

Hybrid particle–grid fluid animation with enhanced details

Chang-bo Wang · Qiang Zhang · Fan-long Kong ·
Hong Qin

Published online: 1 June 2013
© Springer-Verlag Berlin Heidelberg 2013

Abstract Simulating large-scale fluid while retaining and rendering details still remains to be a difficult task in spite of rapid advancements of computer graphics during the last two decades. Grid-based methods can be easily extended to handle large-scale fluid, yet they are unable to preserve sub-grid surface details like spray and foam without multi-level grid refinement. On the other hand, the particle-based methods model details naturally, but at the expense of increasing particle densities. This paper proposes a hybrid particle–grid coupling method to simulate fluid with finer details. The interaction between particles and fluid grids occurs in the vicinity of “coupling band” where multiple particle level sets are introduced simultaneously. First, fluids free of interaction could be modeled by grids and SPH particles independently after initialization. A coupling band inside and near the interface is then identified where the grids interact with the particles. Second, the grids inside and far away from the interface are adaptively sampled for large-scale simulation. Third, the SPH particles outside the coupling band are enhanced by diffuse particles which render little computational cost to simulate spray, foam, and bubbles. A distance function is continuously updated to adaptively coarsen or refine the grids near the coupling band and

provides the coupling weights for the two-way coupling between grids and particles. One characteristic of our hybrid approach is that the two-way coupling between these particles of spray and foam and the grids of fluid volume can retain details with little extra computational cost. Our rendering results realistically exhibit fluids with enhanced details like spray, foam, and bubbles. We make comprehensive comparisons with existing works to demonstrate the effectiveness of our new method.

Keywords Hybrid particle–grid simulation · Fluid animation · Two-way interaction · Enhanced details

1 Introduction

Physically based fluid simulation has been one of active research topics in computer graphics in recent years. However, due to its physics nature and the time and space complexity of simulating physical principles in reality, many of the fluid simulations can only handle fluid in a small and confined region such as a simple cube-shaped bounding box, but for large-scale free-boundary fluid with fine details, such simulation is subject to the limit governed by the grid size, hence the overall resolution of simulation region. In such grid-based simulation, it is unavoidable to optimize the grid refinement and conduct extra simulation in regions of interest in order to expect sub-grid features like foam and spray to occur.

There have been some works on large-scale fluid simulation in recent years, such as grid optimization [7], multi-scale particles [8], 2D and 3D coupling [5, 6], etc. As a result, it is possible to simulate large scenes such as boat floating on the lake, and large ship cruising in ocean. Still,

Electronic supplementary material The online version of this article (doi:10.1007/s00371-013-0849-6) contains supplementary material, which is available to authorized users.

C.-b. Wang (✉) · Q. Zhang · F.-l. Kong
Software Engineering Institute, East China Normal University,
Shanghai 200062, China
e-mail: cbwangcg@gmail.com

H. Qin
Department of Computer Science, Stony Brook University, Stony
Brook, NY, USA
e-mail: qin@cs.sunysb.edu

a common difficulty is the modeling and rendering of surface details. Although naively increasing the grid resolution and/or the number of particles could get the job done towards increased details such as foam and spray, extra computational expenses are becoming prohibitively high which has prevented its practical utility at any consumer-level computing platform.

For enhanced surface details, some methods have also been developed. Nonetheless, they are commonly limited in small-scale scene design such as a glass of water, because such scenario could be easily accommodated by reasonable resolutions in a desktop environment. In principle, the water drops are larger than the computing cells so that they can be accurately retained. In fact, for the two classic fluid simulation methods, Euler method and Lagrangian method have their respective advantages and limitations. Grid-based discretization can easily be utilized to model the large-scale fluid volume, and the particle-based method offers a good strategy to model details of fluid surface. It is our belief that the two-way coupling shall offer a good tradeoff, while taking advantages of both approaches. Yet, how to model the interaction between particles and grids remains elusive in general in spite of recent research progresses in this regard.

In this paper, we focus on large-scale fluid with enhanced surface details and develop a novel hybrid particle–grid method. Our hybrid particle–grid interactions occur at three different levels depending on their whereabouts in the simulation space, and three levels comprise: the grid for inner fluid volume, hybrid coupling band (which exhibits a layered-structure in the vicinity of free surface), and particles on the fluid surface. Their interaction and movement are dynamically synchronized with the evolution of fluid surfaces, and could naturally depict more details for fluid animation without the need of global grid refinement and/or increasing particle densities via splitting. To achieve the three-level model, we proposed a concept of Coupling Band, and accomplish improvement on the previous method in the following aspects.

Particle–grid hybrid modeling with two-way coupling

A hybrid model is proposed to couple the grid-based method for the fluid volume with the particle-based method for the surface details. To model sub-grid details, extra particle-based simulation is performed near the fluid surface. Adaptive grid sampling optimization is adopted for the fluid volume which makes the large fluid scene with local details possible to achieve.

Bidirectional particle–grid interaction with coupling band

The interaction between particles and grids is bidirectional and strong-coupling. A coupling band is based on the distance function which takes the interface between grid and particle as the zero-isosurface. It is a layered-structure which

provides coupling weight for both participating particles and the enclosed cells in order to achieve a true two-way coupling.

Enhanced surface details using SPH and diffuse particles

Local details near the coupling band are enhanced by adding dynamic particles with different motion rules. Three types of particles are introduced with different dynamic interactions. According to the criteria if they are inside, outside or on the coupling band, they are respectively classified as spray, foam, or bubbles.

The rest of this paper is organized as follows. After briefly reviewing the previous works in Sect. 2, the grid-based method and the PLS for free surface tracking are introduced in Sect. 3. In Sect. 4, the coupling band for hybrid particle–grid fluid is elaborated. Then we describe the adaptive grid optimization in Sect. 5 and explain the added diffuse particles for enhanced surface details in Sect. 6, and we then briefly explain our results in Sect. 7. Finally, the conclusion and future work are outlined in Sect. 8.

2 Related work

In recent years some works have already been focused on the simulation of free surface fluid. Based on the level set method, Enright et al. [1] proposed the particle level set (PLS) method which adds more details into the free surface fluid, and greatly improves the visual realism. They combined the level set method with the Lattice Boltzmann method (LBM) [3] and proposed a new free surface model based on the Volume of Fluid (VOF) method [2]. Kwak et al. [18] then presented a hybrid method which couples LBM with PLS. They introduced a complex surface tracking method (like quad-tree) to obtain more accurate surface with features. But the results are only limited to grid-level features, sub-grid features like spray are hardly simulated. On the other hand, the particle-based smooth particle hydrodynamics (SPH) is becoming more popular [4], as its formulation is easy to implement, and details like spray and foam can be modeled in a straightforward fashion. However, it is difficult to extract the surface, and how to select an appropriate smooth kernel function also remains ad hoc. The common shortcoming of the aforementioned fluid simulation methods is that they cannot be directly used to simulate the large-scale fluid with fine surface details.

To simulate the fluid in a region that is as large as possible, grid optimization and hierarchical modeling methods are necessary. Hybrid grids are proposed in [5, 6] which mix the 2D height map with the 3D computing grid, and in [7], the new strategy is to coarsen or refine cells based on its distance to the surface. For the SPH optimization, Solenthaler and Gross [8] presented a two-scale particle method

to enhance the detail at some regions where complex flow behavior emerges by splitting particles into smaller ones. But the referred methods inevitably enlarge the grid resolution in order to model surface details in localized region, and at the same time coarsen the grid to adapt to large-scale region without the need of modeling surface details. Such conflicting requirements make the underlying computational scheme difficult to deploy and manage.

New solutions based on additional simulation are proposed to solve these problems. Chentanez and Muller [9] presented a hybrid method combining grid-based shallow water solver and particle-based non-interacting point masses. The simplified computation makes the real-time simulation possible but limits visual effects of the scene. Losasso et al. [10] coupled the SPH with the particle level set, and simulated the fluid volume with PLS and the surface details with SPH with convincing simulation results. But no optimization is made to the simulation of fluid volume. Ihmsen et al. [11] added diffuse particles which only take simple computations with three types: foam, bubble, and spray to the SPH fluid and obtained attractive visual effects. However, this is only a one-way coupling method which does not offer sufficient functionalities. This paper is inspired by prior works and advances the current state-of-the-art with the new hybrid particle–grid method supporting the two-way coupling.

3 Modeling free surface fluid with PLS

3.1 Grid-based fluid solver with particle level set

Our fluid solver is the grid-based Lattice Boltzmann method (LBM), which uses simple rules of quantifying micro-particle movement to reflect macro-fluid changes. The Bhatnager–Gross–Krook model with single-relaxation can be formulated as

$$f_i(\mathbf{x} + \mathbf{e}_i, t + \Delta t) = f_i(\mathbf{x}, t) - \frac{1}{\tau} (f_i(\mathbf{x}, t) - f_i^{\text{eq}}(\rho, \mathbf{u})), \quad (1)$$

where \mathbf{e}_i is the discrete particle velocity in the direction i , $f_i(\mathbf{x}, t)$ and $f_i^{\text{eq}}(\rho, \mathbf{u})$ are the density distribution function and the equilibrium density distribution function in the cell at \mathbf{x} at time t , τ is relaxation time. In each discrete spatial cell, the density ρ and velocity \mathbf{u} can be formulated as

$$\rho(\mathbf{x}, t) = \sum f_i(\mathbf{x}, t), \quad \mathbf{u}(\mathbf{x}, t) = \frac{1}{\rho} \sum \mathbf{e}_i f_i(\mathbf{x}, t). \quad (2)$$

This distribution function f_i accounts for the number of particles along the direction i , each discrete particle has several directions of possible movement. According to the conservation of energy, different initial conditions and boundary conditions of the recursive evolution for the distribution functions can be derived.

For surface tracking, according to the particle level set method, the evolution equation of the distance function in level set can be written as

$$\phi_t + \mathbf{u} \cdot \nabla \phi = 0, \quad (3)$$

which can be spatially discretized using the 5th-order accurate Hamilton–Jacobi Weighted Essentially Non-Oscillatory (HJ-WENO) scheme [12] and temporally discretized using the 3rd-order Total Variation Diminishing Runge–Kutta (TVD-RK) scheme [13].

To update the distance function, an external velocity field by the numerical solution is needed. To keep the level set as a signed distance function (i.e., that it must meet the equation $|\nabla \phi| = 1$ in the simulation process), the fast marching method (FMM) is used which advects the interface in an up-wind fashion [14]. FMM is also used for the velocity extrapolation which we will address in the following section [15]. The cells to be updated can be reduced by using a narrow band level set method, as described in [16]. In this case, the level set is only propagated in a region within a certain range around the fluid surface, hence the velocities only have to be extrapolated within this region.

As a classic method for surface tracking, the level set method extracts smooth surface of grid-based fluid simulation, however, one weakness is that it smoothes the corner of the simulation result and fails to capture the thin features and the volume of liquid tends to decrease during simulation. To overcome this problem of volume loss, the particle level set method (PLSM) [13] uses massless marker particles in front tracking in sub-grid regions around the interface.

The added particles have the corresponding value of distance function in level set, so the particles with positive distance value are outside the interface. Those particles work to correct the numerical errors generated in the process of the re-initialization of the level set by comparing the distance value from escaped particles and cells [13]. Error correction is used both after the level set has been updated and again after it has been re-initialized.

3.2 Hybrid method for free surface fluid

To combine the grid of fluid with PLSM, we need to obtain the velocity field to advect the level set function and the particles. The velocity of each cell can be calculated using Eq. (2).

However, the evolution equation requires the velocities on both sides of the interface to maintain the validity and smoothness of the signed distance function. Here the fast marching method is used to extrapolate the fluid velocities in the neighboring empty cells in each step.

Thus, the distance functions can be advected by the velocity field. In this paper, we use the semi-Lagrangian advection scheme [17] for the advection method. The basic idea

of the semi Lagrangian method is as follows. Given the level set function and the velocity field \mathbf{u} , the updated value at a point \mathbf{x} after Δt is

$$\phi(\mathbf{x}) = \phi(\mathbf{x} - \mathbf{u}(\mathbf{x})\Delta t). \tag{4}$$

Free surface fluid simulation has to classify the cells into three types: *Gas* for cells full of gas, *Liquid* for cells full of fluid, *Interface* for the cells contain both gas and fluid. The *Liquid* cells can be dealt with using normal LBM cell, and the *Gas* cells do not participate the computation. So we only focus on the update of the *Interface* cells. In [3], the mass exchange is introduced which can record the liquid fraction (ranging from 0 to 1) of a cell so that the cells are classified (1 for *Liquid*, 0 for *Gas*, and other values for *Interface*). This method can guarantee the conservation of mass, but needs computing time for the updating of the mass in the LBM stream step and the switching of the types of cells, especially the redistribution of the extra mass when an *Interface* cell switches to a *Gas* or a *Liquid* cell. In [18], the liquid fraction is obtained by iteratively dividing a *Interface* cell into four cells and calculating the signed distance functions of the new cells until the predefined accuracy is satisfied. This reduces the error in conservation of mass, but also increases the computation cost. We classify the cells based on their distance values: negative value for *Gas*, cells with positive value and gas neighbors are classified as *Interface*, and the others are *Liquid*. This makes the updating of types of cells much faster. We shall classify and update the types of the cells in each time step.

As mentioned above, *Interface* cells must have *Gas* neighbors. So in the stream step of standard LBM solver, the exchange of density distribution functions between *Interface* and *Gas* cells need to be reconstructed (the original equation is Eq. (1)). We take the same reconstruction step as in [3]:

$$f'_i(\mathbf{x}, t + \Delta t) = f_i^{\text{eq}}(\rho_A, \mathbf{u}) + f_{\tilde{i}}^{\text{eq}}(\rho_A, \mathbf{u}) - f_i(\mathbf{x}, t), \tag{5}$$

where \mathbf{x} is the *Interface* cell, i is the direction from \mathbf{x} to the *Gas* neighbor, \tilde{i} is the opposite direction of i , ρ_A is the density of air which is treated as constant 1, \mathbf{u} is the velocity of \mathbf{x} .

To balance the forces on each side of the interface, the distribution functions coming from the direction of the interface normal are also reconstructed [3]. So we have

$$\mathbf{n} \cdot \mathbf{e}_i > 0 \quad \text{with } \mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}, \tag{6}$$

where \mathbf{n} is the surface normal. In our method, the normal is easily computed from the distance function. The main computational steps in the PLS-LBM framework are illustrated in Fig. 1, where grid-based fluid solver provides velocity field for PLS to be advected, and PLS offers different types of all cells according to the distance function.

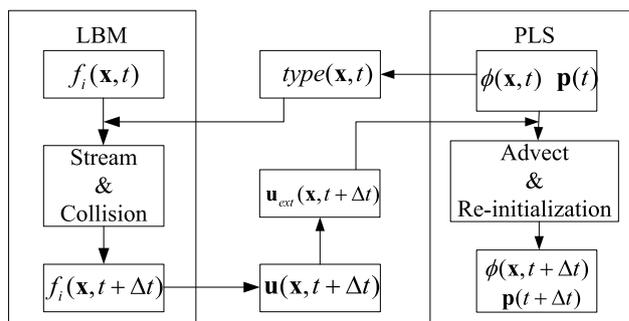


Fig. 1 The computational pipeline in our PLS-LBM model

4 Two-way coupling of grids and particles

For the hybrid method for free surface fluid discussed above, since the PLS and LBM are both based on the grid, in order to simulate the fluid with surface details, excessive sub-grid refinement must be done but at the expense of increasing the global resolution of computation grid. Now, it sets a stage for us to introduce a novel hybrid particle-grid idea. In the PLS-LBM framework, there is a zero-isosurface, i.e., the fluid surface which is called *Border* below to avoid confusion. It may be noted that the *Border* is not the fluid surface in the particle-grid fluid but the interface between grid-based fluid and particle-based fluid.

4.1 Particle-based fluid solver

Our particle-based fluid solver is SPH. SPH smoothes quantities over a neighborhood with the radius h by using a kernel $W(\mathbf{x}_{ij}, h)$ to weigh the contributions according to the distance between two particles i and j . A smoothed, physical quantity q for a particle i can thus be computed by summing up the contributions of the neighboring particles j :

$$q_i = \sum_j \frac{m_j}{\rho_j} q_j W(\mathbf{x}_{ij}, h), \tag{7}$$

where m_j is the mass of particle j and ρ_j is its density. The motion of the particles is governed by the pressure, viscosity and external force [21]:

$$\rho \frac{\partial \mathbf{u}}{\partial t} = f^{\text{pressure}} + f^{\text{viscosity}} + f^{\text{external}}, \tag{8}$$

$$f_i^{\text{pressure}} = -\rho_i \sum_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) m_j \nabla W(r_i - r_j, h), \tag{9}$$

$$f_i^{\text{viscosity}} = \mu \sum_j (\mathbf{u}_j - \mathbf{u}_i) \frac{m_j}{\rho_j} \nabla^2 W(r_i - r_j, h), \tag{10}$$

where μ is the coefficient of viscosity, and ρ_i, ρ_j, p_i, p_j are the density and the pressure of the particles i and j , which

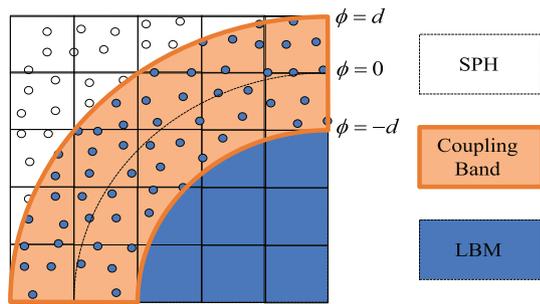


Fig. 2 The coupling band in the hybrid particle–grid method

can be calculated according to Becker’s method [19],

$$p = a \left(\left(\frac{\rho}{\rho_0} \right)^k - 1 \right), \quad k = 7, \tag{11}$$

where ρ_0 is the static density, a is a constant. And the weakly compressible SPH solver is used to keep density variations lower. The only difference is the size of the stiffness value k in the gas equation. Here, k is chosen to be very high so that the density fluctuations are below 1 %.

4.2 Coupling band for particle–grid coupling

To couple the grid with SPH particles for two-way interaction, a coupling band which is based on the distance function in the level set method is proposed. Each cell \mathbf{x} in fluid grid keeps a value of distance from the surface $\phi(\mathbf{x})$. The coupling band means the area which is made up of all the cells whose ϕ is in a given range $[-d, d]$. So these cells are distributed on both sides of the zero-iso-surface where $\phi(\mathbf{x}) = 0$ (i.e., the *Border*) as can be seen in Fig. 2. Only in this band, we perform the interaction between the fluid cells and particles. When a particle moves out of the coupling band, its motion will be defined in the next section.

Grid → Particle: For particles that are in the coupling band, neighboring cells are treated as boundary condition. For a particle \mathbf{x}_p , we can find the cell \mathbf{x}_c that contains this particle. Since the particles are near the surface, they are likely to be driven by the underlying fluid cells, so the velocity of the particle \mathbf{u}_p is partially determined by \mathbf{x}_c . We introduce a coupling weight here for this particle which can be represented as

$$w(\mathbf{x}_p) = \begin{cases} |\phi(\mathbf{x}_p)|/d & \phi(\mathbf{x}_p) \in [-d, d] \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Then we can get its velocity and update its position by

$$\begin{aligned} \mathbf{u}'(\mathbf{x}_p) &= w(\mathbf{x}_p) \cdot \mathbf{u}'_{\text{sph}}(\mathbf{x}_p) + (1 - w(\mathbf{x}_p)) \cdot \mathbf{u}'_{\text{lbm}}(\mathbf{x}_p), \\ \mathbf{x}'_p &= \mathbf{x}_p + \mathbf{u}'(\mathbf{x}_p)\Delta t \end{aligned} \tag{13}$$

where $\mathbf{u}'_{\text{sph}}(\mathbf{x}_p)$ and $\mathbf{u}'_{\text{lbm}}(\mathbf{x}_p)$ are the velocities obtained from SPH solver and the LBM solver, respectively, and $\mathbf{u}'(\mathbf{x}_p)$ is

the resultant velocity. And the closer the distance between particle and *Border*, the greater the drag force it is exerted. When the particle reaches the *Border*, its movement is completely determined by the velocity of the cell where the particle is residing, so that the coupling band makes the coupling from grid to particles more smoothly.

Particle → Grid: For an LBM cell \mathbf{x}_c in the coupling band, the density and velocity of neighboring particles are resampled in the neighboring cell $\mathbf{x}_c + \mathbf{e}_i$ according to

$$\rho'_{\text{lbm}}(\mathbf{x}_c + \mathbf{e}_i) = \rho_{\text{lbm}}(\mathbf{x}_c + \mathbf{e}_i) + \rho_{\text{sph}}(\mathbf{x}_c + \mathbf{e}_i), \tag{14}$$

$$\begin{aligned} \mathbf{u}'_{\text{lbm}}(\mathbf{x}_c + \mathbf{e}_i) &= \frac{\rho_{\text{lbm}}(\mathbf{x}_c + \mathbf{e}_i)}{\rho'_{\text{lbm}}(\mathbf{x}_c + \mathbf{e}_i)} \cdot \mathbf{u}_{\text{lbm}}(\mathbf{x}_c + \mathbf{e}_i) \\ &+ \frac{\rho_{\text{sph}}(\mathbf{x}_c + \mathbf{e}_i)}{\rho'_{\text{lbm}}(\mathbf{x}_c + \mathbf{e}_i)} \cdot \mathbf{u}_{\text{sph}}(\mathbf{x}_c + \mathbf{e}_i), \end{aligned} \tag{15}$$

where $\mathbf{u}_{\text{lbm}}(\mathbf{x}_c + \mathbf{e}_i)$ and $\rho_{\text{lbm}}(\mathbf{x}_c + \mathbf{e}_i)$ are the velocity and density obtained from the cell $\mathbf{x}_c + \mathbf{e}_i$, and the $\mathbf{u}_{\text{sph}}(\mathbf{x}_c + \mathbf{e}_i)$, $\rho_{\text{sph}}(\mathbf{x}_c + \mathbf{e}_i)$ are interpolated from the SPH using Eq. (7). In this way, the distribution density functions are updated according to the resultant density and velocity, so the cell is affected by the neighboring particles. A 2D hybrid particle–grid fluid is presented in the following figures.

In Fig. 3, first, a rectangle represented by SPH particles and a circle represented by LBM grid are shown in the left figure. Then, they fall down due to the gravity and interact with each other. To make it visually clear, the fluid cells in LBM solver are represented in square while ignoring the liquid fraction.

5 Adaptively-refined grids with distance function

The local-grid refinement scheme is first described in [20]. Adaptive coarsening algorithm being applied to fine grids at the area close to interface [7] can build a transition zone between fine cells and coarse ones more effectively. However, they classifies the cells near the interface into 5 types and takes 3 passes to update the cells for refinement and 2 passes for coarsening. Each pass contains neighborhood checking. In our method, each cell keeps a value of distance from the interface, so it can respectively refine or coarsen itself if $\phi(\mathbf{x}) > -\phi_f$ or not, where \mathbf{x} is the position of the center of the cell, ϕ_f is a user-defined parameter which controls how deep the deepest fine cell is from the water surface. To make the computation in the area of coupling band precise, the ϕ_f shall satisfy $\phi_f > d$, where d is the half of the width of coupling band in Eq. (12). So, a larger ϕ_f results in more fine cells in the same situation. And that is to say, fine cells are for the area near the interface, and coarse cells are the ones that are far away from the interface. Figure 4 shows the types of cells.

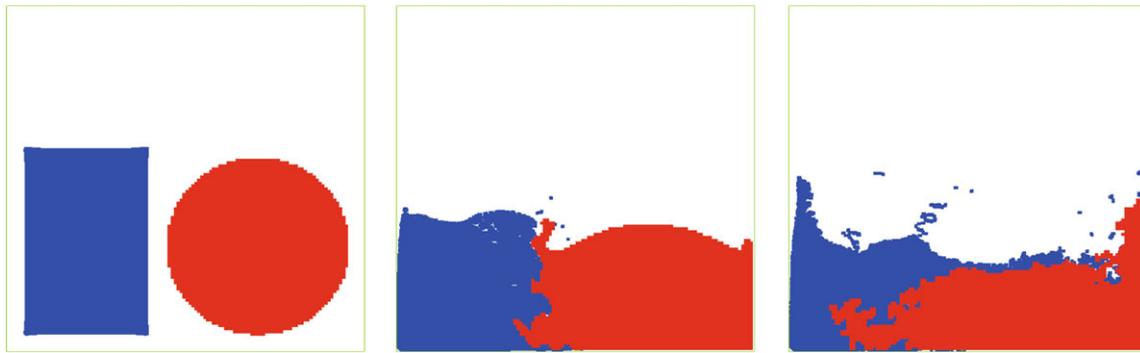


Fig. 3 A 2D hybrid particle–grid fluid simulation starts from a particle-based rectangle fluid and a grid-based circle fluid in the *first figure* and continues their interactions in the middle and *right figures*

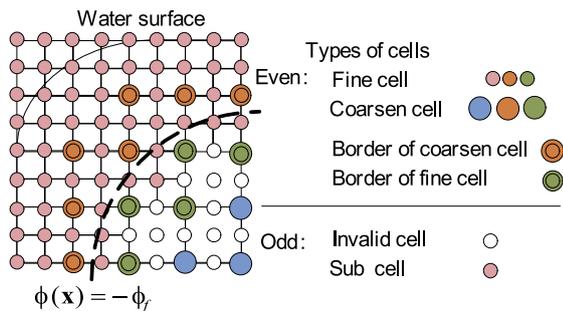


Fig. 4 All types of cells in adaptively-refined grid

There are two kinds according to the indices of the cells: even and odd. Even cells mean the cells with even coordinates on all the axes while odd cells mean the cells with at least one odd coordinate on all the axes. The exchange rules of states are as follows. Fine and coarse occurs only on the even cells, while the state of an odd cell is either sub-cell (work as a fine cell) or invalid cell according to the states of its neighbors: it is sub-cell when it has a fine cell neighbor, otherwise the invalid cell. In Fig. 4, the border of coarse cells and the border of fine cells are both treated as not only fine cell but also coarse cell, so these two types of cells are the key for the coupling of fine and coarse cells. The grid spacing Δx_c and Δt_c on the coarse grids are twice as much as those of the fine grids. So the relaxation time needs to be recalculated by

$$\tau_c = \frac{1}{2} \left(\tau_f - \frac{1}{2} \right) + \frac{1}{2}, \tag{16}$$

where τ_c and τ_f are the relaxation time for coarse and fine grids, respectively. To ensure the velocity and density consistent and continuous in different cells, the non-equilibrium parts of the distribution functions have to be rescaled. The stream and collision steps on the fine and sub-cells perform twice as much as the coarse ones do at each time step. The distribution functions are rescaled directly on the borders of

coarse and fine cells. For the border of fine cells,

$$f_{if} = f_{if}^{eq} + \frac{2(\tau_c - 1)}{\tau_f - 1} (f_{if} - f_{if}^{eq}), \tag{17}$$

and for the border of coarse cells,

$$f_{ic} = f_{ic}^{eq} + \frac{\tau_f - 1}{2(\tau_c - 1)} (f_{ic} - f_{ic}^{eq}), \tag{18}$$

where the subscripts c and f are for coarse and fine cells, respectively. When some neighboring cells of a fine cell are absent (due to the fact that they are coarse cells), sub-cells (in the neighborhood) should be made up by spatial interpolating distribution functions of coarse cells.

Here, the external forces, such as drop gravity, can change the velocity for uniform cells like: $\mathbf{u}' = \mathbf{u} + \tau \mathbf{g}$, where \mathbf{u} is the original velocity, and \mathbf{u}' is the resultant velocity, τ is the relaxation time, and \mathbf{g} is the acceleration of gravity. However, for multi-scale cells, the affection is different for different positions:

$$\mathbf{u}'_f = \mathbf{u}_f + \tau_f \mathbf{g}, \quad \mathbf{u}'_c = \mathbf{u}_c + \tau_c \frac{\Delta x_c}{\Delta x_f} \mathbf{g}, \tag{19}$$

where \mathbf{u}_f , \mathbf{u}_c , Δx_f , Δx_c are the velocity and grid internal of the fine and coarse cells, respectively.

As the simulation continues, the state (fine or coarse) of grid should be updated dynamically according to the distance function. For cells updated from coarse to fine, their neighbors which are all odd cells should become sub-cells as shown in Fig. 4. And the initialization of the cell is obtained from linear interpolation. While for cells updated from fine to coarse, their neighbors which have no fine neighboring cells should become invalid cells, and their properties like the density, velocity and distribution functions are simply discarded.

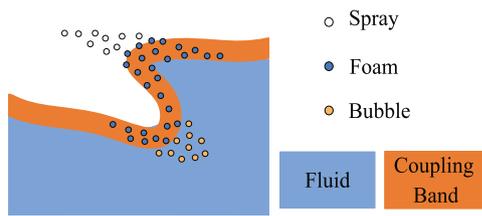


Fig. 5 Classification of *Spray*, *Foam*, and *Bubble* particles

6 Enhanced surface details with spray, foam, and bubbles

In addition to the coupling between particles and grid in the coupling band, more details occurring on the fluid surface are spray, foam and bubbles. Here we treat particles as additional simulation near the fluid surface to model these interesting features. Initially, there is only fluid expressed by grid, particles are adaptively added, classified and deleted also by the coupling band according to the fluid evolution.

6.1 Formation and classification of particles

In particle level set, the massless particles are seeded on both sides of the fluid interface and some are removed when they come across the interface beyond a long distance. These massless particles are only advected by the velocity field of fluid grid and used to correct the distance function of level set as usual. As in [10], we add SPH particles as *Foam* at the positions of the removed negative particles that were originally in the interior of the fluid volume. Here the added SPH particles are different from the particles in level set tracking method. These particles tend to leave the surface of the fluid and move into the air, they will be marked as *Spray* when they leave the coupling band. Additionally, we also add particles as *Foam* at the positions of the removed positive particles for modeling more features, but the particles of this part tend to enter the fluid volume. They will be marked as *Bubble* when they leave the coupling band.

For a level set’s particle p , its original position is \mathbf{x}_p^o and \mathbf{x}_p^t denotes its position at time t . If $\phi(\mathbf{x}_p^o) \cdot \phi(\mathbf{x}_p^t) < 0$, we first delete particle p , then compute the density at \mathbf{x}_p^t ; if $\rho_{\mathbf{x}_p^t} < \rho_0$, we add an SPH particle at \mathbf{x}_p^t , otherwise do nothing. We compare the densities $\rho_{\mathbf{x}_p^t}$ and ρ_0 to avoid too many SPH particles being added in a narrow region. A particle s is classified as *Spray*, *Foam* or *Bubble* according to its distance from the surface $\phi(\mathbf{x}_s)$. Here, if $\phi(\mathbf{x}_s) > d$, particle s is considered *Spray*; if $\phi(\mathbf{x}_s) < -d$, particle s is considered *Bubble*; in all other cases, the particle s is considered to be *Foam*. The positions of *Spray*, *Foam*, and *Bubble* are illustrated in Fig. 5.

6.2 Motion of adaptive particles

We can get the motion of foam particles using Eq. (13). And the motion of spray particle is computed as normal SPH par-

Algorithm 1 The process of the hybrid particle–grid fluid simulation

```

procedure GRIDPARTICLEFLUID(PLS, LBM, SPH)
    Update the LBM and handle the particle–grid coupling by using Eq. (14) and Eq. (15)
    Velocity extrapolation for LBM;
    Update the  $\phi$  of PLS by the  $\mathbf{u}$  of LBM;
    Update the particles of PLS by the  $\mathbf{u}$  of LBM;
    for all particle  $\in$  PLS do
        Fix the corresponding  $\phi$  of PLS
        if particle  $\in$  Removed Particles then
            if  $\rho_{\text{sph}}(\text{particle}) < \rho_0$  then
                Insert an SPH particle
            end if
        end if
    end for
    Re-initialize the  $\phi$  of PLS;
    Fix the  $\phi$  by the particles of PLS
    Update the states (Spray, Foam, Bubble) of the particles of SPH by the  $\phi$  of PLS
    for all particle  $\in$  SPH do
        if particle  $\in$  Spray then
            Take normal SPH steps
        end if
        if particle  $\in$  Foam then
            //handle the particle–grid coupling
            Update particle by Eq. (13)
            Delete the time-up particle
        end if
        if particle  $\in$  Bubble then
            Update particle by Eq. (20) and Eq. (21)
        end if
    end for
end procedure
    
```

ticle. Due to the high density contrast of water and air, the force exerted on the bubbles includes buoyancy which counteracts the gravity, and drag force which makes the particles flow with the fluid volume. Therefore, the velocity of an air bubble can be computed as

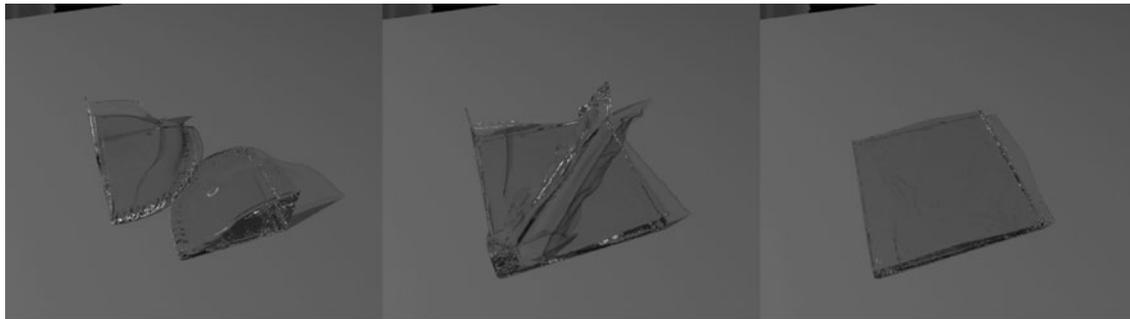
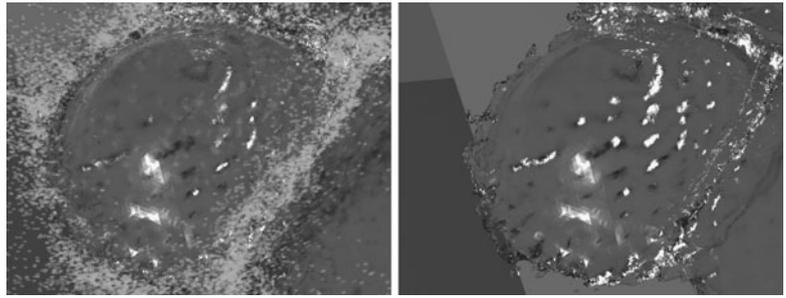
$$\frac{\partial \mathbf{u}_{\text{bub}}}{\partial t} = -w_g \mathbf{g} + w_d \frac{\mathbf{u}_{\text{lbm}}(\mathbf{x}_{\text{bub}}, t + \Delta t) - \mathbf{u}_{\text{bub}}(t)}{\Delta t}, \quad (20)$$

where \mathbf{u}_{bub} is the velocity of the *Bubble* particle, w_g and w_d are the weight of buoyancy and drag force, respectively. Then we update their positions as

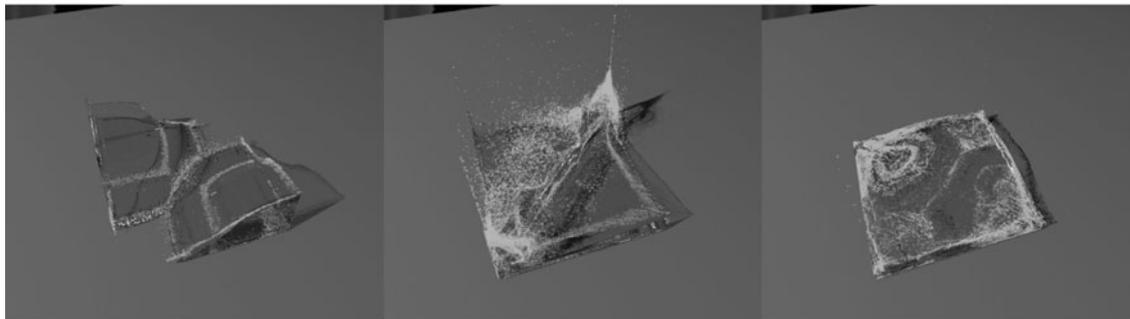
$$\mathbf{x}_{\text{bub}}(t + \Delta t) = \mathbf{x}_{\text{bub}}(t) + \Delta t \mathbf{u}_{\text{bub}}(t + \Delta t). \quad (21)$$

It may be noted that the spray and bubbles could migrate into the coupling band and become foams, so in practice we keep a timer for each foam particle. when the timer expires, the corresponding particle shall be removed.

Fig. 6 Close-up view of the comparison of the results with and without surface details



(a) PLS-LBM without details



(b) PLS-LBM with details

Fig. 7 Comparison between simulated results of our normal PLS-LBM with and without enhanced surface details. The grid size is $150 \times 150 \times 150$

7 Experimental results and analysis

Based on our novel method detailed above, we design different scenes of fluid simulation with and without particles. The attributes such as ρ , \mathbf{u} , f_i , ϕ of the grid are stored in arrays, while the properties of the particles are stored in a point buffer as a list for the frequent inserting and deleting operations. The computational process of the entire algorithm is detailed in Algorithm 1. Our system is implemented using C++. Our hardware platform is PC with Intel (R) Pentium (R) 3.4 GHz CPU, 3 GB memory, and NVIDIA GeForce GTX260 Graphics card. The simulation results are rendered by using V-Ray (www.chaosgroup.com).

The coefficient d of coupling band here is $5\Delta x_f$ where Δx_f is size of a fine cell and to keep the computation in coupling band accurate, the ϕ_f often meets $\phi_f > d$, here we set it

as $8\Delta x_f$. To get better visual results, we choose the lifetime of the foams as 700 steps and add a probability for generating an SPH particle at the position referred to in Sect. 6.2. The probability we used in these scenes is 0.01.

Figure 6 is a close-up view of the comparison of the simulation with and without details. In the left of Fig. 6, sub-grid features are captured by particles which make the scene more realistic.

Figure 7 shows the simulation result by the PLS-LBM framework with and without enhanced surface details. The grid size is $150 \times 150 \times 150$. In Fig. 7(a), only PLS-LBM is used, the surface is well tracked, but due to the inherent weakness of PLS, the surface is obviously over-smoothed and is lacking details on the surface. In this case, it takes about 11 second per frame at most during the simulation.



Fig. 8 An ocean scene with our novel PLS-LBM method with enhanced surface details. The grid size is $400 \times 100 \times 200$

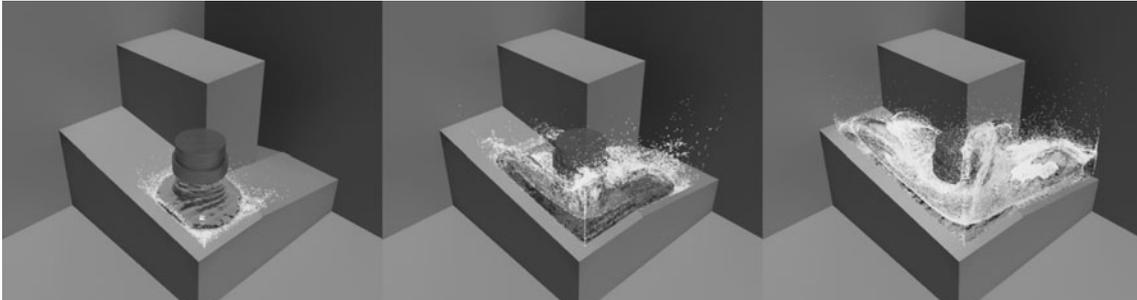


Fig. 9 A pipe scene with our novel PLS-LBM method with enhanced surface details. The grid size is $100 \times 100 \times 100$

Table 1 Timing, grid sizes, and particle counts in our experiments

Scenes	Grid size	Particles	Second/frame	
			LBM	SPH
Fig. 7(a)	$150 \times 150 \times 150$	0	9.3	0.0
Fig. 7(b)	$150 \times 150 \times 150$	300 K	9.3	0.6
Fig. 8	$400 \times 100 \times 200$	1 M	60.3	1.9
Fig. 9	$100 \times 100 \times 100$	160 K	6.4	0.4

While in Fig. 7(b), additional particle-based simulation is added onto the surface and the details are showing with at most 300 K particles for each step of the simulation, and the particle-based simulation costs less than 1 second in total time.

Figures 8 and 9 show the simulation results of an ocean scene and a scene in which a pipe of water stream goes down by using the PLS-LBM method with enhanced surface details. The sizes of the grid are $400 \times 100 \times 200$ and $100 \times 100 \times 100$, respectively. The spray, foam, and bubbles of the ocean are modeled by the SPH particles, and the number of particles is about 1 M and 160 K, respectively. The detailed comparison is documented in Table 1.

8 Conclusion and future work

In this paper, a novel particle–grid hybrid framework for free surface fluid is presented. In a nutshell, our method gener-

alizes PLS to support the layered structure in the vicinity of free surface. We have applied our novel method to fluid simulation with enhanced visual details in regions that enclose free surface. Additionally, we developed a hybrid particle–grid method that tightly couples particle-based simulation with grid-based fluid volume simulation, with a goal to facilitate the large-scale fluid with fine surface details. To add sufficient amount of surface details into fluid simulation, the adaptive refinement method has been deployed to the cells near the surface in order to dynamically update diffuse particles and tightly couple them with surrounding fluid in a bidirectional manner. The animation effects and the time statistics of our method, as well as its comparisons with other related works, are documented in Table 2. We notice that the visual effects we have obtained might not be the most ideal, nonetheless, the simulation time is shorter compared with the methods with better details. The SPH particles allow us to use coarser grid, and the fluid volume comprising cells allow us to reduce the number of the particles. They both reduce the computational cost of the simulation with little effect of details.

There are still limitations to overcome. Our method has not yet considered the local rules for the geometry of fluid surface and the physical movement of particles, and the fluid region which we can simulate is still not large enough.

Future research efforts are: to investigate better grid optimization strategies, and enlarge the simulation region fur-

Table 2 The comparison between our novel method and other recent work

Related work	Key methods	Details	Speed	Largest simulation region
Enright [1]	PLS	+	+++	$140 \times 110 \times 90$
Thurey [7]	LBM	+++	+	$480 \times 480 \times 480$
Losasso [10]	SPH, PLS	+++	+	NA (SPH) + $560 \times 120 \times 320$ (grid)
Ihmsen [11]	SPH, Diffuse particles	+++	++	3.5M (SPH) + 7M (Diffuse)
Our Method	SPH, PLS-LBM	++	+++	300 K (SPH) + $150 \times 150 \times 150$ (grid)

ther. Fluid–solid coupling methods shall also be evaluated within our hybrid particle–grid framework, as there are both particles and cells in the simulation which would make fluid–solid coupling much more challenging. Extending this method to simulate the large-scale disaster scenes also deserves immediate investigation.

Acknowledgements This paper was partially supported by Natural Science Foundation of China (Grant Nos. 61070128, 61272199), National Natural Science Foundation of China (Grant Nos. 61190120, 61190121, and 61190125) and National Science Foundation of USA (Grant Nos. IIS0949467, IIS1047715, and IIS1049448), Innovation Program of the Shanghai Municipal Education Commission (Grant No. 12ZZ042), Fundamental Research Funds for the Central Universities, and Shanghai Knowledge Service Platform for Trustworthy Internet of Things (Grant No. ZF1213).

References

- Enright, D., Marschner, S., Fedkiw, R.: Animation and rendering of complex water surfaces. *ACM Trans. Graph.* **21**(3), 736–744 (2002)
- Thurey, N.: A single-phase free-surface Lattice Boltzmann Method. Master thesis, Dept. of Computer Science, University of Erlangen-Nuremberg (2003)
- Thurey, N., Rude, U.: Free surface lattice-Boltzmann fluid simulations with and without level sets. In: *Proceedings of Workshop on Vision, Modeling and Visualization*, California, USA, 2004, pp. 199–207. IOS Press, Amsterdam (2004)
- Bicknell, G.: The equations of motion of particles in smoothed particle hydrodynamics. *SIAM J. Sci. Stat. Comput.* **12**(5), 1198–1206 (1991)
- Irving, G., Guendelman, E., Losasso, F.: Efficient simulation of large bodies of water by coupling two and three dimensional techniques. *ACM Trans. Graph.* **25**(3), 805–811 (2006)
- Thurey, N., Rude, U., Stamminger, M.: Animation of open water phenomena with coupled shallow water and free surface simulations. In: *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 157–164 (2006)
- Thurey, N., Rude, U.: Stable free surface flows with the lattice Boltzmann method on adaptively coarsened grids. *Comput. Vis. Sci.* **12**(5), 247–263 (2009)
- Solenthaler, B., Gross, M.: Two-scale particle simulation. *ACM Trans. Graph.* **30**(4), 81:1–81:8 (2011)
- Chentanez, N., Muller, M.: Real-time simulation of large bodies of water with small scale details. In: *Proceedings of ACM SIGGRAPH/EUROGRAPHICS Symposium on Computer Animation*, pp. 197–206 (2010)
- Losasso, F., Talton, J., Kwatra, N.: Two-way coupled SPH and particle level set fluid simulation. *IEEE Trans. Vis. Comput. Graph.* **14**(4), 797–804 (2008)
- Ihmsen, M., Akinci, N., Akinci, G., Teschner, M.: Unified spray, foam and bubbles for particle-based fluids. *Vis. Comput.* **28**(6–8), 669–677 (2012)
- Jiang, G., Peng, D.: Weighted Eno schemes for Hamilton–Jacobi equations. *SIAM J. Sci. Comput.* **21**(6), 2126–2143 (1997)
- Enright, D., Fedkiw, R., Ferziger, J., Mitchell, I.: A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.* **183**(1), 83–116 (2002)
- Sethian, J.: Fast marching methods. *SIAM Rev.* **41**(2), 199–235 (1999)
- Adalsteinsson, D., Sethian, J.A.: The fast construction of extension velocities in level set methods. *J. Comput. Phys.* **148**(1), 2–22 (1998)
- Adalsteinsson, D., Sethian, J.A.: A fast level set method for propagating interfaces. *J. Comput. Phys.* **118**(2), 269–277 (1995)
- Enright, D., Losasso, F., Fedkiw, R.: A fast and accurate semi-Lagrangian particle level set method. *Comput. Struct.* **83**(6–7), 479–490 (2005)
- Kwak, Y., Kuo, C.C.J., Nakano, A.: Hybrid lattice-Boltzmann/level-set method for liquid simulation and visualization. *Int. J. Comput. Sci.* **3**(6), 579–592 (2009)
- Becker, M., Teschner, M.: Weakly compressible SPH for free surface flows. In: *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 209–217 (2007)
- Filippova, O., Hanel, D.: Grid refinement for lattice-BGK models. *J. Comput. Phys.* **147**(11), 219–228 (1998)
- Muller, M., Charypar, D., Gross, M.: Particle-based fluid simulation for interactive applications. In: *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 154–159 (2003)



Chang-bo Wang is a Professor of Software Engineering Institute, East China Normal University, P.R. China. He received his Ph.D. degree at the State Key Lab of CAD&CG, Zhejiang University in 2006, and received B.E. degree in 1998 and M.E. degree in Civil Engineering in 2002, respectively, both from Wuhan University of Technology. His research interests include physically based modeling and rendering, computer animation and realistic image synthesis and information visualization.



Qiang Zhang is a graduate student of Software Engineering Institute, East China Normal University, P.R. China. He received his B.E. degree in Mathematics & Applied Mathematics from Zhejiang University of Technology. His research interests are physically based simulation and digital entertainment.



Fan-long Kong is a graduate student of Software Engineering Institute, East China Normal University, P.R. China. He received his B.E. degree in computer science from Central China Normal University. His research interest is fluid simulation based on physics.



Hong Qin is a Full Professor of Computer Science in Department of Computer Science at State University of New York at Stony Brook (Stony Brook University). He received his B.Sc. (1986) degree and his M.Sc. degree (1989) in Computer Science from Peking University in Beijing, China. He received his Ph.D. (1995) degree in Computer Science from the University of Toronto. His research interests include computer graphics, geometric- and physics-based modeling, computer-aided design, computer-aided geometric design, computer animation and simulation, virtual environments and virtual engineering.