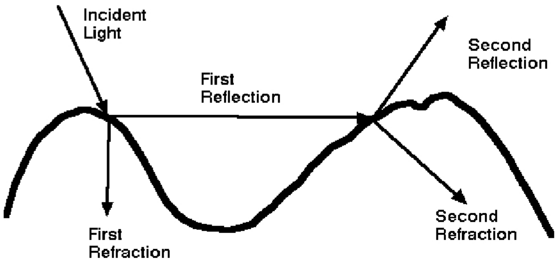

Figure 2: Illustration of multiple reflections and transmission through the air-water interface.

At this point, the schematic representation may have outlived its usefullness because of the complex (and not here defined) meaning of the convolution-like operator $\otimes$, and because the expression for G in terms of P has created an even more schematic view in terms of an exponentiated P. So for now we will leave the schematic representation, and journey on with more concrete quantities the rest of the way through.

The formal schematic discussion put forward here does have a mathematically and physically precise counterpart. The field of study in Radiative Transfer has been applied for some time to water optics, by a large number researchers. The references cited are excellent reading for further information.

As mentioned, there is one additional radiosity scenario that can be important to ocean rendering under certain circumstances, but which we will not consider. The situation is illustrated in figure 2. Following the trail of the arrows, which track the direction light is travelling, we see that sometimes light coming to the surface (from above or below), can reflect and/or transmit through the surface more than once. The conditions which produce this behavior in significant amounts are: the wave heights must be fairly high, and the direction of viewing the waves, or the direction of the light source must be nearly grazing the surface. The higher the waves are, the less grazing the light source or camera need to be. This phenonmenon has been examined experimentally and in computer simulations. It is reasonably well understood, and we will ignore it from this point on.

3 Practical Ocean Wave Algorithms

In this section we focus on algorithms and practical steps to building height fields for ocean waves. Although we will be occupied mostly by a method based on Fast Fourier Transforms (FFTs), we begin by introducing a simpler description called Gerstner Waves. This is a good starting point for several reasons: the mathematics is relatively light compared to FFTs, several important oceanographic concepts can be introduced, and they give us a chance to discuss wave animation. After this discussion of Gerstner waves, we go after the more complex FFT method, which produces wave height fields that are more realistic. These waves, called “linear waves” or “gravity waves” are a fairly realistic representation of typical waves on the ocean when the weather is not too stormy. Linear waves are certainly not the whole story, and so we discuss also some methods by which oceanographers expand the description to “nonlinear waves”, waves passing over a shallow bottom, and very tiny waves about one millimeter across called capillary waves.

In the course of this discussion, we will see how quantities like windspeed, surface tension, and gravitational acceleration come into the practical implementation of the algorithms.

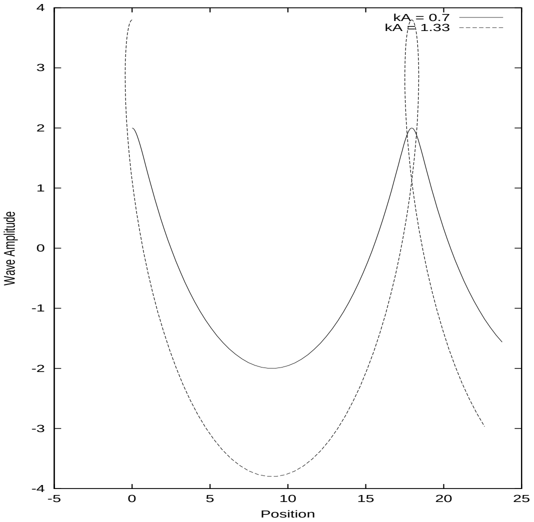

Figure 3: Profiles of two single-mode Gerstner waves, with different relative amplitudes and wavelengths.

3.1 Gerstner Waves

Gerstner waves were first found as an approximate solution to the fluid dynamic equations almost 200 years ago. There first application in computer graphics seems to be the work by Fournier and Reeves in 1986 (cited previously). The physical model is to describe the surface in terms of the motion of individual points on the surface. To a good approximation, points on the surface of the water go through a circular motion as a wave passes by. If a point on the undisturbed surface is labelled $\mathbf{x}_0 = (x_0, z_0)$ and the undisturbed height is $y_0 = 0$, then as a single wave with amplitude A passes by, the point on the surface is displaced at time t to

$$\begin{aligned} \mathbf{x} &= \mathbf{x}_0 - (\mathbf{k}/k) A \sin(\mathbf{k}\cdot\mathbf{x}_0 - \omega t) \qquad (9) \\ y &= A\cos(\mathbf{k}\cdot\mathbf{x}_0 - \omega t)\ . \qquad (10) \end{aligned}$$

In these expressions, the vector $\mathbf{k}$, called the wavevector, is a horizontal vector that points in the direction of travel of the wave, and has magnitude k related to the length of the wave (λ) by

$$k = 2\pi/\lambda \qquad (11)$$

The frequency w is related to the wavevector, as discussed later.

Figure 3 shows two example wave profiles, each with a different value of the dimensionless amplitude kA. For values $kA < 1$, the wave is periodic and shows a steepening at the tops of the waves as kA approaches 1. For $kA > 1$, a loop forms at the tops of the wave, and the “insides of the wave surface are outside”, not a particularly desirable or realistic effect.

As presented so far, Gerstner waves are rather limited because they are a single sine wave horizontally and vertically. However, this can be generalized to a more complex profile by summing a set of sine waves. One picks a set of wavevectors $\mathbf{k}_i$, amplitudes A_i, frequencies ω_i, and phases ϕ_i, for $i = 1, \ldots, N$, to get the expressions

$$\mathbf{x} = \mathbf{x}_0 - \sum_{i=1}^{N} (\mathbf{k}_i/k_i) A_i \sin(\mathbf{k}_i\cdot\mathbf{x}_0 - \omega_i t + \phi_i) \qquad (12)$$

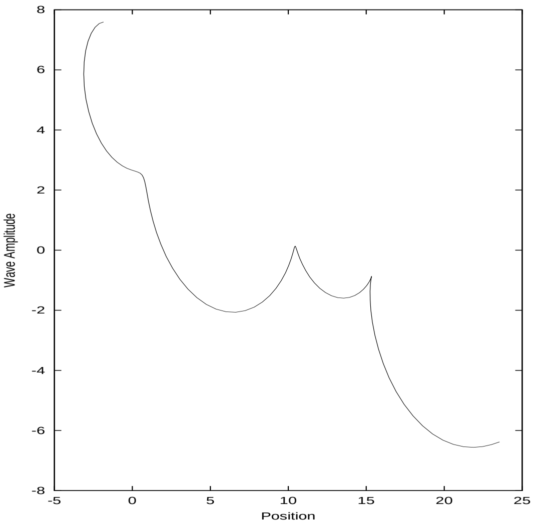

Figure 4: Profile of a 3-mode Gerstner wave.

$$y = \sum_{i=1}^{N} A_i \cos(\mathbf{k}_i \cdot \mathbf{x}_0 - \omega_i t + \phi_i) \ . \tag{13}$$

Figure 4 shows an example with three waves in the set. Interesting and complex shapes can be obtained in this way.

3.2 Animating Waves: The Dispersion Relation

The animated behavior of Gerstner waves is determined by the set of frequencies ω_i chosen for each component. For water waves, there is a well-known relationship between these frequencies and the magnitude of their corresponding wavevectors, k_i. In deep water, where the bottom may be ignored, that relationship is

$$\omega^2(k) = gk \ . \tag{14}$$

The parameter g is the gravitational constant, nominally $9.8m/sec^2$. This dispersion relationship holds for Gerstner waves, and also for the FFT-based waves introduced next.

There are several conditions in which the dispersion relationship is modified. When the bottom is relatively shallow compared to the length of the waves, the bottom has a retarding affect on the waves. For a bottom at a depth D below the mean water level, the dispersion relation is

$$\omega^2(k) = gk \tanh(kD) \tag{15}$$

Notice that if the bottom is very deep, the behavior of the $\tanh$ function reduces this dispersion relation to the previous one.

A second situation which modifies the dispersion relation is surface tension. Very small waves, with a wavelength of about 1 cm or less, have an additional term:

$$\omega^2(k) = gk(1 + k^2L^2) \ , \tag{16}$$

and the parameter L has units of length. Its magnitude is the scale for the surface tension to have effect.

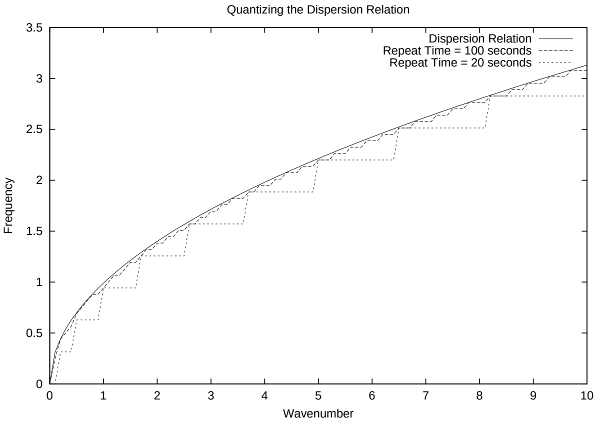

Figure 5: A comparison of the continuous dispersion curve $\omega = \sqrt{gk}$ and quantized dispersion curves, for repeat times of 20 seconds and 100 seconds. Note that for a longer repeat time, the quantized is a closer approximation to the original curve.

Using these dispersion relationships, it is very difficult to create a sequence of frames of water surface which for a continuous loop. In order to have the sequence repeat after a certain amount of time T for example, it is necessary that all frequencies be multiples of the basic frequence

$$\omega_0 \equiv \frac{2\pi}{T} \ . \tag{17}$$

However, when the wavevectors $\mathbf{k}$ are distributed on a regular lattice, itis impossible to arrange the dispersion-generated frequencies to also be on a uniform lattce with spacing ω_0.

The solution to that is to not use the dispersion frequences, but instead a set that is close to them. For a given wavenumber k, we use the frequency

$$\bar{\omega}(k) = \left[\left[\frac{\omega(k)}{\omega_0} \right]\right] \omega_0 \ , \tag{18}$$

where $[[a]]$ means take the integer part of the value of a, and $\omega(k)$ is any dispersion relationship of interest. The frequencies $\bar{\omega}(k)$ are a *quantization* of the dispersion surface, and the animation of the water surface loops after a time T because the quantized frequencies are all integer multiples of ω_0. Figure 5 plots the original dispersion curve, along with quantized dispersion curves for two choices of the repeat time T.

3.3 Statistical Wave Models and the Fourier Transform

Oceanographic literature tends to downplay Gerstner waves as a realistic model of the ocean. Instead, statistical models are used, in combination with experimental observations. In the statistical models, the wave height is considered a random variable of horizontal position and time, $h(\mathbf{x}, t)$.

Statistical models are also based on the ability to decompose the wave height field as a sum of sine and cosine waves. The value of this decomposition is that the amplitudes of the waves have nice mathematical and statistical properties, making it simpler to build models. Computationally, the decomposition uses Fast Fourier Transforms (ffts), which are a rapid method of evaluating the sums.

The fft-based representation of a wave height field expresses the wave height $h(\mathbf{x}, t)$ at the horizontal position $\mathbf{x} = (x, z)$ as the sum of sinusoids with complex, time-dependent amplitudes:

$$h(\mathbf{x}, t) = \sum_{\mathbf{k}} \tilde{h}(\mathbf{k}, t) \, \exp(i\mathbf{k} \cdot \mathbf{x}) \tag{19}$$

where t is the time and $\mathbf{k}$ is a two-dimensional vector with components $\mathbf{k} = (k_x, k_z)$, $k_x = 2\pi n/L_x$, $k_z = 2\pi m/L_z$, and n and m are integers with bounds $-N/2 \le n < N/2$ and $-M/2 \le m < M/2$. The fft process generates the height field at discrete points $\mathbf{x} = (nL_x/N, mL_z/M)$. The value at other points can also be obtained by switching to a *discrete* fourier transform, but under many circumstances this is unnecessary and is not applied here. The height amplitude Fourier components, $\tilde{h}(\mathbf{k}, t)$, determine the structure of the surface. The remainder of this subsection is concerned with generating random sets of amplitudes in a way that is consistent with oceanographic phenomenology.

For computer graphics purposes, the slope vector of the wave-height field is also needed in order to find the surface normal, angles of incidence, and other aspects of optical modeling as well. One way to compute the slope is though a finite difference between fft grid points, separated horizontally by some 2D vector $\Delta\mathbf{x}$. While a finite difference is efficient in terms of memory requirements, it can be a poor approximation to the slope of waves with small wavelength. An exact computation of the slope vector can be obtained by using more ffts:

$$\epsilon(\mathbf{x}, t) = \nabla h(\mathbf{x}, t) = \sum_{\mathbf{k}} i\mathbf{k} \, \tilde{h}(\mathbf{k}, t) \, \exp(i\mathbf{k} \cdot \mathbf{x}) \ . \tag{20}$$

In terms of this fft representation, the finite difference approach would replace the term $i\mathbf{k}$ with terms proportional to

$$\exp(i\mathbf{k} \cdot \Delta\mathbf{x}) - 1 \tag{21}$$

which, for small wavelength waves, does not well approximate the gradient of the wave height. Whenever possible, slope computation via the fft in equation 20 is the prefered method.

The fft representation produces waves on a patch with horizontal dimensions $L_x \times L_z$, outside of which the surface is perfectly periodic. In practical applications, patch sizes vary from 10 meters to 2 kilometers on a side, with the number of discrete sample points as high as 2048 in each direction (i.e. grids that are 2048×2048, or over 4 million waves). The patch can be tiled seamlessly as desired over an area. The consequence of such a tiled extension, however, is that an artificial periodicity in the wave field is present. As long as the patch size is large compared to the field of view, this periodicity is unnoticeable. Also, if the camera is near the surface so that the effective horizon is one or two patch lengths away, the periodicity will not be noticeable in the look-direction, but it may be apparent as repeated structures across the field of view.

Oceanographic research has demonstrated that equation 19 is a reasonable representation of naturally occurring wind-waves in the open ocean. Statistical analysis of a number of wave-buoy, photographic, and radar measurements of the ocean surface demonstrates that the wave height amplitudes $\tilde{h}(\mathbf{k}, t)$ are nearly statistically stationary, independent, gaussian fluctuations with a spatial spectrum denoted by

$$P_h(\mathbf{k}) = \left\langle \left| \tilde{h}^*(\mathbf{k}, t) \right|^2 \right\rangle \tag{22}$$

for data-estimated ensemble averages denoted by the brackets $\langle\ \rangle$.

There are several analytical semi-empirical models for the wave spectrum $P_h(\mathbf{k})$. A useful model for wind-driven waves larger than capillary waves in a fully developed sea is the *Phillips* spectrum

$$P_h(\mathbf{k}) = A \frac{\exp\left(-1/(kL)^2\right)}{k^4} |\hat{\mathbf{k}} \cdot \hat{w}|^2 \ , \tag{23}$$

where $L = V^2/g$ is the largest possible waves arising from a continuous wind of speed V, g is the gravitational constant, and $\hat{w}$ is the direction of the wind. A is a numeric constant. The cosine factor $|\hat{\mathbf{k}} \cdot \hat{w}|^2$ in the Phillips spectrum eliminates waves that move perpendicular to the wind direction. This model, while relatively simple, has poor convergence properties at high values of the wavenumber $|\mathbf{k}|$. A simple fix is to suppress waves smaller that a small length $\ell \ll L$, and modify the Phillips spectrum by the multiplicative factor

$$\exp\left(-k^2 \ell^2\right) \ . \tag{24}$$

Of course, you are free to "roll your own" spectrum to try out various effects.

3.4 Building a Random Ocean Wave Height Field

Realizations of water wave height fields are created from the principles elaborated up to this point: gaussian random numbers with spatial spectra of a prescribed form. This is most efficiently accomplished directly in the fourier domain. The fourier amplitudes of a wave height field can be produced as

$$\tilde{h}_0(\mathbf{k}) = \frac{1}{\sqrt{2}} (\xi_r + i\xi_i) \sqrt{P_h(\mathbf{k})} \ , \tag{25}$$

where ξ_r and ξ_i are ordinary independent draws from a gaussian random number generator, with mean 0 and standard deviation 1. Gaussian distributed random numbers tend to follow the experimental data on ocean waves, but of course other random number distributions could be used. For example, log-normal distributions could be used to produce height fields that are vary "intermittent", i.e. the waves are very high or nearly flat, with relatively little in between.

Given a dispersion relation $\omega(k)$, the Fourier amplitudes of the wave field realization at time t are

$$\begin{aligned} \tilde{h}(\mathbf{k}, t) &= \tilde{h}_0(\mathbf{k}) \exp\{i\omega(k)t\} \\ &+ \tilde{h}_0^*(-\mathbf{k}) \exp\{-i\omega(k)t\} \end{aligned} \tag{26}$$

This form preserves the complex conjugation property $\tilde{h}^*(\mathbf{k}, t) = \tilde{h}(-\mathbf{k}, t)$ by propagating waves "to the left" and "to the right". In addition to being simple to implement, this expression is also efficient for computing $h(\mathbf{x}, t)$, since it relies on ffts, and because the wave field at any chosen time can be computed without computing the field at any other time.

In practice, how big does the Fourier grid need to be? What range of scales is reasonable to choose? If you want to generate wave heights faster, what do you do? Lets take a look at these questions.

How big should the Fourier grid be? The values of N and M can be between 16 and 2048, in powers of two. For many situations, values in the range 128 to 512 are sufficient. For extremely detailed surfaces, 1024 and 2048 can be used. For example, the wave fields used in the motion pictures *Waterworld* and *Titanic* were 2048×2048 in size, with the spacing between grid points at about 3 cm. Above a value of 2048, one should be careful because the limits of numerical accuracy for floating point calculations can become noticeable.

What range of scales is reasonable to choose? The answer to this question comes down to choosing values for L_x, L_z, M, and N. The smallest facet in either direction is $dx \equiv L_x/M$ or $dz \equiv L_z/N$. Generally, dx and dz need never go below 2 cm or so. Below this scale, the amount of wave action is small compared to the rest of the waves. Also, the physics

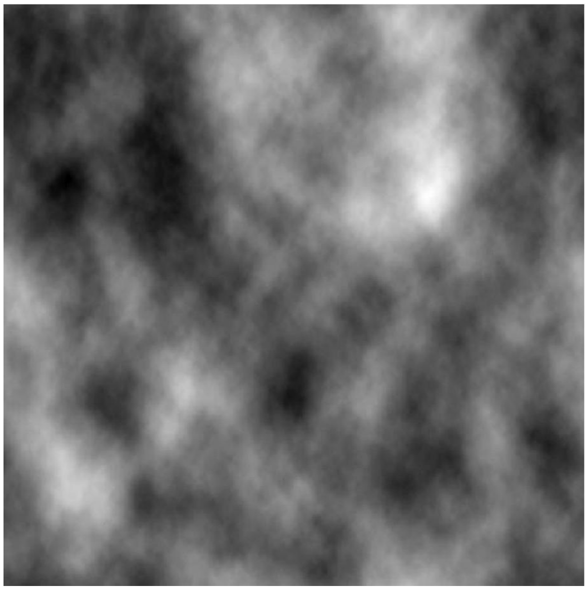

Figure 6: A surface wave height realization, displayed in greyscale.

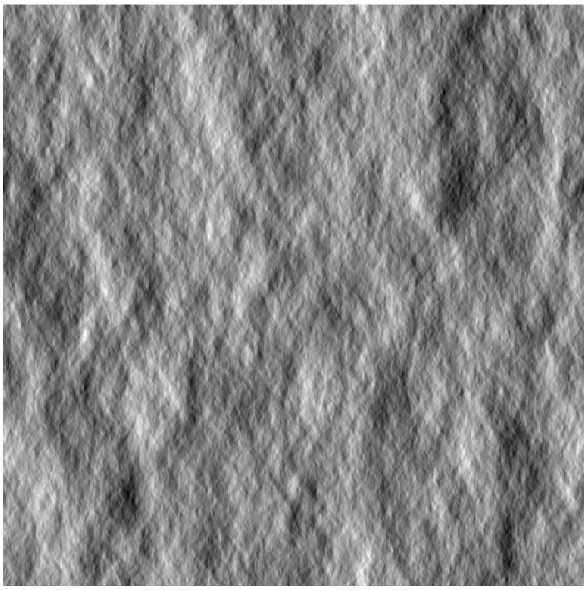

Figure 7: The x-component of the slope for the wave height realization in figure 6.

of wave behavior below 2 cm begins to take on a very different character, involving surface tension and "nonlinear" processes. From the form of the spectrum, waves with a wavelength larger than V^2/g are suppressed. So make sure that dx and dz are smaller than V^2/g by a substantial amount (10 - 1000) or most of the interesting waves will be lost. The secret to realistic looking waves (e.g. figure 9 (a) compared to figure 9 (c)) is to have M and N as large as reasonable.

How do you generate wave height fields in the fastest time? The time consuming part of the computation is the fast fourier transform. Running on a 180 MHz PowerPC 603e processor (under LinuxPPC r4), a 1024x1024 fft takes less than a minute. However, faster times are achieved by setting M and N to smaller powers of 2.

3.5 Examples: Height Fields and Renderings

We now turn to some examples of waves created using the fft approach discussed above. We will show waves in two formats: as greyscale images in which the grey level is proportional to wave height; and renderings of oceanscapes using several different rendering packages to illustrate what is possible.

In the first set of examples, the grid size is set to $M = N = 512$, with $L_x = L_z = 1000$ meters. The wind speed is a gale force at $V = 31$ meters/second, moving in the x-direction. The small-wave cutoff of $\ell = 1$ meter was also used. Figure 6 is a greyscale representation of the wave height: brighter means higher and darker means lower height. Although produced by the fft algorithms described here, figure 6 is not obviously a water height field. It may help to examine figure 7, which is a greyscale depiction of the x-component of the slope. This looks more like water waves that figure 6. What is going on?

Figures 6 and 7 demonstrate a consequence of water surface optics, discussed in the next section: the visible qualities of the surface structure tend to be strongly influenced by the slope of the waves. We will discuss this in quantitative detail, but for now we willl summarize it by saying that the reflectivity of the water is a strong function of the slope of the waves, as well as the directions of the light(s) and camera.

To illustrate a simple effect of customizing the spectrum model, figure 8 is the greyscale display of a height field identical to figure 6, with the exception that the directional factor $|\hat{\mathbf{k}} \cdot \hat{w}|^2$ in equation 23 has been changed to $|\hat{\mathbf{k}} \cdot \hat{w}|^6$. The surface is clearly more aligned with the direction of the wind.

The next example of a height field uses a relatively simple shader in BMRT, the Renderman-compliant raytracer. The shader is shown in the next section. Figure 9 shows three renderings of water surfaces, varying the size of the grid numbers M and N and making the facet sizes dx and dz proportional to $1/M$ and $1/N$. So as we go from the top image to the bottom, the facet sizes become smaller, and we see the effect of increasing amount of detail in the renderings. Clearly, more wave detail helps to build a realistic-looking surface.

As a final example, figure 10 is an image rendered in the commercial package RenderWorld by Arete Entertainment. This rendering includes the effect of an atmosphere, and water volume scattered light. These are discussed in the next section. But clearly, wave height fields generated from random numbers using an fft prescription can produce some nice images.

3.6 Experimental Evidence

This subsection is a diversion away from the main graphics thrust of these notes. It is an account of a relatively simple remote sensing experiment that demonstrates that the algorithms discussed in this section are grounded (wetted ?) in reality. Figure 11 is a frame from a video segment showing water coming into the beach near Zuma Beach, California. The video camera was located on hill overlooking the beach, in 1986. In 1993, the region of video frames indicated in the figure was digitized, to produce a time series of frames containing just water surface.

The analysis of the image data has a statistical character. Recall that the Fourier Transform method of creating surface simulations arises from statistical arguments about the averaged structure of the surface. Recall also that, to some degree, the grayscale display of

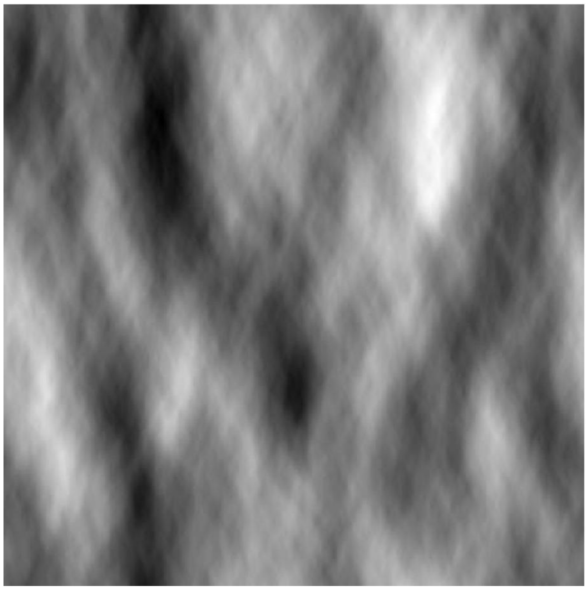

Figure 8: Wave height realization with increased directional dependence.

the slope field is reminiscent of an overhead view of a water surface. Our data analysis here attempts to use these assumptions to compute statistical quantities of the images, that should be related to the wave statistical properties.

From the multiple frames, a three dimensional Power Spectral Density (PSD) was created. The PSD is computed from the images by a two step processes (1) Fourier Transform the images in space and time to create the quantity

$$\tilde{h}(\mathbf{k},\omega) = \int d^2x\; dt\; h(\mathbf{x},t) \exp\left\{-i\mathbf{k}\cdot\mathbf{x} + i\omega t\right\} \ , \tag{27}$$

and (2) form the estimated PSD by smoothing the absolute square of $\tilde{h}$. In mathematical notation, this is

$$PSD(\mathbf{k},\omega) = \overline{\left|\tilde{h}(\mathbf{k},\omega)\right|^2} \tag{28}$$

There is a good reason for creating the 3D PSD as defined here. If the waves animate in time as prescribed in equation 26, that is, with the dispersion relationship $\omega = \omega(k)$, then the PSD will have a large value in the regions where $\mathbf{k}$ and ω satisfy that dispersion relationship, and a smaller value anywhere else.

Figure 12 shows a plot of the dispersion relationship in equation 14. There are two branches, for $+\sqrt{gk}$ and $-\sqrt{gk}$. If the waves in the image data are animating with that dispersion relation, then the PSD will show most of its strength along these two branches, although the actual magnitude of the PSD should vary from point to point. Figure 13 shows the actual 3D PSD from the image data. There are two clear branches along the dispersion relationship we have discussed, with no apparent modification by shallow water affects. There is also a third branch that is approximately a straight line lying between the first two. Examination of the video shows that this branch comes from a surfactant layer floating on the water in part of the video frame, and moving with a constant speed. Excluding the surface layer, this data clearly demonstrates the validity of the dispersion relationship, and demonstrates the usefulness of the statistical, Fourier Transform oriented model of ocean waves.

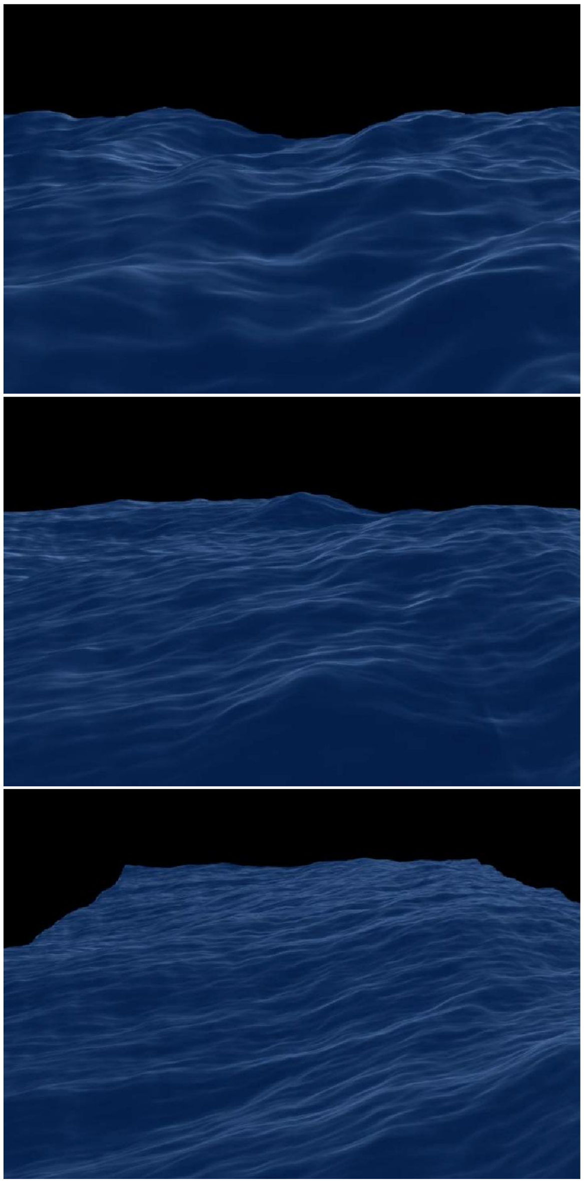

Figure 9: Rendering of waves with (top) a fairly low number of waves (facet size = 10 cm), with little detail; (middle) a reasonably good number of waves (facet size = 5 cm); (bottom) a high number of waves with the most detail (facet size = 2.5 cm).

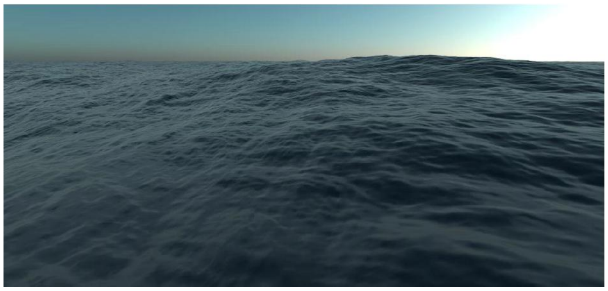

Figure 10: An image of a wave height field rendered in a commercial package with a model atmosphere and sophisticated shading.

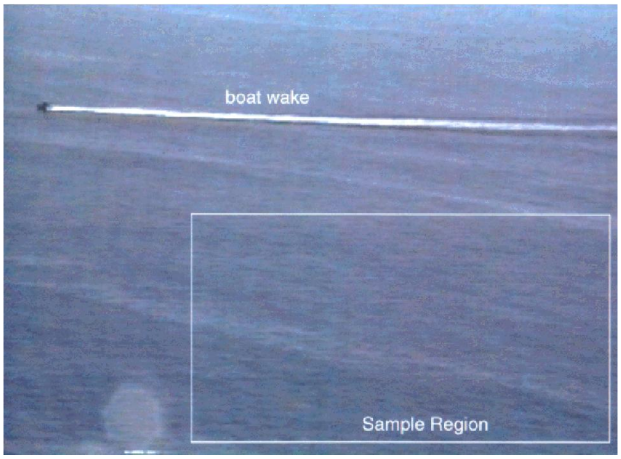

Figure 11: Site at which video data was collected in 1986, near Zuma Beach, California.

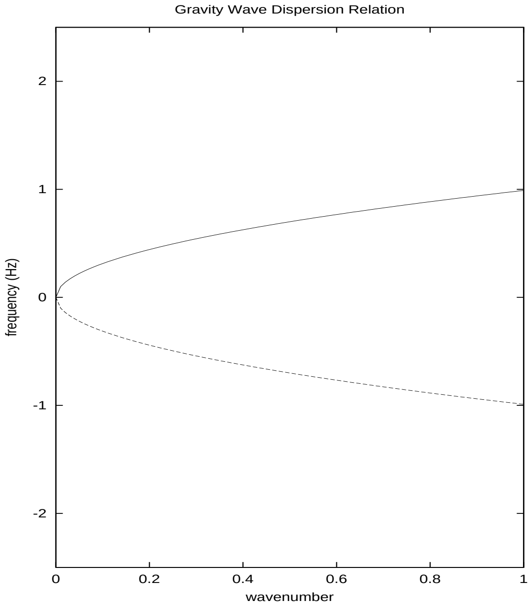

Figure 12: The theoretical dispersion relationship for deep water gravity waves.

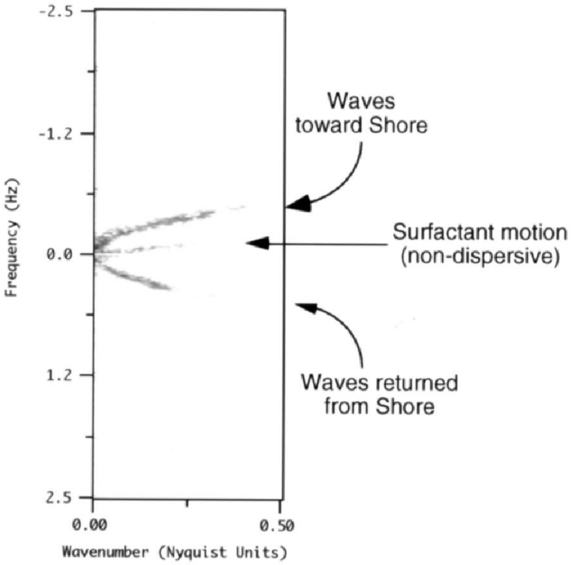

Figure 13: Slice from a 3D Power Spectral Density grayscale plot, from processed video data.

3.7 Choppy Waves

We turn briefly in this section to the subject of creating choppy looking waves. The waves produced by the fft methods presented up to this point have rounded peaks and troughs that give them the appearance of fair-weather conditions. Even in fairly good weather, and particularly in a good wind or storm, the waves are sharply peaked at their tops, and flattened at the bottoms. The extent of this chopping of the wave profile depends on the environmental conditions, the wavelengths and heights of the waves. Waves that are sufficiently high (e.g. with a slope greater than about 1/6) eventually break at the top, generating a new set of physical phenonema in foam, splash, bubbles, and spray.

The starting point for this method is the fundamental fluid dynamic equations of motion for the surface. These equations are expressed in terms of two dynamical fields: the surface elevation and the velocity potential on the surface, and derive from the Navier-Stokes description of the fluid throughout the volume of the water and air, including both above and below the interface. Creamer et al[12] set out to apply a mathematical approach called the ”Lie Transform technique” to generate a sequence of ”canonical transformations” of the elevation and velocity potential. The benefit of this complex mathematical procedure is to convert the elevation and velocity potential into new dynamical fields that have a simpler dynamics. The transformed case is in fact just the simple ocean height field we have been discussing, including evolution with the same dispersion relation we have been using in this paper. Starting from there, the inverse Lie Transform in principle converts our phenomenological solution into a dynamically more accurate one. However, the Lie Transform is difficult to manipulate in 3 dimensions, while in two dimensions exact results have been obtained. Based on those exact results in two dimensions, an extrapolation for the form of the 3D solution has been proposed: a horizontal displacement of the waves, with the displacement locally varying with the waves.

In the fft representation, the 2D displacement vector field is com-

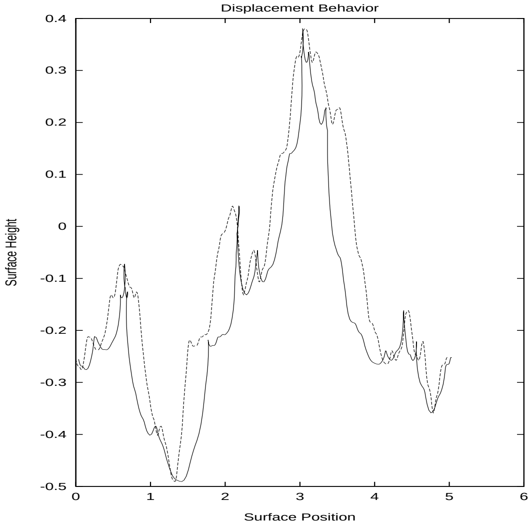

Figure 14: A comparison of a wave height profile with and without the displacement. The dashed curve is the wave height produced by the fft representation. The solid curve is the height field displaced using equation 29.

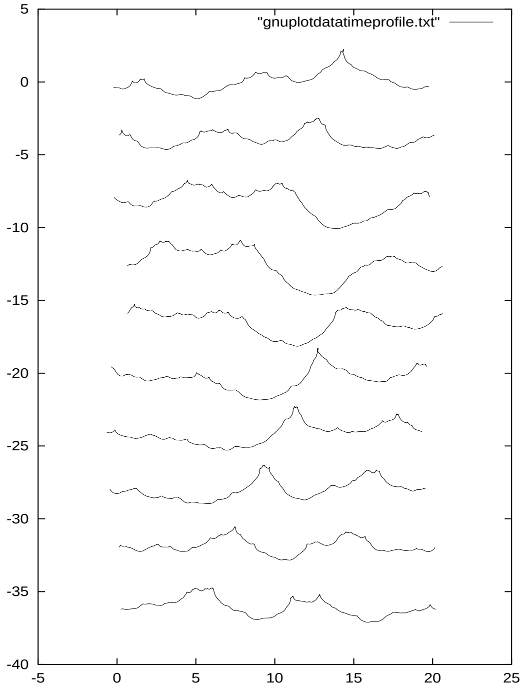

Figure 15: A time sequence of profiles of a wave surface. From top to bottom, the time between profiles is 0.5 seconds.

puted using the Fourier amplitudes of the height field, as

$$\mathbf{D}(\mathbf{x},t) = \sum_{\mathbf{k}} -i\frac{\mathbf{k}}{k}\ \tilde{h}(\mathbf{k},t)\ \exp\left(i\mathbf{k}\cdot\mathbf{x}\right) \tag{29}$$

Using this vector field, the horizontal position of a grid point of the surface is now $\mathbf{x} + \lambda\mathbf{D}(\mathbf{x},t)$, with height $h(\mathbf{x})$ as before. The parameter λ is not part of the original conjecture, but is a convenient method of scaling the importance of the displacement vector. This conjectured solution does not alter the wave heights directly, but instead warps the horizontal positions of the surface points in a way that depends on the spatial structure of the height field. The particular form of this warping however, actually sharpens peaks in the height field and broadens valleys, which is the kind of nonlinear behavior that should make the fft representation more realistic. Figure 14 shows a profile of the wave height along one direction in a simulated surface. This clearly shows that the “displacement conjecture” can dramatically alter the surface.

The displacment form of the this solution is similar to the algorithm for building Gerstner waves [8] discussed in section 3. In that case however, the displacement behavior, applied to sinusoid shapes, was the principle method of characterizing the water surface structure, and here it is a modifier to an already useful wave height representation.

Figure 15 illustrates how these choppy waves behave as they evolve. The tops of waves form a sharp cusp, which rounds out and disappears shortly afterward.

One ”problem” with this method of generating choppy waves can be seen in figure 14. Near the tops of some of the waves, the surface actally passes through itself and inverts, so that the outward normal to the surface points inward. This is because the amplitudes of the wave components can be large enough to create large displacements that overlap. This is easily defeated simply by reducing the magnitude of the scaling factor λ. For the purposes of computer graphics, this might actually be a useful effect to signal the production of spray, foam and/or breaking waves. We will not discuss here how to carry out such an extension, except to note that in order to use this region of overlap, a simple and quick test is needed for deciding that the effect is taking place. Fortunately, there is such a simple test in the form of the Jacobian of the transformation from $\mathbf{x}$ to $\mathbf{x} + \lambda\mathbf{D}(\mathbf{x},t)$. The Jacobian is a measure of the uniqueness of the transformation. When the displacement is zero, the Jacobian is 1. When there is displacement, the Jacobian has the form

$$J(\mathbf{x}) = J_{xx}J_{yy} - J_{xy}J_{yx}\ , \tag{30}$$

with individual terms

$$\begin{aligned} J_{xx}(\mathbf{x}) &= 1 + \lambda\frac{\partial D_x(\mathbf{x})}{\partial x} \\ J_{yy}(\mathbf{x}) &= 1 + \lambda\frac{\partial D_y(\mathbf{x})}{\partial y} \\ J_{yx}(\mathbf{x}) &= \lambda\frac{\partial D_y(\mathbf{x})}{\partial x} \\ J_{xy}(\mathbf{x}) &= \lambda\frac{\partial D_x(\mathbf{x})}{\partial y} = J_{yx} \end{aligned}$$

and $\mathbf{D} = (D_x, D_y)$. The Jacobian signals the presence of the overlapping wave bacause its value is less than zero in the overlap region. For example, figure 16 plots a profile of a basic wave without displacement, the wave with displacement, and the value of J for the choppy wave (labeled ”Folding Map”). The ”folds” or overlaps in the choppy surface are clearly visible, and align with the regions in which $J < 0$. With this information, it should be relatively easy to extract the overlapping region and use it for other purposes if desired.

But there is more that can be learned from these folded waves from a closer examination of this folding criterion. The Jacobian

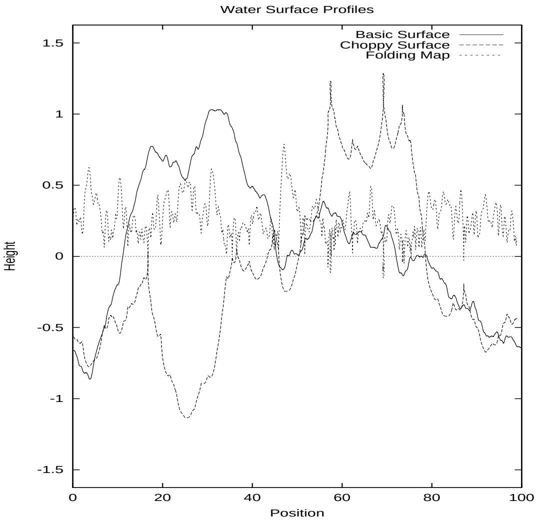

Figure 16: Wave height profile with and without the displacement. Also plotted is the Jacobian map for choppy wave profile.

derives from a 2×2 matrix which measures the local uniqueness of the choppy wave map $\mathbf{x} \longrightarrow \mathbf{x} + \lambda \mathbf{D}$. This matrix can in general be written in terms of eigenvectors and eigenvalues as:

$$J_{ab} = J_- \hat{e}^-_a \hat{e}^-_b + J_+ \hat{e}^+_a \hat{e}^+_b \ , \ \ (a, b = x, y) \qquad (31)$$

where J_- and J_+ are the two eigenvalues of the matrix, ordered so that $J_- \leq J_+$. The corresponding orthonormal eigenvectors are $\hat{e}^-$ and $\hat{e}^+$ respectively. From this expression, the Jacobian is just $J = J_- J_+$.

The criterion for folding that $J < 0$ means that $J_- < 0$ and $J_+ > 0$. So the minimum eigenvalue is the actual signal of the onset of folding. Further, the eigenvector $\hat{e}^-$ points in the horizontal direction in which the folding is taking place. So, the prescription now is to watch the minimum eigenvalue for when it becomes negative, and the alignment of the folded wave is parallel to the minimum eigenvector.

We can illustrate this phenomenon with an example. Figures 17 and 18 show two images of an ocean surface, one without choppy waves, and the other with the choppy waves strongly applied. These two surfaces are identical except for the choppy wave algorithm. Figure 19 shows the wave profiles of both surfaces along a slice through the surfaces. Finally, the profile of the choppy wave is plotted together with the value of the minimum eigenvalue in figure 20, showing the clear connection between folding and the negative value of J_-.

Incidentally, computing the eigenvalues and eigenvectors of this matrix is fast because they have analytic expressions as

$$J_\pm = \frac{1}{2}(J_{xx} + J_{yy}) \pm \frac{1}{2} \left\{ (J_{xx} - J_{yy})^2 + 4 J_{xy}^2 \right\}^{1/2} \qquad (32)$$

for the eigenvalues and

$$\hat{e}^\pm = \frac{(1, q_\pm)}{\sqrt{1 + q_\pm^2}} \qquad (33)$$

with

$$q_\pm = \frac{J_\pm - J_{xx}}{J_{xy}} \qquad (34)$$

for the eigenvectors.

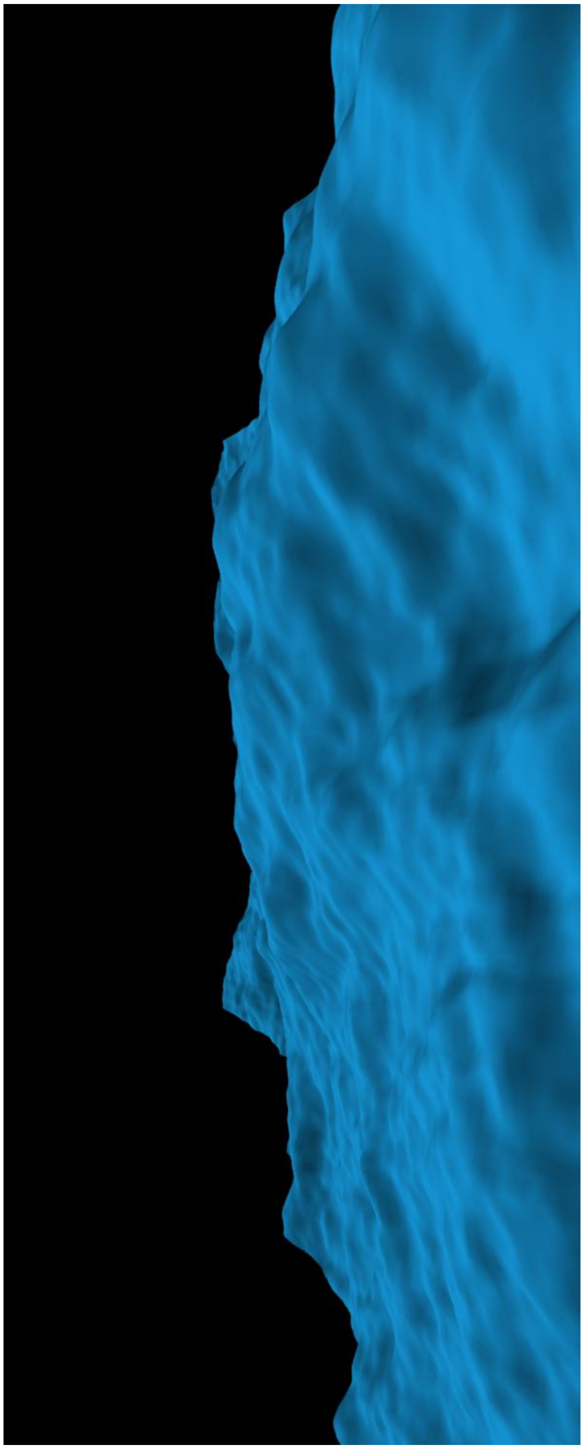

Figure 17: Simulated wave surface without the choppy algorithm applied. Rendered in BMRT with a generic plastic shader.

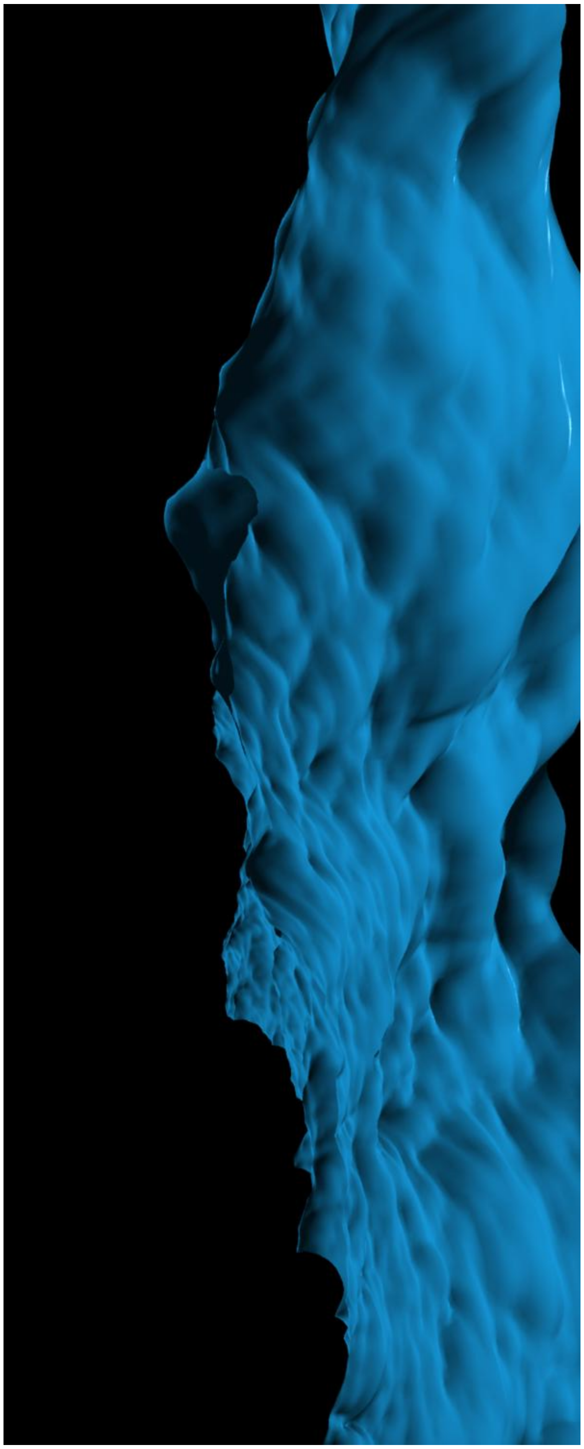

Figure 18: Same wave surface with strong chop applied. Rendered in BMRT with a generic plastic shader.

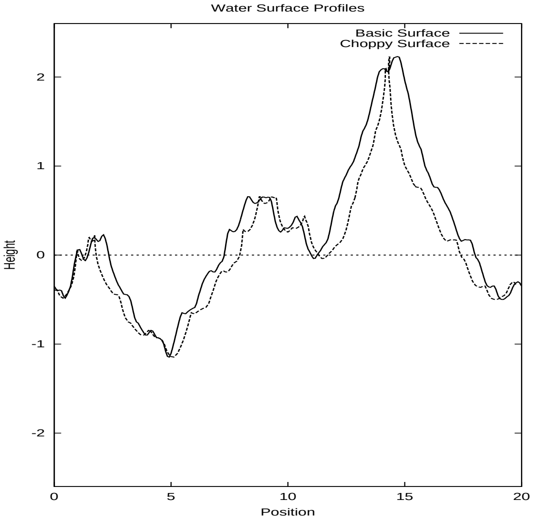

Figure 19: Profiles of the two surfaces, showing the effect of the choppy mapping.

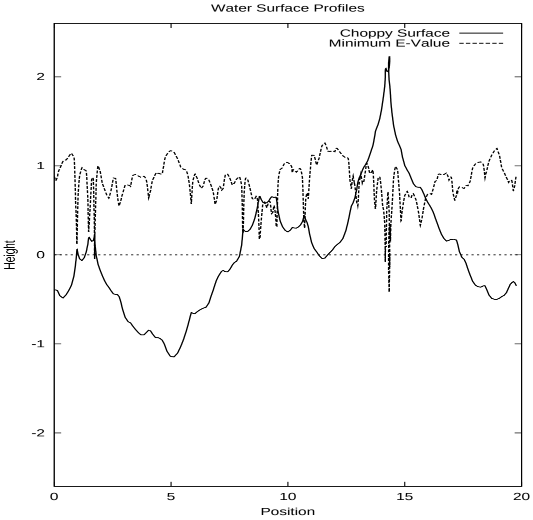

Figure 20: Plot of the choppy surface profile and the minimum eigenvalue. The locations of folds of the surface are clearly the same as where the eigenvalue is negative.

4 Surface Wave Optics

The optical behavior of the ocean surface is fairly well understood, at least for the kinds of quiescent wave structure that we consider in these notes. Fundamentally, the ocean surface is a near perfect specular reflector, with well-understand relectivity and transmissity functions. In this section these properties are summarized, and combined into a simple shader for Renderman. There are circumstances when the surface does not appear to be a specular reflector. In particular, direct sunlight reflected from waves at a large distance from the camera appear to be spread out and made diffuse. This is due to the collection of waves that are smaller than the camera can resolve at large distances. The mechanism is somewhat similar to the underlying microscopic reflection mechnanisms in solid surfaces that lead to the Torrance-Sparrow model of BRDFs. Although the study of glitter patterns in the ocean was pioneered by Cox and Munk many years ago, the first models of this BRDF behavior that I am aware of were developed in the early 1980's. At the end of this section, we introduce the concepts and conditions, state the results, and ignore the in-between analysis and derivation.

Throughout these notes, and particularly in this section, we ignore one optical phenomenon completely: polarization. Polarization effects can be strong at a boundary interface like a water surface. However, since most of computer graphics under consideration ignores polarization, we will continue in that long tradition. Of course, interested readers can find literature on polarization effects at the air-water interface.

4.1 Specular Reflection and Transmission

Rays of light incident from above or below at the air-water interface are split into two components: a transmitted ray continuing through the interface at a refracted angle, and a reflected ray. The intensity of each of these two rays is diminished by reflectivity and transmissivity coefficients. Here we discussed the directions of the two outgoing rays. In the next subsection the coefficients are discussed.

4.1.1 Reflection

As is well known, in a perfect specular reflection the reflected ray and the incident ray have the same angle with respect to the surface normal. This is true for all specular reflections (ignoring roughening effects), regardless of the material. We build here a compact expression for the outgoing reflected ray. First, we need to build up some notation and geometric quantities.

The three-dimensional points on the ocean surface can be labelled by the horizontal position $\mathbf{x}$ and the waveheight $h(\mathbf{x}, t)$ as

$$\mathbf{r}(\mathbf{x}, t) = \mathbf{x} + \hat{\mathbf{y}} h(\mathbf{x}, t) \ , \tag{35}$$

where $\hat{\mathbf{y}}$ is the unit vector pointing straight up. At the point $\mathbf{r}$, the normal to the surface is computed directly from the surface slope $\epsilon(\mathbf{x}, t) \equiv \nabla h(\mathbf{x}, t)$ as

$$\hat{\mathbf{n}}_S(\mathbf{x}, t) = \frac{\hat{\mathbf{y}} - \epsilon(\mathbf{x}, t)}{\sqrt{1 + \epsilon^2(\mathbf{x}, t)}} \tag{36}$$

For a ray intersecting the surface at $\mathbf{r}$ from direction $\hat{\mathbf{n}}_i$, the direction of the reflected ray can depend only on the incident direction and the surface normal. Also, as mentioned before, the angle between the surface normal and the reflected ray must be the same as the angle between incident ray and the surface normal. You can verify for yourself that the reflected direction $\hat{\mathbf{n}}_r$ is

$$\hat{\mathbf{n}}_r(\mathbf{x}, t) = \hat{\mathbf{n}}_i - 2\hat{\mathbf{n}}_S(\mathbf{x}, t) \left(\hat{\mathbf{n}}_S(\mathbf{x}, t) \cdot \hat{\mathbf{n}}_i\right) \ . \tag{37}$$

Note that this expression is valid for incident ray directions on either side of the surface.

4.1.2 Transmission

Unfortunately, the direction of the transmitted ray is not expressed as simply as for the reflected ray. In this case we have two guiding principles: the transmitted direction is dependent only on the surface normal and incident directions, and Snell's Law relating the sines of the angles of the incident and transmitted angles to the indices of refraction of the two materials.

Suppose the incident ray is coming from one of the two media with index of refraction n_i (for air, $n = 1$, for water, $n = 4/3$ approximately), and the transmitted ray is in the medium with index of refraction n_r. For the incident ray at angle θ_i to the normal,

$$\sin\theta_i = \sqrt{1 - (\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_S)^2} = |\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_S| \tag{38}$$

the transmitted ray will be at an angle θ_t with

$$\sin\theta_t = |\hat{\mathbf{n}}_t \times \hat{\mathbf{n}}_S| \ . \tag{39}$$

Snell's Law states that these two angles are related by

$$n_i \sin\theta_i = n_t \sin\theta_t \ . \tag{40}$$

We now have all the pieces needed to derive the direction of transmission. The direction vector can only be a linear combination of $\hat{\mathbf{n}}_i$ and $\hat{\mathbf{n}}_S$. It must satisfy Snell's Law, and it must be a unit vector (by definition). This is adequate to obtain the expression

$$\hat{\mathbf{n}}_t(\mathbf{x}, t) = \frac{n_i}{n_t} \hat{\mathbf{n}}_i + \Gamma(\mathbf{x}, t)\, \hat{\mathbf{n}}_S(\mathbf{x}, t) \tag{41}$$

with the function Γ defined as

$$\begin{aligned} \Gamma(\mathbf{x}, t) \quad &\equiv \quad \frac{n_i}{n_t} \hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_S(\mathbf{x}, t) \\ &\pm \quad \left\{ 1 - \left(\frac{n_i}{n_t}\right)^2 |\hat{\mathbf{n}}_i \times \hat{\mathbf{n}}_S(\mathbf{x}, t)|^2 \right\}^{1/2} \ . \end{aligned} \tag{42}$$

The plus sign is used in Γ when $\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_S < 0$, and the minus sign is used when $\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_S > 0$.

4.2 Fresnel Reflectivity and Transmissivity

Accompanying the process of reflection and transmission through the interface is a pair of coefficients that describe their efficiency. The reflectivity R and transmissivity T are related by the constraint that no light is lost at the interface. This leads to the relationship

$$R + T = 1 \ . \tag{43}$$

The derivation of the expressions for R and T is based on the electromagnetic theory of dielectrics. We will not carry out the derivations, but merely write down the solution

$$R(\hat{\mathbf{n}}_i, \hat{\mathbf{n}}_r) = \frac{1}{2} \left\{ \frac{\sin^2(\theta_t - \theta_i)}{\sin^2(\theta_t + \theta_i)} + \frac{\tan^2(\theta_t - \theta_i)}{\tan^2(\theta_t + \theta_i)} \right\} \tag{44}$$

Figure 21 is a plot of the reflectivity for rays of light traveling down onto a water surface as a function of the angle of incidence to the surface. The plot extends from a grazing angle of 0 degrees to perpendicular incidence at 90 degrees. As should be clear, variation of the reflectivity across an image is an important source of the "texture" or feel of water. Notice that reflectivity is a function of the angle of incidence relative to the wave normal, which in turn is directly related to the slope of the surface. So we can expect that a strong contributor to the texture of water surface is the pattern of slope, while variation of the wave height serves primarily as a wave hiding mechanism. This is the quantitative explanation of why the

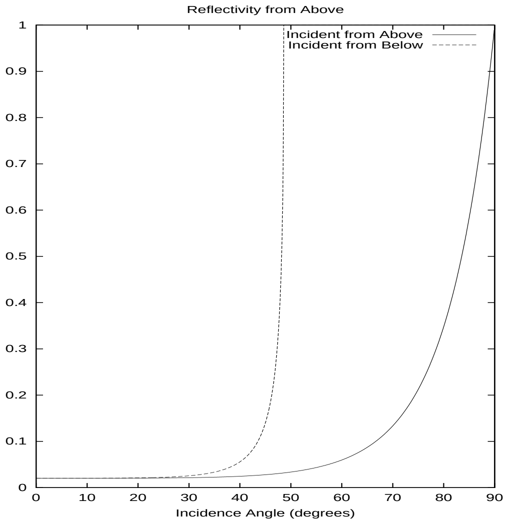

Figure 21: Reflectivity for light coming from the air down to the water surface, as a function of the angle of incidence of the light.

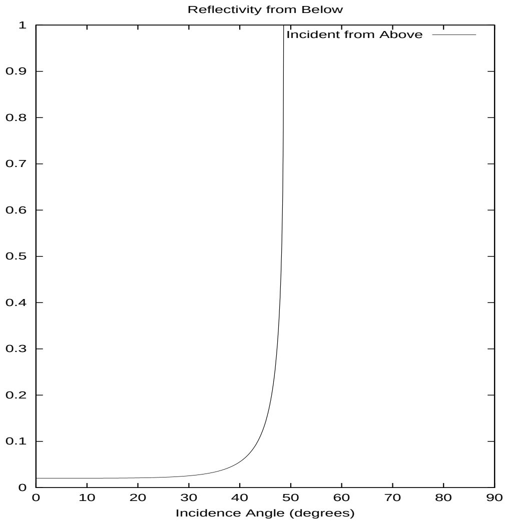

Figure 22: Reflectivity for light coming from below the water surface, as a function of the angle of incidence of the light.

surface slope more closely resembles rendered water than the wave height does, as we saw in the previous section when discussing figure 7.

When the incident ray comes from below the water surface, there are important differences in the reflectivity and transmissivity. Figure 22 shows the reflectivity as a function of incidence angle again, but this time for incident light from below. In this case, the reflectivity reaches unity at a fairly large angle, near 41 degrees. At incidence angles below that, the reflectivity is one and so there is no transmission of light through the interface. This phenomenon is *total internal reflection*, and can be seen just by swimming around in a pool. The angle at which total internal reflection begins is called Brewster's angle, and is given by, from Snell's Law,

$$\sin\theta_i^B = \frac{n_t}{n_i} = 0.75 \tag{45}$$

or $\theta_i^B = 48.6\,\text{deg}$. In our plots, this angle is $90 - \theta_i^B = 41.1\,\text{deg}$.

4.3 Building a Shader for Renderman

From the discussion so far, one of the most important features a rendering must emulate is the textures of the surface due to the strong slope-dependence of reflectivity and transmissivity. In this section we construct a simple Renderman-compliant shader using just these features. Readers who have experience with shaders will know how to extend this one immediately.

The shader exploits that fact that the Renderman interface already provides a built-in Fresnel quantity calculator, which provides R, T, $\hat{\mathbf{n}}_r$, and $\hat{\mathbf{n}}_t$ using the surface normal, incident direction vector, and index of refraction. The shader for the air-to-water case is as follows:

```
surface watercolorshader(
    color upwelling = color(0, 0.2, 0.3);
    color sky       = color(0.69,0.84,1);
    color air       = color(0.1,0.1,0.1);
    float nSnell    = 1.34;
    float Kdiffuse  = 0.91;
    string envmap   = "";
                        )
{
  float reflectivity;
  vector nI = normalize(I);
  vector nN = normalize(Ng);
  float costhetai = abs(nI . nN);
  float thetai = acos(costhetai);
  float sinthetat = sin(thetai)/nSnell;
  float thetat = asin(sinthetat);
  if(thetai == 0.0)
  {
    reflectivity = (nSnell - 1)/(nSnell + 1);
    reflectivity = reflectivity * reflectivity;
  }
  else
  {
    float fs = sin(thetat - thetai)
             / sin(thetat + thetai);
    float ts = tan(thetat - thetai)
             / tan(thetat + thetai);
    reflectivity = 0.5 * ( fs*fs + ts*ts );
  }
  vector dPE = P-E;
  float dist = length(dPE) * Kdiffuse;
  dist = exp(-dist);

  if(envmap != "")
  {
    sky = color environment(envmap, nN);
  }
  Ci =  dist * ( reflectivity * sky
               + (1-reflectivity) * upwelling )
     +  (1-dist)* air;
}
```

There are two contributions to the color: light coming downward onto the surface with the default color of the sky, and light coming upward from the depths with a default color. This second term will be discussed in the next section. It is important for incidence angles that are high in the sky, because the reflectivity is low and transmissivity is high.

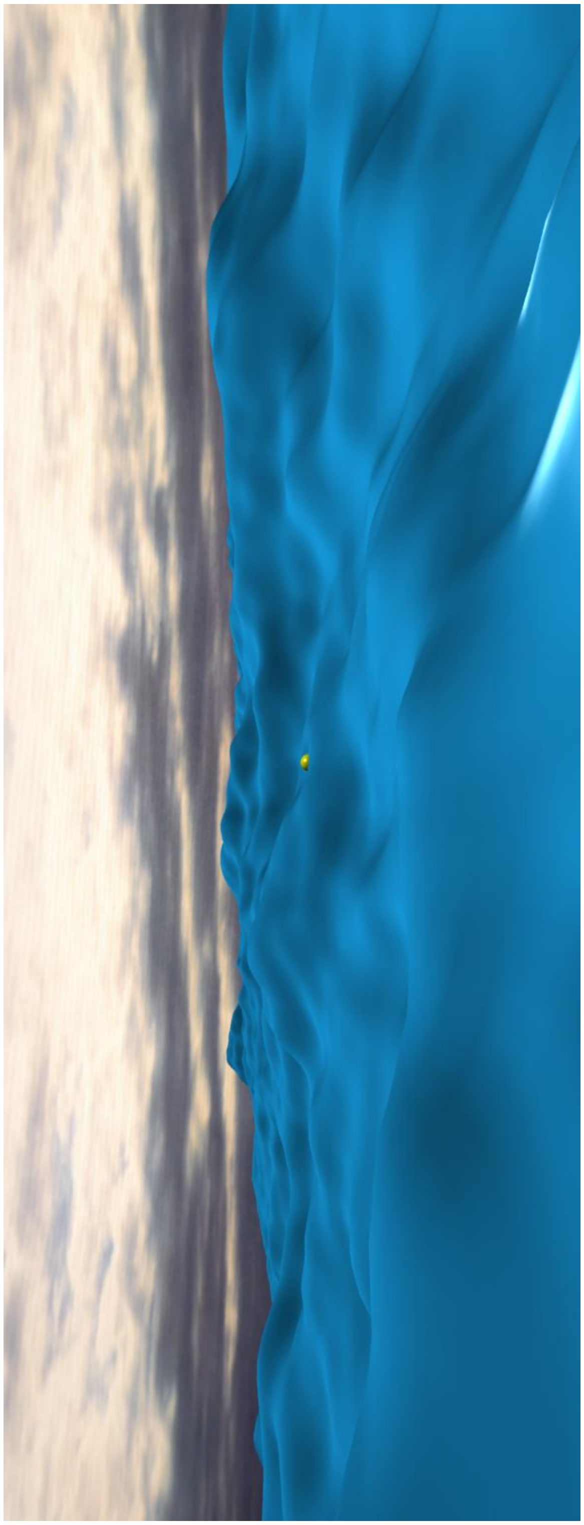

Figure 23: Simulated water surface with a generic plastic surface shader. Rendered with BMRT.

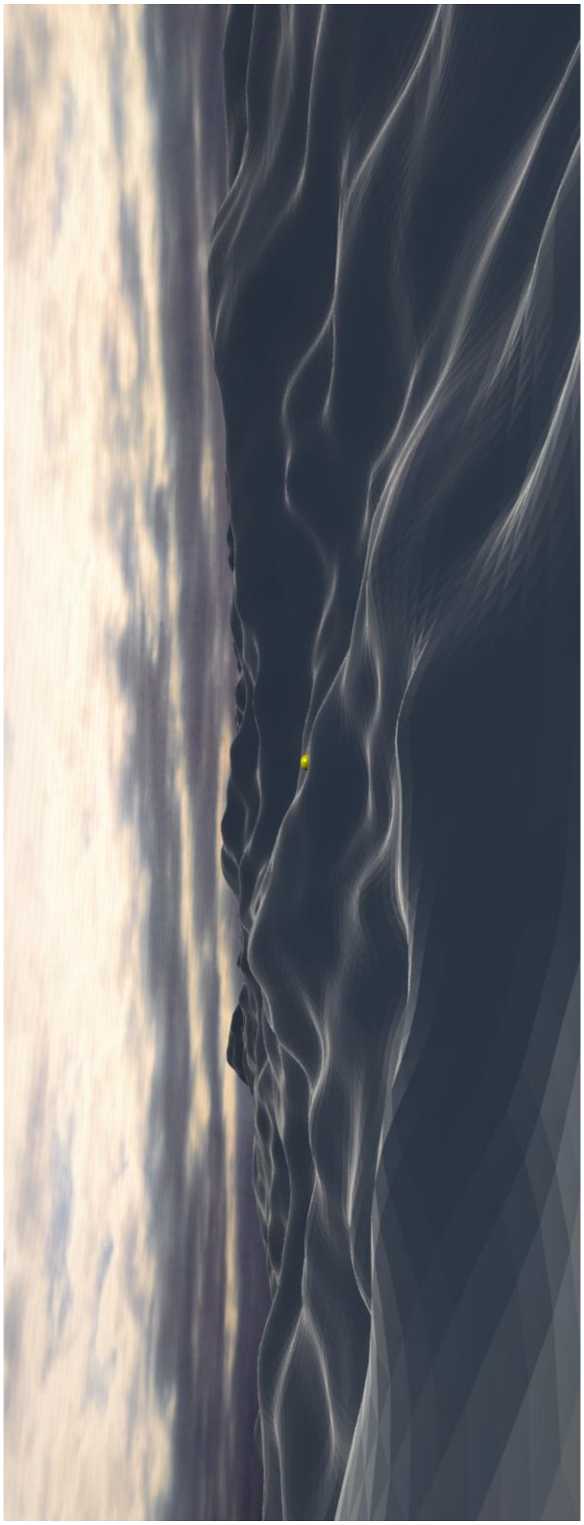

Figure 24: Simulated water surface with a realistic surface shader. Rendered with BMRT.

This shader was used to render the image in figure 24 using the BMRT raytrace renderer. For reference, the exact same image has been rendered in 23 with a generic plastic shader. Note that the realistic water shader tends to highlight the tops of the the waves, where the angle of incidence is nearly 90 degrees grazing and the reflectivity is high, while the sides of the waves are dark, where angle of incidence is nearly 0 that the reflectivity is low.

5 Water Volume Effects

The previous section was devoted to a discussion of the optical behavior of the surface of the ocean. In this section we focus on the optical behavior of the water volume below the surface. We begin with a discussion of the major optical effects the water volume has on light, followed by an introduction to color models researchers have built to try to connect the ocean color on any given day to underlying biological and physical processes. These models are built upon many years of in-situ measures off of ships and peers. Finally, we discuss two important effects, caustics and sunbeams, that sometimes are hard to grasp, and which produce beautiful images when properly simulated.

5.1 Scattering, Transmission, and Reflection by the Water Volume

In the open ocean, light is both scattering and absorbed by the volume of the water. The sources for these events are of three types: water molecules, living and dead organic matter, and non-organic matter. In most oceans around the world, away from the shore lines, absorption is a fairly even mixture of water molecules and organic matter. Scattering is dominated by organic matter however.

To simulate the processes of volumetric absorption and scattering, there are five quantities that are of interest: absorption coefficient, scattering coefficient, extinction coefficient, diffuse extinction coefficient, and bulk reflectivity. All of these coefficients have units of inverse length, and represent the exponential rate of attenuation of light with distance through the medium. The absorption coefficient a is the rate of absorption of light with distance, the scattering coefficient b is the rate of scattering with length, the extinction coefficient c is the sum of the two previous ones $c = a + b$, and the diffuse extinction coefficient K describes the rate of loss of intensity of light with distance after taking into account both absorption and scattering processes. The connection between K and the other parameters is not completely understood, in part because there are a variety of ways to define K in terms of operational measurements. Different ways change the details of the dependence. However, there is a condition called the *asymptotic* limit at very deep depths in the water, at which all operational definitions of K converge to a single value. This asymptotic value of K has been modeled in a variety of ways. There is a mathematically precise result that the ratio K/c depends only on b/c, the single scatter albedo, and some details of the angular distribution of individual scattering distributions. Figure 25 is an example of a model of K/c for reasonable water conditions. Models have been generated for the color dependence of K, most notably by Jerlov. In 1990, Austin and Petzold performed a revised analysis of spectral models, including new data, to produce refined models of K as a function of color. For typical visible light conditions in the ocean, K ranges in value from 0.03/meter to 0.1/meter. It is generally true that $a < K < c$.

One way to interpret these quantities for a simulation of water volume effects is as follows:

1. A ray of sunlight enters the water with intensity I (after losing some intensity to Fresnel transmission). Along a path underwater of a length s, the intensity at the end of the path is $I\exp(-cs)$, i.e. the ray of direct sunlight is attenuated as fast a possible.

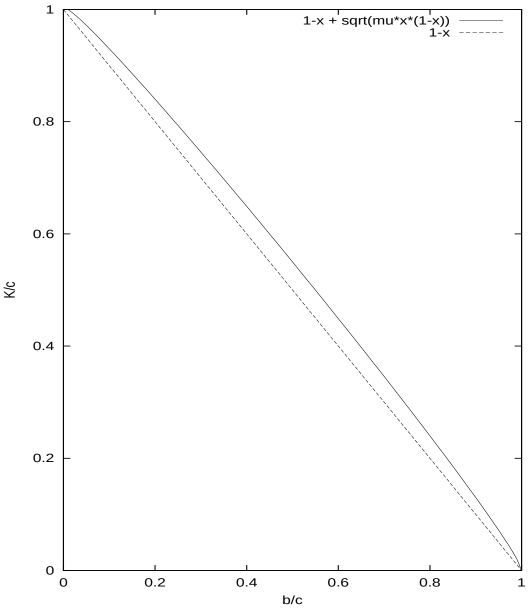

Figure 25: Dependence of the Diffuse Extinction Coefficient on the Single Scatter Albedo, normalized to the extinction.

2. Along the path through the water, a fraction of the ray is scattered into a distribution of directions. The strength of the scattering per unit length of the ray is b, so the intensity is proportional to $bI\exp(-cs)$.

3. The light that is scattered out of the ray goes through potentially many more scattering events. It would be nearly impossible to track all of them. However, the sum whole outcome of this process is to attenuate the ray along the path from the original path to the camera as $bI\exp(-cs)\exp(-Ks_c)$, where s_c is the distance from the scatter point in the ocean to the camera.

A fifth quantity of interest for simulation is the bulk reflectivity of the water volume. This is a quantity that is intended to allow us to ignore the details of what is going on, treat the volume as a Lambertian reflector, and compute a value for bulk reflectivity. That number is sensitive to many factors, including wave surface conditions, sun angle, water optical properties, and details of the angular spread. Nevertheless, values of reflectivity around 0.04 seem to agree well with data.

5.2 The Underwater POV: Refracted Skylight, Caustics, and Sunbeams

Now that we have underwater optical properties at hand, we can look at two important phenomena in the ocean: caustics and sunbeams.

5.2.1 Caustics

Caustics, in this context, are a light pattern that is formed on surfaces underwater. Because the water surface is not flat, groups of

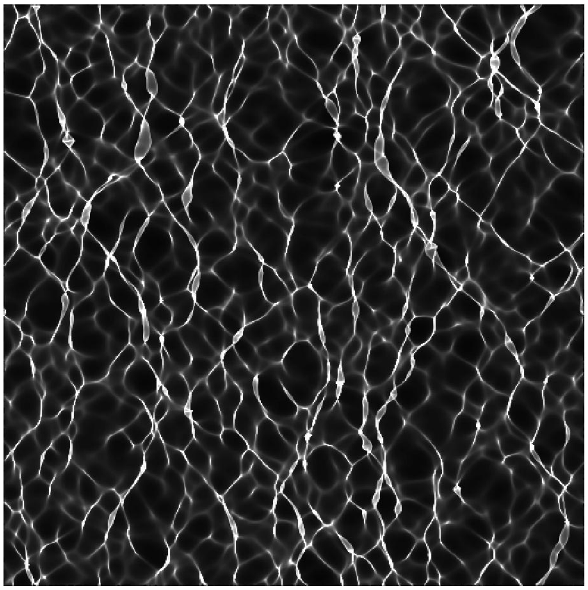

Figure 26: Rendering of a caustic pattern at a shallow depth (5 meters) below the surface.

light rays incident on the surface are either focussed or defocussed. As a result, a point on a fictitious plane some depth below the ocean surface receives direct sunlight from several different positions on the surface. The intensity of light varies due to the depth, original contrast, and other factors. For now, lets write the intensity of the pattern as $I = Ref\ I_0$, with I_0 as the light intensity just above the water surface. The quantity Ref is the scaling factor that varies with position on the fictitious plane due to focussing and defocussing of waves, and is called a *caustic pattern*. Figure 26 shows an example of the caustic pattern Ref. Notice that the caustic pattern exhibits filaments and ring-like structure. At a very deep depth, the caustic pattern is even more striking, as shown in figure 27.

One of the important properties of underwater light that produce caustic patterns is conservation of flux. This is actually a simple idea: suppose a small area on the ocean surface has sunlight passing through it into the water, with intensity I at the surface. As we map that area to greater depths, the amount of area within it grows or shrinks, but most likely grows depending on whether the area is focussed or defocussed. The intensity at any depth within the water is proportional to inverse of the area of the projected region. Another way of saying this is that if a bundle of light rays diverges, their intensities are reduced to keep the product of intensity time area fixed.

Simulated caustic patterns can actually be compared (roughly) with real-world data. In a series of papers published throughout the 1970's, 1980's, and into the 1990's, Dera and others collected high-speed time series of light intensity[17]. As part of this data collection and analysis project, the data was used to generate a probability distribution function (PDF) for the light intensity. Figure 28 shows two PDFs taken from one of Dera's papers. The two PDF's were collected for different surface roughness conditions: rougher water tended to suppress more of the high magnitude fluctuations in intensity.

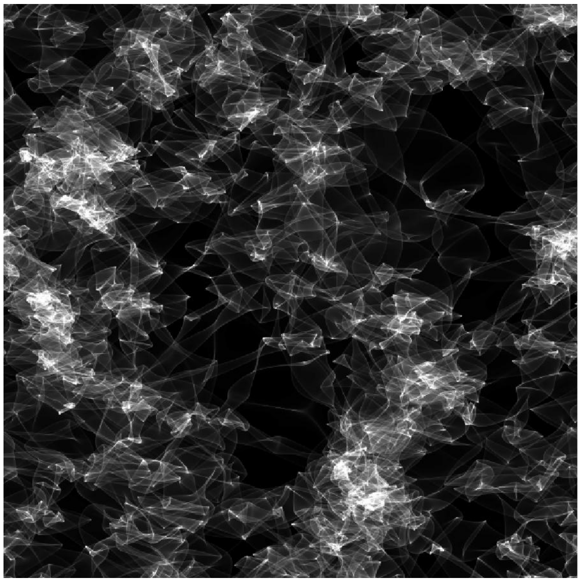

Figure 27: Rendering of a caustic pattern at great depth (100 meters) below the surface.

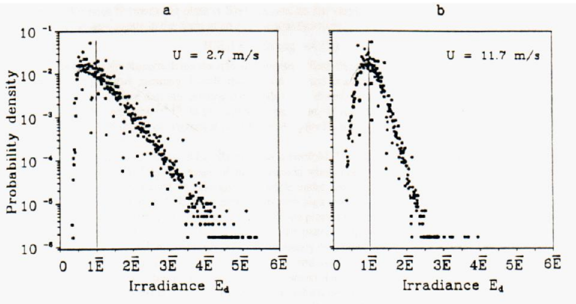

Figure 28: PDF's as measured by Dera in reference [17].

Figure 29 shows the pdf at two depths from a simulation of the ocean surface. These two sets do not match Dera's measurements because of many factors, but most importantly because we have not simulated the environmental conditions and instrumentation in Dera's experiments. Nevertheless, the similarity of figure 29 with Dera's data is an encouraging point of information for the realism of the simulation.

5.2.2 Godrays

Underwater sunbeams, also called godrays, have a very similar origin to caustics. Direct sunlight passes into the water volume, focussed and defocussed at different points across the surface. As the rays of light pass down through the volume, some of the light is scattered in other directions, and a fraction arrives at the camera. The accumulated pattern of scattered light apparent to the camera are the godrays. So, while caustics are the pattern of direct sunlight that penetrates down to the floor of a water volume, sunbeams are scattered light coming from those shafts of direct sunlight in the water volume. Figure 30 demonstrates sunbeams as seen by a camera looking up at it.

References

[1] Jeff Odien, "On the Waterfront", Cinefex, No. 64, p 96, (1995)

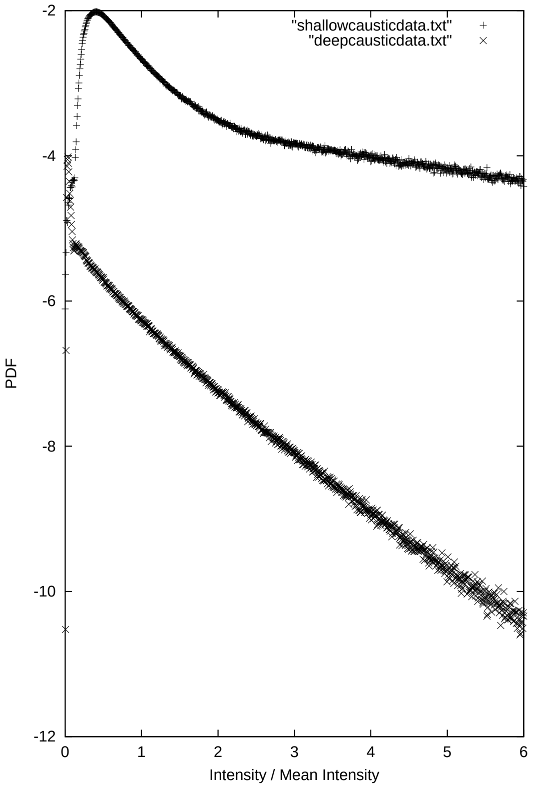

Figure 29: Computed Probability Density Function for light intensity fluctuations in caustics. (upper curve) shallow depth of 2 meters; (lower curve) deep depth of 10 meters.

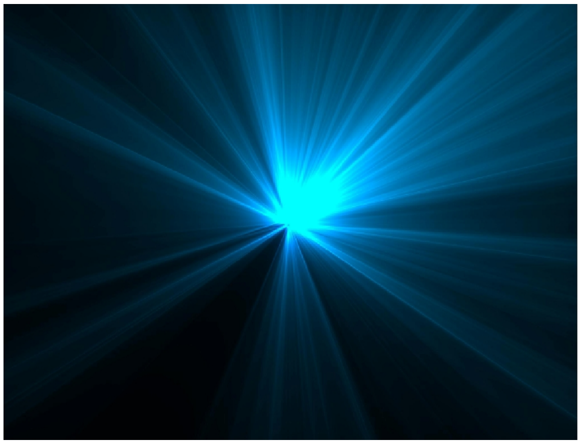

Figure 30: Rendering of sunbeams, or godrays, as seen looking straight up at the light source.

[2] Ted Elrick, "Elemental Images", Cinefex, No. 70, p 114, (1997)

[3] Kevin H. Martin, "Close Contact", Cinefex, No. 71, p 114, (1997)

[4] Don Shay, "Ship of Dreams", Cinefex, No. 72, p 82, (1997)

[5] Kevin H. Martin, "Virus: Building a Better Borg", Cinefex, No. 76, p 55, (1999)

[6] Simon Premože and Michael Ashikhmin, "Rendering Natural Waters," Eighth Pacific Conference on Computer Graphics and Applications, October 2000.

[7] Gary A. Mastin, Peter A. Watterger, and John F. Mareda, "Fourier Synthesis of Ocean Scenes", IEEE CG&A, March 1987, p 16-23.

[8] Alain Fournier and William T. Reeves, "A Simple Model of Ocean Waves", Computer Graphics, Vol. 20, No. 4, 1986, p 75-84.

[9] Darwyn Peachey, "Modeling Waves and Surf", Computer Graphics, Vol. 20, No. 4, 1986, p 65-74.

[10] Blair Kinsman, *Wind Waves, Their Generation and Propagation on the Ocean Surface*, Dover Publications, 1984.

[11] S. Gran, *A Course in Ocean Engineering, Developments in Marine Technology No. 8*, Elsevier Science Publishers B.V. 1992. See also `http://www.dnv.no/ocean/bk/grand.htm`

[12] Dennis B. Creamer, Frank Henyey, Roy Schult, and Jon Wright, "Improved Linear Representation of Ocean Surface Waves." J. Fluid Mech, **205**, pp. 135-161, (1989).

[13] Seibert Q. Duntley, "Light in the Sea," J. Opt. Soc. Am., **53**,2, pg 214-233, 1963.

[14] Curtis D. Mobley, *Light and Water: Radiative Transfer in Natural Waters*, Academic Press, 1994.

[15] *Selected Papers on Multiple Scattering in Plane-Parallel Atmospheres and Oceans: Methods*, ed. by George W. Kattawar, SPIE Milestones Series, **MS 42**, SPIE Opt. Eng. Press., 1991.

[16] R.W. Austin and T. Petzold, "Spectral Dependence of the Diffuse Attenuation Coefficient of Light in Ocean Waters: A Reexamination Using New Data," *Ocean Optics X*, Richard W. Spinrad, ed., SPIE **1302**, 1990.

[17] Jerzy Dera, Slawomir Sagan, Dariusz Stramski, "Focusing of Sunlight by Sea Surface Waves: New Measurement Results from the Black Sea," *Ocean Optics XI*, SPIE Proceedings, 1992.