

Real-Time Simulation of Surface Gravity Ocean Waves Based on the TMA Spectrum

Namkyung Lee¹, Nakhon Baek^{2,*}, and Kwan Woo Ryu¹

¹ Dept. of Computer Engineering, Kyungpook National Univ., Daegu 702-701, Korea
namklee@hotmail.com, kwryu@knu.ac.kr

² School of EECS, Kyungpook National Univ., Daegu 702-701, Korea
oceancru@gmail.com
<http://isaac.knu.ac.kr/~hope/tma.htm>

Abstract. In this paper, we present a real-time method to display ocean surface gravity waves for various computer graphics applications. Starting from a precise surface gravity model in oceanography, we derive its implementation model and our prototype implementation shows more than 50 frames per second on Intel Core2 Duo 2.40GHz PC's. Our major contributions will be the improvement on the expression power of ocean waves and providing more user-controllable parameters for various wave shapes.

Keywords: Computer graphics, Simulation, Ocean wave, TMA.

1 Introduction

Realistic simulation of natural phenomena is one of the interesting and important issues in computer graphics related areas including computer games and animations. In this paper, we are focusing on the ocean waves, for which we have many research results but not a complete solution yet[1].

Waves on the surface of the ocean are primarily generated by winds and gravity. Although the ocean wave includes internal waves, tides, edge waves and others, it is clear that we should display at least the surface gravity waves on the computer screen, to finally represent the ocean. In oceanography, there are many research results to mathematically model the surface waves in the ocean. Simple sinusoidal or trochoidal expressions can approximate a simple ocean wave. Real world waves are a comprised form of these simple waves, and called wave trains.

In computer graphics, we can classify the related results into two categories. The first one uses fluid dynamics equations in a similar way used in the scientific simulation field. We have a number of results with the capability of obtaining realistic animations of complex water surfaces[2,3,4,5,6]. However, these results are hard to apply to large scenes of waters such as oceans, mainly due to their heavy computation.

The other category is based on the ocean wave models from the oceanography, and consists of three approaches. The first group uses the Gerstner swell model. Fournier[7] concentrated on the shallow water waves and surf along a shore line. He started from parametric equations and added control parameters to simulate

* Corresponding author.

various shapes of shallow water waves, but not for the large-scale ocean scenes and/or deep-depth ocean waves. More complex parametric equations to present the propagation of water waves had been introduced by Gonzato[8]. This model is well suited for modeling propagating water of wave front, but its equations are too complex for large-scale ocean waves.

Another group regards the ocean surface as a height field with a prescribed spectrum based on the experimental observations from oceanography. Mastin[9] introduced an effective simulation of wave behavior using Fast Fourier Transform(FFT). The height field is constructed through inverse FFT of the frequency spectrum of the real world ocean waves. It can produce complex wave patterns similar to real world ocean waves. Tessendorf[10] showed that dispersive propagation could be managed in the frequency domain and that the resulting field could be modified to yield trochoid waves. However, the negative aspect of FFT based methods is homogeneity: we cannot handle any local properties such as refraction, reflection, and others.

The last one is the hybrid approach: The spectrum synthesized by a spectral approach is used to control the trochoids generated by the Gerstner model. Hinsinger[11] presented an adaptive scheme for the animation and display of ocean waves in real time. It relied on a procedural wave model which expresses surface point displacements as sums of wave trains. In this paper, we aim to construct an ocean wave model with the following characteristics:

- **Real time capability:** They usually want to display a large scale ocean scene and some special effects may be added to the scene. So, we need to generate the ocean wave in real time.
- **More user-controllable parameters:** We will provide more parameters to generate variety of ocean scenes including deep and shallow oceans, windy and calm oceans, etc.
- **Focusing on the surface gravity waves:** Since we target the large-scale ocean, minor details of the ocean wave are not our major interest. In fact, the minor details can be easily super-imposed to the surface gravity waves, if needed.

In the following sections, we will present a new hybrid approach to finally get a real-time surface gravity wave simulation. Since it is a kind of hybrid approach, it can generate large scale oceans without difficulty, and works in real time, to be sufficiently used with computer generated animations or other special effects. Additionally, we use a more precise wave model and have more controllable parameters including depth of sea, fetch length, wind speed, and so on, in comparison with previous hybrid approaches. We will start from the theoretical ocean wave models in the following section, and build up our implementation model. Our implementation results and conclusions will be followed.

2 The Ocean Wave Model

The major generating force for waves is the wind acting on the interface between the air and the water. From the mathematical point of view, the surface is made

up of many sinusoidal waves generated by the wind, and they are traveling through the ocean. One of the fundamental models for the ocean wave is the Gerstner swell model, in which the trajectory of a water particle is expressed as a circle of radius r around its reference location at rest, (x_0, z_0) , as follows[11]:

$$\begin{aligned} x &= x_0 + r \sin(\omega t - kx_0) \\ z &= z_0 + r \cos(\omega t - kz_0), \end{aligned} \tag{1}$$

where (x, z) is the actual location at time t , $\omega=2\pi f$ is the pulsation with the frequency f , and $k=2\pi/\lambda$ is the wave number with respect to the wave length of λ .

Equation (1) shows a two-dimensional representation of the ocean wave, assuming that the x -axis coincides to the direction of wave propagation. The surface of an ocean is actually made up of a finite sum of these simple waves, and the height z of the water surface on the grid point (x, y) at time t can be expressed as:

$$z(x, y, t) = \sum_i^n A_i \cos(k_i(x \cos \theta_i + y \sin \theta_i) - \omega_i t + \varphi_i), \tag{2}$$

where n is the number of wave trains, A_i is the amplitude, k_i is the wave number, θ_i is the direction of wave propagation on the xy -plane and φ_i is the phase. In Hinsinger[11], they manually selected all these parameters, and thus, the user may meet difficulties to select proper values of them.

In contrast, Thon[12] uses a spectrum-based method to find some reasonable parameter sets. They used the Pierson-Moskowitz(PM) model[13], which empirically expresses a fully developed sea in terms of the wave frequency f as follows:

$$E_{PM}(f) = \frac{0.0081 g^2}{(2\pi)^4 f^5} e^{-\frac{5}{4} \left(\frac{f_p}{f}\right)^4},$$

where $E_{PM}(f)$ is the spectrum, g is the gravity constant and $f_p = 0.13 g/U_{10}$ is a peak of frequency depending on the wind speed U_{10} at a height of 10 meters above the sea surface.

Although Thon used the PM model to give some impressive results, the PM model itself assumes the infinite depth of the ocean and thus may fail to the shallow sea cases. To overcome this drawback, the JONSWAP model and TMA model are introduced. The JONSWAP(Joint North Sea Wave Project) model[14] is developed for fetch-limited seas such as North sea and expressed as follows:

$$E_{JONSWAP}(f) = \frac{\alpha g^2}{(2\pi)^4 f^5} e^{-\frac{5}{4} \left(\frac{f_p}{f}\right)^4} \cdot \gamma e^{-\frac{f/f_p - 1}{2\sigma^2}},$$

where α is the scaling parameter, γ is the peak enhancement factor, and σ is evaluated as 0.07 for $f \leq f_p$ and 0.09 otherwise. Given the fetch length F , the frequency at the spectral peak f_p is calculated as follows:

$$f_p = 3.5 \left(\frac{g^2 F}{U_{10}^3} \right)^{-0.33}.$$

The Texel, Marson and Arsole(TMA) model[15] extends the JONSWAP model to include the depth of water h as one of its implicit parameters as follows:

$$E_{\text{TMA}}(f) = E_{\text{JONSWAP}}(f) \cdot \Phi(f^*, h),$$

where $\Phi(f^*, h)$ is the Kitaigorodskii depth function:

$$\Phi(f^*, h) = \frac{1}{s(f^*)} \left[1 + \frac{K}{\sinh K} \right],$$

with $f^* = f\sqrt{h/g}$, $K = 2(f^*)^2 s(f^*)$ and $s(f^*) = \tanh^{-1}[(2\pi f^*)^2 h]$.

The TMA model shows good empirical behavior even with the water depth of 6 meters. Thus, it is possible to represent the waves on the surface of lake or small-size ponds, in addition to the ocean waves. Additionally, it also includes the fetch length as a parameter, inherited from the JONSWAP model. Thus, the expression power of the TMA model is much increased in comparison with the PM model previously used by other researchers. We use this more improved wave model to finally achieve more realistic ocean scenes with more user-controllable parameters.

3 The Implementation Model

To derive implementation-related expressions, we need to extend the spectrum of TMA model to two dimensional world as follows[14]:

$$E(f, \delta) = E_{\text{TMA}}(f)D(f, \delta),$$

where $D(f, \delta)$ is a directional spreading factor that weights the spectrum at angle δ from the downwind direction. The spreading factor is expressed as follows:

$$D(f, \delta) = N_p^{-1} \cos^{2p} \left(\frac{\delta}{2} \right),$$

where $p = 9.77(f/f_p)^\mu$, $N_p = 2^{1-2p}\pi \cdot \Gamma(2p+1)/\Gamma^2(p+1)$ with Euler's Gamma function Γ and

$$\mu = \begin{cases} 4.06, & \text{if } f < f_p \\ -2.34, & \text{otherwise} \end{cases}.$$

For more convenience in its implementation, we will derive some evaluation functions for the parameters including frequency, amplitude, wave direction, wave number and pulsation. The frequency of each wave train is determined from the peak frequency f_p and a random offset to simulate the irregularity of the ocean waves. Thereafter, the pulsation and wave number is naturally calculated by their definition.

According to the random linear wave theory[16,17,18,19,20], directional wave spectrum $E(f, \delta)$ is given by

$$E(f, \delta) = \Psi(k(f), \delta) \cdot k(f) \frac{dk(f)}{df}, \tag{3}$$

where $k(f) = 4\pi^2 f^2/g$ and $\Psi(k(f), \delta)$ is a wave number spectrum. The second and the third term in Equation (3) can be computed as:

$$k(f) \frac{dk(f)}{df} = \frac{32 \pi^2 f^3}{g^2}.$$

This allows us to re-write Equation (3) as follows[17]:

$$E(f, \delta) = \Psi(k(f), \delta) \frac{32 \pi^2 f^3}{g^2}.$$

From the random linear wave[17,19], the wave number spectrum $\Psi(k(f), \delta)$ can be approximated as:

$$\Psi(k(f), \delta) = \frac{\beta}{4\pi^2} A(f)^2,$$

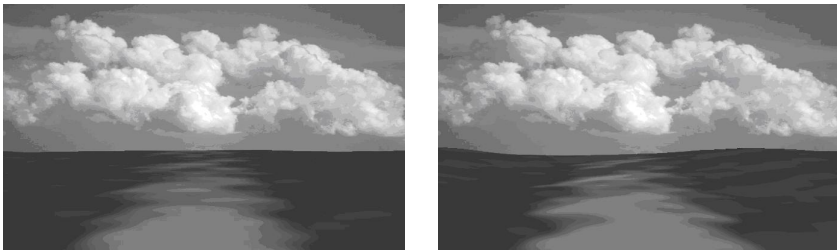
where β is a constant. Finally, the amplitude $A(f)$ of a wave train is evaluated as:

$$A(f) = \sqrt{\frac{E(f, \delta) g^2}{8 f^3 \beta}} = \sqrt{\frac{E_{\text{TMA}}(f) D(f, \delta) g^2}{8 f^3 \beta}}.$$

Using all these derivations, we can calculate the parameter values for Equation (2). And then, we evaluate the height of each grid point (x, y) to construct a rectangular mesh representing the ocean surface.

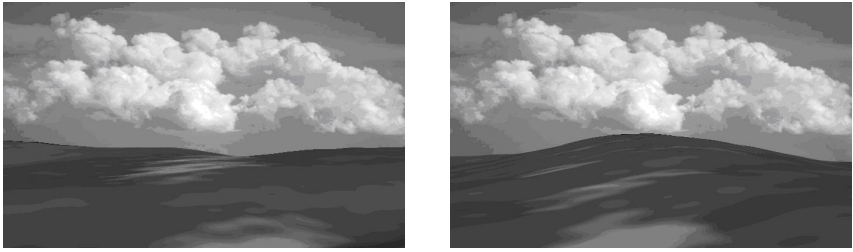
4 Implementation Results

Figures 1, 2 and 3 are some outputs from the prototype implementation. As shown We implemented the ocean wave generation program based on the TMA model presented in the previous section. It uses plain OpenGL library and does not use any multi-threading or hardware-based acceleration techniques. At this time, we focused on the expression power of our TMA model-based implementation, and thus, our prototype implementation lacks some acceleration or optimization factors. Even though, it shows more than 50 frames per second on a PC with Intel



(a) wind speed 3m/s, water depth 5m (b) wind speed 3m/s, water depth 100m

Fig. 1. Ocean waves with different water depths: Even with the same wind speed, different water depths result in very different waves. We use the fetch length of 5km for these images.



(a) wind speed 3m/s, water depth 100m (b) wind speed 6m/s, water depth 100m

Fig. 2. Ocean waves with different wind velocities: Changes in wind speed generate more clam or more choppy waves. The fetch length of 10 km is used for each of these images.

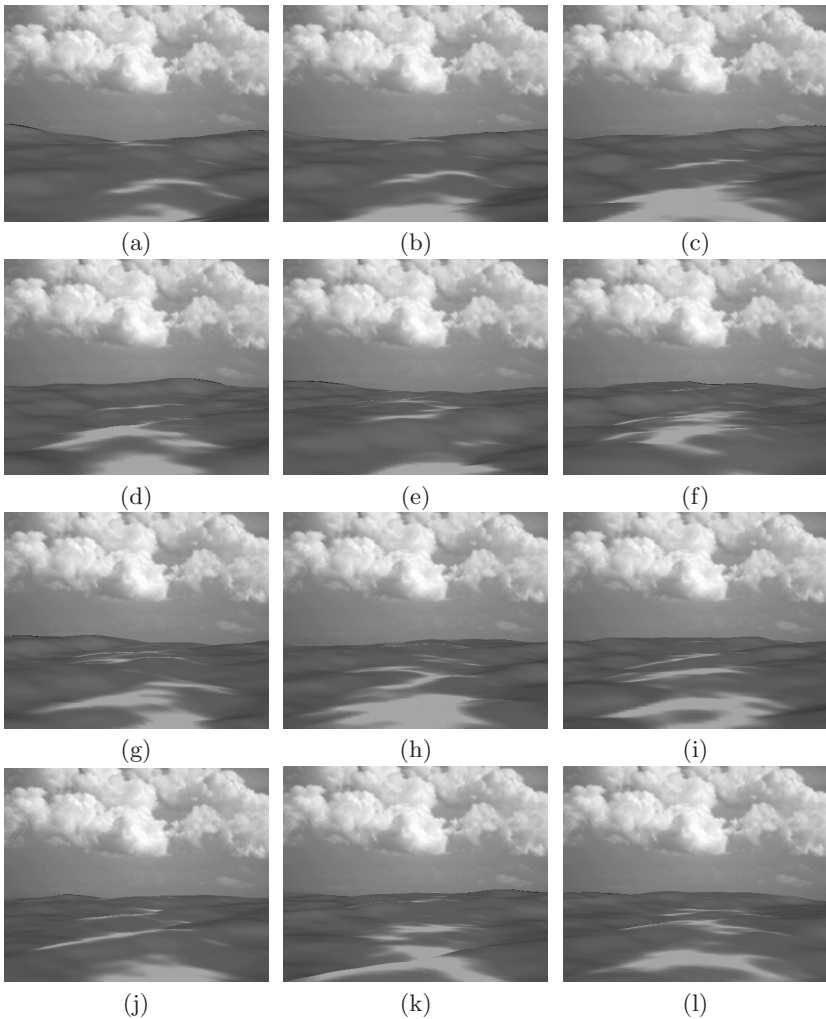


Fig. 3. An animated sequence of ocean waves

Core2 Duo 6600 2.40GHz processor and a GeForce 7950GT based graphics card. We expect that the frame rate will be much better in the next version.

In Figure 1, we can control the depth of the ocean to show very different waves even with the same wind speed and the same fetch length. Especially, the changes in the water depth are acceptable only for the TMA model, while the previous PM model cannot handle it. Figure 2 shows the effect of changing the wind speed. As expected, the high wind speed generates more choppy waves. Figure 3 is a sequence of images captured during the real time animation of the windy ocean. All examples are executed with mesh resolution of 200×200 . More examples are in our web page, <http://isaac.knu.ac.kr/~hope/tma.htm>.

5 Conclusion

In this paper, we present a real-time surface gravity wave simulation method, derived from a precise ocean wave model in the oceanography. We started from a precise ocean wave model of TMA model, which has not been used for a graphics implementation, at least to our knowledge. Since we used a more precise ocean wave model, users can control more parameters to create various ocean scenes.

The two major improvements of our method in comparison with the previous works will be:

- **Enhanced expression power:** Our method can display visually plausible scenes even for shallow seas.
- **Improved user controllability:** Our method provides more parameters such as fetch length and depth of water, in addition to the wind velocity.

We implemented a prototype system, and showed that it can generate animated sequences of ocean waves in real time. We plan to integrate our implementation to large-scale applications such as games, maritime training simulators, etc. Some detailed variations to the ocean waves can also be added to our implementation with minor modifications.

Acknowledgements

This research was supported by the *Regional Innovation Industry Promotion Project* which was conducted by the Ministry of Commerce, Industry and Energy(MOCIE) of the Korean Government (70000187-2006-01).

References

1. Iglesias, A.: Computer graphics for water modeling and rendering: a survey. *Future Generation Comp. Syst.* **20**(8) (2004) 1355–1374
2. Enright, D., Marschner, S., Fedkiw, R.: Animation and rendering of complex water surfaces. In: *SIGGRAPH '02.* (2002) 736–744
3. Foster, N., Fedkiw, R.: Practical animation of liquids. In: *SIGGRAPH '01.* (2001) 23–30

4. Foster, N., Metaxas, D.N.: Realistic animation of liquids. *CVGIP: Graphical Model and Image Processing* **58**(5) (1996) 471–483
5. Foster, N., Metaxas, D.N.: Controlling fluid animation. In: *Computer Graphics International '97*. (1997) 178–188
6. Stam, J.: Stable fluids. In: *SIGGRAPH '99*. (1999) 121–128
7. Fournier, A., Reeves, W.T.: A simple model of ocean waves. In: *SIGGRAPH '86*. (1986) 75–84
8. Gonzato, J.C., Sač, B.L.: On modelling and rendering ocean scenes. *J. of Visualization and Computer Animation* **11**(1) (2000) 27–37
9. Mastin, G.A., Watterberg, P.A., Mareda, J.F.: Fourier synthesis of ocean scenes. *IEEE Comput. Graph. Appl.* **7**(3) (1987) 16–23
10. Tessendorf, J.: Simulating ocean water. In: *SIGGRAPH '01 Course Notes*. (2001)
11. Hinsinger, D., Neyret, F., Cani, M.P.: Interactive animation of ocean waves. In: *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. (2002) 161–166
12. Thon, S., Dischler, J.M., Ghazanfarpour, D.: Ocean waves synthesis using a spectrum-based turbulence function. In: *Computer Graphics International '00*. (2000) 65–
13. Pierson, W., Moskowitz, L.: A proposed spectral form for fully developed wind seas based on the similarity theory of S.A. kitaigorodskii. *J. Geophysical Research* (69) (1964) 5181–5190
14. Hasselmann, D., Duncel, M., Ewing, J.: Directional wave spectra observed during JONSWAP 1973. *J. Physical Oceanography* **10**(8) (1980) 1264–1280
15. Bouws, E., Günther, H., Rosenthal, W., Vincent, C.L.: Similarity of the wind wave spectrum in finite depth water: Part 1. spectral form. *J. Geophysical Research* **90** (1985) 975–986
16. Crawford, F.: *Waves*. McGraw-Hill (1977)
17. Krogstad, H., Arntsen, Ø.: *Linear Wave Theory*. Norwegian Univ. of Sci. and Tech. (2006) <http://www.bygg.ntnu.no/oivarn/>.
18. Seyringer, H.: *Nature wizard* (2006) http://folk.ntnu.no/oivarn/hercules_ntnu/LWTcourse/.
19. Sorensen, R.: *Basic Coastal Engineering*. Springer-Verlag (2006)
20. US Army Corps of Engineers Internet Publishing Group: *Coastal engineering manual – part II* (2006) <http://www.usace.army.mil/publications/engine-manuals/em1110-2-1100/PartII/PartII.htm>.