International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.6 (2015), pp.219-232
http://dx.doi.org/10.14257/ijmue.2015.10.6.22

Realistic Simulation of Incompressible Fluids with Arbitrarily Shaped Boundaries

Xiao Nie [1,2], Leiting Chen [1,2], Chuan Zhou [1,2] and Hang Qiu [1,2]

[1] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2] *Provincial Key Laboratory of Digital Media, Chengdu 611731, China*
niexiao2008@gmail.com

Abstract

We present a particle-based method for realistically modeling the interaction of incompressible fluids with arbitrarily shaped solid boundaries. When dealing with static solids, previous meshless Lagrangian approaches either require uniform sampling of solid boundaries or suffer from unexpected side effects. Our method resolves these issues using a new boundary handling algorithm that represents the solid surface as a single layer of boundary particles with different influence factors. This technique can more efficiently handle arbitrarily shaped boundaries without the side effects of existing approaches, and produces more realistic simulation results. We further propose a new form of weighting kernel that corrects the fluid density estimations and avoids negative pressures at the free surface. We have demonstrated the robustness and effectiveness of our method with a range of scenarios.

Keywords: *physically-based fluid simulation; smoothed particle hydrodynamics; incompressibility; arbitrarily shaped boundaries*

1. Introduction

Physically based fluid simulation techniques have been extensively applied to generate realistic visual effects for movies and commercials. Among these methods, Smoothed Particle Hydrodynamics (SPH) is a popular particle approach. It can conveniently simulate fluid-solid coupling and capture small-scale phenomena. It has been used in a wide range of applications from computational fluid dynamics to real-time 3D games, but the visual artifacts induced by compressibility are a barrier to its use in the film industry. Simulations are more complicated for high-resolution fluid-solid interactions, as both incompressibility and non-penetration must be properly modeled to ensure high accuracy.

Although there are numerous computational models for this phenomenon in computational fluid dynamics, they are not suitable for the graphics community. In computer graphics, existing methods typically require an appropriate sampling quality at the boundaries. In other words, the fluid simulator generates erroneous results that affect the visual realism or lead to blowups, if the boundary sampling density is too high. Additionally, penetration occurs if it is too low. Therefore, we must develop an efficient method without this restriction. In this paper, we focused on efficient solutions to this problem in the context of SPH-based incompressible fluid simulations. Figure 1 illustrates our first example, in which the interaction of incompressible particle fluids and solid letters is modeled using our boundary handling algorithm.

ISSN: 1975-0080 IJMUE

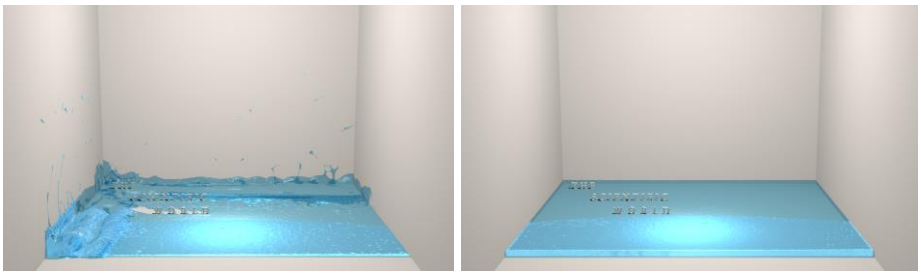

Figure 1. A Corner Breaking Dam (CBD) Flows around Solid Obstacles. The Water was Simulated Using 450k Fluid Particles, and the Solid Letters were Sampled Using 8k Boundary Particles

To further improve the realism of the simulation, we use a corrected kernel function that can prevent underestimations of the density evaluations of the fluid surface particles. Additionally, because CPUs now have an increasing number of cores, current serial algorithms must be modified to better exploit the power of state-of-the-art multi-core processors. To this end, we have rearranged our particle representations and spatial acceleration structures. In this way, we improved the cache-hit rate of our algorithm on multi-core CPU systems.

The contributions of our method are as follows.

(i) A new mass weighted function, which resolves the density estimation issues at the fluid-solid interface.

(ii) A new boundary force, which satisfies the non-penetration condition without consuming extra memory and computational resources.

(iii) An improved weighting kernel that avoids negative pressures at the fluid free surface.

The remainder of this paper is organized as follows. Section 2 discusses previous works, and Section. 3 contains a brief review of the predictive-corrective incompressible SPH (PCISPH) framework. Our arbitrarily shaped boundary handling method is introduced in Section. 4, followed by the implementation details and experimental results in Section. 5., Section 6 presents our conclusions and describes future work.

2. Related Work

SPH is a mesh-free Lagrangian fluid simulation method, which was initially proposed to solve astrophysical problems by Lucy [1] and Gingold and Monaghan [2]. It was first introduced to computer graphics to simulate highly deformable bodies by Desbrun and Gascue [3], and then used for interactive fluid simulation by Müller *et al*. [4]. This technique has been extensively applied over the last decade [5-10].

SPH was originally designed for compressible fluids, in which the pressure is evaluated from the density using the equation of state (EOS) [4]. This typically results in unphysical highly compressible fluids. To avoid this artifact, incompressible SPH (ISPH) [11-14] was proposed, to achieve incompressibility by solving a pressure Poisson equation (PPE) instead of the state equations. Although ISPH allows for relatively large time steps, it consumes much more computational resources per step. An alternative approach is weakly compressible SPH (WCSPH) [15-17], which enforces incompressibility using a stiff EOS. Compared with ISPH, WCSPH reduces the computational cost per physics step. However, it imposes a severe time step restriction. In general, the overall computational cost of both ISPH and WCSPH are too large for high-resolution fluid simulations, making them infeasible for practical applications.

Recently, Solenthaler and Pajarola [18] presented the PCISPH method. It combines the advantages of ISPH and WCSPH, namely the large time steps and low computational overhead per step. In this approach, the pressure force is used as a constraint, which resolves the compression induced by other forces. The pressure values are calculated by iteratively applying a predictor-corrector scheme to all particles. The iterations stop when the density fluctuation is below a user-defined value. Consequently, an incompressible state can be achieved at the next time step of the simulation. Another iterative EOS-based approach is local Poisson SPH (LPSPH), proposed by He *et al.* [19]. In this method, the pressure Poisson equation is solved using a new integral method. Both PCISPH and LPSPH can handle much larger time steps when compared with WCSPH, which results in an overall performance improvement.

Interesting fluid flow behavior generally emerges when arbitrarily shaped boundaries are added to the simulation. This challenges the SPH-based methods, especially when considering incompressible fluids. One challenge is that particle deficiency problems at the fluid-solid interface must be appropriately handled. Another is that non-penetration and incompressibility must be guaranteed. Penalty-based methods [15,20,21] are popular in the SPH literature. These techniques have been applied to model the interactions between compressible SPH fluids and solids. However, large penalty forces must be used. This adds stiffness to the equations, and results in large pressure variations that restrict applications to incompressible fluid simulations. Additionally, the sticking problem occurs because of a lack of neighbors near the boundary. The wall weight function [22] and ghost particle techniques [23-26] have been introduced to overcome the issues of penalty-based approaches. In the wall weight function method, the weighted contribution of the boundary is precomputed, and only relies on the distance between the fluid particle and the boundary. Ghost particles are mirrors of the fluid particles near the boundaries, and have the same physical quantities as their fluid counterparts except the normal component of the velocity. Although these methods avoid sticking artifacts, they are limited to simple boundaries. Frozen particle methods [27,28] are typically used to resolve this problem. The frozen particles are viewed as fluid particles that do not require position updates.

To obtain more control over the boundary condition, Becker *et al.* [17] presented a direct forcing technique. It incorporates control forces to obtain specific relative velocities at the boundaries. This method is much more efficient than penalty-based approaches, because larger time steps can be used. It can also simulate different slip conditions without penetration, because of the prediction-correction-based control forces. Ihmsen *et al.* [29] addressed arbitrarily shaped boundary handling in incompressible fluid simulations with the PCISPH algorithm. This approach combines the advantages of the frozen particle and direct forcing techniques. It can generate smooth density and pressure distributions and allows for larger time steps.

3. Background

SPH is a Lagrangian method that approximates fluid quantities at an arbitrary position using a bunch of discrete particles. That is,

$$A(\mathbf{x}) = \sum_j \frac{m_j}{\rho_j} A_j W(\mathbf{x}_{ij}, h), \qquad (1)$$

where $A(\mathbf{x})$ is the property at location x. m_j, ρ_j, A_j, x_j are the mass, density, physical property, and position of particle j in the support radius of position x, respectively. $W(\mathbf{x}_{ij}, h)$ is the smoothing kernel, $\mathbf{x}_{ij}$ is the distance between particle i

and j, and h is the support radius. Applying Eq. (1) to the fluid density, we can obtain

$$\rho_i = \sum_j m_j W(\mathbf{x}_{ij}, h). \quad (2)$$

Accordingly, we can deduce the pressure and viscosity forces of each particle using

$$\mathbf{F}_i^{pressure} = -\sum_j \frac{m_i}{\rho_i}\frac{m_j}{\rho_j}\frac{p_i + p_j}{2}\nabla W(\mathbf{x}_{ij}, h), \quad (3)$$

and

$$\mathbf{F}_i^{viscosity} = \mu\sum_j \frac{m_j}{\rho_j}(\mathbf{v}_j - \mathbf{v}_i)\nabla^2 W(\mathbf{x}_{ij}, h). \quad (4)$$

In standard SPH, an EOS is used to calculate the pressure, which inevitably causes compressible artifacts that are challenging to hide and produce unrealistic fluid animations. The PCISPH approach [18] has been introduced to avoid this issue. In PCISPH, a prediction-correction scheme is used to iteratively update the fluid pressures that satisfy incompressibility. Specifically, it first predicts the density fluctuation of each particle. Then, it calculates the pressures and pressure forces to reduce the density fluctuation until it is under a user-defined threshold. After that, it performs time integration to advance the positions and velocities of all particles.

4. Our Boundary Handling Method

To satisfy non-penetration conditions, existing particle-based boundary handling methods require either a multi-layer sampling of the solid surface or a positional correction of the fluid particles. This increases the computational overheads and memory footprint. Furthermore, non-uniformly sampled boundary particles at complex solid surfaces result in erroneous fluid density evaluations. This leads to incorrect simulations or stability issues, because of the inaccurate pressure forces. In this section, we present a new boundary handling scheme based on PCISPH, which solves these problems.

4.1. Mass Weighted Function

In PCISPH, we must predict the density fluctuation of each particle at the beginning of each iteration. According to Eq.(2), the neighboring particles should be updated according to their predicted positions. However, their neighbors at the current time step are often used instead, for efficiency reasons. This approximation works fine in the case of density overestimation, but may result in a premature exit from the correction loop when the neighboring particles are not sufficient. These situations may be handled by recalculating the neighborhoods or dramatically lowering the time steps, but this significantly reduces the simulation efficiency.

To address this problem, existing methods typically sample the solid surface with boundary particles, to reduce the density evaluation errors at the fluid-solid interface. We have followed [29,30] in generating a single layer of boundary particles for solid surfaces using the signed distance field (see Figure. 2). For simply shaped solid surfaces, the method can obtain a uniform particle sampling. However, for complex solid surfaces (especially geometrically complex regions), it can lead to irregular particle sampling (see Figure. 3), which induces penetration or stability issues. In other words, at the solid surfaces with small boundary particle density, the boundary forces are not large enough to prevent the fluid particles from penetrating through solid boundaries, whereas at the solid surfaces with a large

number of boundary particles, the fluid density and pressure can be overestimated, which results in unstable simulations (see Figure. 4).

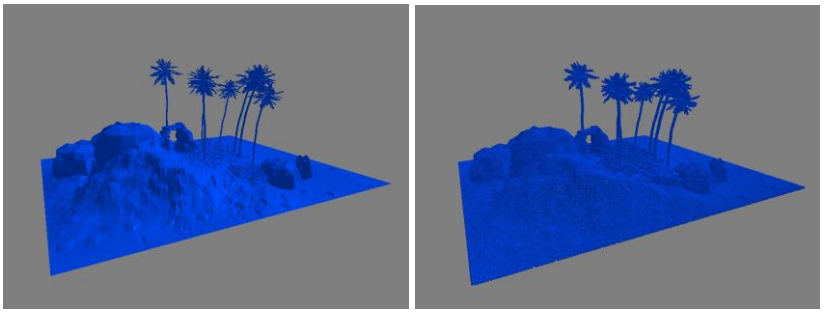

Figure 2. Different Representations of the Solid Island; Left: Triangle Meshes. Right: Boundary Particles

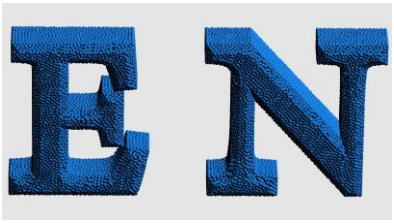

Figure 3. Non-Uniform Sampling of Boundary Particles at the Geometrically Complex Solid Surfaces

To solve this problem, we used an adaptive sampling idea for solid-fluid interactions. Specifically, at the densely sampled solid surfaces, we use boundary particles with small mass to calculate the densities of their neighboring fluid particles. At the sparsely sampled regions, we use boundary particles with large mass to perform the density evaluation (see Figure. 5). To this end, we introduce a mass weighted function based on the local density of the boundary particles. The local number density is

$$\rho_n(\mathbf{b_i}) = \sum_{b_j} W(\mathbf{x_{b_i}} - \mathbf{x_{b_j}}, h), \qquad (5)$$

where b_i is the solid boundary particle with position x_{b_i}; and b_j is its neighboring boundary particle with position x_{b_j}. In this paper, we assume that the incompressible fluid has a rest density close to ρ_0. Therefore, the mass weighted function is

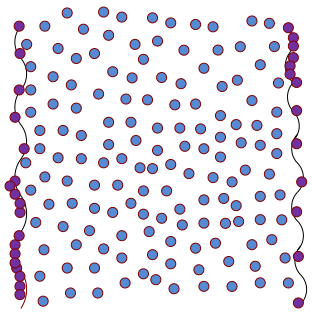

Figure 4. Solid Surfaces Irregularly Sampled with Boundary Particles (Purple). When the Boundary Particles are Dense, the Fluid Particles (Blue) around them have Larger Contributions, which can cause

Stability Issues. However, when the Boundary Particles are Sparsely Distributed, the Generated Boundary Forces may not be Large Enough to Prevent Penetration.

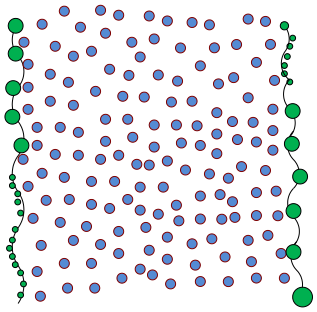

Figure 5. Adaptive Boundary Particle Sampling; The Weighted Mass of the Boundary Particle (Green) is Proportional to its Radius. Thus, the Fluid Particles (Blue) Receive Different Contributions from their Boundary Neighbors

$$m_{weighted}(\mathbf{b_i}) = \frac{\rho_0}{\rho_n(b_i)}, \tag{6}$$

In Eq.(6), the mass weighted function of the boundary particle is inversely proportional to its local number density. This mathematically reflects our adaptive boundary particle sampling idea: we shorten the radius of the densely distributed boundary particles because of their high local number density (which reduces the impact on the neighboring fluid particles), and increase the radius of sparsely distributed boundary particles because of their low local number density (which increases their impact on the neighboring fluid particles).

According to this mass weighted function, the fluid density calculation can be modified to

$$\rho_i = \sum_j m_j W(\mathbf{x}_{ij}, h) + \sum_k m_{weighted}(\mathbf{b_k}) W(\mathbf{x}_{\mathbf{b_i b_k}}, h). \tag{7}$$

There are two benefits of using Eq. (7) for the density evaluation. On one hand, the densities of fluid particles near the solid boundaries will not be too large, so the large pressures and pressure forces that usually result in incorrect results or simulation blowups can be avoided. On the other hand, the iteration loop of the PCISPH algorithm will not be prematurely terminated because of density underestimation, and therefore the compressible artifacts can be avoided.

4.2. Boundary Force Calculation

Similar to [18], we assumed that the pressures of neighboring particles are equal, and that the densities can be approximated using the fluid rest density because we focused on incompressible fluid simulations. Therefore, the pressure force calculation in Eq. (3) can be modified to

$$\begin{aligned}\mathbf{F}_i^{corr\_pressure} &= -\sum_j \frac{m_i}{\rho_i}\frac{m_j}{\rho_j}\frac{p_i+p_j}{2}\nabla W(\mathbf{x}_{ij},h) \\ &= -\sum_j \frac{m_i}{\rho_0}\frac{m_j}{\rho_0}\frac{p_i+p_i}{2}\nabla W(\mathbf{x}_{ij},h) \\ &= -\sum_j m_i m_j \frac{p_i}{\rho_0^{\,2}}\nabla W(\mathbf{x}_{ij},h).\end{aligned} \tag{8}$$

Combined with the mass weighted function, $m_{weighted}(\mathbf{b_i})$ (introduced in the last section), the boundary force from boundary particles can be represented as

$$\mathbf{F}_i^{boundary} = -\sum_k m_i m_{weighted}(\mathbf{b_k})\frac{p_i}{\rho_0^{\,2}}\nabla W(\mathbf{x}_{ik},h)\,. \tag{9}$$

4.3. Negative Pressures and Corrected Kernel

Particles at the fluid free surface do not have enough neighbors, so the evaluated density will be underestimated. Therefore, negative pressures can occur and the iteration loop of PCISPH terminates too early. We address these problems by modifying the corrected SPH kernel [31]:

$$\overline{W_i}(x) = W_i(x)\alpha(x)\,, \tag{10}$$

where

$$\alpha(x) = \frac{1}{\sum_i \frac{m_i}{\rho_i}W_i(x)}, \tag{11}$$

with evaluation position x, weighting kernel $W_i(x)$, mass m_i and density ρ_i at neighboring particle i.

This corrected kernel representation can avoid inaccurate densities and negative pressures at the free surface. Because we have focused on a uniform particle radius sampling for incompressible fluids, the mass and density of each fluid particle can be treated as constant. Therefore, the mass and density terms in Eq. (11) are constant during the simulation and can be precomputed. This avoids another loop over the particles.

Figure 6 shows the results without and with this model. We can see that the particle clumping caused by negative pressures can be eliminated using our modified kernel.

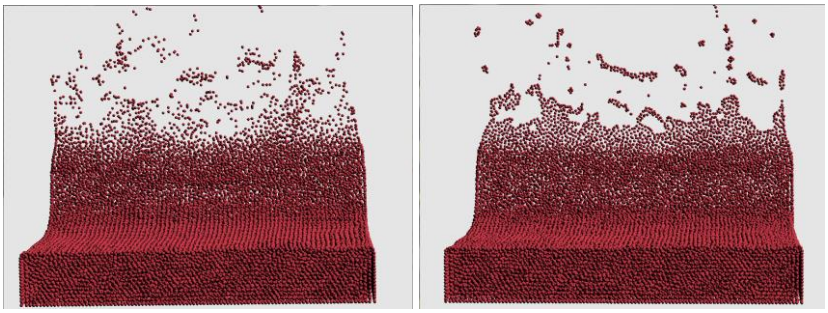

Figure 6. Our Weighting Kernel Model. Left: Without our Model, Particle Clumping Occurs because of the Negative Pressures. Right: with our Model, the Clumping Issue is Avoided

Algorithm 1 Our arbitrarily shaped boundary handling

```
while simulating do
    for all particles do
        perform neighbor searching
    for all boundary particles do
        calculate weighted mass
    for all fluid particles do
        calculate external force
        initialize pressure
        initialize pressure force
    while (density variations are larger than a user-defined threshold) do
        for all fluid particles do
            predict velocity
            predict position
        for all fluid particles do
            predict density
            predict density variation
            update pressure
        for all fluid particles do
            calculate pressure force
            calculate boundary force
    for all fluid particles do
        update velocity
        update position
```

4.4. The Proposed Boundary Handling Algorithm

In this Section, we describe our arbitrarily shaped boundary handling method that is based on PCISPH. We observed that there is some similarity between the pressure force and boundary force in incompressible fluids. That is, the pressure force keeps the distance between neighboring fluid particles above a threshold to avoid compressible artifacts, while the boundary force prevents fluid particles from getting too close to boundary particles, which can lead to penetration. In essence, these two forces are used to try to prevent neighboring particles from getting too close. Based on this observation, we integrated our solid boundary force into the prediction-correction loop of PCISPH (see Alg. 1). In contrast to [18] and [29], we moved the boundary force calculation from the external forces evaluation step to inside the prediction-correction loop, using the rest density of incompressible fluids to predict the boundary force (see Eq.(9)). Then, the boundary force and the adapted inner fluid pressure force (see Eq.(8)) can be predicted and corrected to guarantee incompressibility and non-penetration. Additionally, the benefit of the large time steps in PCISPH is preserved, because our technique can lead to smooth distributions of the physical properties. This makes the proposed method as efficient as PCISPH.

5. Implementation and Results

We implemented the boundary scheme presented by Ihmsen *et al*. [29], the current state-of-the-art PCISPH-based boundary handling technique, and our method to compare the simulation results and performance.

5.1. Implementation

We applied the corrected kernel formulation (see Eq.(10)) to all our equations, using the same cubic spline kernels, $W_i(x)$ [15,18,29]. For the neighborhood search, we used parallel z-index sort [6] together with a uniform grid, to increase the cache-hit rate on multi-core CPUs. For the same purpose, we also applied a cache-friendly data structure to represent the particle data. That is, we used a structure of arrays instead of an array of structures. We used the approach proposed by Solenthaler *et al.* [28] for fluid surface reconstruction. Our code uses OpenMP for parallelization.

In our proposed method, complex solid surfaces such as letters, the armadillo, and the island were sampled by boundary particles, similar to [17]. We used the weighting function approach [22] for geometrically simple regions such as the container, to reduce the computational overhead. We used the surface tension force model from [16], and the leap-frog time integration method.

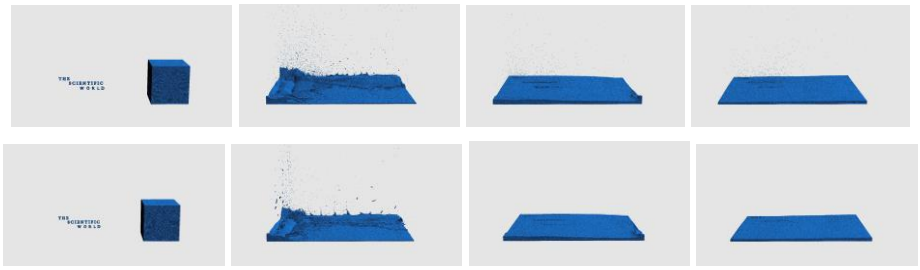

Figure 7. CBD Scene. Top Row: Using Ihmsen *et al.* [29], Some Fluid Particles Unphysically Fly Around After Colliding with the Irregularly Sampled Obstacles, and Do Not Stop. Bottom Row: Using Our Method, All Fluid Particles Rapidly Come to Rest, as Expected.

For all scenarios, we set the fluid particle radius to $r = 0.0125m$, the rest distance between initial particles to $2r$, the support radius to $h = 4r$, the particle mass to $0.002kg$, the fluid rest density to $1000kg/m^3$, and the density fluctuation threshold to 1%.

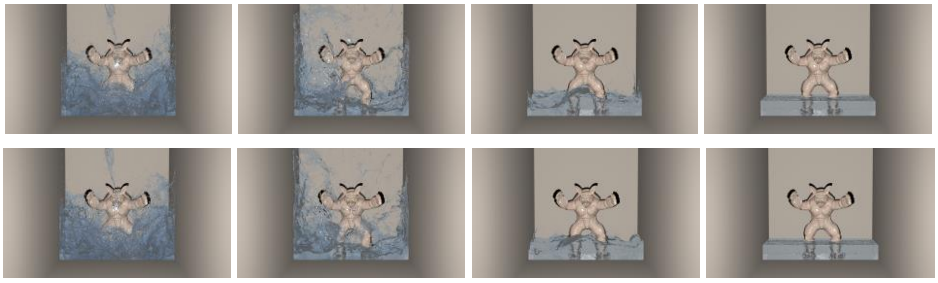

Figure 8. Shock Scene. Fast-Moving Water Drops onto the Solid Armadillo Sampled by Boundary Particles. Top Row: Ihmsen *et al.* [29]. Bottom Row: our Method.

5.2. Results

We implemented the proposed method in C++ on a quad-core Intel Core i7-3770 3.4 GHz CPU with 8 GB memory. All the animations in our experiments were rendered offline with POVRAY. Note that we used constant time steps in all of our simulations, for both Ihmsen *et al.* [29] and our approach.

To show that our approach can handle realistic interactions of incompressible fluids and irregularly sampled arbitrarily shaped boundaries, we simulated a scenario with water flowing around solid letters (see Figure. 7). As shown, the fluid particles unphysically fly around after hitting the solids, using the method in [29] (see Figure. 7, top), whereas they eventually calm down (as expected) using our method (see Figure. 7, bottom). Figure 8 demonstrates a fast-moving fluid falling onto an armadillo, which illustrates the stability of our method under huge impacts. As in [29], large time steps can be used because our approach can also guarantee smooth distributions for the density and pressure.

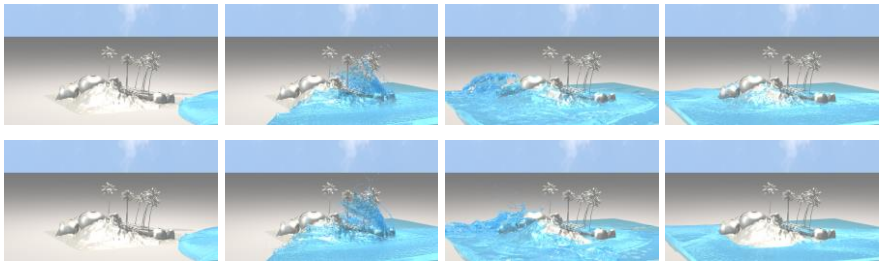

Figure 9. Island Scene. The Island has 155k Boundary Particles, and the Flood is Sampled with 1 Million Fluid Particles. Top Row: Ihmsen *et al.* [29]. Bottom Row: our Method.

Figure 9 shows a more complex flooding island example. Note that, using [29], the fluid adheres to the solid surface even without specifying any external adhesion forces (see Figure. 10, left). However, this does not occur in our approach (see Figure. 10, right). This is because [29] uses a direct forcing-based technique, which constrains the relative velocity and position of the fluid particle at the fluid-solid interface. This results in unphysical behaviors at arbitrarily shaped boundary surfaces. In our approach, the issue of unphysical adhering to solid boundaries is avoided, because we do not apply any artificially imposed interference to the fluid particles at the boundary surfaces. Moreover, we can easily integrate fluid-solid adhesion techniques such as Akinci *et al.* [32] into our framework. This allows for more control, resulting in more physically plausible results.

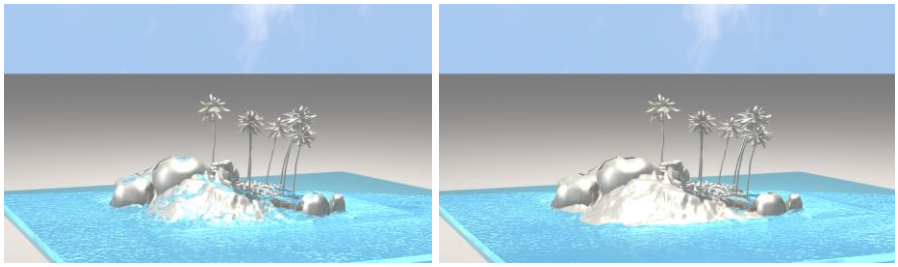

Figure 10. Visual Comparison Between [29] and our Method. Left: Adhesion Artifact Occurs Using Ihmsen *et al.* [29]. Right: the Same Artifact is Avoided in our Method

Table 1 shows the scene settings and timings of different scenarios for the performance comparisons, including the number of particles and the average computational time for each physical simulation update of both approaches. We demonstrated the performance gain of our method when compared with [29] using a single-threaded implementation, where we averaged the performance over 30k time

steps. We can see that our method is more efficient than [29]. This is because our method does not require particle position corrections, and therefore avoids any additional computational overheads.

Table 1. Comparison of Average Performances of Ihmsen *et al.* [29] and our Method, for PCISPH-Based Incompressible Fluid Scenarios with Various Particle Counts; All Simulation Times are Given in Seconds per Time Step

Scene	Time step	# boundary particles	# fluid particles	single-threaded	
				Ihmsen et al. [29]	Our method
CBD	0.0012s	8K	450K	8.48	7.52
Shock	0.0005s	44K	750K	10.97	9.91
Island	0.0008s	155K	1M	19.43	18.25

Table 2 compares the performances of the multi-threaded implementations of both methods and their single-threaded counterparts for different scenes. We can see that the speedups of the parallelized versions of our method when compared with the single-threaded version were 2.24 for the CBD scene, 2.38 for shock, and 2.73 for island. For the parallelized version of Ihmsen *et al.* [29], the speedups were 2.17 for the CBD scene, 2.25 for shock, and 2.59 for island. This table illustrates that our algorithm is very suitable for multi-core architectures.

Table 2. Simulation Speedups of Both Methods Using our Parallel Techniques on a Multi-Core CPU System

Scene	Ihmsen et al. [29]			Our method		
	single-threaded	multi-threaded	Speedup	single-threaded	multi-threaded	Speedup
CBD	8.48	3.92	2.17	7.52	3.35	2.24
Shock	10.97	4.87	2.25	9.91	4.16	2.38
Island	19.43	7.51	2.59	18.25	6.68	2.73

6. Conclusion and Future Work

We have demonstrated a novel boundary condition processing method that can handle arbitrarily shaped boundaries in PCISPH-based fluid simulations. By adaptively adjusting the radius of solid boundary particles at the complex solid surfaces, we can remove evaluation errors of fluid physical properties. We have introduced a new boundary particle mass weighted function, on which the boundary force is based. This results in loose requirements for the boundary sampling quality.

Furthermore, we proposed a corrected kernel to evaluate the density at the fluid free surface, to solve the negative pressure issue. This can generate more realistic visual effects. Moreover, we implemented our algorithm using Open MP, taking full advantage of the computational power of parallel architectures.

Our algorithm could be implemented on state-of-the-art GPUs to accelerate the simulation. In addition, our method could be combined with advanced particle-based

surface tension and adhesion techniques to produce more realistic solid-fluid interactions.

Acknowledgements

The authors would like to thank Prashant Goswami, Fatih Erol and Maxim Makhinya for many insightful discussions. We also thank the anonymous reviewers for their valuable comments that helped to improve the manuscript. This project is supported by National High Technology Research and Development Program of China (No. 2012AA011503), Project on the Integration of Industry, Education and Research of Guangdong Province (No. 2012B090600008) and the pre-research project (No. 51306050102). The armadillo model is courtesy of the Stanford Computer Graphics Lab. The island model is courtesy of www.thefree3dmodels.com.

References

[1] L. B. Lucy, "A numerical approach to the testing of the fission hypothesis," The Astronomical Journal, vol. 82, **(1977)**, pp. 1013-1024.
[2] J. J. Monaghan and R. A. Gingold, "Smoothed particle hydrodynamics-theory and application to nonspherical starts," Royal Astronomical Society, Monthly Notices, vol. 181, **(1977)**, pp. 375-389.
[3] M. Desbrun and M. P. Gascuel, "Smoothed particles: A new paradigm for animating highly deformable bodies," In Eurographics Workshop on Computer Animation and Simulation (EGCAS), **(1996)**, pp. 61–76.
[4] M. Muller, D. Charypar and M. Gross, "Particle-based fluid simulation for interactive applications," In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, **(2003)**, pp. 154-159.
[5] P. Goswami, P. Schlegel, B. Solenthaler, and R. Pajarola, "Interactive SPH simulation and rendering on the GPU," In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, **(2010)**, pp. 55-64.
[6] M. Ihmsen, N. Akinci, M. Becker, and M. Teschner, "A Parallel SPH Implementation on Multi-Core CPUs," Computer Graphics Forum, vol. 30 no. 1, **(2011)**, pp. 99-112.
[7] P. Goswami and R. Pajarola, "Time adaptive approximate SPH," In Proceedings Eurographics Workshop on Virtual Reality Interaction and Physical Simulation, **(2011)**, pp. 19-28.
[8] K. Raveendran, C. Wojtan, and G. Turk, "Hybrid smoothed particle hydrodynamics," In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, **(2011)**, pp. 33-42.
[9] N. Akinci and B. Solenthaler, "Versatile Rigid-Fluid Coupling for Incompressible SPH," ACM Transactions on Graphics (Proceedings SIGGRAPH), vol. 30 no. 4, **(2012)**, pp. 72:1–72:8.
[10] P. Goswami and C. Batty, "Regional Time Stepping for SPH," Eurographics 2014, **(2014)**, pp. 45-48.
[11] S. J. Cummins and M. Rudman, "An SPH projection method," Journal of computational physics, vol. 152, **(1999)**, pp. 584-607.
[12] S. Shao, "Incompressible SPH simulation of wave breaking and overtopping with turbulence modeling," International Journal for Numerical Methods in Fluids, vol. 50, **(2006)**, pp. 597-621.
[13] X. Y. Hu and N. A. Adams, "An incompressible multi-phase SPH method," Journal of Computational Physics, vol. 227, **(2007)**, pp. 264-278.
[14] F. Losasso, J. O. Talton, N. Kwatra and R. Fedkiw, "Two-way coupled SPH and particle level set fluid simulation," IEEE Transactions on Visualization and Computer Graphics, **(2008)**, pp. 797-804.
[15] J. J. Monaghan, "Smoothed particle hydrodynamics," Reports on Progress in Physics, vol. 68, **(2005)**.
[16] M. Becker and M. Teschner, "Weakly compressible SPH for free surface flows," In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, **(2007)**, pp. 209-217.
[17] M. Becker, H. Tessendorf and M. Teschner, "Direct forcing for lagrangian rigid-fluid coupling," IEEE Transactions on Visualization and Computer Graphics, vol. 15, **(2008)**, pp. 493-503.
[18] B. Solenthaler and R. Pajarola, "Predictive-corrective incompressible SPH," ACM Transactions on Graphics (Proceedings SIGGRAPH), vol. 28, **(2009)**, pp. 40:1–40:6.
[19] X. He, N. Liu, S. Li, H. Wang, and G. Wang, "Local Poisson SPH for Viscous Incompressible Fluids," Computer Graphics Forum, **(2012)**, pp. 1948--1958.
[20] J. J. Monaghan and J. B. Kajtar, "SPH particle boundary forces for arbitrary boundaries," Computer Physics Communications, vol. 180, **(2009)**, pp. 1811-1820.
[21] J. J. Monaghan, "Simulating free surface flows with SPH," Journal of Computational Physics, vol. 110, **(1994)**, pp. 399-399.
[22] T. Harada, S. Koshizuka and Y. Kawaguchi, "Smoothed particle hydrodynamics on gpus," In Proceedings of Computer Graphics International, **(2007)**, pp. 63–70.
[23] S. Falappi and M. Gallati, "SPH simulation of water waves generated by granular landslides," In Proceedings of 32nd Congress of IAHR (International Association of Hydraulic Engineering &

Research, **(2007)**, pp. 106.
[24] H. Schechter and R. Bridson, "Ghost sph for animating water," ACM Transactions on Graphics (Proceedings SIGGRAPH), vol. 31, no. 4, **(2012)**, pp. 61:1–61:8.
[25] X. Y. Hu and N. A. Adams, "A multi-phase SPH method for macroscopic and mesoscopic flows," Journal of Computational Physics, vol. 213, **(2006)**, pp. 844--861.
[26] J. P. Morris and J. J. Monaghan, "A Switch to Reduce SPH Viscosity," Journal of Computational Physics, vol. 136, **(1997)**, pp. 41-50.
[27] R. Keiser, B. Adams, P. Dutré, L. J. Guibas and M. Pauly, "Multiresolution particle-based fluids," Technical report, ETH Zurich, **(2006)**.
[28] B. Solenthaler, J. Schlafli and R. Pajarola, "A unified particle model for fluid-solid interactions," Computer Animation and Virtual Worlds, vol. 18, **(2007)**, pp. 69.
[29] M. Ihmsen, N. Akinci, M. Gissler and M. Teschner, "Boundary Handling and Adaptive Time-stepping for PCISPH," VRIPHYS, **(2010)**, pp. 79-88.
[30] N. Bell, Y. Yu and P. J. Mucha, "Particle-based simulation of granular materials," In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, **(2005)**, pp. 77-86.
[31] J. Bonet and S. Kulasegaram, "A simplified approach to enhance the performance of smooth particle hydrodynamics methods," Applied Mathematics and Computation, vol. 126, **(2002)**, pp. 133-155.
[32] N. Akinci, G. Akinci and M. Teschner, "Versatile Surface Tension and Adhesion for SPH Fluids," ACM Transactions on Graphics (Proceedings SIGGRAPH Asia), November **(2013)**.

