Received: 12 July 2023 | Accepted: 22 July 2023
DOI: 10.1002/cav.2202

SPECIAL ISSUE PAPER

WILEY

Efficient and high precision target-driven fluid simulation based on spatial geometry features

Xiangyang Zhou[1,2,3] | **Sinuo Liu**[1,2,3,4] | **Haokai Zeng**[1,2,3] | **Xiaokun Wang**[1,2,3,5] | **Xiaojuan Ban**[1,2,3,6]

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

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

[3]Key Laboratory of Intelligent Bionic Unmanned Systems, University of Science and Technology Beijing, Beijing, China

[4]School of Computer Science, Peking University, Beijing, China

[5]National Centrefor Computer Animation, Bournemouth University, Poole, UK

[6]Institute of Materials Intelligent Technology, Liaoning Academy of Materials, Shenyang, China

Correspondence
Xiaojuan Ban, Beijing Advanced Innovation Center for Materials Genome Engineering, University of Science and Technology Beijing, 30 Xueyuan Road, Haidian District, Beijing, China.
Email: banxj@ustb.edu.cn

Sinuo Liu, School of Computer Science, Peking University, 5 Yiheyuan Road, Haidian District, Beijing, China.
Email: sinuoliu@pku.edu.cn

Funding information
Guangdong Basic and Applied Basic Research Foundation, Grant/Award Numbers: 2022A1515110350, 2021A1515012285, 2023A1515030177; H2020 Marie Skłodowska-Curie Individual Fellowship, Grant/Award Number: 895941; National Key R&D Program of China, Grant/Award Number: 2022ZD0118001; National Natural Science Foundation of China, Grant/Award Number: U22A2022; Postdoctoral Innovation Talents Support Program, Grant/Award Number: BX20220007

Summary
We proposed a novel target-driven fluid simulation method based on the weighted control model derived from the spatial geometric features of the target shape. First, the spatial geometric characteristics of the target model are taken into account to set the color field weights of control particles. This enabled the full expression of geometric characteristics of the target model, and improve the shape accuracy of controlled fluid. Then, the fluid is controlled to form the target shape under driving constraints, wherein we proposed a new adaptive constraint mechanism that enables efficient target shape generation. Finally, a new density constraint between the control particles and the controlled fluid particles is proposed to ensure the incompressibility of fluid during control. Compared to the state-of-the-art target-driven fluid control methods, our method achieves higher precision fluid control with higher efficiency.

KEYWORDS
smoothed particle hydrodynamics, spatial geometry weighting, target-driven fluid simulation

1 | INTRODUCTION

Target-driven fluid control is a hot topic in Computer Graphics, which creates unique simulation results by driving the motion of the fluid to form a specific shape. Since only 3D models are required to generate the desired fluid effect, these methods are widely used in the field of games and films, such as the movie "Terminator", "Spider-Man-Homecoming", and "Frozen". Besides, the fluid control method can also be used for auxiliary medical research. The symptoms of various vascular diseases could be obtained by simulating blood flow using a human blood vessel model. However, to generate the desired visual effect, the time-consuming repeatedly adjusting of a sea of parameters is required. This also makes it hard to accurately fill fluid into a model with complex spatial geometry features. Therefore, it is of great significance to propose a fluid-driven method that automatically matches the target shape according to the user's expectation, and forms the desired fluid result efficiently and accurately.

To control the motion of the fluid, Zhang et al.[1] applied a constraint driving force to the fluid particles and influence their velocity. However, it could only generate an approximate surface of a simple toy object, but cannot handle complex models with a large ratio of surface-to-volume ratio. Lu et al. [2] proposed to drive the movement of fluid more intuitively by rigging-skinning mechanism. This method is effective, but the flow characteristics tend to be lost due to the dense sampling of control particles and the excessive control constraints. Schoentgen et al. [3] proposed to make it easier for artists to achieve their desired effect by using precomputed templates. However, such an expected effect can only be limited to the existing templates, and the preparation is very time-consuming.

In order to solve the above problems, we propose a new controllable fluid simulation method driven by a weighted model of spatial geometric features. Firstly, our proposed weight-based actuation control model tackles the issue of sparse feature points in model details by applying stronger constraints to areas with fewer feature points and vice versa. This approach enables the fluid to conform better to the model's shape, resulting in improved fluid surface generation with the same computational resources. Secondly, our innovative approach introduces an adaptive driving constraint that can regulate the constraint range independently, resulting in a significant reduction in the required number of frames for target model generation. The proposed method generates the controlled result efficiently and effectively by controlling the driving force between the particles and the fluid through adaptive adjustment. Finally, the conventional approaches are prone to fluid compression resulting from control constraints, whereas our novel density-driven constraint model ensures fluid incompressibility and improve the realism of fluid flow, thus offering an improved solution. To summarize, our main contributions include:

1. A novel target-driven fluid simulation method based on spatial geometry features, which achieves high-precision fluid control efficiently, and could maintain subtle details of a complex target model.
2. A novel adaptive constraint model that controls particles based on the distance between fluid particles and control particles, which greatly improves the shape forming efficiency.
3. A new density-driven constraint model to ensure the incompressibility of fluid during the control process, and improve the realism of fluid flow.

2 | RELATED WORK

Target-driven fluid control is an important technology of fluid animation. There are many different approaches to the study of driving. The following aspects of fluid drive are introduced, including drive the fluid in a static shape (2.1), drive the fluid in a motion process (2.2), and the fluid controllable simulation system (2.3).

2.1 | Drive the fluid in a static shape

2.1.1 | Keyframe driven simulation

Keyframe driven simulation is where the final shape of a fluid particle is guided by a specific keyframe. Treuille et al.[4] matched a set of sketchy shapes for a given keyframe by applying forces found by using a gradient-based optimizer. Hong and Kim[5] extracted a potential field from static user-defined keyframes during preprocessing and then applied the driving force to the smoke simulation, proportional to the gradient of this field. Bergou et al.[6] took a rough keyframe animation and augmented it with details from a physical simulation, allowing the artist to choose where to add details. Chu et al.[7] implemented the ability to derive plausible velocity fields from a single frame.

To improve the computing efficiency, Pan et al.[8] edit keyframe liquid simulations locally by solving an optimization problem on a subset of particles. In addition, in order to improve the efficiency by an order of magnitude, Pan and Manocha[9] decomposed the keyframe animation optimization into two subproblems and solved them with the (ADMM). Later, Pan et al.[10] deformed mesh for simulation where smoke can be driven directly by editing the mesh.

2.1.2 | Target model actuation

Target model actuation is the actuation of a fluid to form a desired shape by a specific actuation model or actuation module. Fattal and Lischinski[11] used two forces, a driving force that directs smoke to the density field of the target model and a smoke aggregation force that prevents smoke diffusion. Raveendran et al.[12] took as input a dense sequence of target-driven meshes and used the velocities of these meshes as boundary conditions for pressure resolution. Zhang et al.[1] took a specific model as the driving target, and formed the fluid surface by driving forces including density, spring and velocity. Cornelis et al.[13] realized that the fluid could be converted between the target object and the fluid, enabling a more efficient and simplified solver. Lu et al.[2] also proposed to drive the movement of fluid more intuitively by means of rigging movement and fluid skin. Schoentgen et al.[3] proposed that different driving templates should be prefabricated to make it easier for artists to achieve the desired effect.

2.2 | Drive the fluid in a motion process

2.2.1 | Enhance and drive detail

This is achieved by inputting a low-resolution effect, by adding additional detail and extra actuation to enhance the actuation effect of the fluid. Nielsen et al.[14] first simulated high-resolution smoke by tracking low-resolution inputs, adding additional details while preserving the overall appearance of low-resolution. Since then, they[15] have also proposed an improved mathematical model of Eulerian-based simulation, improved the use of steady-state (constant) guidance that often leads to a lack of sufficient high detail. Later, they also applied these ideas to the driving simulation of fluids.[16]

Through conversion in the frequency domain, Forootaninia et al.[17] showed this effect by combining the high-frequency component of the simulated fluid velocity with the low-frequency component of the input guidance field in order to simulate the smoke faster and maintain the details of the smoke itself. Sato et al.[18] introduced a stream function based on Forootaninia's attempt to ensure efficiency and appearance, which could better drive the shape of smoke.

2.2.2 | Dynamic physical field driven simulation

It works by attaching an additional physical field to the fluid to drive its flow path. McNamara et al.[19] used gradient-based nonlinear optimization to drive physics-based fluid simulations, which was guided early by adding additional constraints to guide the driving fluid. Shi and Yu applied feedback forces in smoke[20] and liquid to compensate for the differences in shape and velocity, matching the controlled fluid to the given target. Sato et al.[21] proposed to calculate the (time-varying) vector potential from a sequence of input velocity field, then deform the vector potential, calculate the deformed velocity field by using the spin operator on the vector potential. Thuerey[22] proposed a mesh-based input signed distance function to compute dense spatiotemporal deformations, combining data-driven fluid simulation and interpolation techniques that match and interpolate phenomena with fundamentally different physics. Inglis et al.[23] used a first-order Primal-Dual method to minimize the difference between the guided velocity field and the target velocity field.

In addition to using material flow fields, Rasmussen et al.[24] guided fluid particles through soft or hard constraints that define several physical properties of fluids by artist-generated control particles. Stomakhin and Selle[25] also used input-driven shapes to enforce boundary conditions and deal with particles reseeding around shapes that describe open boundaries.

2.3 | The fluid controllable simulation system

In the previous fluid actuation application, Foster and Metaxas[26] developed a method to actuate fluid animation by varying physical properties such as pressure and surface tension. The actuator serves as an interface between the animator and a generic tool for computing 3D fluid flow. Mihalef et al.[27] described an intuitive method for generating wave simulations. There are now more freely editable applications, such as Bojsen-Hansen and Wojtan[28] inserted template liquid animation into an existing fluid simulation using special boundary conditions to smooth the transition. The sculpting idea proposed by Manteaux et al.[29] enabled the artist to edit the mesh extracted from the liquid simulation rather than modifying the simulation itself. The sculpture-inspired tool proposed by Stuyck and Dutre[30] also stands out from previous work in that their method is faster, more intuitive, and boasts easier operations. Sato et al.[31] produced the required fluid animation by combining existing streaming data through synthesis. The artist just needs to put the desired result in the corresponding position and interpolate the velocity on the boundary among the flows to ensure the realistic simulation of the boundary.

3 | TARGET-DRIVEN FLUID SIMULATION BASED ON SPATIAL GEOMETRIC FEATURES

In the field of fluid control simulation, achieving higher accuracy and efficiency for driving the fluid remains a formidable challenge. Simply tightening the actuation constraint can cause the fluid to accumulate in areas with higher constraints, making it difficult to populate the entire model. Otherwise, the geometric details of the model will be lost. In addition, it is also vital to improve the efficiency of the driving method. Excessive computing costs are not conducive to the development of art. This paper presents a novel fluid control method that incorporates spatial geometric features of the target model. The proposed method utilizes a depth weighting model to adjust the color field of each control particle during target model sampling to ensure accurate fluid driving constraints in different regions of the model. This approach effectively handles fluid guidance and filling challenges in narrow regions of various models and enhances the fluid driving guidance rate through weighted scaling of control particles. In general, the proposed method demonstrates improved stability and accuracy in the process of fluid guidance, leading to better results in generating fluid surfaces for target models.

3.1 | Position based fluid simulation

Position based fluid (PBF[32]) method is based on position based dynamics (PBD) method and smoothed particle hydrodynamics (SPH) method. It directly updates the object according to the constraints between the spatial positions of the object. PBD changes the position of particles through constraints, and when the particles do not conform to the constraints, they change the position of particles to meet the constraints. PBF introduces the idea of SPH on the basis of PBD, so that the method of PBF can be applied to fluids. Because the N-S equation contains a second-order derivative term, the traditional solution requires a lot of computing resources. The combination of the two can greatly save computing resources and improve the efficiency of computing.

$\boldsymbol{x}$ is a shifted point with coordinates, $C(\boldsymbol{x})$ is the constraint that $C(\boldsymbol{x}) = 0$ in the normal state, which should also be satisfied when the coordinate is shifted by a small amount of $\Delta\boldsymbol{x}$: $C(\boldsymbol{x} + \Delta\boldsymbol{x}) = 0$. Taylor expansion of this equation:

$$C(x + \Delta x) = C(x) + \nabla x C(x) \cdot \Delta x + O\left(|\Delta x|^2\right) = 0, \tag{1}$$

where $C(\boldsymbol{x})$, $\nabla \boldsymbol{x} C(\boldsymbol{x})$can be computed with the definition of the constraint. Using $\boldsymbol{x}$ as the coordinate, the number of particles in three dimensions is n, so computing C for all particles requires $C_i(i = 1, \dots, n)$, $\Delta\boldsymbol{x}$ is changing the particle positions to satisfy the constraints.

Assuming that the point $\boldsymbol{x}$ is moved according to the equation of motion, and $\Delta\boldsymbol{x}$ is moved according to some other constraint C that is different from the equation of motion, then a direction of movement that does not affect the original motion is required. That is, the parallel and rotational movement directions are preserved. Thus Δx has the same direction as $\nabla x C$, and the Lagrange multiplier makes:

$$\Delta x = \lambda \nabla x C(x). \tag{2}$$

Substituting into the formula, we can get:

$$\lambda_i = -\frac{C_i(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)}{\sum_{k \in N} |\nabla \boldsymbol{x}_k C_i(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)|^2}. \tag{3}$$

Combined with the SPH method, by calculating the Lagrange multiplier, the change of position $\Delta \boldsymbol{x}_i$ can be obtained. In order to keep symmetry between the influence of neighbor particles and the influence of neighbor particles, the change of position can be calculated by the following formula:

$$\Delta \boldsymbol{x}_i = \frac{1}{\rho_0} \sum_{j \in N} (\lambda_i + \lambda_j) m_j \nabla W(\boldsymbol{x}_i - \boldsymbol{x}_j, h). \tag{4}$$

Zhang's method[1] is to control particle motion based on PBF through three kinds of constraints, namely density constraint, spring constraint and velocity constraint. By modifying physical quantities such as velocity and viscosity to achieve the purpose of driving fluid, the formation of the fluid surface is driven.

3.2 | Density-based incompressible fluid control

3.2.1 | Generating control particle

The simplest way to create control particles is to use a given precomputed function as described by Foster.[33] In order to simulate the fluid for a given target shape, the initial triangular mesh is sampled periodically, and then the control particles replace each mesh in the animation sequence with the average mesh coordinates. In the proposed method, the control particles are obtained by sampling the mesh model through the existing voxelization method.[34] This method can preserve as many details of the sampled target model as possible, while keeping the computational efficiency within an acceptable range.

3.2.2 | Density-based driving constraint

In the PBF-based fluid actuation method, fluid particles are affected by the forces of control particles, such as traction or repulsion. The set of all these forces is known as the driving constraint. The density constraint is that the control particles treat themselves and their neighboring fluid particles as a system.

However, the density constraint does not consider the overall distribution characteristics of the model, and all control particles and their nearby fluid particles are integrated into the calculation. When the model sampling is too small, or the radius of the fluid particles is too small (e.g., the ratio of sampling distance to the radius of the fluid particles is 1), insufficient control constraints will occur in the special structure part. Simply enlarging the scaling factor will cause the excessive density constraint of the rest of the region and produce unexpected effects.

During the driving process, the control particles are introduced and density constraints are applied to the adjacent fluid. The basic driving effect on the fluid can be realized. This paper introduces a density-driven constraint model in Figure 1, the constraint on the i control particles can be expressed as:

$$C_i(X_i, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \frac{\widetilde{\rho}_i}{\widetilde{\rho}_0} - 1 = 0, \tag{5}$$

$$\widetilde{\rho}_i = \sum_j m_j W(X_i - \boldsymbol{x}_j, H), \tag{6}$$

where $\widetilde{\rho}_0$ is the static density of the control particles, defined by the standard PBF density estimation, and $\widetilde{\rho}_i$ is the fluid density within the support radius of the i control particles. m_j represents the fluid particles mass and H is the control particles support radius.

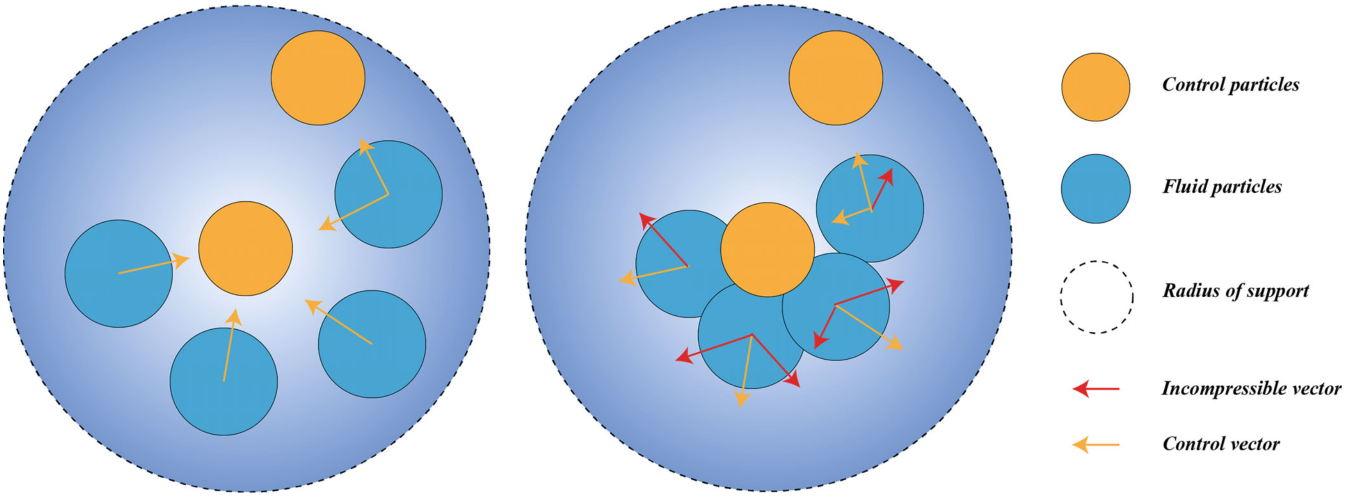

FIGURE 1 This is the driving constraint schematic. Within the influence range (light blue) of the control particle (orange), when the spatial distribution of fluid particles (dark blue) is sparse, the control force will act as attractive force. Otherwise, it will be represented as repulsive force. The orange arrow is the position displacement vector of fluid particles generated by the influence of the control particle, and the red arrow is the displacement vector generated by the excessive density between the fluid particles to ensure incompressibility.

According to the derivation, the sum of the displacement action of the neighboring control particles on each fluid particles can be expressed as:

$$\Delta \boldsymbol{x}_i^{\text{drive}} = -\alpha \frac{1}{\widetilde{\rho}_0} \left(\sum_j \widetilde{\lambda}_j \nabla W\left(X_j - x_i, H\right) + \sum_j \left(\lambda_i + \lambda_j\right) \nabla W\left(\boldsymbol{x}_i - \boldsymbol{x}_j, h\right) \right), \tag{7}$$

where X_j is the position of the control particles j, $\boldsymbol{x}_i$ is the position of the fluid particles, $\boldsymbol{x}_j$ is the position of the neighboring fluid particles, H represents the support radius of the control particles, h stands for the fluid particles support radius, and $\boldsymbol{x}_j$ is the position of the control particles. Generally, it is several times the support radius of the fluid particles. $\widetilde{\lambda}$ represents the constraint value between the control particles and the fluid particles, λ is the constraint value between fluids, and each control particle constraint has the same $\widetilde{\lambda}_i$ for all the fluid particles under the constraint. $\widetilde{\rho}_0$ denotes the static density of the control particles, and α is a scaling factor between 0 and 1.

Through our method, the compression between fluids caused by the actuation constraints is resisted to ensure that the fluid is more efficient in the process of filling the surface. This method can match the shape of the target model more effectively and quickly, and maintain rich fluid characteristics.

3.3 | Adaptive constraint

The adaptive constraint is used to set a consideration point, which has adaptive characteristics and is placed between the fluid and the control particle. The driving surface feature point needs to be selected according to the shortest distance. The adaptive constraint can set different constraint radii for control particle. This allows different degrees of driving force at different distances, enabling the fluid to quickly fill to the top of the driving force and greatly improving the efficiency of the overall fluid generation surface. The farther the driving force increases, the more attractive the surface of the driving force will be to restrain the fluid particles to a closer distance. Otherwise, the constraint has little effect on the fluid particles and therefore no side effects such as artificial viscosity are introduced. The adaptation constraint is formulated as follows:

$$f(d) = \frac{1}{1 + e^{k(H-d)}}, \tag{8}$$

where d represents the distance between the fluid particles and the control particle, H represents the support radius of the driving surface feature point, and k represents the mutation amplitude of the spring constraint at the boundary. In

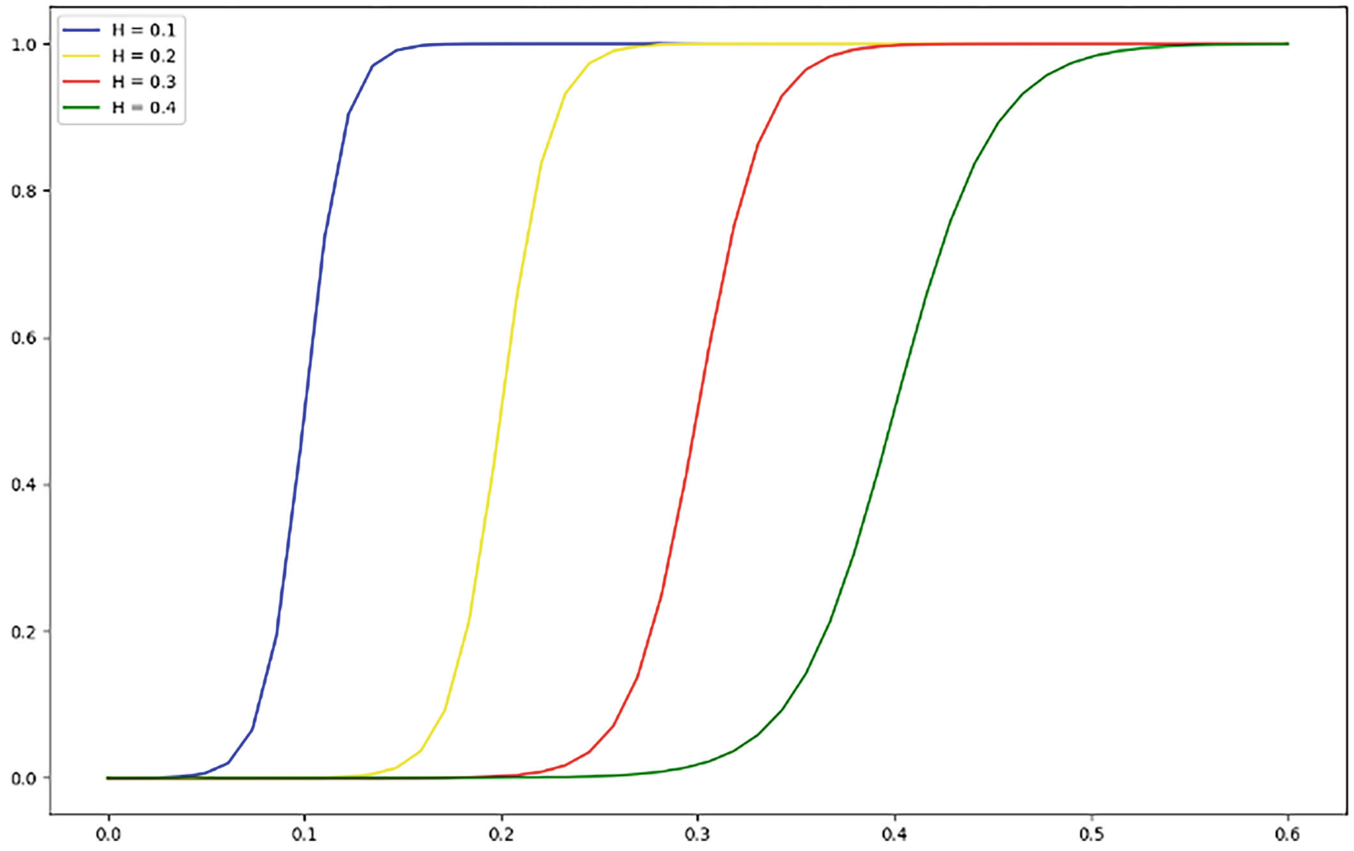

FIGURE 2 The adaptive constraint function constraint applies a limited displacement according to the distance between the control particle and the fluid particles, and adaptively adjusts the range of the constraint according to the different control radii.

the proposed method, the critical distance of the function $f(d)$ has a close relationship with the driving surface. Figure 2 shows the adaptive range and size of the adaptive constraints:

$$\Delta \boldsymbol{x}_i^{\text{adaptive}} = \beta f(d) \frac{\boldsymbol{r}_i - \boldsymbol{x}_i}{d}, \tag{9}$$

where r_i represents the reference control particle position of the fluid particles, and β is a scaling factor between 0 and 1.

3.4 | Depth-weighted model based on spatial geometric features

In this paper, the level set method is used to represent the driving information in three-dimensional space. The length distance of the boundary surface is defined as the distance between the point and the closest point on the surface. Muller[35] SPH represented surfaces in the following form:

$$\phi(\boldsymbol{x}) = \sum_j m_j \frac{1}{\rho_j} W\left(\boldsymbol{x} - \boldsymbol{x}_j, h_j\right) - \chi, \tag{10}$$

where χ is a constant > 0, and the smooth kernel W chooses the appropriate function as appropriate.

Our method aims to enhance the expression ability of surface details of controlled fluids by accurately guiding them in a gravitational field environment, and tracking complex topological target models with high surface-area to volume ratio. This is achieved by using the level set surface tracking method proposed by Zhu et al.[36] as the basis for the method. A fluid depth propagation algorithm based on smooth particle hydrodynamics is implemented in the method. This algorithm enables the ability to exert influence on control particles based on depth weighting, enhancing the internal stability of the filled area and improving the driving ability of the outer surface of the target model.

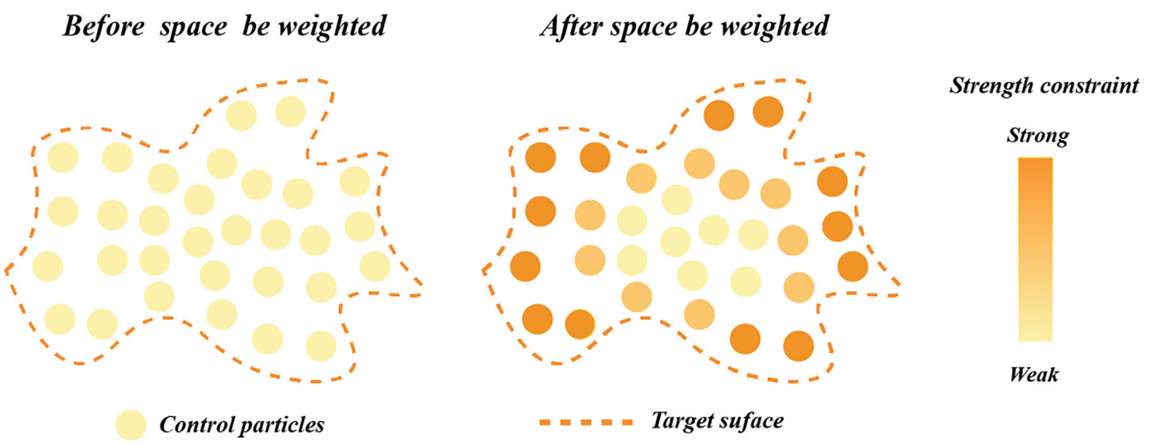

FIGURE 3 The control weight based on the spatial geometry features of the target shape. The control particles near the surface require greater driving force to ensure the fluid fills the target shape efficiently and accurately.

To drive the particles ' depth track in the model, the first particle is marked as i for all models model particles neighbor search and the neighbors particles are marked as j, for a two-way relationship between neighbors. Considering the spatial and temporal distribution characteristics of control particles, the proposed method uses the octree method to improve the efficiency of neighbor search. After the neighbor particles search, the SPH method is used to calculate the neighbor particles contribution weight. Based on the position of each control particle in space, the weighted average position of the control particles is calculated according to the neighbor particles contribution weight $\overline{\boldsymbol{x}}_i$:

$$\omega_{ij} = \frac{W_{ij}}{\sum_j W_{ij}}, \overline{\boldsymbol{x}}_i = \sum_j \omega_{ij}\boldsymbol{x}_j. \tag{11}$$

The sum of weights generated by all neighbor control particles (including themselves) for control particles i is $1\left(\sum_i \omega_{ij} = 1\right)$, and the closer the neighbor particles is, the greater the effect is. This step eliminates the numerical instability caused by the heterogeneous arrangement of particles. Figure 3 is a schematic diagram of different constraint strengths:

$$\widetilde{\phi}_i = \left|x_i - \overline{\boldsymbol{x}}_i\right| - r, \tag{12}$$

$$\phi_i = \begin{cases} -0.85r & Neighbor \leq 10 \\ \widetilde{\phi}_i & else \end{cases}, \tag{13}$$

where $\widetilde{\phi}_i$ is called the temporary color field of the particles, and r is the uniform distribution radius of the control particles. The position migration characteristics of the control particles can be determined according to the numerical value of the color field, and the control particles of the target can be determined. Based on the temporary color field, the color field ϕ_i of each control particle at each time is further determined. In the experiment, the control particles of $\phi_i \geq 0.85r$ is selected as the outermost particles, and then the driving weight of the control particle decreases according to the depth.

In order to obtain the depth information of the control particles, the color field propagation algorithm is used in the method design in this paper. The control particles depth is calculated recursively according to the surface control particles. In each recursion, firstly, all non-surface control particles are traversed, and whether any neighbor control particles are surface control particles or shallow control particles is examined. If so, the following calculation is made afterwards:

$$\phi_i := \max\left(\phi_{j'} - \left|\boldsymbol{x}_i - \boldsymbol{x}_{j'}\right|, \phi^{\min}\right), \tag{14}$$

where j' said neighbors control particles or shallow control particles surface, $\phi^{\min}$ for color floor a threshold, the symbol ":=" to update assignment, to update the color in the previous step field selected by numerical calculation. This formula

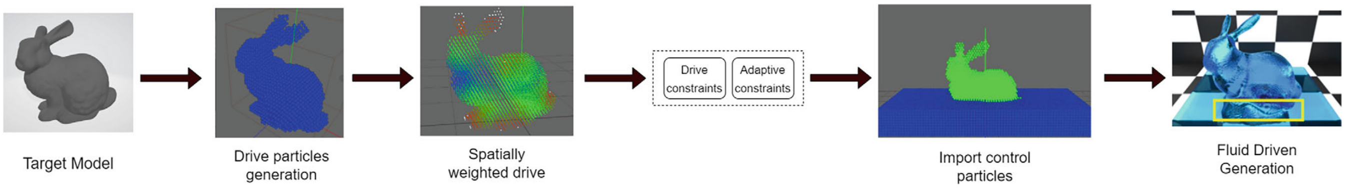

FIGURE 4 Our method applies different driving forces to fluid particles adaptively to drive the fluid forming the target shape.

means that if there is a neighbor particles j' that meets the criteria, then all color field shifts about j' are calculated and the largest among them is selected. Maximum value, and then to $\phi^{\min}$, compared with the maximum update particles color fields. In the experiment, $\phi^{min} = -18r$ was selected, and the non-surface control particles larger than the threshold were selected as shallow control particles. This step is calculated iteratively at each time step in the modeling process until the color field is stable and the color field propagation is completed.

Finally, the density constraint process is modified. The final constraint size is scaled by adding color field weighting. Figure 4 introduces the flow of the entire project. Before the adjustment, the density constraint strength at any position in the space is the same, especially the part close to the surface in the figure. It results in the situation that the fluid particles cannot be fully guided. After the color field weighting adjustment, the closer the depth information is to the surface, the higher the density constraint strength will be. On the contrary, the density constraint strength of the control particles in the interior will be weakened. It avoids the problem that the fluid particles can not move to the surface where there are few control particles, and enhances the stability of the fluid inside the model.

In the solution formula, Δx_j needs to adopt a position-based dynamic method. According to PBD, Δx is estimated to be:

$$\Delta \boldsymbol{x}_j = \widetilde{\lambda}_i \nabla_{x_j} C_i(P_i, x_1, \ldots, \boldsymbol{x}_n), \tag{15}$$

where $\widetilde{\lambda}_i$is the scalar value in the calculation of each control particle i, which is modified by color field weighting in this method:

$$\widetilde{\lambda}_i' = \phi_i \widetilde{\lambda}_i, \tag{16}$$

Finally, the updated scalar value is substituted into the formula for displacement calculation. The algorithm of our method is shown in Algorithm 1. The blue part is our innovative approach.

4 | RESULTS

To validate the performance of our method, we set several scenarios to drive the fluid in a static shape. We perform the simulation of the same scene multiple times using different constraint parameters, constraint modes, and number of fluid particles. Our method is primarily compared to two existing techniques for fluid control: Position-based fluid control (PBFC)[1] and Particle-based Liquid Control using Animation Templates (LCAT).[3] The CPU is INTEL i7-9700 with 16GB of RAM. Our code is implemented on the open-source library SPlishSPlasH. Off-line rendering is performed using the open source software Blender, in order to verify the effectiveness of the method, the existing volume sampling method is used for all target models without fine adjustment. Figure 5 shows the placement of the control particles in each of our experimental scenes.

To better demonstrate the impact of the two reconstruction methods, we compare the differences in particle positions between their respective results. The positional information allows for a visual comparison of the appearance and shape of generated particles, which is conducted in three dimensions. Additionally, we are comparing discrepancies in the number of fluid particles that require control. The modification for the density constraint is illustrated in Figure 6, which employs fewer fluid particles and yields superior results compared to other experiments where additional fluid particles were utilized to fill intricate details. Detailed data are given in Tables 1 and 2.

Algorithm 1. Overall simulation loop

for fluid particles i **do**
  apply forces $v_i \Leftarrow v_i + \Delta t f_{external}(x_i)$
  predict position $x_i^* \Leftarrow x_i + \Delta t v_i$
end for
for fluid particles i **do**
  find neighboring fluid particles
end for
for control particles i **do**
  calculate $\widetilde{\lambda}_i$
  calculate ϕ_i
  calculate $\widetilde{\lambda}_i' = \phi_i \widetilde{\lambda}_i$
end for
for fluid particles i **do**
  calculate $\Delta \boldsymbol{x}_i^{\text{drive}}$ and$\Delta \boldsymbol{x}_i^{\text{adaptive}}$
  update position $x_i^* \Leftarrow x_i^* + \Delta \boldsymbol{x}_i^{\text{drive}} + \Delta \boldsymbol{x}_i^{\text{adaptive}}$
end for
for iter< solverIterations **do**
  for fluid particles i **do**
    calculate λ_i and λ_j
  end for
  for fluid particles i **do**
    calculate $\Delta \boldsymbol{x}_i$
    collision detection and response
    update position $x_i^* \Leftarrow x_i^* + \Delta \boldsymbol{x}_i$
  end for
end for
for fluid particles i **do**
  update velocity $v_i \Leftarrow \frac{1}{\Delta t}(x_i^* - x_i)$
  update position $x_i \Leftarrow x_i^*$
end for

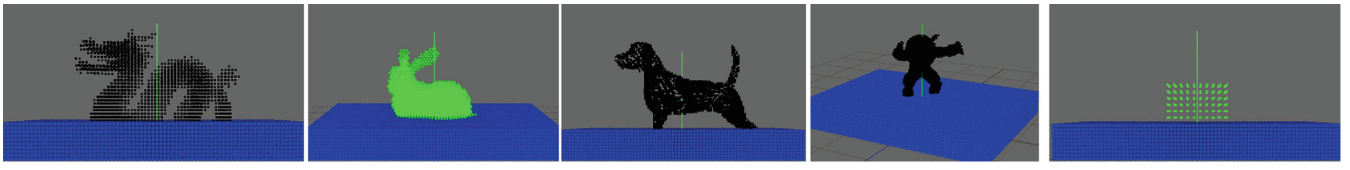

FIGURE 5 Placement of control particles in the scene.

The depth propagation algorithm is used to realize the depth tracking of the target model. The method binds a color field weighting value to each control particle to represent the information characteristics of the current space. The color field weighting is then incorporated into the density constraint calculation of the fluid driving basic constraint. Accurate drive release and drive of the driving force are realized. This paper addresses the issue of insufficient control force in driving the target 3D model. To achieve this, the geometric characteristics of the target 3D model are fully considered, and information about the key geometric characteristics of the model is incorporated into the fluid control process.

4.1 Shape accuracy

Figure 7 shows the comparison of the effects of different methods of generating the dragon model. In the case of the same time frame, the proposed method can drive the fluid motion faster. In the part circled in yellow in frame 102, the

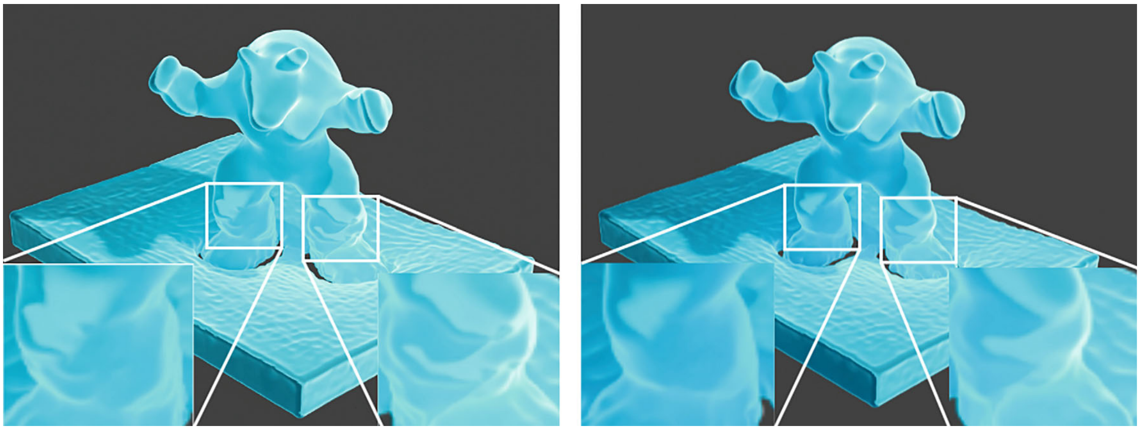

FIGURE 6 Comparison of our method with (right) and without (left) the newly proposed density-driven constraint model with an armadillo-shaped target.

TABLE 1 The data of different experimental scenarios in PBFC method.

Target model	Figure 7	Figure 8	Figure 9	Figure 6
Fluid particles	20,886	27,848	46,610	46,610
Control particles	6978	7706	4366	6428
Constraint	0.20	0.8	0.25	0.02
Controlled fluid particles	10,465	9763	3130	17,456
Y-axis range of control particles	0–1.875	0–1.811	0–2.779	0–2.812
X-axis range of control particles	−1.176–1.173	−0.778–0.801	−1.839–1.641	−0.788–0.857
Z-axis range of control particles	−0.535–0.515	−0.576–0.599	−0.557–0.562	−1.011–0.766
Y-axis range of controlled fluid particles	0–1.69391	0–1.63916	0–0.92523	0–2.79307
X-axis range of controlled fluid particles	−1.11028–1.19130	−0.77301–0.83183	Failure	−0.81328–0.89130
Z-axis range of controlled fluid particles	−0.56358–0.54136	−0.40315–0.63092	Failure	−1.00325–0.78151
Total frame	128	195	Failure	65
Time (s)	312	360	Failure	1560
Time per frame (s)	2.438	1.846	Failure	24

TABLE 2 The data of different experimental scenarios in our method.

Target model	Figure 7	Figure 8	Figure 9	Figure 6
Fluid particles	20,886	27,848	46,610	46,610
Control particles	6978	7706	4366	6428
Constraint	0.20	0.25	0.25	0.02
Controlled fluid particles	10,671	10,087	16,645	17,171
Y-axis range of control particles	0–1.875	0–1.811	0–2.779	0–2.812
X-axis range of control particles	−1.176–1.173	−0.778–0.801	−1.839–1.641	−0.788–0.857
Z-axis range of control particles	−0.535–0.515	−0.576–0.599	−0.557–0.562	−1.011–0.766
Y-axis range of controlled fluid particles	0–1.81865	0–1.81302	0–2.72202	0–2.79391
X-axis range of controlled fluid particles	−1.11028–1.19130	−0.78085–0.83875	−1.94563–1.58756	−0.81336–0.89225
Z-axis range of controlled fluid particles	−0.55711–0.542662	−0.57798–0.63307	−0.57812–0.58686	−1.01528–0.78232
Total frame	128	195	70	65
Time (s)	302	357	661	1318
Time per frame (s)	2.359	1.831	9.443	20.277

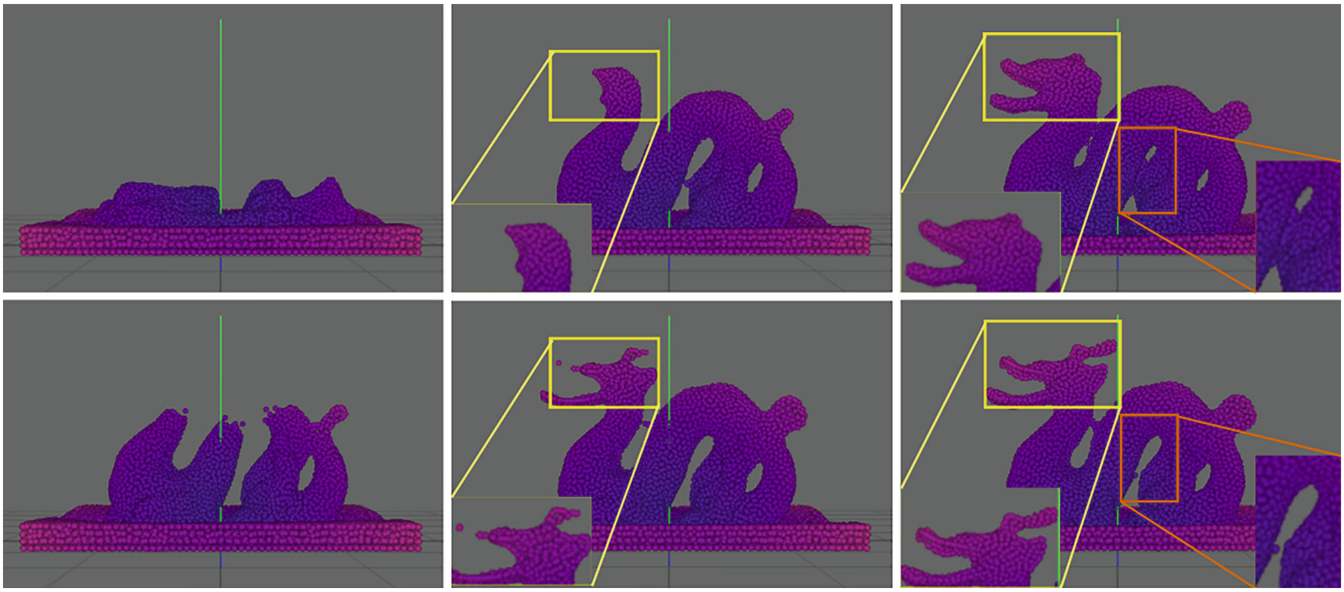

FIGURE 7 The PBFC (top) and our method (bottom) are compared in the process of driving and guiding the flow shape of the dragon. The first to third columns are frames 88, 102, and 128, respectively.

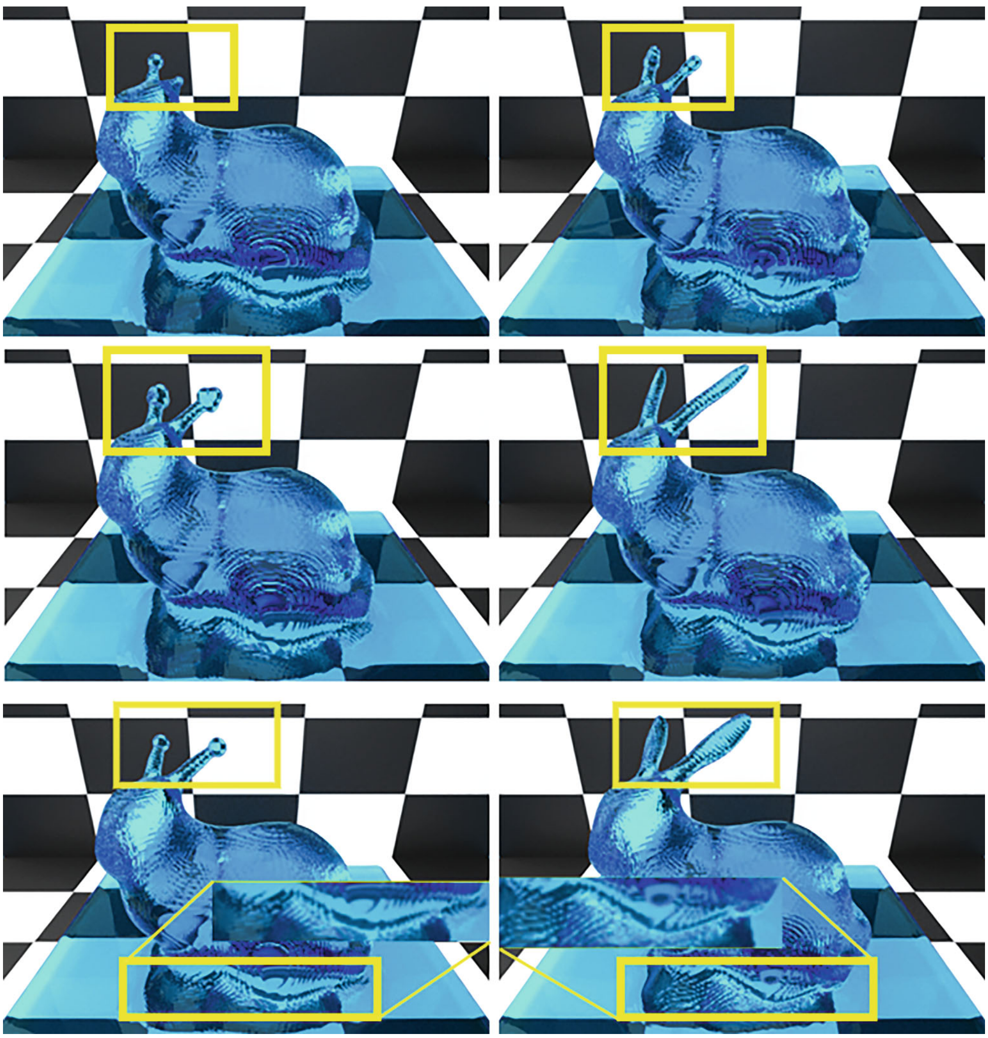

FIGURE 8 Comparison of the control result derived by the PBFC method (left) and our method (right) with a bunny-shaped target. Our method is able to handle the area with complex shapes like ear and foot, while the PBFC method cannot present these details accurately.

traditional scheme has not fully formed the head, meanwhile the proposed method has begun to drive the fluid toward the dragon's mouth and the dragon's horn. At the same time, the guidance effect of the proposed method on the final model is better than that of PBFC. At the 128th frame, the fluid guided by the proposed method has fully displayed the head details (circled in yellow), and in other body parts, there is no excessive constraint that leads to morphological expansion. The body parts will be connected (circled in orange). Under the premise of ensuring stability, the proposed method significantly improves the efficiency of the fluid guiding drives. Compared with PBFC, the formation process of the proposed method is faster.

Figure 8 shows the comparison of the effects of different methods used to generate the bunny model. At frame 78, after the head region was guided, both methods begin to guide the rabbit's ear region. Compared with PBFC, the proposed method is faster than the traditional scheme. In the 136th frame, both methods fill the rabbit ear of the target model to different degrees, but PBFC does not weight the control particles of the rabbit ear to enhance them, resulting in insufficient driving constraints. As circled in the figure, the rabbit ear under the action of PBFC appears the phenomenon of water droplets overflowing, while the proposed method has stronger constraints in this area. A natural form of transition is presented. At the bottom of the model, the proposed approach is more compact and fits the shape of the model better.

4.2 | Efficiency

Figure 9 shows the difference in the model for generating dog. Therefore, the proposed method is more efficient. By comparing 35 frames, 50 frames, and 70 frames of the simulation with a dog-shaped target model, it is clear

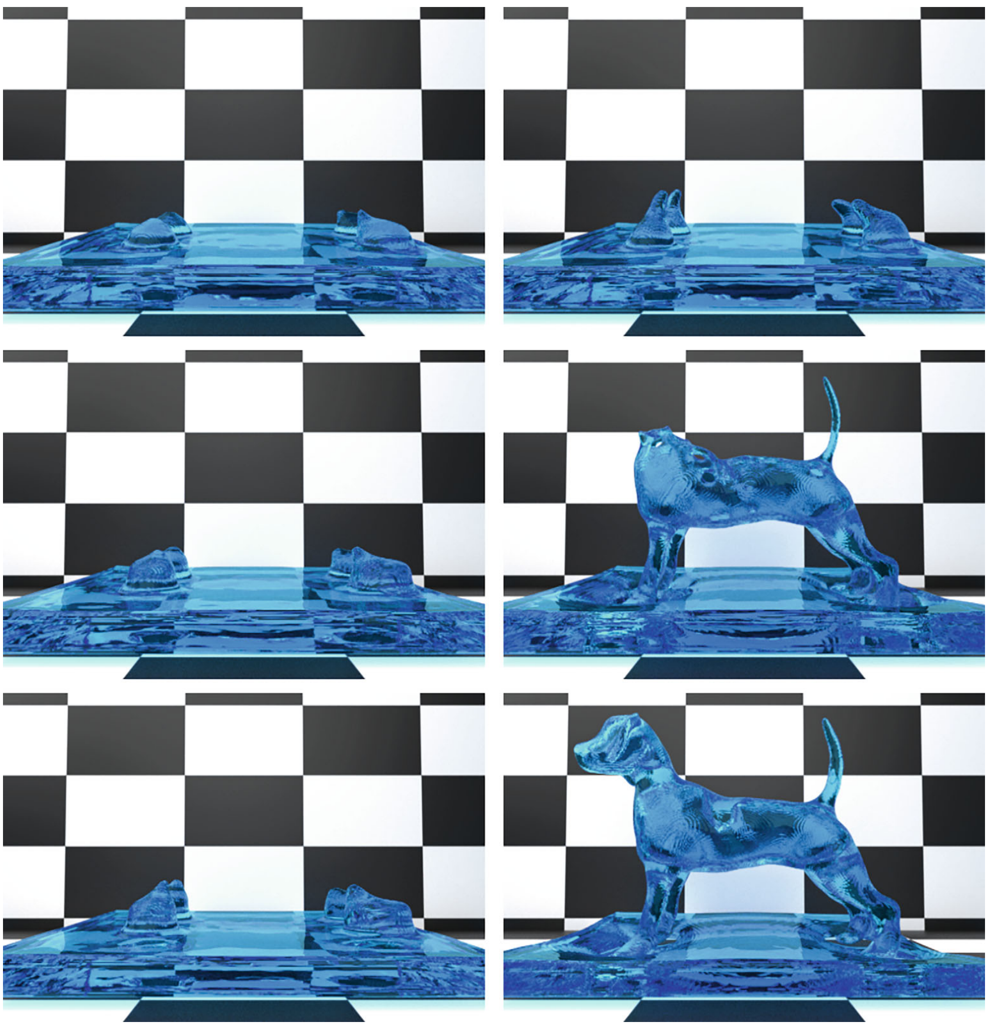

FIGURE 9 Comparison of the control result derived by the PBFC method (left) and our method (right) with a dog-shaped target. PBFC method cannot handle the shape with very narrow channels (dog legs), while our method can form the shape efficiently and accurately.

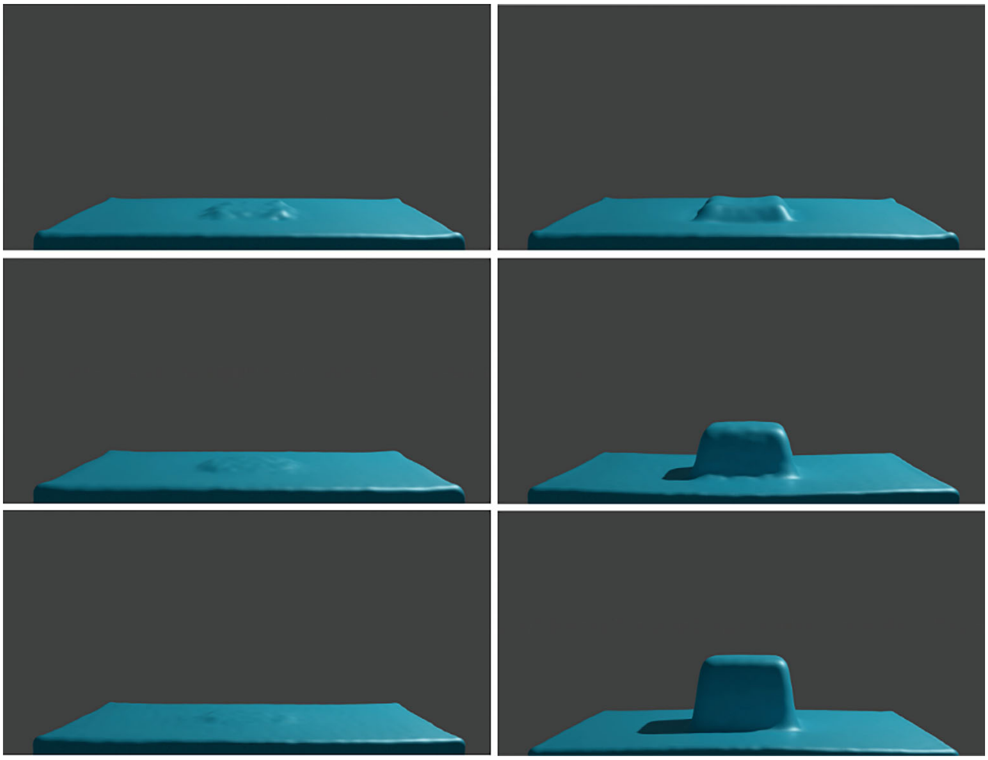

FIGURE 10 The comparison is made between the control results obtained using the LCAT (left) and our method (right) with the objective of generating a cube. LCAT method can only control neighboring fluid particles, so it requires raising the control particles from beneath the water surface. In terms of static shapes, LCAT method can only control the portions closest to the water surface. In contrast, our method focuses on static shape control and produces smoother shapes compared to LCAT method.

that the proposed approach, with the new adaptive constraints, can form a more complete model at every moment. The PBFC method runs to 421 frame and still does not fill the entire model. The proposed method has comparable computational time per frame compared to previous methods, but the fluid could be filled more rapidly with fewer frames.

4.3 | Realism

Figure 6 shows the difference between models that generate armadillo only with density-driven changes. The new density-driven constraint compared to the previous method, through the experiment it can be seen that there are better realism with the bottom details.

LCAT method can only enable control particles to influence neighboring fluid particles. If there are no fluid particles surrounding the control particles, the fluid particles will not be guided towards the control particles. As a result, the control cannot effectively shape static models and can only be applied for dynamic control, where the control particles are lifted from the water surface. However, the shapes generated through this approach have defects. Figure 10 shows the difference in the model for generating cube. Through our experiments, we have observed that even compared to dynamic control, our method produces smoother shapes, and the generated shapes are defect-free.

5 | CONCLUSION

In this paper, we have presented an accurate and controllable method for driving fluid simulation based on weighted spatial geometric features. This method is suitable for solving efficient driving fluid generation with shape features that better fit the target model while maintaining the characteristics realism of fluid flow. The proposed method mainly considers the depth characteristics of the target model space, and adaptively modifies the driving force of the driving surface, so as to have a specific driving constraint effect when the fluid moves to different spatial positions. At the same time, our method has advantages in dealing with detailed regions, which can generate more complex models in a short time without causing fluid instability due to insufficient control particles. Additionally, the experiment revealed that when faced with fluid particles that are too large in comparison to the sampling distance, PBFC methods tend to produce rough surfaces. In extreme cases, these methods may still result in insufficient filling. By comparison, the superiority of the proposed method is demonstrated. The LCAT method is effective in controlling motion states, but it lacks efficacy in regulating static shapes. Finally, our method is also flawed in some way. If the control time exceeds a certain threshold, it may lead to particle accumulation. This will be an area of focus for our future research.

ACKNOWLEDGMENTS

This research was funded by: National Key R&D Program of China (2022ZD0118001), National Natural Science Foundation of China grant (No. U22A2022), Postdoctoral Innovation Talents Support Program (BX20220007), Guangdong Basic and Applied Basic Research Foundation (2021A1515012285, 2022A1515110350, 2023A1515030177), H2020 Marie Skłodowska-Curie Individual Fellowship (895941).

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Xiangyang Zhou https://orcid.org/0000-0002-9614-939X
Xiaokun Wang https://orcid.org/0000-0002-4449-591X

REFERENCES

1. Zhang S, Yang X, Wu Z, Liu H. Position-based fluid control. Proceedings of the 19th Symposium on Interactive 3D Graphics and Games. 2015:61–68.
2. Lu JM, Chen XS, Yan X, Li CF, Lin M, Hu SM. A rigging-skinning scheme to control fluid simulation. Comput Graph Forum. 2019;38(7):501–12.
3. Schoentgen A, Poulin P, Darles E, Meseure P. Particle-based liquid control using animation templates. Comput Graph Forum. 2020;39(8):79–88.
4. Treuille A, McNamara A, Popović Z, Stam J. Keyframe control of smoke simulations. ACM Siggraph 2003 Papers. 2003: 716–723.
5. Jm H, Kim C. Controlling fluid animation with geometric potential. Comput Animation Virtual Worlds. 2004;15(3-4):147–57.
6. Bergou M, Mathur S, Wardetzky M, Grinspun E. Tracks: toward directable thin shells. ACM Trans Graph. 2007;26(3):50.
7. Chu M, Thuerey N, Seidel HP, Theobalt C, Zayer R. Learning meaningful controls for fluids. ACM Trans Graph. 2021; 40(4):1–13.
8. Pan Z, Huang J, Tong Y, Zheng C, Bao H. Interactive localized liquid motion editing. ACM Trans Graph. 2013;32(6):1–10.
9. Pan Z, Manocha D. Efficient solver for spacetime control of smoke. ACM Trans Graph. 2017;36(4):1.
10. Pan Z, Manocha D. Editing smoke animation using a deforming grid. Comput Visual Media. 2017;3:369–78.
11. Fattal R, Lischinski D. Target-driven smoke animation. ACM Siggraph 2004 Papers. 2004:441–448.
12. Raveendran K, Thuerey N, Wojtan CJ, Turk G. Controlling liquids using meshes. Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 2012.
13. Cornelis J, Ihmsen M, Peer A, Teschner M. Liquid boundaries for implicit incompressible SPH. Comput Graph. 2015;52:72–8.
14. Nielsen MB, Christensen BB, Zafar NB, Roble D, Museth K. Guiding of smoke animations through variational coupling of simulations at different resolutions. Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 2009:217–226.
15. Nielsen MB, Christensen BB. Improved variational guiding of smoke animations. Comput Graph Forum. 2010;29(2):705–12.
16. Nielsen MB, Bridson R. Guide shapes for high resolution naturalistic liquid simulation. ACM SIGGRAPH 2011 Papers. 2011:1–8.
17. Forootaninia Z, Narain R. Frequency-domain smoke guiding. ACM Trans Graph. 2020;39(6):1–10.
18. Sato S, Dobashi Y, Kim T. Stream-guided smoke simulations. ACM Trans Graph. 2021;40(4):1–7.
19. McNamara A, Treuille A, Popović Z, Stam J. Fluid control using the adjoint method. ACM Trans Graph. 2004;23(3):449–56.
20. Shi L, Yu Y. Controllable smoke animation with guiding objects. ACM Trans Graph. 2005;24(1):140–64.

21. Sato S, Dobashi Y, Yue Y, Iwasaki K, Nishita T. Incompressibility-preserving deformation for fluid flows using vector potentials. Visual Comput. 2015;31:959–65.
22. Thuerey N. Interpolations of smoke and liquid simulations. ACM Trans Graph. 2016;36(1):1–16.
23. Inglis T, Eckert ML, Gregson J, Thuerey N. Primal-dual optimization for fluids. Comput Graph Forum. 2017;36(8):354–68.
24. Rasmussen N, Enright D, Nguyen D, et al. Directable photorealistic liquids. Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 2004:193–202.
25. Stomakhin A, Selle A. Fluxed animated boundary method. ACM Trans Graph. 2017;36(4):1–8.
26. Foster N, Metaxas D. Controlling fluid animation. Proceedings Computer Graphics International. 1997:178–188.
27. Mihalef V, Metaxas D, Sussman M. Animation and control of breaking waves. Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 2004:315–324.
28. Bojsen-Hansen M, Wojtan C. Generalized non-reflecting boundaries for fluid re-simulation. ACM Trans Graph. 2016;35(4):1–7.
29. Manteaux PL, Vimont U, Wojtan C, Rohmer D, Cani MP. Space-time sculpting of liquid animation. Proceedings of the 9th International Conference on Motion in Games. 2016:61–71.
30. Stuyck T, Dutré P. Sculpting fluids: a new and intuitive approach to art-directable fluids. ACM Siggraph 2016 Posters. 2016:1–2.
31. Sato S, Dobashi Y, Nishita T. Editing fluid animation using flow interpolation. ACM Trans Graph. 2018;37(5):1–12.
32. Macklin M, Müller M. Position based fluids. ACM Trans Graph. 2013;32(4):1–12.
33. Foster N, Fedkiw R. Practical animation of liquids. Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. 2001:23–30.
34. Dionne O, Lasa DM. Geodesic voxel binding for production character meshes. Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 2013:173–180.
35. Müller M, Charypar D, Gross M. Particle-based fluid simulation for interactive applications. Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 2003:154–159.
36. Zhu Y, Bridson R. Animating sand as a fluid. ACM Trans Graph. 2005;24(3):965–72.

AUTHOR BIOGRAPHIES

Xiangyang Zhou is a postgraduate student in Artificial Intelligence Science and Engineering at University of Science and Technology Beijing. He received an undergraduate Diploma in Inner Mongolia University of Science and Technology, in 2022. His research field is computer graphics, especially physically-based fluid simulation.

Sinuo Liu is a postdoc in School of Computer Science, Peking University, China. She received a PhD degree in Computer Science and Technology from University of Science and Technology Beijing, in 2021. Her research interests include computer graphics and virtual reality.

Haokai Zeng is a postgraduate student in Artificial Intelligence Science and Engineering at University of Science and Technology Beijing. He received an undergraduate Diploma in University of Science and Technology Beijing, in 2023. His research field is computer graphics, especially physically-based fluid simulation.

Xiaokun Wang is an associate professor in School of Computer and Communication Engineering, University of Science and Technology Beijing, China. He received a PhD degree in Computer Science and Technology from University of Science and Technology Beijing, in 2017. His research interests include computer graphics, virtual reality and human-computer interaction.

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

How to cite this article: Zhou X, Liu S, Zeng H, Wang X, Ban X. Efficient and high precision target-driven fluid simulation based on spatial geometry features. Comput Anim Virtual Worlds. 2024;35(1):e2202. https://doi.org/10.1002/cav.2202