CIT. Journal of Computing and Information Technology, Vol. 25, No. 1, March 2017, 63–76
doi: 10.20532/cit.2017.1003327

Ocean Wave Rendering with Whitecap in the Visual System of a Maritime Simulator

Lining Chen[1], Yicheng Jin[2] and Yong Yin[2]

[1]School of Navigation, Guangzhou Maritime University, Guangzhou, China
[2]Laboratory of Marine Dynamic Simulation and Control, Dalian Maritime University, Dalian, China

The whitecap is an important oceanographic phenomenon. However, existing whitecap rendering methods do not successfully generate realistic whitecaps. To solve this problem, this paper presents a real-time whitecap rendering method applied to the visual system of a maritime simulator. The method takes the vertical acceleration on the wave crest as the criterion of whitecap generation. The Fourier coefficient of the vertical acceleration is provided, and a continuous mathematical model computing the whitecap coverage is built. The vertical acceleration is the variable of the model. The lifetime of the whitecap's existence can be controlled by the parameter of the model, and the parameter is computed with the genetic algorithm. The average of the computed whitecap coverage is equal to the whitecap coverage computed by the stochastic method and is close to the whitecap coverage computed by the empirical formula. The whitecap coverage is used as the blending factor to blend the pixel color of the whitecap texture and that of the sea surface. The presented method has sound theoretical support, with small computational complexity. The rendered whitecap is closer to the description of the Beaufort wind force scale than before.

ACM CCS (2012) Classification: Computing methodologies → Computer graphics → Animation → Physical simulation

Human-centered computing → Visualization → Visualization techniques → Graph drawings

Keywords: maritime simulator, Gerstner wave, ocean wave rendering, whitecap, whitecap coverage, Beaufort scale

1. Introduction

A maritime simulator, as defined by Det Norske Veritas (DNV), is a creation of certain conditions by means of a model to simulate situations within maritime operation [1]. It plays an important role in mariners' training and certification [2], [3]. It is also used in marine accident analysis and navigation security assessment. The full mission and multi-task simulators must have a visual system that simulates the actual nautical environment [1]: the ocean wave, sky, island, ship, and shore building should all be rendered in the system. The visual system uses computer graphics techniques to generate the nautical scene, which is projected onto the screen, simulating a realistic environment for the user [4].

The ocean wave creates the whitecap of bubbles and streaks on the breaking crest. During navigation, the mariner optically observes the size and shape of the ocean wave and whitecap to infer the wind velocity, wind direction and sea state, and properly maneuvers the ship according to the observed conditions [5]. Therefore, the whitecap is significant for the mariner, and a variety of research has been carried out on ocean wave rendering and animation. Although these works do improve the environmental realism of animation and increase the frame rate, research on whitecap rendering is still limited. Some research has been done on whitecap rendering, but these works have obvious flaws. For example, some methods of whitecap rendering lack theoretical support in marine engineering and stochastic wave theories, and the details of whitecap rendering methods in other references are too brief. Whitecap rendering is significant for improving the environmental realism of the rendered ocean wave, so ocean wave rendering methods without whitecap generation are not satisfactory.

To solve these problems, we present a whitecap rendering method used in the visual system of a marine simulator. The method takes the vertical acceleration on the wave crest as a criterion of whitecap generation. The whitecap coverage is computed by a continuous mathematical model with the acceleration as its variable. The computed whitecap coverage is used as the blending factor to blend the pixel color of whitecap texture and the pixel color of the sea surface. The presented method has theoretical support, and its computational complexity is small. The rendered whitecap is closer to the description of the Beaufort wind force scale and is similar to the whitecap in the photo of the actual sea.

2. Related Work

2.1 The Request of a Maritime Simulator on Ocean Wave Rendering

Whitecap rendering is a part of ocean wave rendering. Consequently, the whitecap rendering method depends on the ocean wave rendering method used.

To fulfill mariners' need, the visual system of the full mission and multi-task simulator should meet below requirements [1]:

- the frame rate should be higher than 30 f/s, i.e., rendering is real-time;
- ocean wave spectra should be used in ocean wave rendering;
- the rendered ocean wave should be variable in direction height and period;
- the visual system should provide a realistic set of wind waves including whitecaps according to the Beaufort wind force scale.

Because the visual system simulates large scale ocean waves, the selected rendering method should be fit for large scale ocean waves.

Different models can be used to render ocean waves: models from Computational Fluid Dynamics (CFD) [8] – [10], descriptions from video data [11], [12], Perlin noise [13], and the Gerstner wave. When the CFD model is used to animate large scale ocean waves, the computation is complex and the frame rate is low. The model described from video data is mainly used to rebuild the surface of the fluid in video. For the Perlin noise model, the rendered ocean wave cannot reflect the wind influence over the sea surface. The Gerstner wave model uses wave spectra and meets the above requirements [1], so we selected this model to render ocean waves in the visual system of maritime simulator.

2.2. Ocean Wave Rendering Based on the Gerstner Wave Model

The Gerstner wave model sums a variety of cosine waves to reproduce the shape of the incompressible sea surface [14], and the amplitude of the cosine wave can be computed with the wave spectrum [6].The expression of the model is briefly written as:

$$\begin{cases} \eta(\mathbf{x}_{pq}, t) = \sum_{i,j} A_{ij} \cos(\mathbf{k}_{ij} \cdot \mathbf{x}_{pq} - \omega t + \varphi_{ij}) \\ \mathbf{D}(\mathbf{x}_{pq}, t) = -\sum_{i,j} \frac{\mathbf{k}_{ij}}{k} A_{ij} \sin(\mathbf{k}_{ij} \cdot \mathbf{x}_{pq} - \omega t + \varphi_{ij}) \end{cases} \tag{1}$$

where

η is the elevation of the sea surface;

$\mathbf{x}_{pq} = (x_p, y_q)$ is the horizontal position vector;

t is time;

A_{ij} is the amplitude of the cosine wave;

$\mathbf{k}_{ij} = (k_{x,i}, k_{y,j}) = (k \cos\theta, k \cos\theta)$ is the wave number vector, and $k = |\mathbf{k}_{ij}|$;

ω is the circular frequency;

$\varphi_{ij} \in [0,2\pi]$ is an independent random number;

and $\mathbf{D}(\mathbf{x}_{pq}, t)$ is the Choppy wave vector.

Fournier *et al.* [15] introduced the Gerstner wave model to ocean wave graphics. Thon *et al.* [16] and Fréchot [17] employed the directional spectrum to compute the amplitude of the cosine wave. The spectrum is obtained from oceanographic observation, so the works of [16], [17] combined the observational oceanographic data with the Gerstner wave model. Some approaches to optimize the geometry of the sea surface include the perspective grid [18] and Level of Detail (LOD) technique [19], [20]. The Bidirectional Reflectance Distribution Function (BRDF) was adopted to make illumination more realistic [21]. Horvath [22] used the TMA spectrum to construct a directional spectrum, and the Gerstner wave model was used to render sea swell. Some approaches were tried to animate the interaction between the ocean wave and solid body, such as wave front tracking algorithm [23] and wave particle [24].

The direct use of the Gerstner wave model is very convenient. However, if a variety of cosine waves are involved, the number of direct summations is huge. Inverse Fast Fourier Transform (IFFT) is a cheaper alternative to direct summation, i.e., the IFFT Gerstner wave model. IFFT is efficiently sped up by GPU [25] and FPGA [26]. Tessendorf [27] presented the method in SIGGRAPH'99. The expression of the IFFT Gerstner wave model is written as

$$\begin{cases} \eta(\mathbf{x}_{pq},t) = \sum\limits_{i,j} \tilde{\eta}(\mathbf{k}_{ij},t)\exp(\sqrt{-1}\mathbf{k}_{ij}\cdot\mathbf{x}_{pq}) \\ \mathbf{D}(\mathbf{x}_{pq},t) = -\sum\limits_{i,j} \dfrac{\mathbf{k}_{ij}}{k}\tilde{D}(\mathbf{k}_{ij},t)\exp(\sqrt{-1}\mathbf{k}_{ij}\cdot\mathbf{x}_{pq}) \end{cases} \tag{2}$$

where

$\mathbf{x}_{pq} = (x_p, y_q) = (Lp/n, Lq/n)$;

$\mathbf{k}_{ij} = (k_{x,i}, k_{y,j}) = (2\pi i/L, 2\pi j/L)$;

L is the width of the rendered sea surface;

$p, q, i, j \in [-n/2, n/2]$ are integers;

$n = 2^m$, and m is a positive integer;

and $\sqrt{-1}$ is the imaginary unit.

$\tilde{\eta}(\mathbf{k}_{ij},t)$ and $\tilde{D}(\mathbf{k}_{ij},t)$ are, respectively, expressed as

$$\tilde{\eta}(\mathbf{k}_{ij},t) = \tilde{\eta}_0(\mathbf{k}_{ij})\exp(\sqrt{-1}\omega t) + \tilde{\eta}_0^*(-\mathbf{k}_{ij})\exp(-\sqrt{-1}\omega t) \tag{3}$$

$$\tilde{D}(\mathbf{k}_{ij},t) = \tilde{D}_0(\mathbf{k}_{ij})\exp(\sqrt{-1}\omega t) + \tilde{D}_0^*(-\mathbf{k}_{ij})\exp(-\sqrt{-1}\omega t) \tag{4}$$

where * is the notation for the complex conjugate. $\tilde{\eta}_0(\mathbf{k}_{ij})$ and $\tilde{D}_0(\mathbf{k}_{ij})$ are, respectively, written as

$$\tilde{\eta}_0(\mathbf{k}_{ij}) = \frac{1}{\sqrt{2}}(\varepsilon_1 + \sqrt{-1}\varepsilon_2)\sqrt{\Psi(\mathbf{k}_{ij})} \tag{5}$$

$$\tilde{D}_0(\mathbf{k}_{ij}) = \frac{1}{\sqrt{2}}(\varepsilon_2 - \sqrt{-1}\varepsilon_1)\sqrt{\Psi(\mathbf{k}_{ij})} \tag{6}$$

where both ε_1 and ε_2 are independent Gauss randoms, and $\Psi(\mathbf{k}_{ij})$ is the wave number spectrum. We use E and V to denote mean value and variance, respectively, $E(\varepsilon_1)=E(\varepsilon_2)=0$ and $V(\varepsilon_1)=V(\varepsilon_2)=0$.

Some approaches were employed to optimize the geometry of the IFFT sea surface, such as the perspective LOD grid [28], concentric circle grid [29] and spherical projected grid [30]. Wave number spectrum sampling was optimized [31], [32]. To add sea surface detail, texture mapping techniques, such as bumping mapping [33] and height map [34], were applied in rendering. Won *et al.* [35] introduced BRDF in illumination calculating. Some research focused on the interaction between the ship and the IFFT sea surface. The interaction mainly includes the ship wave and ship motion model. To simulate the ship wave, several approaches were tried, such as the moving grid with particle-based ship motion [36], Perfectly Matched Layers grid [37] and multi-LOD grid [38]. Additionally, Weerasinghe *et al.* [39] combined the ship motion model with the geometry of the IFFT sea surface.

The most frequently used wave number spectrum in rendering is the Phillips spectrum [27] – [39]. It is expressed as

$$\Psi_{\mathrm{Ph}}(\mathbf{k}) = A_{\mathrm{Ph}}\frac{1}{k^4}\exp\left(-\frac{g^2}{U^4k^2}\right)\cos^2(\theta-\alpha) \tag{7}$$

where A_{Ph} is the spectral constant, U is the wind velocity, and α is the angle between the wind direction and x axis.

Although the Phillips spectrum has widely been used, none of [27] – [39] provided the value of A_{Ph}. The author of this paper analyzed the structure of the spectrum and found that $A_{\mathrm{Ph}} = 3.48 \times 10$, and U, the wind velocity, should be $U_{19.5}$ [40]. However, when the computed results of [40] were applied in rendering, the amplitude of the generated ocean wave was malformed [41]. None of [27] – [39] gave the derivation of the IFFT Gerstner wave model. Consequently, the author of this paper provided complete derivation, and found that both expressions (5) and (6) omitted the area of the discrete integral domain, denoted by Δs_k. $\Delta s_k = (2\pi/L)^2$. To correct the malformation, expressions (5) and (6) should be written as

$$\tilde{\eta}_0(\mathbf{k}_{ij}) = \frac{1}{\sqrt{2}}(\varepsilon_1{}' + \sqrt{-1}\varepsilon_2{}')\sqrt{\Delta s_k \Psi(\mathbf{k}_{ij})} \tag{8}$$

$$\widetilde{D}_0(\mathbf{k}_{ij})=\frac{1}{\sqrt{2}}(\varepsilon_2{}'+\sqrt{-1}\varepsilon_1{}')\sqrt{\Delta s_k\Psi(\mathbf{k}_{ij})} \quad (9)$$

When expressions (8) and (9) are used, the amplitude malformation is corrected [41].

2.3. Whitecap Rendering

In marine engineering and stochastic wave theories, commonly used criteria of whitecap generation include the vertical acceleration on the wave crest, wave steepness, and horizontal velocity of the sea surface particle, etc [42].

As introduced above, we find that the existing research on Gerstner wave rendering focuses mainly on basic rendering methods [15], [17], wave spectrum sampling [16], [17], [31], [32], the sea surface geometry grid [18] – [20], [28] – [30], texture mapping [33], [34], illumination [21], [35] and the interaction between the ocean wave and the environment [24], [36] – [39]. Although many works have been carried out, research on whitecap rendering is limited. When the whitecap is rendered, commonly used criteria include the elevation of the sea surface, the Jacobian determinant, and the vertical acceleration of the wave crest.

Because elevation of the sea surface can be directly computed with expression (2), Grindstad *et al.* [43] take the elevation as a criterion of whitecap generation. When the elevation exceeds a threshold, the ocean wave breaks and the whitecap is generated [43]. The method is simple, and the computational complexity is small. However, it has some flaws:

- taking the elevation as a criterion lacks theoretical support in marine engineering and stochastic wave theories;
- the value of the threshold is not provided;
- the shape of the rendered whitecap obviously differs from that of the actual whitecap.

If the Jacobian determinant is negative, the surface of the sea surface passes through itself and inverts. This is an effect to signal the production of breaking and whitecap [27]. The Jacobian determinant is written as:

$$J(\mathbf{x}_{pq},t)=J_{xx}(\mathbf{x}_{pq},t)J_{yy}(\mathbf{x}_{pq},t) \\ -J_{xy}^2(\mathbf{x}_{pq},t). \quad (10)$$

$$\begin{cases} J_{xx}(\mathbf{x}_{pq},t)=1+\lambda\partial D_x(\mathbf{x}_{pq},t)/\partial x, \\ J_{yy}(\mathbf{x}_{pq},t)=1+\lambda\partial D_y(\mathbf{x}_{pq},t)/\partial y, \\ J_{xy}(\mathbf{x}_{pq},t)=\lambda\partial D_x(\mathbf{x}_{pq},t)/\partial y. \end{cases} \quad (11)$$

Obviously, IFFT should be individually performed when $\partial D_x(\mathbf{x}_{pq},t)/\partial x$, $\partial D_y(\mathbf{x}_{pq},t)/\partial y$ and $\partial D_x(\mathbf{x}_{pq},t)/\partial y$ are computed. This means that three times of IFFT are needed for one computation of $J(\mathbf{x}_{pq},t)$.

Wang *et al.* [44] used the Jacobian determinant to detect the spray position, and generated particle spray. Dupuy *et al.* [45] employed the Jacobian determinant to compute the whitecap coverage which is used to compute the sea surface shading. Advantages of the method adopted by [45] are clear:

- the Jacobian determinant can test the overlapping area on the sea surface;
- the Jacobian determinant is used to compute the whitecap coverage, the coverage is employed in sea surface shading, and the shaded color of the generated whitecap changes gradually.

The method also has some flaws:

- the computational complexity is great: three times of IFFT are needed for one computation of the Jacobian determinant;
- taking the Jacobian determinant as a criterion lacks theoretic support in marine engineering and stochastic wave theories, and the rendered whitecap is obviously different from the actual whitecap;
- the method using the Jacobian determinant to test wave breaking suffers from temporal instability [46].

Although the flaws are obvious, the approach that the whitecap coverage is used in sea surface shading is novel, effective and practical. The approach combines the whitecap coverage with texture blending, producing better rendered results. Therefore, it is worth learning and using. We also employ the approach in whitecap rendering, but the variable that we use to compute the whitecap coverage should be a physical quantity that is a criterion in marine engineering and stochastic wave theories.

In deep water, if the wave steepness exceeds the limit value, the ocean wave is unstable, and wave breaking and whitecap creation occur [42], [47]. However, the value is difficult to observe [48]. Therefore, some alternative criteria are often used. When the steepness exceeds the limit value, the vertical acceleration on the wave crest crosses the limit acceleration, thus the vertical acceleration is also taken as a criterion [42], [47]. Schwenger *et al.* [49] rendered the whitecap, taking the vertical acceleration as a criterion. Unfortunately, the detail of the whitecap rendering method was not provided.

3. Whitecap Rendering Method Taking the Vertical Acceleration as a Criterion

The approach of [45] combines the whitecap coverage with texture blending, rendering a gradually changing whitecap that inspires us to further develop our method. Since the vertical acceleration is taken as a criterion of whitecap generation [42], [47], we may use the acceleration to compute the whitecap coverage, which is applied in whitecap rendering. Therefore, we have to overcome two problems:

- how to conveniently compute the vertical acceleration?
- how to use the vertical acceleration to compute the whitecap coverage which will improve rendering results?

Both problems are solved in this section.

3.1. Computing the Vertical Acceleration on the Wave Crest

We use $a_z(\mathbf{x}_{pq},t)$ to denote vertical acceleration of the wave crest. $a_z(\mathbf{x}_{pq},t)$ is expressed as

$$a_z(\mathbf{x}_{pq},t) = \mathrm{d}^2\eta(\mathbf{x}_{pq},t)/\mathrm{d}t^2 = -\sum_{i,j} A_{ij}\omega^2 \cos(\mathbf{k}_{ij}\cdot\mathbf{x}_{pq} - \omega t + \varphi_{ij}) \tag{12}$$

To conveniently compute $a_z(\mathbf{x}_{pq},t)$ we need to write the expression of $a_z(\mathbf{x}_{pq},t)$ in the IFFT form. First, we write expression (12) in the complex form as:

$$a_z(\mathbf{x}_{pq},t) = -\frac{1}{2}(C_1 + C_2) \tag{13}$$

where

$$C_1 = \sum_{i=-N/2}^{N/2-1}\sum_{j=-N/2}^{N/2-1} \omega^2 \exp\left(\sqrt{-1}(\mathbf{k}_{ij}\cdot\mathbf{x}_{pq})\right)A_{ij} \times \exp\left(\sqrt{-1}(\varphi_{ij} - \omega t)\right),$$

$$C_2 = \sum_{i=-N/2}^{N/2-1}\sum_{j=-N/2}^{N/2-1} \omega^2 \exp\left(-\sqrt{-1}(\mathbf{k}_{ij}\cdot\mathbf{x}_{pq})\right)A_{ij} \times \exp\left(\sqrt{-1}(\omega t - \varphi_{ij})\right).$$

$-\mathbf{k}_{ij}=\mathbf{k}_{-i,-j}$ and $A_{ij} = \sqrt{2\Psi(\mathbf{k}_{ij})\Delta s_{\mathbf{k}}}$ [40], [42], so expression (13) can be written in the IFFT form as:

$$a_z(\mathbf{x}_{pq},t) = -\sum_{i,j} \tilde{a}_z(\mathbf{k}_{ij},t)\exp(\sqrt{-1}\mathbf{k}_{ij}\cdot\mathbf{x}_{pq}) \tag{14}$$

$$\tilde{a}_z(\mathbf{k}_{ij},t) = \tilde{a}_{z,0}(\mathbf{k}_{ij})\exp(\sqrt{-1}\omega t) + \tilde{a}^*_{z,0}(-\mathbf{k}_{ij})\exp(-\sqrt{-1}\omega t) \tag{15}$$

$$\tilde{a}_{z,0}(\mathbf{k}_{ij}) = \frac{1}{\sqrt{2}}\omega^2(\varepsilon_2{}' + \sqrt{-1}\varepsilon_1{}')\sqrt{\Delta s_k \Psi(\mathbf{k}_{ij})} \tag{16}$$

where $\tilde{a}_z(\mathbf{k}_{ij},t)$ is the Fourier coefficient of $a_z(\mathbf{x}_{pq},t)$. Therefore, $a_z(\mathbf{x}_{pq},t)$ can be computed by IFFT, and only one time of IFFT is needed in one computation of $a_z(\mathbf{x}_{pq},t)$. Therefore, computing is less expensive than computing $J(\mathbf{x}_{pq},t)$, the Jacobian determinant.

3.2. Computing the Whitecap Coverage

The whitecap coverage is defined as the area of whitecaps of unit sea surface [42]. In marine engineering and stochastic wave theories, the whitecap coverage is deemed as a stochastic feature and can be computed by the empirical formula and the stochastic model.

By observing ocean waves with some instruments, researchers have worked out the empirical formula. Such instruments include aerial cameras [50], the Coastal Zone Color Scanner

(CSCZ) [51], and digital cameras [52]. The observed data is analyzed, and the empirical formula is obtained based on the observational data. The commonly used empirical formulas include [50] – [52]:

$$W_o = 4.4\times10^{-4}U_{10}^2 \quad (17)$$

$$W_o = 2.96\times10^{-6}U_{10}^{3.75} \quad (18)$$

$$W_o = 1.90\times10^{-6}U_{10}^{3.51} \quad (19)$$

where W_0 is the whitecap coverage computed using the empirical formula, U_{10} is the wind velocity at the height of 10.0 m above the sea surface, and $U_{10} \approx 0.9016\, U_{19.5}$.

The whitecap coverage can also be computed using the stochastic model, i.e., the cumulative distribution function of the concerned ocean wave parameter is employed to compute the whitecap coverage. According to marine engineering and stochastic wave theories, the vertical acceleration, denoted by a_z, is a criterion of whitecap generation. If $a_z \le -\beta g$, wave breaking occurs and the whitecap is generated. $\beta = 0.388 \approx 0.39$ [42]. Expression (12) indicates that a_z distributes normally, $E(a_z) = 0$, and $V(a_z) = m_4$. m_4 is the 4th order spectral moment. The whitecap coverage computed by the stochastic model can be expressed as:

$$W_{a_z} = \int_{-\infty}^{-\beta g} f(a_z)\mathrm{d}a_z = 1 - \Phi\left(\beta g / m_4^{1/2}\right) \quad (20)$$

where $f(x)$ is the probability density function of a_z, and $\Phi(x)$ is the cumulative distribution function of the standard normal distribution.

If the wind parameter is fixed, both W_e and W_{a_z} are constants.

Dupuy *et al.* [45] used the whitecap coverage as a random value, and the computed whitecap coverage was used as the blending factor to blend the pixel color of the whitecap texture and that of the sea surface. This approach makes the shaded color of the rendered whitecap change gradually, and the rendered result is encouraging, thus we also use the approach. Let C be the shaded color of the sea surface after texture blending, which is expressed as

$$C = (1 - W_R)C_F + W_R C_W \quad (21)$$

where C_F and C_W, respectively, denote the pixel color of the whitecap-free sea surface and the whitecap texture, W_R is the whitecap coverage, and $W_R \in [0,1]$. W_R is a random variable and must be computed in real-time. When W_R is computed, the following aspects should be considered:

- the variable used to compute W_R should be a reasonable physical quantity, so that the computation of W_R has sound theoretical support;
- the variable is conveniently computed, and the computation does not reduce the frame rate;
- $E(W_R)$ should be close to the whitecap coverage in marine engineering and stochastic wave theories. Thus the shape and area of the rendered whitecap should vary according to the wind parameter, and the shape and area of the rendered whitecap will reflect the wind's influence over the sea surface.

Considering these aspects, we use a_z to compute W_R.

3.3. Continuous Mathematical Model of the Whitecap Coverage Function in Rendering

Figure 1 is the diagram of η and a_z of a sea surface particle. In the figure, the particle is on the crest when $t = t_1$, and falls to the trough. It should be noted that a_z increases monotonically when the particle moves from the crest to the trough. If $a_{z,1} \le -\beta g$, the whitecap is generated and W_R is positive when $t = t_1$. In regard to $t = t_4$, the whitecap disappears and W_R is 0. During the process, W_R changes from positive to 0.

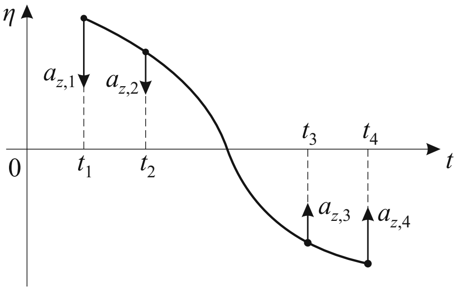


Figure 1. Diagram of η and a_z of a sea surface particle.

As mentioned above, if $E(W_{\mathrm{R}})$ is close to the whitecap coverage in marine engineering and stochastic wave theories, the shape and area of the rendered whitecap will reflect the wind's influence over the sea surface, and the rendering result will be satisfactory. Therefore, the model of the whitecap coverage function in rendering can be built on the condition that $E(W_{\mathrm{R}}) = W_{a_z}$.

We have built a discontinuous mathematical model to compute W_{R}. The discontinuous model is expressed as

$$W_{\mathrm{R}} = \begin{cases} 0, \text{ if } a_z \in \langle -\beta \mathrm{g}, \infty \rangle \\ 1, \text{ if } a_z \in \langle -\infty, -\beta \mathrm{g}] \end{cases} \tag{22}$$

where $\beta = 0.39$. If W_{R} is computed by expression (22), $E(W_{\mathrm{R}}) = W_{a_z}$. When the model is used in rendering, C is equal to either C_{F} or C_{W}, the shaded color of the generated whitecap changes suddenly, and the rendered result is not desirable. So a continuous model may be a better choice.

To improve the rendered results, we have built a continuous mathematical model to compute W_{R}. Let α_1 and α_2 be positive, $\alpha_1 \in \langle 0,\beta \rangle$ and $\alpha_2 \in \langle \beta,\infty \rangle$. Our strategy is as follows:

- if $a_z \in \langle -\infty, -\alpha_2 \mathrm{g}]$, $W_{\mathrm{R}} = 1$, and $C = C_{\mathrm{F}}$;
- when $a_z \in [-\alpha_1 \mathrm{g}, \infty\rangle$, $W_{\mathrm{R}} = 0$ and $C = C_{\mathrm{W}}$;
- if $a_z \in \langle -\alpha_2 \mathrm{g}, -\alpha_1 \mathrm{g} \rangle$, C is computed by blending C_{F} and C_{W}, $W_{\mathrm{R}} \in \langle 0,1 \rangle$, and W_{R} is the blending factor.

The shaded color of the rendered whitecap changes gradually, and the rendered result is improved. The continuous mathematical model is expressed as

$$W_{\mathrm{R}} = \begin{cases} 0, \text{ if } a_z \in [-\alpha_1 \mathrm{g}, \infty \rangle \\ f_1(a_z), \text{ if } a_z \in \langle -\alpha_2 \mathrm{g}, -\alpha_1 \mathrm{g} \rangle \\ 1, \text{ if } a_z \in \langle -\infty, -\alpha_2 \mathrm{g}] \end{cases} \tag{23}$$

$$f_1(a_z) = \beta_1 \left(\exp\left(\frac{-a_z - \alpha_1 \mathrm{g}}{2m_4} \right) - 1 \right) \tag{24}$$

where $\beta_1 = \left(\exp\left(\frac{\alpha_2 \mathrm{g} - \alpha_1 \mathrm{g}}{2m_4} \right) - 1 \right)^{-1}$. In expression (24), α_1 and α_2 control the range of a_z, and β_1 limits W_{R} to between 0 and 1. Figure 2 shows the curve of the continuous mathematical model. $E(W_{\mathrm{R}})$ is expressed as

$$E(W_{\mathrm{R}}) = \int_{-\infty}^{-\alpha_2 \mathrm{g}} 1 \cdot f(a_z) \mathrm{d}a_z + \int_{-\alpha_2 \mathrm{g}}^{-\alpha_1 \mathrm{g}} f_1(a_z) f(a_z) \mathrm{d}a_z + \int_{-\alpha_1 \mathrm{g}}^{\infty} 0 \cdot f(a_z) \mathrm{d}a_z \tag{25}$$

If $E(W_{\mathrm{R}}) = W_{a_z}$, we obtain

$$\int_{-\alpha_2 \mathrm{g}}^{-\alpha_1 \mathrm{g}} f_1(a_z) f(a_z) \mathrm{d}a_z = \int_{-\alpha_2 \mathrm{g}}^{-\beta \mathrm{g}} f(a_z) \mathrm{d}a_z$$

$$C_1 C_2 - C_3 = \Phi(\alpha_2 \mathrm{g} / m_4^{1/2}) - \Phi(\beta \mathrm{g} / m_4^{1/2}) \tag{26}$$

where

$$C_1 = \beta_1 \exp\left(\left(-\alpha_1 \mathrm{g} + 1/4 \right) / \left(2m_4 \right) \right);$$
$$C_2 = \Phi\left(\left(\alpha_2 \mathrm{g} + 1/2 \right) / m_4^{1/2} \right) - \Phi\left(\left(\alpha_1 \mathrm{g} + 1/2 \right) / m_4^{1/2} \right);$$
$$C_3 = \beta_1 \left(\Phi\left(\alpha_2 \mathrm{g} / m_4^{1/2} \right) - \Phi\left(\alpha_1 \mathrm{g} / m_4^{1/2} \right) \right).$$

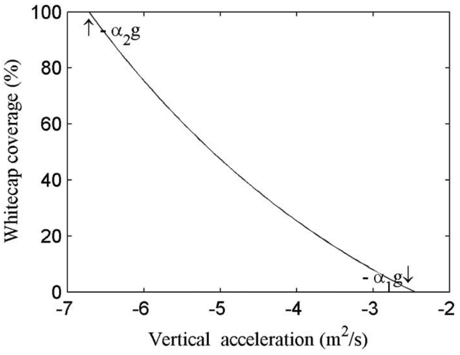


Figure 2. Curve of the continuous mathematical model.

α_1 and α_2 can be computed from expression (27). To facilitate the computation, we set α_1 as a constant, and compute the corresponding values of α_2 and β_1. It is difficult to obtain the analytical solution for α_2, so the optimization algorithm should be employed to compute the optimum solution of α_2. We use the Standard Genetic Algorithm (SGA) [54] to compute the optimum solution. The objective function of SGA is written as:

$$f_{\mathrm{O}}(\alpha_2) = \min_{\alpha_2 \in [\beta, 1.5]} \left| C_1 C_2 - C_3 - \Phi\left(\alpha_2 \mathrm{g} / m_4^{1/2}\right) \right. \\ \left. + \Phi\left(\beta \mathrm{g} / m_4^{1/2}\right) \right| \qquad (27)$$

The parameters of the SGA are set in Table 1. Figure 3 is the evolution process of the optimum solution. Table 2 gives α_1, α_2 and β_1 computed by the SGA. The smaller α_1 is, the longer the lifetime of the whitecap. Our rendering method does the SGA with off-line computation, so the frame rate is not affected by the SGA.

Table 1. Parameters of SGA.

Parameter	Value
Population size	40
Maximum generation	60
Genes value	20
Generation gap	0.6
Crossover probability	0.7
Mutation probability	0.9

Table 2. α_1, α_2 and β_1 computed by SGA (β = 0.39).

$U_{19.5}$ (m/s)	α_1	α_2	$\alpha_2 - \alpha_1$	β_1
10	0.250	0.6849	0.4349	0.5747
	0.300	0.5254	0.2254	1.4574
15	0.250	0.6058	0.3558	1.4455
	0.300	0.4904	0.1904	3.0775
20	0.250	0.5066	0.2566	3.7401
	0.300	0.4742	0.1742	5.7301

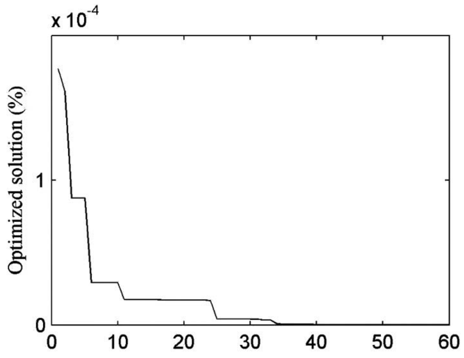


Figure 3. Evolution process of the optimum solution.

4. Rendered Results

The Beaufort wind force scale, or the Beaufort scale, is an empirical measure that relates to observed conditions at sea or on land. The scale is widely used in navigation and meteorology. Table 3 is part of the Beaufort scale. The P-M spectrum is used to compute the spectral constant of the Phillips spectrum [40]. The wind velocity of the P-M and Phillips spectra is $U_{19.5}$, in the range of 10.0 ~ 20.0 m/s. Therefore, the Phillips spectrum is effective at scale numbers 5, 6 and 7.

The 3D graphics tool of the maritime simulator's visual system is OpenSceneGraph (OSG). OSG uses a tree-structure to manage the graph data [55]. The sea surface is connected to the scene tree as a geode, i.e., leaf node, and the sea surface geometries are connected to the geode. The number of the geometry vertex depends on the distance between the position of the observer's eye and the center of the bounding box of the geometry. The longer the distance, the lower the number of the geometry vertex, and the shorter the distance, the higher the number of the geometry vertex. G_{ij}, $\mathbf{x}_{\mathrm{eye}}$ and $\mathbf{c}_{\mathrm{eye}}$ denote the sea surface geometry, observer's eye and the center of the bounding box of the geometry, and the geometric grid of the rendered sea surface is shown in Figure 4. We use 35 × 35 geometries to construct the sea surface. The size of the geometry is 512 m × 512 m, and the number of the geometry vertex closest to the position of the observer's eye is 256 × 256. The program is implemented on a PC with a Geforce GTX 660 GPU, Intel Core(TM) i5 – 4460 CPU and 4 GB memory, and the frame rate is 71 f/s. Figure 5 shows the rendered results of our method.

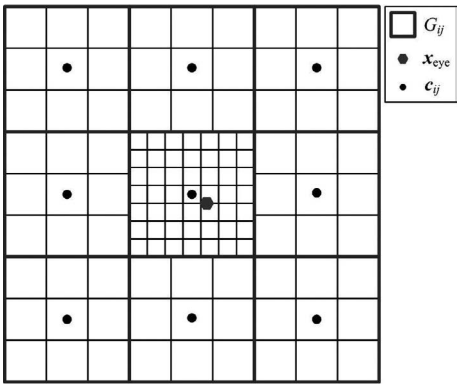
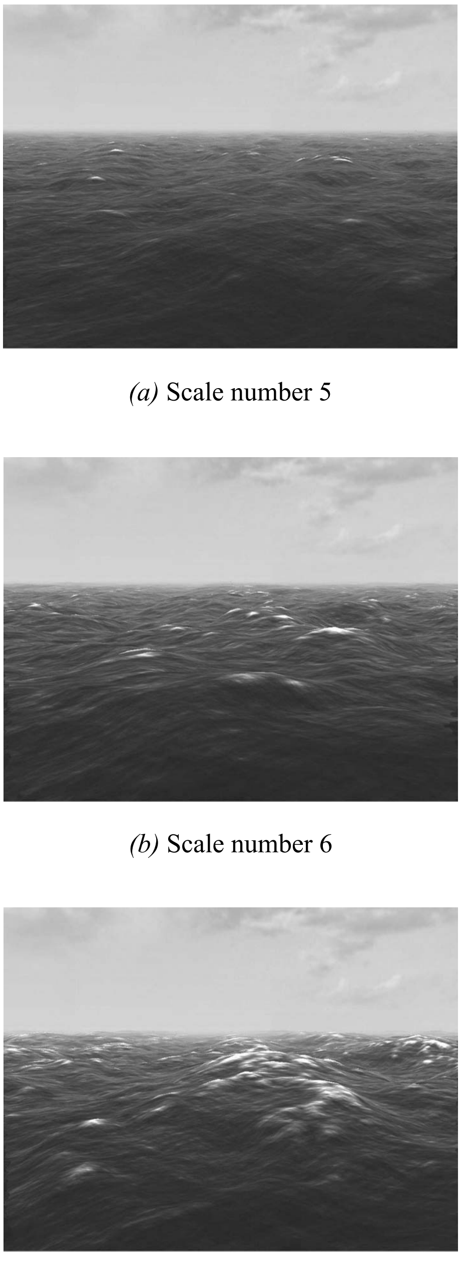


Figure 4. Geometric grid of the rendered sea surface.

Table 3. Beaufort scale (scale number 5 ~ 7).

Scale number	Description	Wind velocity (U_{10})	Sea conditions
5	Fresh breeze	8.0 ~ 10.7 m/s	Moderate waves of some length. Many whitecaps. Small amounts of spray.
6	Strong breeze	10.8 ~ 13.8 m/s	FLong waves begin to form. White foam crests are very frequent. Some airborne spray is present.
7	Near gale	13.9 ~ 17.1 m/s	Sea heaps up. Some foam from breaking waves is blown into streaks along wind direction. Moderate amounts of airborne spray.



(a) Scale number 5

(b) Scale number 6

(c) Scale number 7

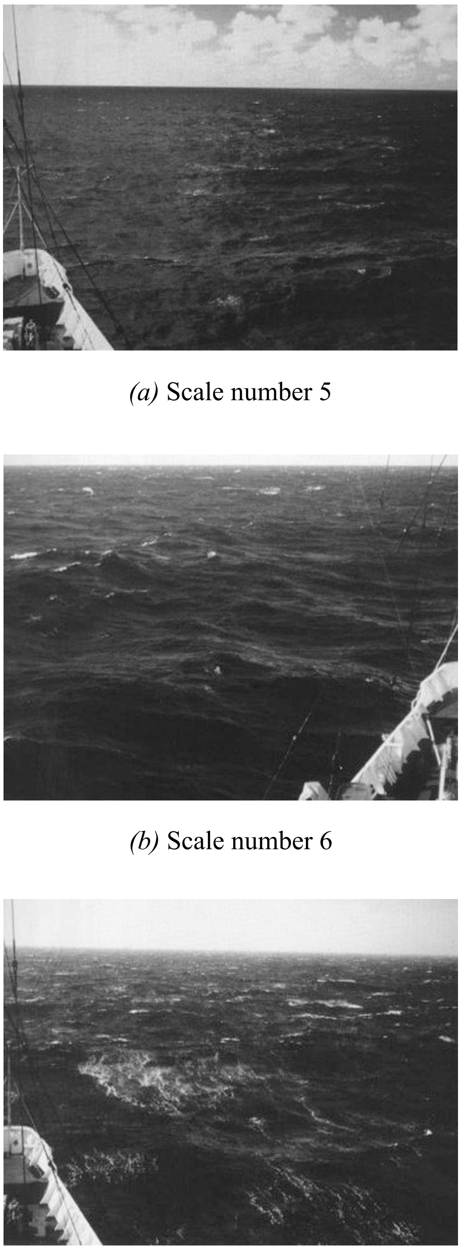
Figure 5. Rendered results of the method raised in the paper.

5. Rendered Result Analysis and Comparison

We compare our rendered results with the whitecaps in photos of the actual sea surface and the whitecap described by the Beaufort scale. Figure 6 includes the actual surface [5]. Observing Figure 5, we note that the shape of the whitecap rendered with our method varies according to the wind velocity. In Figure 5(a), the rendered result of scale number 5, there are many whitecaps over the sea surface. In Figure 5(b), the rendered result of scale number 6, we note that white foam crests are very frequent. In Figure 5(c), the rendered result of scale number 7, we find that some foam from breaking waves is blown into streaks in the direction of the wind. Obviously, the rendered result of our method is similar to the actual whitecap, and is in accordance with the Beaufort scale. Figure 7 shows the rendered results of the open source code offered by [45]. We note that the whitecap in Figure 7 is distinctly different from the actual whitecap in Figure 6. Figure 8 shows the rendered results of [43], [49]. Obviously, the rendered whitecap in Figure 8(a) enormously differs from the actual whitecap. The whitecap in Figure 8(b) is similar to our method, but [49] did not provide details of the whitecap rendering method. Figure 9 shows curves of the whitecap coverage. It demonstrates that $E(W_R)$ is close to the whitecap coverage computed as the empirical formula obtained by observation [50] – [52].

6. Conclusions

In whitecap rendering based on the Gerstner wave model, the most frequently used criteria of whitecap generation include the elevation of

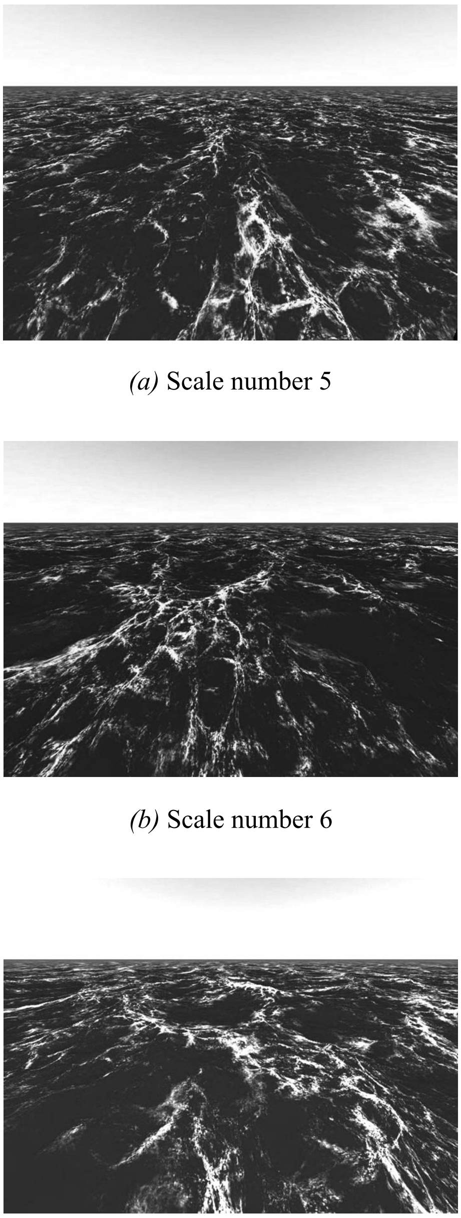


(a) Scale number 5

(b) Scale number 6

(c) Scale number 7

Figure 6. Photos of the actual sea surface.



(a) Scale number 5

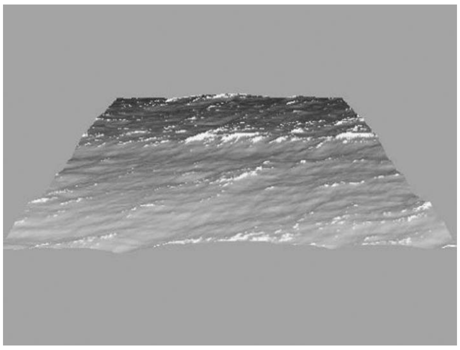
(b) Scale number 6

(c) Scale number 7

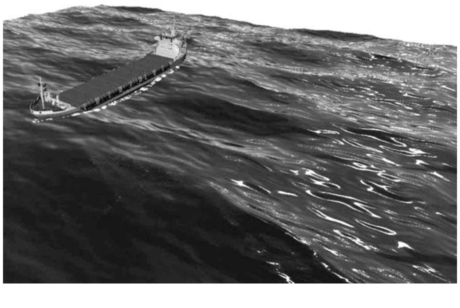
Figure 7. Rendered results of the method in [45].

the sea surface [43], the Jacobian determinant [44], [45] and the vertical acceleration of the wave crest [49]. The method taking the elevation or Jacobian determinant as a criterion lacks theoretical support in marine engineering and stochastic wave theories, and the rendered results differ enormously from the actual whitecap. Additionally, computing the Jacobian determinant is expensive. The method taking the vertical acceleration as a criterion has sound theoretical support, but [49] did not provide details of the whitecap rendering method.

To solve these problems, we present a whitecap rendering method applied to the visual system of a maritime simulator. The method takes the vertical acceleration of the wave crest as a criterion of whitecap generation. We provide the Fourier coefficient of the vertical acceleration, so the vertical acceleration can be computed in real time. Then, a continuous mathematical model computing the whitecap coverage is built. Vertical acceleration is the variable of the continuous model, and the parameter of the model is computed by the SGA. The lifetime of the whitecap's existence is controlled by the parameter in the model. The average of the computed whitecap coverage is equal to the whitecap coverage computed by the stochastic model and is close to the whitecap coverage computed by the empirical formula that is obtained from observational data. The computed whitecap coverage is used as the blending factor to blend the pixel color of the whitecap texture and that of the sea surface, and the whitecap is real-time rendered and animated. The computation of the vertical acceleration is less expensive than the computation of the Jacobian determinant. Comparisons show that the rendered whitecap of our method is in accordance with both the whitecap description of the Beaufort scale and the actual whitecap. Our method that takes the vertical acceleration of the wave crest as a criterion has sound theoretical support, and the rendered whitecap meets the concerned requirements raised by the DNV.



(a) Rendered result of [43]



(b) Rendered result of [49]

Figure 8. Rendered results of [43], [49].

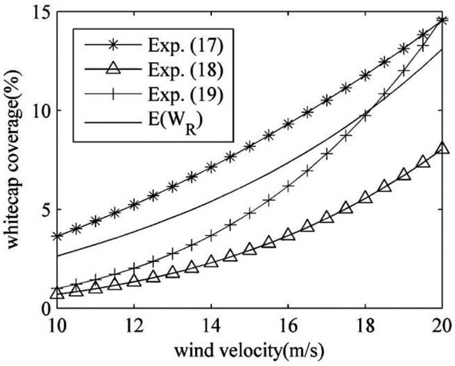


Figure 9. Curves of the whitecap coverage.

Acknowledgement

The work is supported by the National High Technology Research of China (863 Project) (No. 2015AA016404) and the Fundamental Research Funds for the Central Universities of China (No. 3132013302).

References

[1] Det Norske Veritas, "Standard for Certification No. 2.14 Maritime Simulator Systems", 2011. https://rules.dnvgl.com/docs/pdf/DNV/stdcert/2011-01/Standard2-14.pdf

[2] A. Weintrit, "Marine Navigation and Safety of Sea Transportation: Advances in Marine Navigation", *CRC Press*, London, 2013, pp. 95–145. http://dx.doi.org/10.1201/b14961

[3] The Nautical Institute, "Dynamic Positing Operator's Training and Certification Scheme" (version 2.1), 2012. http://www.nialexisplatform.org/media/1351/dp-operator-certificate-information-jul-2012.pdf

[4] Y. Zan *et al.*, "Research on Real-Time Simulation System of Ship Motion Based on Simulink", *The Open Mechanical Engineering Journal*, vol. 8, pp. 820–827, 2014. http://dx.doi.org/10.2174/1874155X01408010820

[5] N. Bowditch, "The American Practical Navigator", Bicentennial Edition, Bethesda, MD: National Imagery and Mapping Agency, 2002, pp. 537–544.

[6] COST Action 714 Working Group 3, "Measuring and Analyzing the Directional Spectrum of Ocean Waves", Luxembourg, Office for Official Publications of the European Communities, 2005, pp. 16–45.

[7] E. Darles *et al.*, "A survey of Ocean Simulation in Computer Graphics", *Computer Graphics Forum*, vol. 30, no.1, pp. 43–60, 2011.
http://dx.doi.org/10.1111/j.1467-8659.2010.01828.x

[8] Q. Wang *et al.*, "Efficient Rendering of Breaking Waves Using MPS Method", *Journal of Zhejiang University, Science A*, vol. 7, no. 6, pp. 1018–1025, 2006.
http://dx.doi.org/10.1631/jzus.2006.A1018

[9] J. Dominguez *et al.*, "New Multi-GPU Implementation for Smoothed Particle Hydrodynamics on Heterogeneous Clusters", *Computer Physics Communications*, vol. 184, no.8, pp. 1848–1860, 2013.
http://dx.doi.org/10.1016/j.cpc.2013.03.008

[10] J. Ojeda and A. Susin, "Real-Time Lattice Boltzmann Shallow Waters Methods for breaking Wave simulations", *Communications in Computer and Information Science*, vol. 458, pp. 3–18, 2014.

[11] G. B. Gallego, "Variational Image Processing Algorithms for the Stereoscopic Space-Time Reconstruction of Water Waves", Ph.D Thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 2011.

[12] C. Li *et al.*, "Water Surface Modeling from a single Viewpoint Video", *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 7, pp. 1242–1251, 2013.
http://dx.doi.org/10.1109/TVCG.2012.302

[13] A. Puig-Centelles *et al.*, "View-Dependent Tessellation and Simulation of Ocean Surfaces", *The Scientific World Journal*, vol. 2014, article no. 979418, 2014.
http://dx.doi.org/10.1155/2014/979418

[14] D. D. A. Craik, "The Origins of Water Wave Theory", *Annual Review of Fluid Mechanics*, vol. 36, pp. 1–28, 2004.
http://dx.doi.org/10.1146/annurev.fluid.36.050802.122118

[15] A. Fournier and W. Reeves, "Simple Model of Ocean Waves", *Computer Graphics*, vol. 20, no. 4, pp. 75–84, 1986.
http://dx.doi.org/10.1145/15886.15894

[16] S. Thon and D. Ghazanfarpour, "Ocean Waves Synthesis and Animation Using Real World Information", *Computers & Graphics*, vol. 26, no.1, pp. 99–108, 2002.
http://dx.doi.org/10.1016/S0097-8493(01)00161-3

[17] J. Frechot, "Realistic Simulation of Ocean Surface Using Wave Spectra", in *Proceedings of the 1st International Conference on Computer Graphics Theory and Applications*, Setubal, Portugal: INSTICC, 2006, pp. 76–83.
http://dx.doi.org/10.1155/2014/979418

[18] D. Hinsinger *et al.*, "Interactive Animation of Ocean Waves", in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, New York, ACM Press, 2002, pp. 161–166.
http://dx.doi.org/10.1145/545261.545288

[19] X. Zhao *et al.*, "Ocean Wave Simulation near the Seashore", *Journal of Beijing Institute of Technology*, vol.18, no. 2, pp. 181–185, 2009.
http://dx.doi.org/10.15918/j.jbit1004-0579.2009.02.019

[20] M. N. Fan and N. Zhang, "Simulation Method of Random Ocean Waves Based on Fractal Interpolation", *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 6, no. 6, pp. 411–420, 2013.
http://dx.doi.org/10.14257/ijsip.2013.6.6.37

[21] E. Bruneton *et al.*, "Real-Time Realistic Ocean Lighting Using Seamless Transitions from Geometry to BRDF", *Computer Graphics Forum*, vol. 29, no. 2, pp. 487–496, 2010.
http://dx.doi.org/10.1111/j.1467-8659.2009.01618.x

[22] C. J. Horvath, "Empirical Directional Wave Spectra for Computer Graphics", in *Proceedings of the 2015 Symposium on Digital Production*, New York, ACM Press, 2015, pp. 29–39.
http://dx.doi.org/10.1145/2791261.2791267

[23] S. Jeschke and C. Wojtan, "Water Wave Animation via Wavefront Parameter Interpolation", *ACM Transactions on Graphics*, vol. 34, no. 3, article no. 27, 2015.
http://dx.doi.org/ 10.1145/2714572

[24] C. Yuksel *et al.*, "Wave Particles", *ACM Transactions on Graphics*, vol. 26, no. 3, article no. 1276501, 2007.
http://dx.doi.org/10.1145/2791261.2791267

[25] B. Van De Wiele *et al.*, "Fast Fourier Transforms for the Evaluation of Convolution Products: CPU versus GPU Implementation", *International Journal of Numerical Modeling, Electronic Networks, Devices and Fields*, vol. 27, no. 3, pp. 495–504, 2014.
http://dx.doi.org/10.1002/jnm.1960

[26] M. Fortin *et al.*, "Implementation Strategies for a Software-Compensated FFT-Based Geeric Acquisition Architecture with Minimal FPGA Resources", *Navigation, Journal of the Institute of Navigation*, vol. 62, no. 3, pp. 171–188, 2015.
http://dx.doi.org/10.1002/navi.110

[27] J. Tessendorf, "Simulating Ocean Water", In SIGGRAPH, 1999, Course Note
http://graphics.ucsd.edu/courses/rendering/2005/jdewall/tessendorf.pdf

[28] M. D. Lee and K. S. Lee, "Synthesis of Ocean Wave Models and Simulation Using GPU", *KIPS Transactions on Computer and Communication Systems*, vol. 14, no. 6, pp. 421–434, 2007. (in Korean)
http://dx.doi.org/10.3745/KIPSTA.2007.14-A.7.421

[29] H. Ren *et al.*, "Real-Time Rendering of Ocean in Marine Simulator", in *2008 Asia Simulation Conference – 7th International Conference on*

System and Scientific Computing, Los Alamitos, CA, IEEE Computer Society Press, 2008, pp. 1133–1136.
http://dx.doi.org/10.1109/ASC-ICSC.2008.4675536

[30] Y. Xiong and S. Liu, "Large-Scale Water Simulation Based on Spherical Projected Grid (SPG)", in *SIGGRAPH Asia 2014 Posters*, New York, ACM Press, 2014, article no. A35.

[31] G. LeBlanc *et al.*, "Multi-Band Fourier Synthesis of Ocean Waves", *Journal of Graphics Tools*, vol. 16, no. 2, pp. 57–70, 2012.
http://dx.doi.org/10.1080/2165347X.2012.680854

[32] M. J. Varela and G. C. Soares, "Ring Discretization of the Wave Spectrum for Sea Surface Simulation", *IEEE Computer Graphics and Applications*, vol. 34, no. 2, pp. 58–71, 2014.
http://dx.doi.org/10.1109/MCG.2013.102

[33] Y. Hu *et al.*, "Realistic, Real-time Rendering of Ocean Waves", *Computer Animation and Virtual Reality Worlds*, vol. 17, no. 1, pp. 59–67, 2006.
http://dx.doi.org/10.1002/cav.74

[34] E. Miandji *et al.*, "Real-Time Multi-Band Synthesis of Ocean Water with New Iterative Up-Sampling Technique", *Visual Computer*, vol. 25, no. 5–7, pp. 697–705, 2009.
http://dx.doi.org/10.1007/s00371-009-0352-2

[35] Y. Won, "Real-Time Ocean Rendering in Creating Marine Content Utilizing Real-World Illumination", *Journal of the Island Culture*, vol. 45, no. 6, pp. 209–238, 2015.

[36] H. Cords and O. Staadt, "Real-Time Water Environment with Interacting Objects", in *The 4th Eurographics Workshop on Natural Phenomena (NHP 2009)*, Geneve, Switzerland, Eurogrphics Association, 2009, pp. 35–42.

[37] S. Liu and Y. Xiong, "Fast and Stable Simulation of Virtual Water Scenes with Interactions" , *Virtual Reality*, vol. 17, no. 1, pp. 77–88, Feb, 2013.
http://dx.doi.org/10.1007/s10055-013-0222-0

[38] Y. Li *et al.*, "Real-Time Simulation of Large-Scale Water Environments with Interacting Objects", *Journal of Computational Information Systems*, vol. 11, no. 9, pp. 3257–3264, May, 2015.

[39] M. Weerasinghe *et al.*, "A Novel Approach to Simulate Wind-Driven Ocean Waves in the Deep Ocean", in *Proceedings of International Conference on Advances in ICT for Emerging Regions*, Los Alamitos, CA, IEEE Computer Society Press, 2013, pp. 28–37.
http://dx.doi.org/10.1109/ICTer.2013.6761151

[40] L. Chen *et al.*, "On the Wave Spectrum Selection in Ocean Wave Scene Simulation of the Maritime Simulator", in *Communications in Computer and Information Science*, vol. 402, Heidelberg, Germany: Springer-Verlag, 2013, pp. 453–465.
http://dx.doi.org/10.1007/978-3-642-45037-2_50

[41] L. Chen *et al.*, "Amplitude Malformation in the IFFT Ocean Wave Rendering under the Influence of the Fourier Coefficient", *Electronics*, vol. 18, no. 2, pp. 89–97, 2014.
http://dx.doi.org/10.7251/ELS1418089C

[42] M. K. Ochi, "Ocean Waves: the Stochastic Approach", Cambridge, UK, Cambridge University Press, 1998, pp. 60–70, 218–229.
http://dx.doi.org/10.1017/CBO9780511529559

[43] T. Grindstad and R, Rasmussen, "Deep Water Ocean Surface Modeling with Ship Simulation", M. S. Thesis, Marine Technology Dept, Norwegian University of Science and Technology, Trondheim, Norway, 2011.

[44] C. C. Wang *et al.*, "Ocean Wave Simulation in Real-Time Using GPU", in *2010 International Computer Symposium*, Los Alamitos, CA, IEEE Computer Society Press, 2010, pp. 28–37.
http://dx.doi.org/10.1109/COMPSYM.2010.5685474

[45] J. Dupuy and E. Bruneton, "Real-Time Animation and Rendering of Ocean Whitecaps", in *SIGGRAPH Asia 2012 Technical Briefs (SA 2012)*, New York, ACM Press, 2012, article no. A15.
http://dx.doi.org/10.1145/2407746.2407761

[46] B. M. Nielsen *et al.*, "Synthesizing Waves from Animated Height Fields", *ACM Transactions on Graphics*, vol. 32, no. 1, article no. 2, 2013.
http://dx.doi.org/10.1145/2421636.2421638

[47] A. V. Babanin, "Breaking of Ocean Surface Waves", *Acta Physica Skivaca*, vol. 59, no. 4, pp. 305–335, 2009.
http://dx.doi.org/10.2478/v10155-010-0097-5

[48] A. Toffoli *et al.*, "Maximum Steepness of Oceanic Waves: Field and Laboratory", *Geophysical Research Letters*, vol. 35, no. 5, article no. L05603, 2010.
http://dx.doi.org/10.1029/2009GL041771

[49] F. Schwenger and A. Repasi, "Sea Surface Simulation in the Infrared Modeling and Validation", in *Proceedings of SPIE*, vol. 6239, Bellingham, WA: SPIE Press, 2006, article no. 6239J.
http://dx.doi.org/10.1117/12.664838

[50] D. C. Blanchard, "The Electrification of the Atmosphere by Particles from Bubbles in the Sea", *Progress in Oceanography*, vol. 1, no. 1, pp. 73–202, 1963.
http://dx.doi.org/10.1016/0079-6611(63)90004-1

[51] E. S. Monahan and I. O. Muircheartaigh, "Optimal Power-Law Description of Oceanic Whitecap Coverage Dependence on Wind Speed", *Journal of Physical Oceanography*, vol. 10, no. 12, pp. 2094–2099, 1980.
http://dx.doi.org/10.1175/1520-0485(1980)010<2094:OPLDOO>2.0.CO;2

[52] C. Lafon *et al.*, "Whitecap Coverage in Coastal Environment for Steady and Unsteady Wave Field Conditions", *Journal of Marine Systems*,

vol. 66, no. 1–4, pp. 38–46, 2007. http://dx.doi.org/10.1016/j.jmarsys.2006.02.013

[53] M. A. Srokosz, "On the Probability of Wave Breaking in Deep Water", *Journal of Physical Oceanography,* vol. 16, no.2, pp. 382–385, 1985. http://dx.doi.org/10.1175/1520-0485(1986)016<0382:OTPOWB>2.0.CO;2

[54] J. McCall, "Genetic Algorithms for Modeling and Optimisation", *Journal of Computational and Applied Mathematics*, vol. 184, no. 1, pp. 205–222, 2005. http://dx.doi.org/10.1016/j.cam.2004.07.034

[55] R. Wang and X. Qian, "OpenSceneGraph 3.0 Beginner's Guide", Birmingham, UK, PACKET Publishing, 2010, pp. 50–55.

Received: June 2016
Revised: December 2016
Accepted: December 2016

Contact addresses:

Lining Chen
School of Navigation
Guangzhou Maritime University
510725 Guangzhou
Huangpu District, China
e-mail: lnchen1981@163.com

Yicheng Jin
Laboratory of Marine Dynamic Simulation and Control
Dalian Maritime University
116023 Dalian
Ganjingzi District, China
e-mail: jycdmu@dlmu.edu.cn

Yong Yin
Laboratory of Marine Dynamic Simulation and Control
Dalian Maritime University
116023 Dalian
Ganjingzi District, China
e-mail: bushyin@163.com

Lining Chen is a lecturer Guangzhou Maritime University. He received his PhD degree in traffic information engineering and control from the Navigation School, Dalian Maritime University in 2016, Master's degree in traffic information engineering and control from the Navigation School, Dalian Maritime University in 2010, and Bachelor's degree in logistics engineering from the Transportation Planning and Management School, Dalian Maritime University in 2005. His research interests are computer graphics and navigation simulation.

Yicheng Jin is professor and doctoral advisor at Dalian Maritime University, China. He graduated from Zhejiang University, China, in 1966, with a major in electrical engineering. His research interests are computer graphics, virtual reality and navigation science and engineering.

Yong Yin is professor and doctoral advisor of Dalian Maritime University, China. He received his PhD degree in traffic information engineering and control from the Navigation School, Dalian Maritime University in 2001, Master's degree in traffic information engineering and control from the Navigation School, Dalian Maritime University in 1994, and Bachelor's degree in marine communication from the Navigation School, Dalian Maritime University in 1991. His research interests are computer graphics, virtual reality and navigation science and engineering.