Single-Phase Trapped Air Simulation in Water Flow

Seungtaik Oh
ETRI
218 Gajeong-ro
Yusong
305-700, Daejon, Korea
stoh@etri.re.kr

Il Kwon Jeong
ETRI
218 Gajeong-ro
Yusong
305-700, Daejon, Korea
jik@etri.re.kr

ABSTRACT

We introduce a novel practical single-phase particle simulation for trapped air bubbles in a turbulent water flow. Our model for a trapped air bubble is a low-density rigid body with a spherical shape, and our bubble interacts with water and other rigid bodies in a fully two-way manner. Our bubble is created at a trapped air pocket computed from the water volume. Stable and realistic bubble interactions are achieved using an impulse-based boundary force with non-positive coefficients of restitution. Subgrid-scale bubbles are also created to add more details using precomputed bubble data and an oscillating bubble mesh is used in rendering stage instead of a spherical shape for a soft look of the bubble surface. Our method can be easily implemented by extending an existing rigid body interaction of fluid solver, and it is fast compared to two-phase simulation because we do not simulate the air part.

Keywords

Bubble, Fluid, SPH.

1 INTRODUCTION

The existence of trapped air bubbles is one of the most attractive features in water flow. Trapped air bubbles mainly come from trapped air pockets captured by turbulent water flow. Two-phase simulation is definitely the most accurate way to capture trapped air pockets in water flow and to simulate the complex behaviour of air bubbles, such as breaking and merging [Col03, Hon03, Hon05, Hon08, Son05, Zhe06]. However, a two-phase simulation of water and air suffers from heavy computational cost and slow simulation time since the air part should be simulated in the same way as the water part.

In the present paper we propose a novel and practical single-phase trapped air bubble simulation, within a particle-based framework, Smoothed Particles Hydrodynamics (SPH). Our main idea is to model a trapped air bubble as a low-density rigid body with a spherical shape. In our approach, the air bubbles are naturally coupled to the water and the other rigid bodies in a fully two-way manner since a trapped air bubble is simulated as a rigid body. We follow the impulse-based approach [Oh09] for stable and realistic rigid body interaction in SPH, and in particular we apply nonpositive impulses for the interaction between bubbles. Nonpositive impulses produce a weak repulsion, and so it allows approaching bubbles to overlap. A natural buoyant force for our bubble model is also induced by the impulse-based boundary force. Thanks to the single-phase feature of our method, one can perform an air bubble simulation as fast as a normal single-phase water simulation with a fairly small amount of additional memory.

Our approach provides some degree of freedom for user to enhance existing results. In the rendering stage, our spherical bubble is replaced by an elliptically oscillating shape whose axis are dynamically changed according to the bubble velocity. In addition, sub-grid scale bubbles can be recursively added in a post-processing step. The creation and amount of sub-grid scale bubbles are controlled by the gap between an air pocket and bubbles inside.

This paper is organized as follows. In the next section, the prior literature is surveyed. Section 3 explains our new bubble model and related simulation issues. Section 4 explains how to create an oscillating bubble mesh and subgrid-scale bubbles. Section 5 presents and discusses the simulation results and conclusion is presented in the last section.

2 PREVIOUS WORK

Two phase simulations have been studied for realistic air bubbles in grid-based Eulerian framework [Kan00, Hon03, Hon05, Son05, Zhe06]. Some hybrid methods

have also been presented to capture small scale air bubbles [Hon08, Mih09].

A real-time method is suggested for bubbles and foams in a shallow water simulation in [Thu07]. Millions of dispersed bubbles were simulated using a stochastic solver at a subgrid level in [Kim10].

In particle-based Lagrangian approaches, two-phase simulations with a high density ratio were studied by making some modifications to the SPH equations in [Col03, Sol08]. A two-phase SPH fluid in a weak sense was simulated using a dynamic particle layer at water-bubble interface in [Mue05]. Small scale bubbles including foam were simulated within an SPH framework by [Cle07].

Trapped air bubbles were simulated as spheres in [Gre04]. The bubbles interact with water in an one-way manner and several sophisticated bubble interactions are applied. The bubble simulation is performed in a sub-simulation of a water simulation with a much smaller time step. Water might exist inside the bubble because the bubble cannot affect the water flow. Nonetheless, the idea that a trapped air bubble is represented by a sphere was practically simple and it became a motivation for our work. The underlying idea of handling trapped air bubbles as spheres are the same, but now a two-way coupling between the water and the trapped air bubble increases the realism of our method.

3 TRAPPED AIR BUBBLES

Let us start with a brief introduction to SPH fluid simulation.

3.1 SPH Fluid Simulation

The fluid is updated by the two discrete SPH equations derived from the momentum and continuity equations [Mon94]:

$$\frac{du_a}{dt} = -\sum_b m_b \left(\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2} + \Pi_{ab} \right) \nabla_a W_{ab} + f_a \quad (1)$$

$$\frac{d\rho_a}{dt} = \sum_b m_b (u_a - u_b) \nabla_a W_{ab}, \quad (2)$$

where the summations run over all neighboring particles within the support of the kernel W, u is the velocity, p is the pressure, ρ is the density, Π is the artificial viscosity, and f is the body force such as gravity. The pressure is given by an equation of state

$$p = p_0((\rho/\rho_0)^\gamma - 1), \quad (3)$$

where $\gamma = 7$ for water. With this formulation, we obtain a weakly compressible fluid and the density variation is determined by the ratio of the maximum velocity to the speed of sound, c_s, more precisely: $d\rho/\rho \approx (u_{max}/c_s)^2$. For an incompressible fluid such as water, we set $c_s = 10u_{max}$ and obtain a 1% density variation. All simulations in this paper use $u_{max} = 10$ m/s.

3.2 Bubble Model

Our method simulates a trapped air bubble by a low-density rigid body. Every trapped air bubble is represented by a spherical rigid body, and its shape is unchanged during the simulation. For a shape change of bubble, the spherical bubbles are replaced with dynamically changing shapes at rendering stage, which is computationally efficient compared to applying soft body simulation for each bubble.

The stability of a weakly compressible SPH is guaranteed only under the condition that the density ratio is less than ten to one in the fluid particles evolving in the simulation [Col03][Sol08], and the same is true for our rigid body interaction, such as water with bubbles. For this reason, we would make use of a relaxed density for our bubble model and set the density of our bubble to one-tenth of that of water, although the real density ratio is one-one thousandth. A density difference between water and a boundary object generates a natural buoyancy on the boundary object in our rigid body interaction scheme.

3.3 Bubble Dynamics

Once a trapped air bubble is modeled as in the previous manner, the interactions of the bubbles with the water and other rigid bodies are almost straightforward since the bubble is just a rigid body.

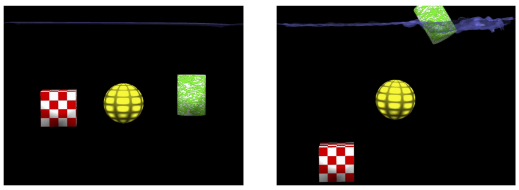
3.3.1 Impulse-based Boundary Force

Monaghan suggested a *boundary force scheme*, which is a common method of solving rigid body interactions in SPH [Mon94][Mon03]. An improved boundary force based on impulse is suggested for stable and realistic complex rigid body interactions such as a huge stacking problem [Oh09]. Given an impulse J and a time step Δt, the complete impulse-based boundary force F_I for a fluid particle and a rigid body is

$$F_I = J/\Delta t + F_f, \quad (4)$$

where the fluid force F_f consists of the pressure of the fluid particle and the friction between the fluid particle and the rigid body.

A benefit in using the impulse-based boundary force is that the density difference between the fluid and the bubble generates a natural buoyancy for our bubble model. The impulse-based boundary force has a water pressure term F_f in Equation (4), which make a force difference between on the top and the bottom of the bubble. Fig. 1 shows three rigid bodies with different densities experiencing the correct buoyancy from the impulse-based boundary force. At the beginning, the objects have zero relative velocity with respect to the water, that is, $J = 0$, but it experiences a non-zero buoyancy boundary force from the fluid pressure.



(a) (b)

Figure 1: Buoyancy test of three objects. From the left the density ratios of objects to water are 10 (cube), 1 (sphere), and 0.1 (cylinder). The buoyant force is naturally generated by the impulse-based boundary force without artificial buoyancy.

An auxiliary buoyant force is added to increase the rising velocity, as in [Mue05, Mih09]. The Stokes velocity we are going to use is the terminal velocity of an air bubble maintaining its spherical shape while rising in water, which is given by

$$v_s = \frac{2}{9}\frac{gR^2}{\nu}, \tag{5}$$

where g is the gravity, R is the radius of the bubble, and ν is the kinematic viscosity of water (see [Bat67], p. 234). The auxiliary buoyant force controls the rising velocity of bubbles in the vertical direction.

A drag force can also be simulated by the fluid force term, more precisely, the friction. The frictional force is given by a damping model and its magnitude depends on the fluid viscosity and the object's friction coefficient:

$$F_{drag} = -(\mu + k)\Delta v, \tag{6}$$

where μ is the fluid viscosity, k is the object's friction coefficient, and Δv is the relative viscosity between the fluid and the object.

3.3.2 *Nonpositive Impulse for Bubble Interaction*

The bubble interaction can be also attained by the impulse-based boundary force in a unified way since our bubble model is rigid. For the interaction between bubbles, we use a modified impulse with nonpositive coefficient of restitution(COR) denoted by ε. Generally, only $\varepsilon \geq 0$ is considered for non-penetration in interaction. However, in principle, a nonpositive COR is also admissible and it is observed that $\varepsilon = -1$ generates no interaction force and, for $-1 < \varepsilon < 0$, a weak repulsive force occurs. Based on this, we control COR by setting $\varepsilon \leq 0$ for proper bubble interactions such as merging and splitting. Attraction between bubbles can be realized by setting $\varepsilon_1 < 0$ for approaching bubbles. For overlapping bubbles, the overlapping can be accelerated or decelerated by setting another ε_2 with $\varepsilon_1 < \varepsilon_2 \leq 0$.

	procedures
1	compute neighboring info
	- construct search grid for current particle positions
	- **find trapped air pocket and create bubbles**
2	calculate particle interactions
	- **compute bubble interactions**
	- **delete bubbles**
3	update particles position and velocity
	- **update bubble motion**
4	go back to 1

Table 1: Procedures of trapped air simulation

3.4 Bubble Creation and Deletion

For the creation of a bubble, we follow an approach based on the flood fill algorithm, similar to that in [Gre04]. The existing neighbor-searching grid is used for the grid structure for the flood fill algorithm to minimize the overhead. We choose an atmosphere cell with no particles first and then all atmosphere cells can be found by the flood fill algorithm. The empty cells not marked as atmosphere become candidates for air pockets. For a randomly chosen air pocket component, a bubble is created by computing the center and the optimal size of bubble fit to the air pocket component.

A bubble is deleted when one of the following condition is satisfied:(1)Bubble age is too high. (2) A bubble has been floating on the free surface too long. (3) Two bubbles overlap too much (4) A floating bubble experiences a strong impact by water.

The overall procedure for the trapped air simulation in the SPH formulation is shown in Table 1.

4 BUBBLE MESHING AND SUB-GRID SCALE BUBBLES

4.1 Oscillating Bubble Mesh

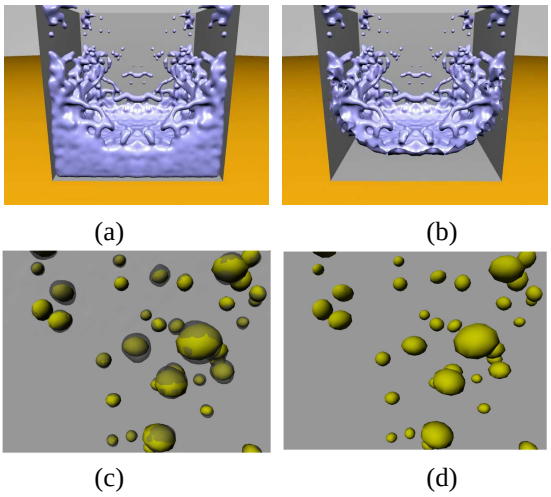
If a bubble is rendered as a sphere, the bubble surface looks hard and unrealistic. In order to overcome this defect, the spherical bubble is replaced by an oscillating bubble mesh during the rendering stage. The oscillating bubble mesh has an elliptic shape whose axis is dynamically rotating according to the bubble velocity direction.

An arbitrarily oriented ellipsoid with center c is given by $(x-c)^T A(x-c) = 1$, where A is a positive definite symmetric matrix. In fact, $A = R^T \Lambda R$ for a rotation matrix R and a diagonal matrix Λ. For our bubble mesh, R is chosen to be a rotation matrix mapping the direction of the bubble's velocity v to the z-axis, that is, the rotation axis is $e_z \times v$ and the rotation angle is $\measuredangle(e_3, v)$. Letting $\Lambda = diag(\alpha^{-2}, \beta^{-2}, \gamma^{-2})$, we set

$$\alpha(t) = r + \delta(t) \tag{7}$$

$$\beta(t) = \alpha(t) \tag{8}$$

$$\gamma(t) = \frac{3V}{4\pi\alpha(t)\beta(t)}, \tag{9}$$



(a) (b)

(c) (d)

Figure 2: Free surface mesh and its application to bubble visualization. The upper row shows a comparison between the full surface mesh (a) and the free surface mesh (b) in a general water simulation. In the lower row, the full surface mesh (c) and the free surface mesh (d) results in bubble visualization are given. In (c), one can see some artifacts caused by the full surface mesh (grey) at the water–bubble interface.

where r is the radius of the bubble, $\delta(t)$ is an oscillating function with an attenuation in time, and V is the bubble's volume. The oscillating function $\delta(t)$ is given by

$$\delta(t) = \begin{cases} kr(t/T-1)^2 \sin(2n\pi t/T) & : 0 < t \leq T \\ 0 & : t > T \end{cases} \tag{10}$$

where $k \ll 1$ is the magnitude, T is the maximum oscillation time, and n is the frequency. From the the definition of $\gamma(t)$, we see that the volume of our oscillating bubble is preserved. This oscillating bubble mesh creates the look of an elastic bubble without the need for soft body modelling of the bubble in the simulation.

A free surface extraction for SPH water particles is needed for effective rendering of bubble mesh. In our situation, a conventional water mesh extraction from the water particles causes a rendering problem in the proximity of bubbles since two interfaces, water and the bubble mesh, co-exist around a bubble (Fig. 2(c)). To remove this problem, we modify our Marching Cubes algorithm by skipping grid cells close to rigid boundaries or bubbles so that the water mesh has a free surface part only. The free surface mesh has no polygons near a boundary object (Fig. 2(b)). Since our bubble is simulated as a boundary object, the free surface only extraction is also available for our bubble. A free surface mesh is useful in a general situation since the size of the mesh data is significantly reduced and a clear rendering around the interface of a boundary object is possible(Fig. 2(d)). Once a free surface only mesh is obtained, it is straightforward to substitute a spherical bubble shape with the oscillating bubble shape.

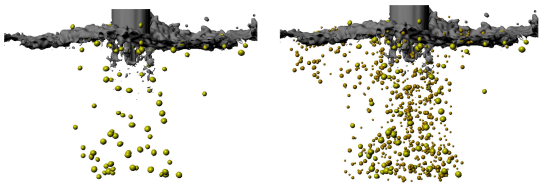


Figure 3: Comparison for subgrid-scale bubbles: without subgrid-scale bubbles (a) and with subgrid-scale bubbles (orange)(b).

4.2 Sub-grid Scale Bubble Generation

Our trapped air bubble is slightly smaller than the corresponding trapped air pocket consisting of cubic cells in the neighbor searching grid. Sub-scale bubbles whose size is less than the search grid size are created at the gap between the bubble and the pocket. The exact computation of the gap size, however, is expensive and impractical. Roughly, the gap size is inversely related to the area fraction in water, which is computed as the ratio of the number of the surface particles of a bubble insider the water to the number of all surface particles of the bubble. With the area fraction in water, we can determine the numbers and positions of the subgrid-scale bubbles around a newly created trapped air bubble. The creation sites exist outside of the trapped air bubble and the sites are randomly chosen from the mesh vertices of three concentric spheres whose center is that of the existing trapped air bubble.

The subgrid-scale bubbles represented by particles are passively advected by the water flow and the second order Runge–Kutta method is used for the time integration of its motion. The vertical component of the bubble velocity is also controlled by the Stokes velocity and u_{max} in the same way as before. Sub-grid scale bubble generation can be done recursively as decreasing the size so that one can obtain a sufficient number of subgrid-scale bubbles. A comparison result for subgrid-scale bubbles is shown in Fig. 3.

This process is a post-process, performed after the simulation, and so artists have some control to make their own subgrid-scale bubbles as needed.

5 RESULTS AND DISCUSSION

Some examples are presented to show the effectiveness of our method. All the simulations were performed on a machine with two quad core CPUs at 2.33 GHz. In all simulations, free surface meshes were used for rendering the water and a trapped air bubble was rendered using an oscillating bubble mesh. The frames per second rate (FPS) was set to 30 and so a frame represents 1/30 sec.

(a) (b)

Figure 4: Pouring water.

The first example is water pouring (Fig. 4). Water is poured by an inflow into an initial volume of water with 474k particles. The maximum number of water particles reaches 923k at the end of simulation. Totally 1140 trapped air bubbles are created and in order to add details 3050 subgrid-scale bubbles are generated by a post-process. It takes 5–12 minutes for a single frame, and the overhead for the trapped air bubbles increases the simulation time by 1.9% only for the main simulation, and only by 5.8% including the subgrid-scale bubble generation.

The second example is a diving bunny simulation (Fig. 5). As mentioned earlier, a trapped air bubble is controlled so that it is not created in the interior of the bunny. In total, 61 trapped air bubbles are created and 1716 subgrid-scale bubbles are added by the post-processing. The simulation has 474k water particles and it takes about five minutes to simulate one frame of the water and trapped air bubbles. In this case, the overhead for the main trapped air bubbles is negligible, and, including the subgrid-scale bubble generation, the overhead is estimated to be 8.1%. An artificial buoyant force based on Stokes law was applied to increase the rising velocity of the subgrid-scale bubbles. Note, too, that since we had no water mesh around the bunny from the free surface mesh extraction, the bunny is able to be clearly rendered, which is the main benefit of free surface mesh extraction.

Our current method has the following limitations. The density ratio of the water to our trapped air bubble is set to ten to one, which is one hundred times smaller than the real density ratio. We expect that more realistic results can be achieved if we can increase the density ratio while preserving the time step and stability. Some advanced techniques could improve the reality of the oscillating bubble mesh by considering the nearby dynamics of water, such as the pressure and vorticity. Although the oscillating bubble mesh is used in rendering instead of a spherical bubble, we still have a problem in simulating a natural bubble's merging and splitting. It should also be mentioned that our SPH solver performance is not good enough because no acceleration method has been applied as yet, so we expect that some GPU techniques can improve the performance significantly.

6 CONCLUSION

We have presented a new practical method to simulate water and trapped air bubbles in a single-phase SPH framework. Our trapped air bubble is modeled as a light weight spherical rigid body and its interaction is computed through a stable and realistic impulse-based boundary force. The trapped air bubbles interact with the water in a fully two-way manner, which increases the realism of the water simulation. Sub-grid scale bubbles are also created to add details using the existing trapped air bubble data and oscillating bubble meshes are rendered instead of spherical shapes to provide a soft look to the bubble surfaces. Finally, our method is almost as fast as single-phase water simulation because we don't simulate the air, and it has been shown that we have a trapped air simulation with an overhead less than 10% of that for water-only simulations.

7 ACKNOWLEDGMENTS

We also give many thanks to our colleagues in ETRI for their continuous support. This work was supported by the 'Cross-Ministry Giga KOREA Project' of the Ministry of Science, ICT and Future Planning, Republic of Korea[GK130100, Development of Interactive and Realistic Massive Giga-Content Technology].

8 REFERENCES

[Bat67] G.K. Batchelor, *An Introduction to Fluid Dynamics*, Cambridge University Press, 1967.

[Cle07] P.W. Cleary, S.H. Pyo, M. Prakash and B.K. Koo, Bubbling and frothing liquids, *ACM Trans. Graph.(SIGGRAPH 07)* 26(3), Article 97, 2007.

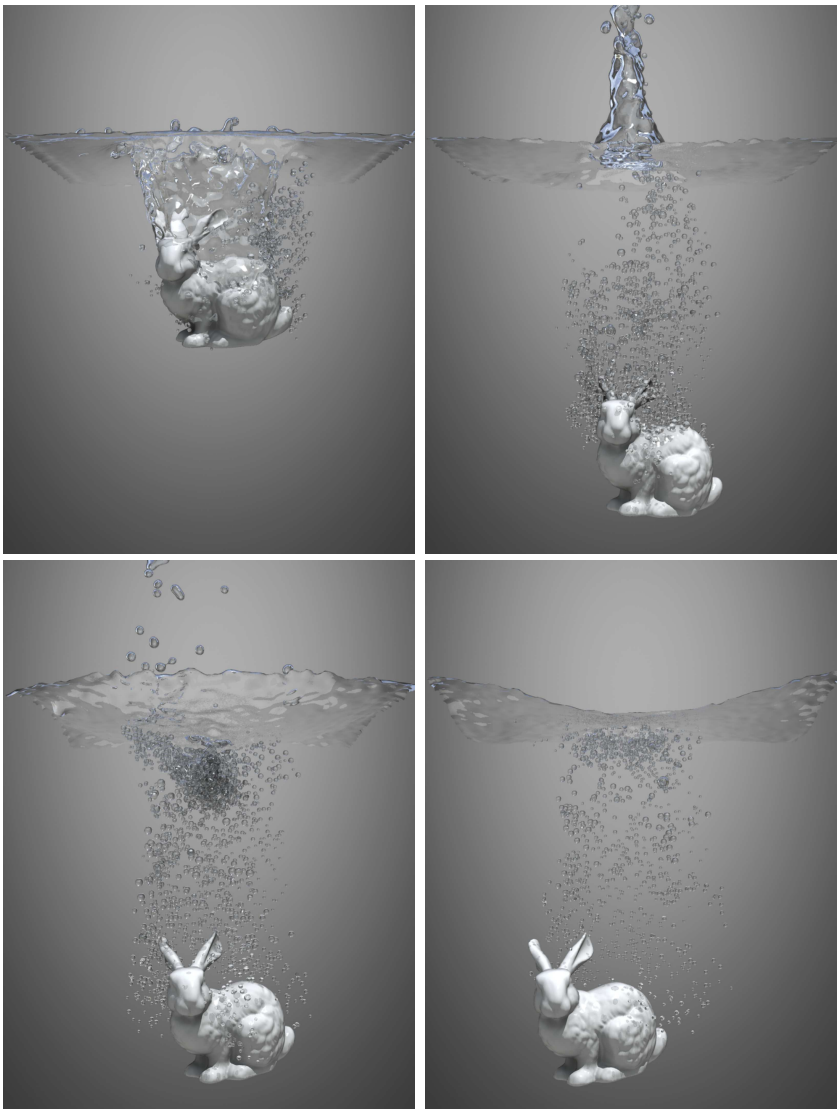


Figure 5: Diving bunny.

[Col03] A. Colagrossi and M. Landrini, Numerical simulation of interfacial flows by smoothed particle hydrodynamics, *J. Comput. Phys.* 192(2), 448-475, 2003.

[Gre04] S.T. Greenwood and D.H. House, Better with bubbles: Enhancing the visual realism of simulated fluid, In *Proceedings of SCA 04*, 287–296, 2004.

[Hon03] J.-M. Hong and C. Kim, Animation of bubbles in liquid. *Comput. Graph. Forum(Eurographics 03)*, 22(3), 253–262, 2003.

[Hon05] J.-M. Hong and C. Kim, Discontinuous fluids, *ACM Trans. Graph.(SIGGRAPH 05)*, 24(3), 915–920, 2005.

[Hon08] J.-M. Hong, H.-Y. Lee, J.-C. Yoon and C. H. Kim, Bubbles alive, *ACM Trans. on Graph.(SIGGRAPH 08)*, 27(3), 48:1–48:4, 2008.

[Kan00] M. Kang, R. Fedkiw and X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, *J. Sci. Comput.*, 15(3), 323–360, 2000.

[Kim10] D. Kim, O.-Y. Song and H.-S. Ko, A practical simulation of dispersed bubble flow, *ACM Trans. Graph.(SIGGRAPH 10)*, 29(4), 70:1–70:5, 2010.

[Mih09] V. Mihalef, D. Metaxas and M. Sussman, Simulation of two-phase flow with sub-scale droplet and bubble effects, *Comput. Graph. Forum(Eurographics 09)*, 28(2), 229–238, 2009.

[Mon94] J.J. Monaghan, Simulating free surface flows with sph, *J. Comput. Phys.*, 110(2), 399–406, 1994.

[Mon03] J.J. Monaghan, A. Kos and N. Issa, Fluid motion generated by impact, *J. Waterw., Port, Coastal, Ocean Eng.* 129(6), 250–259, 2003.

[Mue05] M. Müller, B. Solenthaler, R. Keiser and M. Gross, Particle-based fluid–fluid interaction, In *Proceedings of SCA 05*, 237–244, 2005.

[Oh09] S. Oh, Y. Kim and B.-S. Rho, Impulse-based rigid body interaction in SPH, *Comp. Anim. Virtual Worlds* 20(2-3), 215–224, 2009.

[Sol08] B. Solenthaler and R. Pajarola, Density contrast SPH interfaces, In *Proceedings of SCA 08*, 211–218, 2008.

[Son05] O.-Y. Song, H. Shin and H.-S. Ko, Stable but nondissipative water, *ACM Trans. Graph.*, 24(1), 81–97, 2005.

[Thu07] N. Thürey, F. Sadlo, S.Schirm, M. Müller-Fischer and M. Gross, Real-time simulations of bubbles and foam within a shallow water framework, In *Proceedings of SCA 07*, 191–198, 2007.

[Zhe06] W. Zheng, H.-J. Yong and J.-C. Paul, Simulation of bubbles, In *Proceedings of SCA 06*, 325–333, 2006.