

Simulation of Shallow-Water Waves in Coastal Region for Marine Simulator

Yongjin Li^{1,2*}

Yicheng Jin^{1†}

Yong Yin^{1‡}

Helong Shen^{1§}

¹Dalian Maritime University

²Public Security Marine Police Academy

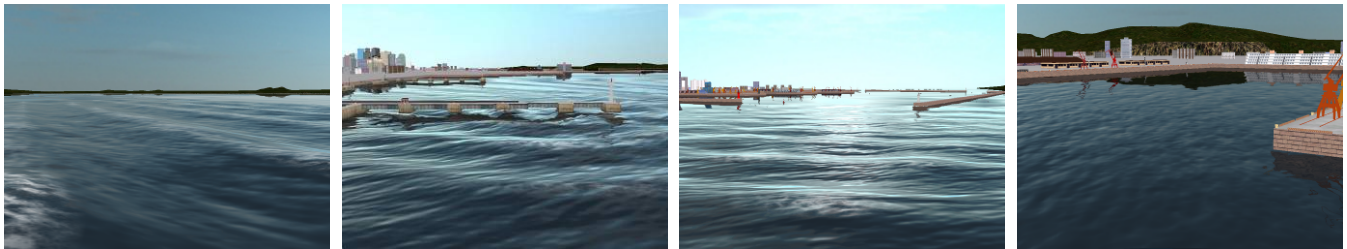


Figure 1: Sample frames captured from our real-time simulation system at different viewpoints in the Dalian Port scene. Appearances of the sea surface vary significantly with locations by the impact of the topography and coast constructions.

Abstract

This paper presents a new method for simulating shallow-water waves for the marine simulator. Firstly, a sequence of sea surface height fields is achieved by solving 2D Boussinesq type equations. These height fields can exhibit the combined effect of the most shallow-water waves in the coastal region, such as shoaling, refraction, diffraction, reflection and non-linear wave-wave interaction, etc. Secondly, these height fields are synthesized to a new unlimited long sequence by rearranging their order according to their similarity. Finally, the height fields are used as vertex textures sampled by a view-dependent sea surface grid in the new order. Experimental results show that the simulated shallow-water waves have realistic effect with fast rendering speed. It is suitable for the applications of real-time simulation, the marine simulator especially.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: shallow-water waves simulation, physically based wave modeling, real-time wave rendering, marine simulator

1 Introduction

Simulating ocean waves has been among the most important problems in the study of the marine simulator, because the realistic waves can not only increase the realism of the virtual ocean scene, but also provide correct wave-parameters to the ship maneuvering model. It has also been one of the most challenging problems in the research of the marine simulator. Firstly, ocean waves are an important factor for ship maneuvering, which requires accurate wave models. Secondly, the visual systems in marine simulators are com-

mon multi-channel, so the ocean waves in each channel must stitch up seamlessly. Finally, the virtual scene also contains many other objects (ships, islands and buildings, for example), so the ocean waves must be rendered enough quickly to ensure that the whole scene is rendered in real-time.

There are many different types of waves in the ocean: tide, tsunami, wind wave, ship wave, deep-water wave and shallow-water wave etc. This paper focuses on the simulation of shallow-water waves because the vast majority of virtual scenes in the marine simulator are in coastal region. The significant wave in this region is the shallow-water wave, which is generated by deep-ocean waves transmitting to the shore. The simulation of shallow-water waves is more difficulty than that of deep-water waves because shallow-water waves have more complex phenomena due to the influence of the topographic and marine structures.

There are two majors companies in the market of marine simulator: Norcontrol and Transas. These two companies offer a large range of marine simulators. Nevertheless, we cannot find any realistic simulation of shallow-water waves in their simulators. The marine simulators produced by us before also used the deep-ocean wave model in the coast region [Xie et al. 2004; Yin et al. 2003].

In this paper, we introduce a method for simulating shallow-water waves in the coast region for the marine simulator, which is capable of exhibiting the combined behavior of shallow-water waves such as shoaling, refraction, diffraction, reflection, non-linear wave-wave interaction, etc.

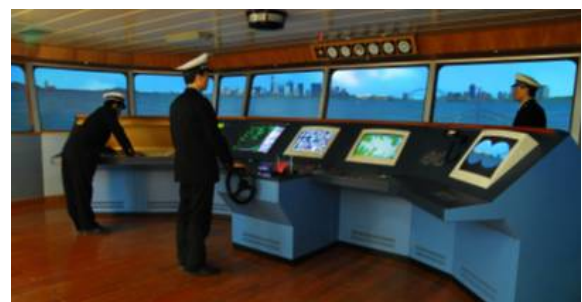


Figure 2: A photo taken in one of our marine simulators.

Figure 1 was created by capturing frames from the real-time simulation of Dalian Port scene at different viewpoints. The waves in

*e-mail: liyj.cn@gmail.com

†e-mail: jycdmu@dlmu.edu.cn

‡e-mail: bushyin.dmu@263.net

§e-mail: shenhelong@126.com

Copyright © 2008 by the Association for Computing Machinery, Inc.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail permissions@acm.org.

VRCAI 2008, Singapore, December 8–9, 2008.

© 2008 ACM 978-1-60558-335-8/08/0012 \$5.00

different regions have different appearances, which correctly reflect the impact on waves by the environment. Figure 2 is a photo of our marine simulator. Our method will be applied in this system.

2 Previous Work

In this section we briefly overview some of the previous work on shallow water simulation in computer graphics. For an review in engineering we recommend [Feng et al. 2003].

Some previous work on water simulation modeled water as a surface using parametric representations [Cieutat et al. 2001; Fournier and Reeves 1986; Gonzato and Saëc 1997; Peachey 1986; Ts'o and Barsky 1987]. Wave refraction caused by water depth varying was simulated using wave tracing. [Cieutat et al. 2001] used this method for simulating the wave near the shore in their maritime training simulator. [Gonzato and Saëc 2000] proposed a geometrical model accounting for refraction, diffraction, reflection, transmission and multi-waves trains, and described a specific algorithm for the rendering of multi-waves trains. These methods can only simulate water waves in the simple environment, and the accuracy of wave movement is low.

Others rely on solving the hydrodynamics equations to simulate shallow-water waves. [Kass and Miller 1990] used finite differences to solve simplified 2D shallow water equations over a dynamic height field. This method can generate the effects of wave refraction with depth, wave reflection, etc. [Layton and Panne 2002] made the computation more stable by using the implicit semi-Lagrangian integration for 2D shallow water equations.

There is a lot of work has been done on fluids simulation by solving 3D Navier-Stokes equations [Enright et al. 2002; Foster and Metaxas 1996; Takahashi et al. 2003]. Although these methods can generate high quality simulating result of shallow-water waves, all of these methods are computationally expensive. They can not been used for simulating large area of near-shore wave therefor.

3 Sea Surface Motion Model

Because all of the current models in computer graphics fields can not meet our demand for simulating large area of shallow-water waves we have to find a model in the fields of engineering. There are several different models for shallow-water waves simulation. The 2D Boussinesq equations model is the most suitable for our needs. The careful comparison of each model was given in [Feng et al. 2003]. The details about Boussinesq equations used in our method can refer to [Abbott et al. 1978; Madsen et al. 1991; Madsen and Sørensen 1992].

P, Q	depth integrated velocity in the x,y direction, m ² /s
ψ_x, ψ_y	Boussinesq dispersion term, m ² /s ²
F_x, F_y	horizontal stress term in x,y direction, m ² /s ²
B	Boussinesq dispersion factor
C	Chezy resistance number, m ^{0.5} /s
x, y	Cartesian coordinates, m
d	still water depth, m
g	gravitational acceleration(= 9.8 m/s ²)
h	total water depth(= $d + \xi$), m
n	porosity
t	time, s
α	resistance coefficient for laminar flow in porous media
β	resistance coefficient for turbulent flow in porous media
ξ	water surface level above datum, m

Table 1: Symbol list in Boussinesq equations

The symbols we used are listed in Table 1. The enhanced 2D Boussinesq equations in our method are:

$$n \frac{\partial \xi}{\partial t} + \frac{\partial P}{\partial x} + \frac{\partial Q}{\partial y} = 0, \quad (1)$$

$$n \frac{\partial P}{\partial t} + \frac{\partial}{\partial x} \left(\frac{P^2}{h} \right) + \frac{\partial}{\partial y} \left(\frac{PQ}{h} \right) + n^2 g h \frac{\partial \xi}{\partial x} + F_x + n^2 P \left[\alpha + \beta \frac{\sqrt{P^2 + Q^2}}{h} \right] + \frac{gP \sqrt{P^2 + Q^2}}{h^2 C^2} + n\psi_x = 0, \quad (2)$$

$$n \frac{\partial Q}{\partial t} + \frac{\partial}{\partial y} \left(\frac{Q^2}{h} \right) + \frac{\partial}{\partial x} \left(\frac{PQ}{h} \right) + n^2 g h \frac{\partial \xi}{\partial y} + F_y + n^2 Q \left[\alpha + \beta \frac{\sqrt{P^2 + Q^2}}{h} \right] + \frac{gQ \sqrt{P^2 + Q^2}}{h^2 C^2} + n\psi_y = 0, \quad (3)$$

where Boussinesq dispersion term ψ_x and ψ_y are defined as:

$$\begin{aligned} \psi_x = & - \left(B + \frac{1}{3} \right) d^2 (P_{xxt} + Q_{xyt}) - nBg d^3 (\xi_{xxx} + \xi_{xyy}) \\ & - dd_x \left(\frac{1}{3} P_{xt} + \frac{1}{6} Q_{yt} + nBg d (2\xi_{xx} + \xi_{yy}) \right) \\ & - dd_y \left(\frac{1}{6} Q_{xt} + nBg d \xi_{xy} \right), \end{aligned} \quad (4)$$

$$\begin{aligned} \psi_y = & - \left(B + \frac{1}{3} \right) d^2 (Q_{yyt} + P_{xyt}) - nBg d^3 (\xi_{yyy} + \xi_{xxy}) \\ & - dd_y \left(\frac{1}{3} Q_{yt} + \frac{1}{6} P_{xt} + nBg d (2\xi_{yy} + \xi_{xx}) \right) \\ & - dd_x \left(\frac{1}{6} P_{yt} + nBg d \xi_{xy} \right), \end{aligned} \quad (5)$$

where the subscripts x , y and t denote partial differentiation with respect to space and time, respectively.

The horizontal stress terms are described using a gradient-stress relation, which reads:

$$F_x = - \left(\frac{\partial}{\partial x} \left(v_t \frac{\partial P}{\partial x} \right) + \frac{\partial}{\partial y} \left(v_t \left(\frac{\partial P}{\partial y} + \frac{\partial Q}{\partial x} \right) \right) \right), \quad (6)$$

$$F_y = - \left(\frac{\partial}{\partial y} \left(v_t \frac{\partial Q}{\partial y} \right) + \frac{\partial}{\partial x} \left(v_t \left(\frac{\partial Q}{\partial x} + \frac{\partial P}{\partial y} \right) \right) \right), \quad (7)$$

where v_t is the horizontal eddy viscosity.

The equations were solved using an implicit finite difference technique with variables defined on a space-staggered rectangular grid. We can get a sea surface elevation (total water depth, h) at every grid point and two flux (depth integrated velocity, P, Q) in x and y directions through every grid after each time step. We define sea surface elevations at a time step as a height field and the fluxes at a time step as a motion field. Because we can not solve it in real-time, we save the results in a period of time into a file.

4 Sea Surfaces Reordering

In order to render the finite sea surfaces in unlimited time continually, we developed a method similar to the one in [Schödl et al.

2000]. We consider the sea wave as a Markov process, with each state corresponding to a single sea surface, and the probabilities corresponding to the likelihood of transitions from one to another. Through reconstructing the sequence of the sea surface, we make the wave “timeless”.

The first step in creating the “timeless” sequence is to compute similarity between all pairs of sea surfaces. We define the similarity between two sea surfaces as:

$$D_{ij} = \frac{\|h_i - h_j\|_2}{\max(\|h_i - h_j\|_2)} + \delta \frac{\|P_i - P_j\|_2 + \|Q_i - Q_j\|_2}{\max(\|P_i - P_j\|_2 + \|Q_i - Q_j\|_2)}. \quad (8)$$

The first part in the right of the definition denotes the distance between two height fields and the second part in the right denotes the distance between two motion fields, the constant δ used to control the relative weight between shape similarity and dynamic similarity. We not only use the height fields but also the motion fields to measure the similarity. The reason is that the two height fields may be similar to each other, but their movement may be in opposite directions.

The next step is to compute the transition probability from one surface to another. The basic idea will be to create transition from surface i to surface j anytime the successor of i is similar to j —that is whenever $D_{i+1,j}$ is small. A simple way to do this is to map these similarities D to probabilities T through an exponential function:

$$T_{ij} \propto \exp(-D_{i+1,j}/\sigma), \quad (9)$$

where constant σ adjusts the mapping between similarity and probability. Smaller values of σ make better transitions, while larger values of σ allow for greater variety at the cost of poorer transitions. The probabilities in every row of T must normalized to ensure the $\sum_j T_{ij} = 1$. Further improvement of the probability matrix T can be considered similar to [Schödl et al. 2000].

The last step is to synthesize a new “timeless” sequence of sea surfaces. Two different algorithms can be used: random and loop. For random method, the rendering begins from any height field before the last non-zero transition probability. Then, the next height field is selected according to the transition probability. For loop method, you can select a sequence of sea surfaces with the largest transition probability and rendering repeatedly.

5 Sea Surface Rendering

Because our simulation regions are usually very wide, every height field is very large. In order to get fast rendering speed we don’t render the height fields as regular grid directly. We create the sea surface with view-dependent grid and use the height fields as vertex textures sampled by the sea surface.

5.1 Geometry of Sea Surface

We can use two kinds of view-dependent grids, one is a projected grid and another is a radial grid, as shown in Figure 3. In projected grid method, the vertexes of the grid are even-spaced in view plan and then project to sea level. In radial grid method, the center position of the radial grid is the camera position on sea level and it provides more detail closer to the viewer. They all have LOD effects. We recommend that the interested reader refer to [Johanson 2004] and [Kryachko 2005] for detailed explanations separately.

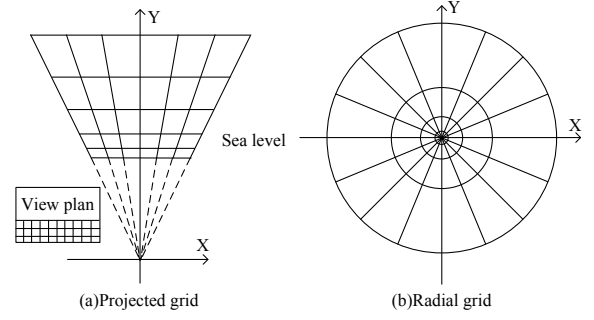


Figure 3: Projected grid and radial grid.

5.2 Color of Sea Surface

The color formulas of the sea surface in our method are:

$$C_{final} = S(h) * C_{surface} + (1 - S(h)) * C_{foam} \quad (10)$$

$$C_{surface} = F(\theta) * C_{reflect} + (1 - F(\theta)) * C_{refract} \quad (11)$$

$$C_{reflect} = C_{sky} + C_{sun} + C_{localreflect} \quad (12)$$

$$C_{refract} = (e^{-atten*dist}) * C_{object} + (1 - e^{-atten*dist}) * C_{water} \quad (13)$$

$$S(h) = \text{saturation} \left(\frac{h - h_0}{h_{max} - h_0} \right) \quad (14)$$

The terms used in these formulas are summarized in Table 2.

C_{final}	Final color of the sea surface
$C_{surface}$	Sea surface color without foam effect
C_{foam}	Color of the foam
$C_{reflect}$	Reflect color from outside of water
$C_{refract}$	Refract color from inside of water
C_{sky}	Color of the sky
C_{sun}	Height light color from the sun
$C_{localreflect}$	Reflect color from object near to camera
C_{object}	Color of object in water
C_{water}	Color of water itself
$S(h)$	Dense factor of foam
$F(\theta)$	Fresnel factor, saved in an 1D texture
h	Height of current pixel
h_0	Height begin to generate foam
h_{max}	Height at which foam is maximum
θ	Angle of view direction to current normal
$atten$	Attenuation factor
$dist$	Distance from sea surface to object in water

Table 2: Terms used in rendering.

6 Implementation Details

Here we give an overview of our implementation and provide some details about nontrivial steps. The method can be divided into two stages: pre-computing and real-time computing, see Figure 4.

In pre-computing stage, the main work is solving Boussinesq equations to get a sequence of height fields of the simulated region. In order to ensure the solving stable, water depth must be considered when setting wave parameters. The sponge layer is used to eliminate the wave and the porosity layer is used to let the wave transmission through the porous object. After a sequence of height fields

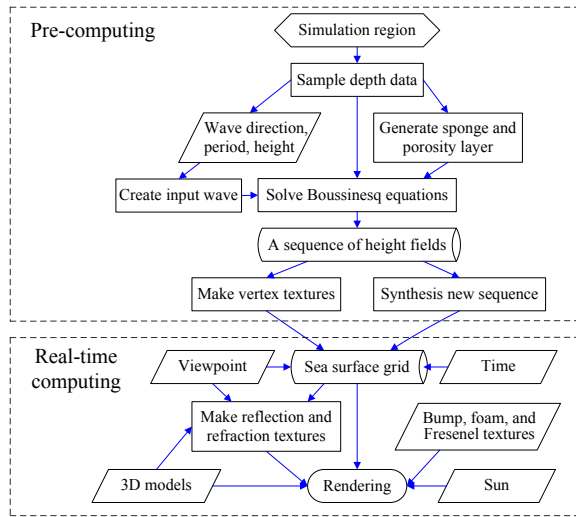


Figure 4: System overview diagram.

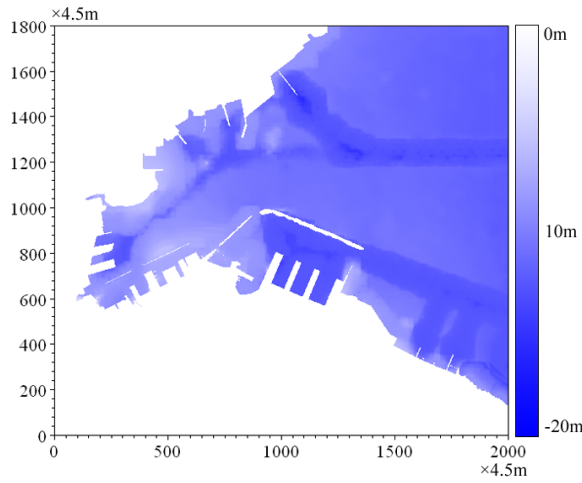


Figure 5: Water depth of the simulation region.

have been gotten, we make vertex textures from them and compute a new “timeless” sequence.

In real-time rendering stage, the view-dependent sea surface samples the vertex textures to get the height of vertexes. Trilinear filtering was used to ensure the visual quality when sampling. Two methods were used to simulate the reflect effect of the water surface. We use the cube map method to compute the reflect color of sky and object far to camera, and use the mirror reflection method to compute the reflect color of object near to camera. Bump textures are used to simulate the little wave on the sea surface. Foam textures are used to simulate the foam after wave break. The highlight effect of sun is also considered when rendering.

7 Results

This section shows an application of our method for simulating shallow-water waves in Dalian Port. The region of simulation is $9000\text{m} \times 8100\text{m}$, and the grid interval is 4.5m , the number of grid points is 2001×1801 . The ocean wave comes from east and the wave height is 4m , the wave period is 12s . The Figure 5 is the

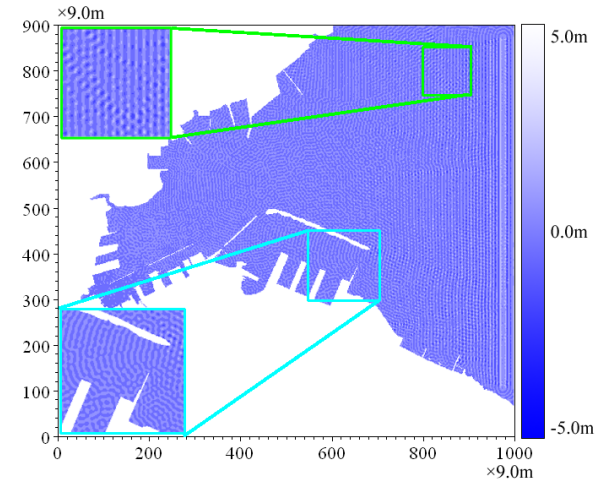


Figure 6: Height field generated by the regular wave.

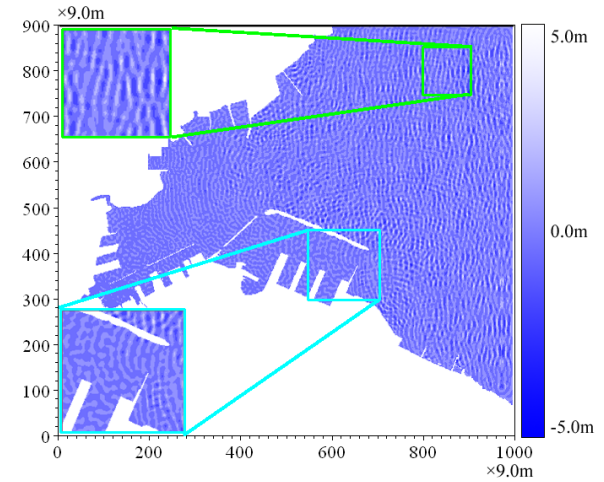


Figure 7: Height field generated by the irregular wave.

bathymetry of simulation region. The Figure 6 and Figure 7 are height fields of sea surface generated by regular wave and irregular wave respectively. It is clear to see the wave of refraction, reflection, diffraction and non-linear wave-wave interaction effect from the pictures.

We rendered the scene on a PC of Pentium 4 2.6GHz CPU, 1.5G RAM and a Geforce 8500GT graphics card. The frame rate is 126 fps when not rendering harbor object. The frame rate range from 120 fps to 20 fps according the number of objects in the view frustum. Figure 8 shows some frames captured from our system in different air conditions.

8 Conclusion

In this paper, we propose a new method of simulation large area of shallow-water waves in the coastal region. Using this method, the near-shore wave can be rendered with realistic effect in real-time. The Boussinesq equations guarantee the accuracy of the sea surface motion. By reordering the sequence of sea surface, we make the animation of wave “timeless”. The view-dependent sea surface grid and vertex textures make the system can render large area of sea surface in real-time.



Figure 8: Frames of rendering result in different weather.

However the method can not have the real effect of wave curling and breaking due to using 2D Boussinesq equations to get the height fields of the sea surface. How to introduce these effects to our system will be studied later.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments and the support by the 973 Program of China (No. 2002CB312103).

References

- ABBOTT, M. B., PETERSEN, H. M., AND SKOVGAARD, O. 1978. On the Numerical Modelling of Short Waves in Shallow Water. *Journal of Hydraulic Research* 16, 3, 173–203.
- CIEUTAT, J. M., GONZATO, J. C., AND GUITTON, P. 2001. A new efficient wave model for maritime training simulator. In *Proceedings of the 17th Spring conference on Computer graphics (SCCG'01)*, IEEE Computer Society, Budmerice, Slovakia, 202–209.
- ENRIGHT, D., MARSCHNER, S., AND FEDKIW, R. 2002. Animation and rendering of complex water surfaces. *ACM Transactions on Graphics* 21, 3, 736–744.
- FENG, M., SHA, W., AND ZHOU, S. 2003. Study on some numerical methods calculating waves in shore. *Marine Forecasts* 20, 1, 52–59.
- FOSTER, N., AND METAXAS, D. 1996. Realistic animation of liquids. *Graphical Models and Image Processing* 58, 5, 471–483.
- FOURNIER, A., AND REEVES, W. T. 1986. A simple model of ocean waves. *Computer Graphics* 20, 4, 75–84.
- GONZATO, J. C., AND SAËC, B. L. 1997. A phenomenological model of coastal scenes based on physical considerations. *Computer Animation and Simulation* 97, 137–148.
- GONZATO, J. C., AND SAËC, B. L. 2000. On modelling and rendering ocean scenes. *The Journal of Visualization and Computer Animation* 11, 1, 27–37.
- JOHANSON, C. 2004. *Real-time Water Rendering*. Master thesis, Lund University.
- KASS, M., AND MILLER, G. 1990. Rapid, stable fluid dynamics for computer graphics. *Computer Graphics* 24, 4, 49–57.
- KRYACHKO, Y. 2005. Using vertex texture displacement for realistic water rendering. In *GPU Gems 2*, M. Pharr, Ed., 1 ed., GPU Gems. Addison-Wesley Professional, 283–294.
- LAYTON, A. T., AND PANNE, M. V. D. 2002. A numerically efficient and stable algorithm for animating water waves. *The Visual Computer* 18, 41–53.
- MADSEN, P. A., AND SØRENSEN, O. R. 1992. A new form of the boussinesq equations with improved linear dispersion characteristics. part 2. a slowly-varying bathymetry. *Coastal Engineering* 18, 183–204.
- MADSEN, P. A., MURRAY, R., AND SØRENSEN, O. R. 1991. A new form of the boussinesq equations with improved linear dispersion characteristics. *Coastal Engineering* 15, 371–388.
- PEACHEY, D. R. 1986. Modeling waves and surf. *Computer Graphics* 20, 4, 65–74.
- SCHÖDL, A., SZELISKI, R., SALESIN, D. H., AND ESSA, I. 2000. Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques (SIGGRAPH'00)*, ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, New Orleans, Louisiana, 489–498.
- TAKAHASHI, T., FUJII, H., KUNIMATSU, A., HIWADA, K., SAITO, T., TANAKA, K., AND UEKI, H. 2003. Realistic animation of fluid with splash and foam. *Computer Graphics Forum* 22, 3, 391–400.
- TS'O, P. Y., AND BARSKY, B. A. 1987. Modeling and rendering waves: wave-tracing using beta-splines and reflective and refractive texture mapping. *ACM Transactions on Graphics* 6, 3, 191–214.
- XIE, C., JIN, Y., AND LIU, X. 2004. Real-time ocean wave in multi-channel marine simulator. In *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry (VRCAI'04)*, ACM Press New York, NY, USA, Singapore, 332–335.
- YIN, Y., JIN, Y., AND REN, H. 2003. Wave simulation of visual system in marine simulator based on wave spectrums. In *Proceedings of International Conference On Marine Simulation and Ship Maneuverability (MARSIM'03)*, vol. 1, RA–28–RA–28–8.