**Adaptive smoothing length method based on weighted average of neighboring particle density for SPH fluid simulation**

Rongda ZENG¹, Zihao WU¹, Shengbang DENG¹, Jian ZHU⁴*, Xiaoyu CHI²*

1. School of Computer Science, Guangdong University of Technology, Guangzhou 510006, China
2. Qingdao Research Institute of Beihang University, Qingdao 266000, China

*Corresponding author, dr.zhuj@gmail.com; terry.chi@goertek.com

**Received:** 10 October 2018 **Accepted:** 18 December 2018

**Supported by** National Natural Science Foundation of China (61976052, 61876043); National Key R&D Program of China (2017YFB1002701, 2017YFB1201203).

**Citation:** Rongda ZENG, Zihao WU, Shengbang DENG, Jian ZHU, Xiaoyu CHI. Adaptive smoothing length method based on weighted average of neighboring particle density for SPH fluid simulation. Virtual Reality & Intelligent Hardware, 2021, 3(2): 129—141

DOI: 10.1016/j.vrih.2018.12.001

**Abstract**

**Background** In the smoothed particle hydrodynamics (SPH) fluid simulation method, the smoothing length affects not only the process of neighbor search but also the calculation accuracy of the pressure solver. Therefore, it plays a crucial role in ensuring the accuracy and stability of SPH. **Methods** In this study, an adaptive SPH fluid simulation method with a variable smoothing length is designed. In this method, the smoothing length is adaptively adjusted according to the ratio of the particle density to the weighted average of the density of the neighboring particles. Additionally, a neighbor search scheme and kernel function scheme are designed to solve the asymmetry problems caused by the variable smoothing length. **Results** The simulation efficiency of the proposed algorithm is comparable to that of some classical methods, and the variance of the number of neighboring particles is reduced. Thus, the visual effect is more similar to the corresponding physical reality. **Conclusions** The precision of the interpolation calculation performed in the SPH algorithm is improved using the adaptive-smoothing length scheme; thus, the stability of the algorithm is enhanced, and a larger timestep is possible.

**Keywords** Fluid simulation; SPH; Smoothing length; Adaptive; Particle density

1 **Introduction**

In the field of physics-based fluid simulation, the smoothed particle hydrodynamics (SPH)⁴¹ method is being increasingly employed for various applications, including entertainment technology and engineering design. The SPH method, which is particle-based, separates a continuous fluid into interacting particles⁴². By calculating the force on each particle, the particles are continuously moved, mimicking the flowing fluid. Owing to its convenience and flexibility in complex fluid simulations, SPH has been widely used in digital entertainment and various other fields; moreover, it has already become one of the main tools for fluid simulation.

---

2096-5796©Copyright 2021 Beijing Zhongke Journal Publishing Co. Ltd., Publishing services by Elsevier B.V. on behalf of KeAi Communication Co. Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by/4.0/).
In the SPH method, the smoothing length significantly affects the simulation efficiency and accuracy. Improper setting of the smoothing length can reduce the simulation accuracy considerably. In three-dimensional (3D) simulations, employing 30–40 particles in the support domain generally yields good accuracy. In traditional SPH, the particle support domain is fixed; thus, different regions differ with regard to the number of neighboring particles in the support domain. It is difficult to obtain a consistent kernel approximation in the calculation domain, which leads to excessive particle interpolation errors, as well as the loss of minor details during the fluid simulation. Therefore, in accordance with the idea of allocating more resources to areas of visual interest, researchers investigated adaptive sampling techniques for improving the computational efficiency without compromising the visual quality\textsuperscript{[4–6]}.

In this paper, an adaptive smoothing length SPH (ASLSPH) fluid simulation method is proposed for reducing the calculation error caused by the fixed smoothing length of particles. By establishing the relationship between the smoothing length and the density of particles and adjusting the smoothing length adaptively according to the changes in the particle density, the method can reduce the calculation error, improve the accuracy of interpolation calculations, and achieve simulation results that are closer to the physical reality.

A limitation of our method is that the neighbor search efficiency is lower than that of the traditional method, because the smoothing lengths differ among particles. Therefore, a fast neighbor search scheme based on an auxiliary grid is proposed for increasing the simulation speed. Additionally, to solve the problem of particle interaction asymmetry caused by the variable smoothing length, a symmetric kernel function is designed, which maintains the stability of the algorithm.

The main contributions of this paper are as follows:

• A density-based adaptive smoothing length scheme to improve the calculation accuracy;
• A symmetric interpolation scheme for the kernel function to maintain the stability of the algorithm;
• A grid-based fast neighbor search scheme for particles with a variable smoothing length.

2 Related work

2.1 SPH foundation

The first particle-based method was proposed by Miller et al.\textsuperscript{(7)} Among the particle-based methods, SPH\textsuperscript{(8)} is the most commonly used. In 2003, Muller et al. introduced the SPH method to the field of computer graphics\textsuperscript{(9)}. Since then, scientists have continuously investigated the method, which has promoted its rapid development and progress\textsuperscript{[9–12]}. The incompressibility of fluids—particularly liquids—significantly affects the visual quality of the fluid animation. Thus, scientists have proposed various pressure solvers to ensure fluid incompressibility. Becker and Teschner proposed weakly compressible SPH (WCSPH) based on the Tait equation\textsuperscript{(13)}. Solenthaler and Pajarola proposed predictive–corrective incompressible SPH (PCISPH) to ensure that the density is constant throughout the simulation\textsuperscript{(14)}. Ihmsen et al. proposed implicit incompressible SPH (IISPH)\textsuperscript{(15)}. Bender and Koschier combined two new implicit pressure solvers and proposed divergence-free SPH (DFSPH) to limit volume compression and prevent divergence in the velocity field\textsuperscript{[16,17]}.

2.2 Adaptive sampling methods

Several methods have been proposed by graphics researchers to achieve better simulation results and a higher efficiency. Generally, a larger number of particles corresponds to a higher calculation accuracy, but simulations with a large number of particles are time-consuming. It is difficult to simulate a large number
of particles in a given time range. Therefore, the adaptive sampling technique has been investigated for improving the allocation of computational resources.

The spatial adaptive method attempts to re-sample the particles so that the non-uniform sizes of the particles can be employed to obtain animation results of higher quality. The idea of the adaptive particle size\(^4\) is to dynamically adjust the radius of the particle according to the complexity and visual importance of the fluid region. To minimize the sampling error, the Lagrange multiplier\(^5\) is typically used to minimize the local error function in computational fluid dynamics. In computer graphics, simple and more efficient sampling operations have been used, such as particle merging or splitting\(^6\). In the high-resolution region, a particle is split into a certain number of child particles, whereas in the low-resolution region, they merge into a single parent particle.

In addition to spatial adaptive sampling, timestep adaptation is helpful for improving the simulation efficiency. The adaptive-timestep method\(^7\) dynamically adjusts the integral timestep in each simulation step to automatically adapt to the Courant-Friedrichs-Lewy condition, thus improving the simulation efficiency.

The aforementioned adaptive methods were mainly used to improve the calculation efficiency, whereas the adaptive smoothing length methods are used to improve the calculation accuracy and algorithm stability. Gingold and Monaghan proposed an adaptive smoothing length method for assigning independent smoothing lengths to each particle\(^8\). Subsequently, Nelson and Papaloizou studied the relationship between the smoothing length variation and the momentum and energy equations; further, they proposed a variable smoothing length method to reduce the energy error and improve the calculation accuracy\(^9\). Sigalotti et al. introduced adaptive density kernel estimation and applied an adaptive smoothing length to free surfaces\(^10\). Olejnik et al. used the Okubo-Weiss parameter to update the smoothing length\(^11\). Ren et al. proposed dual-support SPH, which allows the use of a variable smoothing length and satisfies the conservation of momentum and energy\(^12\). The foregoing methods improved the simulation accuracy, but the algorithms are complex, which reduces the simulation efficiency. We propose an adaptive smoothing length algorithm based on particle density variation, which can not only improve the computational accuracy but also maintain a high computational efficiency.

### 2.3 Neighbor search

The neighboring particles of each particle in SPH change dynamically over time. Therefore, a neighborhoods search is required in each timestep to calculate the interpolation weight of the neighboring particles. This is usually the most time-consuming step in the SPH simulation process. For an SPH simulation with a fixed smoothing length, a uniform grid\(^13\) can be used to improve the efficiency of the neighbor search. By reordering the particles\(^14\), the search speed can be further improved. For SPH simulations with a multiresolution particle size or a non-uniform smoothing length, hierarchical data structures such as trees\(^15\) are typically used to conduct the neighbor search. Gong et al. proposed a parallel background grid neighbor-search method based on adaptive mesh refinement, which can significantly improve the neighbor search efficiency\(^16\). Additionally, because there is almost no data dependency, the SPH method can be easily mapped to a graphics processing unit (GPU)\(^17\) to perform parallel computation. A GPU-based neighbor list algorithm\(^18\) can be used to simulate large-scale fluid animation owing to its performance advantages.

### 3 Adaptive-smoothing length method

In contrast with the Eulerian grid method, in SPH, it is difficult to ensure the incompressible property of
the fluid, which may lead to an uneven particle distribution in which the particles are excessively dense or sparse in some regions. The conventional SPH fluid simulation method usually adopts a fixed particle smoothing length for the neighbor search and interpolation calculation. Owing to the uneven distribution of particles, particularly at the free surface, the fixed smoothing length of particles may lead to a large difference in the number of neighbors among particles, which affects the interpolation accuracy.

In Figure 1, particles \( i \) and \( j \) are at different positions in the flow field. The particles around \( i \) are densely distributed, e.g., in the deep-water area, while the particles around \( j \) are sparsely distributed, e.g., near the fluid surface. With a fixed smoothing length \( h \), owing to the insufficient number of neighboring particles, the interpolation accuracy for the physical quantities on particle \( j \) is reduced. To solve this problem, an ASLSPH method based on the particle density is proposed. By establishing the relationship between the smoothing length of the particles and the weighted average of the density of the neighbors of the particles, the smoothing length of the particles can be adjusted adaptively. Even in the case of a special particle distribution, e.g., a lack of neighboring particles or an uneven distribution of particles, the interpolation can maintain a high accuracy, and this yields improved interpolation accuracy and stability of the simulation.

In the proposed method, the smoothing length of each particle is a variable so that the number of neighboring particles can be kept within a stable range; thus, a high interpolation accuracy can be maintained for each particle. Additionally, to ensure that the computed forces satisfy Newton's third law, we designed a corresponding neighbor search scheme and interpolation kernel function for maintaining the stability of the algorithm. The simulation process is presented in Algorithm 1, where the improved parts (relative to the standard SPH method) are shown in bold.

---

**Algorithm 1: ASLSPH method**

1. while simulation do
2. for each particle \( i \) do
3. calculate the smoothing length of \( i \) (Section 3.1)
4. end for
5. adaptive smoothing length neighbor search (Section 3.2)
6. for each particle \( i \) do
7. calculate the density of particle \( i \) (symmetric interpolation, Section 3.3)
8. end for
9. for each particle \( i \) do
10. calculation the pressure force, viscous force and external force of particle \( i \) (symmetric interpolation, Section 3.3)
11. end for
12. for each particle \( i \) do
13. update the velocity of particle \( i \)
14. update the position of particle \( i \)
15. end for
16. end while

---

### 3.1 Density-based adaptive smoothing length

To obtain a high approximation accuracy of the interpolation kernel for each particle, the smoothing length is adjusted adaptively according to a defined scaling factor, which is controlled by the weighted average of
the density of the neighboring particles. Thus, the number of neighboring particles is maintained within a stable range. Specifically, the smoothing length \( h_i \) of particle \( i \) is defined as
\[
h_i' = \alpha_i h,
\]
where \( h \) represents the initial radius of each particle, and \( \alpha_i \) is a scale factor used to adaptively control the smoothing length. \( \alpha_i \) is defined as follows:
\[
\alpha_i = \left( \frac{\rho_i}{\bar{\rho}} \right)^{\frac{1}{2}}
\]
where \( d \) represents the dimension of the simulation (2 and 3 for two-dimensional (2D) and 3D simulations, respectively), \( \rho_i \) represents the density of particle \( i \), and \( \bar{\rho} \) represents the weighted average of the density of the neighboring particles. \( \bar{\rho} \) is given as follows:
\[
\bar{\rho} = \frac{\sum_{j=1}^{n} \rho_j}{\sum_{j=1}^{n} d_j^d}
\]
where \( n \) represents the number of neighboring particles for particle \( i \), \( \rho_j \) represents the density of the \( j \)th neighboring particle, and \( d_j^d \) represents the squared Euclidean distance between particle \( i \) and its \( j \)th neighboring particle. \( d_j^d \) is given as follows:
\[
d_j^d = \left\| x_i - x_j \right\|_2^2
\]
where \( x_j^d \) represents the position of the \( j \)th neighboring particle.

When \( n = 0 \), particle \( i \) is considered as an isolated particle, and we set \( h_i = h \). If \( \rho_i < \bar{\rho} \), the distribution of particles around particle \( i \) is sparse, and the number of neighbors is too small. If the smoothing length is fixed, an interpolation error is generated owing to the insufficient neighboring particles. However, in our method, according to Equation (2), \( \alpha_i > 1 \); thus, a smoothing length of \( >h \) is eventually assigned to \( h_i \). With a greater smoothing length, the number of neighboring particles increases accordingly, and eventually the interpolation error is reduced. In contrast, in places where the particles are densely distributed, the smoothing length \( h_i \) decreases to a value smaller than \( h \), thereby avoiding an excessive number of neighboring particles. As shown in Figure 1, after the smoothing length of particle \( j \) was adaptively adjusted, the number of neighboring particles was approximately identical to that for particle \( i \). Thus, the calculation accuracy of the interpolation kernel can be improved.

### 3.2 Neighbor search scheme

Owing to the inconsistency in the smoothing length of the particles, a problem with our adaptive method is that the determination of the neighboring particles is asymmetric. Suppose that the smoothing length of particle \( i \) exceeds that of particle \( j \); i.e., \( h_i > h_j \). When the distance \( d_{ij} \) between the two particles satisfies \( h_i < d_{ij} < h_j \), particle \( i \) regards particle \( j \) as its neighbor, but particle \( j \) does not. This asymmetry results in a force imbalance between particles \( i \) and \( j \), which is inconsistent with Newton’s third law. To solve this problem, we employ the average value of \( h_i \) and \( h_j \), i.e., \( (h_i + h_j)/2 \), as the comparison benchmark. As long as \( d_{ij} \) is not greater than \( (h_i + h_j)/2 \), \( i \) and \( j \) are considered to be neighboring particles.

Additionally, an improved auxiliary background grid method is used to accelerate the neighborhood search process. The role of the auxiliary grid is to quickly determine the search scope. The side length of the grid cell is the maximum smoothing length of all the particles; this ensures that all neighboring particle pairs are in the same cell or adjacent cells. During the search process, each particle does not need to match
all the particles but only needs to search in its own grid cell or the neighboring grid cells (9 and 27 cells in 2D and 3D, respectively). Thus, the computational complexity of the neighbor search is reduced from \(O(n^3)\) to \(O(nm)\), where \(n\) and \(m\) represent the number of particles in the scene and the average number of particles in the grid cell, respectively. Normally, \(m \ll n\); thus, the search efficiency is significantly improved.

### 3.3 Interacting forces

In SPH, the value of physical quantity \(A\) at position \(x\) is determined by the weighted interpolation sum of its neighboring particles:

\[
A(x_i) = \sum_j m_i \frac{A}{\rho_j} W \left( x_i - x_j, h \right) \tag{5}
\]

where \(m_i\) represents the mass of particle \(j\), \(\rho_j\) represents the particle density, \(W\) represents the interpolation kernel function, and \(h\) represents the smoothing length. Using Equation (5), the pressure on particle \(i\) can be calculated as follows:

\[
P(x_i) = \sum_j m_i \frac{P}{\rho_j} W \left( x_i - x_j, h \right) \tag{6}
\]

where \(P_j\) represents the pressure on particle \(j\). Thus, the net pressure force on particle \(i\) is given as:

\[
f_{\text{pressure}}^{i - j} = -\nabla P = -\sum_j m_i \frac{P}{\rho_j} \nabla W \left( x_i - x_j, h \right) = -\sum_j V_j \nabla W \left( x_i - x_j, h \right) \tag{7}
\]

where \(V_j\) represents the particle volume. By integrating the pressure over the volume of the particle, the whole pressure force can be determined as follows:

\[
f_{\text{pressure}}^{i - j} = -V_i \nabla P = -V_i \sum_j V_j \nabla W \left( x_i - x_j, h \right). \tag{8}
\]

This pressure force is not symmetric; thus, we replace \(P_j\) with \((P_i + P_j)/2\), yielding:

\[
f_{\text{pressure}}^{i - j} = -V_i \sum_j V_j \frac{P_i + P_j}{2} \nabla W \left( x_i - x_j, h \right). \tag{9}
\]

Because we assign different smoothing lengths to the particles, a problem arises: the forces exerted onto adjacent particles differ. For example, the pressure force exerted by particle \(j\) onto particle \(i\) is given as:

\[
f_{\text{pressure}}^{j - i} = -V_j \left( \frac{P_i + P_j}{2} \right) \nabla W \left( x_i - x_j, h_i \right) \tag{10}
\]

while the pressure force exerted by particle \(i\) onto particle \(j\) is given as:

\[
f_{\text{pressure}}^{j - i} = -V_j \left( \frac{P_i + P_j}{2} \right) \nabla W \left( x_i - x_j, h_i \right). \tag{11}
\]

Because we normally have \(h_i \neq h_j\), the calculation results of the interpolation kernel function in the above equations are different, i.e., \(W \left( x_i - x_j, h_i \right) \neq W \left( x_i - x_j, h_j \right)\). Thus, the forces between the neighboring particles are different, i.e., \(f_{\text{pressure}}^{j - i} \neq f_{\text{pressure}}^{i - j}\), which clearly violates Newton’s third law. To solve this problem, we used a combination of their kernel functions. The proposed kernel function \(W_\varphi\) is computed as the average of the kernel functions of the neighboring particles.

\[
W_\varphi = W \left( x_i - x_j, (h_i + h_j)/2 \right). \tag{12}
\]

According to the proposed symmetric kernel function \(W_\varphi\), we have \(f_{\text{pressure}}^{i - j} = f_{\text{pressure}}^{j - i}\). Similarly, other forces such as the viscous force can be defined as symmetric ones.

### 4 Experimental results

We tested the proposed ASLSPH algorithm and compared it with state-of-the-art methods, including...
Rongda ZENG et al: Adaptive smoothing length method based on weighted average of neighboring particle density for SPH fluid simulation

WCSPH[13], PCISPH[14], IISPH[15], and DFSPH[16,17]. All the experimental results reported in this section were obtained using a desktop computer with an Intel i7-7700k central processing unit and 8 GB of memory.

4.1 Number of neighbors

First, we compared the ASLSPH method and the other four methods with regard to the number of neighbors via a 2D dam-break experiment. As shown in Figure 2, different colors were used to represent the number of neighboring particles for each particle, and a change in color from purple to red indicates that the number of neighboring particles changed from 0 to 20. Compared with that for the other methods, the number of neighboring particles was more evenly distributed for the ASLSPH method; this is because the smoothing length was adaptively adjusted to keep the number of neighbors as consistent as possible.

![Figure 2: Comparison of the number of neighbors in the 2D dam-break experiment.](image)

Additionally, we compared the variance of the number of particle neighbors among the various methods at different timesteps, as shown in Figure 3. The X-axis indicates the timestep (the sampling interval was 20), and the Y-axis indicates the variance of the number of particle neighbors for each method at the corresponding timestep. The variance of the ASLSPH method was significantly smaller than those of the other methods.

![Figure 3: Variance of the number of neighbors for each method.](image)

4.2 Density field

We also compared the density field of the ASLSPH method and those of the other four methods in the foregoing 2D dam-break experiment. As shown in Figure 4, the density distribution corresponding to the ASLSPH method was significantly more uniform than that of WCSPH. The iterative correction algorithm of PCISPH failed on the left side of the fluid, exhibiting an uneven density distribution. IISPH and DFSPH exhibited uniform density distributions, but both algorithms required substantially more computations than those required by ASLSPH; this is because they needed to solve the complex pressure Poisson equations.
4.3 Visual effects

Next, we compared the visual quality exhibited by the different methods. Various colors were used to represent the smoothing lengths of the particles in the particle view, and a color change from blue to red indicated that the smoothing length changed from small to large.

Figure 5 shows a simulated 3D dam-break experiment with the water column initially located in the middle. The top row presents the results for ASLSPH in the particle view, and the bottom row presents the corresponding surface rendering results. As shown, the simulation effect was highly realistic, with rich details. Figure 6 compares the simulation results of ASLSPH (left), PCISPH (middle), and WCSPH (right). Owing to the lack of neighboring particles on the free surface, the smoothing lengths of the particles on the free surface increased in ASLSPH, and the colors of the particles on the surface gradually changed to green, yellow, and even red. Clearly, more water droplets were captured compared with the case of PCISPH, and even the effect of a rolling wave was captured. WCSPH uses a gas state equation to calculate the pressure, and the fluid settles down quickly, making it difficult to capture the wave curling effect.
As shown in Figure 7, we simulated the effect of a water column falling onto a dragon-shaped obstacle. The simulation results of ASLSPH and IISPH are presented in the top and bottom rows, respectively. Near the free surface, the ASLSPH method effectively adjusted the smoothing length of the particles, improving the calculation accuracy. Compared with IISPH, ASLSPH captured richer surface spray details.

![Figure 7 Water column falling into a dragon-shaped obstacle (top: ASLSPH; bottom: IISPH).](image)

As shown in Figure 8, we simulated the animation result of a bunny-shaped fluid falling into a water tank. The simulation results of ASLSPH and DFSPH are presented in the top and bottom rows, respectively. For the aforementioned reason, the ASLSPH method dynamically adjusted the smoothing length of the particles near the free surface. Initially, DFSPH captured rich fluid details because of its ability to maintain the incompressibility of fluids. However, this capability decreased over time, and DFSPH did not maintain sufficient details (in contrast to ASLSPH), as shown in the last column of Figure 8.

![Figure 8 Bunny-shaped fluid falling into a water tank (top: ASLSPH; bottom: DFSPH).](image)

### 4.4 Simulation efficiency

To evaluate the time efficiencies of the different methods, we simulated the 3D dam-break effect shown in Figure 5 with two different resolutions (particle numbers of 9360 and 89216). Table 1 presents the performance measurements for the first 15s, where \( \text{Num} \) represents the particle number, \( \Delta t \) represents the timestep, \( T_{\text{sim}} \) represents the total simulation time, \( T_{\text{ns}} \) represents the time cost for the neighbor search, and \( \text{Speed-up} \) represents the acceleration ratio of each method to WCSPH.

Although the calculation times of PCISPH, IISPH, and DFSPH in a single timestep were longer than that
of WCSPH owing to the iterative calculation of the pressure, larger timesteps could be used (32.5, 37.5, and 37.5, respectively, times that of WCSPH at the low resolution). Consequently, the overall acceleration ratio reached 15.72, 20.35, and 40.87, respectively, in the low-resolution simulation. Owing to the high interpolation accuracy and stability, ASLSPH can also adopt a large timestep. Additionally, similar to WCSPH, ASLSPH does not need to solve the pressure iteratively. ASLSPH achieved an exceedingly high computational efficiency, reaching an acceleration ratio of 48.63 times that of WCSPH. In the high-resolution simulation, the acceleration ratios of PCISPH, IISPH, DFSPH, and ASLSPH were 16.91, 23.18, 40.58, and 44.96 times larger, respectively, than that of WCSPH.

ASLSPH did not perform well with regard to the time required for the neighbor search, even though we used a scheme (see Section 3.2) for acceleration. In the low-resolution mode, the neighbor search times of the five methods accounted for 28.95%, 27.78%, 28.19%, 28.32%, and 33.30% of the total time, respectively (calculated as $T_w/T_{tot}$). The proportion was largest for ASLSPH. In the high-resolution mode, as the number of particles increased, ASLSPH’s neighbor search became even more time-consuming, accounting for 36.46% of the total time. This is because the size of the auxiliary grid depends on the maximum smoothing length of the particles, and the efficiency of the neighbor search method in ASLSPH decreases when the smoothing length varies over a large range. Therefore, enhancing the performance of the neighbor search scheme can improve the overall simulation efficiency of ASLSPH.

## 5 Conclusion and future work

An ASLSPH method is proposed to improve the accuracy of interpolation kernels in SPH by adjusting the smoothing length of particles adaptively. An effective neighboring particle search scheme and a symmetric kernel function are used to increase the time efficiency and maintain the stability of the algorithm, respectively. Thus, a larger timestep can be used in ASLSPH to improve the simulation efficiency. Compared with other state-of-the-art methods, ASLSPH is competitive with regard to both the visual quality and the simulation efficiency.

The ASLSPH method also has shortcomings. When the smoothing length of the particles varies widely, the efficiency of the grid-assisted neighborhood search scheme decreases. In the future, we will investigate better neighbor search schemes to further improve the simulation efficiency. Additionally, we are planning to transplant ASLSPH to a GPU for a large-scale fluid simulation.

**Declaration of competing interest**

We declare that we have no conflict of interest.
References


3 Koschier D, Bender J, Solenthaler B. Smoothed particle hydrodynamics techniques for the physics based simulation of fluids and solids. Eurographics, 2019 DOI: 10.2312/egt.20191035


16 Bender J, Koschier D. Divergence-free smoothed particle hydrodynamics. In: Proceedings of the 14th ACM
A point-based method for animating incompressible flow
Hodgins J K Bargteil A W 2009
Proceedings of the

A parallel SPH implementation on multi-core CPUs
Becker M

Smoothed particle hydrodynamics on GPUs
Kawaguchi Y

A dual-support smoothed particle hydrodynamics for weakly compressible fluid
Olejnik M Pozorski J Szewc K Donoso A Daza J Di L Condensed Sigalotti G

Kernel estimates as a basis for general particle methods in hydrodynamics
Monaghan J J

Time adaptive approximate SPH
Goswami P Pajarola R

Temporal blending for adaptive SPH
Kolb A

Adaptive sampling and rendering of fluids on the GPU
Zhang Y C Solenthaler B Pajarola R

Multiresolution particle-based fluids
Dutré P Basel M H

Adaptive particle distribution for smoothed particle hydrodynamics
Quinlan N Bender J


Virtual Reality & Intelligent Hardware 2010 New Orleans, New York


The Eurographics Association 2010


DOI: 10.1111/j.1467-8659.2012.03186.x


DOI: 10.1007/978-3-7091-7486-9_5

Ihmsen M, Akinci N, Gissler M. Boundary handling and adaptive time-stepping for PCISPH. The Eurographics Association, 2010

DOI: 10.2312/PE/vriphys/vriphys10/079-088


DOI: 10.2312/PE/vriphys/vriphys11/019-028


DOI: 10.1016/0021-9991(82)90025-0


DOI: 10.1093/mnras/270.1.1


DOI: 10.5488/cmp.9.2.359


DOI: 10.1016/j.jcp.2017.07.023


DOI: 10.32604/cmes.2019.05146


DOI: 10.1142/S0219876207000972


DOI: 10.1111/j.1467-8659.2010.01832.x

ACM Press, 2009, 247–255
DOI:10.1145/1599470.1599502
DOI:10.1086/191344
DOI: 10.3969/j.issn.1001-246X.2016.02.008
DOI: 10.2312/SCA/SCA10/055-064
DOI:10.1145/2461912.2461984
DOI:10.1145/3099564.3099580
DOI:10.3389/fenrg.2020.00086
DOI:10.1007/s40571-020-00347-0