

# Virtual Noctiluca: Interaction between Light and Water using Real-time Fluid Simulation and 3D Motion Measurement

Kyouhei Aida and Noriko Nagata

Kwansei Gakuin University, School of Science and Technology 2-1 Gakuen, Sanda,  
669-1337 Japan

{ aida, nagata } @kwansei.ac.jp  
<http://ist.ksc.kwansei.ac.jp/~nagata/>

**Abstract.** In recent years, with the rapid improvement of the performance of computers, new possibilities for real-time simulation technologies are emerging. In this study, we simulated the behavior of water in real time and combined this simulation with a measurement of the user's 3D motion to simulate the interaction between water and light, as observed in Noctiluca. Noctiluca is oceanic plankton that produces light when physically stimulated. Stirring the surface of water containing Noctiluca in a completely dark place causes the surface to glow, and an observer can gain the mystical and fantastic experience of watching the glow becoming dim with the flow of the water.

**Key words:** Interactive art, GPU, stereo vision, Smoothed Particle Hydrodynamics, animation, illumination

## 1 Introduction

Simulations based on physical interactions have been widely used in computer graphics to produce animations. In particular, many methods have been developed to visualize fluids realistically simulations; however, the computational cost of these methods have always been high [1][2]. With the recent developments of techniques for obtaining faster simulations using high performance-computers, it has now become possible to calculate fluid simulations in real time. Therefore, there is a possibility that interactive applications using real-time fluid simulations will be developed soon.

In this study, we simulated the behavior of water in real time and combined this simulation with a measurement of a user's 3D motion; in this manner, we simulated the interaction between water and light, as observed in Noctiluca. Noctiluca is oceanic plankton that produces light when physically stimulated. Stirring the surface of water containing Noctiluca in a completely dark place causes the surface to glow, and an observer can gain a fantastic and almost mystical experience of watching the glow becoming dim with the flow of the water. We use Smoothed Particle Hydrodynamics (SPH), which is a method of

particle-based fluid simulation, to simulate realistic water flow. Furthermore, we increased the simulation speed using graphics processing units (GPUs) so that we can receive the response quickly. We also used a stereo camera to measure the 3D motions of inputs such as a user's hand. By using 3D motions as the input data, we can experience "virtual" Noctiluca.

## 2 Related Work

Fluid simulation can be categorized into grid methods and particle methods. In grid methods, many grids are needed for exact computations, so it is not suited for real-time applications. In particle methods, Moving Particle Semi-implicit (MPS) [7] and Smoothed Particle Hydrodynamics (SPH) [8] are typical methods. SPH methods are suitable for real-time simulations because of their low computational cost [6].

For real-time simulations, the growth in computational power of GPUs has materially contributed to their efficiency. GPUs were originally designed for 3D graphics tasks and have also been used to speed non-graphic tasks such as cellular automata simulation, particle simulation, solving of linear equations, and so on. Regarding fluid simulation, several studies have been done on increasing speed [9, 10], and Harada et al. recently proposed a method for SPH on GPUs [11].

On the other hand, several studies have been conducted on the use of physically based simulations in interactive applications. Cassinelli et al. developed an application in which a user can send portions of a projected image forward or backward in time by actually touching and deforming the projection screen, using a CCD camera [12]. Ariga et al. presented a system to show wind flow, computed by fluid simulation, projected on the screen, and then obstructed by the shadows of observers. Matsuo developed an application that controls a group of brilliant butterflies by operating a glowing ball [13]. No previous studies, however, have examined the same interactions between water and light as we propose here. In this study, integration of real-time fluid simulation and stereo vision has been realized.

## 3 Simulation Of Water In Smoothed Particle Hydrodynamics (SPH)

### 3.1 Governing Equations

The velocity of a fluid is influenced by the forces associated with it, such as pressure, viscosity, and external forces. In incompressible fluids, the density remains constant even with the application of pressure. Viscosity is a force that makes the velocity constant. And the only external force is gravitational force. This indicates that the changes in the velocity of incompressible fluids is dependent on the sum of the forces associated with the fluid. The governing equations for

incompressible flow are expressed by the mass conservation equation and the momentum conservation equation as follows

$$\frac{D\rho}{Dt} = 0 \quad (1)$$

$$\rho \frac{Dv}{Dt} = -\nabla p + \mu \nabla^2 v + f \quad (2)$$

where  $\rho$ ,  $v$ ,  $p$ ,  $\mu$ , and  $f$  are density, velocity, pressure, dynamic viscosity coefficient of the fluid, and external force, respectively. The left hand side of Lagrange's differential equation, indicates the changes in the physical values of the flow. It is expressed by the following equation

$$\frac{D\phi}{Dt} = \frac{\partial\phi}{\partial t} + v \nabla \phi \quad (3)$$

The second term on the right hand side of Eq.(2) indicates advection phase. When the fixed grid method is used, we must calculate the parameters associated with the advection phase. This method requires computations by linear interpolation because the grid points have physical values. This leads to serious problems in that the velocity of the fluid decreases to zero and the mass on the surface of the fluid decreases. Simulation with small time steps minimizes the problems, but this is not sufficient these simulations would fail with real-time applications. Lagrange's differential equation indicates the differential calculus on the moving particles. A particle method such as SPH, that problem does not occur because we can calculate the advection phase only by movement of particles.

### 3.2 Discretization

SPH is a particle method in which the fluid is regarded as a group of particle. In SPH, a physical value at a position  $x$  is calculated as a weighted sum of the physical values  $\phi_j$  of neighboring particles  $j$  as follows:

$$\phi(x) = \sum_j m_j \frac{\phi_j}{\rho_j} (x - x_j) \quad (4)$$

where  $m_j$ ,  $\rho_j$ , and  $x_j$  represent the mass, density, and position of particle  $j$ , respectively, and  $W$  is a weight function. The density of the fluid is calculated using Eq.(4) as

$$\rho(x) = \sum_j m_j \nabla W_{den}(x - x_j) \quad (5)$$

Since  $W$  is distant from the position  $x$ , the value of  $W$  becomes approximately 0. Therefore,  $W$  is the sum of the physical values of neighboring particles. In this section, we explain a method to discretize Eq.(2) with Eq.(4). In order to calculate acceleration, we separately compute pressure, viscosity, and external

forces. The acceleration is then calculated as a sum of these forces. The pressure on the fluid is calculated using the constitutive equation

$$p = p_0 + k(\rho - \rho_0) \quad (6)$$

where,  $p_0$  and  $\rho_0$  are the pressure at rest and density, respectively. To compute the momentum conservation equation, gradient and laplacian operators, which are used to solve for the pressure and viscosity forces on particles, have to be modeled. The pressure force  $f^{press}$  and the viscosity force  $f^{vis}$  are computed as follows:

$$f_i^{press} = - \sum_j m_j \frac{p_i + p_j}{2\rho_j} \nabla W_{press}(x_i - x_j) \quad (7)$$

$$f_i^{vis} = \mu \sum_j m_j \frac{v_i + v_j}{2\rho_j} \nabla W_{vis}(x_i - x_j) \quad (8)$$

where  $x_i$  and  $x_j$  are the positions of particles  $i$  and  $j$ , respectively. The weight functions used by Müller et al. are also used in this study [6]. The weight functions for the pressure, viscosity, and other terms are designed as follows.

$$\nabla W_{press}(r) = -\frac{45}{\pi r_e^6} (r_e - |r|)^2 \frac{r}{|r|} \quad (9)$$

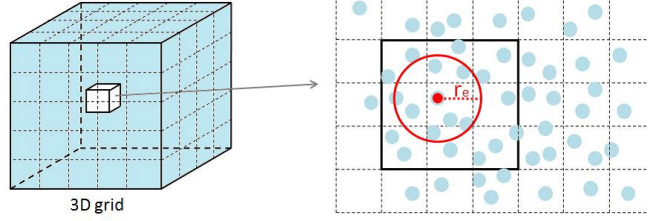
$$\nabla W_{press}(r) = \frac{45}{\pi r_e^6} (r_e - |r|) \quad (10)$$

$$\nabla W_{press}(r) = \frac{315}{64\pi r_e^9} (r_e^2 - |r|^2)^3 \quad (11)$$

In these functions, the value is 0 outside the effective radius  $r_e$ .

### 3.3 Neighbor Search

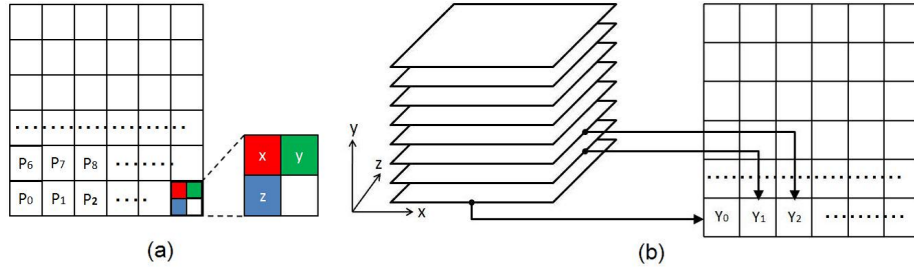
In SPH, each particle must be used to search for neighboring particles in order to calculate the interaction among the particles. The computational cost of searching for neighboring particles is high when a large number of particles are used. To reduce this computational cost, a 3D grid covering the computational region, called a bucket, is introduced, as described by Mishra et al. [14]. Each voxel encoded as a pixel in the texture is assigned a 3D computational space. Then, for each particle, we compute a voxel to which the particle belongs and store the particle index in the voxel. If this bucket can be obtained, we do not have to search for the neighboring particles of a particle  $i$  because the neighboring particles are present in the voxels surrounding the voxel to which the particle  $i$  belongs.



**Fig. 1.** Neighbor search using a bucket

### 3.4 SPH on Graphics Processing Units (GPUs)

To compute SPH on GPUs, physical values are stored as textures in video memories. The textures of position, velocity, density, and bucket are prepared. A texel of a texture corresponds to a particle, and the physical values of a particle are stored in the texel, as shown in Fig. 2 (a). Although a bucket is a 2D array, the current GPUs cannot write to a 3D buffer directly. Therefore, we employed a flat 3D texture in which a 3D array is divided into set of 2D arrays. We divided the 3D textures into 2D arrays, as shown in Fig. 2 (b). The indices of the



**Fig. 2.** (a) All the particles positions are stored at each texel of the position texture.  
(b) The 3D array is divided into a set of 2D arrays.

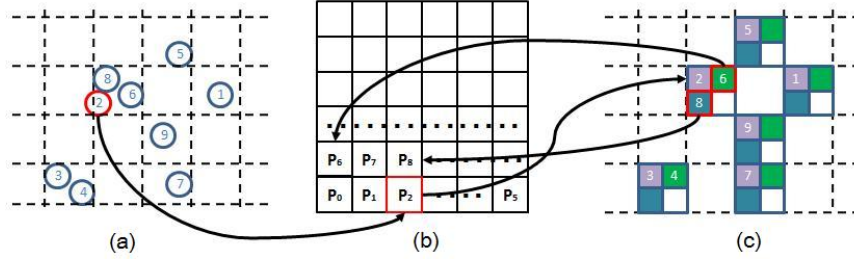
neighboring particles of a particle  $i$  can be found using the generated bucket texture. Using the index of the particle, the position can be obtained from the position texture. The density of a particle  $i$  is then calculated by the weighted sum of mass of the neighboring particles, which is then written into the density texture. The pressure can be calculated from the density and position textures. Further, the viscosity force can be calculated from the position, velocity, and density textures. These forces are computed using Eq.(7) and (8). Assuming that the external force is only gravitation, we can update the velocity, and by using this updated velocity texture, the position is calculated using an explicit

Euler integration as follows:

$$v_i^{t+\Delta t} = v_i^t \frac{\Delta t}{\rho_i} (f_i^{press} + f_i^{vis} + f_i^g) \quad (12)$$

$$x_i^{t+\Delta t} = x_i^t + v_i^{t+\Delta t} \Delta t \quad (13)$$

where  $x_i^t$  and  $v_i^t$  are the previous position and velocity of particle  $i$ , respectively. Although there are higher order schemes, they were not introduced because we did not encounter any stability problems.



**Fig. 3.** Neighbor search using bucket texture. (a) disposition of particles in the region. (b) position texture (c) bucket texture comprising stored particles.

## 4 Manipulation of a Stream of Water

We used a stereo camera (Bumblebee2, Point Grey Research Inc.) to measure the 3D motion of the user's hands and the parts moved by the user in real time; the 3D coordinates of the point sequence of the object surface are entered into the system. Table 1 shows the specifications of the stereo camera. The stereo camera measures the 3D points by employing the principle of stereo vision. The stereo camera cannot adjust the angle of view because the position of the two lens is fixed. However, the stereo camera can measure without calibration since it has defined parameters of the camera.

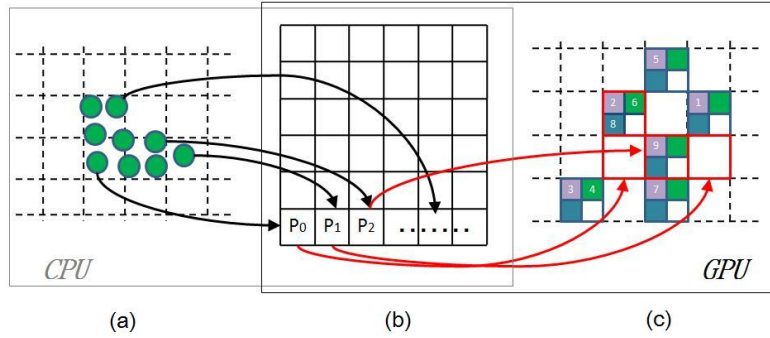
### 4.1 Collision Detection

In this section, we explain our method for detecting collisions between water particles, which are simulated by using SPH, and the 3D coordinates of the point sequence on the object surface. The computational cost of calculating the collision detection among particles is high, and hence, we compute it on GPUs as well as SPH. As described in chapter 3, four types of textures are prepared in SPH on GPUs. Further, the texture of the object's position, called object texture, is prepared to compute the collision detection. The positions of the points measured

**Table 1.** Specifications of the stereo camera

Imaging Sensor	Sony 1/3 " progressive scan CCD
	ICX204 (1032 × 776 max pixels)
	4.65 μ m square pixels
Baseline	12 cm
A/D Converter	12-bit analog-to-digital converter
Frame Rates	20 FPS
Camera Specification	IIDC 1394-based Digital Camera Specification v1.31

by using the stereo camera are stored in each texel, and the empty texels have a value of 0. By using the object texture, we calculate the positions of the object's points and mapped them into a bucket texture. The collision detection between particles is generally computed in cases where the distance between the particles is 0 or below an optional constant value. Our method of collision detection is decided from each voxel without computing distance. We regard the water particles that are stored in the texel of the mapping object's particles as collided particles. In such a manner, we reduce the computational cost, and the method retains precision because each voxel is sufficiently small.



**Fig. 4.** Collision detection on GPUs. (a) 3D coordinates of the point sequence on the object surface. (b) Object texture. (c) Bucket texture.

## 4.2 Taking into Consideration User's Motion

We also need to consider interference from users on the behavior of water; hence the average of all object positions is calculated for each frame. This is expressed by the following equation

$$\Delta v^t = \frac{1}{N} \sum_i x_i^t - \frac{1}{M} \sum_j x_j^{t-1}, (i = 0, \dots, N; j = 0, \dots, M) \quad (14)$$

where  $v^t$  is velocity of the water particle,  $N$  and  $M$  are the number of points measured by stereo camera, and  $x$  is the position of the points. The rate of change in the value is considered to be the speed of the collided water particles.

## 5 Result

The system was implemented using a system equipped with a Core(TM)2 CPU and a GeForce 7950GTX. The programs were written in C++ and DirectX, and the shader programs were written in HLSL. The system comprise a PC, which simulates the behavior of water and regulates the overall system, and a stereo camera, which measures the user's 3D motion. Fig. 5 shows the composition of the system. Collisions between the surface point sequence and water particles

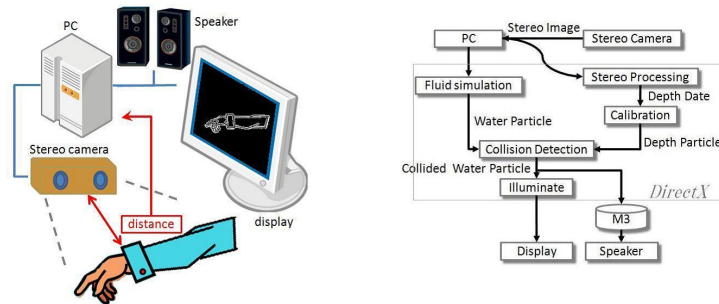


Fig. 5. Overview of the system

were detected by the camera, and the water particles that were involved in the collision were made to produce light. In addition, in order to simplify the regulation of parameters for light production, various patterns of light production were developed by using intuitive expressions such as “ beautiful, ” mystical, ” and “ fantastic. ” Fig. 6 illustrates a demonstration. Our system was used represent a stream of water by fluid simulation. Furthermore, we adopted sound effects

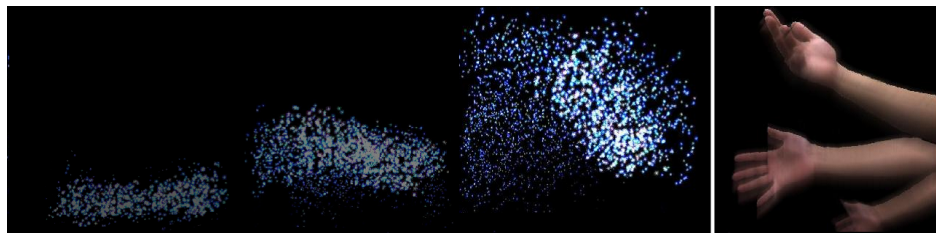
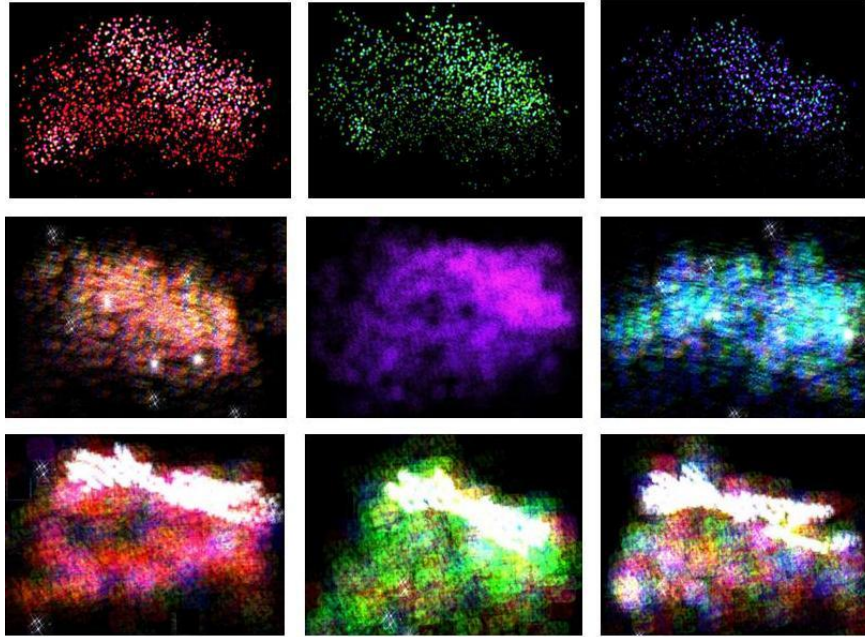


Fig. 6. The motions of a user's hand and the change in the illuminations.





**Fig. 7.** Various examples of illuminations

that change according to the number of water particles as well as their speed, using a music visualization system [15]. This system produced synergistic effects with visual sensations and engaged users. Even with such a complex pattern of light production, we realized a frame rate of 10 fps. The system allows users to experience the natural luminescence of Noctiluca.

## 6 Conclusion

In this study, we have simulated the interactions of Noctiluca with water by using real-time flow simulations and 3D motion measurement. Although computational cost of fluid simulation is high, we can receive the response more quickly using GPUs effectively. Furthermore we can operate our application intuitively measuring user's 3D motion with stereo camera. We developed the interesting application, which is not only the sense of sight, but also hearing, by adopting sound effects that change according to the user's input. Using this application, mysterious and fantastic sensations that cannot be experienced in daily life can be felt.

## References

1. Geoffrey Irving, Eran Guendelman, Frank Losasso, Ronald Fedkiw, Efficient Simulation of Large Bodies of Water by Coupling Two and Three Dimensional Techniques,

- SIGGRAPH2006, ACM TOG 25, pp.805-811, (2006).
2. Losasso, Tamar Shinar, Andrew Selle, Ronald Fedkiw, Multiple Interacting Liquids, SIGGRAPH2006, ACM TOG 25, pp.812-819, (2006).
  3. Nick Foster and Dimitris Metaxas. Modeling the motion of a hot, turbulent gas. In SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp. 181-188. ACM Press/Addison- Wesley Publishing Co., (1997).
  4. Jos Stam. Stable fluids. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp. 121-128. ACM Press/Addison-Wesley Publishing Co., (1999).
  5. Mark Carlson, Peter J. Mucha, and Greg Turk. Rigid fluid: animating the interplay between rigid bodies and fluid. ACM Trans. Graph., Vol. 23, No. 3, pp. 377-384, (2004).
  6. M. Müller, D. Charypar, and M. Gross. Particle-based fluid simulation for interactive applications. Proc. of Siggraph Symposium on Computer Animation, pages 154-159, (2003).
  7. S. Koshizuka and Y. Oka. Moving-particle semi-implicit method for fragmentation of incompressible fluid. Nucl.Sci.Eng., 123:421-434, (1996).
  8. J.J. Monaghan. Smoothed particle hydrodynamics. Annu.Rev.Astrophys, 30:543-574, (1992).
  9. P.Kipfer, M.Segal, and R.Westermann. Uberflow: A gpu-based particle engine. Proceedings of the ACM SIGGRAPH/EUROGRAPHICS 6 Conference on Graphics Hardware, pages 115-122, (2004).
  10. A.Kolb, L.Latta, and C.Rezk-Salama. hardware based simulation and collision detection for large particle systems. Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware, pages 123-131, (2004).
  11. Takahiro Harada, Seiichi Koshizuka, Yoichiro Kawaguchi. Smoothed Particle Hydrodynamics on GPUs. Computer Graphics International, (2007)
  12. Alvaro Cassinelli, Masatoshi Ishikawa. Khronos projector, ACM SIGGRAPH 2005 Emerging technologies, Article No. 10, (2005)
  13. Takahiro Matsuo. Phantasm, ACM SIGGRAPH 2008 art gallery, Pages 101-101, (2008)
  14. B.K. Mishra. A review of computer simulation of tumbling mills by the discrete element method: Parti-contact mechanics. International Journal of Mineral Processing, 71(1):73- 93, (2003).
  15. Fujisawa T. X., Tani, M., Nagata, N., and Katayose, H. Music Mood Visualization based on Quantitative Model of Chord Perception. J. Information Processing Society of Japan, 50(3), 1133-1138, (2009).