

Research Article

A Coupled Grid-Particle Method for Fluid Animation on GPU

Fengquan Zhang^{1,2}, Qiuming Wei,¹ and Zhaohui Wu^{2,3}

¹School of Information Science and Technology, North China University of Technology, Beijing, China

²State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China

³China Academy of Transportation Sciences, Beijing, China

Correspondence should be addressed to Fengquan Zhang; fqzhang@ncut.edu.cn

Received 18 March 2020; Accepted 6 May 2020; Published 23 May 2020

Academic Editor: Huimin Lu

Copyright © 2020 Fengquan Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In digital production environments, high-quality visual effects play a key role in our mobile device such as game and film. The simulation of fluid animation with free surface is an important area in computer graphic. However, the tracking of fluid surface is a challenging problem because of its instability. In this paper, a coupled grid-particle method for fluid animation surface tracking and detail preserving is proposed. Firstly, based on the nonequilibrium extrapolation method, we design a novel method for reconstructing distribution functions (DFs) of interface grids of lattice Boltzmann method (LBM) and couple the reconstruction method with LBM and volume of fluid (VOF) to track the free surface, which can obtain the accurate surface. Secondly, in order to avoid the loss of details caused by weaknesses in the traditional LBM-VOF method, we design a coupled grid-particle method that not only makes full use of the advantages of the coupled grid-particle method but also realizes the two-way coupling between grid method and particle method. Furthermore, for achieving the real-time requirements of fluid animation, we use GPU parallel computing to accelerate the simulation and use an improved screen space fluid (SSF) rendering method for realistic rendering. The various experiments show that this work can track the fluid surface with high precision and preserve the details of the fluid surface, and it also achieves good real-time performance in large-scale fluid simulation.

1. Introduction

Small-scale features, such as fluid drop, splash, and ripple, show interesting details in realistic physical simulation based on fluid animation. But how to track and preserve them from the numerical model is a challenging problem in computer graphics. In addition, due to the difficulty and high cost of animation production, especially of large-scale natural phenomena, traditional methods are gradually replaced or complemented by virtual reality technology. As one of the important areas in virtual reality, fluid simulation has made remarkable achievements in animation production.

Wen and Ma [1] presented a vorticity preserving lattice Boltzmann method (VPLBM) to simulate high-resolution motion of smoke in real time. Stomakhin et al. [2] obtained the fluid animation based on the material point method, and it was used in the movie *Frozen*; and Jiang et al. [3] have developed the affine particle in cell method to simulate the animation of water, which has also been used in many

movies. The free surface of fluid tracking plays a key technique in many animation simulations. However, the free surface is often limited by the accuracy of solving the advection equation, and errors occur in the visual effect of the animation. In view of the above problems, there are two mainstream solutions: one is to radically solve the problem, that is, to improve the accuracy of solving the advection equation, just like the least-squares volume of fluid (VOF) interface reconstruction algorithm [4]. The other method to solve the problem from visual effect is the coupled grid-particle method. The principle of coupled grid-particle method is to use the grid method with high accuracy to solve the velocity field to evolve the main fluid and then use the particle level set method (PLSM) to track the free surface. The particles beyond the free surface are evolved by the particle method with more details, and the grids and particles are coupled on the free surface [5]. The first solution to improve the accuracy of solving advection equation solution can only reduce but not avoid the error of solving the advection equation.

The second solution to solve the problem from visual effect treats the particles as separate details from the free surface and then uses the particle method to render, focusing on increasing the surface details of fluid. In addition, the accuracy of surface simulation is closely related to the number of particles in PLSM. But too many particles will cause a serious computational burden. Therefore, reasonable preserving and enhancing of the details of the fluid surface become the key to improving the fidelity of visual effects.

This paper focuses on solving the weakness problems using the grid method to improve the fidelity of visual effect. Based on the LBM-VOF method [6], this paper proposes a coupled grid-particle method which can solve the problem of low accuracy of reconstructing DFs of interface grids and abnormal interface grid in the LBM-VOF method. And the contributions of this article are as follows:

- (1) This paper proposes a new method for reconstructing DFs of interface grids and couple the reconstruction method with LBM-VOF. This method enhances the accuracy of the reconstruction, to achieve the purpose of preserving the fluid surface detail
- (2) Aiming at the problem of anomalous interface grids, this paper proposes a coupled grid-particle method. This method replaces abnormal interface grids with particles and uses the advantage of the particle method to express details to enhance the details of fluid surface. And particles and grids are coupled through a two-way coupling model
- (3) For real-time fluid simulation, this paper designs high-performance simulation algorithms and rendering algorithms on GPU

The rest of this paper is organized as follows. After briefly reviewing the related work in Section 2, Section 3 introduces our coupled grid-particle method for fluid animation and describes our improved rendering method on GPU. Finally, experimental results are analyzed in Section 4, and the conclusion and future work are outlined in Section 5.

2. Related Work

In recent years, the simulation of free surface fluid based on LBM has attracted some research interests. Körner et al. [6] proposed the LBM-VOF coupled algorithm for simulating the free surface of the water. However, the accuracy of solving the advection equation is low and the mass exchange error of the interface grid is large, which leads to the suspension of the abnormal interface grid. Thürey and Rude [7] compared the LBM-VOF coupled algorithm and the coupled algorithm of LBM and the level set method. Their experiment shows that the LBM-VOF coupled algorithm has higher numerical accuracy. Based on the LBM-VOF algorithm, Thürey [8] proposed a method of artificial intervention to solve the suspension problem. Although this method solves the problem of suspension, it leads to mass nonconservation and the evolution of fluid can be greatly interfered with by a human. Aiming at solving the problem of surface reconstruction in

LBM-VOF, Janssen and Krafczyk [9] adopted a higher accuracy piecewise linear interface reconstruction (PLIC) method, which fundamentally improved the solution accuracy of the advection equation. However, this method can only improve the accuracy to a certain extent, and it cannot avoid the abnormal interface suspension problem in [8]. Janssen and Krafczyk [10] took advantage of the high parallelism of LBM and implemented the LBM-VOF algorithm on GPGPU, focusing on alleviating the computational burden of fluid simulation through a parallel computing acceleration algorithm. So far, the LBM-VOF method is still the most common free surface tracking algorithm used by studies in LBM, which this paper is focusing on for its improvement. But, the problem of abnormal interface suspension in the LBM-VOF method is urgently needed, but no suitable solution has been proposed yet.

Other than VOF, the LBM method can also be coupled with other surface tracking methods. Zheng et al. [11] built a higher accuracy interface capturing equation and proposed a two-dimensional surface tracking method based on LBM. This method does not require interface reconstruction as needed by the LBM-VOF method, but it costs more computation time than the LBM-VOF method. Kaneda et al. [12] used the coupled level set with VOF (CLSVOF) method to track the two-dimensional free surface of LBM and proposed the LB-CLSVOF model. Although the volume error is reduced, the visual effect still has the problem of interface grid suspension. Wang et al. [5] used PLSM to track the surface of fluid animation based on LBM and SPH to evolve some separated particles. This method coupled LBM and SPH on the free surface. In this method, the particles are regarded as details of the fluid that are separated from the free surface, which are used to increase the free surface details of the fluid animation. In addition, the accuracy of the free surface is closely related to the number of particles in PLSM, and too many particles will cause a serious computational burden.

For detail preserving fluid animation, the research work is relatively sparse. Zhang et al. [13, 14] presented a real-time fluid simulation tool based on the particle method, which can obtain vivid fluid animation. Therefore, in order to obtain realistic fluid animation, efficient detail preserving methods and realistic rendering methods are very important. For particle-based fluid animation, most of the research work focuses on avoiding the loss of details in fluid deformation, such as [15–17]. Wang et al. [18] proposed a new method combined with the Marching Cubes and free surface algorithm to extract the fluid surface and used adaptive surface tension combining the wave-particle equation to show the details of fluid surface. For grid-based fluid animation, Thürey et al. [19] also focuses on fluid control technique instead of detail preserving of fluid surface and proposes a new fluid control technique that used scale-dependent force control to preserve small-scale fluid detail. In addition, some neural network methods such as [20–22] are used in fluid animation recently, which can get fine fluid details.

3. Proposed Method

3.1. The Framework of our Method. As shown in Figure 1, the framework of our method is divided into four parts: LBM,

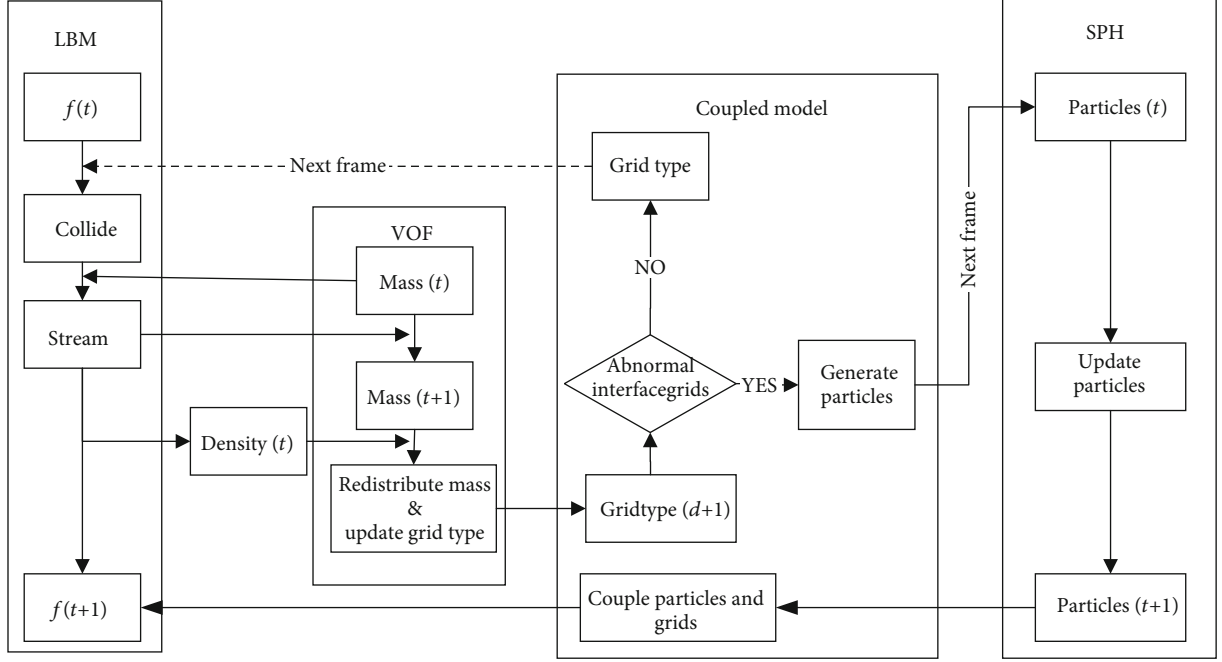


FIGURE 1: The computational pipeline in our method.

VOF, the coupled model, and SPH. The main part of the fluid adopts the grid-based Lattice Boltzmann method, which can fully utilize the advantages of the grid method in solving the fluid velocity field with high accuracy. In LBM's stream step, a new reconstruction method with second-order accuracy for DFs of interface grid is proposed, which can effectively preserve fluid details. Next, the mass and density between cells are calculated in the stream step, and according to the calculated mass and density, free surface tracking is performed by the VOF method. Then, anomalous interface grids are transformed into particles and particles are coupled with the non-anomalous interface grids through our coupled model. Finally, the particles evolve through the SPH method and use the advantage of the SPH method to express details to enhance surface details.

A new reconstruction method for DFs of interface grid proposed for the LBM-VOF method, but our work is mainly focused on the coupled model. There are three steps in our proposed model. Firstly, the abnormal interface grids in the fluid surface tracked by VOF are extracted while grid types of the nonanomalous interface grids will be used in the next frame. Secondly, the abnormal interface grids are converted to particles that will participate in the evolution in the next frame. Finally, the updated particles of this frame are coupled with the nonanomalous interface grids of the current frame.

Our method is to deal with the abnormal grid generated by the LBM-VOF method through SPH, to achieve detail preserving and enhancing. Moreover, our method can take advantage of the SPH method to depict and enrich the details of fluid surface. In short, our approach combines the advantage of the LBM method which is good at solving the velocity field and the advantage of SPH which is good at depicting surface details.

3.2. LBM-VOF Method for Free Surface Tracking. For free surface tracking, Körner et al. [6] proposed the LBM-VOF coupled algorithm. The fluid solver uses the grid-based LBM, which described the fluid with cellular automation to compute the transport equations. Each cell stores a discrete number of velocities particle. The Bhatnagar-Gross-Krook (BGK) model is applied widely to solve collisions between cells [23], and it can be formulated as

$$f_i(x + e_i, t + \delta_t) - f_i(x, t) = -\frac{1}{\tau} \cdot (f_i(x, t) - f_i^{(eq)}(x, t)), \quad (1)$$

where e_i is the discrete cell velocity in the direction i , τ is the relaxation time, and $f_i(x, t)$ and $f_i^{(eq)}(x, t)$ are the density DF and the equilibrium density DF in the cell at x at time t , respectively. In 2D simulations, nine directions are usually used in the D2Q9 model, while for 3D simulation, the D3Q19 model with nineteen velocities is the most common one. In our work, we use the velocity vectors as shown in Figure 2.

This paper uses the D3Q19 model and the equilibrium DF, $f_i^{(eq)}(x, t)$, which can be obtained with

$$f_i^{(eq)}(x, t) = \omega_i \rho \left[1 + 3e_i \cdot u + \frac{9}{2}(e_i \cdot u)^2 - \frac{3}{2}(u \cdot u) \right], \quad (2)$$

where $\omega_i = 1/3$ for $i = 0$, $\omega_i = 1/18$ for $i = 1..6$, $\omega_i = 1/36$ for $i = 7..18$, and ρ and u are the density and the velocity of each discrete spatial cell, respectively; they can be obtained by the summation of all DFs for one cell:

$$\rho = \sum_i f_i(x, t), \quad (3)$$

$$\rho u = \sum_i f_i(x, t) \cdot e_i. \quad (4)$$

After the values computed with Equation (1) are stored at DFs for time $t + \delta_t$ (i.e., collide step), each discrete spatial cell needs the DFs of the adjacent cells from the previous time step (i.e., stream step). The collide step and stream step are repeated, and the fluid flows. Additionally, the external forces can be applied in the collide step. This paper uses the GZS model which is of benefit in designing LBM models for fluids exposed to external and internal forces [24]. For boundary conditions, it can be implemented simply by the no-slip boundary condition or the idea extrapolation of the nonequilibrium which has second-order accuracy.

The next step is to update the grid type; the VOF method is used, which is based on tracking the mass of fluid throughout the computational cell. Compared with the level set method, VOF has higher numerical accuracy. Generally, the cells are divided into three types. The cells without fluid are defined as gas phase (G), the partially filled cells are defined as interface (I), and the cells completely are filled as fluid (F). All these cells form a closed layer, as shown in Figure 3.

In other words, “F” cannot be adjacent to “G” directly and only “I” can change to “G” or “F.” The interface cells are updated by tracking mass exchange using all neighboring fluid cells and interface cells; the mass exchange is obtained by subtracting the outgoing DFs from the incoming ones during the streaming step:

$$\Delta m_i = \varepsilon \cdot (f_{\text{in}} - f_{\text{out}}) \quad (5)$$

$$\varepsilon = \frac{(\varepsilon_1 + \varepsilon_2)}{2}$$

Here, Δm_i denotes the exchange quantity of mass in direction i . f_{in} and f_{out} are the incoming DF and outgoing DF, while ε_1 and ε_2 are the fraction of the cells that are filled with fluid. As Figure 4 shows, the gas phase does not participate in evolution, so the DFs of “G” are unknown, and its DFs need to be reconstructed.

Körner et al. [6] provided a reconstruction method based on the momentum exchange:

$$\hat{f}_i(x + e_i \delta_t, t + \delta_t) = \hat{f}_i^{(\text{eq})}(\rho_A, u) + \hat{f}_i^{(\text{eq})}(\rho_A, u) - f_i(x, t), \quad (6)$$

where $\hat{f}_i^{(\text{eq})}(\rho, u)$ is the DF opposite to $f_i^{(\text{eq})}(\rho_A, u)$, $\hat{f}_i(x + e_i \delta_t, t + \delta_t)$ is the reconstructed DF of an interface cell which comes from a gas cell, u is the velocity of the interface cell, and ρ_A is an atmospheric pressure parameter. To balance the forces on each side of the interface, the DFs coming from the normal direction of the interface are also reconstructed.

However, this reconstruction method is not accurate. Based on the nonequilibrium extrapolation method, we design a novel method for reconstructing the DFs of the interface grids. The analysis and theory are detailed in Section 3.3.

3.3. A New DF Reconstruction Method for Preserving Surface Detail. The LBM-VOF method for surface tracking has been

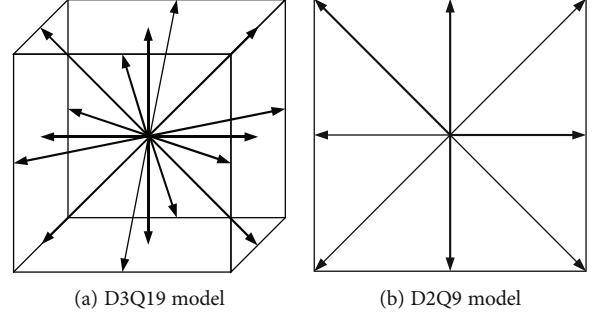


FIGURE 2: The model of velocity vectors.

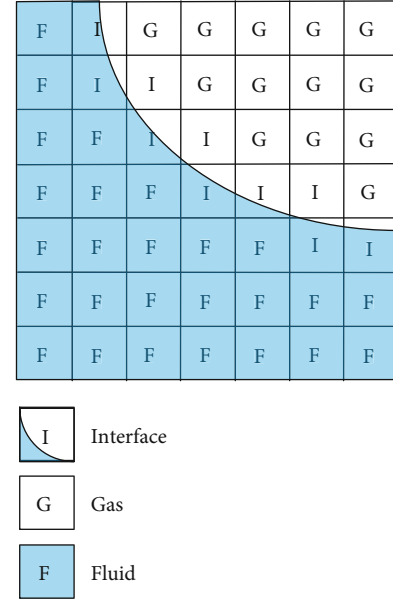


FIGURE 3: The different grid types required for the free surface.

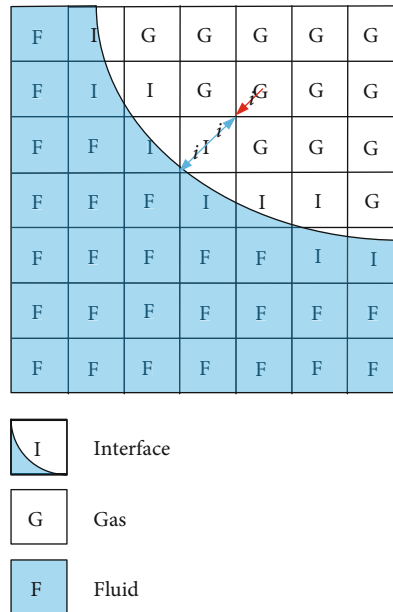


FIGURE 4: The DFs from the gas phases are unknown.

detailed in Section 3.2, but the poor accuracy of the reconstructed DFs can lead to some artifacts. A typical artifact is that the fluid is initially symmetric, but after a period of evolution, it is clearly asymmetric, which shows that the reconstruction error of DFs is still large.

Guo et al. [25] proposed a nonequilibrium extrapolation method to deal with solid boundaries, whose basic principle is that according to Chapman-Enskog expansion, the DF can be expressed as the sum of the equilibrium function and the nonequilibrium function so that the approximate values can be solved by using the information of adjacent lattice points. This method can be further extended to reconstructing DFs of the interface grids, as shown in Equations (7)–(9):

$$f_{i(\text{Gas})} = f_{i(\text{Gas})}^{(\text{eq})} + \frac{1}{\tau} \cdot f_{i(\text{Gas})}^{(\text{neq})}, \quad (7)$$

$$f_{i(\text{Gas})}^{(\text{eq})} = f_i^{(\text{eq})}(\rho_A, u), \quad (8)$$

$$f_{i(\text{Gas})}^{(\text{neq})} = f_{i(\text{Interface})}^{(\text{neq})} - f_i^{(\text{eq})}(\rho_{\text{Interface}}, u), \quad (9)$$

where the interface grid is in direction i of the gas grid; $f_{i(\text{Gas})}$ is the DF of gas in the direction of i ; $f_{i(\text{Gas})}^{(\text{eq})}$ is the equilibrium function of $f_{i(\text{Gas})}$; τ is the relaxation time; $f_{i(\text{Gas})}^{(\text{neq})}$ is the non-equilibrium function of $f_{i(\text{Gas})}$; ρ_A is an atmospheric pressure parameter, $\rho_{\text{Interface}}$ and u are density and velocity of the gas grid, respectively; $f_{i(\text{Interface})}$ is the DF of the interface grid in direction i ; and $f_i^{(\text{eq})}(\rho_{\text{Interface}}, u)$ is the equilibrium function of the interface grid in direction i which is calculated by $\rho_{\text{Interface}}$ and u .

In the case of low speed, the density and speed of adjacent grids are similar, so the transfer of fluid DF can be considered the transfer of nonequilibrium function. According to Equation (3) and Equation (7), it is obvious that

$$\sum_i f_i^{(\text{neq})} = 0. \quad (10)$$

If only considering direction i and direction \tilde{i} , we can obtain

$$f_i^{(\text{neq})} + f_{\tilde{i}}^{(\text{neq})} = 0. \quad (11)$$

According to Equation (7), Equation (8), Equation (9), and Equation (11), we can obtain that

$$f_{i(\text{Gas})} = f_i^{(\text{eq})}(\rho_A, u) + \frac{1}{\tau} \cdot \left(f_{\tilde{i}}^{(\text{eq})}(\rho_{\text{Interface}}, u) - f_{i(\text{Interface})}^{(\text{eq})} \right). \quad (12)$$

In this paper, we use Equation (12) instead of Equation (6) to reconstruct DFs of interface grids, as balancing the forces on each side of the interface is no longer needed. We combine this method with LBM-VOF and develop a surface tracking algorithm with higher accuracy than [6, 8].

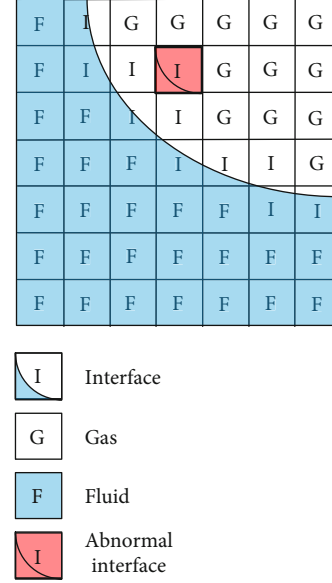


FIGURE 5: The abnormal interface grid.

3.4. A Coupled Grid-Particle Method for Enhancing Fluid Detail. For free surface tracing, the key is to solve the advection equation, which can be expressed as

$$\frac{\partial \phi}{\partial t} = -u \cdot \nabla \phi. \quad (13)$$

Here, the left side of the formula represents the increment of fluid per unit time, and the right side of the formula represents the amount of fluid inflow per unit time.

The advection equation solution directly determines the accuracy of the free surface of the fluid animation. The key to solving the advection equation is to determine the interface within the grid, which is difficult to solve with high accuracy. As mentioned in Section 3.2, due to the difficulty of accurate determination of the interface within the grid, both the advection equation solution and the mass exchange are inaccurate, resulting in the abnormal suspension of the cells. Typically, when most fluid regions are in equilibrium, there are still many cells suspended in the air. This paper looks at this problem from the perspective of visual effect, that is, solving the cell suspension problem, to provide better visual effects.

Different from [8], this paper does not fully rely on manual intervention but uses a coupled method: the main part of the fluid animation uses LBM with high accuracy to solve the velocity field, and the abnormal grids use SPH to evolve. On the premise of following the physical laws, it can not only solve the cell suspension problem but can also enrich the surface details through particles to a certain extent. This method can be divided into two parts: the generation and evolution of particles and the coupling of LBM and SPH.

3.4.1. The Generation and Evolution of Particles. Based on the LBM-VOF method, the interface grids must be adjacent to the fluid grids. However, as shown in Figure 5, due to the

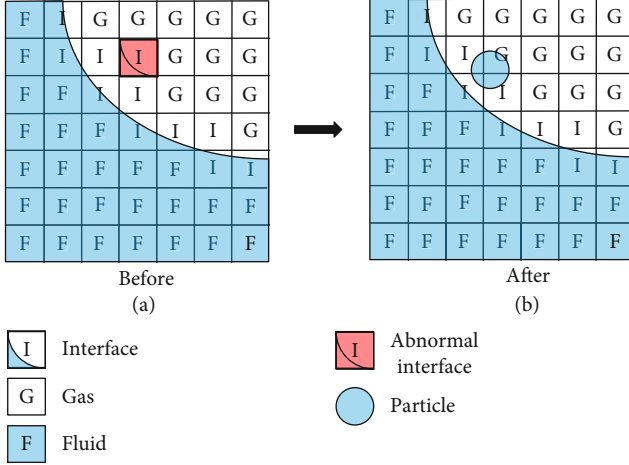


FIGURE 6: Schematic diagram of particle replacing the abnormal grid: (a) before; (b) after.

low accuracy of mass exchange between interface grids, some interface grids are not adjacent to the fluid grid, and these grids are abnormal interface grids.

To solve the problem of abnormal grids and make abnormal grids flow reasonably, as shown in Figure 6, this paper uses particles instead of these abnormal grids to evolve. There are three steps to generate particles: firstly, generate a particle at the position $\text{pos}_{\text{particle}}$ that is obtained based on the position of the interface grid. Secondly, the physical quantities (mass $\text{mass}_{\text{particle}}$, velocity v_{particle} , and density ρ_{particle}) of the particles are calculated according to the physical quantities of the abnormal interface grids, and then, the abnormal interface grids are improved. Finally, in the next frame, the particles evolve in the method of SPH, while the grids evolve with the improved LBM-VOF. The equations used in the three steps are shown as follows:

$$\text{pos}_{\text{particle}} = \text{pos}_{\text{grid}} - (1 - \varepsilon) \cdot n, \quad (14)$$

$$\text{mass}_{\text{particle}} = \text{mass}_{\text{grid}}, \quad (15)$$

$$v_{\text{particle}} = v_{\text{grid}}, \quad (16)$$

$$\rho_{\text{particle}} = \varepsilon \cdot m_{\text{grid}}, \quad (17)$$

where pos_{grid} is the position of the abnormal interface grid, ε is the ratio of mass to the density of the abnormal interface grid, and n is the normal vector of the abnormal interface grid. $\text{mass}_{\text{grid}}$, v_{grid} , and ρ_{grid} represent mass, velocity, and density of the abnormal interface grid, respectively.

The SPH method is implemented based on [13, 14]. The density ρ_i , pressure ∇p_i , and viscous force $\mu \nabla \cdot \nabla \mathbf{u}$ of the particles can be calculated by the following three equations:

$$\begin{aligned} \rho_i &= \sum_{j=1}^n m_j W(x_i - x_j, h), \quad i \neq j, \\ \nabla p_i &= -\rho_i \sum_{j=1}^n \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) m_j \nabla W(x_i - x_j, h), \quad i \neq j, \\ \mu \nabla \cdot \nabla \mathbf{u} &= \mu \sum_{j=1}^n (\mathbf{u}_j - \mathbf{u}_i) \frac{m_j}{\rho_j} \nabla^2 W(x_i - x_j, h), \quad i \neq j, \end{aligned} \quad (18)$$

where h and $W(x_i - x_j, h)$ are the radius of the smooth kernel and the smooth kernel function of the SPH method, respectively. Specifically, the Poly6 smooth kernel is used in this paper.

3.4.2. The Coupling of LBM and SPH. After dealing with the abnormal grids, the previously evolved particles may coincide with the interface grids, and the physical quantities of the particles need to be transferred to the interface grids in the coincided part. There are six steps in this part:

- (1) Calculate the position of the grid pos_{grid} based on where the particle is, defined in the following equation:

$$\text{pos}_{\text{grid}} = \lfloor \text{pos}_{\text{particle}} \rfloor \quad (19)$$

Here, $\text{pos}_{\text{particle}}$ is position of the particle.

- (2) As shown in Figure 7, if the grid type of pos_{grid} is interface, the particle needs to be coupled to the grid and execute step (3):

$$\begin{cases} \varepsilon_{\text{interface}} + k \cdot \varepsilon_{\text{particle}} \geq 1 \longrightarrow \text{need coupling,} \\ \varepsilon_{\text{interface}} + \varepsilon_{\text{particle}} < 1 \longrightarrow \text{no coupling required,} \end{cases}$$

$$k = \begin{cases} 1, & |\text{pos}_{\text{particle}} - \text{pos}_{\text{grid}}| \leq 0.5, \\ 1 - (|\text{pos}_{\text{particle}} - \text{pos}_{\text{grid}}| - 0.5), & \text{otherwise,} \end{cases} \quad (20)$$

where $\varepsilon_{\text{interface}}$ and $\varepsilon_{\text{particle}}$ are volume integrals of the interface grid and particle, respectively

- (3) Calculate the new mass of the interface grid m'_{grid} by Equation (21).

$$m'_{\text{grid}} = m_{\text{particle}} + m_{\text{grid}}, \quad (21)$$

where m_{particle} is the mass of the particle and m_{grid} is the mass of the grid

- (4) According to the law of conservation of momentum, the new velocity v'_{grid} and density ρ'_{grid} of the

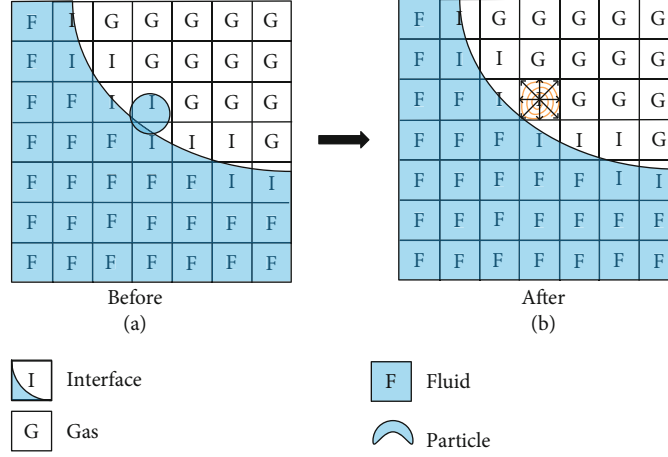


FIGURE 7: Schematic diagram of coupling of grid and particle: (a) before; (b) after.

```

Update physical information of particles based on SPH
Update physical information of grids based on improved LBM-VOF
for all grid  $\in$  interface grids do
  if grid  $\in$  abnormal grids then
    Generate particle at position  $pos_{particle}$  by using Eq.(14)
    Calculate physical information for the particle by using Eq.(15) - Eq.(17)
    Delete the abnormal grid
  end if
end for
for all particle  $\in$  SPH do
  Calculate the grid position  $pos_{grid}$  of the particle by using Eq.(19)
  if  $pos_{grid} \in$  interface grid positions then
    Calculate physical information for the grid by using Eq.(21) - Eq.(23)
    Recalculate DF of the grid by using Eq.(1)
    Delete the particle
  end if
end for

```

ALGORITHM 1: Coupled grid-particle method for the free surface.

coupled grid can be calculated by Equation (22)–Equation (23):

$$v'_{grid} = (m_{particle} v_{particle} + m_{grid} v_{grid}) / m'_{grid}, \quad (22)$$

$$\frac{m'_{grid}}{\rho'_{grid}} = \frac{m_{particle}}{\rho_{particle}} + \frac{m_{grid}}{\rho_{grid}}, \quad (23)$$

where $v_{particle}$ and $\rho_{particle}$ are the velocity of the particle and v_{grid} and ρ_{grid} are the velocity and density of the old grid

- (5) Then, according to v'_{grid} and ρ'_{grid} obtained in step (4), the DF of the grid needs to be recalculated by Equation (1)
- (6) Delete the particle and repeat step (1)–step (6) until all particles are traversed

The above two parts describe the coupled particle-grid method in this paper. Algorithm 1 summarizes the detailed algorithm flow of this method. This method solves the grid suspension problem from the perspective of visual effect and also enriches the details of the fluid interface to a certain extent. The more detailed experimental analysis will be presented in Section 4.

3.5. Real-Time Fluid Surface Rendering. To meet the real-time requirements of fluid animation, this paper uses GPU to realize the real-time fluid animation simulation. First of all, GPU intuitively supports parallel computing to speed up the whole algorithm. Moreover, the traditional rendering method is abandoned, and the GPU-based screen space fluid (SSF) method is adopted, which does not involve the matching and reconstruction of the fluid surface mesh, and the implementation is based on GPU [26, 27].

In particular, SSF is proposed for particle-based fluid. For rendering grid-based fluid, we improve on the radius and the position of the sprite points. For the radius of the sprite point,

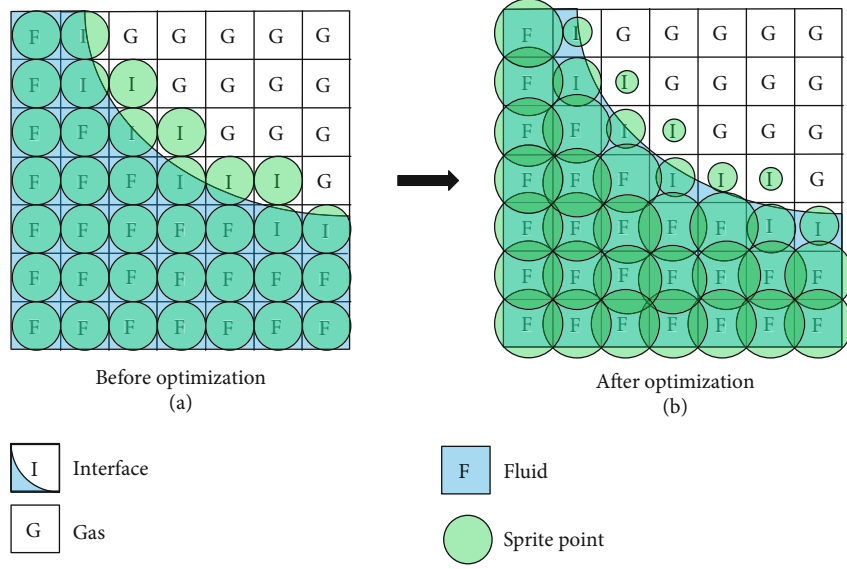


FIGURE 8: Size optimization of sprite points: (a) before optimization; (b) after optimization.

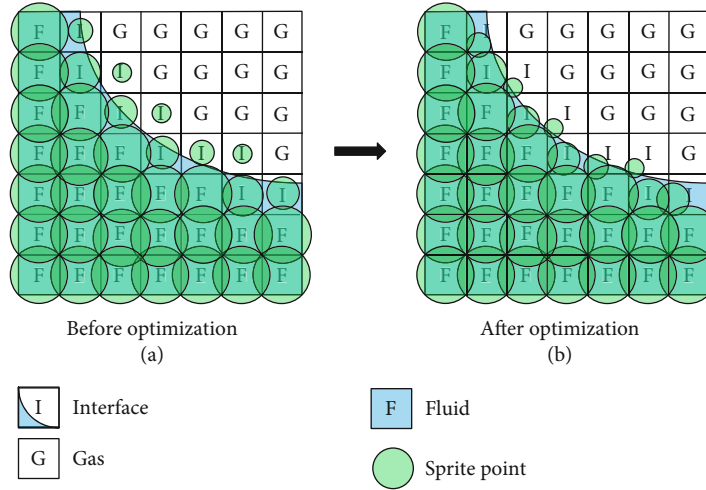


FIGURE 9: Position optimization of sprite points: (a) before optimization; (b) after optimization.

to make the transformation of the fluid more continuous, the radius of the sprite point is set to K times the density of the grid. K is $\sqrt{2}/2$ in two-dimensional space or $\sqrt{3}/2$ three-dimensional space. As shown in Figure 8, when a grid is filled, the size of the sprite point just covers the grid completely.

For the position of the sprite point, as shown in Figure 9(a), if the sprite points are drawn directly in the center of the grid, there will be obvious gaps between the interface sprite points and the fluid sprite points, which visually represent the discontinuity of fluid flow. For a continuous visual effect, we use Equation (14) to calculate the position of the interface sprite points, and the final schematic diagram is shown in Figure 9(b).

4. Experiment and Analysis

Our experiment is running on a PC with Intel® CPU i5-8300H and NVIDIA GTX 1060 graphics card. All parallel

algorithms are implemented by Compute Shader of DirectX, and all fluid rendering algorithms are implemented by CG language.

Our fluid parameter of $1/\tau$ is 1.85, and the fluid parameter of the acceleration of gravity is 0.005, which can be calculated by [8].

Figure 10 shows the visual effect comparison before and after the optimization of the screen space fluid (SSF) rendering method. Figure 10(a) illustrates the rendering before the optimization, the interface sprite points are separated from the fluid sprite points highlighted in the circle, showing an obvious gap between the interface sprite points and the fluid sprite points. There are even gaps between the fluid sprite points themselves. Figure 10(b) illustrates the rendering after the optimization, the interface sprite points are closer to the fluid sprite points, and the whole surface is smoother.

Figure 11 compares the DF reconstruction method based on [8] and our method at frame 215. Here, $1/\tau$ is 1.85, the

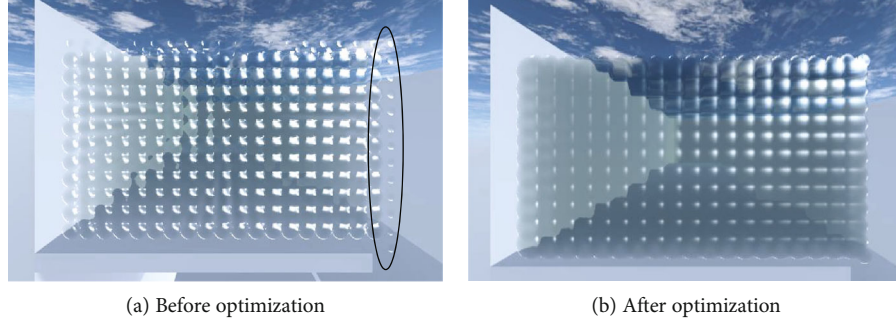


FIGURE 10: Comparison of before and after optimization of SSF.

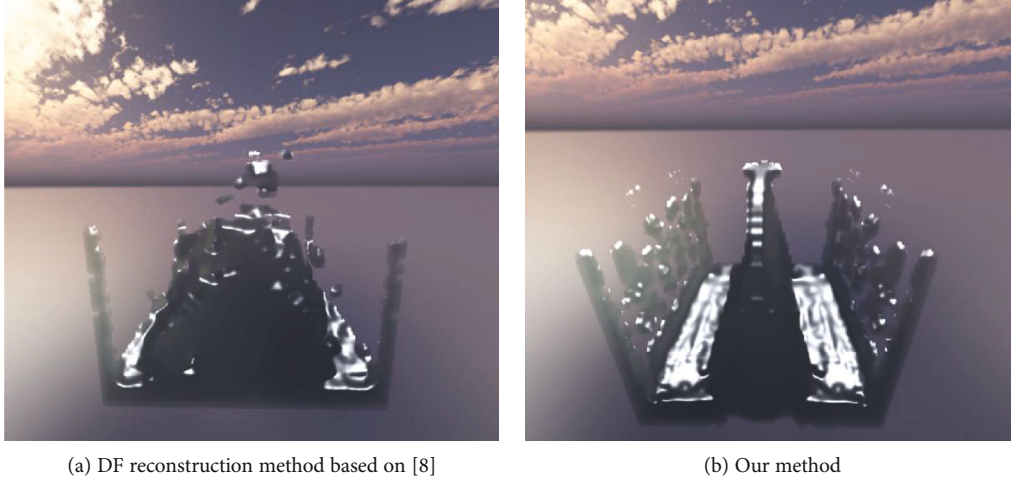


FIGURE 11: High-accuracy reconstruction method of the DFs of the interface grids.

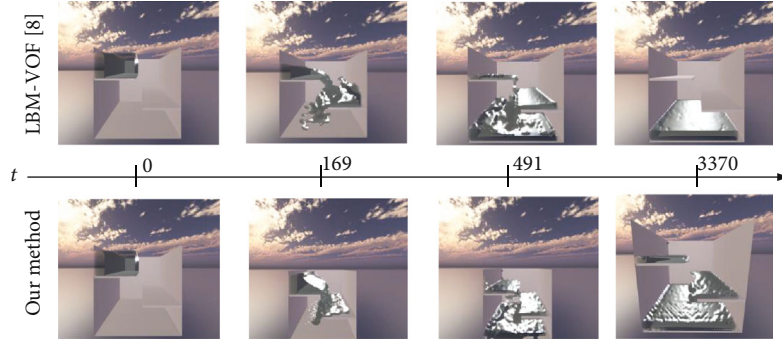


FIGURE 12: The comparison of the LBM-VOF method [8] and our grid-particle coupled method.

grid size is $40 \times 40 \times 40$, and the fluid parameter of the acceleration of gravity is 0.005 which can be calculated by [8]. Figure 11(a) shows that the fluid no longer remains symmetrical after a period of evolution, while Figure 11(b) shows that most of the fluids on the right and left are symmetrical. The experimental results show that our method has higher accuracy. However, since our method is developed based on the theory of the low-speed model, our method is only used for low-speed fluid simulation.

A comparative experiment is shown in Figure 12, rendered with a frame rate of 89.1 FPS. The initial condition ($t = 0$) of both methods is the same. Here, the parameters remain the same; we only change the initial positions of the

fluid and solid. After the fluids have evolved 169 time steps ($t = 169$), our method shows more interface details as shown in the Figure 12. At the time of $t = 491$, our method shows more interface details that are highlighted by the small circle. The large circle shows the physical phenomenon of the fluid interface affected by the surface tension. This phenomenon is benefited from the high-accuracy DF reconstruction method of interface grids in our method. Finally, when the fluid tends to be more stable ($t = 3370$), the animation based on the LBM-VOF method [8] does not leave any fluid on the obstacles due to the low accuracy of mass exchange and excessive manual intervention, while in our coupled method, there are many fluids left on the obstacles. Specially, our coupled

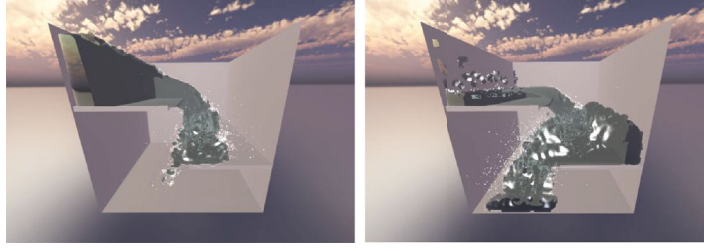


FIGURE 13: The scene of fluid flowing from the stair.

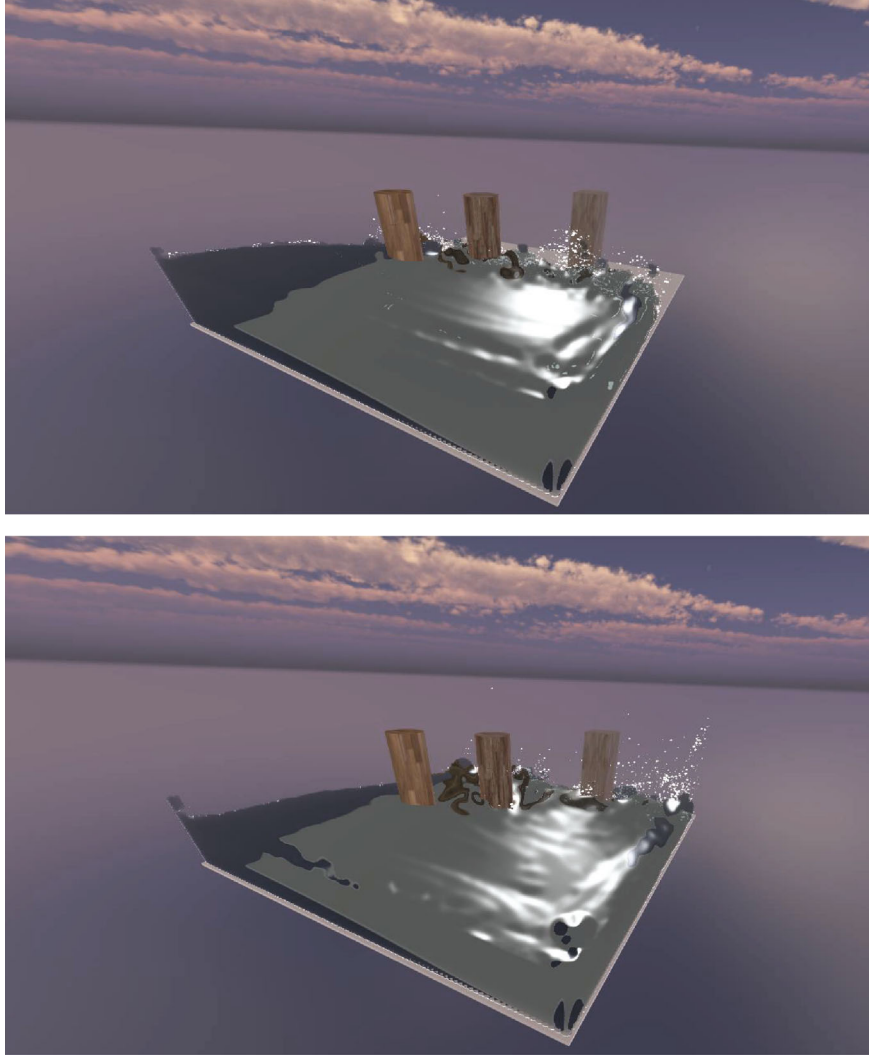


FIGURE 14: The scene of fluid-solid interaction animation with our method.

model is a simplified physical model, and numerical dissipation inevitably occurs during the numerical interpolation between particles and grids. Therefore, our method mainly solves visual problems and cannot be used for numerical simulation which has high requirements for numerical accuracy.

Figure 13 shows a large-scale fluid animation with an average frame rate of 29.2 FPS, and we increase the grid size to $100 \times 100 \times 100$. Here, it can be seen that our method can not only solve the problem of abnormal interface grid

suspension but can also enrich the surface details under the premise of mass conservation. Meanwhile, our method still has good real-time performance in large-scale fluid animation. Similar large-scale fluid simulations are also shown in Figure 14. The number of grid in Figure 13 is 148862, and 96628 in Figure 14.

Table 1 shows the performance of our method in different scenes, and grid size is $100 \times 100 \times 100$. The table shows not only the number of particles at different times but also the average number of frame rate. It can be seen from the

TABLE 1: The performance of our method in different scenes.

| Scene | The number of particles at $t = 100$ s | The number of particles at $t = 200$ s | The number of particles at $t = 400$ s | The number of particles at $t = 800$ s | The averaged time (ms) |
|--------------------------------------|--|--|--|--|------------------------|
| Fluid flowing from stair (Figure 13) | 3089 | 12163 | 23098 | 13225 | 30.645 |
| Fluid-solid interaction (Figure 14) | 585 | 5135 | 1635 | 786 | 27.279 |

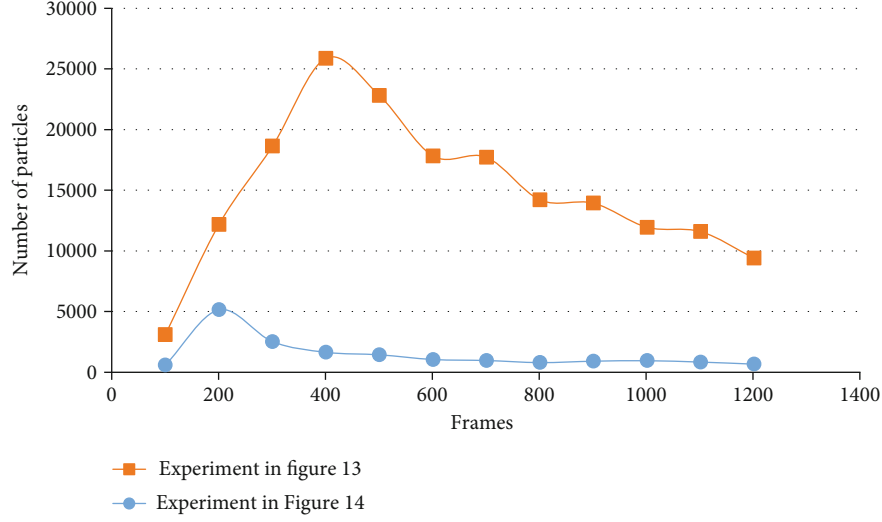


FIGURE 15: Graph of the change in the number of particles.

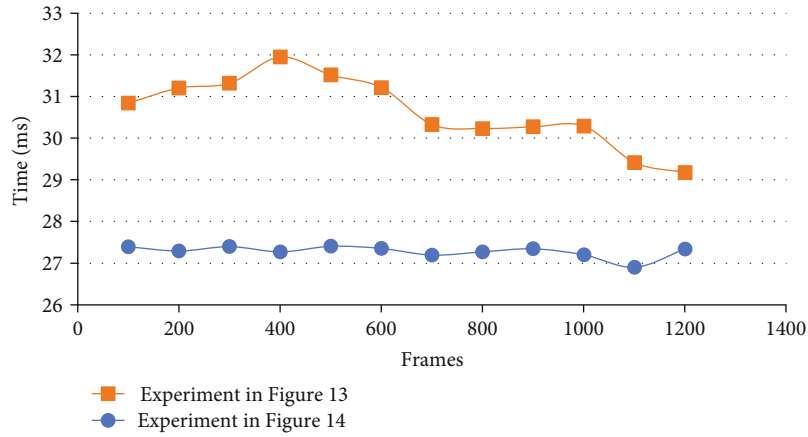


FIGURE 16: Real-time performance of fluid simulation.

table that the number of particles in Figure 13 is more than the number of particles in Figure 14 at different times, but their average time is about 30 ms. This shows that the number of particles in the optimized algorithm based on GPU does not significantly affect the frame rate of fluid simulation. In fact, the number of particles is orders of magnitude smaller than the number of grids. Finally, more detailed changes in the number of particles during the simulation are shown in Figure 15.

Figure 16 shows the real-time performance data of our GPU algorithm at different times. It is not seen that the performance fluctuation is small during the whole simulation

process, and it proves again that the influence of the number of particles on the performance is smaller than that of the fluid grids. And Figure 17 shows more detailed performance data. It can be seen the time in computing takes much more than the rendering. Although the algorithm has reached real-time simulation, there is still a lot of room for optimization in physical computing.

5. Conclusion and Future Work

In this paper, we obtain vivid fluid animation with our proposed method. We improve some key problems in the

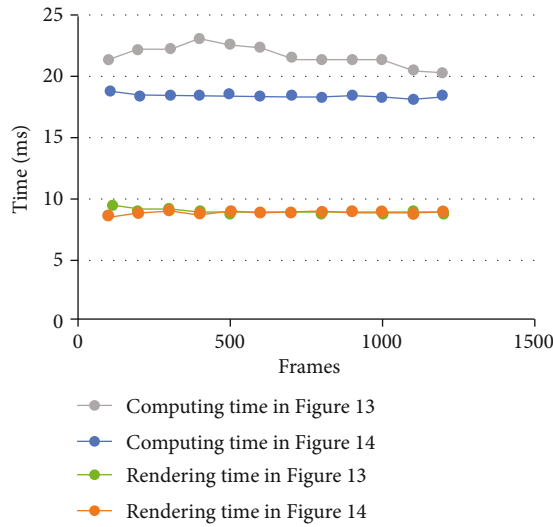


FIGURE 17: Computing and rendering performance of fluid simulation.

traditional LBM-VOF method. For the first problem of DF reconstruction of the interface grids, we adopted the nonequilibrium extrapolation method which has second-order accuracy to solve this problem and improved this method for reconstructing DFs of interface grids. For the second problem of the abnormal grid suspension caused by the inaccurate mass exchange of interface grids, we first removed the manual intervention which could lead to serious mass nonconservation, and then, we proposed a coupled grid-particle method which combined LBM, VOF, and SPH. Our coupled method not only solved the original problems but also provided enriching details of the fluid interface. Meanwhile, taking advantage of our method's high parallelism, we also used GPU parallel computing to speed up the algorithm and realized real-time rendering.

There are still limitations to overcome. Achieving high-accuracy coupling of grid and particle is still a difficult problem, for example, the transfer of physical properties. Furthermore, when the particles are adjacent to the interface grids, the force between the particles and the interface grids is not considered in this paper. In the future, we will continue to improve the simulation accuracy for more complex fluid animation by extending our method, such as multiple fluid interactions, multiphase flow, and fluid-solid coupling.

Data Availability

The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is supported by Beijing Natural Science Foundation (No. 4182018, No. 4154067); Beijing Social Science Foundation (No. 18YTC038); Humanities and Social Sciences Fund of the Ministry of Education (No. 19YJC760150); National Natural Science Foundation (No. 61402016); the open funding project of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (Nos. VRLAB2018A05 and VRLAB2020B10); Beijing Youth Talent Foundation (No. 2016000026833ZK09); and NCUT Foundation (No. XN018001).

References

- [1] J. Wen and H. Ma, "Real-time smoke simulation based on vorticity preserving lattice Boltzmann method," *The Visual Computer*, vol. 35, no. 9, pp. 1279–1292, 2019.
- [2] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle, "A material point method for snow simulation," *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 1–10, 2013.
- [3] C. Jiang, C. Schroeder, A. Selle, J. Teran, and A. Stomakhin, "The affine particle-in-cell method," *ACM Transactions on Graphics*, vol. 34, no. 4, pp. 1–10, 2015.
- [4] E. Aulisa, S. Manservigi, R. Scardovelli, and S. Zaleski, "Interface reconstruction with least-squares fit and split advection in three-dimensional Cartesian geometry," *Journal of Computational Physics*, vol. 225, no. 2, pp. 2301–2319, 2007.
- [5] C.-B. Wang, Q. Zhang, F.-l. Kong, and H. Qin, "Hybrid particle-grid fluid animation with enhanced details," *The Visual Computer*, vol. 29, no. 9, pp. 937–947, 2013.
- [6] C. Körner, M. Thies, T. Hofmann, N. Thürey, and U. Rüde, "Lattice Boltzmann model for free surface flow for modeling foaming," *Journal of Statistical Physics*, vol. 121, no. 1-2, pp. 179–196, 2005.
- [7] N. Thürey and U. Rüde, "Free surface lattice-Boltzmann fluid simulations with and without level sets," in *Proceedings of the Vision, Modeling, and Visualization Conference 2004 (VMV 2004)*, Stanford, CA, USA, November 2004.
- [8] N. Thürey, *Physically based animation of free surface flows with the lattice Boltzmann method*, [Ph. D. thesis], University of Erlangen, 2007.
- [9] C. Janssen and M. Krafczyk, "A lattice Boltzmann approach for free-surface-flow simulations on non-uniform block-structured grids," *Computers & Mathematics with Applications*, vol. 59, no. 7, pp. 2215–2235, 2010.
- [10] C. Janssen and M. Krafczyk, "Free surface flow simulations on GPGPUs using the LBM," *Computers & Mathematics with Applications*, vol. 61, no. 12, pp. 3549–3563, 2011.
- [11] H. W. Zheng, C. Shu, and Y. T. Chew, "Lattice Boltzmann interface capturing method for incompressible flows," *Physical Review E*, vol. 72, no. 5, article 056705, 2005.
- [12] M. Kaneda, T. Ueda, and K. Suga, "Hybrid model of lattice Boltzmann and CLSVOF methods for immiscible two-fluid flow," *Progress in Computational Fluid Dynamics, an International Journal*, vol. 13, no. 3/4, pp. 152–161, 2013.
- [13] F. Zhang, Z. Wang, J. Chang, J. Zhang, and F. Tian, "A fast framework construction and visualization method for particle-based fluid," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, Article ID 79, 2017.

- [14] F. Zhang, Q. Wei, and L. Xu, "An fast simulation tool for fluid animation in VR application based on GPUs," *Multimedia Tools and Applications*, pp. 1–24, 2019.
- [15] F. Gang and L. Shiguang, "Detail-preserving shape deformation in SPH fluid control," *Journal of System Simulation*, vol. 30, no. 6, article 2368, 2018.
- [16] G. Feng and S. Liu, "Detail-preserving SPH fluid control with deformation constraints," *Computer Animation and Virtual Worlds*, vol. 29, no. 1, article e1781, 2018.
- [17] F. Dagenais, J. Gagnon, and E. Paquette, "Detail-preserving explicit mesh projection and topology matching for particle-based fluids," *Computer Graphics Forum*, vol. 36, no. 8, pp. 444–457, 2017.
- [18] C. Wang, Q. Zhang, Z. Zhang, P. Yang, and Z. Xia, "Detail-preserving rendering of free surface fluid with Lattice Boltzmann," in *Transactions on Edutainment VI. Lecture Notes in Computer Science*, vol. 6758, Z. Pan, A. D. Cheok, and W. Müller, Eds., pp. 216–226, Springer, Berlin, Heidelberg, 2011.
- [19] N. Thürey, R. Keiser, M. Pauly, and U. Rüdè, "Detail-preserving fluid control," *Graphical Models*, vol. 71, no. 6, pp. 221–228, 2009.
- [20] Y. Xie, E. Franz, M. Chu, and N. Thürey, "tempoGAN: a temporally coherent, volumetric GAN for super-resolution fluid flow," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–15, 2018.
- [21] W. Lu, X. Zhang, H. Lu, and F. Li, "Deep hierarchical encoding model for sentence semantic matching," *Journal of Visual Communication and Image Representation*, article 102794, 2020.
- [22] Y. Zhang, W. Lu, W. Ou et al., "Chinese medical question answer selection via hybrid models based on CNN and GRU," *Multimedia Tools and Applications*, 2019.
- [23] P. Bhattachagor, E. Gross, and M. Krook, "A model for collision processes in gases," *Physical Review*, vol. 94, no. 3, p. 511, 1954.
- [24] Z. Guo, C. Zheng, and B. Shi, "Discrete lattice effects on the forcing term in the lattice Boltzmann method," *Physical Review E*, vol. 65, no. 4, article 046308, 2002.
- [25] G. Zhao-Li, Z. Chu-Guang, and S. Bao-Chang, "Non-equilibrium extrapolation method for velocity and pressure boundary conditions in the lattice Boltzmann method," *Chinese Physics*, vol. 11, no. 4, pp. 366–374, 2002.
- [26] L. Van, J. Wladimir, S. Green, and M. Sainz, "Screen space fluid rendering with curvature flow," in *I3D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games*, Boston, MA, USA, February 2009.
- [27] N. Truong and C. Yuksel, "A narrow-range filter for screen-space fluid rendering," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 1, no. 1, pp. 1–15, 2018.