IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 28, 2019, accepted December 10, 2019, date of publication December 24, 2019,
date of current version January 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962082

Survey on Smoothed Particle Hydrodynamics and the Particle Systems

RUNPING XI[1], ZHANGCAI LUO[1,3], DAVID DAGAN FENG[2], YANNING ZHANG[1], XIAOPENG ZHANG[4], AND TIANYI HAN[1]

[1]School of Computer Science, Northwest Polytechnical University, Xi'an 710072, China
[2]School of Information and Technology, The University of Sydney, Sydney 2006, Australia
[3]Shaanxi Provincial Key Laboratory of Speech and Image Information Processing, Xi'an 710129, China
[4]LIAMA–NLPR, CAS Institute of Automation, Beijing 100190, China

Corresponding author: Runping Xi (xrp@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572405.

ABSTRACT Particle techniques mainly deal with physically-based particle variations for phenomena and shapes in computer animation and geometric modeling. Typical particle techniques include smoothed particle hydrodynamics (SPH) simulation for animation and particle systems for shape modeling. SPH simulation and particle systems are meshfree methods and have been widely applied in engineering and applied sciences. Here, we provide a comprehensive survey on the recent developments, achievements, and future expectations of SPH and particle systems, with a detailed description of the classification and methods used in the field. Particularly, it has not only summarized the cutting-edge research of SPH methods and the particle systems, with a description of basic theories behind the particle system and method of the structuring mode for stimulating the irregular mode, but also proposed measures to improve the particle system. These applied methods mainly include the Lagrangian method, for simulating fluid phenomena, such as smoke, fire, explosion, wave, bubble, and free surface, which have been the focus of little attention in the past. Finally, future research directions and expectations are thoroughly discussed with the detail, acceleration, and control techniques used to meet the pressing real demand for real-time performance and flexibility.

INDEX TERMS Particle technique in modeling and simulation, particle system, smoothed particle hydrodynamics (SPH), fluid simulation, data visualization technology.

I. INTRODUCTION

As a common phenomenon in nature, objects with ''blurred'' shape, such as flame, cloud, spray, smoke and cloud, have no fixed shape and regular geometric shape. More importantly, their appearance changes uncertainly with time. There is a basic law in physics, ''Objects are made up of elementary particles'' Based on this idea, Reeves [6] proposed the particle system in 1983, and it is recognized as the most successful method of graphics generation for simulating irregular blurred objects. The particle system defines tens of thousands of irregular and randomly distributed particles moving continuously and transforming to form the overall shape, characteristics and dynamic changes of the scenery.

Water, smoke and other fluids have always been representative objects in realistic visual scenes. Because of their large degree of freedom, many factors affecting motion and complex physical equations, traditional grid-based methods confine all simulations to fixed grids in advance, and their computations are complex. Compared with the traditional mesh or grid-based numerical modeling methods, the meshfree simulation methods have some special advantages [3] for computer animation. The main idea of the meshfree methods is to provide accurate and stable numerical solutions to integral equations or all types of possible boundary conditions using a set of arbitrarily distributed nodes or particles.

Smoothed particle hydrodynamics (SPH) [1] is a ''truly'' meshfree, particle method was used for continuous scale applications, and SPH may be regarded as the oldest modern meshfree particle method. SPH was developed by Gingold and Monaghan [1] and Lucy [2] initially for astrophysical problems in 1977. However, it is a meshfree Lagrangian method, in which the coordinates move with the fluid, and the application of the method can be easily adjusted with

The associate editor coordinating the review of this manuscript and approving it for publication was Shuhan Shen.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

respect to variables such as density, velocity, acceleration, and energy. SPH has been used in many research fields, such as astrophysics, ballistics, volcanology, and oceanography.

Here, we focus on SPH simulation and the particle system that could be used for computer animation and geometric modeling. In Section 1, the background of particle techniques is introduced. In Sections 2, 3, 4, and 5, the invention, theoretical process features, boundary processing and computational efficiency of SPH are described. In Section 6, the applications of the SPH method are discussed. In Sections 7, 8, and 9, the concepts, ideas and model of the particle system are explained. In Section 10, the applications of particle modeling are discussed. In Section 11, the advantages and main attributes of SPH simulation and the particle system are discussed. In Section 12, the advice and concluding remarks are presented.

II. SMOOTHED PARTICLE HYDRODYNAMICS (SPH)

A. BASIC THEORY OF SPH

SPH was originally developed by Lucy [2], Gingold and Monaghan [1] to simulate astrophysics, but it also has wide applicability in other dynamic simulations. SPH is an interpolation method based on particle system, which enables the physical field defined at discrete points to be evaluated at any position in space. The method uses a radially symmetric smooth kernel function, also known as a kernel function, which distributes the field in the local neighborhood of each particle, which is called the support region. The physical field A at position $\mathbf{r}$ can be calculated by interpolating the weights of all particles in the support domain:

$$A_s(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h) \tag{1}$$

Among them, j denotes any particle in the support domain at the current position $\mathbf{r}$, m_j denotes the mass of particle j, $\mathbf{r}_j$ denotes the position of particle j, ρ_j denotes the density of particle j, A_j denotes a physical field at the position $\mathbf{r}_j$, and $W(\mathbf{r}\text{-}\mathbf{r}_j, h)$ is a smooth kernel function, where h is the radius of the support domain, also known as the smooth core radius. If W is even, standardized and second-order, then there are:

$$\int W(\mathbf{r})d\mathbf{r} = \mathbf{1} \tag{2}$$

In most dynamic equations, the derivative of physical field to position will appear, which needs to be solved. When using SPH method, the derivatives of the physical field loaded by particles only affect the smooth kernel function because the physical field loaded by particles only changes with time from the Lagrangian perspective. The gradient of physical field A can be expressed as:

$$\nabla A_s(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j, h) \tag{3}$$

The Laplace operator of physical field A can be expressed as:

$$\nabla^2 A_s(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h) \tag{4}$$

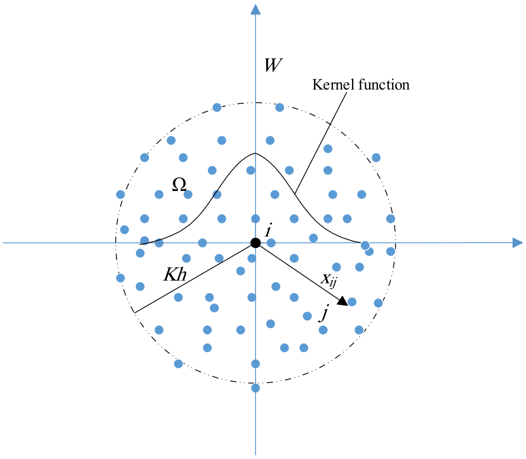

FIGURE 1. Approximate schematic of SPH particles in a two-dimensional space. *W* is a smooth function and supports domain *Kh*, and *p* is the surface of the calculation area. [3].

In SPH simulation, every fixed time step Δt, the position and velocity of each particle will be calculated once for image rendering. After obtaining the acceleration of fluid particles at the current time, the appropriate numerical method can be selected to estimate the velocity and position of particles at the next time, so that the whole simulation process can be continued.

SPH method can deal with the flow system with complex wall, free surface, multi-phase moving interface, and large deformation and breakage of solid relatively simply. As a particle method, mass is strictly conserved. In addition, its concept is concise, it is easy to add various special physical or chemical effects, and it can be directly extended from two-dimensional system to three-dimensional system, which is easy to program. Because of the short-range interaction between particles, it has good parallelism and can make full use of cluster system for large-scale computation.

Figure 1 shows a schematic representation of the SPH particle approximation in a two-dimensional space. Techniques for deriving SPH formulations for the Navier-Stokes (N-S) equations are reviewed in a previous article [3].

B. SMOOTHING FUNCTIONS

In SPH, the smoothing function is used by a kernel for particle approximation. The smoothing function is extremely important in SPH because it determines the pattern to interpolate and define the threshold of the support area of a particle.

Like the mathematical approach in hydrodynamics, the SPH algorithm involves the concept of ‘‘smooth kernel,’’ which can be understood in this manner, and the properties of the particles are ‘‘diffused’’ to the surroundings and gradually become small as the distance increases. The function of attenuation with distance is called the ‘‘smooth kernel’’ function, and the maximum influence radius is the ‘‘smooth kernel radius.’’

When SPH method is used for fluid simulation, the stability, accuracy and speed of calculation depend highly on the selected smooth kernel function. Different smoothing functions have been used in SPH, and their major requirements or properties have been analyzed as follows [3]:

(a) Regularization conditions:

$$\int_{\Omega} W\left(x-x', h\right) \mathrm{d}x' = 1 \tag{5}$$

(b) Tightness:

$$W\left(x-x', h\right) = 0, \quad \left|x-x'\right| > \kappa h \tag{6}$$

where κ is the constant of the smooth function at a point and κh determines the effective range of the smooth function (i.e., the support domain)

(c) Nonnegativity:

In general meshless methods, the weight function can be negative. In the study of hydrodynamic problems, the traditional smoothing function usually uses non-negative function. Negative smoothing functions may lead to negative physical parameters, such as negative energy, density and mass, which violate physical properties in the calculation process..

(d) Attenuation:

The smoothing function value of particles should monotonously decrease with the increase of distance from particles. On the contrary, the interaction force decreases with the increase of distance between two interacting particles.

(e) Dirac Functionality:

$$\lim_{h\to 0} W\left(x-x', h\right) = \delta\left(x-x'\right) \tag{7}$$

(f) Even Functionality:

$$\int_{\Omega} (x-x')w\left(x-x', h\right) \mathrm{d}x' = 0 \tag{8}$$

This property indicates that the same distance from a given particle to different particles at the different positions has the same effect on a given particle.

Lucy in the original SPH paper [2] used a bell-shaped function. Gingold and Monaghan [1] selected Gaussian kernel to simulate nonspherical stars. The most frequently used smoothing function is the cubic B-spline function [5].

Table 1 lists the existing appropriate smooth functions.

C. ACCURACY OF SPH

SPH is usually considered a second-order precision method [50]. On the right side of (3) on the Taylor to start, only to retain the first derivative, you can obtain the following [3]:

$$\begin{aligned} < f(\mathrm{x}) > &= \int_{\Omega} [f(\mathrm{x}) + f'(\mathrm{x}-\mathrm{x}') + r((\mathrm{x}'-\mathrm{x})^2)]\mathrm{W}(\mathrm{x}-\mathrm{x}', \mathrm{h})dx' \\ &= f(\mathrm{x}) \int_{\Omega} W(\mathrm{x}-\mathrm{x}', \mathrm{h})\mathrm{dx}' \\ &\quad + f'(\mathrm{x}) \cdot \int_{\Omega} (\mathrm{x}-\mathrm{x}')\mathrm{W}(\mathrm{x}-\mathrm{x}', \mathrm{h})\mathrm{dx}' + r(\mathrm{h}^2) \end{aligned} \tag{9}$$

where r is the remainder of the Taylor expansion because the smooth kernel function satisfies the regularization and symmetry conditions. The above equation can be written as follows:

$$< f(\mathrm{x}) >= f(\mathrm{x}) + \mathrm{r}(\mathrm{h}^2) \tag{10}$$

However, this second-order accuracy may not be true for all situations. Liu *et al.* [53] analyzed a situation in which a particle is near the boundary of the computational region, the support domain of the particle intersects the computational region, and the support domain is truncated by the boundary. Neither regularization nor symmetry conditions can be satisfied. At this time, the SPH kernel approximation of the function no longer has second-order accuracy.

In terms of kernel approximation, SPH has second-order accuracy. By using SPH to approximate the partial differential equation, the precision ultimately depends on the discrete form of the particle approximation, rather than the continuous form of kernel approximation [53]. At the same time, the distribution of particles, the choice of the smoothing function, and the calculation accuracy of smoothing length have considerable influence. In addition, the boundary processing method directly affects the SPH numerical simulation and calculation accuracy.

III. THEORETICAL PROCESS OF SPH

The simulation of incompressible fluid by SPH can be divided into two methods: weak compressible SPH (WCSPH) [51], and incompressible SPH (ISPH). Monaghan [54] first applied the SPH to an incompressible free surface flow simulation. The traditional WCSPH is simpler than other methods because it uses the artificial state equation to explicitly correlate the fluid pressure with the local density. By choosing the appropriate equation of state to ensure that the Mach number is less than 0.1 in the flow process, flow is achieved in this approximation of incompressibility. Compared with WCSPH, ISPH is a semi-implicit approach whose pressure has a Poisson equation iterative solution to solve the velocity explicitly. The following two methods are introduced.

A. WEAK COMPRESSIBLE SPH

WCSPH addresses the incompressible fluid as a weakly compressible flow by limiting the rate of change in fluid density, typically less than 1%. The time-consuming solver for the Poisson equation is avoided by allowing small, user-defined density fluctuations compared with the commonly employed projection methods that strictly enforce incompressibility. WCSPH has been used in free surface flow and low Reynolds number incompressible fluid simulation. WCSPH is easy to implement in the programming owing to the use of explicit integration. Figure 2 shows the comparison of the simulation results obtained by WCSPH using the improved gas state equation.

TABLE 1. Appropriate smooth functions.

Smoothing function	Formulation	α_d in 1D, 2D, and 3D
Bell-shaped [2]	$W(R,h)=\alpha_d \left\{\begin{array}{ll} (1+3R)(1-R)^3 & R\le 1 \\ 0 & R>1 \end{array}\right\}$	$\frac{5}{4h}\ \frac{5}{\pi h^2}\ \frac{105}{16\pi h^3}$
Gaussian kernel [1]	$W(R,h)=\alpha_d e^{-R^2}$	$\frac{1}{\sqrt{\pi}h}\ \frac{1}{\pi h^2}\ \frac{1}{\sqrt{\pi^3}h^3}$
Cubic B-spline function [5]	$W(R,h)=\alpha_d \left\{\begin{array}{ll} \frac{2}{3}-R^2+\frac{1}{2}R^3 & 0\le R\le 1 \\ \frac{1}{6}(2-R)^3 & 1\le R\le 2 \\ 0 & R\ge 2 \end{array}\right\}$	$\frac{1}{h}\ \frac{15}{7\pi h^2}\ \frac{3}{2\pi h^3}$
Quartic spline [16] [17]	$W(R,h)=\alpha_d \left\{\begin{array}{ll} (R+2.5)^4-5(R+1.5)^4+10(R+0.5)^4 & 0\le R\le 0.5 \\ (2.5-R)^4-5(1.5-R)^4 & 1.5\le R<1.5 \\ (2.5-R)^4 & 1.5\le R\le 2.5 \\ 0 & R\ge 2.5 \end{array}\right\}$	$\frac{1}{24h}$
Quintic spline [16] [17]	$W(R,h)=\alpha_d \left\{\begin{array}{ll} (3\text{-}R)^5-6(2\text{-}R)^5+15(1-R)^5 & 0\le R\le 1 \\ (3-R)^5-6(2-R)^5 & 1\le R<2 \\ (3-R)^5 & 2\le R\le 3 \\ 0 & R\ge 3 \end{array}\right\}$	$\frac{120}{h}\ \frac{7}{478\pi h^2}\ \frac{3}{359\pi h^3}$
Quadratic [18]	$W(R,h)=\alpha_d(\frac{3}{16}R^2-\frac{3}{4}R+\frac{3}{4}) \quad 0\le R\le 2$	$\frac{1}{h}\ \frac{2}{\pi h^2}\ \frac{5}{4\pi h^3}$
Super-Gaussian kernel [5]	$W(R,h)=\alpha_d(\frac{3}{2}-R^3)e^{-R^2} \quad 0\le R\le 2$	$\frac{2}{\sqrt{\pi}}$
New quartic spline	$W(R,h)=\alpha_d \left\{\begin{array}{ll} (\frac{2}{3}-\frac{9}{8}R^2+\frac{19}{24}R^2-\frac{5}{32}R^4) & 0\le R\le 2 \\ 0 & R\ge 2 \end{array}\right\}$	$\frac{1}{h}\ \frac{15}{7\pi h^2}\ \frac{315}{208\pi h^3}$
Poly6 kernel [15]	$W_{poly6}(r,h)=\alpha_d \left\{\begin{array}{ll} (h^2-r^2)^3, 0\le r\le h \\ 0, & \text{oherwise} \end{array}\right\}$	$\frac{4}{\pi h^8}\ \frac{315}{64\pi h^9}$ (for 2D,3D)
Spiky kernel [15] [49]	$W_{spiky}(r,h)=\alpha_d \left\{\begin{array}{ll} (h-r)^3, 0\le r\le h \\ 0, & otherwise \end{array}\right\}$	$\frac{10}{\pi h^5}\ \frac{15}{\pi h^6}$ (For 2D,3D)
Viscosity kernel [15]	$W_{viscoisty}(r,h)=\alpha_d \left\{\begin{array}{ll} -\frac{r^3}{2h^3}+\frac{r^2}{h^2}+\frac{h}{2r}-1, 0\le r\le h \\ 0, & otherwise \end{array}\right\}$	$\frac{10}{3\pi h^2}\ \frac{15}{2\pi h^3}$ (For 2D,3D)

Notes: h is the smooth length that defines the influence or support area of the smoothing function W. R is the relative distance between two particles

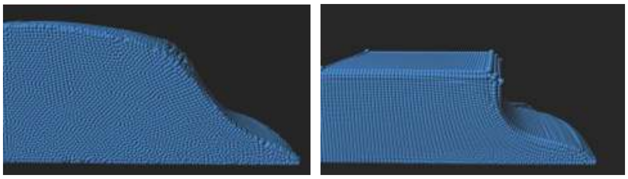

FIGURE 2. Corner-breaking dam using the tait equation (left) and the ideal gas equation (right). The gas equation leads to high compressibility. [51].

B. INCOMPRESSIBLE SPH

In computational fluid dynamics, incompressibility is usually applied by the incompressible SPH (ISPH) [56]–[58] method based on pressure projection, and pressure is obtained by solving the pressure Poisson equation by using the conjugate gradient method. This method has also been introduced into the field of graphics [138]. The ISPH first uses a force other than pressure to predict the intermediate velocity of the particle and then uses the discrete pressure Poisson equation to project the intermediate velocity into the non-dispersion vector space and calculate the pressure and pressure accordingly. The ISPH can allow large time steps and impose strict incompressibility, but it requires considerable time to calculate linear equations in each time step, and the overall computational overhead is high.

C. PREDICTIVE-CORRECTIVE INCOMPRESSIBLE SMOOTHED PARTICLE HYDRODYNAMICS (PCISPH)

PCISPH is a new SPH approach that combines the benefits of WCSPH with those of the ISPH, which performs convergence loops in each physical update, including prediction and cor-

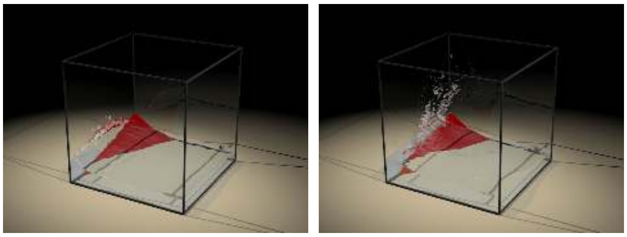

FIGURE 3. Comparison of WCSPH (left, 17k particles) and PCISPH (right, 100k particles) with equal computation times. [59].

rection iterations [59]. In this process, the position and density of a new particle are predicted, and the change in the reference density is calculated. Furthermore, a formula is obtained that correlates density fluctuations and the pressure, reducing the density error and approaching incompressibility. Less consumption is calculated per physical update that allows for great time steps. Thus, the experiment proved that the acceleration obtained by WCSPH was more than an order of magnitude, and good results were obtained. Figure 3 shows the effect of PCISPH and the WCSPH simulation when simulating the same time. The former method allows a large number of examples to be used, and the simulation effect is realistic.

D. IMPLICIT INCOMPRESSIBLE SPH

Ihmsen *et al.* [148] proposed a new implicit incompressible SPH method in 2015. The method takes the density invariant condition as the source term of the pressure Poisson equation, and combines the discretization form of the fluid continuity equation with the pressure term of SPH to obtain the discretized pressure Poisson equation. The relaxation Jacobian method is used to solve the pressure iteratively. IISPH improves the convergence rate of the solver by considering the contribution of pressure in solving the pressure Poisson equation. Different from PCISPH and LPSPH, IISPH updates fluid density information by velocity iteration, which improves the stability of time integration process. IISPH can simulate incompressible fluids with lower compressibility, allowing larger time steps and high scalability. It is very suitable for realistic fluid simulation in large-scale scenarios.

E. OTHER IMPROVEMENTS

Given that the conventional SPH is insufficient, several new high-precision methods were proposed as follows.

Reproducing kernel particle method [60]: a correction function is introduced and applied for regeneration conditions, which satisfy the boundary compatibility conditions and improve the solution accuracy for large deformation analysis.

Moving least squares particle dynamics [61]: the method has a tight support domain of the weight function, which is obtained by the moving least squares interpolation of the shape functions and the shape functions for solving fluid mechanics equations.

Corrected smooth [62] and finite particle methods [63]: by a Taylor series expansion, traditional discrete SPH way is used to describe the correction.

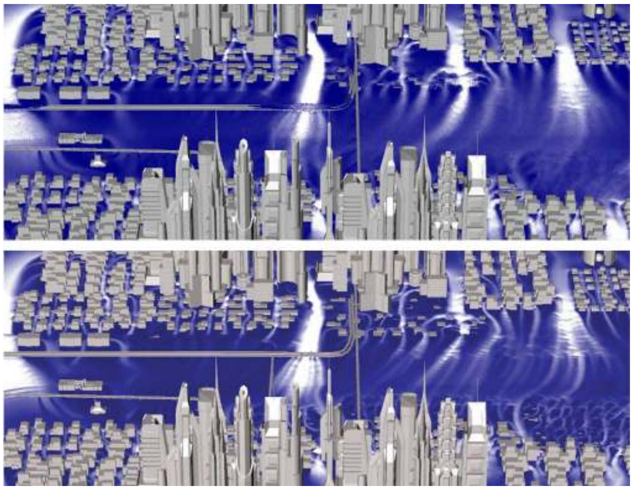

FIGURE 4. A city with an area of 5 × 10^6 m^2 is flooded at low resolution (top), 6 million particles, and high resolution (bottom), 40 million particles. Velocities are color coded [148].

Symmetric smooth particle dynamics [64]: the SSPH method was proven highly suitable for elastic solid simulation by constructing a new basis function. However, the construction of its base function is overly complex, and this complexity should be difficult to popularize to the free surface of the simulation.

IV. BOUNDARY PROCESSING IN SPH

In the application of the SPH, the boundary condition has the advantages of the method and the current weak link because the SPH was originally proposed in the absence of boundary conditions in the field. Later, it was gradually extended to some boundary applications in industries. SPH uses a moveable particle system to represent the contemplated continuum, which creates difficulties for boundary processing and conditions [140].

Presently, no general solutions to solid wall boundary problems are universally accepted. Generally, existing methods cannot easily account for the accuracy of the calculation, the complexity of the calculation of the region, and the flexibility of the calculation. These methods are broadly divided into four categories: boundary force, mirror, reflection, and coupling boundary methods.

A. CLASSICAL SOLID WALL BOUNDARY PROCESSING

The boundary force method works by setting a series of the relative positions of particles around the fixed solid wall boundary, and force is exerted on the flow of particles by boundary particles. This force is repulsive and decreases as the distance increases. The boundary force method can handle complex boundary conditions, but the choice of force is usually only based on experience and is difficult to specify. An excessively small force cannot push the particles through the border, but if the force is too large, then moving particles could penetrate through the border [65].

The mirror particle method works by mapping boundary particles to the solid wall to represent the solid region [33].

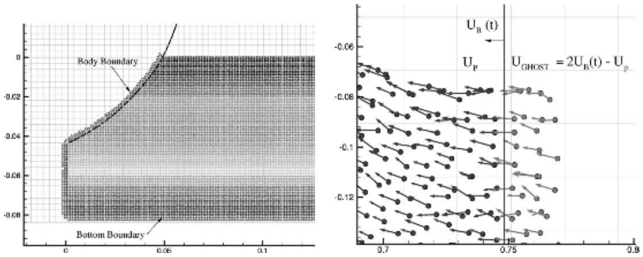

FIGURE 5. Ghost particles. Left: example of curved rigid boundary modeled by ghost particles. Right: enlarged view with the details of the mirrored velocity field [33].

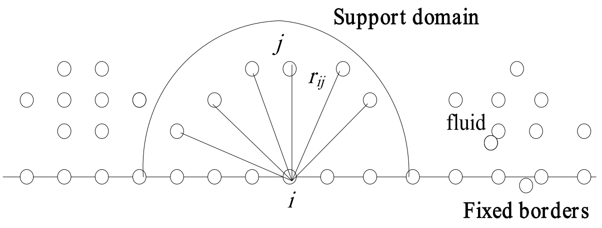

FIGURE 6. Fixed boundary particle method. [67].

The boundary particles have the same mass and density as the corresponding flow particles, with an opposing velocity in the same direction. Figure 5 shows the principal details of themirror particle method. The mirror particle method has a good property of conservation and accuracy, but it is difficult to generalize to a complex and less accurate calculation area.

The coupling boundary method works by converting the solid walls into a series of boundary points, where these boundary points are some properties (mass, density, and pressure) of the fluid mass. In the solution process, the presence of boundary points is akin to the situation to solve a continuous equation or a state equation, such that the boundary problem is coupled with the process of solving an equation, and the explicit boundary condition solution is avoided [66], [3].

B. IMPROVED METHODS

1) FIXED BOUNDARY PARTICLE METHOD

To obtain an exact solution of the pressure on the fixed boundary and to save computation time, Zheng *et al.* [67] proposed a new method, as shown in Figure 6, by fixing the boundary area close to particle point *i*. The pressure at particle *i* is estimated by the particle *j* adjacent to particle *i*.

Figure 6 is a schematic diagram of the principle of the improved method. The distance between the boundary points *i* and *j* is selected if less than *h* (*h* for the smooth length) for all internal particles. Thereafter, we calculate internal particle *j* and boundary particle *i* spacing r_{ij}. We approximate the following formula for pressure point *i*

$$p_{\mathrm{i}} = \frac{\sum_{j=1}^{N} p_{\mathrm{j}}(\mathrm{h} - r_{ij})}{\sum_{j=1}^{N} (\mathrm{h} - r_{ij})} \tag{11}$$

This method does not need to conduct particle search and sorting in program implementation, such that the calculation time could be saved and the method reflects well the actual distance between the two points on the same level of effect owing to the same pressure.

2) IMPROVED COUPLING BOUNDARY METHOD

The pressure on a boundary point obtained by solving the continuous and state equations is not a matter of true pressure. In response to this deficiency, Gong *et al.* [68] improved the method of solving particle pressure, in which the pressure was obtained by setting a pressure interpolation for fluid particles. Thus, the problem that the original coupling boundary method is not allowed has been improved on the boundary. The boundary pressure interpolation method was improved owing to the interpolation method proposed by Oger *et al.* [147].

V. COMPUTATIONAL EFFICIENCY OF SPH

SPH simulation needs a large amount of calculation for the consumption of particles in the interaction. Studying how to improve the efficiency of SPH computation for a large-scale simulation is necessary. Three aspects are worthy of improvement: to improve the efficient adjacent particle search, to improve the construction of efficient parallel computation, and to improve other methods of coupling.

A. NEAREST NEIGHBOR PARTICLE APPROACH

Nearest neighbor search: in SPH, the calculation of the related examples is only related to the particles in the support domain due to the presence of support domains. The particles that are included in these support domains are called the nearest neighbor particles (NNP) of the relevant particles. The process of finding the nearest neighbor particle is called nearest neighbor particle search (NNPS) [141]. The most commonly used NNPS has three types:

- full-matching search method
- linked list search method
- tree search method

The full-matching search method calculates the distance between all the particles and particle *i* in the problem domain, searching for all the part icles in particle *i* support domain. This search process for all particles has the time complexity of $O(n^2)$; thus, the calculation time consumption is large and is not suited to the huge number of particles.

In the realization of a linked list search method [69]–[73], the first problem in the domain lies on a layer of a temporary grid. The grid size should match the size of the supported domain. For given particle *i*, its adjacent particles can only be in the same grid or in close adjacent units. The list search method distributes each particle in the grid cell and concatenates all the particles within each grid with a simple storage rule; its time complexity is $O(n)$ [142]. Figure 7 is a schematic diagram of a two-dimensional space linked list search method. The problem with the chain search is that when the smooth length varies, the grid space cannot be

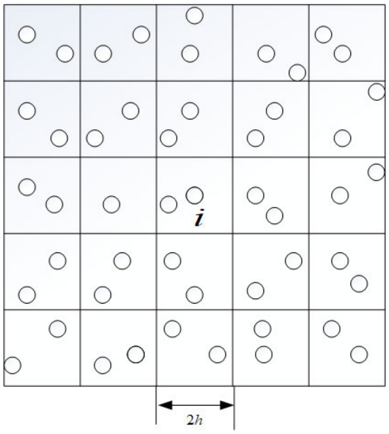

FIGURE 7. Link list search method in two-dimensional space. [70].

adapted to each particle. If the list search method is applied, the search efficiency will be low.

The tree search method [74] recursively divides the largest problem domain into a hexagram until each of all divisions contains only one particle. For any particle i, within the $2\kappa h$ cube around it, if the cube of the tested particle does not overlap with the cube range of particle i, the search will be stopped. When overlap occurs, the search will continue to the next layer until only one particle is at the current node. If particle i is in the support domain of the particle, then a neighboring particle of particle i will be found. The order of its time complexity is $O(N\lg N)$. The tree search method is highly efficient for solving problems with various smooth lengths and large numbers of particles; therefore, it is very suitable for smooth length concept SPH fluid simulation, with strong robustness [143].

B. EFFICIENT PARALLEL ALGORITHMS

Even with the use of efficient neighboring particle search, the calculation remains very time-consuming if the number of particles is huge. In recent years, the use of GPU-accelerated SPH has been a new development of SPH calculations. By means of modern GPUs with multiple cores, highly parallel structure characteristics could be applied. The GPU uses the streaming parallel mode data in operation; that is, a plurality of data can be processed simultaneously [144].

SPH is a meshless particle method; thus, its calculation is by a series of fluid-particle interactions; in this process, each particle of the calculations performed is exactly the same; that is, in different data on the implementation of the same program [144]. Thus, for fluid simulation, SPH has good parallel properties, and using the GPU to achieve parallel computing is suitable. In SPH fluid simulation, all arrays are mapped to the GPU cache; thus, all the particles of the calculation can be performed in parallel. Therefore, particles can be assigned to each GPU thread, fully using the GPU's parallelism. Its performance has been greatly improved, such that real-time simulation becomes possible.

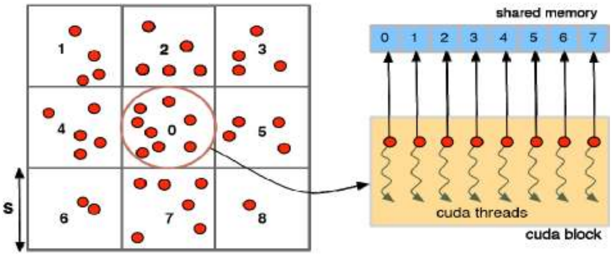

FIGURE 8. Each CUDA thread in a block computes attribute, for one particle and at the same time copies particles from a neighboring block into its shared memory [75].

Parallel SPH implementation generally comprises a physical simulation, including neighborhood search, density, associated force, and final rendering [145], [146]. The current proposed GPU-based particle simulation requires CPU involvement, which uses CPU–GPU computing, and transmission becomes the algorithm's bottleneck. Many scholars have improved this issue. Goswami *et al.* [75] proposed a method by generating a distance field volume as the core to accelerate the rendering process for particle simulation and visualization entirely on the GPU, avoiding CPU–GPU overhead, and good results are achieved. Figure 8 is a schematic diagram of an improved method.

C. COUPLING OF SPH WITH OTHER METHODS

The SPH method can be coupled with the grid method; thus, it has respective advantages. The coupling of SPH and FEM has been very common, which is suitable for solving large deformation problems, such as collision impact [76]–[78]. The SPH method could be coupled with the discrete element method (DEM) to solve the problem of particle and droplet impact [79]. SPH can also be coupled with other particle methods, such as MD [80] (molecular dynamics coupling) and DSMC (direct simulation Monte Carlo) for chemical separation flow simulation.

VI. APPLICATIONS OF SPH

As a pure Lagrangian method, SPH is liberated from the traditional mesh processing method. Particle motion makes the SPH method can be used to simulate various complex flows. It has obvious advantages in dealing with moving boundary problems, especially large-scale distortion and free surface problems. The applications of SPH to some areas are as follows.

A. HIGH SPEED COLLISION AND BLASTING

The original application of SPH is in astrophysical phenomena [4] because celestial motion can also be modeled by simulating a compressible fluid. SPH was introduced to a vast range of problems in both fluid and solid mechanics for a strong capability to incorporate complex physics into the SPH formulations [5], including high explosive detonation and explosion [19]–[23], microfluidics or micro drop dynamics [24]–[31], heat conduction [10], fluid-solid interaction [11], and metal forming [5].

Given that the SPH method has the advantages of the Lagrangian and meshless methods, it is very suitable for simulating complex deformation problems such as an explosion. Zong *et al.* [81] studied the SPH method of simulating two-dimensional fluid underwater explosion. Xiubo studied the detonation simulation using SPH and compared it with the earlier method. Yang *et al.* [82] applied SPH to simulate the near-surface explosion and reproduce the interaction process of detonation products, water, and air.

The current SPH method of the study of the explosion remains only in a relatively simple situation because additional complex surface chemical phenomena cannot be simulated. Furthermore, in the case of water media simulation, the explosive gas particles easily penetrate the water medium, showing a need for effective interface treatment.

B. FREE SURFACE FLOW

Vertical pressure of free surface flow transits continuously and smoothly to zero, but there exists a discontinuous sudden change of density from fluid density to zero below the surface. The boundary conditions on arbitrarily moving surface are special. At present, MAC and VOF methods are ideal methods to deal with this situation, but both of them have numerical diffusion problems caused by convection term in N - S equation. Because of its Lagrangian characteristics, SPH method solves this problem well and is especially suitable for the simulation of free interface flow.

Dam break simulation is one of the earliest applications of SPH in hydrodynamics. The dam break is a type of wave motion simulation problem. In 1994, J. J. Monaghan first created a model for simulating incompressible fluids by simulating dam break problems [54]. Thereafter, many scholars on this issue have sought in-depth studies. Shao *et al.* [83] studied the dam break problem, combined with a large eddy simulation, and discussed the climbing of isolated waves. Colagross and Landrini [33] modeled the flow of water against an upright wall by setting up an upright wall downstream. Gesteira and Dalrymple [66] used WCSPH to simulate the movement of dam-break water flow and analyzed the impact of dam-breakwater waves on high-rise buildings and wave breaking. Jian *et al.* [84] used WCSPH and ISPH to simulate the collision process between dam breakwater flow and moving and fixed obstacles. Figure 9 shows the use of WCSPH to simulate a dam break scenario at different times.

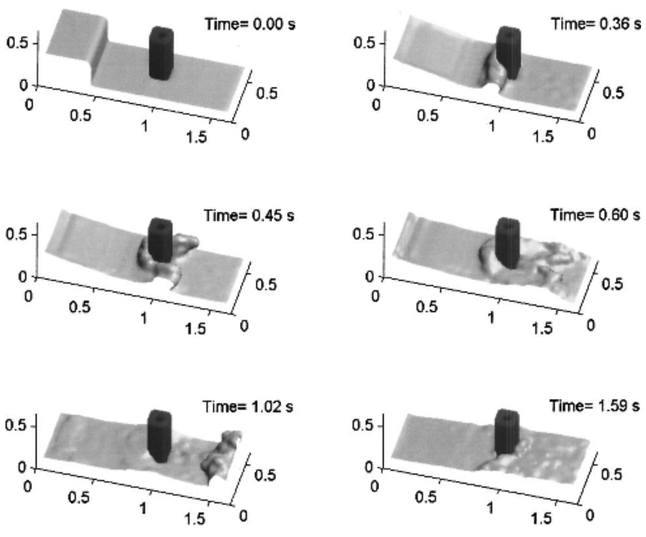

FIGURE 9. **Wave evolution (T = 0.0s) initial configuration; (T = 0.36s) waves hitting the front of the structure; (T = 0.45s) waves wrapping around the structure; (T = 0.60s) waves colliding after passing the structure; (T = 1.02s) wave colliding with the opposite wall of the tank; (T = 1.59s) reflected waves hitting the back of the structure. [66].**

The SPH simulations of incompressible flows are mainly classified as weak compressible models (WCSPH) and strict incompressible models (ISPH). The actual fluid is compressible, and the incompressible fluid is an idealized mechanical model. The liquid compression coefficient is very small, and the density is almost constant in the range of considerable pressure changes. Therefore, the liquid balance and the motion problem all treat the liquid as an incompressible fluid for theoretical analysis. The gas compressibility is much larger than that of the liquid. In all natural atmospheric movements, air velocity is small; it is much smaller than the speed of sound. The flow process density did not change significantly and can still be used as incompressible fluid treatment. For the simulation of incompressible flow in the dam, wave motion, river hydrodynamics, and multiphase flow and its interaction with other structures have been developed by leaps and bounds. WCSPH establishes the relational expression of the density error and the pressure using the Tait equation, ensuring the incompressibility of the fluid by using large stiffness coefficients to control the error. When simulating by ISPH, the pressure of the particle is obtained by the discrete pressure Poisson formula, allowing for a large time step and becoming a common method in computational fluid dynamics and computer graphics. Although the WCSPH and ISPH can be used to impose incompressibility, they have low computational efficiency. In response to this problem, Solenthaler and Pajarola [59] proposed the PCISPH method, which uses split ideas to split the forces on the particles into pressure and other forces, eliminating the limitations of the time step such that it can be used for interactive real-time simulation.

Free surface flow problems are also widespread in the ship, aerospace, water conservancy, civil engineering, and other fields. SPH is also suitable for ship hydrodynamics. The study of ship hydrodynamics mainly includes hull resistance, tank sloshing, and overturning. In the sloshing problem, SPH gradually became one of the main tools in the numerical analysis of the problem. Iglesias et al. used SPH to study the problem of the moment amplitude in the sloshing. Iglesias *et al.* [85] studied the sloshing problem of damping tanks Shao *et al.* [86] used the improved fixed boundary condition to study the sloshing problem. Rafiee *et al.* [87] used the three-dimensional SPH model to study the problem of sloshing and obtained results and compared them with the experimental findings. Turbulence simulation has always been a hot topic in SPH method.. Dalrymple and Rogers [39] improved the SPH algorithm to handle turbulence, the fluid viscosity and

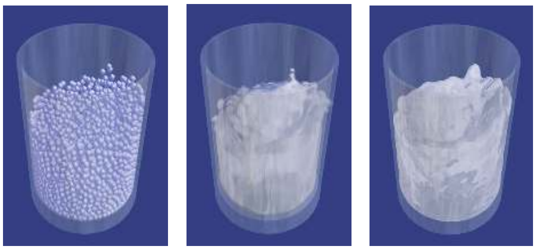

FIGURE 10. A swirl in a glass induced by a rotational force field. (a) Particles, (b) surface using point splatting and (c) iso-surface triangulated via marching cubes. [89].

density, and a different time stepping. The model is shown to be able to model breaking waves on beaches in two and three dimensions, green water overtopping of decks, and wave–structure interaction.

C. INTERFACIAL FLOW

Interfacial flow usually refers to a liquid-liquid and gas-liquid two-phase flow system with a variable phase interface. Fluid and fluid interaction, likewise known as multiphase flow, is highly common. In reality, similar to water and oil mixing, the clear majority of grid-based simulation methods are difficult to manage as systems that contain two or more fluids, and may not even be feasible. For SPH, the addition of a new fluid only means the introduction of a new type of particle, which is relatively simple. Compared with the mesh-based Euler method, particle representation could clearly distinguish the different phase fluid interfaces. Thus, the particle method is suitable for the treatment of the multiphase flow.

Colagrossi *et al.* [33] implemented the SPH method to manage two-dimensional interfacial flows. The results are consistent with those obtained by other solution techniques. Chen *et al.* [14] proposed a particle-based solution to simulate the interaction between blood flow and vessel wall for virtual surgery. By coupling two particle-based techniques, SPH and mass-spring model (MSM), the author seamlessly simulated the blood flow and vessel deformation. Experimental results demonstrate the potential value of the proposed method in providing real-time and realistic interactions for virtual vascular surgery systems.

For the first time, Morris achieved the liquid two-phase flow SPH simulation [88]. However, this model is highly sensitive to the non-uniformity of the particle distribution, and a large error may exist in the correlation calculation. Müller *et al.* [89] were able to simulate the multiphase flow with a low-density ratio by directly applying the SPH formula to different fluid particles. Solenthaler and Pajarola [90] introduced an improved density formula into the Navier-Stokes equations and the calculation formula of other forces, which can be used to solve the problem of false interface effect and to realize the simulation of a stable higher density than incompatible fluid. Figure 10 shows a schematic of the liquid vortex in a glass under direct particle use and different surface extraction methods.

Compared with the liquid two-phase flow, the gas–liquid two-phase flow has a greater density ratio. Grenier *et al.* [91] established a Hamiltonian description for interface and free surface flows, simulating the rise of bubbles in water. Cleary *et al.* [92] used gas dissipation to simulate the bubbles, dealing with the gas and the liquid separately. The bubbles were simulated with a discrete solid of fixed shape, the fluid was simulated using the SPH method, and the effect of the bubble on the fluid was ignored. Hong *et al.* [93] proposed a coupling of meshless and grid methods, simulating gas using a meshless SPH method, and liquid with a grid-based method. The copling between the two methods is achieved by the velocity field. Schechter and Bridson [94] achieved a one-way coupling of gas and water, where the gas particles interacted only with the density field and not with the liquid, except for a passive interaction with the velocity field of the fluid. Ihmsen *et al.* [95] implemented a two-way coupling of gas and liquid, which can be used to simulate the effect of resistance and buoyancy.

An important feature of the fluid-solid coupling problem is the interaction between the two-phase media; the solid moves under the action of the fluid, and the movement of the solid in turn affects the flow field of the fluid [96]. In the fluid-solid coupling problem, the properties of solid materials can usually be divided into two types, rigidity and elasticity. However, serious local errors exist in the Arbitrary Lagrangian-Eulerian (ALE) method, and the Immersed Boundary Method (IBM) does not accurately express the solid shape. Compared with the traditional ALE and IBM, an application of SPH method to deal with the fluid-solid coupling problem is more flexible and convenient.

Monaghan *et al.* [97] first examined the process by which a rectangular object slid down along the slope and collided with the water surface. Oh et al. [98] combined the SPH method with the impulse-based rigid body dynamics to simulate the fluid-solid coupling phenomenon, but the method is not based on hydrodynamics and cannot guarantee non-penetrability. Becker *et al.* [99] used a predictive-corrected method to achieve bi-directional coupling between the rigid body and the fluid, ensuring that the fluid does not cross the solid boundary. However, the conservation of momentum is not guaranteed; thus, the authenticity of results is affected. Shao *et al.* [100] simulated the fluid-solid coupling phenomenon using a method based on velocity-position correction, which ensures momentum conservation and allows the use of a large time step to simulate. Macklin *et al.* [101] proposed a unified particle dynamics simulation framework, based on the location of the dynamics, the interaction between various substances into a unified particle processing, and the flexibility to achieve various object interaction simulation. However, in the above simulation, the particles can only be treated as pure liquid and solid. Yan *et al.* [102] extended the existing multiphase flow framework to achieve the interaction between deformed bodies, solids, particles, multiple fluids, and variable solids, as well as the dissolution of deformable solids. Figure 11 shows the process in which instant coffee and candy pieces are gradually dissolved in water under the action of agitation.

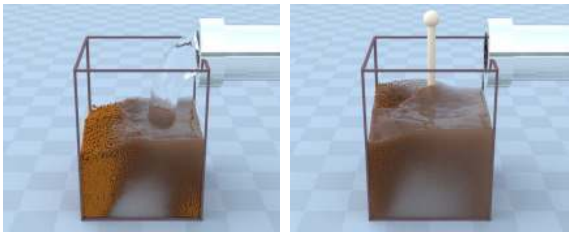

FIGURE 11. Instant coffee and a soft candy dissolving in water. [102].

D. LARGE DEFORMATION PROBLEM

High-speed impact, projectile penetration and space destruction are common phenomena in life and scientific research and production. In the calculation of the impact of such a large deformation problem, the grid method often produces a serious distortion phenomenon. The meshfree method is used to solve the difficulties encountered by the grid method. Batra and Zhang [32] used the modified smoothed particle hydrodynamics (MSPH) method to analyze shear strain localization in elasto-thermo-viscoelastic materials that exhibit strain- and strain-rate hardening as well as thermal softening. Libersky and Petschek [103], [104] first considered the partial stress and elastic-plastic constitutive equation, successfully simulating the high-speed collision problem. Thereafter, Johnson *et al.* [18], [105], [107] conducted a series of work on the high-speed impact problem by improving the SPH method. Benz and Asphaug [108] first used the SPH method with the fracture algorithm and analyzed the related problems. Chen *et al.* [109] simulated the anti-penetration of composites on the basis of the SPH method. The process was simulated, and the results were consistent to the experiment.

Zhou et al. [38] revisited the Taylor-Bar-Impact focusing on the variation of results corresponding to different model parameters that represent varied SPH implementation in a series of three-dimensional computational simulations. The study provided informative data on appropriate SPH implementation options. The normal and oblique hypervelocity impacts of a sphere on the thin plate were likewise investigated and produced large structure deformation. Liu *et al.* [110] used the improved ASPH method to simulate the Taylor collisions and proved its superiority in dealing with large deformation problems compared with the traditional SPH method.

The SPH method has been intensively used for simulating high strain hydrodynamics with material strength due to its special features of meshfree, Lagrangian, and particle nature. L.D. Libersky and his colleagues [35]–[37] carried out the pioneering work of applying the SPH method to high strain hydrodynamic problems, including hyper velocity impact (HVI), fracture, and fragmentation. The research group modeled high velocity impacts ranging from the very small to very large size.

VII. PARTICLE SYSTEM

In 1983, Reeves put forward the concept of particle system for the first time [111]. The term particle system referred to a computer graphics technique that uses a large number of tiny sprites or other graphic objects to simulate certain kinds of ‘‘fuzzy’’ phenomena, which are otherwise difficult to reproduce with conventional rendering techniques—usually highly chaotic systems, natural phenomena, or processes caused by chemical reactions. The commonality of these objects is that they have no fixed shape, no regular geometric shape, and more importantly, their appearance changes with time uncertainty. The particle system uses ‘‘particles’’ to simulate objects in nature, and it may be a small ball, ellipsoid, cube, or other shapes. The size and shape of the particles vary with time. Other properties and particle transparency, color, and movement vary randomly. In addition, the particle movement path can be described in a kinematic manner or by a given force such as a gravity field. In this study, the focus is not on the particle shape in a particle system, but rather on the shape and changes of the simulated object. The particle system is one of the most successful graphics generation algorithms in simulating irregular fuzzy scenes. Numerous natural phenomena can be simulated, including fire, explosion, smoke, flow, leaves, cloud, smog, snow, dust, meteor track, or even nonobjective vision effects such as glow tracks.

After 30 years of development, the particle system gradually formed a set of feasible theory and methods. On the basis of its different applications, the particle system is divided into random, fluid, and directional particle systems as well as structured example system [112]. David and Englandstraat [113] summarized the work of predecessors, and divided the particle system into the independent, fixed connection, and the dynamic coupling. In his paper, Chen [114] divided the particle system into simple and structured according to the particle composition. According to the positional relationship between particles, the particle system can be divided into two types: continuous and discrete. Among them, different classification methods represent different objects that are most suitable for certain simulation. For instance, due to the random attribute of particles, random particle system is particularly suitable for the simulation of natural scenes and irregular objects. Directional particle systems handle interactions and therefore are often used for rigid body and textile simulation. The structured particle system is used to simulate objects or phenomena with a certain structure, such as virtual crops.

VIII. BASIC IDEA OF PARTICLE SYSTEM

Particle system is a method for irregular blurred object modeling and image generation. It was proposed by Reeves in 1983. This method uses a completely different set of methods from previous modeling and rendering systems to construct and render scenery. Scenery is defined as a group of thousands of irregular and randomly distributed particles. As a result, each particle has a certain life cycle, and constantly moves and transforms its shape at every moment. The whole shape, characteristics and dynamic changes of the scenery are formed by the aggregation of many particles rather than individual particles. Particle system fully reflects the dynamics, randomness and discreteness of irregular fuzzy objects. It can simulate natural scenes such as fire, cloud, water, forest and wilderness well. Therefore, it is recognized as the most successful method of graph generation for simulating irregular fuzzy objects.

The particle system theory consists of the following components [116]:

(a) Material particle composition assumptions. The particle system represents the object as a collection of particles, which fill the space and continue to move.

(b) Particle independent assumptions. First, particles do not intersect with other particles in other scenes; second, the particles have no interaction.

(c) The particle property hypothesis. Each particle in the system is not abstract and has a series of column attributes, including quality, spatial location, appearance (such as color, brightness, shape, and size), sports (such as speed and acceleration), and survival (life). The speed attribute, including position, color, and brightness, may change over time.

(d) Particle life mechanism. Every particle in a particle system has a certain life cycle within a certain period, the particles undergo three basic life processes: birth, activity, and demise.

(e) Particle movement mechanism. Particles in the survival of a movement always follow a certain manner. Particle motion attributes are likewise different for the simulation of different scenes.

(f) Particle rendering algorithm. The authenticity of the physical simulation requires the rendering of generated particles to composite the visual effect.

IX. BASIC MODEL OF PARTICLE SYSTEM

Particle system is composed of a large number of simple voxels called particles. These particles have their own set of attributes, such as position, speed, color, size, life cycle and so on. Every particle has to undergo a complete life cycle: generation, movement and death. Particles are usually generated in a designated area by a random process, which constantly updates their properties and eventually dies. Because of the continuous movement of particles, the simulated scene has a certain dynamic nature, so using particle system to simulate irregular objects will have a unique effect.

In general, the basic steps of particle system simulation are as follows:

A. PARTICLE GENERATION

For a fuzzy object, particles in the system are randomly generated within a predetermined location termed the *generation shape*, which may change over time. The number of generated particles is important as it strongly influences the density of the fuzzy object. In the model design, the actual number of particles generated at frame f could be specified as follows:

$$N_{Partsf} = M\text{ean}_{Partsf} + Rand() * Var_{Partsf} \tag{12}$$

where $Rand()$ is a random number between -1 to $+1$, $Mean_{Partsf}$ is the mean number of particles, and Var_{Partsf} is its variance.

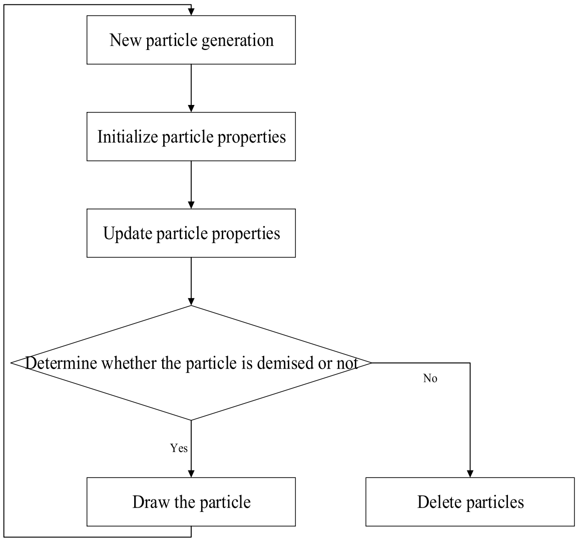
New particle generation
Initialize particle properties
Update particle properties
Determine whether the particle is demised or not
No
Yes
Draw the particle
Delete particles

FIGURE 12. Particle generation.

The number of new particles may depend on the screen size of the object. The corresponding equation is the following:

$$N_{Partsf} = (M\text{ean}_{Partsf} + Rand() * Var_{Partsf}) * ScreenArea \tag{13}$$

where $Mean_{Partsf}$ is the mean number of particles per screen area, Var_{Partsf} is its variance, and $ScreenArea$ is the particle systems' screen area.

In addition, we can use a simple linear function

$$\text{Mean}_{Partsf} = \text{InitalMean}_{Partsf} + \text{Delta}MeanParts(f - f_0) \tag{14}$$

where f is the current frame, f_0 is the first frame during which the particle system is alive, $InitialMean_{Parts}$ is the mean number of particles at this first frame, and $DeltaMean_{Parts}$ is its rate of change.

Therefore, the designer specifies f_0 and any of the parameters$InitialMean_{Parts}$, $DeltaMean_{Parts}$, and Var_{Partsf} to control the particle generation of a particle system. Figure 12 shows a schematic of the particle generation process.

B. PARTICLE ATTRIBUTES

For each newly generated particle, the system must specify its certain attribute. When a particle is created, the distribution of its properties should be random. Otherwise, they may become highly consistent and cause an extremely regular ultimate effect. In different systems, the particle properties should vary but generally include location, color, sports, and survival attributes [117], [130], [131], [133], [134].

(a) Location attribute: A particle system has a location controlled by the initial position of its particles. This attribute contains its initial position coordinates and its direction. The coordinates (x, y, z) of the particles in the three-dimensional space determine its location. The two-dimensional vector

$(angx, angy)$ determines the rotation angles of the particles in the local coordinate system from the x to the y axis.

(b) Color attribute: The initial color of a particle is specified as its primary color value (RGB) and its transparency value. Particle transparency and size are likewise determined by this method

$$\begin{aligned} InitialColor(R, G, B) &= MeanColor(R, G, B) \\ &\quad + Rand() * VarColor(R, G, B) \\ InitialColor(Alpha) &= 1.0. \end{aligned} \tag{15}$$

When the system simulates an irregular object, the initial transparent value can be set to 1.0, that is, opaque. The necessity for transparency change can be determined by the following formula

$$\begin{aligned} InitialTransparency &= MeanTansparency \\ &\quad + Rand() * VarTransparency \end{aligned} \tag{16}$$

(c) Speed property: A typical particle can be implemented by a sphere of radius r. The initial direction in which a new particle moves should be described, and the initial speed of a particle is determined by the following:

$$IntitalSpeed = MeanSpeed + Rand() * VarSpeed \tag{17}$$

where *MeanSpeed* is the mean speed and *VarSpeed* its variance.

(d) Survival attributes: The particle's living attribute is the life parameter, which determines its time of existence in the system. The unit of time can be the number of frames or the system's clock cycle. Using the number of frames to describe the benefits of particle life is straightforward and easy to implement because the particle life in the T-th frame is given by the following:

$$Lifetime(T) = InitialLifeTime - T * Attenuationperframe \tag{18}$$

where *InitialLifeTime* is the initial lifetime of the particle and is set by the system when the particle is generated, and *Attenuationperframe* is the rate of particle life decay per frame, representing the *number* of lives reduced by one particle per frame. The disadvantage of using the number of frames as a particle lifetime unit is that the complexity of generating the graph affects the particle lifetime and hence the uncertainty of the time unit.

Another means to describe the life cycle of a system is to use the system clock to interrupt the lifetime unit of the particle. The advantage of this method is that the unit of time is independent of the complexity of generating graphics primitives and removes the dependency of the number of frames as the unit of time on the background complexity of rendering. Thus, the realism of presentation is improved accordingly. However, the disadvantage of this method is that the system needs to set up a message interrupt for the clock of the particle for the particle lifetime. Each time the clock is interrupted, the system modifies the parameters for each particle, which increases the system overhead and reduces the real-time performance to a certain extent.

C. PARTICLE MOTION

Individual particles within a particle system move in 3D space and change over time in their attributes [134]. To move a particle from one frame to the next is a simple matter of adding its velocity vector to its position. An acceleration factor can be used to modify the velocity of its particles. With this parameter, the model designer can simulate the gravity phenomenon and cause the particle to move in parabolic arcs rather than in straight lines. The particle color, transparency, and size change over time in the same way as prescribed by the rate-of-color-change parameter.

D. PARTICLE EXTINCTION

When a particle is generated, its lifetime measured in frames could be specified. A particle will be demised when its lifetime reaches zero or a threshold value. In addition, other criteria may cause the premature termination of a particle [6]:

(a) Running out of bounds - If a particle moves out of the viewing area and does not reenter it, then there is no reason to keep the particle active

(b) Hitting the ground-particles that run into the ground are assumed to burn out and can no longer be seen.

(c) Some attribute reaches a threshold - For instance, if the particle color is so close to black that it will not contribute any color to the final image, then it can be safely demised.

E. PARTICLE RENDERING

Once the parameters of all particles are calculated, the rendering algorithm starts. The general particle-rendering problem is as complicated as the rendering of objects composed of numerous common graphical primitives.

The methods of rendering are likewise different from one another due to the different forms of the particles themselves [115], [133], [135].

1) POINT PARTICLE MAPPING

In the particle system, a large number of particles are used to ensure realistic simulation but simultaneously consumes enormous time and space consumption for calculation and rendering. Therefore, the following are assumed to improve the rendering speed in the Reeves-based particle system:

- All particle systems include particles that are assumed to be a point source. This avoids numerous complicated operations, such as the shadow of a single particle being no longer considered, regardless of the particle reflection on light. Particles that map to the same pixels in the image are additive, and the pixel color is simply the sum of the color values of all the particles that map to it. No hidden surface algorithms are needed to render the image due to such assumption; the particles are simply rendered in order.

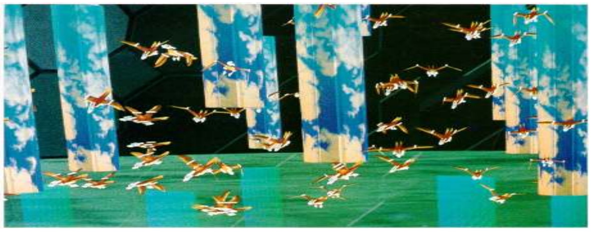

FIGURE 13. Natural scenes simulated by surface particles. [128].

- The relationship between particles, such as collision and fusion, are ignored. Collision detection between particles is a very complex problem because this study focuses on the overall shape of the particles. Such detection is neither realistic nor necessary.

2) SURFACE PARTICLE RENDERING

To show great spatial information, Reynolds [128] proposed the method of surface particles model as very small patches that reflect the directional light source and are used to construct a discrete stream face by face. Each face particle has its own geometric and temporal properties. A normal is added to the face particle that is related to the type of display face, which in turn, is the shape of the particle source and the release time of particles to decide to calculate the particle lightness. The particle source defines the starting position, and the initial particles can be regularly or randomly distributed over the length of the line segment, polygon region, or the entity face. The particles may be continuously or discretely released at regular or random intervals, and the face transparency depends on the particle density. If the particle source continuously releases the spot particles, then a streamline is generated; if the particle source continuously releases the particle particles, then a flow surface is created. Brightness and the color of surface particles are first calculated before rendering, and then particle positions are transformed into the screen space and finally rendered after the scan conversion. Figure 13 shows the natural scene generated using the surface particle method simulation.

3) LINEAR PARTICLE RENDERING

Simulations propose a general and flexible method of drawing particles that is suitable for parallel implementation. Each particle is defined by a header and a trailer, with their own position, radius, color, opacity, and other attributes. The shape of a moving particle consists of two circles, as shown in Figure 14. The middle is connected by a tangent. All parameters can be obtained by a linear interpolation from start to end. The opacity gradually decreases from 1.0 at the center of the disc to 0.0 at the edge by a linear or Gaussian function. When linear particles are rendered, the head and tail positions as well as radii of the particles are first converted to the screen coordinate system. The particles are then divided into pixel-level substrates that contain color, transparency, and depth information, which are subsequently sorted by depth for hidden surface removal. The color information of the pixel is finally displayed. Figure 14 is a schematic diagram of the linear particle method.

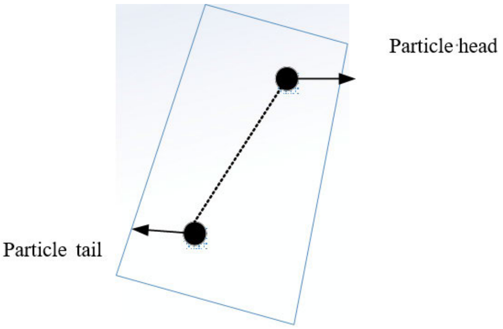

FIGURE 14. Linear particle diagram. [133].

4) RENDERING OF RANDOM SHAPE PARTICLES

Hin and Post [129] used the particle method to display the three-dimensional turbulent flow field. Starting from the particle trajectory, the turbulence is decomposed into forward and turbulent motions. The forward motion is directly described by an average velocity field. The turbulent motion consists of one vortex diffusion equation that describes each step of the particle path generation. A perturbation determined by the local eddy diffusivity is added, which results in a random particle motion. The overall behavior of many particles can be used to simulate an excellent turbulence effect.

X. APPLICATIONS OF PARTICLE SYSTEM

The application based on particle methods is extensive. Since the 1980s, the particle system has been used to model wall fire, explosions, fireworks, and grass. The achievements of particle methods are evident in the simulation of fuzzy and irregular objects. For instance, in the explosion effect of the 1982 film "Star Trek II: The Wrath of Khan", the dynamic particle system concept was first used for the simulation of galactic explosion.

Natural phenomena are particularly suitable to simulate by using particle methods due to their irregularity, dynamic, randomness, and other characteristics. In 1985, Reeves and Blau [115] improved the particle system and realized the simulations of trees and flowers drifting with the wind. On the basis of the theory of Reeves' particle system, Hmonen and Konikanen [48] provided a second order particle system, which presented uniformity and is easily used to simulate phenomena such as smog, cloud, and explosion.

In 1986, Fournier and Reeves [116] used the particle system to simulate waves for the wave model and proposed the simulation conditions. In 1998, Unbescheiden and Trembilski [43] used the particle system by starting from the physical principles of the cloud and combining texture mapping tecniques to establish a cloud model. However, particle numbers are reduced to decrease the computation and rendering times. Through the establishment of velocity field in the three-dimensional framework of flame, Beaudoin *et al.* [118] used particles to render a three-dimensional skeleton that simplifies the movement of flame particles and simulates the flame combustion process. Csuri *et al.* [119] used particles to simulate the smoke that appears in chimneys without using kinetics and stochastic control methods. In 1999, on the basis of the particle system, Xie *et al.* [117] proposed a rain and snowflakes landing real-time simulation algorithm. Hu and Li [44] designed and implemented a waterfall in 3D dynamic terrain with alterations of range and density of particles effects. Kong [125] used the Bezier curve in generating the veins and edge curves to form the basic framework of the leaves of aquatic plants, and the plants are formed close to the natural state.

In addition to the simulation of natural objects and events, the particle system has likewise been effective in simulating life. In 1990, Sim [120] studied the particle animation and rendering algorithm. The proposed parallel particle rendering system could draw different particle shapes, sizes, colors, and transparencies using the parallel property of particle system. Goss [121] proposed a real-time simulation method of the wake of a ship based on the particle system. In 1992, Loke *et al.* [42] modeled, rendered, and animated realistic firework displays by using the particle attributes of fireworks, in which a rendering engine was described that consists of various modules that individually handle a particle property. With the Mass-Spring Model proposed by Provot [45], a cloth object is first approximated to a deformable surface composed of a network of masses and springs, the movement of which is evaluated using the numerical integration of the fundamental law of dynamics. In 1996, Eberhardt *et al.* [122] proposed a fast and flexible method for modeling drape drains. The method is faster than several earlier methods and allows the simulated behavior to change over time. The trajectories of particles were calculated in the system and not just their final positions. In 1998, Wan *et al.* [126] proposed a method of simulating real-time fountain flow based on fluid mechanics and particle system, obtaining a realistic visual effect of fountain flow. In 2003, Shi *et al.* [127] analyzed the Chinese ink painting materials and put forward a simulation model on the basis of the particle system to realize the typical artistic effect of Chinese ink painting. Moreover, Sims [40] provided a method on particle animation and rendering using data parallel computation. Stolk and Van Wijk [41] presented surface particles for 3D flow visualization. Figure 15 shows the life scene simulated using the parallel particle rendering system.

Plant simulation has always been an exciting but difficult point in computer graphics research. The Iteration Function System (IFS) and L-system are typical simulations, but the IFS method is based on the idea of fractal rendering, making the simulation of the specified plant model difficult. Meanwhile, using the L-system, determining the initial and last plant growth is difficult. The trajectories of particles are simulated in the particle system, and the characteristics of the interaction between the plots of the structured particle system are used to simulate the leaves with greater accuracy. At the same time, due to its expansion, the particle system can be macroscopic and be used to vividly simulate the natural landscape of forest grassland. Reeves and Blau [115] used the particle system to simulate grassland as a background environment. Literature [46] [47] based on particle system proposed the simulation of plant organs and their growth model. In 1996, Song and Lai [123] used the particle system to simulate trees. The study achieved good simulation results of tree height, width, branches, and leaves. Wang and Lai [124] proposed a framework based on the simulation of tree skeletons for the static background of the model, and various forms of trees are simulated. Figure 16 shows a tree growth map simulated using the particle method.

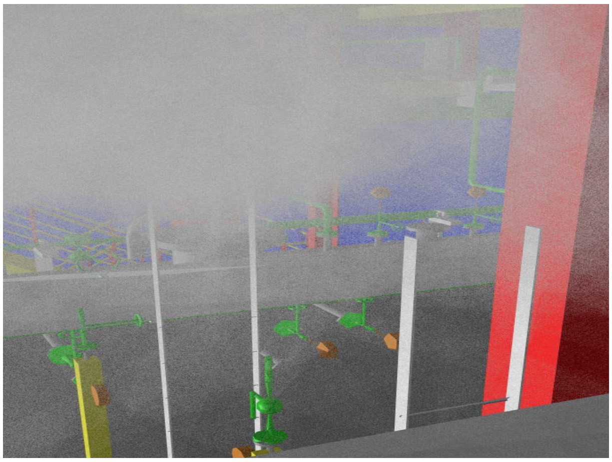

FIGURE 15. The user is standing 'inside' the gas stream. The gas cloud density in the upper part of the picture makes it impossible to see the objects behind. [120].

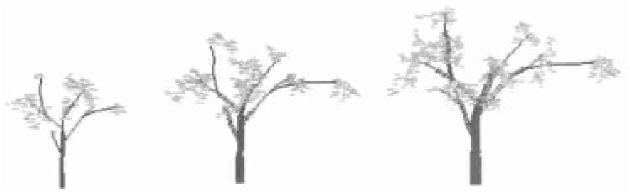

FIGURE 16. Simulation of tree growth. [124].

Surface particle is a combination of particle tracing and shaded surface rendering, thereby capturing the advantages of both: surface interpretation and a moving particle texture. A surface particle is very small and flat, modeled as a point with a surface normal. Their applications in 3D flow visualization are described [11]. The stream surface can be considered a surface traced by particle trajectories.

XI. DISCUSSIONS ON SPH AND PARTICLE SYSTEM

Objects are made of fundamental particles. Natural objects can thus be modeled with several particles. The particle system can be used to model irregular natural scenes in different environments. The randomness of particles can be used to simulate dynamic scenes and achieve good results. In theory, accurately describing objects and their dynamics is possible with enough particles.

SPH is an interpolation method used for continuous scale applications. A simple division of SPH and Particle system is that SPH focuses on modeling shape with weakly free particles, like flows, and particle system works on modeling shape with strongly free particles, like fuzzy phenomena, so they are all useful for computer animation and geometric modeling.

As a meshless method, SPH is likewise classified as a ''particle'' method because of its Lagrangian characteristics. With the continuous improvement of the calculation accuracy and stability, the SPH method has been widely used in numerous science and engineering fields. This method uses particle dispersion and represents the simulated medium to estimate the governing equation of the entire approximate medium motion based on the particle system. SPH is essentially one of, and has a simulation process similar to, the particle methods. Here, the particle system and SPH method are described and collectively referred to as the particle method. The advantages and disadvantages are summarized below.

Compared with the traditional physical method modeling, the particle system has its own unique **advantages** [136]:

(a) Choice of flexibility: The particle system is a set of a large number of particles or geometric primitives combined to form large-scale natural scenes. For different scenes, the number of particles and the particle primitives have considerable flexibility in the choice. Points or variability can be used as the basic primitives, such that the strict polygon approach of traditional graphics may not be necessary. Of course, the option remains for using 10,000 particles to simulate large-scale explosions in the battlefield. We can likewise simulate the flame of a candle with 3,000 particles. Depending on the number of particles and the size of scene, these flexible processes are simplified in algorithms and in real-time.

(b) Dynamic flexibility: Different from traditional modeling methods, the modeling of particle system is a dynamic process that abandons the original static system. Particle systems can be used to simulate the dynamic characteristics of objects. The particles change over time, such as their displacement, velocity, color, and shape. These changing properties of particles cause the overall system morphology to change.

(c) Stochastic flexibility: The operation of the particle system is in accordance with the random process. The values of various particle properties are randomly generated within a certain range, and various properties randomly change over time. Adjusting the parameters can change the visual effects.

The particle system has the following disadvanges:

(a) The number of particles in a particle system directly affects the real-time performance of rendering. Several large-scale scenes require rendering numerous particles. Therefore, the overall system cost is huge, and real-time performance is reduced. Therefore, in the simulation of large-scale objects in the scene, the number of particles must be grasped to balance authenticity and real-time, which likewise requires numerous experiments to verify the completion.

(b) Particle system simulation mainly aims at irregular objects. The particle system method may not be able to achieve the best results and, on the contrary, may lose several details. Thus, the simulation of irregular objects and of changes with details will be a big challenge.

The most important advantage of particle models is their dynamics: good dynamics are the key to making objects look real. Particle number is another, and the problem with the effect and computation in particle modeling. In natural scenery simulation by particle modeling, authenticity and real-time are conflicting requirements. In particle modeling, realism and real-time must be dynamically balanced to obtain the best state that could be solved on the GPU and with improving performance in computing hardware. Particle modeling method has a great potential in numerous engineering and science applications. However, designing an interactive system with real-time and completely true reflection of the nature of various flow phenomena in the computer remains challenging. Numerous issues are worthy of further study. Work on the particle modeling for realistic and real-time simulation with economic computation cost strategies is necessary to speed up the entire calculation. GPU programming, LOD technology, particle swarm theory, collision detection between particles and other objects, simulation of physical mechanism and processes, as well as rendering improvements are promising. The performance of particle modeling could be explored in the coming work in this topic. Future work should similarly be developed, including detail, acceleration and control techniques, to meet the demands for realism, real-time, and flexibility.

XII. CONCLUSION

The idea that matter consists of very small particles called molecules or atoms is widely accepted. In many research fields, modeling substances by particles is significant not only to use particles as the computational frame for interpolation or differencing, but also to carry material properties. The particle is the basic and tiny element of objects, carrying material properties to model or simulate the natural material or natural phenomena. The recent developments of numerical algorithms for particle modeling and applications have been reviewed. Particles can be used to simulate numerous objects in different situations. SPH and the particle system are meshfree particle modeling methods. Through the study and research based on previous works, this article summarizes the fundamental methods of modeling and simulation based on particles.

In this paper, the basic ideas of the SPH method and the particle system are reviewed, and an overview of their numerical methods and applications is provided. For the SPH method, its basic idea is presented, different smooth kernel functions are reviewed, and the latest theoretical progress is described. Several numerical methods are studied, including the boundary treatment method and the computational efficiency of the SPH method. The application of the SPH method is highlighted, including different applications of compressible and incompressible fluids as well as elasto-plastic problems. The successful applications of the SPH method are likewise shown. The basic idea of the SPH method is similar to that of the particle system. As an important aspect of particle system, the main content of the particle method is introduced, and at the same time the concrete particle system steps are elaborated. As with the SPH method, a specific overview of its application is necessary.

The particle method has achieved good simulation results at the macro and micro levels, but numerous shortcomings remain, which can be divided into real-time and visualization effects, which can be improved from the following aspects:

(a) The GPU parallel method is used to accelerate simulation. The simulation speed of the particle method is quite slow when the number of particles is large. Each particle has the same attributes, and the calculation process of each particle is identical and relatively independent. These characteristics are highly suitable for accelerated calculation using GPU. Presently, CUDA is widely used in the field of graphics to accelerate research on the basis of the SPH method, and the related calculation process of particles is handed over to GPU to realize large-scale real-time simulation.

(b) High-quality surface reconstruction and physics-based rendering. From the perspective of realistic simulation, the authenticity of using particle technology to simulate objects requires improvement. On the one hand, the surface of an object must be precisely reconstructed to achieve the real-world effect. On the other hand, the simulation of various optical phenomena is necessary to make the fluid simulation realistic, such as the reflection, refraction of light, generation of bubbles, and other physical phenomena everywhere. However, the addition of these effects requires significant additional computational effort, which poses a challenge to the real-time simulation of fluids. The simulation of these physical phenomena has been likewise one of the hot topics of fluid simulation in recent years.

(c) Multiphase flow simulation. Two-way coupling simulation is common in life, such as boats moving with the surge of waves or skin balls floating on the water surface. Multiphase flow simulation could considerably increase the simulation authenticity.

(d) Coupling of variable solids with fluids. In recent years, the coupling between variable objects and fluids has been likewise worthy of attention, such as the dissolution of instant coffee and the deformation of hair in water. These problems have great complexity, and their simulation is similarly a hot topic for researchers.

REFERENCES

[1] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics: Theory and application to non-spherical stars," *Monthly Notices Roy. Astron. Soc.*, vol. 181, no. 3, pp. 375–389, 1977.

[2] L. B. Lucy, "A numerical approach to the testing of the fission hypothesis," *Astron. J.*, vol. 82, pp. 1013–1024, Dec. 1977.

[3] M. B. Liu and G. R. Liu, "Smoothed particle hydrodynamics (SPH): An overview and recent developments," *Arch. Comput. Methods Eng.*, vol. 17, no. 1, pp. 25–76, 2010.

[4] J. J. Monaghan and J. C. Lattanzio, "A simulation of the collapse and fragmentation of cooling molecular clouds," *Astrophys. J.*, vol. 375, pp. 177–189, 1991.

[5] J. J. Monaghan and J. C. Lattanzio, "A refined particle method for astrophysical problems," *Astron. Astrophys.*, vol. 149, no. 149, pp. 135–143, 1985.

[6] W. T. Revees, "Particle system—A technique for modeling a class of fuzzy objects," *ACM Trans. Graph.*, vol. 2, no. 2, 1983, pp. 91–108.

[7] W. T. Revees and R. Blau, "Approximate and probabilistic algorithms for shading and render structured particle system," *ACM SIGGRAPH Comput. Graph.*, vol. 19, no. 3, pp. 313–322, 1985.

[8] R. Goldstein and G. Wainer, "Simulation of a presynaptic nerve terminal with a tethered particle system model," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Sep. 2009, pp. 3877–3880.

[9] E. M. Husni, K. Hamdi, and T. Mardiono, "Particle system implementation using smoothed particle hydrodynamics (SPH) for lava flow simulation," in *Proc. Int. Conf. Electr. Eng. Informat., United States, IEEE Comput. Soc.*, Aug. 2009, pp. 216–221.

[10] Y. Liu, X. Liu, H. Zhu, and E. Wu, "Physically based fluid simulation in computer animation," *J. Comput.-Aided Des. Comput. Graph.*, vol. 17, no. 2, p. 5281, 2005.

[11] J. Stolk and J. J. Wijk, "Surface-particles for 3D flow visualization," in *Advances in Scientific Visualization*. Berlin, Germany: Springer, 1992, pp. 119–130.

[12] M. B. Liu, G. R. Liu, and K. Y. Lam, "Constructing smoothing functions in smoothed particle hydrodynamics with applications," *J. Comput. Appl. Math.*, vol. 155, no. 2, pp. 263–284, 2003.

[13] H. S. Fang, K. Bao, and J. A. Wei, "Simulations of droplet spreading and solidification using an improved SPH model," *Numer. Heat Transf. A, Appl.*, vol. 55, no. 2, pp. 124–143, 2009.

[14] J. K. Chen, J. E. Beraun, and T. C. Carney, "A corrective smoothed particle method for boundary value problems in heat conduction," *Int. J. Numer. Methods Eng.*, vol. 46, no. 2, pp. 231–252, 2015.

[15] M. Muller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, 2003, pp. 154–159.

[16] J. P. Morri, *Analysis of Smoothed Particle Hydrodynamics with Applications*. Melbourne, VIC, Australia: Monash University, 1996.

[17] J. P. Morris, "A study of the stability properties of smooth particle hydrodynamics," *Publications Astronom. Soc. Aust.*, vol. 13, no. 1, pp. 97–102, 1996.

[18] G. R. Johnson, R. A. Stryk, and S. R. Beissel, "SPH for high velocity impact computations," *Comput. Methods Appl. Mech. Eng.*, vol. 139, nos. 1–4, pp. 347–373, 1996.

[19] J. W. Swegle and S. W. Attaway, "On the feasibility of using smoothed particle hydrodynamics for underwater explosion calculations," *Comput. Mech.*, vol. 17, no. 3, pp. 151–168, 1995.

[20] M. B. Liu, G. R. Liu, K. Y. Lam, and Z. Zong, "Smoothed particle hydrodynamics for numerical simulation of underwater explosion," *Comput. Mech.*, vol. 30, no. 2, pp. 106–118, 2003.

[21] M. B. Liu, G. R. Liu, Z. Zong, and K. Y. Lam, "Computer simulation of high explosive explosion using smoothed particle hydrodynamics methodology," *Comput. Fluids*, vol. 32, no. 3, pp. 305–322, 2003.

[22] A. Alia and M. Souli, "High explosive simulation using multimaterial formulations," *Appl. Thermal Eng.*, vol. 26, no. 10, pp. 1032–1042, 2006.

[23] H. U. Mair, "Hydrocodes for structural response to underwater explosions," *Shock Vib.*, vol. 6, no. 2, pp. 81–96, 1999.

[24] M. B. Liu and G. R. Liu, "Meshfree particle simulation of microchannel flows with surface tension," *Comput. Mech.*, vol. 35, no. 5, pp. 332–341, 2005.

[25] R. Garg, C. Narayanan, D. Lakehal, and S. Subramaniam, "Accurate numerical estimation of interphase momentum transfer in Lagrangian-Eulerian simulations of dispersed two-phase flows," *Int. J. Multiphase Flow*, vol. 33, no. 12, pp. 1337–1364, 2007.

[26] A. Tartakovsky and P. Meakin, "Modeling of surface tension and contact angles with smoothed particle hydrodynamics," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 72, no. 2, p. 026301, 2005.

[27] R. E. Apfel, Y. Tian, J. Jankovsky, T. Shi, X. Chen, R. G. Holt, E. Trinh, A. Croonquist, K. C. Thornton, and J. A. Sacco, "Free oscillations and surfactant studies of superdeformed drops in microgravity," *Phys. Rev. Lett.*, vol. 78, no. 10, pp. 1912–1915, 1997.

[28] H. Lopez and L. D. G. Sigalotti, "Oscillation of viscous drops with smoothed particle hydrodynamics," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 73, no. 5, p. 051201, 2006.

[29] Y. Melean and L. G. D. Sigalotti, "Coalescence of colliding van der Waals liquid drops," *Int. J. Heat Mass Transf.*, vol. 48, nos. 19–20, pp. 4041–4061, 2005.

[30] M. Zhang, L. Zhang, and L. Zheng, "Numerical investigation of substrate melting and deformation during thermal spray coating by SPH method," *Plasma Chem. Plasma Process.*, vol. 29, no. 1, pp. 55–68, 2009.

[31] G. Zhou, W. Ge, and J. Li, "A revised surface tension model for macro-scale particle methods," *Powder Technol.*, vol. 183, no. 1, pp. 21–26, 2008.

[32] R. C. Batra and G. M. Zhang, "Analysis of adiabatic shear bands in elasto-thermo-viscoplastic materials by modified smoothed-particle hydrodynamics (MSPH) method," *J. Comput. Phys.*, vol. 201, no. 1, pp. 172–190, 2004.

[33] A. Colagrossi and M. Landrini, "Numerical simulation of interfacial flows by smoothed particle hydrodynamics," *J. Comput. Phys.*, vol. 191, no. 2, pp. 448–475, 2003.

[34] J. Qin, W. M. Pang, B. P. Nguyen, D. Ni, and C.-K. Chui, "Particle-based simulation of blood flow and vessel wall interactions in virtual surgery," in *Proc. Symp. Inf. Commun. Technol.*, 2010, pp. 128–133.

[35] L. D. Libersky and A. G. Petschek, "Smooth particle hydrodynamics with strength of materials," in *Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method*. Berlin, Germany: Springer, 1991, pp. 248–257.

[36] P. W. Randles and L. D. Libersky, "Smoothed particle hydrodynamics: Some recent improvements and applications," *Comput. Methods Appl. Mech. Eng.*, vol. 139, nos. 1–4, pp. 375–408, 1996.

[37] L. D. Libersky, A. G. Petschek, T. C. Carney, J. R. Hipp, and F. A. Allahdadi, "High strain Lagrangian hydrodynamics: A three dimensional SPH code for dynamic material response," *J. Comput. Phys.*, vol. 109, no. 1, pp. 67–75, 1993.

[38] C. E. Zhou, G. R. Liu, and K. Y. Lou, "Three-dimensional penetration simulation using smoothed particle hydrodynamics," *Int. J. Comput. Methods*, vol. 4, no. 4, pp. 671–691, 2007.

[39] R. A. Dalrymple and B. D. Rogers, "Numerical modeling of water waves with the SPH method," *Coastal Eng.*, vol. 53, no. 2, pp. 141–147, 2006.

[40] K. Sims, "Particle animation and rendering using data parallel computation," *Comput. Graph.*, vol. 24, no. 4, pp. 405–413, 1990.

[41] J. Stolk and J. J. Van Wijk, "Surface particle for 3D flow visualization," in *Advances in Scientific Visualization*. Berlin, Germany: Springer, 1992, pp. 119–130.

[42] T.-S. Loke, D. Tan, H.-S. Seah, and M.-H. Er, "Rendering fireworks displays," *IEEE Comput. Graph. Appl.*, vol. 12, no. 3, pp. 33–43, May 1992.

[43] M. Unbescheiden and A. Trembilski, "Cloud simulation in virtual environments," in *Proc. Virtual Reality Annu. Int. Symp.*, 1998, pp. 98–104.

[44] X. Hu and D. Li, "Designing and implementation of waterfall in 3D dynamic terrain," *Computerized Tomogr. Theory Appl.*, vol. 13, no. 2, pp. 38–40, 2004.

[45] X. Provot, "Deformation constraint in a mass-spring model to describe rigid cloth behavior," in *Proc. Graph. Interface*, 1995, pp. 147–154.

[46] H. Xiong, L. H. Jiang, and Y. X. Luo, "A physically restrained plant root growing model based on particle system," *Comput. Appl.*, vol. 22, no. 7, pp. 39–41, 2002.

[47] Q. Yuan, S. Zhou, and X. Guo, "Research on the application of particle system for virtual crop organ development," *Microcomput. Inf.*, vol. 23, no. 7, 2007, Art. no. 262264.

[48] T. Hmonen and J. Konikanen, "The second order particle system," in *Proc. WSCF*, 2003.

[49] M. Desbrun and M. Cani, "Smoothed particles: A new paradigm for animating highly deformable bodies," in *Computer Animation and Simulation*. Vienna, Austria: Springer, 1996, pp. 61–76.

[50] J. J. Monaghan, "Smoothed particle hydrodynamics," *Annu. Rev. Astron. Astrophys.*, vol. 30, no. 1, pp. 543–574, 1992.

[51] M. Becker and M. Teschner, "Weakly compressible SPH for free surface flows," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, 2009, pp. 209–217.

[52] M. Hirschler, M. Huber, W. Säckel, P. Kunz, and U. Nieken, "An application of the Cahn-Hilliard approach to smoothed particle hydrodynamics," *Math. Problems Eng.*, vol. 2014, Feb. 2014, Art. no. 694894.

[53] M. Liu, Z. Zong, and J. Z. Chang, "Developments and applications of smoothed particle hydrodynamics," *Adv. Mech.*, vol. 41, no. 2, pp. 219–236, 2011.

[54] J. J. Monaghan, "Simulating free surface flows with SPH," *J. Comput. Phys.*, vol. 110, no. 2, pp. 399–406, 1994.

[55] J. P. Morris, J. P. Fox, and Y. Zhu, "Modeling low reynolds number incompressible flows using SPH," *J. Comput. Phys.*, vol. 136, no. 1, pp. 214–226, 1997.

[56] S. Shao and E. Y. M. Lo, "Incompressible SPH method for simulating newtonian and non-newtonian flows with a free surface," *Adv. Water Resour.*, vol. 26, no. 7, pp. 787–800, 2003.

[57] E. S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, and P. Stansby, "Comparisons of weakly compressible and truly incompressible algorithms for the SPH mesh free particle method," *J. Comput. Phys.*, vol. 227, no. 18, pp. 8417–8436, 2008.

[58] X. Hu and N. A. Adams, "A constant-density approach for incompressible multi-phase SPH," *J. Comput. Phys.*, vol. 228, no. 6, pp. 2082–2091, 2009.

[59] B. Solenthaler and R. Pajarola, "Predictive-corrective incompressible SPH," *ACM Trans. Graph.*, vol. 28, no. 3, p. 40, 2009.

[60] W. K. Liu, S. Jun, and Y. F. Zhang, "Reproducing kernel particle methods," *Int. J. Numer. Methods Fluids*, vol. 20, nos. 8–9, pp. 1081–1106, 1995.

[61] G. A. Dilts, "Moving-least-squares-particle hydrodynamics consistency and stability," *Int. J. Numer. Methods Eng.*, vol. 44, no. 8, pp. 1115–1155, 1999.

[62] J. K. Chen, J. E. Beraun, and T. C. Carney, "A corrective smoothed particle method for boundary value problems in heat conductions," *Int. J. Numer. Methods Eng.*, vol. 46, no. 2, pp. 231–252, 1999.

[63] M. B. Liu, W. P. Xie, and G. R. Liu, "Modeling incompressible flows using a finite particle method," *Appl. Math. Model.*, vol. 29, no. 12, pp. 1252–1270, 2005.

[64] G. M. Zhang and R. C. Batra, "Symmetric smoothed particle hydrodynamics (SSPH) method and its application to elastic problems," *Comput. Mech.*, vol. 43, no. 3, pp. 321–340, 2009.

[65] J. J. Monaghan, "Simlulating free surface flows with SPH," *J. Comput. Phys.*, vol. 110, no. 2, pp. 399–406, 1994.

[66] M. Gómez-Gesteira and R. A. Dalrymple, "Using a three-dimensional smoothed particle hydrodynamics method for wave impact on a tall structure," *J. Waterway Port Coastal Ocean Eng.*, vol. 130, no. 2, pp. 63–69, 2004.

[67] K. Zeng, Z. C. Sun, Z. M. Zhang, F. Zhou, and X. Z. Zhao, "On methods of processing solid boundary condition in SPH simulation," *Port Waterway Eng.*, vol. 1, no. 8, pp. 1002–4972, 2009.

[68] K. Gong, H. Liu, and B. Wang, "An improved boundary treatment approach for SPH method," *Chin. Quart. Mech.*, vol. 29, no. 4, pp. 507–514, 2007.

[69] J. J. Monaghan and R. A. Gingold, "Shock simulation by the particle method SPH," *J. Comput. Phys.*, vol. 52, no. 2, pp. 374–389, 1983.

[70] J. J. Monaghan, "Particle methods for hydrodynamics," *Comput. Phys. Rep.*, vol. 3, no. 2, pp. 71–124, 1985.

[71] R. W. Hockney and J. W. Eastwood, "Computer simulations using particles," *Adam Hilger*, vol. 3, no. 2, pp. 71–124, 1985.

[72] C. E. Rhoades, "A fast algorithm for calculating particle interactions in smooth particle hydrodynamic simulations," *Comput. Phys. Commun.*, vol. 70, no. 3, pp. 478–482, 1992.

[73] J. C. Simpson, "Numerical techniques for three-dimensional smoothed particle hydrodynamics simulations: Applications to accretion disks," *Astrophys. J.*, vol. 448, p. 822, Aug. 1995.

[74] L. Hernquist and N. Katz, "Tree SPH—A unification of SPH with the hierarchical tree method," *Astrophys. J. Suppl. Ser.*, vol. 70, pp. 419–446, Jun. 1989.

[75] P. Goswami, P. Schlegel, B. Solenthaler, and R. Pajarola, "Interactive SPH simulation and rendering on the GPU," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, 2010, pp. 55–64.

[76] W. G. Wang, H. L. Liu, Y. M. Chen, and Z. C. Zhang, "Coupled SPH-FEM method for analyzing touchdown explosion in sand foundation," *J. PLA Univ. Sci. Technol. (Natural Sci. Ed.)*, vol. 14, no. 3, pp. 271–276, 2013.

[77] L. Ma, W. M. Tao, and Y. M. Guo, "Elastic/plastic impact simulation of water jet using smoothed particle hydrodynamics and finite element method," *J. Zhejiang Univ. Eng. Sci.*, vol. 42, no. 2, p. 259, 2008.

[78] H. T. Zhang, M. Y. Zhao, L. Ren, and B. W. Mi, "Bird impact on aircraft leading edge simulation with SPH and FEM coupling method," *Sci. Technol. Eng.*, vol. 7, p. 28, Jul. 2009.

[79] Z. C. Wang, "Research on droplet impact to discrete particle based on a SPH-DEM coupling method,"

[80] B. Q. Liu, J. Liu, W. Q. Lu, and M. J. Ni, "Simulation of liquid argon flow in micro-channel by an SPH-MD coupling method," *J. Eng. Thermophys.*, vol. 33, no. 6, p. 23, 2012.

[81] Z. Zong, L. Zou, M. B. Liu, and X. J. Wang, "SPH simulation of two-dimensional underwater explosion," *J. Hydrodyn.*, vol. 22, no. 1, pp. 61–67, 2007.

[82] G. Yang, X. Han, and S. Y. Long, "Simulation of underwater explosion near air-water surface by SPH method," *Eng. Mechanicsm*, vol. 4, 2008.

[83] S. C. Shao, C. Ji, D. I. Graham, D. E. Reeve, P. W. James, and A. J. Chadwick, "Simulation of wave overtopping by an incompressible SPH model," *Coastal Eng.*, vol. 53, no. 9, pp. 723–735, 2006.

[84] J. Wei, D. Liang, S. Shao, R. Chen, and K. Yang, "Smoothed particle hydrodynamics simulations of dam-break flows around movable structures," *Int. J. Offshore Polar Eng.*, vol. 26, no. 1, pp. 33–40, 2016.

[85] A. S. Iglesias, L. P. Rojas, and R. Z. Rodriguez, "Simulation of antiroll tanks and sloshing type problems with smoothed particle hydrodynamics," *Ocean Eng.*, vol. 31, nos. 8–9, pp. 1169–1192, 2004.

[86] J. R. Shao, H. Q. Li, G. R. Liu, and M. B. Liu, "An improved SPH method for modeling liquid sloshing dynamics," *Comput. Struct.*, vol. 100, pp. 18–26, 2012.

[87] A. Rafiee, F. Pistani, and K. Thiagarajan, "Study of liquid sloshing: Numerical and experimental approach," *Comput. Mech.*, vol. 47, no. 1, pp. 65–75, Aug. 2011.

[88] J. P. Morris, "Simulating surface tension with smoothed particle hydrodynamics," *Int. J. Numer. Methods Fluids*, vol. 33, no. 3, pp. 333–353, 2000.

[89] M. Miiller, B. Solenthaler, R. Keiser, and M. Gross, "Particle-based fluid-fluid interaction," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, 2005, pp. 237–244.

[90] B. Solenthaler and R. Pajarola, "Density contrast SPH interfaces," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animation*, 2008, pp. 211–218.

[91] N. Grenier, M. Antuono, A. Colagrossi, D. L. Touzé, and B. Alessandrini, "An Hamiltonian interface SPH formulation for multi-fluid and free surface flows," *J. Comput. Phys.*, vol. 228, no. 22, pp. 8380–8393, 2009.

[92] P. W. Cleary, S. H. Pyo, M. Prakash, and B. K. Koo, "Bubbling and frothing liquids," *ACM Trans. Graph.*, vol. 26, no. 3, p. 97, 2007.

[93] J. M. Hong, H. Y. Lee, J. C. Yoon, and C. H. Kim, "Bubbles alive," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–4, 2008.

[94] H. Schechter and R. Bridson, "Ghost SPH for animating water," *Trans. ACM Graph.*, vol. 31, no. 4, pp. 1–8, 2012.

[95] M. Ihmsen, J. Bader, G. Akinci, and M. Teschner, "Animation of air bubbles with SPH," *Comput. Graph. Theory Appl.*, vol. 11, pp. 225–234, Mar. 2011.

[96] G. Zhou and W. Ge, "Progress of smoothed particle hydrodynamics in complex flows," *CIESC J.*, vol. 65, pp. 1145–1161, Apr. 2014.

[97] J. J. Monaghan, A. Kos, and N. Issa, "Fluid motion generated by impact," *J. Waterway, Port, Coastal, Ocean Eng.*, vol. 129, no. 6, pp. 250–259, 2003.

[98] S. Oh, Y. Kim, and B. S. Roh, "Impulse-based rigid body interaction in SPH," *Comput. Animation Virtual Worlds*, vol. 20, nos. 2–3, pp. 215–224, 2009.

[99] M. Becker, H. Tessendorf, and M. Teschner, "Direct forcing for Lagrangian rigid-fluid coupling," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 3, pp. 493–503, May 2009.

[100] X. Shao, Z. Zhou, N. Magnenat-Thalmann, and W. Wu, "Stable and fast fluid–solid coupling for incompressible SPH," *Comput. Graph. Forum*, vol. 34, no. 1, pp. 191–204, 2015.

[101] M. Macklin, M. Ller, N. Chentanez, and T. Y. and Kim, "Unified particle physics for real-time applications," *ACM Trans. Graph.*, vol. 33, no. 4, p. 153, 2014.

[102] X. Y. C. Yan Jiang Li and R. S. Martin Hu, "Multiphase SPH simulation for interactive fluids and solids," *ACM Trans. Graph.*, vol. 35, no. 4, p. 79, 2016.

[103] L. D. Libersky and A. G. Petschek, "Smoothed particle hydrodynamics with strength of materials," in *Advances in the Free-Lagrange Method*. Berlin, Germany: Springer, 1990, pp. 248–257

[104] L. D. Libersky and A. G. Petschek, "High strain Lagrangian hydrodynamics: A three-dimensional SPH code for dynamic material response," *J. Comput. Phys.*, vol. 109, no. 1, pp. 67–75, 1993.

[105] G. R. Johnson, R. A. Stryk, S. R. Beissel, and T. J. Holmquist, "An algorithm to automatically convert distorted finite elements into meshless particle during dynamics deformation," *Int. J. Impact Eng.*, vol. 27, no. 10, pp. 997–1013, 2002.

[106] G. R. Johnson, R. A. Stryk, and S. R. Beissel, "SPH for high velocity impact computations," *Comput. Methods Appl. Mech. Eng.*, vol. 139, nos. 1–4, pp. 347–373, 1996.

[107] G. R. Johnson, R. A. Stryk, and S. R. Beissel, "Interface effects for SPH computations," in *Structures Under Shock and Impact IV*. Boston, MA, USA: Computational Mechanical, 1996, pp. 285–294.

[108] W. Benz and E. Asphaug, "Simulations of brittle solids using smooth particle hydrodynamics," *Comput. Phys. Commun.*, vol. 87, no. 1, pp. 253–265, 1995.

[109] J. K. Chen, F. A. Allahdadi, and T. C. Carney, "High-velocity impact of graphite/epoxy composite laminates," *Compos. Sci. Technol.*, vol. 57, nos. 9–10, pp. 1369–1379, 1997.

[110] M. B. Liu, G. R. Liu, and K. Y. Lam, "Adaptive smoothed particle hydrodynamics for high strain hydrodynamics with material strength," *Shock Waves*, vol. 15, no. 1, pp. 21–29, 2006.

[111] W. T. Reeves, "A Particle systems—A technique for modeling a class of fuzzy objects," *ACM SIGGRAPH Comput. Graph.*, vol. 17, no. 3, pp. 359–375, 1983.

[112] Y. Xu, *Model Reseach of Irregular Scenerie Based on Particle System Abstract*. Jinan, China: Shandong Normal Univ., 2009.

[113] T. David and R. Englandstraat, "Particle systems for artistic expression," in *Proc. Subtle Technol. Conf.*, 2001, pp. 17–20.

[114] Y. Chen, *Particle Systems' Applied and Achieved in The Virtual Environment*. Chengdu, China: Univ. of Electronic Science and Technology of China, 2012.

[115] W. T. Reeves and R. Blau, "Approximate and probabilistic algorithms for shading and rendering structured particle systems," *ACM SIGGRAPH Comput. Graph.*, vol. 19, no. 3, pp. 313–322, 1985.

[116] A. Fournier and W. T. Reeves, "A simple model of ocean waves," *ACM Siggraph Comput. Graph.*, vol. 20, no. 4, pp. 75–84, 1986.

[117] J. Xie, J. Hao, X. Cai, and M. Sun, "Real-time simulating algorithm of rain and snow descending based on particle system," *J. Image Graph.*, vol. 4, no. 9, pp. 734–738, 1999.

[118] P. Beaudoin, S. Paquet, and P. Poulin, "Realistic and controllable fire simulation," in *Proc. Graph. Interface Ontario, Canada Inf. Process. Soc.*, 2001, pp. 159–166.

[119] C. Csuri, R. Hackathorn, R. Parent, W. Carlson, and M. Howard, "Towards an interactive high visual complexity animation system," in *Proc. ACM SIGGRAPH Comput. Graph.*, New York, NY, USA, 1979, pp. 289–299.

[120] K. Sim, "Particle animation and using data parallel computation," *Comput. Graph.*, vol. 24, no. 4, pp. 405–413, 1990.

[121] M. E. Goss, "A real time particle system for display of ship wakes," *IEEE Comput. Graph. Appl.*, vol. 10, no. 3, pp. 30–35, May 1990.

[122] B. Eberhardt, A. Weber, and W. Strasser, "A fast, flexible, particle-system model for cloth draping," *J. Comput. Aided Des. Comput. Graph.*, vol. 16, no. 5, pp. 52–59, Sep. 1996.

[123] W. Song and J. Lai, "Fireworks and trees modeling on the approach of particle system," *J. Comput. Aided Design Comput. Graph.*, vol. 8, no. 4, pp. 254–258, 1996.

[124] X. M. Wang and L. Lin, "Particle system model of tree simulation and its implementation," *J. South China Normal Univ.*, vol. 1, no. 3, pp. 49–52, 2003.

[125] X. Kong, *Simulation of Dynamic Water Grass Based On GPU Abstract*. Suzhou, China: Soochow Universitry, 2012.

[126] H. Wan, X. Jin, and Q. Peng, "Physically based real time fountain simulation," *Chin. J. Comput.*, vol. 21, pp. 774–779, 1998.

[127] Y. Shi, "Graphical simulation algorithm for Chinese ink wash drawing by particle system," *J. Comput.-Aided Des. Comput. Graph.*, vol. 15, no. 6, pp. 667–672, 2003.

[128] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *Proc. Conf. Comput. Graph. Interact. Techn.*, 1987, p. 4.

[129] A. J. S. Hin and F. H. Post, "Visualization of turbulent flow with particles," in *Proc. IEEE Conf. Vis.*, Oct. 1993, pp. 46–53.

[130] J. Wang, *Research on Dynamic Natural Scene Simulation Based on Particle System*. Nanjing, China: Nanjing Univ. of Aeronautics and Astronautics, 1999.

[131] A. Pang, "Spray rendering," *IEEE Comput. Graph. Appl.*, vol. 14, no. 5, pp. 57–63, Sep. 1994.

[132] T. Ruofeng, C. Lingjun, and W. Guozhao, "A method for quick smog simulation," *J. Softw.*, vol. 10, no. 6, pp. 647–651, 1999.

[133] L. Chen, *Study on Particle System Theory and Its Application in Real-time Visual Scene Simulation of Flight Simulator*. Changchun, China: Jilin Univ., 2004.

[134] Y. Che, *Particle System's Applied and Achieved in The Virtual Environment*. Chengdu, China: Univ. of Electronic Science and Technology of China, 2012.

[135] D. E. Breen, D. H. House, and P. H. Getto, "A physically-based particle model of woven cloth," *Vis. Comput.*, vol. 8, nos. 5–6, pp. 264–277, 1992.

[136] B. Zhan, J. Quan, and P. Wang, "Principles based on particle system modeling," *Sci. Technol. Vis.*, vol. 24, no. 19, p. 105, 2016.

[137] H. Liu, *Simulation of 0-9 Digital Shaping Fireworks Based on the Particle System*. Hefei, China: Anhui Univ., 2013.

[138] F. Losasso, J. Talton, N. Kwatra, and R. Fedkiw, "Two-way coupled SPH and particle level set fluid simulation," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 4, pp. 797–804, Mar. 2008.

[139] M. Macklin and M. Müller, "Position based fluids," *ACM Trans. Graph.*, vol. 32, no. 4, p. 104, 2013.

[140] H. Zhang, *Fluid Motion and Solid-Liquid Coupling Simulation Based on SPH Method*. Hefei, China: China Univ. of Science and Technology, 2015.

[141] M. Liu, Z. Zong, and J. Chang, "Development and application of smooth particle dynamics," *Progr. Mechanics*, vol. 41, no. 2, pp. 217–234, 2011.

[142] M. P. Allen, D. J. Tildesley, *Computer Simulation of Liquids*. Oxford, U.K.: Clarendon, 1987.

[143] M. Zhang, *Numerical Simulation of Impact Load Based on SPH Method*. 2016.

[144] *Large-Scale Numerical Simulation of SPH Fluid Motion on Liu Yang*, CUDA Platform, Yanshan Univ., Qinhuangdao, China, 2005.

[145] H. Zhou, *Parallelization and Application of Smooth Particle Hydrodynamics*. Changsha, China: National Univ. of Defense Science and Technology, 2011.

[146] K. Shi, *Research on Fluid Simulation Method Based on CUDA Technology*. Xi'an, China: Chang'an Univ., 2011.

[147] G. Oger, M. Doring, B. Alessandrini, and P. Ferrant, "Two-dimensional SPH simulations of wedge water entries," *J. Comput. Phys.*, vol. 213, no. 2, pp. 803–822, 2006.

[148] M. Ihmsen, J. Cornelis, B. Solenthaler, C. Horvath, and M. Teschner, "Implicit incompressible SPH," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 3, pp. 426–435, Mar. 2014.

RUNPING XI received the Ph.D. degree in computer science and technology from Northwestern Polytechnical University, in 2011. He is currently an Associate Professor with the School of Computer, Northwestern Polytechnical University. His current research interests include computer graphics, image processing, computer vision, and multimedia information processing.

ZHANGCAI LUO received the M.S. degree in computer science and technology from Northwestern Polytechnical University, in 2019. His research interests include computer graphics, image processing, and computer vision.

DAVID DAGAN FENG received the M.E. degree in EECS from SJTU, in 1982, and the M.Sc. degree in biocybernetics and the Ph.D. degree in computer science from UCLA, in 1985 and 1988, respectively, where he received the Crump Prize for Excellence in Medical Engineering. In conjunction with his team members and students, he has been responsible for more than 50 key research projects, published over 900 scholarly research articles, pioneered several new research directions, and made a number of landmark contributions in his field. His research interests include biomedical systems modeling, functional imaging, biomedical information technology, and multimedia computing.

YANNING ZHANG received the Ph.D. degree in computer science and technology from Northwestern Polytechnical University. She is currently a Professor with the School of Computer, Northwestern Polytechnical University. Her research interests include image processing, computer vision, pattern recognition, artificial intelligence, and machine learning.

XIAOPENG ZHANG received the Ph.D. degree in computer application from the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. He is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His main research interests include computer graphics, computer vision, and artificial intelligence.

TIANYI HAN received the B.S. degree in computer science and technology from Northwestern Polytechnical University, in 2018, where he is currently pursuing the master's degree in computer science and technology. His research interests include image processing, computer vision, and machine learning

• • •