Received: 3 August 2022 | Revised: 11 December 2022 | Accepted: 13 December 2022

DOI: 10.1002/cav.2136

SPECIAL ISSUE PAPER

WILEY

Spatial adaptivity with boundary refinement for smoothed particle hydrodynamics fluid simulation

Yanrui Xu[1] | **Chongming Song**[1] | **Xiaokun Wang**[1,2,3] | **Xiaojuan Ban**[1,4,5] | **Jiamin Wang**[1] | **Yalan Zhang**[1] | **Jian Chang**[3]

[1]School of Intelligence Science and Technology, University of Science and Technology Beijing, Beijing, China

[2]Beijing Key Laboratory of Knowledge Engineering for Materials Science, University of Science and Technology Beijing, Beijing, China

[3]National Centre for Computer Animation, Bournemouth University, Poole, UK

[4]Beijing Advanced Innovation Center for Materials Genome Engineering, University of Science and Technology Beijing, Beijing, China

[5]Key Laboratory of Perception and Control of Intelligent Bionic Unmanned Systems, Ministry of Education, University of Science and Technology Beijing, Beijing, China

Correspondence
Xiaokun Wang and Xiaojuan Ban, School of Intelligence Science and Technology, University of Science and Technology Beijing, Beijing, China.
Email: wangxiaokun@ustb.edu.cn and banxj@ustb.edu.cn

Funding information
Horizon 2020-Marie Skłodowska-Curie Action-Individual Fellowships, Grant/Award Number: 895941; National Natural Science Foundation of China, Grant/Award Number: 61873299; Key Research and Development Project of Hainan Province, Grant/Award Number: ZDYF2020031; Fundamental Research Funds for the Central Universities, Grant/Award Number: QNXM20220043

Abstract

Fluid simulation is well-known for being visually stunning while computationally expensive. Spatial adaptivity can effectively ease the computational cost by discretizing the simulation space with varying resolutions. Adaptive methods nowadays mainly focus on the mechanism of refining the fluid surfaces to obtain more vivid splashes and wave effects. But such techniques hinder further performance gain under the condition where most of the vast fluid surface is tranquil. Moreover, energetic flow beneath the surface cannot be adequately captured with the interior of the fluid still being simulated under coarse discretization. This article proposes a novel boundary-distance based adaptive method for smoothed particle hydrodynamics fluid simulation. The signed-distance field constructed with respect to the coupling boundary is introduced to determine particle resolution in different spatial positions. The resolution is maximal within a specific distance to the boundary and decreases smoothly as the distance increases until a threshold is reached. The sizes of the particles are then adjusted towards the resolution via splitting and merging. Additionally, a wake flow preservation mechanism is introduced to keep the particle resolution at a high level for a period of time after a particle flows through the boundary object to prevent the loss of flow details. Experiments show that our method can refine fluid–solid coupling details more efficiently and effectively capture dynamic effects beneath the surface.

KEYWORDS

boundary handling, computer animation, fluid simulation, spatial adaptivity

1 | INTRODUCTION

Fluid simulation plays an essential role in visual effects, for the physical-based result can produce magnificent eye-catching effects to maximize the sense of reality. Nevertheless, higher demand for simulation details usually means the rapid growth of computational cost due to the need for the finer discretization of multidimensional space and time. Though the choice of time step (*temporal adaptivity*) can be readily adapted using Courant–Friedrichs–Lewy (CFL) condition, the mechanism of *spatial adaptivity* can still be further exploited considering variable conditions for better efficiency. The spatial adaptivity mechanism enables achieving exquisite fluid behaviors with more affordable expenses by optimizing the interval of sampling points to refine local areas with pertinence.

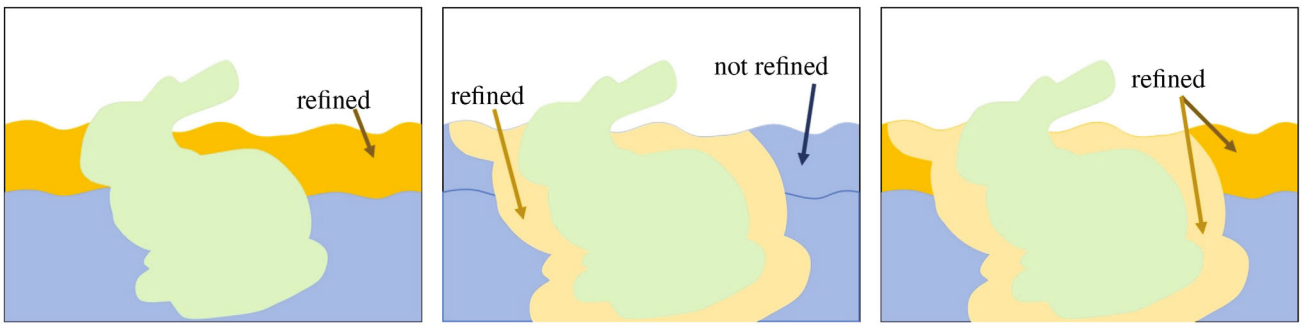
Nowadays, research on spatial adaptivity for Lagrangian fluid simulation mainly targets surface-based optimization, aiming at refining the surface details like splashes, waves and thin films to enrich the visual effects. Horvath and Solenthaler[1] use a higher resolution for particles within a certain distance to the free surface of the fluid; Winchenbach et al.[2] make the desired size of particles increase smoothly with the distance to the surface.

However, the surface-based strategy has the disadvantage of an inflexible intervention mechanism, which causes low computational efficiency and the failure to enhance specific dynamic areas. On the wide fluid surface, the area with the most high-frequency details is often the part where the fluid is coupled with other objects, such as a boat sailing across the water surface, arousing the waves around and behind the body. Meanwhile, the boat's rotating propeller below the surface can constantly transmit kinetic energy into the water, further producing splashes to the surface. When such a scenario is simulated using coarse discretization with surface-based refinement, the vast and calm surface brings a large computational burden, causing computational resources to be wasted in these unimportant areas. Also, the energy dissipation from the propeller area cannot be restrained effectively.

To conduct spatial adaptivity with more flexibility and efficiency, we propose a novel adaptivity method with boundary refinement for smoothed particle hydrodynamics (SPH) fluid simulation, where particle resolution increases as the distance to the interested boundary object decreases (see Figure 1). The sizes of the particles are adjusted towards the resolution via splitting and merging. Moreover, a wake flow preservation mechanism is introduced to retain the refined resolution for a specific period after a particle flows through the boundary object to prevent the loss of flow details. The results show that our method can enrich fluid details more economically and produce more accurate visual effects in contrast to the surface-based strategy.

2 | RELATED WORK

Methods for spatial adaptivity in fluid simulation depend heavily on the fluid simulation approaches on which they are based. For Eulerian approaches that simulate the fluid with grids, a standard scheme uses octrees to split some grids into finer grids.[3] Meanwhile, tilted grids can also be used[4] to avoid numerical issues. For particle-based Lagrangian approaches, the sizes of particles are adaptively adjusted.[2] For hybrid approaches that inherit the traits of both Eulerian



(a) Surface-based refinement (b) Our boundary refinement. (c) Ours + surface-based.

FIGURE 1 Contrary to the previous surface-based refinement method (a), our method (b) refines fluid particles near boundary objects to allow the fine-scale fluid-boundary coupling to enhance boundary coupling details. Our method can also be combined with the surface-based method to refine near the boundary and the surface (c).

and Lagrangian methods, one can improve the efficiency by limiting the regions where particles are used[5] or combining octrees with an adjustable number of particles based on regional characteristics.[6]

SPH[7] is a Lagrangian fluid simulation method that possesses the advantages of natural mass conservation and simplicity of advection computation. For adaptive SPH, Adams et al.[8] resampled particles with different radii based on the size of local geometric features. Solenthaler and Gross[9] introduced a two-scale mechanism that coupled a low-resolution simulation with a high-resolution one, and particles were directly inserted or deleted at the boundary between the two resolutions, which broke the mass-preserving condition. Horvath and Solenthaler[1] extended the previous work[9] to support multiple levels of resolution and conserve mass. Orthmann and Kolb[10] increased the spatial resolution by splitting particles using a 1:2 pattern and applied a temporal blending technique to achieve a smooth particle splitting process by maintaining continuous physical fields. Vacondio et al.[11] used the variational principle to search for an optimized splitting pattern that minimized density error. They also introduced a coalescing scheme to realize the dynamic reduction of resolution. Winchenbach et al.[2] achieved adaptivity in incompressible SPH by introducing a new split-merge mechanism that produced an approximately continuous resolution based on the distance from the free surface of the fluid. Winchenbach and Kolb[12] later proposed a method to optimize the density error from the splitting process into an arbitrary number of particles by optimizing the position and mass of the split particles to increase the stability of the previous method.[2]

A popular approach for boundary handling of SPH fluids is particle-based methods that represent the solid objects with particles.[13,14] However, for fluid simulation using adaptive methods, particle-based boundary representation may encounter the problem of size discrepancy between fluid and boundary particles. Another approach for boundary handling is the boundary integral method. Fujisawa et al.[15] used empirically derived functions to handle triangle-mesh boundaries. Koschier and Bender[16] precomputed the integral of boundary on a fixed grid, where the precomputing was rather expensive. Bender et al.[17] improved this method by precalculating a volume term. Chang et al.[18] integrated computer-aided design mesh file boundary representation into SPH by converting the volume integral to surface integral. Winchenbach et al.[19] proposed a boundary integral method that supports adaptive SPH, which approximated the boundary integral locally at each particle with a plane based on the signed-distance field (SDF) of the boundary object. Though the boundary handling for adaptive SPH has been well-studied, more attention needs to be paid to designing the adaptive mechanism dedicated to the fluid–solid boundary. So in this article, we present a spatial adaptivity method with the boundary refinement mechanism for SPH fluid simulation to capture detailed interaction effects.

3 | ADAPTIVE SPH

3.1 | SPH basics

To simulate fluid behavior, the SPH method[7] describes physical fluid motion according to the Navier–Stokes equation:

$$\rho\frac{D\mathbf{v}}{Dt} = -\nabla p + \rho\upsilon\nabla^2\mathbf{v} + \mathbf{f}_{\text{ext}}, \tag{1}$$

where the change rate of the velocity $\mathbf{v}$ is determined by pressure p, density ρ, dynamic viscosity υ and external body force $\mathbf{f}_{\text{ext}}$.

To perform numerical computation, SPH discretizes the fluid into particles and calculates the physical attributes of each particle using the attributes from neighboring particles and a kernel function W, as:

$$A_i = \sum_j A_j \frac{m_j}{\rho_j} W(\mathbf{x}_i - \mathbf{x}_j, h_{ij}), \tag{2}$$

where A is the physical attribute. i is the current particle. j denotes neighboring particles of i, whose distance to i is smaller than the support radius h_{ij}. $h_{ij} = (h_i + h_j)/2$, where $h_i = 2r_i$ and r is the particle diameter. m is mass. $\mathbf{x}$ is position. This article uses a cubic spline function for the kernel function W. Any value followed by the subscript i or j means the value of that particle.

According to Equation (2), density located at particle i can be approximated as:

$$\rho_i = \sum_j m_j W_{ij}, \tag{3}$$

where W_{ij} is short for $W(\mathbf{x}_i - \mathbf{x}_j, h_{ij})$.

Using the state equation from Becker and Teschner,[20] the pressure at the particle i can be calculated as:

$$p_i = B\left(\left(\frac{\rho_i}{\rho_0}\right)^{\gamma} - 1\right), \tag{4}$$

where $B = \rho_0 c_s^2/\gamma$. ρ_0 is the rest density denoting the density of the fluid without any compression. c_s is the speed of sound in the fluid, which is often set to 100 m/s. γ is set to $\gamma = 7$ for weakly compressible fluids.

The pressure force on the particle can then be computed as:

$$\mathbf{F}_i^p = -m_i \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2}\right) \nabla W_{ij}. \tag{5}$$

The viscosity force can be approximated as follows:

$$\mathbf{F}_i^{\text{vis}} = m_i \sum_j m_j \upsilon \left(\frac{\mathbf{v}_{ij} \cdot \mathbf{x}_{ij}}{|\mathbf{x}_{ij}|^2 + 0.01h_i^2}\right) \nabla W_{ij}, \tag{6}$$

where $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ is the relative velocity between i and j. $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$.

The total force on a particle can be expressed as:

$$\mathbf{F}_i = \mathbf{F}_i^p + \mathbf{F}_i^{\text{adv}} + \mathbf{F}_i^{\text{bound}}, \tag{7}$$

where $\mathbf{F}^{\text{adv}}$ is the advection force consisting of gravity $\mathbf{G}$, viscosity force $\mathbf{F}^{\text{vis}}$, and surface tension.[21] $\mathbf{F}^{\text{bound}}$ is the force from the boundary to the fluid, which is explained in Section 4. The velocity and position of the particle are updated each time step with $\mathbf{v}_i := \mathbf{v}_i + \Delta t \mathbf{F}_i/m_i$, $\mathbf{x}_i := \mathbf{x}_i + \Delta t \mathbf{v}_i$, where Δt is the length of the time step.

3.2 | Split-merge based adaptivity

Winchenbach et al.[2] achieved SPH adaptivity by splitting and merging particles with strict conservation of the total fluid mass. In this method, the desired size for each particle is computed using a sizing function based on the distance to the fluid's free surface:

$$m_i^{\text{opt}} = m^{\text{base}}\left(\frac{\min(|\phi_i^f|, |\phi_{\max}|)}{|\phi_{\max}|}(1-\alpha) + \alpha\right), \tag{8}$$

where m^{opt} is the optimal mass that is the desired mass for the particle. α is the adaptivity ratio that denotes the largest mass ratio allowed between particles. m^{base} is the largest allowed particle mass. ϕ^f is the distance to the fluid's free surface similar to Horvath and Solenthaler,[1] which is negative inside the fluid. $\phi_{\max}$ is the max distance to the surface within which particles are refined. Equation (8) ensures that the optimal mass decreases linearly and smoothly as the distance to the free surface decreases.

Splitting or merging is applied to each particle to adjust its mass towards m^{opt} if necessary. To determine the operation each particle needs, the particles are classified into five categories according to the ratio between the particle's mass and the optimal mass, $m_i^{\text{rel}} = m_i/m_i^{\text{opt}}$. The five classes are S ($m_i^{\text{rel}} < 0.5$), s ($0.5 \leq m_i^{\text{rel}} \leq 0.9$), o ($0.9 < m_i^{\text{rel}} < 1.1$), l ($1.1 \leq m_i^{\text{rel}} \leq 2$), and L ($2 < m_i^{\text{rel}}$)

Particles of class L undergo splitting to become multiple smaller particles (Section 3.2.1); particles of class s are merged into nearby s or S particles by distributing all of its mass to nearby particles (Section 3.2.2); and particles of class l redistribute their excess mass to nearby s particles similarly to merging (Section 3.2.2). The process is shown schematically in Figure 2.

3.2.1 | Splitting

Particles of class L are split into n children particles where $n = \lceil m_i/m_i^{\text{opt}} \rceil$. The mass and positions of children particles are determined using precomputed split patterns. They are further optimized online by solving a minimization problem

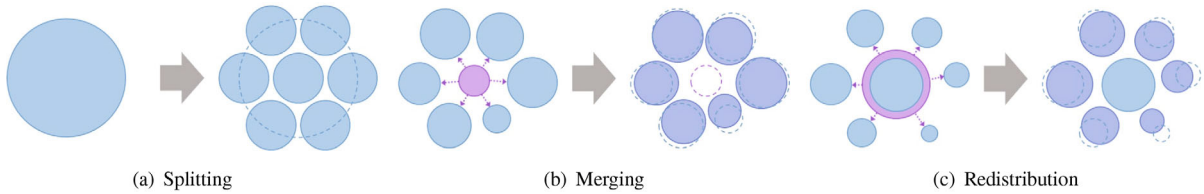


FIGURE 2 Schematic diagram of particle (a) splitting, (b) merging, and (c) redistribution.

on density error to reduce the error induced by splitting.[12] The other physical attributes of children particles are inherited from their parents.

3.2.2 | Merging and redistribution

Particles of class s undergo merging, distributing all of their mass to nearby s or S particles, while particles of class l redistribute their excess mass $m_i - m_i^{\text{opt}}$ to nearby s particles. The distance between the mass-distributing particle i and the mass-receiving particle j must be within $h_i/2$. The mass to distribute m^{dist} is equally divided among the mass-receiving particles, while the other physical attributes are weight-averaged between i and j as:

$$A_j^* = \frac{\frac{m^{\text{dist}}}{n} A_i + m_j A_j}{\frac{m^{\text{dist}}}{n} + m_j}, \tag{9}$$

where A^* is the attribute's new value after redistribution, and n is the number of mass-receiving particles.

3.2.3 | Techniques to increase stability

Even with optimized splitting patterns and online optimization, splitting and merging can still introduce some density errors.[12] Temporal blending[2] and local viscosity[12] can increase the stability of particle splitting and merging.

Temporal blending is applied to particles that recently participated in splitting, merging, or mass redistribution. The original particle o before splitting or redistributing mass is stored. Then, o is used to modify the density and velocity of its children or particles that received mass from it. The blended density is computed as:

$$\rho_i^{\text{blended}} = (1 - \beta)\rho_i + \beta\rho_o, \tag{10}$$

where ρ_o is the density of the original particle, computed using Equation (3), but ignoring the density contribution from its children. β is a temporal blending factor that is initialized to 0.5 for particles generated in splitting and 0.2 for particles involved in redistribution or merging. Then, β is decreased by 0.1 every time step until it reaches 0. ρ_i^{blended} is used instead of ρ_i in further computations. The velocity of the original particle $\mathbf{v}_o$ is set to the average velocity of all its children particles and is used to update the original particle's position $\mathbf{x}_o$. The blended velocity $\mathbf{v}_i^{\text{blended}} = (1 - \beta)\mathbf{v}_i + \beta\mathbf{v}_o$ is used instead of $\mathbf{v}_i$ to update the particle's position $\mathbf{x}_i$.

In the local viscosity technique, the viscosity coefficient υ in Equation (6) is multiplied with a factor $1 + 0.5(\beta_i + \beta_j)/2$ to increase the viscosity of newly split, merged, or redistributed particles locally, thus increasing stability.

4 | SEMI-ANALYTIC BOUNDARY HANDLING

A suitable boundary handling method for adaptive SPH is proposed in Reference 19. This method uses a local plane to approximate the boundary at each particle. The plane approximation is based on a discrete SDF evaluated before simulation. The plane's distance d to the particle equals the SDF magnitude at the particle's position, and the normal vector of

the plane is the normalized SDF gradient. To approximate the boundary's contribution, the integral of the kernel function W is evaluated over the region on the side of the plane that contains the boundary:

$$\lambda'(d) = \begin{cases} \frac{1}{30}[81q^6 - 144q^5 + 80q^3 - 42q + 15] & 0 \le q \le 0.5 \\ \frac{-8}{15}[2q^6 - 9q^5 + 15q^4 - 10q^3 + 3q - 1] & 0.5 < q \le 1 \\ 1 - \lambda'(-q) & -1 \le q < 0 \end{cases}, \tag{11}$$

where $q = d/h_i$.

A penalty term is introduced to penalize solid penetration: $\beta(d) = 1 - d/h_i$. The penalty term is directly multiplied onto $\lambda'(d)$, obtaining the corrected value $\lambda(d) = \beta(d)\lambda'(d)$.

The density equation Equation (3) is then modified to consider the boundary:

$$\rho_i = \sum_j m_j W_{ij} + \sum_b \rho_0 \lambda_i^b, \tag{12}$$

where b denotes the boundary object(s). $\lambda_i^b = \lambda(\text{sdf}^b(\mathbf{x}_i))$, where $\text{sdf}^b(\mathbf{x}_i)$ is the signed-distance to b from the position of i.

The pressure at the boundary can be mirrored[13] or extrapolated[14] from the pressure of nearby fluid particles. In the case of pressure mirroring, the pressure force from the boundary is:

$$\mathbf{F}^p_{i \leftarrow b} = -m_i \rho_0 \frac{p_i}{\rho_i^2} \nabla \lambda_i^b. \tag{13}$$

The friction between the fluid and solid follows the Coulomb model, which is proportional to the boundary's pressure force on the particle:

$$\mathbf{F}^{fric}_{i \leftarrow b} = \mu ||\mathbf{F}^p_{i \leftarrow b}|| \mathbf{t}_{ib}, \tag{14}$$

where $\mathbf{t}_{ib}$ is a unit vector pointing to the direction of relative tangential velocity between fluid and boundary.

The total force from the boundary objects to a particle is $\mathbf{F}_i^{\text{bound}} = \sum_b (\mathbf{F}^p_{i \leftarrow b} + \mathbf{F}^{fric}_{i \leftarrow b})$.

5 | ADAPTIVE BOUNDARY COUPLING

The region near the boundary objects is often of interest in fluid-boundary coupling. Therefore it is desirable to use the refined particles near the important boundary objects to emphasize the details of boundary coupling. However, refining the entire surface can cause an unnecessary increase in particle number. Previous work[2] refines particles only near the free surface of the fluid, which cannot cover the entire boundary region and can cause detail loss in boundary coupling. Moreover, in some cases, only the region near the boundary object is of interest, such as a ship sailing on a vast water surface.

Given the above problems, we propose a boundary refinement method to refine particles near specific boundary objects. The method can either be used alone to refine only near the boundary or combined with surface-based refinement schemes[2] to refine both the surface and the boundary regions, which is outlined in Algorithm 1.

To achieve particle refinement near boundary objects, we modify the sizing function of Equation (8) to consider the distance to boundary objects. Instead of optimizing particle mass according to the distance to the free surface ϕ^f, we use the distance to boundary objects to define a value ϕ^b to adjust the particle size:

$$\phi_i^b = \min\left(\max_{b^*}\left(-\text{sdf}^{b^*}(\mathbf{x}_i) + \phi_{b^*}^{\text{fine}}\right), 0\right), \tag{15}$$

where b^* denotes the interested boundary objects near which the fluid particles should be refined. $\phi_{b^*}^{\text{fine}}$ is a user-controlled parameter: all particles with distance to b^* smaller than $\phi_{b^*}^{\text{fine}}$ are refined to the smallest scale. We use $\phi_{b^*}^{\text{fine}} = h_i$ unless explicitly mentioned.

Algorithm 1. Boundary refinement for SPH. New steps in our method are marked blue

SPH computation
Establish neighbor list
Compute and blend density
Calculate advection forces
Compute fluid pressure
Apply boundary pressure and friction
Update and blend velocity
Update position
Adaptive mechanism
Compute ϕ^b from boundary SDF (Equation 15)
Wake flow preservation (Algorithm 2)
if combine surface-based: detect surface
Calculate optimal mass (Equation 16)
Splitting, merging, and redistribution

To only refine the region near the boundary, one can replace ϕ_i^f with ϕ_i^b in Equation (8), namely:

$$m_i^{\text{opt}} = m^{\text{base}} \left(\frac{\min(|\phi_i^b|, |\phi_{\max}|)}{|\phi_{\max}|}(1-\alpha) + \alpha \right), \tag{16}$$

and to refine both near the boundary region and the fluid surface, one should substitute ϕ_i^f in Equation (8) with $\phi_i = \max(\phi_i^b, \phi_i^f)$.

5.1 | Wake flow preservation by delaying merge

For scenarios where the fluid flows past the boundary object or the boundary object moves through the fluid, interesting wake flow effects are generated behind the boundary object, such as the wave behind a ship moving through water. However, when using ϕ_i^b to determine particle size, the wake flow loses detail because the optimal mass quickly returns to the largest value as the boundary object moves away from the particle. Consequently, the particles in the wake flow merge into larger particles, reducing the detail level.

To preserve wake flow detail, we inhibit the particle merging process if the particle was recently in the vicinity of the boundary, as Algorithm 2 shows. For each interested boundary object b^*, the user can set a duration to delay merging $\tau_{b^*}^{\max}$ for particles that have flowed around it, and each particle tracks its remaining duration to delay merging, denoted as τ_i. Every time step, for each particle i and each interested boundary object b^*, if the particle satisfies $\text{sdf}^{b^*}(\mathbf{x}_i) \le h_i$, we set its remaining duration $\tau_i := \tau_{b^*}^{\max}$ if the duration is not already longer. The duration is decremented at each time step by the time step length Δt as $\tau_i := \tau_i - \Delta t$. As long as the particle satisfies $\tau_i > 0$, ϕ_i^b is not allowed to decrease: if the newly computed value $\phi_{i,t}^b$ for the time step t is lower than the previous value $\phi_{i,t-1}^b$, we set it to $\phi_{i,t}^b := \phi_{i,t-1}^b$. To

Algorithm 2. Wake flow preservation

for each particle i
for each interested boundary object b^*:
if $\text{sdf}^{b^*}(\mathbf{x}_i) \le h_i$:
set duration $\tau_i := \max(\tau_{b^*}^{\max}, \tau_i)$
decrement duration $\tau_i := \tau_i - \Delta t$
if $\tau_i > 0$:
$\phi_{i,t}^b := \max(\phi_{i,t}^b, \phi_{i,t-1}^b)$

retain the smoothness of the ϕ^b field, the ϕ_i^b values for $\tau_i > 0$ are then propagated to surrounding particles using the method from Horvath and Solenthaler.[1] This approach effectively delays particle merging in the wake flow by the time length of $\tau_{b^*}^{\max}$.

6 | RESULTS AND DISCUSSIONS

To test the effectiveness and capabilities of our method, in this section, we use multiple scenarios to compare the proposed approach with simulations of uniform particle scale and the surface-based state-of-the-art SPH adaptive scheme,[2] which adapts resolution near the free surface.

We use weakly compressible SPH[20] for fluid simulation and the semi-analytic boundary handling method proposed in Reference 19 for boundary coupling. The time step length is determined using the CFL condition with a scaling parameter of 0.5. Surface reconstruction and rendering are conducted using Houdini and mantra with Intel Xeon Gold 5218. All simulations are coded using the Taichi programming language[22] and run on an NVIDIA Tesla V100 GPU.

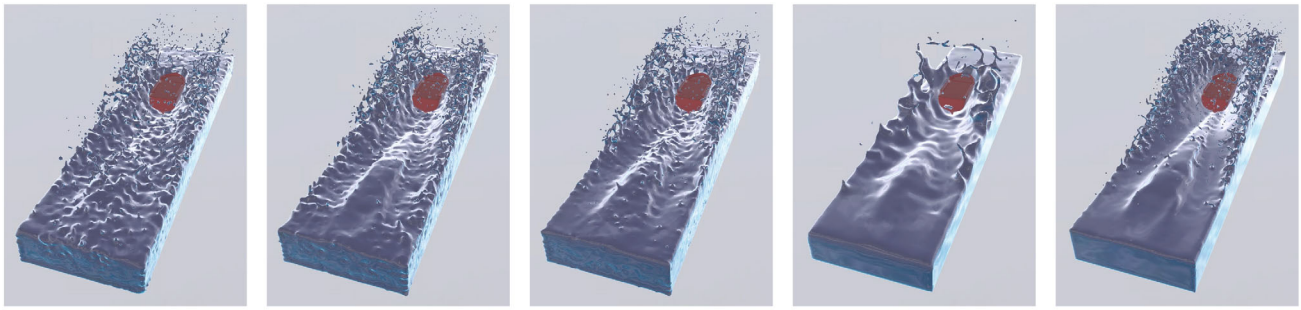
6.1 | Efficiency comparison

Efficiency is a crucial aspect of adaptive simulation mechanisms. We show that our method can produce a similarly detailed result using fewer particles than the surface-based adaptive mechanism[2] and is more vivid than a low-resolution simulation.

6.1.1 | Boat-sailing

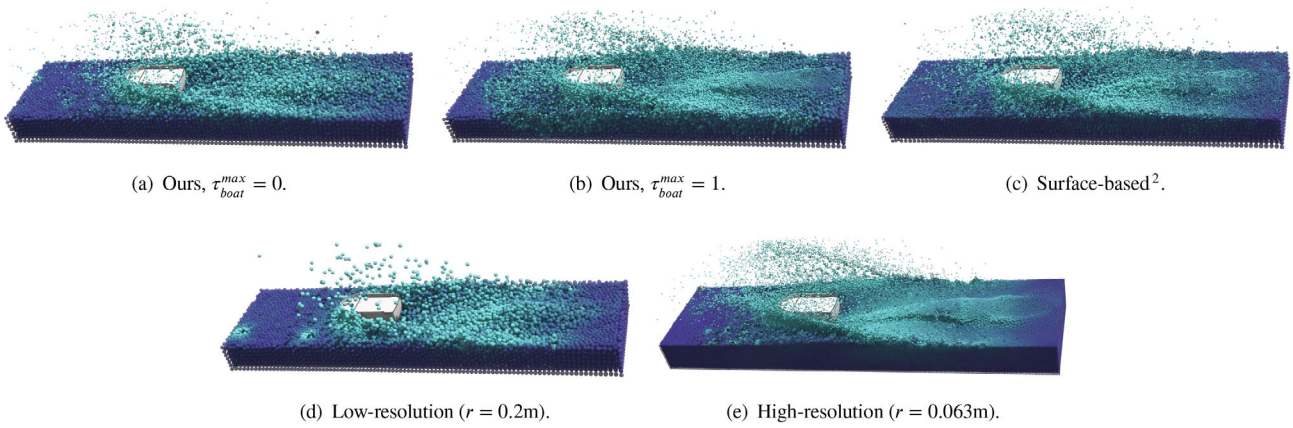
In this scenario, a boat sails quickly across a calm fluid surface, stirring up waves and splashes around and behind the body. For the adaptive methods, we use the max particle size $r_{\text{base}} = 0.2$ and an adaptive ratio $\alpha = 32$ and choose $\phi_{b^*}^{\text{fine}} = 0$ in our method. For uniform size simulations, we perform a low-resolution and a high-resolution simulation using $r = 0.2$ and $r = 0.063$, where the particle size respectively equals the largest and smallest particle size in adaptive simulation. The particles are color-coded according to velocity.

Figures 3 and 4 show the experimental results rendered with reconstructed fluid surface and particles, respectively. Considering the splashes around the ship, the visual level of detail is similar between the previous surface-based method[2] (Figure 3c) and our method with or without wake flow preservation (Figure 3b,a). The three adaptive results are all much more detailed than the uniform low-resolution simulation (Figure 3d), though they may lack some detail compared to high-resolution (Figure 3e). For the wake flow effect, the result of our method without wake flow preservation (Figure 3a) is relatively coarse, but when wake flow preservation is added to our method (Figure 3b), the visual detail level is comparable to the surface-based method (Figure 3c).



(a) Ours, $\tau_{boat}^{max} = 0$. (b) Ours, $\tau_{boat}^{max} = 1$. (c) Surface-based[2]. (d) Low-resolution. (e) High-resolution.

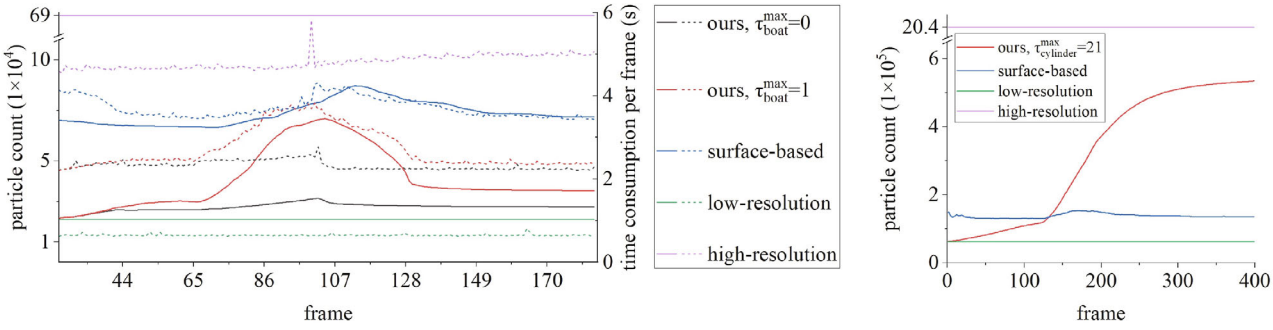
FIGURE 3 Surface reconstruction result of the boat-sailing experiment: (a–c) Are adaptive simulations using max particle size $r_{\text{base}} = 0.2$ and adaptive ratio $\alpha = 32$; (d, e) are uniform scale simulations with $r = 0.2$ and $r = 0.063$. (a) Ours, $\tau_{\text{boat}}^{\max} = 0$; (b) ours, $\tau_{\text{boat}}^{\max} = 1$; (c) surface-based;[2] (d) low-resolution; (e) high-resolution



(a) Ours, $\tau_{boat}^{max} = 0$. (b) Ours, $\tau_{boat}^{max} = 1$. (c) Surface-based[2].

(d) Low-resolution ($r = 0.2$m). (e) High-resolution ($r = 0.063$m).

FIGURE 4 Boat-sailing experiment. Adaptive methods use $r_{\text{base}} = 0.2$ and $\alpha = 32$. (a) Ours, $\tau_{\text{boat}}^{\max} = 0$; (b) ours, $\tau_{\text{boat}}^{\max} = 1$; (c) surface-based;[2] (d) low-resolution ($r = 0.2$ m); (e) high-resolution ($r = 0.063$ m)



(a) Boat-sailing experiment. (b) Cylinder moving experiment.

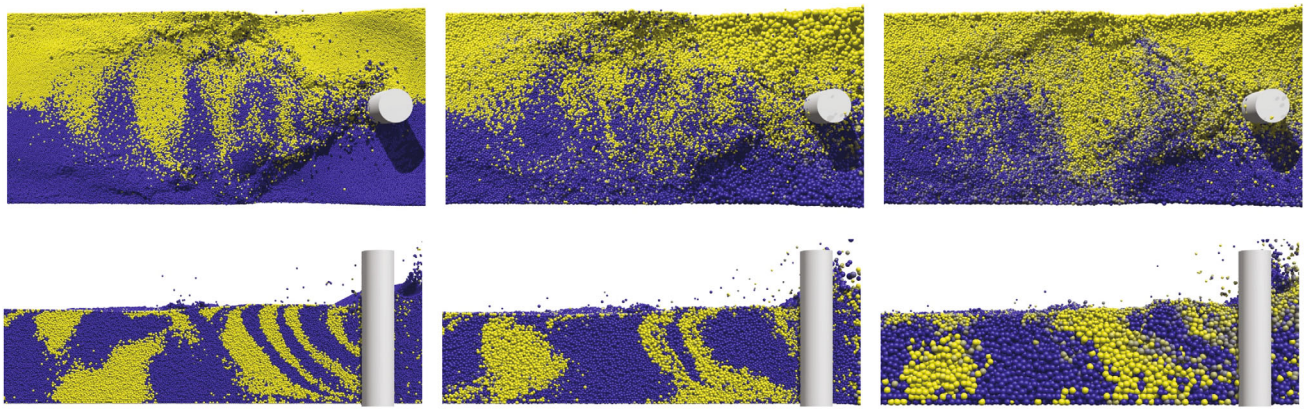
FIGURE 5 Particle count and time consumption in (a) boat-sailing and (b) cylinder moving experiment. Solid lines denote particle count; dashed lines denote time consumption. One frame equals 1/24 s.

To evaluate the efficiency of the proposed method, the particle count and time consumption using different methods are plotted in Figure 5a. The solid lines show the particle count, and the dashed lines show the time consumption. It can be seen that the uniform low-resolution simulation uses the least particles. While our adaptive method with $\tau_{\text{boat}}^{\max} = 0$ increases the particle count slightly. Further, our method with $\tau_{\text{boat}}^{\max} = 1$ begins with a low particle count, which increases since frame 65 when the boat starts moving and producing wake flow, and decreases back to a low value after frame 107 when the boat leaves the simulated zone because the duration to delay merging for the wake flow has expired. Under both conditions, our method significantly reduces the average particle count over time compared to the surface-based method[2] and the high-resolution simulation. The time consumption curves follow a similar trend. This experiment shows that our method can reduce the time cost compared to the state-of-the-art surface-based adaptive mechanism[2] with no or negligible negative influence on visual quality.

6.2 | Effect evaluation

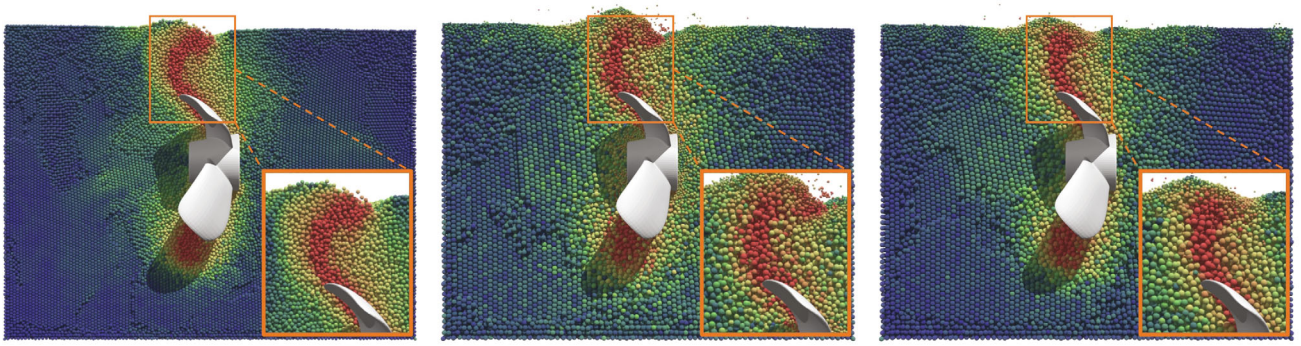
6.2.1 | Cylinder moving

Figure 6 shows an experiment where a cylinder is moved horizontally through a water tank in a straight line to generate a wave pattern. The particles are colored according to whether their initial positions are on one side of the moving path of the cylinder. A jagged pattern can be seen from the vertical view (first row), and curved patterns can be seen from the side view of a cross-section (second row). Figure 6a shows the result of a uniformly scaled high-resolution simulation performed using the smallest allowed particle size in the two adaptive simulations, $r = 0.063$, which should be the most accurate among the three simulations. Figure 6b uses our boundary refinement method where particles are adaptively



(a) High-resolution ($r = 0.063$m). (b) Ours, $\tau^{max}_{cylinder} = 21$. (c) Surface-based[2].

FIGURE 6 Cylinder moving experiment, where the cylinder moves horizontally in the tank. The first row is the vertical view; the second row is a cross-section of the side view. Our method better reproduces the jagged (vertical view) and curved (side view) wave patterns in the high-resolution simulation. Adaptive methods use max particle size $r_{\text{base}} = 0.2$ and $\alpha = 32$. (a) High-resolution ($r = 0.063$ m); (b) ours, $\tau^{\text{max}}_{\text{cylinder}} = 21$; (c) surface-based[2]



(a) High-resolution ($r = 0.063$m). (b) Ours + surface-based[2]. (c) Only surface-based.

FIGURE 7 A cross-section of the propeller spinning experiment where color denotes velocity. Adaptive methods use $r_{\text{base}} = 0.1$, $\alpha = 32$, $\tau^{\text{max}}_{\text{propeller}} = 0$. In the zoomed-in areas, our method (b) generates a more detailed vortex compared to (c), closer to the high-resolution result (a). (a) High-resolution ($r = 0.063$ m); (b) ours + surface-based;[2] (c) only surface-based

refined near the cylinder. We use $\tau^{\text{max}}_{\text{cylinder}} = 21$, which is longer than the simulation duration, to prevent wake flow merging for the entire simulation to achieve a high level of detail. Figure 6c uses surface-based refinement.[2] The two adaptive simulations use $r_{\text{base}} = 0.2$ and $\alpha = 32$. Compared to the surface-based method, our method is more similar to the high-resolution simulation in terms of the size and number of jags from the vertical view and the shape of the curved pattern in the cross-section, indicating our method can achieve a better accuracy compared to surface-based method[2] under certain settings.

The particle count of the cylinder moving experiment is displayed in Figure 5b. Our method uses a larger number of particles compared to the surface-based method[2] to achieve higher accuracy, but the particle count is still considerably lower than the high-resolution simulation.

6.2.2 | Propeller spinning

In this experiment, a propeller is fully submerged in the fluid and spins horizontally to generate turbulence. We use this scenario to demonstrate the ability of our method to be combined with surface-based refinement.[2]

Figure 7 shows the particle view of a cross-section of the propeller spinning experiment. From Figure 7b, it can be seen that the particle size is refined both near the surface and around the propeller. Compared with only using surface-based

refinement as in Figure 7c, the combined method can generate more details in a vortex, as shown in the zoomed-in areas, and also has a more similar shape to the vortex in a high-resolution simulation (Figure 7a). This experiment shows that adding our method to the surface-based method[2] can enhance dynamic details produced by boundary coupling.

7 | CONCLUSION AND DISCUSSION

7.1 | Limitations

7.1.1 | Visual effects

In the surface-based only SPH adaptive mechanism, the sizes of the particles are always transformed smoothly in space according to a continuous and stable distance. However, our boundary-based adaptive method may encounter the difficulties of particle merging. The divided small particles, as a part of the waves stirred up by the solid boundary, may not have suitable *s* or *S* neighbor particles to be merged when falling back to the fluid surface.

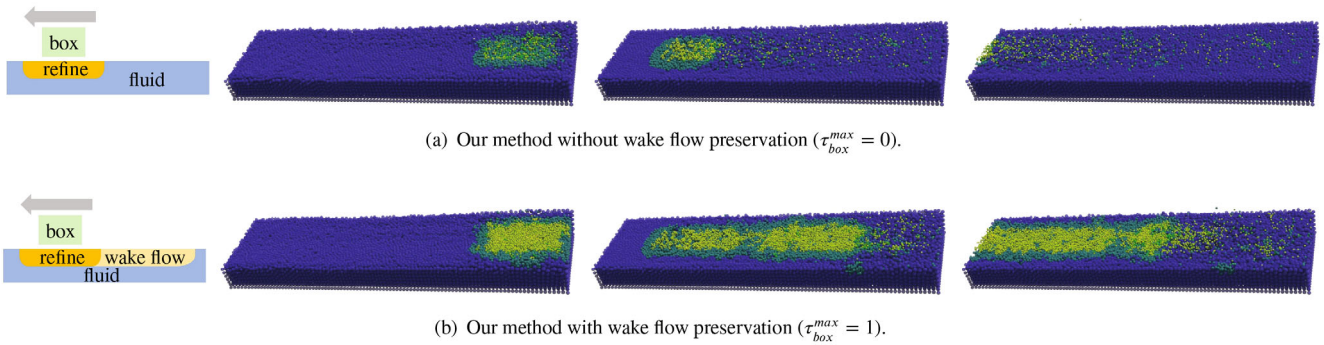
As shown in Figure 8, we make a solid box fly over the fluid surface without substantive contact to demonstrate this issue. Only the sizes of the fluid particle will be affected by the signed distance field from the solid object. We carried out this experiment separately with and without the wake flow preservation mechanism. The sizes of the particles are color-coded as smaller particles have brighter appearances. Here we can see that our wake flow preservation mechanism delays the merging process effectively.

However, we can also observe that when the solid object leaves the fluid, some small particles are left unmerged on the surface. This is because our method only allows the merging procedure to take place between particles of *s* or *S* class. When small particles begin to merge, there may be no *s* or *S* particles in the range of merging. On the other hand, the surface-based adaptive mechanism naturally guarantees fluid particles near each other obtain similar sizes as it only depends on particles' distances to the fluid surface, so particles of suitable class are easier to be found.

7.1.2 | Energy conservation

The particle splitting and merging techniques applied in this article would also introduce potential energy conservation violations into the simulation. We measured the total energy (including kinetic and gravitational potential energy) for the boat-sailing experiment in Figure 4 and the moving box experiment in Figure 8 and drew the statistical diagram in Figure 9.

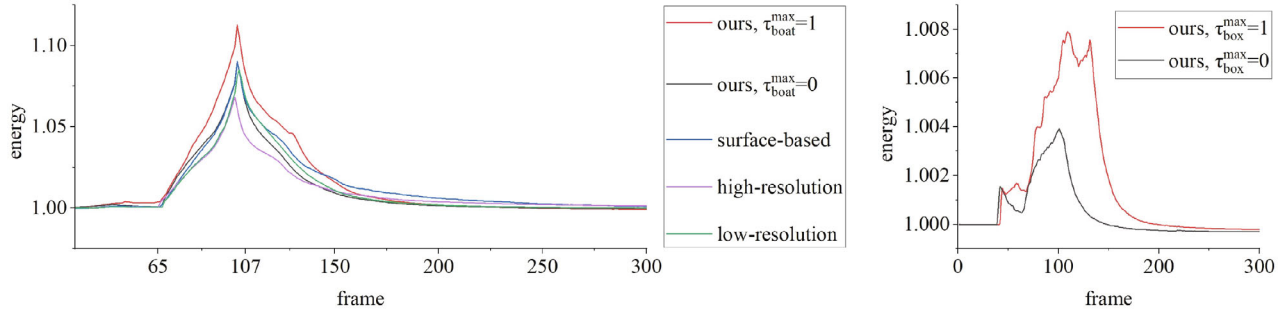
In Figure 9a, we can see that for the boat-sailing experiment, both our method and surface enhancement method[2] obtain energy higher than both high-resolution and low-resolution simulation results. This manifests in the random



(a) Our method without wake flow preservation ($\tau_{box}^{max} = 0$).

(b) Our method with wake flow preservation ($\tau_{box}^{max} = 1$).

FIGURE 8 Moving box experiment. In this experiment, a box-shaped boundary object moves close to the fluid surface but does not touch the fluid, causing particle refinement near it but not disturbing it. The leftmost column shows the experimental setup; the other columns show the result. Color denotes particle size. Our method uses $r_{\text{base}} = 0.2$ and $\alpha = 32$. The distance between the box and the fluid surface is 0.1. (a) Our method without wake flow preservation ($\tau_{\text{box}}^{\max} = 0$); (b) our method with wake flow preservation ($\tau_{\text{box}}^{\max} = 1$)



(a) Boat-sailing experiment. (b) Moving box experiment.

FIGURE 9 Statistical analysis of energy change (sum of total kinetic and gravitational potential energy) in (a) boat-sailing and (b) moving box experiments with a frame rate of 24 fps. The energy change with time is expressed with the ratio of the current time's energy to the first frame's.

spatter of fluid particles during the simulation, indicating the instability issue of the method. This issue is further demonstrated in Figure 9b, where, as the chart shows, the violation of energy conservation energy happens during the merging process. The energy cannot remain stable when the solid box starts to cause the fluid to split and merge. Moreover, for the experiment applying wake flow preservation, the period of conserving energy performs worse. Though this instability is relatively small (generally less than 1‰ energy fluctuation), it can generate unexpected artifacts if the simulation scenario is not carefully configured.

7.2 | Summary

We propose a new boundary refinement mechanism for adaptive SPH to refine particles near the coupling boundary and retain resolution for the flow generated by solid objects. Experiments show that, compared to the surface-based only refinement mechanism[2] (Appendix S1) , our method can improve the computation efficiency significantly for enhancing fluid details when simulating fluid–solid coupling scenarios. Further, the proposed method can produce more accurate visual results for underwater coupling scenarios such as propeller spinning. Therefore, our method can be widely applied to most fluid simulation scenarios to improve efficiency and visual quality since many fluid simulations focus on the interaction between fluid and solid. A limitation of our method is that when a solid particle leaves the refined region due to splashing and so forth it would lack merge partners and remain in the unrefined region, causing a small redundancy in particles. Also, the splitting and merging process of our method would cause a potential violation of energy conservation and generate visual artifacts such as unrealistic splashes.

In the future, we will explore a more stable and accurate particle splitting-merging mechanism for adaptive SPH fluid simulation. The critical issue is balancing the particles' sizes close to each other and creating a progressive merging process to avoid a sudden change of local density field.

ACKNOWLEDGMENTS

This research was funded by: Horizon 2020-Marie Skłodowska-Curie Action-Individual Fellowships (No. 895941), National Natural Science Foundation of China (No. 61873299), Key Research and Development Project of Hainan Province (No. ZDYF2020031), Fundamental Research Funds for the Central Universities (QNXM20220043).

ORCID

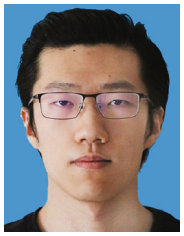
Yanrui Xu https://orcid.org/0000-0002-2154-1178
Xiaokun Wang https://orcid.org/0000-0002-4449-591X

REFERENCES

1. Horvath C, Solenthaler B. Mass preserving multi-scale SPH. Pixar technical memo #13-04; 2013.
2. Winchenbach R, Hochstetter H, Kolb A. Infinite continuous adaptivity for incompressible SPH. ACM Trans Graph. 2017;36(4):1–10. https://doi.org/10.1145/3072959.3073713

3. Ando R, Batty C. A practical octree liquid simulator with adaptive surface resolution. ACM Trans Graph. 2020;39(4):32. https://doi.org/10.1145/3386569.3392460
4. Xiao Y, Chan S, Wang S, Zhu B, Yang X. An adaptive staggered-tilted grid for incompressible flow simulation. ACM Trans Graph. 2020;39(6):1–5. https://doi.org/10.1145/3414685.3417837
5. Sato T, Wojtan C, Thuerey N, Igarashi T, Ando R. Extended narrow band FLIP for liquid simulations. Comput Graph Forum. 2018;37(2):169–77. https://doi.org/10.1111/cgf.13351
6. Nakanishi R, Nascimento F, Campos R, Pagliosa P, Paiva A. RBF liquids: an adaptive PIC solver using RBF-FD. ACM Trans Graph. 2020;39(6):1–13. https://doi.org/10.1145/3414685.3417794
7. Koschier D, Bender J, Solenthaler B, Teschner M. Smoothed particle hydrodynamics techniques for the physics based simulation of fluids and solids. Eurographics; 2019. p. 1–41.
8. Adams B, Pauly M, Keiser R, Guibas LJ. Adaptively sampled particle fluids. ACM Trans Graph. 2007;26(3):48–es. https://doi.org/10.1145/1276377.1276437
9. Solenthaler B, Gross M. Two-scale particle simulation. ACM Trans Graph. 2011;30(4):1-8. https://doi.org/10.1145/2010324.1964976
10. Orthmann J, Kolb A. Temporal blending for adaptive SPH. Comput Graph Forum. 2012 dec;31(8):2436–49. https://doi.org/10.1111/j.1467-8659.2012.03186.x
11. Vacondio R, Rogers BD, Stansby PK, Mignosa P. Variable resolution for SPH in three dimensions: towards optimal splitting and coalescing for dynamic adaptivity. Comput Methods Appl Mech Eng. 2016;300:442–60. Available from: https://www.sciencedirect.com/science/article/pii/S0045782515003813
12. Winchenbach R, Kolb A. Optimized refinement for spatially adaptive SPH. ACM Trans Graph. 2021;40(1). https://doi.org/10.1145/3363555
13. Akinci N, Ihmsen M, Akinci G, Solenthaler B, Teschner M. Versatile rigid-fluid coupling for incompressible SPH. ACM Trans Graph. 2012;31(4):1-8. https://doi.org/10.1145/2185520.2185558
14. Band S, Gissler C, Peer A, Teschner M. MLS pressure boundaries for divergence-free and viscous SPH fluids. Comput Graph. 2018;76:37–46. Available from: https://www.sciencedirect.com/science/article/pii/S009784931830116X
15. Fujisawa M, Miura KT. An efficient boundary handling with a modified density calculation for SPH. Comput Graph Forum. 2015;34(7):155–62. https://doi.org/10.1111/cgf.12754
16. Koschier D, Bender J. Density maps for improved SPH boundary handling. Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation. SCA '17. New York, NY: Association for Computing Machinery; 2017. p. 1–10. https://doi.org/10.1145/3099564.3099565.
17. Bender J, Kugelstadt T, Weiler M, Koschier D. Volume maps: an implicit boundary representation for SPH. Motion, interaction and games. MIG '19. New York, NY: Association for Computing Machinery; 2019. p. 1–10. https://doi.org/10.1145/3359566.3360077
18. Chang Y, Liu S, He X, Li S, Wang G. Semi-analytical solid boundary conditions for free surface flows. Comput Graph Forum. 2020;39(7):131–41. https://doi.org/10.1111/cgf.14132
19. Winchenbach R, Akhunov R, Kolb A. Semi-analytic boundary handling below particle resolution for smoothed particle hydrodynamics. ACM Trans Graph. 2020;39(6). https://doi.org/10.1145/3414685.3417829
20. Becker M, Teschner M. Weakly compressible SPH for free surface flows. Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Vol. 9; 2007. p. 209–17.
21. Akinci N, Akinci G, Teschner M. Versatile surface tension and adhesion for SPH fluids. ACM Trans Graph. 2013;32(6):1–8. https://doi.org/10.1145/2508363.2508395
22. Hu Y, Li TM, Anderson L, Ragan-Kelley J, Durand F. Taichi: a language for high-performance computation on spatially sparse data structures. ACM Trans Graph. 2019;38(6):1-16. https://doi.org/10.1145/3355089.3356506

AUTHOR BIOGRAPHIES



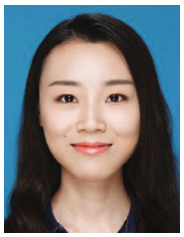
Yanrui Xu is a Ph.D. student at the School of Intelligence Science and Technology, University of Science and Technology Beijing. He received his Master's degree from the University of Science and Technology Beijing in 2020. His research interests include physical-based simulation and fluid animation.

Chongming Song is a postgraduate student at the School of Intelligence Science and Technology, University of Science and Technology Beijing. She received an undergraduate Diploma in University of Science and Technology Beijing, in 2021. Her research field is computer graphics, especially physically-based fluid simulation.

Xiaokun Wang is an associate professor in Intelligence Science and Technology, University of Science and Technology Beijing, China. He received a Ph.D. degree in Computer Science and Technology from the University of Science and Technology Beijing, in 2017. He is currently working at the National Centre for Computer Animation at Bournemouth University funded by the EU's Horizon 2020 Marie Curie Individual Fellowship. His research interests include computer graphics, virtual reality, and human–computer interaction.

Xiaojuan Ban is a professor at the School of Intelligence Science and Technology, University of Science and Technology Beijing. She is the leader of the Artificial Intelligence and 3D Visualization Group at University of Science and Technology Beijing, China. She received her master's degree in computer application and Ph.D. degree in control theory and control engineering from the University of Science and Technology Beijing. Her research interests include computer graphics, artificial intelligence, human–computer interaction, big data analysis, and 3D visualization.

Jiamin Wang is a Ph.D. student at the School of Intelligence Science and Technology, University of Science and Technology Beijing. She received an undergraduate Diploma in University of Science and Technology Beijing, in 2021. Her research field is computer graphics, especially physically-based fluid simulation.



Yalan Zhang received her Ph.D. degree in Computer Science and Technology from University of Science and Technology Beijing, China, in 2020. Currently, she is a lecturer in School of Intelligence Science and Technology, University of Science and Technology Beijing. Her research interests include computer graphics, 3-D visualization, and computer vision.

Jian Chang is a professor in National Centre for Computer Animation at Bournemouth University, UK. He received his Ph.D. degree in Computer Graphics from Bournemouth University, in 2007. His research interests include physics based modeling (deformation & fluid), motion synthesis, virtual reality (surgery simulation), and novel HCI (eye tracking, gesture control, and haptic).

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Xu Y, Song C, Wang X, Ban X, Wang J, Zhang Y, Chang J. Spatial adaptivity with boundary refinement for smoothed particle hydrodynamics fluid simulation. Comput Anim Virtual Worlds. 2023;e2136. https://doi.org/10.1002/cav.2136