Vis Comput (2010) 26: 865–872
DOI 10.1007/s00371-010-0439-9

ORIGINAL ARTICLE

Solving the Shallow Water equations using 2D SPH particles for interactive applications

Hyokwang Lee · Soonhung Han

Abstract In this paper, we introduce a 2D particle-based approach to achieve realistic water surface behaviors for interactive applications. We formulate 2D particle-based Shallow Water equations using the Smoothed Particle Hydrodynamics. Particles defined with specific amount of water volume interplay with each other, which generates the horizon flow and the water surface motion. By the application of the particle-based Lagrangian framework to the 2D Shallow Water simulation, our method allows the water particles to move freely without being confined to a grid. The motion of the particles can represent global flow with dynamic waves covering a large area while avoiding extensive 3D fluid dynamics computation. The 2D particle-based Shallow Water equations are straightforward and computed fast with the GPU-based implementation. Experiments on a standard hardware demonstrate the performance of our approach which is running on the GPU, and the results show a realistic motion of the water surface at interactive rates.

Keywords Physics-based modeling · Shallow Water equations · SPH · Real-time simulation · GPU

1 Introduction

These days work on water simulation in Computer Graphics shows highly realistic results with physics-based methods. Flow, interactions with objects, bubbles, and foam and spray with breaking are computed and described in detail. The physics-based methods usually involve modeling with Navier–Stokes equations (NSE) along with computational fluid dynamics (CFD). Since it is very time consuming to compute NSE for a large volume of water body, simplified models or ad hoc methods are proposed to simulate the water surface behavior at interactive frame rates. For example, water surface which covers a large area or has no bound like ocean is usually simulated with wave models focused on the surface motion without considering the water volume. For a more detailed simulation, the volume is simulated with 2D techniques, while the surface is described with wave equations or with 3D techniques for a small local region.

In this paper, we introduce a 2D particle-based method to simulate the water surface behavior for interactive applications. We start with Shallow Water equations (SWE) and formulate 2D particle-based SWE by applying Smoothed Particle Hydrodynamics (SPH). SPH modeling of the shallow water equations has been done in the computational science for flood simulation such as the wetting-dry phenomena or flows by dam break [2, 8, 31], where the mesh-based methods are rather inadequate to cover the huge possibly flooded domain. In Computer Graphics, SWE have been used to compute water flow for a large area with the Eulerian gridded architecture [16, 17, 38], where the flow details are dependent upon the grid size and resolution, and flow directions are confined to the grid shape, which is usually rectangular for an easy implementation. In our particle-based approach, the simulation details are not relevant to the grid but to the characteristics of the particle system, and the water particles can move in any direction representing the flow of water. Interactions with objects are integrated using the virtual particles method [19]. Virtual particles are included in the objects, and the forces between a water particle and a virtual particle are calculated in the same way as the forces between water particles. As our solution is a 2D SPH system, it can cover a larger water area than 3D SPH systems

---

H. Lee (✉) · S. Han
Dept. of Mechanical Engineering, KAIST, 335 Gwahak-ro,
Yuseong-gu, Daejeon, South Korea
e-mail: adpc9@kaist.ac.kr

with a same number of particles. 2D water particles interplay with each other and generate water surface height field where the water particles are finally located.

The next section reviews previous work on fluid simulation, for water surface behavior in particular. Section 3 briefly overviews SWE and SPH, and the proposed method is explained in Sect. 4. Section 5 illustrates the implementation of our approach, and the results of the implementation are discussed in Sect. 6. Finally we conclude in Sect. 7 including further extension of our approach.

2 Previous work

In this section we present a brief overview of the previous works on fluid simulations for water surface behavior in Computer Graphics and the SPH-based methods for SWE in nongraphics literatures.

There have been a number of approaches applying NSE to 3D Eulerian gridded solutions. [34] proposed a implicit solver with Semi-Lagrangian method which results in stable fluid simulation. [9, 10] showed realistic fluid simulation by computing NSE, where they used particle level set method to track the fluid surface. [20] solved NSE in octree data structure instead of uniform grid, and [21] presented a simulation method for resulting liquid from melting solid. [22] proposed a two-way coupled simulation framework where grid-based liquid by particle level set method can be coupled with SPH volumes. [35] used NSE for a base fluid solver and added particle systems for splash and foam. [3] presented the rigid fluid method where two-way interactions between fluids and solid are simulated. [15] simulated large bodies of water by coupling uniform 3D grid with Semi-Lagrangian method and adaptive 2D grid with height field approach. [38] proposed a similar method for free surface water simulation where Shallow Water simulation and full 3D surface simulation are coupled. Although the methods mentioned above result in highly realistic fluid animation, they are not suitable for real-time fluid simulation.

Many works used 2D NSE instead of 3D NSE which require extensive computation, to simulate a wide water area for interactive applications. [16] used SWE, which are simplified 2D NSE for Shallow Water simulation. In the method of [27], each cell's water height changes with the net flow rate through neighbor cells by pressure, which is basically same as the Shallow Water simulation, and particle systems are added for splashing simulation. [4, 5] proposed height field simulation methods with 2D Navier–Stokes solution for interactive applications, and [17] presented a stable algorithm for SWE with Semi-Lagrangian method. [37] used SWE for water body behavior and added wave simulation method for the simulation of breaking waves. Although the vertical motion of the water volume cannot be represented in their 2D grid-based methods, a large area of water surface can be represented with the simulation of horizon water flow which results in plausible wave motion on the water surface at interactive rates, and other methods for more details can be easily coupled.

Particle systems with Lagrangian framework have been applied in fluid simulations. [25] proposed an SPH-based method for fluid simulation with free surfaces at interactive rates, which was implemented to run on the GPU in [1]. [30] suggested Moving Particle Semi-Implicit method with which mixing of two fluids with different properties is simulated, and [26] used SPH to simulate fluid–fluid interactions. Particle-based approaches were proposed for viscoelastic fluid simulation in [6, 28]. [7] proposed a hybrid method where SPH for the liquid flow and 2D wave equations for surface simulation are coupled for interactive applications. As shown above, particle systems can be used to describe fluid flow and to represent bubble, splash, and foam. However, for an interactive application, the number of particles are limited, and fast surface tracking and visualization techniques are required.

Early graphics approaches on fluid simulation focused on modeling the water surface wave motion where the flow of water volume was ignored. [11] used Gerstner wave model for water surface motion including effects of depth by wave refraction, and [29] used a quadratic function for the wave shape and applied Airy model to represent wave refraction by depth. [39] proposed the wave-tracing method for the refraction in the shallow water. For real-time graphics, [14] developed an adaptive scheme for an unbounded ocean with the Gerstner wave, and [23] presented a multiband Fourier domain approach for deep water ocean waves running on the GPU. While their methods mainly focused on the generation of realistic waves particularly for a vast ocean area, [12, 33] used 2D wave equations for representing surface behaviors, and [36] developed a surface motion equation with the linearized Bernoulli's equation including a vertical derivative operator. [40] proposed a particle-based approach called Wave Particles for representing the wave motion including two-way interactions with objects. The methods based on wave equations are highly applicable to interactive applications for the fact that they incur very little computation. However, since the global flow cannot be represented, other techniques need to be coupled for the water flow as in [7].

SPH method has been applied to the solution of SWE for nongraphics applications such as flood simulation in the computational science. [2] introduced a technique based on the Riemann solver to improve the stability of the SPH method for SWE with a new artificial viscosity derived. [31] presented the variational SPH formulation of Lagrangian SWE over a general terrain. Flooding simulations of complex 3D bathymetry is performed in [8]. As they aim more at the simulation accuracy and stability than at the performance, their solutions are rather complicated and inadequate for real-time simulation, and reformulation and simplification of these works are necessary.

To represent the wafer flow with waves on the surface for interactive applications, we formulate an SPH-based form of SWE working on Lagrangian framework by modifying the standard SPH equations. Our method is straightforward to implement for real-time simulation. And since it is a 2D solution where the particles are always located on the water surface, the surface tracking is not required.

3 Overview of SWE and SPH

3.1 Shallow Water equations

SWE are derived from the full NSE by assuming inviscid and incompressible 2D flow [17]:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+g\frac{\partial h}{\partial x}=0, \tag{1}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+g\frac{\partial h}{\partial y}=0, \tag{2}$$

$$\frac{\partial h}{\partial t}+\frac{\partial}{\partial x}[u(h-b)]+\frac{\partial}{\partial y}[v(h-b)]=0, \tag{3}$$

where u and v are fluid velocity components in the x and y direction, h and b are the height of the water surface and the height of the ground in the z direction, and g denotes the gravitational constant. (1) and (2) are the momentum conservation equations, and (3) is the continuity equation. We add the assumption that the ground depth is not considered, $b=0$. Then the continuity equation becomes

$$\frac{\partial h}{\partial t}+\frac{\partial (uh)}{\partial x}+\frac{\partial (vh)}{\partial y}=0. \tag{4}$$

Equations (1), (2), and (4) can be rewritten in Lagrangian form as

$$\frac{D\mathbf{u}}{Dt}=-g\nabla h, \tag{5}$$

$$\frac{Dh}{Dt}=-h\nabla\cdot\mathbf{u}, \tag{6}$$

where $\mathbf{u}=(u,v)$ is the horizontal fluid velocity, $\frac{D}{Dt}=\frac{\partial}{\partial t}+u\frac{\partial}{\partial x}+v\frac{\partial}{\partial y}$ refers to the material derivative, and ∇ refers to the gradient. (5) is the momentum conservation equation, and (6) is the continuity equation.

3.2 Smoothed Particle Hydrodynamics

We give a brief overview on SPH especially for fluid simulation. Details are well explained in [18, 24].

SPH is a particle-based meshfree approach which is applicable to hydrodynamic simulation. The system is represented by a finite number of particles that carry individual mass and occupy individual space. The numerical interpolant of any function $A(\mathbf{r})$ is approximated as

$$A_S(\mathbf{r})=\sum_{j=1}^{N}m_j\frac{A_j}{\rho_j}W(\mathbf{r}-\mathbf{r}_j,h), \tag{7}$$

where N is the number of particles in the support domain, j is a particle index, m_j is the mass of particle j, $\mathbf{r}=(x,y,z)$ denotes the location, ρ_j and A_j are the density and the value of A at $\mathbf{r}_j$, and $W(\mathbf{r},h)$ is a smoothing kernel function with smoothing length h. Then the value of $A(\mathbf{r})$ at particle i is approximated as

$$A_i=\sum_{j=1}^{N}m_j\frac{A_j}{\rho_j}W_{ij}, \tag{8}$$

where

$$W_{ij}=W(\mathbf{r}_i-\mathbf{r}_j,h). \tag{9}$$

The gradient and the Laplacian of A_i, ∇A_i and $\nabla^2 A_i$, can be obtained with the gradient and the Laplacian of the smoothing kernel as follows:

$$\nabla A_i=\sum_{j=1}^{N}m_j\frac{A_j}{\rho_j}\nabla W_{ij}, \tag{10}$$

$$\nabla^2 A_i=\sum_{j=1}^{N}m_j\frac{A_j}{\rho_j}\nabla^2 W_{ij}. \tag{11}$$

If the function $A(\mathbf{r})$ is substituted with the density function ρ in (8), the density at particle i, ρ_i, is obtained as

$$\rho_i=\sum_{j=1}^{N}m_jW_{ij}, \tag{12}$$

which means the density approximation. The density at a position is determined by the distribution of particles of mass m_j.

4 Shallow Water Particles

In this section, we describe our 2D SPH-based shallow water simulation method, which we call Shallow Water Particles (SWP) for brevity. It is assumed that the flow is inviscid, and water particles are incompressible.

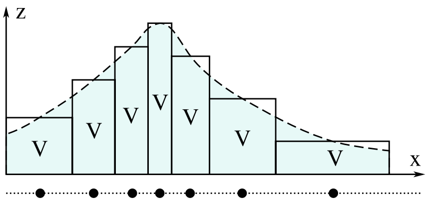

Fig. 1 The concept of SWP in 1D

4.1 SWP formulation

For 2D SPH-based approach, we define a new 2D water particle with a constant amount of mass m and the corresponding constant volume V with the assumption that the water is incompressible with the constant density of water, ρ_w, where $V = m/\rho_w$. So, the volume V_j of a particle j stays unchanged in our 2D SPH method as the mass m_j is constant in 3D SPH. 2D water particles are supposed to move horizontally on the xy plane. Figure 1 shows the concept of SWP in 1D. If particles are getting together at a place, the height will be increasing. On the contrary, if they are getting away from each other near a place, the height near the place will be decreasing. This means that the height at a place is determined by the distribution of particles of constant volume V. This is analogous to the density approximation in (12), where the density is determined by the distribution of particles of mass m. As a result, we can define an equation of the height approximation as

$$h_i = \sum_{j=1}^{N} V_j W_{ij}, \tag{13}$$

where W_{ij} is a 2D version of the smoothing kernel. Table 1 shows the analogy of variables between 3D and 2D SPH equations where the units are represented with two primary units, mass M and length L. The mass of a 3D particle corresponds to the volume of a 2D particle, and the density in 3D SPH corresponds to the height in 2D SPH. With this analogy, (7), (8), (10), and (11) become

$$A_S(\mathbf{r}) = \sum_{j=1}^{N} V_j \frac{A_j}{h_j} W(\mathbf{r} - \mathbf{r}_j, h), \tag{14}$$

$$A_i = \sum_{j=1}^{N} V_j \frac{A_j}{h_j} W_{ij}, \tag{15}$$

$$\nabla A_i = \sum_{j=1}^{N} V_j \frac{A_j}{h_j} \nabla W_{ij}, \tag{16}$$

$$\nabla^2 A_i = \sum_{j=1}^{N} V_j \frac{A_j}{h_j} \nabla^2 W_{ij}. \tag{17}$$

Table 1 Analogy between 3D and 2D SPH

3D SPH		2D SPH	
Variables	[units]	Variables	[units]
m_j	[M]	V_j	[L^3]
ρ_j	[ML^{-3}]	h_j	[L]
W_{ij}	[L^{-3}]	W_{ij}	[L^{-2}]
$\rho_i = \sum m_j W_{ij}$		$h_i = \sum V_j W_{ij}$	

If $A(\mathbf{r})$ is replaced with $h(\mathbf{r})$, (16) becomes

$$\nabla h_i = \sum_{j=1}^{N} V_j \nabla W_{ij}. \tag{18}$$

We formulate 2D SPH equations by modifying 3D SPH equations as a special case where 2D water particles are defined with individual volume. The 2D SPH equations can be applied to SWE, (5) and (6), written in Lagrangian form. In 3D SPH, as the mass in 3D water particles stays unchanged, the mass conservation is satisfied. In the same way, as the volume in 2D water particles is unchanged, it is conserved, by which the mass is also conserved with the incompressible condition. Therefore, we do not need to consider the continuity equation (6). For a 2D water particle i, the momentum conservation equation (5) becomes

$$\frac{D\mathbf{u}_i}{Dt} = -g\nabla h_i, \tag{19}$$

where (18) can be applied, and then (19) is rewritten as

$$\frac{D\mathbf{u}_i}{Dt} = -g \sum_{j=1}^{N} V_j \nabla W_{ij}. \tag{20}$$

(13) and (20) are the 2D SPH-based representation of SWE, or the SWP equations, where (13) means the height approximation, and (20) means the pressure force.

4.2 Viscous force

Although the flow is assumed to be inviscid in SWE, the simulation can be unstable without damping. Therefore, we add viscous terms to the SWE for stable and realistic surface motion behavior. When SWE are derived from NSE, if viscous terms are considered together, the momentum conservation equations (1) and (2) become

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + g\frac{\partial h}{\partial x} = \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \tag{21}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + g\frac{\partial h}{\partial y} = \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right), \tag{22}$$

where ν denotes the dynamic viscosity. Equations (21) and (22) can be rewritten in Lagrangian form as

$$\frac{D\mathbf{u}}{Dt} = -g\nabla h + \nu\nabla^2\mathbf{u}, \tag{23}$$

where $\nu\nabla^2\mathbf{u}$ means the viscous force. If $A(\mathbf{r})$ is substituted with $\mathbf{u}$ in (17), the viscous force is represented as

$$\frac{D\mathbf{u}_i}{Dt} = \nu\sum_{j=1}^{N} V_j \frac{\mathbf{u}_j}{h_j}\nabla^2 W_{ij}. \tag{24}$$

4.3 Smoothing kernels

We apply two smoothing kernels proposed in [25] to the SWP equations. $W_{\text{poly6}}(\mathbf{r},h)$ is used for the height approximation in (13), and $W_{\text{spiky}}(\mathbf{r},h)$ is used to compute pressure and viscous force in (20) and (24). As the two kernels are designed for 3D SPH simulation, their coefficients should be modified to satisfy the normalization condition in 2D space,

$$\int W(\mathbf{r})\,d\mathbf{r} = 1, \tag{25}$$

where $\mathbf{r} = (x, y)$. By (25), the kernels for the 2D SPH-based solution become

$$W_{\text{poly6}}(\mathbf{r},h) = \frac{4}{\pi h^8}\begin{cases}(h^2 - r^2)^3, & 0 \le r \le h,\\ 0 & \text{otherwise},\end{cases} \tag{26}$$

$$W_{\text{spiky}}(\mathbf{r},h) = \frac{10}{\pi h^5}\begin{cases}(h - r)^3, & 0 \le r \le h,\\ 0 & \text{otherwise},\end{cases} \tag{27}$$

where h denotes the smoothing length, and $r = |\mathbf{r}|$.

4.4 Simulation

SWP simulation is forwarded with the execution of next process in a single time step.

Integration
Height approximation
Force computation

Integration updates the particle's velocity and position, and moves the particle on the xy plane. The velocity is updated with the pressure and viscous force computed at the previous time step, and the position is updated with the new velocity.

Height approximation determines the height of particles by (13). The z value of each particle is updated with the vertical motion of particles in z direction. As a result, the particles are positioned on the water surface.

Force computation computes the force interaction between particles where the pressure and viscous force are determined by (20) and (24).

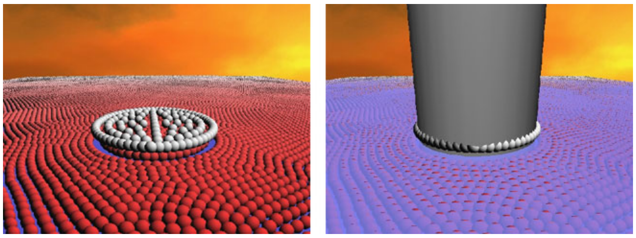

Fig. 2 Virtual particles placed in an object

4.5 Solid-to-fluid interaction

One-way interactions from solid-to-fluid are handled in a simple way using the virtual particles proposed in [19] to treat the solid boundary conditions. An object is assumed to occupy an area on the 2D space where the virtual particles are located as shown in Fig. 2. The virtual particles are treated as 2D water particles with a different amount of water volume from that of the original water particles, and the forces from the virtual particles to the water particles are integrated in the same way as the forces between water particles are computed. When an object is moving, the virtual particles which belong to the object have the same motion.

5 Implementation

We implemented our system to work on the GPU using [13] which supports a GPU-based particle simulation with a uniform hash grid where the particles are sorted by the parallel radix sort algorithm [32]. We integrated our SWP method to work with the hash-based particle simulation structure. Virtual particles and water particles are identified by simply indexing in groups, and the forces from virtual particles to water particles and the forces between water particles are only computed.

To render the water surface in our demo system, the height field of the surface are obtained with the render-to-framebuffer method supported by OpenGL. After a time step of the SWP simulation, the positions of the particles are computed. Then, the particles are projected onto the xy plane and rendered as sprites with a kernel texture which represents the 2D smoothing kernel function. Figure 3 shows the height field obtained by the render-to-framebuffer, which can be used as a height map. This process works same as the height approximation by (18). As a result, all the steps from the particle simulation to rendering are done on the GPU.

6 Results and discussion

Figure 4 shows the images of our SWP simulation. When the particles get closer over a place, the water surface becomes

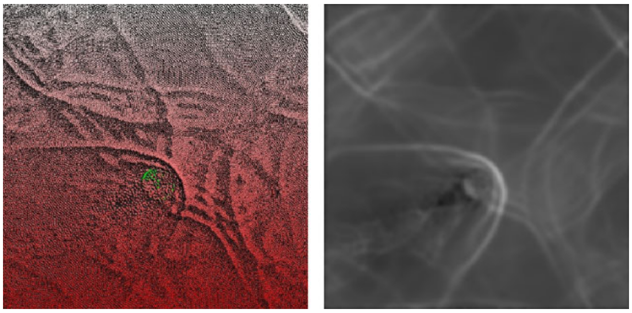

Fig. 3 Height approximation with render-to-framebuffer

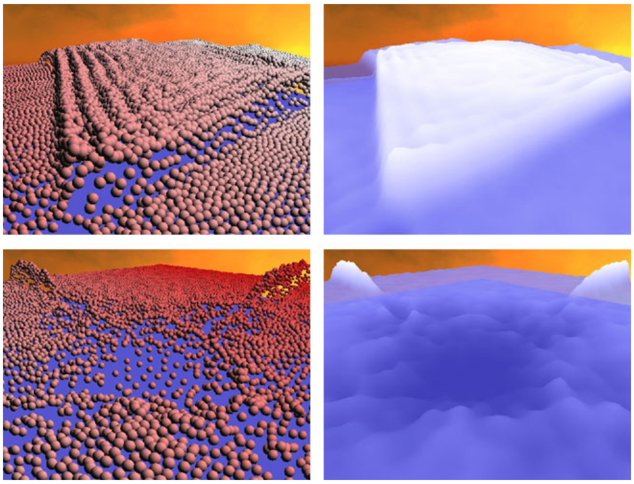

Fig. 4 Particles placed with high density (*top*) and with low density (*bottom*)

higher, while the surface becomes lower when the particles are far from each other. It is because the height in our SWP method means the 2D spatial distribution of particles, or the 2D density, which is different from the constant water density ρ_w. Therefore, near the peak of waves with higher 2D density, the water surface waves can be represented in more detail than around. On the contrary, in the shallow region with lower 2D density, the water surface is represented in less detail than around. This shows that the distribution of particles results in adaptive sampling effect. However, if the particles are far away than the smoothing length in the shallow region, the under-sampling causes bumpy surface when rendered, because we use kernels with a fixed smoothing length. For favorable adaptive sampling and rendering of the surface, the variable smoothing length SPH method [31] can be a solution with more computational cost.

In our method, there is no predefined connectivity between particles, and they can move freely around, which can show the horizon water flow more efficiently than the previous methods based on Eulerian framework with 2D NSE [16, 17, 37] or with SWE [4, 5, 27]. The surface wave motion can be represented as is in the approaches based on wave equations [12, 33, 36, 40], but they cannot simulate the horizon flow. As all the water particles are located on the surface, our 2D particle system can cover a large volume of water than 3D particle system with a same number of particles, and the surface extraction is not required.

Table 2 Performance of the SWP simulation in frame rate

Num. of particles	Hash grid size 16 × 16	32 × 32	64 × 64
16384	37	140	310
32768	13	49	135
65536	3	14	43

Interactions for solid-to-fluid coupling are shown in Fig. 5. Virtual particles are simply located inside an object and move with it. The water particles near the object show a plausible surface behavior. In this paper we have handled a simple case of interactions; however, we need to integrate more advanced technique to deal with the interactions between virtual particles and water particles.

To demonstrate the performance of our approach, we simulated on a standard PC with a 1.86 GHz Core2 Duo processor and GeForce 9800 graphics card. Using the GPU-based particle simulation framework [13], we can simulate about 32,000 particles with the SWP method at 130 fps. Table 2 shows the speed of our method for a number of cases. As the hash grid size increases, the frame rate increases because the number of particles in the neighbor cells is reduced and more cells can be processed in parallel. Moreover, the more the particles are evenly spaced, the faster it works because the maximum number of particles in a cell becomes lower. The results show that our approach works fast and are highly applicable to interactive applications.

7 Conclusion

We have introduced a 2D SPH particle-based solution of SWE to simulate a realistic water surface behavior over a large volume of water for interactive applications. We formulated 2D SPH equations for Shallow Water simulation by analogizing the height approximation in 2D SPH with the density approximation in 3D SPH. Both the horizon flow and the surface motion of water can be described with the method, and we have shown that it works fast enough for real-time simulation with the GPU-based implementation.

As a future plan, we will extend our approach to reflect the depth effect with the consideration of the ground depth and to include the fluid-to-solid interactions with more sophisticated virtual particles technique.

Acknowledgements This research was supported by WCU (World Class University) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R31-2008-000-10045-0).

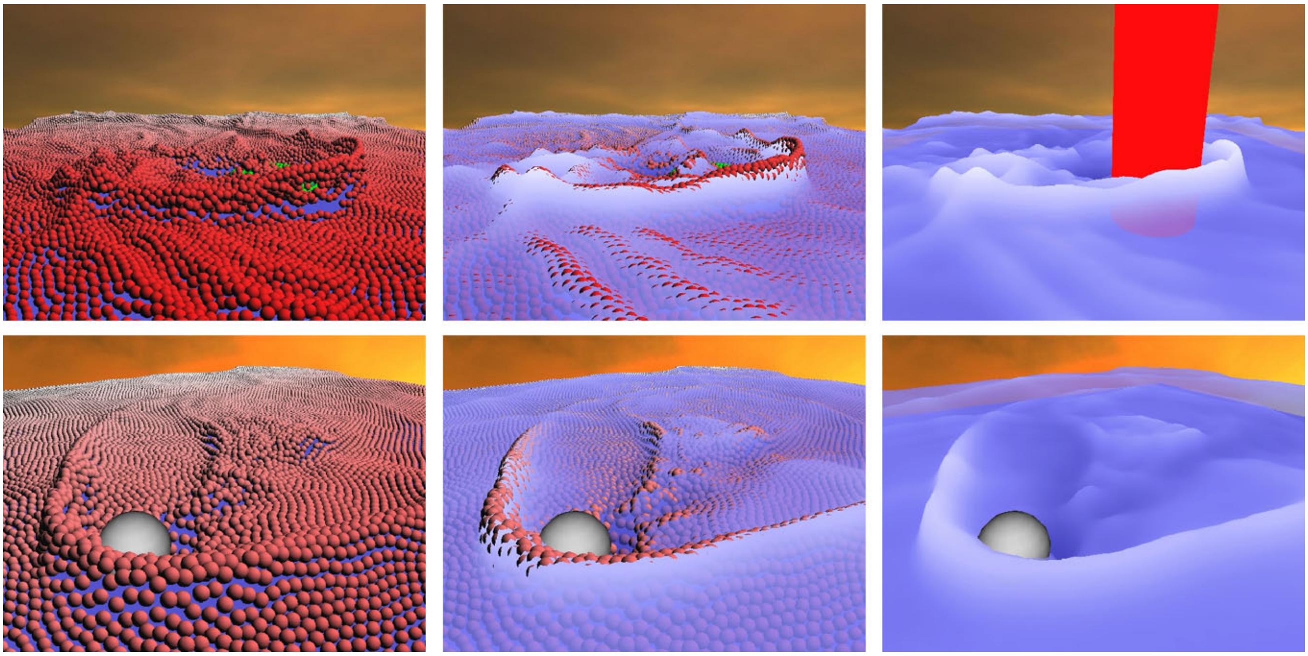

Fig. 5 Rendered images of solid-to-fluid interactions with 32768 particles and a 64 × 64 hash grid

References

1. Amada, T., Imura, M., Yasumuro, Y., Manabe, Y., Chihara, K.: Particle-based fluid simulation on GPU. In: ACM Workshop on General-Purpose Computing on Graphics Processors, vol. 41, p. 42 (2004)
2. Ata, R., Soulaïmani, A.: A stabilized SPH method for inviscid shallow water flows. Int. J. Numer. Methods Fluids **47**(2), 139–159 (2005)
3. Carlson, M., Mucha, P.J., Turk, G.: Rigid fluid: animating the interplay between rigid bodies and fluid. ACM Trans. Graph. **23**(3), 377–384 (2004)
4. Chen, J.X., Lobo, N.D.V.: Toward interactive-rate simulation of fluids with moving obstacles using Navier–Stokes equations. Graph. Models Image Process. **57**(2), 107–116 (1995)
5. Chen, J.X., Lobo, N.D.V., Hughes, C.E., Moshell, J.M.: Real-time fluid simulation in a dynamic virtual environment. IEEE Comput. Graph. Appl. **17**(3), 52–61 (1997)
6. Clavet, S., Beaudoin, P., Poulin, P.: Particle-based viscoelastic fluid simulation. In: SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 219–228. ACM Press, New York (2005)
7. Cords, H.: Mode-splitting for highly detailed, interactive liquid simulation. In: GRAPHITE '07: Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia, pp. 265–272 (2007)
8. de Leffe, M., Le Touzé, D., Alessandrini, B.: SPH modeling of shallow-water coastal flows. In: ICHD '08: Proceedings of the 8th International Conference on Hydrodynamics (2008)
9. Enright, D., Marschner, S., Fedkiw, R.: Animation and rendering of complex water surfaces. ACM Trans. Graph. **21**(3), 736–744 (2002)
10. Foster, N., Fedkiw, R.: Practical animation of liquids. In: SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 23–30 (2001)
11. Fournier, A., Reeves, W.T.: A simple model of ocean waves. SIGGRAPH Comput. Graph. **20**(4), 75–84 (1986)
12. Gomez, M.: Interactive simulation of water surface. In: Deloura, M. (eds.) Game Programming Gems, pp. 187–194. Charles River Media, Boston (2000)
13. Green, S.: CUDA particles (2007). NVIDIA CUDA SDK v2.2
14. Hinsinger, D., Neyret, F., Cani, M.P.: Interactive animation of ocean waves. In: SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 161–166. ACM Press, New York (2002)
15. Irving, G., Guendelman, E., Losasso, F., Fedkiw, R.: Efficient simulation of large bodies of water by coupling two and three dimensional techniques. ACM Trans. Graph. **25**(3), 805–811 (2006)
16. Kass, M., Miller, G.: Rapid, stable fluid dynamics for computer graphics. In: SIGGRAPH '90: Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques, pp. 49–57 (1990)
17. Layton, A., van de Panne, M.: A numerically efficient and stable algorithm for animating water waves. Vis. Comput. **18**(1), 41–53 (2002)
18. Liu, G.R., Liu, M.B.: Smoothed Particle Hydrodynamics: A Meshfree Particle Method. World Scientific, Singapore (2003)
19. Liu, M.B., Liu, G.R., Lam, K.Y.: Investigations into water mitigation using a meshless particle method. Shock Waves **12**(3), 181–195 (2002)
20. Losasso, F., Gibou, F., Fedkiw, R.: Simulating water and smoke with an octree data structure. ACM Trans. Graph. **23**(3), 457–462 (2004)
21. Losasso, F., Irving, G., Guendelman, E., Fedkiw, R.: Melting and burning solids into liquids and gases. IEEE Trans. Vis. Comput. Graph. **12**(3), 343–352 (2006)
22. Losasso, F., Talton, J., Kwatra, N., Fedkiw, R.: Two-way coupled SPH and particle level set fluid simulation. IEEE Trans. Vis. Comput. Graph. **14**(4), 797–804 (2008)
23. Mitchell, J.: Real-time synthesis and rendering of ocean water. Tech. rep., ATI Research Technical Report (2005)
24. Monaghan, J.J.: Smoothed particle hydrodynamics. Ann. Rev. Astron. Astrophys. **30**(1), 543–574 (1992)
25. Müller, M., Charypar, D., Gross, M.: Particle-based fluid simulation for interactive applications. In: SCA '03: Proceedings of the

2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 154–159. ACM Press, New York (2003)
26. Müller, M., Solenthaler, B., Keiser, R., Gross, M.: Particle-based fluid-fluid interaction. In: SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 237–244. ACM Press, New York (2005)
27. O'Brien, J.F., Hodgins, J.K.: Dynamic simulation of splashing fluids. In: CA '95: Proceedings of the Computer Animation, p. 198. IEEE Computer Society, Los Alamitos (1995)
28. Paiva, A., Petronetto, F., Lewiner, T., Tavares, G.: Particle-based viscoplastic fluid/solid simulation. Comput. Aided Des. **41**(4), 306–314 (2009)
29. Peachey, D.R.: Modeling waves and surf. In: SIGGRAPH '86: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, pp. 65–74 (1986)
30. Premože, S., Tasdizen, T., Bigler, J., Lefohn, A., Whitaker, R.: Particle-based simulation of fluids. Comput. Graph. Forum **22**(3), 401–410 (2003)
31. Rodriguez-Paz, M., Bonet, J.: A corrected smooth particle hydrodynamics formulation of the shallow-water equations. Comput. Struct. **83**(17–18), 1396–1410 (2005)
32. Satish, N., Harris, M., Garland, M.: Designing efficient sorting algorithms for manycore GPUs. In: IPDPS '09: Proceedings of the 23rd IEEE International Parallel and Distributed Processing Symposium, pp. 1–10 (2009)
33. Schneider, J., Westermann, R.: Towards real-time visual simulation of water surfaces. In: VMV '01: Proceedings of the Vision Modeling and Visualization Conference 2001, pp. 211–218 (2001)
34. Stam, J.: Stable fluids. In: SIGGRAPH '99: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, pp. 121–128 (1999)
35. Takahashi, T., Fujii, H., Kunimatsu, A., Hiwada, K., Saito, T., Tanaka, K., Ueki, H.: Realistic animation of fluid with splash and foam. Comput. Graph. Forum **22**(3), 391–400 (2003)
36. Tessendorf, J.: Interactive water surfaces. In: Kirmse, A. (ed.) Game Programming Gems4, pp. 265–274. Charles River Media, Boston (2004)
37. Thürey, N., Muller-Fischer, M., Schirm, S., Gross, M.: Real-time breaking waves for shallow water simulations. In: PG '07: Proceedings of the 15th Pacific Conference on Computer Graphics and Applications, pp. 39–46 (2007)
38. Thürey, N., Rüde, U., Stamminger, M.: Animation of open water phenomena with coupled shallow water and free surface simulations. In: SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 157–164. ACM Press, New York (2006)
39. Ts'o, P.Y., Barsky, B.A.: Modeling and rendering waves: wave-tracing using beta-splines and reflective and refractive texture mapping. ACM Trans. Graph. **6**(3), 191–214 (1987)
40. Yuksel, C., House, D., Keyser, J.: Wave particles. ACM Trans. Graph. **26**(3), 99 (2007)

Hyokwang Lee is a master-doctoral integrated course student in the Intelligent CAD laboratory of the Department of Mechanical Engineering at KAIST. He received his B.S. degree from the Department of Mechanical Engineering at Yonsei University in 2003. His current research interests include Computer Aided Design, Computer Aided Engineering, Modeling and Simulation, and real-time simulation and visualization.

Soonhung Han is the Chair of the Division of Ocean Systems Engineering (www.kaist-bocean.org) and a professor at the School of Mechanical Engineering (http://me.kaist.ac.kr) of KAIST (www.kaist.edu). He is leading the Intelligent CAD laboratory (http://icad.kaist.ac.kr) of KAIST and the STEP community of Korea (www.kstep.or.kr). His research interests include STEP (ISO standard for the exchange of product model data), VR (virtual reality) for engineering design, and knowledge-based design systems. His domain of interests include shipbuilding and automotive. More information can be found from his personal web page at http://icad.kaist.ac.kr/~shhan.