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

Lining Chen¹, Yicheng Jin² and Yong Yin²

¹School of Navigation, Guangzhou Maritime University, Guangzhou, China
²Laboratory 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.

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’ needs, the visual system of the full-motion and multi-task simulator should meet below requirements [1]:

- the frame rate should be higher than 30 fps, 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:

\[
\begin{align*}
\eta(x, y, t) &= \sum_{i,j} A_i \cos(k_i x - \omega_i t + \phi_i) \\
D(x, y, t) &= \sum_{i,j} \frac{k_i}{k} A_i \sin(k_i x - \omega_i t + \phi_i)
\end{align*}
\]

where \( \eta \) is the elevation of the sea surface; \( x = (x_p, y_p) \) is the horizontal position vector; \( t \) is time; \( A_i \) is the amplitude of the cosine wave; \( k_i \) is the wave number vector, and \( k = |k|; \)
\( \omega_i \) is the circular frequency; \( \psi_i \in [0, 2\pi] \) is an independent random number; and \( D(x, y, t) \) is the Choppy wave vector.

Fournier et al. [15] introduced the Gerstner wave model to ocean wave graphics. Thon et al. [16] and Fréchet [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].

2.2.1. Ocean Wave Rendering with Whitecap in the Visual System

The direct use of the Gerstner wave model is problematic when the wave cannot reflect the wind influence over the sea surface. 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:

\[
\begin{align*}
\eta(x, y, t) &= \sum_{i,j} A_i \cos(k_i x - \omega_i t + \phi_i) \\
D(x, y, t) &= \sum_{i,j} \frac{k_i}{k} A_i \sin(k_i x - \omega_i t + \phi_i)
\end{align*}
\]

where \( \eta \) is the elevation of the sea surface; \( x = (x_p, y_p) \) is the horizontal position vector; \( t \) is time; \( A_i \) is the amplitude of the cosine wave; \( k_i \) is the wave number vector, and \( k = |k|; \)
\( \omega_i \) is the circular frequency; \( \psi_i \in [0, 2\pi] \) is an independent random number; and \( D(x, y, t) \) is the Choppy wave vector.

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

\[ \Psi_{\eta}(k) = A_{\eta} \frac{1}{k} \exp \left( - \frac{\pi^2}{U^2 k^2} \right) \cos^2(\theta - \alpha) \]

where \( A_{\eta} \) is the spectral constant, \( U \) is the wind velocity, and \( \alpha \) 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 \( A_{\eta} \). The author of this paper analyzed the structure of the spectrum and found that \( A_{\eta} = 3.48 \times 10^{-16} \), \( U \) is the wind velocity, should be \( U_{0.05} \) [40]. However, when the computed results [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 \( \Delta k, \Delta x = (2\pi / L)^2 \). To correct the malformation, expressions (5) and (6) should be written as

\[ \tilde{\eta}(k) = \frac{1}{\sqrt{2}} \left( \frac{1}{\Delta k^2} + \frac{1}{\Delta x^2} \right) \tilde{\Psi}(k) \]
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' needs, the visual system of the full mission and multi-task simulator should meet below requirements [1]:

- the frame rate should be higher than 30 fps, 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:

\[
\eta(x_{pq}, t) = \sum_{ij} A_i \cos(k_{ij} x_{pq} - \omega t + \phi_{ij})
\]

\[
D(x_{pq}, t) = -\sum_{ij} \frac{k_{ij}}{k} A_i \sin(k_{ij} x_{pq} - \omega t + \phi_{ij})
\]

where \( \eta \) is the elevation of the sea surface; \( x_{pq} = (x_p, y_q) \) is the horizontal position vector; \( t \) is time; \( A_i \) is the amplitude of the cosine wave; \( k_{ij} = (k_i, k_j) = (k \cos \theta, k \sin \theta) \) is the wave number vector, and \( k = k_i \); \( \omega \) is the circular frequency; \( \phi_{ij} \in [0, 2\pi] \) is an independent random number; and \( D(x_{pq}, t) \) 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 IFFT sea surface include the perspective grid [18] and Level of Detail (LOD) technique [19], [20]. The Bidirectional Reflection Distribution Function (BRDF) was adopted to make illumination more realistic [20]. Harrath [22] used the TMA spectrum to compute the directional spectrum, and the Gerstner wave model was used to render sea swell. Some approaches 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:

\[
\eta(x_{pq}, t) = \sum_{ij} D_{pq}(k_{ij}, t) \exp(\sqrt{-1}k_{ij} \cdot x_{pq})
\]

\[
D(x_{pq}, t) = -\sum_{ij} \frac{k_{ij}}{k} D_{pq}(k_{ij}, t) \exp(\sqrt{-1}k_{ij} \cdot x_{pq})
\]

where \( x_{pq} = (x_p, y_q) = (Lp/n, Lq/n) \); \( k_{ij} = (k_i, k_j) = (2\pi \eta L, 2\pi \eta L) \); \( L \) is the width of the rendered sea surface; \( p, q, i, j \in [-n/2, n/2] \) are integers; \( n = 2^n \), and \( m \) is a positive integer; and \( \sqrt{-1} \) is the imaginary unit.

\[
\eta(k_{ij}, t) \text{ and } D(k_{ij}, t) \text{ are, respectively, expressed as}
\]

\[
\eta(k_{ij}, t) = \eta_0(k_{ij}) \exp(\sqrt{-1} \omega t) + \eta_0(-k_{ij}) \exp(-\sqrt{-1} \omega t)
\]

\[
D(k_{ij}, t) = D_0(k_{ij}) \exp(\sqrt{-1} \omega t) - D_0(-k_{ij}) \exp(-\sqrt{-1} \omega t)
\]

where \( \eta_0(k_{ij}) \) and \( D_0(k_{ij}) \) are the complex conjugate. \( \eta_0(k_{ij}) \) and \( D_0(k_{ij}) \) are, respectively, written as

\[
\eta_0(k_{ij}) = \frac{1}{\sqrt{2}} \left( \epsilon_1 + \sqrt{1-\epsilon_1^2} \right) \Psi(k_{ij})
\]

\[
D_0(k_{ij}) = \frac{1}{\sqrt{2}} \left( \epsilon_2 - \sqrt{1-\epsilon_2^2} \right) \Psi(k_{ij})
\]

where both \( \epsilon_1 \) and \( \epsilon_2 \) are independent Gauss randoms, and \( \Psi(k_{ij}) \) is the wave number spectrum. We use \( E \) and \( V \) to denote mean value and variance, respectively, \( E(\epsilon_1) = E(\epsilon_2) = 0 \) and \( V(\epsilon_1) = V(\epsilon_2) = 1 \).

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 bump 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

\[
\Psi_p(k) = A_p \frac{1}{k} \exp \left( -\frac{p^2}{U^2 k^4} \right) \cos^2(\theta - \alpha)
\]

where \( A_p \) is the spectral constant, \( U \) is the wind velocity, and \( \alpha \) 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 \( A_p \). The author of this paper analyzed the structure of the spectrum and found that \( A_p = 3.48 \times 10^3 \) and \( U \), the wind velocity, should be \( U_{10 m} \) [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 \( \Delta x \). \( \Delta x = (2\pi/L)^2 \). To correct the malformation, expressions (5) and (6) should be written as

\[
\eta_0(k_{ij}) = \frac{1}{\sqrt{2}} \left( \epsilon_1 + \sqrt{1-\epsilon_1^2} \right) \Psi(k_{ij})
\]

\[
D_0(k_{ij}) = \frac{1}{\sqrt{2}} \left( \epsilon_2 - \sqrt{1-\epsilon_2^2} \right) \Psi(k_{ij})
\]
\[
D_0(k_x) = \frac{\rho}{\sqrt{2}} \left( e^{+i\eta_0} - e^{-i\eta_0} \right) \Lambda \theta \Phi(k_y)
\]  
(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), Grandet 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:

\[
J(x_p, t) = J_0(x_{p0}, t) + J_1(x_{p1}, t) - J_2(x_{p2}, t).
\]  
(10)

Obviously, IFFT should be individually performed when \( \partial D_0(x_{p0}, t) / \partial x \) and \( \partial D_0(x_{p0}, t) / \partial y \) are computed. This means that three times of IFFT are needed for one computation of \( J(x_{p}, t) \).

Wang et al. [44] used the Jacobian determinant to detect the spray position, and generated particle-spray 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;
- the assumption that the vertical acceleration on the wave crest is used 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 in a 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 \( a_z(x_{p0}, t) \) to denote vertical acceleration of the wave crest. \( a_z(x_{p0}, t) \) is expressed as

\[
a_z(x_{p0}, t) = \frac{d^2 p}{dt^2}(x_{p0}, t) = \frac{d^2 p}{dt^2}(x_{p0}, t) = -\sum_{i,j} A_j a_z^2 \cos(k_j \cdot x_{p0} - \omega t + \phi_j)
\]  
(12)

To conveniently compute \( a_z(x_{p0}, t) \) we need to write the expression of \( a_z(x_{p0}, t) \) in the IFFT form. First, we write expression (12) in the complex form as:

\[
a_z(x_{p0}, t) = -\frac{1}{2} (C_j + C_{j*})
\]  
(13)

where

\[
C_j = \sum_{i=0}^{N/2-1} \left[ \sum_{j=0}^{N/2-1} a_z^2 \exp\left( -\frac{1}{2} (k_j \cdot x_{p0} - \omega t + \phi_j) \right) \right]
\]  
(14)

\[
C_{j*} = \sum_{i=0}^{N/2-1} \left[ \sum_{j=0}^{N/2-1} a_z^2 \exp\left( -\frac{1}{2} (k_j \cdot x_{p0} - \omega t + \phi_j) \right) \right]
\]  
(15)

\[
-\frac{1}{2} \frac{\sqrt{\Lambda \theta \Phi(k_y)}}{\sqrt{2}} + \frac{\sqrt{\Lambda \theta \Phi(k_y)}}{\sqrt{2}} \]  
(16)

where \( a_z(k_j) \) is the Fourier coefficient of \( a_z(x_{p0}, t) \). Therefore, \( a_z(x_{p0}, t) \) can be computed by IFFT, and only one time of IFFT is needed in one computation of \( a_z(x_{p0}, t) \). Therefore, computing is less expensive than computing \( J(x_{p}, t) \), 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...
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 [35], [36], 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), Grandstaff 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:

\[
J_{\omega}(\mathbf{x}, t) = \frac{1}{\sqrt{2}} \left( \omega_x \cdot \mathbf{x} + \mathbf{I}_{10} \right) \sqrt{\Delta \omega \mathcal{P}(\mathbf{k})}
\]  

(9)

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

Obviously, IFFT should be individually performed when \( \partial \omega / \partial x \) \( \mathbf{x} + \Delta \omega / \partial y \mathbf{y} \) and \( \partial \omega / \partial y \mathbf{y} \) \( \mathbf{y} \). This means that three times of IFFT are needed for one computation of \( \omega(x, y, t) \).

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;
- the Jacobian determinant is 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 temporary 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 \( a_z(x, y, t) \) to denote vertical acceleration of the wave crest. \( a_z(x, y, t) \) is expressed as:

\[
a_z(x, y, t) = \frac{d^2 \mathbf{x}(x, y, t)}{dt^2} = \frac{\partial^2 \mathbf{x}(x, y, t)}{\partial t^2}
\]  

(12)

To conveniently compute \( a_z(x, y, t) \) we need to write the expression of \( a_z(x, y, t) \) in the IFFT form. First, we write expression (12) in the complex form as:

\[
a_z(x, y, t) = -\frac{1}{2} (C_1 + C_2)
\]  

(13)

where

\[
C_1 = \sum_{i=-N/2}^{N/2-1} \sum_{j=-N/2}^{N/2-1} a_i^2 \exp \left( -i \mathbf{k} \cdot \mathbf{x} \right) a_j \exp \left( -i \mathbf{k} \cdot \mathbf{x} \right)
\]

(14)

\[
C_2 = \sum_{i=-N/2}^{N/2-1} \sum_{j=-N/2}^{N/2-1} a_i^2 \exp \left( -i \mathbf{k} \cdot \mathbf{x} \right) a_j \exp \left( -i \mathbf{k} \cdot \mathbf{x} \right)
\]

(15)

where \( \mathbf{k} \) is the Fourier coefficient of \( a_z(x, y, t) \). Therefore, \( a_z(x, y, t) \) can be computed by IFFT, and only one time of IFFT is needed in one computation of \( a_z(x, y, t) \). Therefore, computing is less expensive than computing \( J_{\omega}(x, y, t) \), 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.
where \( W_e \) 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 \( U_{19.5} = 0.9016 \) \( U_{19.5} \).

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 \( a_z \leq -\beta g \), wave breaking occurs and the whitecap is generated. \( \beta = 0.388 \) \[=0.39\] \[=0.39\] \[=0.39\] \[=0.39\] \[=0.39\]. Expression (12) indicates that \( a_z \) distributes normally, \( a_z \sim N(0,1) \), and \( F(a_z) = \Phi \left( \frac{a_z}{\sigma} \right) \) is the probability density function of \( a_z \), \( \Phi \) is the cumulative distribution function of the standard normal distribution.

If the wind parameter is fixed, both \( W_e \) and \( W_a \) 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

\[
C = (1 - W_e)C_T + W_eC_W
\]  

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

- the variable used to compute \( W_e \) should be a reasonable physical quantity, so that the computation of \( W_e \) has sound theoretical support;
- the variable is conveniently computed, and the computation does not reduce the frame rate;
- \( E(W_e) \) 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 \( W_e \).

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

Figure 1 is the diagram of \( \eta \) 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 when \( t = t_2 \). It should be noted that \( a_z \) increases monotonically when the particle moves from the crest to the trough. If \( a_z \leq -\beta g \), the whitecap is generated and \( W_e \) is positive when \( t = t_1 \). In regard to \( t = t_2 \), the whitecap disappears and \( W_e = 0 \). During the process, \( W_e \) changes from positive to 0.

As mentioned above, if \( E(W_e) \) 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 \( E(W_e) = W_e \).

We have built a discontinuous mathematical model to compute \( W_e \). The discontinuous model is expressed as

\[
W_e = \begin{cases} 
0, & \text{if } a_z \in \left(-\beta g, \infty\right) \\
1, & \text{if } a_z \in \left(-\infty, -\beta g\right)
\end{cases}
\]  

where \( \beta = 0.39 \). \( W_e \) is computed by expression (22), \( E(W_e) = W_e \). When the model is used in rendering, \( C_r \) is equal to either \( C_T \) or \( C_W \), the shaded color of the generated whitecap changes slightly, 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 \( W_e \). Let \( a_1 \) and \( a_2 \) be positive, \( a_1 \leq \beta g \), and \( a_2 \leq \beta g \). Our strategy is as follows:

- if \( a_z \in \left(-\alpha g, -\beta g\right), \text{ } W_e = 1, \text{ and } C = C_W; \)
- if \( a_z \in \left(-\alpha g, -\beta g\right), \text{ } W_e = 0, \text{ and } C = C_T; \)
- \( a_1 \) and \( a_2 \) can be computed from expression (27).

To facilitate the computation, we set \( a_1 \) as a constant, and compute the corresponding values of \( a_2 \) and \( \beta_1 \). It is difficult to obtain the analytical solution for \( a_2 \), so the optimization algorithm should be employed to compute the optimum solution of \( a_2 \). We use the Standard Genetic Algorithm (SGA) [54] to compute the optimum solution. The objective function of SGA is written as:

\[
\Omega(E) = \frac{E(W_e) - \min(E(W_e))}{\max(E(W_e)) - \min(E(W_e))}
\]  

The value of \( \Omega(E) \) should be close to 1, and \( \beta \_1 \) should be computed such that \( \Omega(E) \) is equal to \( \Omega(E) \).

\[
\beta_1 = \frac{\exp \left(-\frac{a_2 - a_1}{2m_4}\right) - 1}{\exp \left(-\frac{a_2 - a_1}{2m_4}\right) - 1}
\]  

where \( a_1 \) and \( a_2 \) can be computed from expression (27).
where \( W_a \) 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 \( U_{19.5} = 0.916 U_{10} \).

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 whitecap generation can be computed using the empirical formula, i.e., the cumulative distribution function of the concerned ocean wave parameter is employed to compute the whitecap coverage function.

As mentioned above, if \( E(W_a) \) 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 \( E(W_a) = W_a \).

We have built a discontinuous mathematical model to compute \( W_a \). The discontinuous model is expressed as:

\[
W_a = 0, \text{ if } a_c \in (-\infty, -\beta g \sigma) \\
1, \text{ if } a_c \in (-\beta g \sigma, \infty)
\]

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

- the variable used to compute \( W_a \) should be a reasonable physical quantity, so that the computation of \( W_a \) has sound theoretical support;
- the variable is conveniently computed, and the computation does not reduce the frame rate;
- \( E(W_a) \) 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 \( a_c \) to compute \( W_a \).

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

Figure 1 is the diagram of \( \eta \) and \( a_c \) 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 \( a_c \) increases monotonically when the particle moves from the crest to the trough. If \( a_c \leq -\beta g \sigma \), the whitecap is generated and \( W_R \) is positive when \( t = t_1 \). In regard to \( t = t_2 \), the whitecap disappears and \( W_R = 0 \). During the process, \( W_R \) changes from positive to 0.

As mentioned above, if \( E(W_a) \) 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 \( E(W_a) = W_a \).

We have built a discontinuous mathematical model to compute \( W_a \). The discontinuous model is expressed as:

\[
W_a = 0, \text{ if } a_c \in (-\infty, -\beta g \sigma) \\
1, \text{ if } a_c \in (-\beta g \sigma, \infty)
\]

where \( a_c \) and \( \sigma \) control the range of \( a_c \) and \( \beta g \) limits \( W_R \) to between 0 and 1. Figure 2 shows the curve of the continuous mathematical model. \( E(W_a) \) is expressed as

\[
E(W_a) = \frac{-\beta g \sigma}{2} f(a_c) da_c = \int_{-\beta g \sigma}^{0} f(a_c) da_c + \int_{-\beta g \sigma}^{\infty} 0 \cdot f(a_c) da_c
\]

If \( E(W_a) = W_a \), we obtain

\[
C_C C_w = C + \Phi(a_c g / m_1^2) - \Phi(\beta g / m_1^2)
\]

where \( C_C \) and \( C_w \) are constants. The shaded color of the generated whitecap changes gradually, and the rendered result is desirable. So a continuous model may be a better choice.

To improve the rendering results, we have built a continuous mathematical model to compute \( W_a \). Let \( a_c \) and \( \sigma \) positive, \( a_c \in (0, \beta g \sigma) \), and \( \sigma \in (0, \infty) \). Our strategy is as follows:

- if \( a_c \in (-\infty, -\beta g \sigma) \), \( W_R = 0 \), and \( C = C_w \);
- if \( a_c \in (-\beta g \sigma, \infty) \), \( W_R = 0 \), and \( C = C_w \);
- if \( a_c \in (-\beta g \sigma, 0) \), \( C \) is computed by blending \( C_C \) and \( C_w \), \( W_R \in (0,1) \), 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

\[
W_a = \begin{cases} 
0, & \text{if } a_c \in (-\beta g \sigma, 0) \\
1, & \text{if } a_c \in (-\beta g \sigma, \infty)
\end{cases}
\]

where \( \beta g \) is a constant. The shaded color of the rendered whitecap is computed as

\[
f(a_c) = \beta g \left( \exp \left( a_c g / m_1^2 \right) \right) - 1
\]

In expression (24), \( a_c \) and \( \beta g \) control the range of \( a_c \) and \( \beta g \) limits \( W_R \) to between 0 and 1. Figure 2 shows the curve of the continuous mathematical model. \( E(W_a) \) is expressed as

\[
E(W_a) = \frac{-\beta g \sigma}{2} f(a_c) da_c + \int_{-\beta g \sigma}^{\infty} 0 \cdot f(a_c) da_c
\]

where \( C_C \) and \( C_w \) are constants. The shaded color of the generated whitecap changes gradually, and the rendered result is desirable. So a continuous model may be a better choice.
The parameters of the SGA are set in Table 1. Figure 3 is the evolution process of the optimum solution. Table 2 gives $\alpha_1$, $\alpha_2$, and $\beta_1$ computed by the SGA. The smaller $\alpha_1$ is, the longer the lifetime of the whitecap. Our rendering method differs from the actual 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 $E(W_e)$ is close to the whitecap coverage computed as the empirical formula obtained by observation [50]—[52].

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 $E(W_e)$ 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...
The parameters of the SGA are set in Table 1. Table 2 gives \(\alpha_0\), \(\alpha_2\) and \(\beta_1\) computed by the SGA. The smaller \(\alpha_0\), the longer the lifetime of the whitecap. Our rendering method differs from the actual whitecap. The wind velocity of the Phillips spectrum is \(U_{10}\), 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. \(G_i\), \(x_{eye}\) and \(x_{geode}\) 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.

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 and Phillips spectra is used to compute the spectral constant of the Phillips spectrum [40]. The wind velocity of the Phillips spectrum is \(U_{10}\), in the range of 10.0 – 20.0 m/s. Therefore, the Phillips spectrum is effective at scale numbers 5, 6 and 7.

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 and Phillips spectra is used to compute the spectral constant of the Phillips spectrum [40]. The wind velocity of the Phillips spectrum is \(U_{10}\), in the range of 10.0 – 20.0 m/s. Therefore, the Phillips spectrum is effective at scale numbers 5, 6 and 7.

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 \(E(W_g)\) 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

\[
f_{\alpha}(\alpha_2) = \min_{\alpha_2 > \beta_1} \left\{ C_1 C_2 - C_1 - \Phi\left(\alpha_2 g / m_1^2\right) + \Phi\left(\beta_2 g / m_1^2\right) \right\}
\]

(27)
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 as a criterion has sound theoretical support, and the rendered whitecap meets the concerned requirements raised by the DNV.

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


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.

Acknowledgement

The work is supported by the National High Technology Research of China (863 Project) (No. 2015A010404) and the Fundamental Research Funds for the Central Universities of China (No. 3132013302).

References


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.