International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM)
First Special Issue on “Computer Graphics and Geometric Modeling” (2008)
Selected Papers from CGGM’2007 Workshop

A Real-Time Method for Ocean Surface Simulation using the TMA Model

Namkyung Lee
Dept. of Computer Engineering
Kyungpook National Univ.
Daegu, 702-701, Korea
namklee@hotmail.com

Nakhoon Baek*
School of EECS
Kyungpook National Univ.
Daegu, 702-701, Korea
oceancru@gmail.com
* *correspoding author*

Kwan Woo Ryu
Dept. of Computer Engineering
Kyungpook National Univ.
Daegu, 702-701, Korea
kwryu@knu.ac.kr

Abstract— **In the field of computer graphics, we have several research results to display the ocean waves on the screen, while we still not have a complete solution yet. Though ocean waves are constructed from a variety of sources, the dominant one is the surface gravity wave, which is generated by the gravity and the wind. In this paper, we present a real-time surface gravity wave simulation method, derived from a precise ocean wave model in the oceanography. While there are many ocean wave models in the field of oceanography, they have used relatively simple models in computer graphics area. Especially, there are research results based on the Pierson-Moskowitz(PM) model, which assumes infinite depth of water and thus shows some mismatches in the case of shallow seas. In contrast, we started from the Texel, Marsen and Arsloe(TMA) model, which is a more precise wave model and thus can be used to display more realistic ocean waves. We derived its implementation model for the graphics applications and our prototype implementation shows more than 50 frames per second on the Intel Core2 Duo 2.4GHz-based personal computers. Our major contributions to the computer graphics area will be the improvement on the expression power of ocean waves and providing more user-controllable parameters for various wave shapes.**

Index Terms— **computer graphics simulation, ocean wave, TMA model, real-time simulation.**

1 Introduction

Representing natural phenomena more realistically is one of the most important factors to increase reality, especially in the various fields related to the computer graphics applications. For example, in the case of three-dimensional computer games, more realistic behaviors in the gaming environment will deliver more immersion to the users. In this paper, among the various natural phenomena, 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 ocean wave-related results into two categories: fluid dynamics-based methods[2, 3, 4, 5, 6] and oceanography spectrum analysis-based methods[7, 8, 9]. Recently, some hybrid methods[10, 11] are also available. Although these methods have capability of generating much realistic images, they all still have some drawbacks including too much computations, restricted expression powers, limited number of controllable parameters, etc.

In this paper, we aim to construct an ocean wave model based on a surface gravity model, which is newly introduced into the computer graphics field. Our ocean wave model can overcome the limitations in the previous hybrid ocean-wave models. Our method has 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.
- **Cooperative to other wave generation methods:** Based on the mesh structure, our method can be used simultaneously with other wave generation methods. Furthermore, most geometric operations are easily applicable to our resulting mesh structure.

In the next section, previous results on the ocean surface waves are presented. 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 Previous Works

There are two approaches for ocean wave models. The first approach uses fluid mechanics equations such as those used in scientific simulations. These physical equations are mainly derived from the Navier-Stokes equations introduced in 1827, which are the governing laws of fluid mechanics. Foster and Metaxas[3] modified the classic marker and cell(MAC) method[7] to obtain realistic fluids behavior. They extended their method later[4] to include simple control mechanisms at the level of physical parameters. Stam[12] departed from the finite-difference scheme used by Foster and Metaxas[3] and introduced the stable semi-Lagrangian methods for computing the advection part of the Navier-Stokes equations. Foster and Fedkiw introduced in [5] a hybrid liquid model, combining implicit surfaces and particles, while Enright et al.[6] improve the hybrid model devising the particle level set method, which is presently one of the methods of choice for obtaining very realistic animations of complex water surfaces. But these models are only for representing small water surfaces and never for ocean waves. In fact, the representation of large scenes of ocean waves is an open problem for methods based on fluid mechanics, because the computation of water motion over wide areas entails important memory storage and computation time problems.

The second approach consists of three groups. One simulates a train of trochoids based on the water wave model by Gerstner. Fournier[2] had modeled shallow water waves and surf along a shore line. His model uses two parametric equations to be represented allow any wave shape. Fournier added size and shape parameters to a basic wave equation to simulate various shape waves. More complex parametric equations to solve the propagation of water waves had been used by Gonzato[9]. This model is well suited for modeling propagating water of wave front, but its equations become quite complex as more shape manipulations are imposed.

Another group is to synthesize the ocean surface as a height field with a prescribed spectrum based on experimental observations. Mastin[13] introduced an effective simulation of wave behavior using the Fast Fourier. The idea is to produce a height field having the same spectrum as the ocean surface. This can be done by filtering a white noise with filter obtained observations in oceanography and then calculating its FFT. Despite its simplicity, it produces complex wave patterns similar to real ocean waves. Tessendorf[12] showed that dispersive propagation could be managed in the frequency domain and that the resulting field could be modified to yield trochoid waves. The negative aspect of FFTs is homogeneity. No local property can exist, so no refraction can be handled.

The other is an hybrid approach. The spectrum synthesized using a spectral approach is used to control the trochoids generated by a Gerstner model[14]. This is only applicable in the calm sea case, where trochoids of small amplitude are very similar to sines. Smaller scale waves are obtained by directly tuning some extra Perlin turbulence function[10]. Hinsinger[11] presented an adaptive scheme for the animation and display of ocean waves. It relied on a procedural wave model which expresses surface point displacements as sums of wave-trains. Hinsinger obtained parameters manually, but it is hard to use.

In this paper, we propose a new hybrid technique and generate ocean waves using spectrum. Our technique uses parameter such as, depth of sea, fetch length and wind speed, so user can generate various ocean waves under different condition easily and simply with proposed technique.

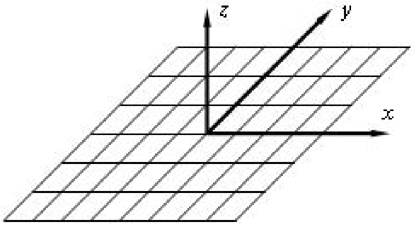


Figure 1: The mesh structure for the ocean waves

3 The Ocean Wave Model

Since we are focusing on the surface gravity waves, the ocean itself can be modeled as a height field where a two-dimensional coordinate (x, y) represents a location in the interesting area and its corresponding height z is used to express the water level at that location. For rapid display, we arranged the location (x, y)'s to the rectangular grid points, as shown in Figure 1. In this case, the ocean surface can be expressed as an ordinary mesh structure, and we can use many acceleration techniques to display it on the screen.

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

$$k = \frac{2\pi}{\lambda}$$

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:

$$z(x,y,t) = \sum_{i}^{n} A_i \cos\left(k_i(x\cos\theta_i + y\sin\theta_i) - \omega_i t + \varphi_i\right), \quad (2)$$

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[15] uses a spectrum-based method to find some reasonable parameter sets. They used the Pierson-Moskowitz(PM) model[16], which empirically expresses a fully developed sea in terms of the wave frequency f as follows:

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

where $E_{\mathrm{PM}}(f)$ is the spectrum, g is the gravity constant and

$$f_{\mathrm{p}} = \frac{0.13g}{U_{10}}$$

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[8] is developed for fetch-limited seas such as North sea and expressed as follows:

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

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

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

The Texel, Marson and Arsole(TMA) model[17] extends the JONSWAP model to include the depth of water h as one of its implicit parameters as follows:

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

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

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

with

$$f^* = f\sqrt{\frac{h}{g}},$$

$$K = 2(f^*)^2 s(f^*)$$

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.

4 The Implementation Model

To derive implementation-related expressions, we need to extend the spectrum of TMA model to two dimensional world as follows[8]:

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

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

$$p = 9.77\left(\frac{f}{f_{\mathrm{p}}}\right)^{\mu},$$

$$N_p = \frac{2^{1-2p}\pi\Gamma(2p+1)}{\Gamma^2(p+1)}$$

with Euler's Gamma function Γ and

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

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[18, 19, 20, 21, 22], directional wave spectrum $E(f, \delta)$ is given by

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

where

$$k(f) = 4\pi^2 \frac{f^2}{g}$$

and $\Psi(k(f), \delta)$ is a wave number spectrum. The second and the third term in Equation (3) can be computed as:

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


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


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

Figure 2: 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 2m/s, water depth 10m


(b) wind speed 2m/s, water depth 50m

Figure 3: Another example for different water depths. We use the fetch length of 3km for these images.

This allows us to re-write Equation (3) as follows[19]:

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

From the random linear wave[19, 21], 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:

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

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.

5 Implementation Results

Figures 2, 3, 4 5 and 6 are some outputs from the prototype implementation. As shown in these examples, 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 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 Figures 2 and 3, 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. Figures 4 and 5 show the effect of changing the wind speed. As expected, the high wind speed generates more choppy waves. Figure 6 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.

6 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.

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.

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

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

Figure 4: Ocean waves with different wind velocities: Changes in wind speed generate more clam or more choppy waves. The fetch length of 10km is used for each of these images.

(a) wind speed 2m/s, water depth 50m

(b) wind speed 5m/s, water depth 50m

Figure 5: Another example for different wind velocities. The fetch length of 5km is used for each of these images.

- **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. Since we used a more precise ocean wave model, users can control more parameters to create various ocean scenes. Additionally, we expect that the execution speed will be more improved using GPU-based techniques.

In near future, 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 work was supported by the IT R&D program of MIC/IITA. [2007-S001-01, Development of Multi-Core CPU & MPU-Based Cross-Platform Game Technology]

Bibliography

[1] A. Iglesias. Computer graphics for water modeling and rendering: a survey. *Future Gener. Comput. Syst.*, 20(8):1355–1374, 2004.

[2] Alain Fournier and William T. Reeves. A simple model of ocean waves. In *SIGGRAPH ’86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 75–84, New York, NY, USA, 1986. ACM Press.

[3] Nick Foster and Dimitris N. Metaxas. Realistic animation of liquids. *CVGIP: Graphical Model and Image Processing*, 58(5):471–483, 1996.

[4] Nick Foster and Dimitris N. Metaxas. Controlling fluid animation. In *Computer Graphics International*, pages 178–188, 1997.

[5] Nick Foster and Ronald Fedkiw. Practical animation of liquids. In *SIGGRAPH ’01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23–30, New York, NY, USA, 2001. ACM Press.

[6] Douglas Enright, Stephen Marschner, and Ronald Fedkiw. Animation and rendering of complex water surfaces. In *SIGGRAPH ’02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 736–744, New York, NY, USA, 2002. ACM Press.

[7] Francis H. Harlow and J. Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8(12):2182–2189, 1965.

[8] D. E. Hasselmann, M. Dunckel, and J. A. Ewing. Directional wave spectra observed during JONSWAP 1973. *Journal of Physical Oceanography*, 10(8):1264–1280, 1980.

[9] Jean-Christophe Gonzato and Bertrand Le Saëc. On modelling and rendering ocean scenes. *Journal of Visualization and Computer Animation*, 11(1):27–37, 2000.

[10] Ken Perlin. An image synthesizer. In *SIGGRAPH ’85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 287–296, New York, NY, USA, 1985. ACM Press.

[11] Damien Hinsinger, Fabrice Neyret, and Marie-Paule Cani. Interactive animation of ocean waves. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 161–166, New York, NY, USA, 2002. ACM Press.

[12] Jos Stam. Stable fluids. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[13] Gary A. Mastin, Peter A. Watterberg, and John F. Mareda. Fourier synthesis of ocean scenes. *IEEE Comput. Graph. Appl.*, 7(3):16–23, 1987.

[14] Oliver Deusen, David S. Ebert, Ron Fedkiw, F. Kenton Musgrave, Przemyslaw Prusinkiewicz, Doug Roble, Jos Stam, and Jerry Tessendorf. The elements of nature: interactive and realistic techniques. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes*, page 32, New York, NY, USA, 2004. ACM Press.

[15] Sebastien Thon, Jean-Michel Dischler, and Djamchid Ghazanfarpour. Ocean waves synthesis using a spectrum-based turbulence function. In *CGI '00: Proceedings of the International Conference on Computer Graphics*, page 65, Washington, DC, USA, 2000. IEEE Computer Society.

[16] W. J. Pierson and L. Moskowitz. A proposed spectral form for fully developed wind seas based on the similarity theory of S.A. kitaigorodskii. *Journal of Geophysical Research*, (69):5181–5190, 1964.

[17] E. Bouws, H. Günther, W. Rosenthal, and C. L. Vincent. Similarity of the wind wave spectrum in finite depth water: Part 1. spectral form. *Journal of Geophysical Research*, 90:975–986, 1985.

[18] F. Crawford. *Waves*. McGraw-Hill, 1977.

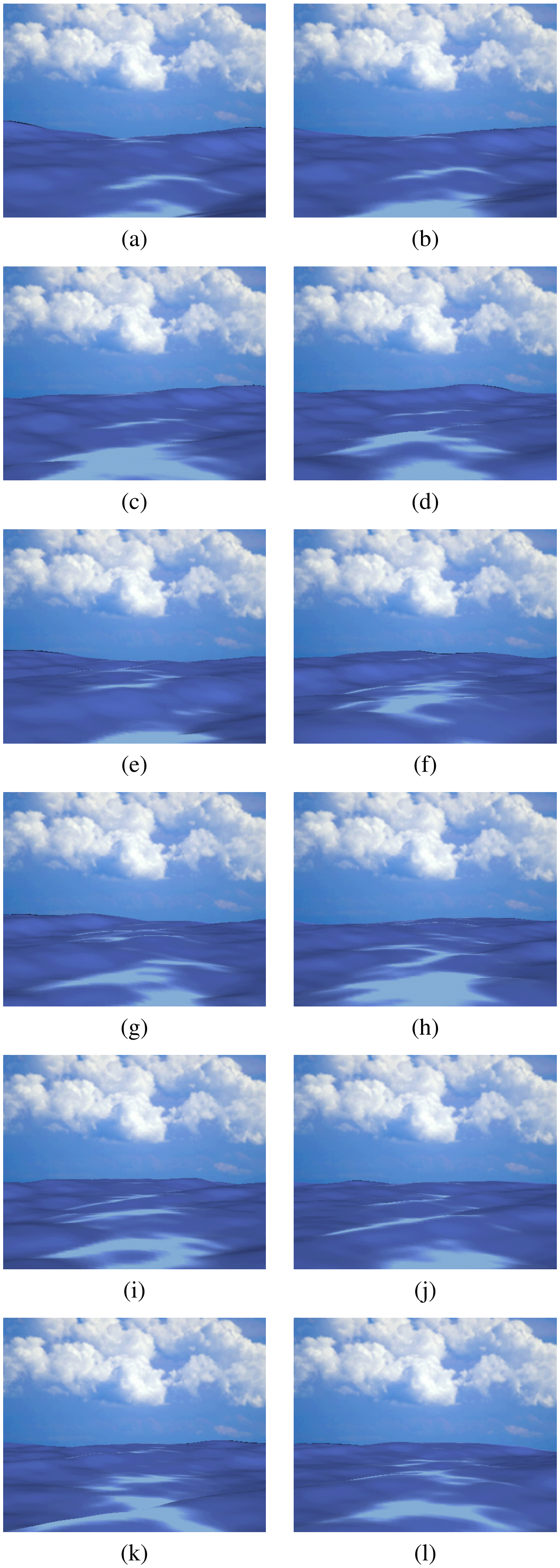
[19] H. Krogstad and Ø. Arntsen. *Linear Wave Theory*. Norwegian Univ. of Sci. and Tech., 2006. http://www.bygg.ntnu.no/ oivarn/.

[20] H. Seyringer. Nature wizard, 2006. http://folk.ntnu.no/-oivarn/hercules_ntnu/LWTcourse/.

[21] R. Sorensen. *Basic Coastal Engineering*. Springer-Verlag, 2006.

[22] US Army Corps of Engineers Internet Publishing Group. Coastal engineering manual – part II, 2006. http://www.usace.army.mil/publications/eng-manuals/-em1110-2-1100/PartII/PartII.htm.

Namkyung Lee received the BS, MS, and Ph.D degrees in Computer Engineering from the Kyungpook National University, Korea, in 1998, 2000, and 2007, respectively. He is currently a lecturer in the Department of Computer Engineering at Kyungpook National University. His primary research interests are physics-based animation.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Figure 6: An animated sequence of ocean waves.

Nakhoon Baek is currently an associate professor in the School of Electrical Engineering & Computer Science at Kyungpook National University(KNU), Korea. He received his BA, MS, and Ph.D degree in Computer Science from the Korea Advanced Institute of Science and Technology(KAIST) in 1990, 1992, and 1997, respectively. His research interests include real-time rendering and graphics algorithms.

Kwan Woo Ryu received the BS degree in electrical engineering from the Kyungpook National University, Korea, in 1980, and the MS degree in computer science from the Korea Advanced Institute of Science and Technology, Korea in 1982. He received his Ph.D degree from the Department of Computer Science at the University of Maryland, College Park, in 1990. Currently, he is a professor in the Department of Computer Engineering, Kyungpook National University, Korea. His main research interests are design and analysis of paralled algorithms for combinatorial problems, and computer graphics.