Ships on Real-time Rendering Dynamic Ocean Applied in 6-DOF Platform Motion Simulator.

Ching-Tang Chou, and Li-Chen Fu, *Fellow, IEEE*

***Abstract*—This paper is focused on constructing a physical dynamic ocean surface for the ship model and motion simulation applied for virtual reality. At the first we introduce the real-time physically-based deep ocean as our virtual environment. We adopt the GPU hardware acceleration to obtain the real-time and realistic rendering ocean scenery with ocean's reflection and refraction phenomenon, and calculate the real force and torque from the generated dynamic waves, which are based on the hydrodynamic model and transferred to our ship model. Finally we obtain the acceleration and angle velocity of the ship which interact with the ocean, and integrate it on the 6 degree-of-freedom motion platform to create a virtual reality environment.**

I. Introduction

In the past few decades, motion vehicle simulator has been used to simulate the mobile vehicle behavior, and widely developed for the driver training, vehicle design, military science, and entertainment. In the field of motion vehicle simulator, planes, cars and ships are the most popular and common simulators that have been usually designed for mobile vehicle. The flight simulator model has been commonly used and is the earlier issue [1]. Ships play important roles in national defense, fishery, and over-sea transportation. Meanwhile, as people view the significance of ships, the simulation of ships is widely developed. Due to the complexity of the ocean wave modeling and rendering, the real-time ship simulator has rarely been issued compared to the other two models.

Ocean is unpredictable and changed from different locations and climates. Currently, the bottleneck that we confronted is the ocean wave modeling, and especially the topic of how to present a close-to-real virtual ocean wave behavior is highly discussed and continually analyzed. Besides, how to diminish the computation time and to render perfect scenario after wave dynamics is well-constructed are the main concerns in the study of computer graphics.

Ocean surface simulation has a large number of documentation and has been applied in many fields of computer graphics and oceanography, and it is also highly developed and has great achievement in games and film industries. From the point of computer graphics, it has been discussed specifically in fluid mechanics, which is the Computational Fluid Dynamics (CFD). Numerical methods and algorithms are used to solve and analyze the problems of fluid in CFD. However, it is still hard to accomplish the simulation of fluid accurately even though supercomputers or simplified equations are used.

The Navier-strokes equations are the most fundamental equations in computational fluid dynamics (CFD). Scholars view it as the essence to express the dynamics of fluids[2-6]. And then, after the discretization of Navier-strokes equations, real physically-based wave model is obtained. In addition, this equation can be simplified by removing the terms of viscosity, and Euler equations are attained. There are some other methods which are based on these two equations that discretize them into small cell to form volume mesh or grid. Unfortunately, real-time simulation usually fails to accomplish due to the complex and massive computation which is involved in solving this type of equation.

In order to achieve real-time simulation, many researchers try to use the method, particle system, to simulate fluid body. In this method, many particles or fluid elements divided from fluid are used to present the fluid effect. Thus, spray splashing can be presented. However, this method has a serious problem in the real-time application. It is impossible to form a water surface by simple points, balls, or lines, and constructing discrete curves requires huge computation. As a result, how to obtain interactive rate simulation is our core trade-off.

Around fifteenth-century, Chinese Ming dynasty Cheng-her, Christopher Columbus, the first people discovered the America continent, and Ferdinand Magellan, the first people traveled around the world, launched the importance of navigation and made a breakthrough epoch. Since then, the stability and construction of the ocean vessels were widely discussed and investigated by shipbuilding and control engineers. Thus, we have to consider about the interactive physical behavior between the hull and wave on the ship modeling. Therefore, we need a suitable mathematical model to present a 6 degree-of-freedom ship maneuver due to the wave disturbance.

The 2D height map method is the simplest and fastest computer algorithm for simulating and rendering fluid body.

Ching-Tang Chou is with the Electrical Engineering Department, National Taiwan University, Taipei, Taiwan, R.O.C. (phone: +886-2-23635251 ext. 205; e-mail: r94921016@ntu.edu.tw).

Li-Chen Fu is with the Electrical Engineering Department, National Taiwan University, Taipei, Taiwan, R.O.C. (e-mail: lichen@ntu.edu.tw).

I will present this method that simplify the wave modeling and also make a shader by GPU in order to obtain the real-time and fidelity ship simulator according to our ship modeling.

II. PREVIOUS WORKS

Claude Navier (1822), and George Stokes (1845) formulated the famous Navier-Stokes equations that described the real dynamics of fluids. Physically-based fluid simulation is adapted on these equations. There are two general approaches in analyzing fluid mechanics problems [7]:

The first method, called Eulerian method, applies the field concept. In this case, all the positions that fluid flows through it were treated as fixed points. Then, describing all the fluid properties (pressure, velocity, density) vary on different spaces and times. This is also a grid-based or mesh-based method [8, 9]

Foster adopted the 3D Navier-Stokes equations to simulate the fluids. He marked the cells to obtain the solution. But this calculation required plenty of time [10]. Chen presented another method to solve the Navier-Stokes equations, diminishing the 3D equations to 2D equations to get the velocity gradient and solved the height field according to the pressure Bernoulli's equation. It could avoid the huge computation to accelerate the simulation time [11].

Stam rewrited the explicit Eulerian method. He used the Lagrangian equations and implicit method to calculate the Navier-Stokes Equations, and developed a stable solver to simulate the stable fluid [9].

The second method, called the Lagrangian method, divided the fluid into particles or a small fluid element. For each particle, its properties (pressure, velocity, density) change as a function of time. This is what we called particle method.

Lucy, and Gingold et al. introduced the Lagrangin particle-based method called Smoothed Particle Hydrodynamics (SPH). Muller et al. also use a Smoothed Particle Hydrodynamics (SPH) based approach to simulate free surface flows. The surface tension also considered in this approach which could simulate 5000 particles at interactive rates. The fluid surface was rendered by the marching cube method at the end [12]. Premoe et al. use a moving particle semi-implicit (MPS) technique to simulate the flow of different kinds of fluid [13]. The Navier-Stokes Equations recast to the interactive particle format. These approaches are hard to construct the smooth surface. If so, complex calculation is required. Meanwhile large amount of calculation is involved as increase of the particles.

Under the circumstances of real-time simulation, we regressed to adopt the spectral method to generate the fluid. Peachey presented a linear superposition of wave functions and simplified it as wave shape function and phase combinations [14]. Thus, the surface height field could be attained. This method is only processed for simulating shore wave. Unfortunately, large and wide ocean surface scenes could not be simulated. Thon et al. presented an intermediate model with the advantage of physical and empirical model. The procedural wave model was defined by the combination of two level-of-details [15].

Fournier adopted the Lagrangian particle to simulate the surface parameters of Gerstner wave model. The sinusoid function of the ocean surface could be tuned according to the ocean depth and inclination of ocean floor [16]. Tessendorf utilized the statistics and FFT model to model the middle scale ocean wave mesh. This model was constructed by some statistical parameters [17]. Moreover, Hinsinger improved this method and added the adaptive camera algorithm to make the ocean mesh unbounded [18].

Davidson who is the first person presented the ship steering theory [19]. Abkowitz adopted the multivariable Taylor series expansion of the forces and moments about some initial equilibrium condition. It could be used in large ships simulator [20].

Hirano and Inoue et al. developed the compact mathematical model including the related hydrodynamic effects caused from seakeeping and maneuvering. They demonstrated the sufficient accurate theoretical and empirical formula based on experimental tests [21-23].

Browning modified the mathematical model so that it could be used in small boat. He also presented the simulation framework which could be expanded to other marine vehicles [24].

III. OCEAN'S WAVE MODELING.

In this section, I will introduce the multi-variables function to describe the ocean surface. The main representation of this function is the height and the geometry of the ocean wave surface:

$$y = f\left(x, z, t\right). \tag{1}$$

In this representation, we consider the free surface is constructed by lots of particles (points). Each particle (i.e. each point) is continuous arrangement and is up and down around its rest position (*x, y, z*). Where the (*x, y, z*) is the usual 3D Cartesian modeling space with the Y axis directed upward, and the XZ plane is the plane of the sea at rest.

The wave behavior is depicted by superposition a 2D sinusoid function. Airy (1845) found out the wave profile is somewhat like a sinusoid wave:

$$h(x,t) = A\sin(kx - wt) \tag{2}$$

This is a function to represent a water surface height on the Y axis direction. A is the wave amplitude, k is the wave number, this number is defined as $2\pi/\lambda$ by the wave length λ. w is the pulsation which is defined as the $2\pi f$ by the frequency *f*. A, k, and f are all varied by the time. The ocean surface represented by this model is unable to break up. Thus, the spray and some complex wave phenomena can not be modeled by this method. However, its characteristics are simple and real-time to render the ocean surface.

Gerstner brought out the approximate solution of the fluid dynamics in 1804. It was the earliest wave profile was acknowledged in oceanography. The ocean wave usually gets the sharp crest and rounded troughs characteristic. The Gerstner model equation described the motion of each water particle in deep water is:

$$\begin{aligned} x &= x_0 + r \cdot \sin\left(kx_0 - \omega t\right) \\ z &= z_0 + r \cdot \cos\left(kz_0 - \omega t\right) \end{aligned} \quad (3)$$

The water surface can be obtained as a trochoid. The shape of the trochoid is determined by multiplying the rk. For values $0 < rk < 1$, we can get the sharp crest and rounded troughs wave shape. For $rk > 1$, the crest is deformed in Fig. 1, the original wave surface inside turns over to outside. For this situation the ocean wave surface is not a desired and realistic shape.

Fig. 1. The rk value of the trochoid on left side is smaller than 1, but on the right side is bigger than 1.

For the large ocean surface implementation, we extend the 1D sinusoid equation (2) into 2D to form a surface. Thus, the water height *y* on any point (*x*, *z*) of a surface plane according to the time *t* is given by:

$$y = h(x,z,t) = A\sin(k(x\cos\theta + \mathrm{z}\sin\theta) - wt + \varphi) \quad (4)$$

Here, θ is the direction angle of the wave and φ is the initial phase which can be selected randomly between $0 \sim 2\pi$. However, we can sum up several these sinusoid functions to approach the Gerstmer wave profile. So the wave model become to:

$$y = h(x,z,t) = \sum_{i=1}^{n} A_i \sin(k_i(x\cos\theta_i + z\sin\theta_i) - w_i t + \varphi_i) \quad (5)$$

,where *n* is the number of different sinusoid functions with different A_i, k_i, θ_i, w_i, and φ_i parameters. These parameters can be defined by the user. Due to the natural phenomena are always chaotic. The noise functions are used to decide those parameters. Finally, the wave can be generated by random variables.

IV. OCEAN SURFACE CONSTRACTION AND RENDERING.

After modeling the ocean wave, the triangle mesh was constructed to implement the sinusoid ocean model. The triangle mesh is union of triangles,

$$M = \bigcup_{i \in [1,\mathrm{N}]} T_i \quad (6)$$

,where N is the number of the triangles. At first, we generated the mesh based on fixed interval which size is defined by the user in a special space Ω. Then the motion of the triangle mesh up and down liking the wave we defined.

In order to control the triangle mesh, control points was sampled to move the mesh according to the ocean wave model. On construction the smooth curve of the ocean wave we choose an easier and more convenient method. The method is using the mesh points be our sampling points. The sampling points are the set in our special space,

$$\rho_i\left(x_i, y_i, z_i\right) \in \Omega \quad (7)$$

,where the *i* is the number of our sampling points. From the sampling points we can generate the two-dimension surface with triangle meshes by defining the interval of $\left(x_i, z_i\right)$. The mesh points were considered as the surface particles. Thus, the wave surface dynamic motion can describe as the motion of particles. The height value of the wave particles are described by y_i value of the sampling points. Therefore, the three-dimension wave which display in fig. 2. was created according to this 3D triangle mesh.

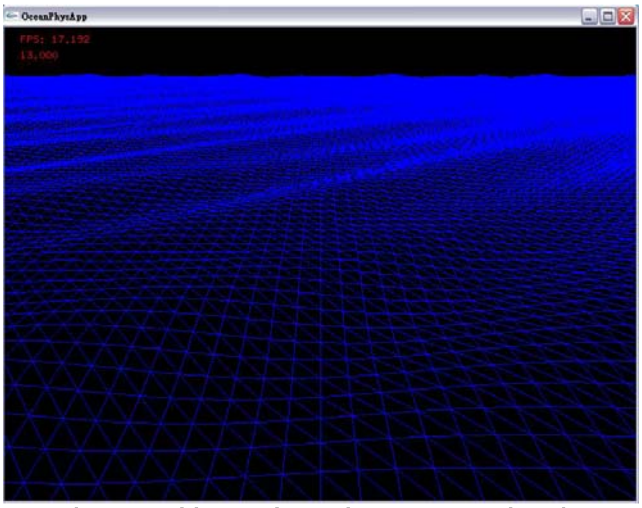

Fig. 2. The wave model was implemented on 512x512 triangle mesh.

Most of the visual effect of the water was described by reflection and refraction. After an incident ray hits to the water surface, part of it will reflect to the air, and the other part of it will transmit into the water. The former ray is usually seen by camera directly, and the latter ray scatters or hits to the object inside the water then transmits the surface again from water to the air or to the camera. All of these effects can be correctly described by complex rendering equation. However, this equation is too sophisticated to implement in real-time. Here the simplified first order equation was taken.

Due to the ocean surface is the highly reflective surface. When you look at the ocean surface on the boat which is sailing in the ocean, the scene you really see is the environment of the sky or the inverted reflection of the cloud or sun. The characteristics of the reflected ray depend on the object surface normal N and the incident ray vector I:

$$R = I - 2N(N \cdot I) \quad (8)$$

,where R is the reflection vector. The environment mapping can easily be presented from this simplified equation. But we assume that the environment is infinitely distance from the object. For the deep ocean which is simulated here, the refraction situation was omitted. The bottom of the ocean seems impossible to be seen. Here, Fresnel term is an important visual essential of the water rendering. It is a weight for the blending the reflection and refraction. For the deep ocean simulation here, we modified this term to let the visual effect only have the relationship with reflection.

V. PRELIMINARY RIGID BODY DYNAMICS AND THE SYSTEM COORDINATE DEFINITION

The position and orientation of a rigid body was updated by the following:

$$\dot{x} = v, \quad \dot{q} = \tfrac{1}{2}\mathrm{quat}(\omega)q \quad (9)$$

,where x and q are the position and the orientation of the rigid body respectively, and the v and ω are the linear and angular velocities respectively of the rigid body. Function $\mathrm{quat}(\omega)$ constructs a quaternion with zero scalar part from an angular velocity vector.

Using the forward Euler integration, the position can be updated by

$$x^{n+1} = x^n + \Delta t v^n \quad (10)$$

, and the orientation can be defined by

$$q^{n+1} = \hat{q}(\Delta t w^n)q^n \quad (11)$$

,where $\hat{q}(\omega) = \left(\cos(|\omega|/2), \sin(|\omega|/2)\,\omega/|\omega|\right)$ is a unit quaternion.

Thus the motion of the rigid body can be specified by *Newton's second law*, the velocities are updated by *Newton-Euler formulas* as following two equations:

$$\dot{v} = \frac{F}{m}, \quad \frac{d(I\omega)}{dt} = \tau \quad (12)$$

where F is the net external force, m the mass of the rigid body, and τ is the net external torque, I is the inertia tensor of the form $I = RDR^T$ (R is the rotation matrix and D is the diagonal inertia tensor in body coordinate).

According to the Newton-Euler formulas, the velocity term can be updated by the applying force and torque. Using forward Euler integration, we get the new velocity term by integrating (12),

$$\begin{aligned} v^{n+1} &= v^n + F/m\,\Delta t \\ \omega^{n+1} &= \omega^n + I^{-1}\left(\tau - \omega^n \times I\omega^n\right)\Delta t \end{aligned} \quad (13)$$

In this phase, we can obtain the velocity and angular velocity updating of the rigid body by the external force and torque. The force term applied here should be accumulated in one time step in my simulation pass.

VI. SHIP MODELING AND DYNAMICS

In order to simulate the ship sailing motion on the ocean, the ship modeling established is the first thing we do. The numerical model we adopted here is referred to [24, 25]. The ship can be viewed as a rigid body. Therefore, the equation of the ship modeling also can be summarized by *Newton's second law* . The six degree of freedom was expressed below:

Surge $m(\dot{u} + vq - wr) = X_{sw} + X_G + X_W + X_H + X_R + X_P$

Sway $m(\dot{v} - vp + ur) = Z_{sw} + Z_G + Z_W + Z_H + Z_R + Z_P$

Heave $m(\dot{w} + wp - uq) = Y_{sw} + Y_G + Y_W + Y_H + Y_R + Y_P$

Roll $I_{xx}\dot{p} = L_{sw} + L_G + L_W + L_H + L_R$

Pitch $I_{zz}\dot{q} = M_{sw} + M_G + M_W + M_H + M_R$

Yaw $I_{yy}\dot{r} = N_{sw} + N_G + N_W + N_H + N_R$

In order to describe this model, two sets of Cartesian coordinate system are defined. The first one is the world-coordinate system $(O\text{-}XYZ)$. The other is body-coordinate system $(G\text{-}xyz)$ of the ship. The XZ plane is the horizontal plan and Y axis is vertically upward. G is the center of the ship mass. The X, Y, Z and L, M, N are the hydrodynamics forces and torque applying on the ship in X-direction, Y-direction and Z-direction. The u, v and w are the linear velocity of the ship along X, Z and Y direction, respectively. Based on the same concept, the p, q and r are the angular velocity of the ship along X, Z and Y direction. The *m* is the mass of the ship and the *I* is the inertial tensor with respect to the X, Z and Y axes. The subscripts SW, G, W, H, R and P are respect to the force and moment of the sea wave, gravity, wind, ship hull, rudder, and propeller, respectively.

Due to the two sets of the Cartesian coordinate system, the state function has been defined here to express the relationship between the body-coordinate moving on the ocean and the world-coordinate.

$$\begin{bmatrix} X \\ Z \\ Y \end{bmatrix} = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ z \\ y \end{bmatrix} + \begin{bmatrix} Ut\cos\beta \\ Ut\sin\beta \\ 0 \end{bmatrix} \quad (14)$$

,where the β is the angle between the OX and Gx, and U is speed of the ship when t equals to the zero. Based on this equation we assume that the ship is attached sailing on the ocean. In the other words, the heave motion of the ship can be omitted here. Thus, we can easily change the position and orientation from the body-coordinate system to the world-coordinate system or from the world-coordinate system to the body-coordinate system.

The real motion of the ship is very complicated and difficult to implement here, because of the plenty unexpected variable. Therefore, we present a simplified way on this model to simulate the ship dynamics. However, on this assumption, some hypotheses have to be claim first:

1. The ship is rigid, it is impossible to deform.
2. The motion of the ship moves in the small amplitude ocean, the force can be treated in linear superposition theorem. Nonlinear effect can be eliminated.

3. Using one way coupling on the force applying: The wave can apply force and torque on the ship, but the ship doesn't influence the wave motion at all.
4. The viscosity of the fluid doesn't exist.
5. The ship hull is totally a flat surface.
6. The water particle only can oscillate up and down. Thus, when a rigid object is floating on the water, the vertically force is the only external force on the object. Therefore, the yaw motion is not presented here.

The ship dynamics model according to the wave motion is defined as below:

$$F = \left[-\omega^2 \left(M_0 + A \right) - \omega B + C \right] \times \xi \tag{15}$$

,where the ξ is the motion displacement of the wave particle. ω is encounter frequency between the ship bottom and water particle. A, B, C, M_0 and F are added virtual mass, water damping coefficient, water restoring force, ship mass and external force, respectively. The ship is divided into several equal parts along the length and width by the user then we apply the well-defined force which is modeled on each part bottom. After that, the ship dynamics according to the sea wave (subscript SW), gravity (subscript G) and ship hull (subscript H) can be replaced by this approach.

The propeller (subscript P) force is directly applied on the ship's center of the mass. The sway and heave degree of freedom is too slight to be influenced by the propeller here. For this reason, I just omit these two terms here.

The rudder force (subscript R) is also directly be put on the center of mass. Thus, the steering of the ship can be determined directly.

Finally, the wind force (subscript W) is applied by the fundamental Bernoulli's principal. The aerodynamics reference to [26]. First, the dynamic pressure has been defined:

$$Q = \frac{1}{2} \rho v_T^2 \tag{16}$$

,where ρ is the density of atmosphere, and v_T is the relative velocity of wind and the triangle mesh of the ship. For each triangle mesh, the relative velocity is determined:

$$v_{T,t} = v_{wind} - \left(v^n + \omega^n \times r_{c,t} \right) \tag{17}$$

,where $r_{c,t}$ is the vector from center of mass to the centered of triangle t, and v_{wind} s the current velocity of wind. The dynamics pressure projected to the transverse and longitudinal areas of the ship. Thus, we can get the wind force from the below equation:

$$F_t = \frac{1}{2} \rho v_T^2 \left(\frac{v_T}{\left\| v_T \right\|} \cdot \vec{n}_t \right) \vec{n}_t \tag{18}$$

Up to now, all the forces have been used from the simplified approach based on my assumption.

VII. EXPERIMENT RESULT

We implement the simulation system with C++, and we also use OpenGL and nVidia Cg HLSL to render the results on-line. All the examples are tested on a laptop equipped with Intel Duo 2.00GHz CPU, 1GB RAM, and NVIDIA GeForce Go 7400.

The ship simulation was implemented here, the ship hull was divided into 1600 segments to approach the volume of the ship.

Here, we simply combined two different waves to simulate the water surface. The wave amplitude are chosen to be 0.2 and 0.3, respectively. The wave length and wave frequency was given according to the wind speed. We default our wind speed is 3m/s. The motion of the ship is presented below:

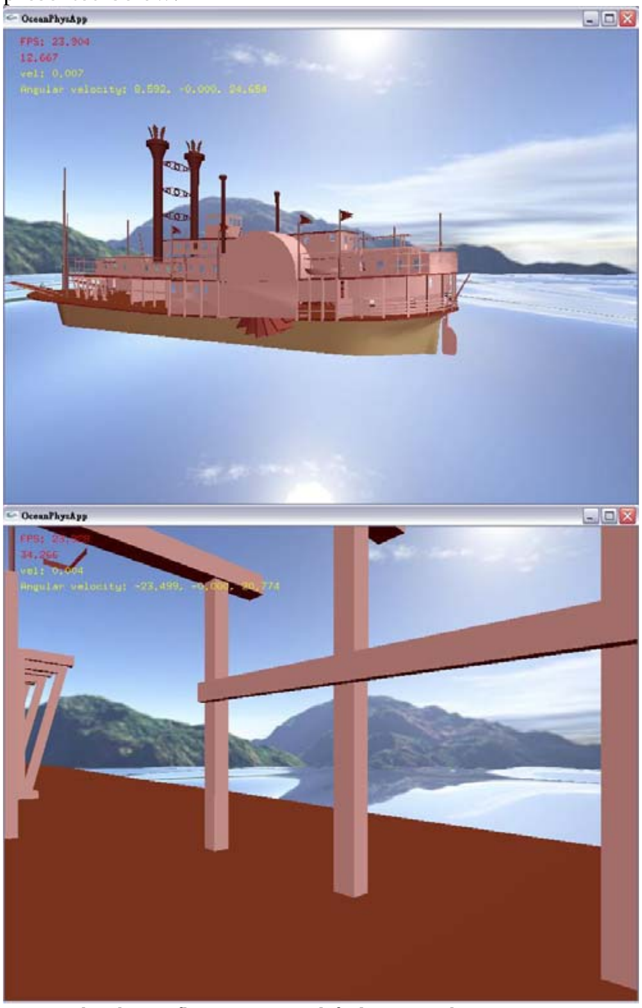

Fig. 3. The ship is floating on my default wave. The yaw motion is not present here, when ship does not turn on the engine. We also can see the reflection image is generated due to the environment.

The ship dynamics are decided by those parameters in my dynamic model, the parameters of the added virtual mass, water damping coefficient, water restoring force coefficient and ship mass are 0.0001, 0.014, 0.16 and 8, respectively. The frame rate of this system can approach to 23.9 FPS, real time approach can be implemented. The most influence effect of the frame rate is the size of the 2D surface grid. Here, we used the 512x512 size to demo this result.

VIII. CONCULUSION AND FUTUREWOEK

The physically based ship dynamics and the simply ocean wave model was demonstrated here. This method is simple and robust, especially on the parameters tuning. The simplified ship model is proposed. The simulation of the ship dynamics is also fidelity to real. The wind effect is also forced on the ship based on the physical approach.

The ocean wave will be modified by the spectrum method to get the more real wave in the future. The shape of the ship body will be considered in this simulation, so that the spray effect will be present due to add the water particles in this system.

IX. REFERENCE

[1] D. M. Bourg, *Physics for Game Developers*: O'Reilly, 2002.

[2] M. Carlson, P. J. Mucha, and G. Turk, "Rigid fluid: animating the interplay between rigid bodies and fluid," *ACM Transactions on Graphics (TOG),* vol. 23, pp. 377-384, 2004.

[3] S. Clavet, P. Beaudoin, and P. Poulin, "Particle-based viscoelastic fluid simulation," *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 219-228, 2005.

[4] D. Enright, S. Marschner, and R. Fedkiw, "Animation and rendering of complex water surfaces," *ACM Transactions on Graphics (TOG),* vol. 21, pp. 736-744, 2002.

[5] N. Foster and R. Fedkiw, "Practical animation of liquids," *Proceedings of the 28th annual conference on Computer graphics and interactive techniques,* pp. 23-30, 2001.

[6] N. Foster and D. Metaxas, "Controlling fluid animation," *Computer Graphics International,* vol. 97, pp. 178-188, 1997.

[7] D. F. Young, B. R. Munson, and T. H. Okiishi, "Brief Introduction to Fluid Mechancis."

[8] Y. Liu, X. Liu, and E. Wu, "Real-time 3D fluid simulation on GPU with complex obstacles," *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pp. 247-256, 2004.

[9] J. Stam, "Stable fluids," *Proceedings of the 26th annual conference on Computer graphics and interactive techniques,* pp. 121-128, 1999.

[10] N. Foster and D. Metaxas, "Realistic animation of liquids," *Graphical Models and Image Processing,* vol. 58, pp. 471-483, 1996.

[11] J. X. Chen, N. V. Lobo, C. E. Hughes, and J. M. Moshell, "Real-time fluid simulation in a dynamic virtual environment," *Computer Graphics and Applications, IEEE,* vol. 17, pp. 52-61, 1997.

[12] M. Müller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," *Proceedings of 2003 ACM SIGGRAPH Symposium on Computer Animation,* vol. 159, 2003.

[13] S. Premože, T. Tasdizen, J. Bigler, A. Lefohn, and R. T. Whitaker, "Particle-based simulation of fluids," *Computer Graphics Forum*, vol. 22, pp. 401-411, 2003.

[14] D. R. Peachey, "Modeling waves and surf," *Proceedings of the 13th annual conference on Computer graphics and interactive techniques,* pp. 65-74, 1986.

[15] S. Thon, J. M. Dischler, and D. Ghazanfarpour, "Ocean waves synthesis using a spectrum-based turbulence function," 2000, pp. 65-72.

[16] A. Fournier and W. T. Reeves, "A simple model of ocean waves," *ACM SIGGRAPH Computer Graphics*, vol. 20, pp. 75-84, 1986.

[17] J. Tessendorf, "Simulating ocean water," *Siggraph Course Notes*, vol. 2, 1999.

[18] D. Hinsinger, F. Neyret, and M. P. Cani, "Interactive animation of ocean waves," *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 161-166, 2002.

[19] K. S. M. Davidson and S. L.I., "Turning and Course Keeping Qualities of Ships," *SNAME Transaction* 1946.

[20] M. A. Abkowitz, "Lectures on Ship Hydrodynamics-Steering and Manoeuvrability," *Report Hy-5, Hydro-and Aerodynamic Laboratory, Lyngby, Denmark*, 1964.

[21] M. Hirano, "Calculation Method of Ship Maneuvering Motion at Initial Design Phase," *J. SOC. NAVAL ARCHIT. JAPAN,* vol. 147, pp. 144-153, 1980.

[22] M. Hirano, J. Takashina, and Y. Takaishi, "Ship turning trajectory in regular waves," *Transaction of the West-Japan Society of Naval Architects,* vol. 60, pp. 17-31, 1980.

[23] S. Inoue, M. Hirano, K. Kijima, and J. Takashina, "A Practical Calculation Method of Ship Maneuvering Motion," *International Shipbuilding Progress*, vol. 28, pp. 207-222, 1981.

[24] A. W. Browning, "A mathematical model to simulate small boat behaviour," *SIMULATION,* vol. 56, p. 329, 1991.

[25] M. C. Fang, K. W. Lin, and Z. H. Shu, "An Indigenous PC-Based Ship Simulator Incorporating the Hydrodynamic Numerical Model and Virtual Reality Technique," *Journal of the Society of Naval Architects and Marine Engineers*, vol. 23, pp. 87-95, 2004.

[26] C.-D. Lee, "Impulse-Based Dynamic Simulation of Articulated Rigid Bodies with Aerodynamics," vol. Master. Taipei: National Taiwan University, 2006.