SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2022, Vol. 65 212102:1–212102:16
https://doi.org/10.1007/s11432-020-3322-x

Reconstructing and editing fluids using the adaptive multilayer external force guiding model

Xiaoying NIE[1], Yong HU[2,1], Xukun SHEN[1,2*] & Zhiyuan SU[1]

[1]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China;
[2]School of New Media Art and Design, Beihang University, Beijing 100191, China

Received 30 December 2020/Revised 23 April 2021/Accepted 30 June 2021/Published online 25 October 2022

Abstract Reconstructing monocular fluids usually involves a tedious trial-and-error process, and editing desired fluid behaviors is notoriously difficult to predict and control. To address these problems, we propose an adaptive multilayer external force guiding model that alleviates the challenging parameter tuning and satisfies user-defined requirements. External forces cause the effect of target particles on fluid particles. The adaptive multilayer scheme makes the whole 3D fluid volume subject to the shape and motion of the water captured by the input video or designed by users. Therefore, we can avoid the tedious and laborious parameter tuning and easily balance the smoothness of fluid volume and the details of the water surface. Simultaneously, to vividly reproduce the inflow and outflow of the video scene, we construct a generation and extinction model to add or delete fluid particles according to the three-dimensional velocity field of target particles calculated using a hybrid model coupling shape-from-shading with optical flow. Besides, we edit fluids by subtly extracting features selected by users using the off-screen rendering method and seamlessly integrating them using the dynamic weight approach. Experiments show that our approach is comparable to the state-of-the-art reconstruction quality and is remarkably convenient in authoring flows by editing fluids. Furthermore, our results can be effectively applied to any desired new scenario.

Keywords reconstructing fluids, editing fluids, monocular videos, adaptive multilayer external force guiding model, generation and extinction model

Citation Nie X Y, Hu Y, Shen X K, et al. Reconstructing and editing fluids using the adaptive multilayer external force guiding model. Sci China Inf Sci, 2022, 65(11): 212102, https://doi.org/10.1007/s11432-020-3322-x

1 Introduction

Monocular fluid reconstruction is challenging because of the complex topological variation and frequent occlusion of water [1–3]. Currently, there is no convenient method of editing fluids. The traditional physics-based simulation methods [4–6] are consuming, challenging and have tedious parameters tuning to generate desirable fluid behaviors captured from the real world [7]. The shape-from-shading (SfS) methods [8–10] can reconstruct the 3D surface of the water using its 2D intensity image. However, they are incapable of recovering the internal volume and physical property.

The latest research [11] has demonstrated the power of the external force scheme, which can solve 3D fluid reconstruction from a monocular video by constructing the mapping between the SfS method and the smoothed particle hydrodynamics (SPH) model. The external force scheme formulates the guidance of turning the input monocular video to a 3D fluid volume and enables the recovery of the water surface and volume. However, owing to the enormous height difference of the reconstructed water surface via SfS, maintaining the fluid surface details is difficult while eliminating the fluid volume holes. Consequently, to obtain satisfactory visual results, tedious and laborious trial-and-error parameter manually tuning steps are inevitable. In practice, the external force scheme requires considerable time overhead because besides parameter turning, fluid reconstruction can also consume several hours to check intermediate results.

In this paper, we considerably improve Nie's [11] work and propose an adaptive multilayer external force guiding model that compensates the imperfection in the height field, and decreases the effect of parameter

* Corresponding author (email: xkshen@buaa.edu.cn)

© Science China Press and Springer-Verlag GmbH Germany, part of Springer Nature 2022 info.scichina.com link.springer.com

values on the results. We follow the definition of fluid and target particles from [11] and combine SfS with the optical flow to estimate the position and velocity field of target particles. Furthermore, we adaptively interpolate the multilayer target particles with geometric and physical properties. To be consistent with the fluid scene in the input video, we revise the initialization of fluid particles in [11] and develop their generation and extinction model, whose inflow and outflow are embedded in the SPH model to form a stable fluid with open boundaries. Besides, we edit fluids based on the off-screen rendering technique and the dynamic weight method. Our work develops a low-consumption and easy-to-use reconstructing and editing framework to intuitively and efficiently reconstruct fluids from real-world videos; we further edit them under the users' specified target. The main contributions of this paper are as follows.

- We propose an adaptive multilayer external force guiding model to reconstruct and edit fluids. We show that the advantage of the "adaptive multilayer" framework based on the SPH model is that the guiding strategy can be significantly simplified because there is no need to tune parameters through laborious and tedious trial-and-error manual processes.
- We introduce a generation and extinction model of fluid particles to reproduce the inflow and outflow behaviors of the fluid in the video, preserving the observed fluid properties in the real world. We believe this is an essential step toward bridging the gap between the reconstructed result and the input video.
- We introduce the off-screen rendering technique and present the dynamic weight method to edit fluids. This method can extract features from arbitrary target particles and combine them to create the desired animation. Furthermore, we provide real-time visual feedback that allows users to set features in the required editing process.

2 Related work

In this section, we summarize previous studies that are related to our approach. We focus our discussion on physics-based simulation and video-based reconstruction approaches for monocular fluid reconstruction and fluid editing.

2.1 Physics-based simulation

In computer graphics communities, forward simulation of fluids has been an important research topic and a fast-growing research field. Physics-based simulation methods track the fluid motion to generate desirable effects by solving Navier-Stokes equations. To track smoke's 3D density and motion based on single-view video sequences, Eckert et al. [12] used Eulerian simulation algorithms to resolve motion uncertainty along the lines of sight in the single-camera view and disambiguate a sparse set of 2D observations. Takahashi et al. [7] proposed a real-to-virtual parameter transfer framework to determine the set of physical variables and viscosity parameters from example videos capturing fluid flows in real-world phenomena. Their method avoids tedious trial-and-error physics-based simulation parameter tuning steps.

Recently, physics-based fluid simulation has achieved interesting results by efficiently controlling the dynamic behavior of the fluid. The previous physics-based simulation work concentrated on the forward control method to control fluid motion using parameters tuning [13–16]. The forward control method requires a trial-and-error approach to achieve the desired scenes, because the efficiency is very low. Here, we aim to develop a reverse control method that will effectively control fluid keyframes and efficiently match the target shape [17, 18]. Similar to the keyframe-based control method, particle-based control method [19–21], guiding method [22–25], rigid inspired method [26–28], and skeleton-based control method [29] have performed well and generated desirable animation results, for instance, a fluid water man rising, dancing, fighting, and running, and a fluid character jumping, collapsing, and finally forming other characters.

Our work focuses on the natural water phenomena, which play an essential role in numerous special effect applications. For liquids, particle-based Lagrangian approaches such as the SPH method are trending [30] among researches in this area. Thürey et al. [19] used the SPH method to calculate the control force that makes the particles control the fluid to fill the target shape. Yang et al. [31] demonstrated how their pairwise-force SPH model could capture various interactions, such as interfaces between egg white, egg yolk, air, and solid, a stream of strawberry sauce flows over a ball, and the close connection between droplets and bubbles. SPH method can be implemented and has a beneficial effect on fluid detail expression and scalability.

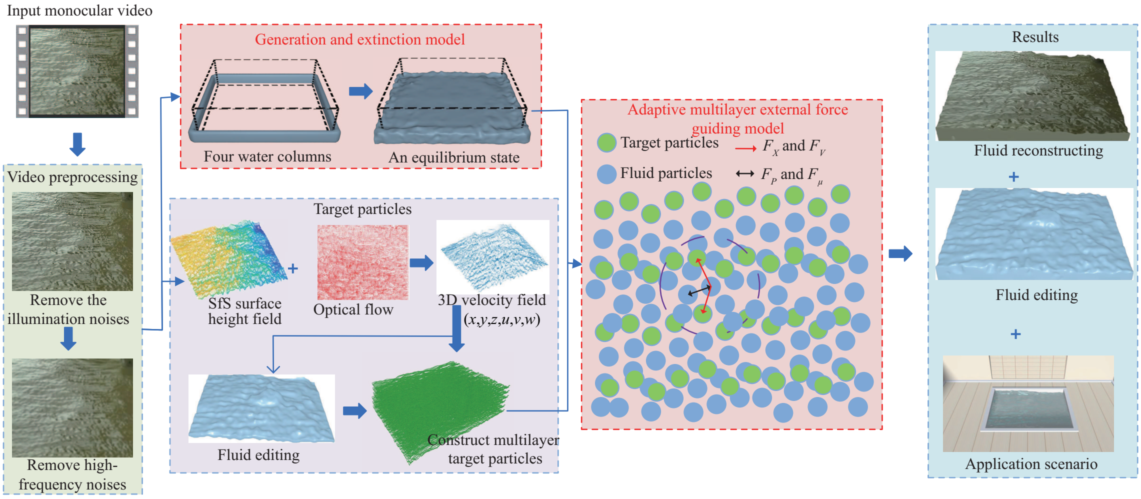


Figure 1 (Color online) An overview of our method. The input is a monocular video, and the output is a 3D fluid volume. Based on the denoising video, we construct fluid particles using the generation and extinction model and adaptively generate multilayer target particles with 3D position data and velocity field. Under the action of the adaptive multilayer external force guiding model, the reconstructed or edited 3D fluid volume can represent the geometric appearance and 3D motion of the video to life. Thus, it can be effectively applied to any desired new scenario.

2.2 Video-based reconstruction

Monocular video-based 3D fluid reconstruction approaches use SfS. The SfS method takes a single image [32] or a sequence of images as input and uses image formation assumptions to recover surface shape.

Murase [33] reconstructed the surface shape of water from clear motion of the observed pattern based on the optical and statistical analysis of the distortions. However, the limitations of the water are strict: (1) the average slant of the wave surface over a long time is zero; (2) the water is transparent and refractive; (3) the pattern in the water is static and fat. These assumptions make the method impossible to be applied to large-scale outdoor water.

Li et al. [34, 35] made significant progress in reconstructing the water surface from monocular video. They discovered that SfS could reconstruct the appearance and dynamic behaviors of water using example videos of the public DynTex dataset [36], which are recorded using a single static camera in the outdoor environment. Although water surface properties normally violate the Lambertian condition of SfS, the water scenes in this dataset [36] appear visually opaque owing to their depth and suspension of dirt, mud, and air. The SfS reproduces a water surface animation with a similar dynamic behavior. As dynamics of the surface (e.g., the evolution of waves with time) play a more significant role in human perception and are the more noticeable part, the geometric inaccuracies on such a free-form surface are considered inconsequential and within acceptable limits. Regrettably, Li et al. [34, 35] only reconstructed the geometry shape and 3D velocity field of a single-layer water surface, lacking further simulation of 3D fluid volume.

Following this landmark study, Nie et al. [11] introduced fluid external forces to a tailored combined SfS and SPH model, which successfully reconstructs the volume movement and surface details of the fluid based on a single monocular video. They used target particles to represent the surface height field recovered using SfS and fluid particles to represent 3D fluid volume. Then, they quantified the relationship between the 3D fluid volume and the input video by calculating the external forces applied by the target particles to the fluid particles. Their reconstructed approach can achieve desirable favorable effects in terms of global fluid volume motion features and fluid surface details. Nonetheless, this method requires elaborate manual parameter tuning. The tedious and laborious trial-and-error challenges human patience and persistence.

3 Overview

Our aim is to reconstruct and edit fluids as close as possible to the water phenomena captured from the monocular video or a user-desired target. Figure 1 illustrates that our framework comprises two stages: the particle preparation stage and the external force guidance stage. In the particle preparation stage,

Figure 2 (Color online) The sketches of target particles. The green dots represent target particles, and the blue dots represent fluid particles. (a) Target particles of [11] were placed directly above the 3D fluid volume; (b) the top sub-figure shows the smaller ΔH, and the bottom sub-figure shows the larger ΔH in [11]; (c) the target particles of our adaptive multilayer external force guiding model.

our framework takes a large-scale and opaque water video as input. First, we preprocess the video by removing the illumination and high-frequency noises. Then, we construct multilayer target particles with positional data and the 3D velocity field using the SfS and optical flow method. Simultaneously, we directly deform the corresponding target particles to edit the 3D fluid volume. Besides, we initialize the fluid particles based on the generation and extinction model. The external force guidance stage is an iterative process. First, using the SPH model, we perform fluid simulation by calculating the internal forces between fluid particles. Next, we employ target particles to guide fluid particles based on the adaptive multilayer external force guiding model to vividly reproduce the geometric appearance and motion field of the input video. Our framework iteratively adds or deletes fluid particles based on the generation and extinction model to realize the inflow and outflow of the 3D fluid volume.

4 Adaptive multilayer external force guiding model

We uniformly represent inputs as target particles $\mathcal{G} = \{g \in \mathcal{G} \mid g = 1, 2, 3, \ldots, n_g\}$, and sample the fluid continum to fluid particles $\mathcal{I} = \{i \in \mathcal{I} \mid i = 1, 2, 3, \ldots, n_i\}$. Here n_i and n_g denote the number of fluid particles and target particles, respectively. We transform the reconstructing and editing problems into the external forces of target particles on fluid particles. To achieve the desired behavior, each target particle g should act as a local magnet to attract nearby fluid particles. Moreover, the motions of target particles should be like wind force to drive fluid particles along their moving path [37]. Therefore, each fluid particle i is subject to two external forces of its neighbor target particle g:

$$F_X(i) = w_a \sum_{g=1}^{g \in \mathcal{G}} \alpha_g \frac{X_g - X_i}{||X_g - X_i||} W(X_{gi}, h), \tag{1}$$

$$F_V(i) = w_V \sum_{g=1}^{g \in \mathcal{G}} (V_g - V_i) W(X_{gi}, h), \tag{2}$$

where $F_X(i)$ and $F_V(i)$ are determined by the distance and velocity difference between the fluid particle i and the target particle g, respectively. $\alpha_g = 1 - \min(1, \sum_{i=1}^{n_i} \frac{m_i}{\rho_i} W(X_{gi}, h))$ scales down the $F_X(i)$ force when the target particle's influence region is already sufficiently covered with fluid. w_a and w_V are global constants that define the strength of the two external forces, respectively. $X_{gi} = ||X_g - X_i||$ denotes Euclidean distance between the target particle g and the fluid particle i. V_i, X_i, ρ_i and m_i denote the velocity, position, density, mass of the fluid particle i, respectively. V_g and X_g denote the velocity, position of the target particle g, respectively. h denotes the smooth radius, and W denotes the target particle kernel function. Here we use a normalized spline kernel [4, 19] to be the target particle kernel function the same as the density approximation.

In [11], target particles were placed directly above the 3D fluid volume as shown in Figures 2(a) and (b), so only the fluid particles on the surface are subject to external forces. They caused the parameter tuning process too time-consuming, especially the overall relative height difference ΔH of target particles and fluid particles. Smaller ΔH results in some holes in the 3D fluid volume, and larger ΔH results in too flat surface and misses surface details. Meanwhile, ΔH value is also related to the smooth radius h, the external force weight coefficients w_a and w_V, the time step Δt, and other physical parameters.

To address this problem, we adopt adaptive multilayer external forces to guide fluid particles on the surface and inside the 3D fluid volume. We only need to ensure target particles are just above the initialized 3D fluid volume as follows:

$$\Delta H \geqslant \max\ (z_i(t=0)), \tag{3}$$

where $z_i(t=0)$ is the vertical component of the position $X_i(x,y,z)$ of the fluid particle i in the initial state $t=0$. The max is a function of the maximum. Note, the initial state $t=0$ refers to the equilibrium state in the initialization stage of the 3D fluid volume (Figure 3(c)). The specific solution process of ΔH involves the following steps: determining the time $t=0$, counting the fluid depth at $t=0$, calculating the maximum value of the fluid depth $\max(z_i(t=0))$, determining the value range of ΔH according to (3). The value of the ΔH only needs to satisfy (3) in theory, but it is not appropriate to be too large, otherwise it will increase the calculation of particles, and it is slightly larger than $\max(z_i(t=0))$.

To enable $F_X(i)$ and $F_V(i)$ to act on the fluid particles inside of the 3D fluid volume, we interpolate target particles along with the depth of 3D fluid volumes as follows:

$$X_{g+\lambda}(x,y,z) = X_g(x,y,z-\lambda\times h), \tag{4}$$

where λ is an adaptive interpolation coefficient:

$$\lambda \leftarrow \lambda + 1, \tag{5}$$

and λ value is a non-negative integer. The number of layers of target particles is determined by the adaptive coefficient radius of λ. When $\lambda=0$, target particles are only single-layer as in [11]. The larger λ value, the more layers target particles have. So fluid particles will be guided by multilayer external forces. Note that the maximum value of λ needs to ensure that target particles cannot cross the bottom of the 3D fluid volume. As shown in Figure 2(c), these target particles are interpolated in the vertical direction as a whole without changing the relative horizontal position and velocity field. In (4), the smooth radius h determines the interpolation step size of multilayer target particles and determines the spacing of fluid particles. Accordingly, the target particles maintain spatial consistency with the fluid particles.

The set of multilayer target particles is $\hat{\mathcal{G}} = \{g \in \hat{\mathcal{G}} \mid (g = 1,2,3,\ldots,n_g,\ldots,1+\lambda\times n_g, 2+\lambda\times n_g,\ldots,n_g+\lambda\times n_g,\ldots)$. Accompanied by the generation of multilayer target particles, the 3D fluid volume is guided by multilayer external forces. Eqs. (1) and (2) can be written as

$$F_X(i) = w_a \sum_{g=1}^{g\in\hat{\mathcal{G}}} \alpha_g \frac{X_g - X_i}{||X_g - X_i||} W(X_{gi}, h), \tag{6}$$

$$F_V(i) = w_V \sum_{g=1}^{g\in\hat{\mathcal{G}}} (V_g - V_i) W(X_{gi}, h). \tag{7}$$

We embed the adaptive multilayer external force guiding model into the SPH method. The SPH governs fluid particles by calculating the internal forces, including pressure force F_P and viscosity effect force $F_\mu(i)$. F_P and $F_\mu(i)$ can be written as

$$F_P(i) = -\sum_{j=1}^{N} m_j \left(\frac{P_i + P_j}{2\rho_j}\right) \nabla W_{\text{spiky}}(X_i - X_j, h), \tag{8}$$

$$F_\mu(i) = \mu \sum_{j=1}^{N} m_j \left(\frac{V_j - V_i}{\rho_j}\right) \nabla^2 W_{\text{viscosity}}(X_i - X_j, h), \tag{9}$$

where j denotes the index of neighbor fluid particles of i, N is the number of neighbor particles, W_{spiky} and $W_{\text{viscosity}}$ are smoothing kernel functions, μ is viscosity coefficient, and P is pressure. The pressure field of fluid is computed by WCSPH [38]. Finally, the total force function for the fluid particle i can be described as follows:

$$F_{\text{total}}(i) = F_P(i) + F_\mu(i) + F_X(i) + F_V(i) + G(i), \tag{10}$$

where $G(i)$ is gravity.

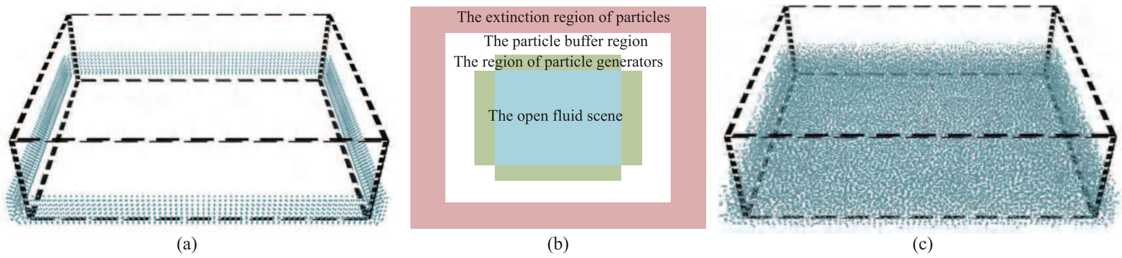


Figure 3 (Color online) The generation and extinction model of fluid particles. (a) Four particle generators (blue points) outside the open bounding box (black dashed frame); (b) the region of the generation and extinction; (c) an equilibrium state of the 3D fluid volume (blue points).

5 Particles representation

Our adaptive multilayer external force guiding model is based on SPH methods, so the initialization of fluid particles is essential. We propose a generation and extinction model to initialize fluid scenarios. Besides, we borrow the concept of target particles [11] to describe the inputs like videos or user's requirements. We combine SfS and optical flow to generate target particles from a video.

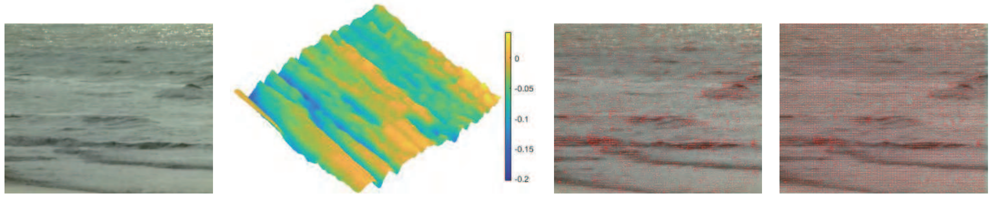
5.1 The generation and extinction model

Minor changes in the initial conditions can drastically change the resulting animation. Hence, the initialization of fluid particles has tremendous significance for the layout of the fluid scene. The up-to-date research [11] generated fluid particles according to the aspect ratio of video frames. These fluid particles move in a fixed bounding box to produce a surface appearance similar to target particles. For the water example with apparent inflow and outflow, a fixed number of fluid particles cannot accurately track the flow movement at a fixed boundary. To overcome these problems, we propose a generation and extinction model to add or delete fluid particles to approximate the water's dynamic variation in videos.

As shown in Figure 3(a), our scene is a bounding box whose size is proportional to the input video frame's height and width, and only the bottom is fixed, and the other five sides are open. Clinging close to the front, back, left, right outer edges of the bounding box, we place four particle generators. The particle generators iteratively generate fluid particles with the initial velocity of $V_i = (0, 0, 0)$ and the particle spacing of h (i.e., the smooth radius) according to a fixed frequency, as shown in Figure 3(a). The region length of each particle generator is set as the same as the corresponding bounding, and the width and height of each other are the same. The settings of the four particle generators are the same to ensure that the generation of particles in the four directions is balanceable, to avoid unexpected turmoil of the 3D fluid volume.

The generated particles flow to both sides of particle generators under the action of gravity, which will add particles to our scene and form the phenomenon of fluid inflow. The particles flow out to the boundary during the free motion of particles, and the newly generated particles also flow out of the boundary under the push of the outflow particles. These outflow particles form the phenomenon of fluid outflow. Open fluid boundaries and fluid generators have met the demand for water inflow and outflow. The open fluid boundaries and particles generators have already met the demand of water inflow and outflow, but the reconstruction will become more and more time-consuming with the increase of the number of particles. In addition, it makes little sense to calculate the physical quantities of particles beyond the fluid scene area. To solve this problem, we introduce the extinction model. As shown in Figure 3(b), when a fluid particle flows out of the central region surrounded by particle generators for a certain distance, the life cycle of the particle is considered being over and the particle is deleted from the reconstruction system. We set a particle buffer region with a width of ω between the particle generation region and the extinction region to avoid the newly generated particles being immediately deleted in subsequent iterations, resulting in the loss of support to the boundary and the excessive loss of fluid particles.

Throughout the fluid reconstruction, we add the particles entering the bounding box into the 3D fluid volume and treat this process as the generation of fluid particles, that is, the inflow phenomenon of fluid. Likewise, we remove the particles that flow out the bounding box boundary from the 3D fluid volume and treat this process as the extinction of fluid particles, that is, the outflow phenomenon of fluid.



(a) (b) (c) (d)

Figure 4 (Color online) The height field and velocity field of target particles. (a) Input video frame; (b) the height field; (c) $(u_{g=k}, v_{g=k})$; (d) $(\hat{u}_{g=k}, \hat{v}_{g=k})$.

In the initialization stage of the 3D fluid volume, it is reasonable to generate a certain amount of particles within the scene boundary before entering the reconstruction stage. This is because fluid reconstruction without particles is meaningless. We drive the particles flowing into the bounding box to reach an equilibrium state (Figure 3(c)), i.e., relative stillness only under the action of gravity and internal forces in SPH. There is no violent movement such as water splashing in the equilibrium state. In this way, it can also suppress the falling, splashing, collision of the generated particles caused by the influence of gravity during the reconstruction and editing stage.

5.2 Target particles generation

To reconstruct the desired liquid motion from the input video, the generation of target particles is of utmost importance. In general, users often design expected flow shapes using an implicit function, a sequence of target shapes or another fluid simulation to generate target particles [19]. As with [11, 35], we sample the height field (as shown in Figure 4(b)) of water surface acquired by the SfS method to produce target particles. Thus, each target particle's position information is obtained and expressed as $X_g(x, y, z)$.

In this paper, we use (u, v, w) to represent the velocities along (x, y, z) directions. In [11], they ignored the horizontal velocity u and v of target particles when external forces were applied. Therefore, fluid particles were only affected by the force F_V of the target particle in the vertical direction. Here, the fluid particles only moved up and down in the z-direction, without front, back, left and right movements in the x- and y-directions. This was extremely inconsistent with the fluid scenes with inflow and outflow.

To solve this issue, we combine SfS and optical flow to estimate the 3D velocity field $V_g(u, v, w)$ of the target particle. This hybrid algorithm is described in detail in [34]. As shown in Figure 4(c), in the area with gentle motion, the horizontal velocity field approaches $(0, 0)$. This result seriously violates water motion's physical laws; that is, there are no stationary particles in the flowing water. Since the fluid's motion in the peaceful area accords with the overall fluid scene's moving trend, we use the global direction of motion and the average speed to amend the velocity field in this area. As shown in Figure 4(d), the amending problem is formulated as

$$(\hat{u}_{g=k}, \hat{v}_{g=k}) = \left(\frac{\sum_{g=1}^{n_g} u_g}{n_g - n_0}, \frac{\sum_{g=1}^{n_g} v_g}{n_g - n_0} \right), \quad (u_{g=k}, v_{g=k}) \to (0, 0), \tag{11}$$

where $(\hat{u}_{g=k}, \hat{v}_{g=k})$ denotes corrected horizontal velocity field, k denotes the index of the target particles. n_0 denotes the number of particles whose horizontal velocity field approaches $(0, 0)$. To reduce the error effect of the wrong velocity field, we subtract the number of such particles when calculating the flow field's average velocity; otherwise, the average velocity value will be relatively smaller.

6 Fluid editing

Authoring fluid to match what the user intended is notoriously tricky. There is to date no convenient way of editing a fluid. Here we edit the 3D fluid volume by directly deforming the corresponding target particles. First, we accurately extract users' interest feature area based on the off-screen rendering technique. Then, we realize the seamless fusion of features based on the dynamic weight method.

6.1 Feature extraction based on off-screen rendering

Our feature extraction method is to identify and display target particles by drawing twice in OpenGL drawing mode. The first drawing is the off-screen rendering that helps select objects by drawing some auxiliary graphics in the frame cache. These auxiliary graphics are invisible. So after determining the selected features, we should remove the auxiliary graphics from the frame cache before the second drawing. The second drawing is the on-screen rendering. Here we utilize the screen space fluid rendering method to display target particles in real time.

The process of the feature extraction method based on off-screen rendering is as follows. The first step is to draw the auxiliary graphics. Auxiliary graphics usually use color to distinguish between selected and unchecked objects. We paint the 'Fountain' feature area in red and the other target particles in blue. The red area's target particles are the selected feature points and constitute an auxiliary graphic in the off-screen rendering. Since our input is a set of target particle sequences, we use the ID value of particles for coloring to ensure the selected feature area's consistency. The second step is to search for target particles. By traversing all particles, we determine whether to select the particle depending on the auxiliary graphic's information, such as the color of its position. In the last step, we render all target particles using the SSFR method in our system. To achieve real-time editing, we interactively select desired features from the top view perspective. Users click one pixel on the screen, and then resize the operations' radius, so they could locate the feature area.

6.2 Feature fusion based on dynamic weight

Our feature selection method can select any particles' area, such as the edited area $\mathbb{M}$. Given an interpolation weight in [39], they produced an intermediate space-time mesh by linearly interpolating between the positions of vertices of $\mathbb{S}$ and their corresponding points on $\mathbb{M}$. This blending approach is completely unproductive because we focus on fluid editing in a local region. In [40], they progressively dilated the grid form of $\mathbb{M}$ until it filled the bounding box of size $\mathbb{S}$. Then, they oriented the displacements to deform the $\mathbb{M}$ by using the difference of normals between $\mathbb{S}$ and $\mathbb{M}$. However, our extent of the editing area $\mathbb{M}$ completely depends on users' preferences and wishes and is not affected by the size of the selected feature area $\mathbb{S}$. The method in [40] is just a special case that the edited area is the same size as the selected feature area $\mathrm{size}(\mathbb{M}) = \mathrm{size}(\mathbb{S})$. Moreover, our editing object is particles, rather than grids.

To realize the seamless fusion of $\mathbb{S}$ and $\mathbb{M}$, we propose a feature fusion method based on the dynamic weight ζ as follows:

$$H_{\widehat{\mathbb{M}}} = H_{\mathbb{M}} \times (1-\zeta) + H_{\mathbb{S}} \times \zeta, \tag{12}$$

where $\widehat{\mathbb{M}}$ is the edited area after fusion features. And H is the height field and attracts our undivided attention. The horizontal range of $\widehat{\mathbb{M}}$ is determined only by the user-defined $\mathbb{M}$ that $r(x,y)_{\widehat{\mathbb{M}}} = r(x,y)_{\mathbb{M}}$. ζ is a dynamic weight which is associated with the distance $\mathcal{D}$ between the particle g in the edited area and the edge of the edited area $\mathbb{M}_{\mathrm{edge}}$.

Inspired by the linked-list nearest neighboring particle searching algorithm in the SPH method, we propose a definition of the distance $\mathcal{D}$ between g and $\mathbb{M}_{\mathrm{edge}}$. We overlay a temporary mesh on the edited area $\mathbb{M}$ and the selected feature area $\mathbb{S}$. Because the target particle is a single layer of water surface, a two-dimensional horizontal mesh is enough to location g's position and edge $\mathbb{M}_{\mathrm{edge}}$. We initialize each grid value using the edited area's height field $H_{\mathbb{M}}$. Then for a given target particle g ($g \in \mathbb{M}$), we take it as the origin and count separately the total number of particles (signified by C_1, C_2, C_3, C_4) in each grid cell in its four quadrants. We use the minimum C_1, C_2, C_3, C_4 to define the distance $\mathcal{D}$ between the particle g in the edited area and the edge of the edited area $\mathbb{M}_{\mathrm{edge}}$.

$$\mathcal{D} = \min(C_1, C_2, C_3, C_4). \tag{13}$$

According to (12), when g is close to the edge $\mathbb{M}_{\mathrm{edge}}$, ζ is as small as possible to ensure the seamless fusion of $\mathbb{M}$ and $\mathbb{S}$. When g is within the $\mathbb{M}$, the bigger ζ is better to retain the information of $\mathbb{S}$ as much as possible. Given a threshold ϖ of particles number according to statistics of all C_1, C_2, C_3, C_4, we quantitatively estimate g and $\mathbb{M}_{\mathrm{edge}}$ as

$$\begin{aligned} &\zeta \propto 1/\mathcal{D}, \quad \text{when } \mathcal{D} < \varpi, \\ &\zeta \propto \mathcal{D}, \quad\quad \text{when } \mathcal{D} > \varpi. \end{aligned} \tag{14}$$

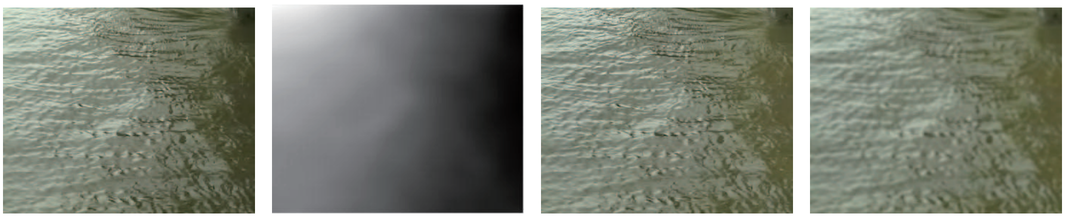


Figure 5 (Color online) Video preprocessing. First, we extract the illumination component (b) of input video (a). Then, we remove the illumination noise (c). Finally, we remove the high-frequency noise (d).

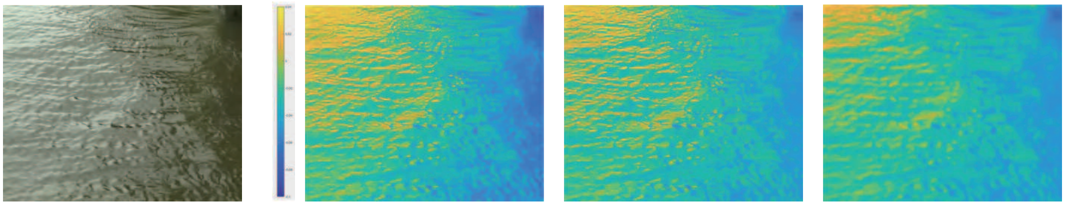


Figure 6 (Color online) Comparison between the effect of video denoising on the height field recovered by the SfS method. (a) The original video; (b) before video denoising; (c) after removing the illumination noise; (d) after removing the video noise.

Eq. (14) can be unified into

$$\zeta = \min\ (\mathcal{D}, \varpi) \times \tau, \tag{15}$$

where τ is the proportion coefficient.

7 Results

We tested our method on a PC with a 3.6 GHz Intel Core i7-7700 CPU. We implement our iterative process with C++ and CUDA. Houdini render is utilized to render our results. In the following, we first remove the video noise and verify our reconstruction method's reliability with experiments on the synthetic data, and then present comparisons to state-of-the-art studies. Subsequently, we evaluate our model with fluid videos in the public Dyntex dataset [36] or on the Internet. Each video is about 10 s (250 frames) in length, and resolution is 352×288. The overall computation time depends on the video resolution, the number of video frames, and the experimental environments. Then, we show the editing effect of the fluid. Finally, we provide some application scenarios for the reconstructed 3D fluid volume.

7.1 Video preprocessing

Owing to the influence of weather, illumination, and reflective characteristics of the object surface, the illumination distribution in the obtained fluid video is not uniform. It not only seriously affects the visual effects of an image but also leads to a distorted surface height field during the capture and reconstruction processes (Figure 5(b)). We use the Gaussian function to extract the illumination component of the water video and then use their distribution characteristics to adjust the 2D gamma function parameters so that the intensity value can be increased in low-illumination regions and decreased in high-illumination regions [41–45]. As shown in Figures 5(b) and (c), this can reduce the negative impacts such as distortion and aberration from non-uniformity of illumination conditions and enhance the visual quality across different water scenarios.

There are also high-frequency phenomena such as foam, splash, or spray in video frames, which cause a large amount of noise in the acquired height field and significantly reduce the accuracy of the fluid reconstruction (Figures 6(b) and (c)). For fluid reconstruction, the low-frequency variation is often the most exciting factor in the time series, while the high-frequency fluctuation quickly hides the visual movement trend. For this reason, we use low-pass Gaussian filtering to blur water image and control the standard deviation of the Gaussian to determine smoothing [46–49]. As shown in Figure 6(d), we remove high spatial frequency components from an image and preserve movement trends and appearance of water details simultaneously.

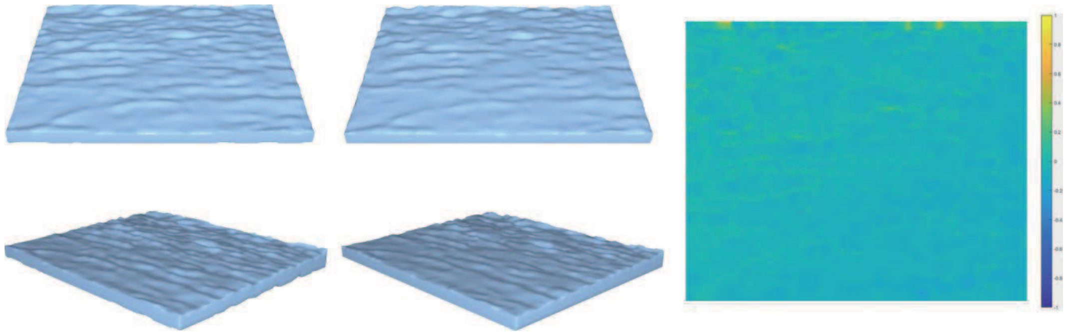


Figure 7 (Color online) Validation with ground truth. Multiple comparisons between synthetic data (first column) and our reconstructed 3D fluid volume (second column). The third column is the residual depth.

Table 1 Statistics results of the depth errors

Frame number	Error	Frame number	Error	Frame number	Error
1	0.0052	61	0.0032	121	0.0036
11	0.0026	71	0.003	141	0.0018
21	0.0026	81	0.0023	151	0.002
31	0.0067	91	0.0039	161	0.0035
41	0.0074	101	0.0027	171	0.0023
51	0.0022	111	0.0083	181	0.021

In Figure 6, we compared the effect of video denoising on the height field recovered by the SfS method. Figure 6(b) shows the height field recovered directly from a frame of the original video (Figure 6(a)). The height field of the high-illumination regions is larger than low-illumination regions. The denoising for the illumination reduces the height field difference, as shown in Figure 6(c). Since white splashes have no concept of height field, we filter out this high-frequency noise to eliminate their negative affects on the height field (Figure 6(d)).

7.2 Validation with synthetic data

To verify our model, we generated a video using our SPH fluid solver and used the synthetic data as input. This experiment validates that our 3D fluid volume can reproduce the geometric appearance and 3D motion field. As shown in Figure 7, the first column shows the ground truth, and the second column shows the reconstructed 3D fluid volume. The first and second rows show comparisons from different render views. The third column shows the residual depth of this frame from a top view. The visual differences between the ground truth and the reconstructed result are not significant. For each frame, we evaluate the depth error of the ground truth z_i^{GT} and the reconstruction results z_i^{RE} based on the formula $e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(z_i^{\mathrm{GT}} - z_i^{\mathrm{RE}})^2}$ [11]. Table 1 shows the quantitative analysis of the reconstructed errors. The maximum standard deviation of depth is 0.021. This shows that the deviation from the reconstruction results is slight and acceptable.

7.3 Comparative experiments to verify the innovation points

We compare with [11] using the same fluid video with the identical running environment. To reconstruct a similar 3D fluid volume as shown in Figure 8, Nie et al. [11] spent an average of 291.1 s, while we took an average of 220.3 s. Figure 9 shows the time-consuming contrast of reconstructing a video sequence. Although our adaptive multilayer external force guide model requires more target particles, our generation and extinction model reduces the number of fluid particles, thus immensely reducing time consumption. However, this is only the consumption when Nie et al. [11] chose the appropriate parameters that take into account both the volume smoothness and the surface details. Even worse, the time, physical and mental sports involved in trial-and-error parameter tuning steps behind are immeasurable, taxing, and oppressive.

Our adaptive multilayer external force guiding model solves the difficulty of balancing holes and details. This is because our model expands the influence range of external forces so that the whole 3D fluid volume is uniformly guided. The advantage of our model makes it easy to determine an appropriate overall relative

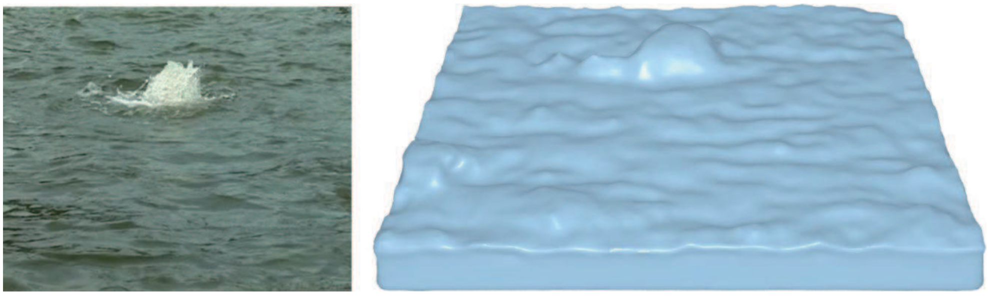


Figure 8 (Color online) A result example of Nie et al. [11]. The left is the input and the right is the reconstructed 3D fluid volume.

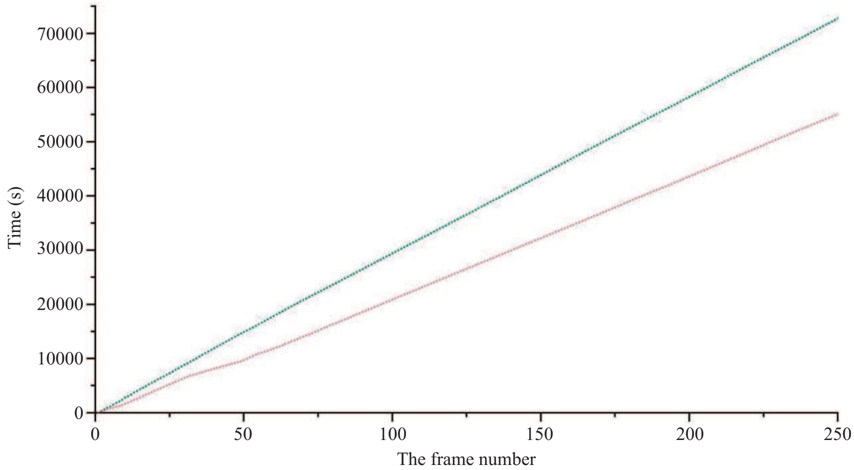


Figure 9 (Color online) The time-consuming comparison between [11] (green) and our method (red).

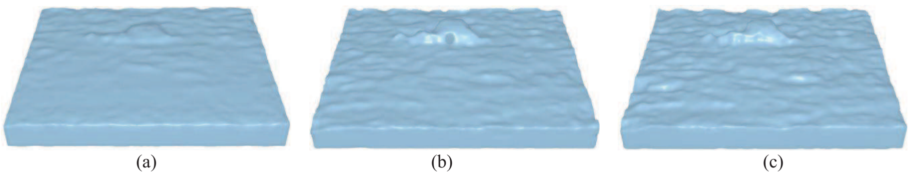


Figure 10 (Color online) Comparison between Nie et al.'s method and ours. (a) $\Delta H = 0$ of Nie et al.'s. The reconstructed result is too smooth and missing surface details because the ΔH value is enormous. (b) $\Delta H = -0.015$ of Nie et al.'s. There are some holes in the 3D fluid volume because the ΔH value is small. (c) Our reconstruction result. We preserve rich details and maintain the smoothness of the 3D volume.

height ΔH of target particles and fluid particles. Even ΔH is too large, the surface of the 3D fluid volume can also be correctly guided. This is because multilayer target particles are generated downward from the surface layer for external force guidance. Figures 10(a) and (b) show Nie et al.'s reconstruction results in the same rendering environment, where $\Delta H = 0$ and $\Delta H = -0.015$, respectively. As shown in Figure 10(a), the reconstructed surface is very flat. This is due to the extreme value of ΔH, which results in a significant portion of the surface fluid particles not being guided by sufficiently large external forces. There are some obvious holes on the surface in Figure 10(b). This is because target particles only guide the surface fluid particles but have a feeble influence on the internal fluid particles. This leads to the inner hollow of the protruding surface and results in the instability of the geometric details of the fluid surface. Figure 10(c) shows our reconstruction result. Because of the multilayer external forces, the whole 3D fluid volume is guided to preserve rich details and maintain the smoothness of the 3D volume.

Furthermore, we designed an experiment to verify the effectiveness of our generation and extinction model. In the initialization stage of the 3D fluid volume, a ball drops into the still water. Then, the ball particles move respectively under the action of Nie et al.'s [11] model and our model. Figure 11 shows the four states of the ball's falling process, falling into the 3D fluid volume, guiding movement, and the end of reconstruction. The first row shows the results of Nie et al. We can see that the number of ball

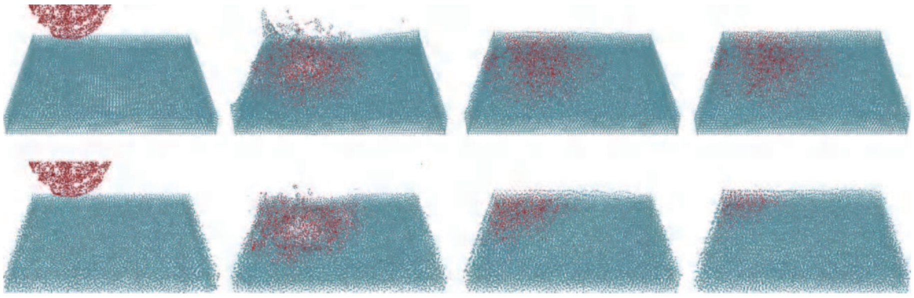


Figure 11 (Color online) The verified experiment of the generation and extinction model. The first row shows Nie et al.'s results, and the second row shows our results. The columns show the four states of the ball particles (red points) at the corresponding moment. The blue points denote the 3D fluid volume.

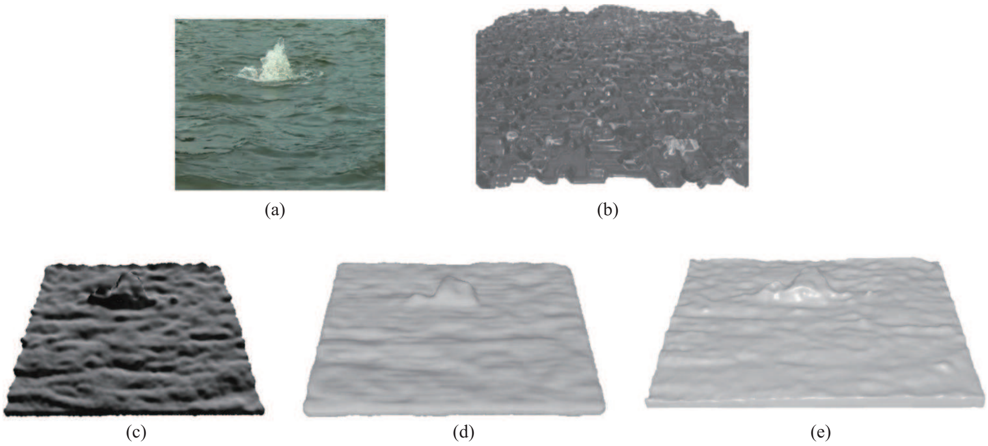


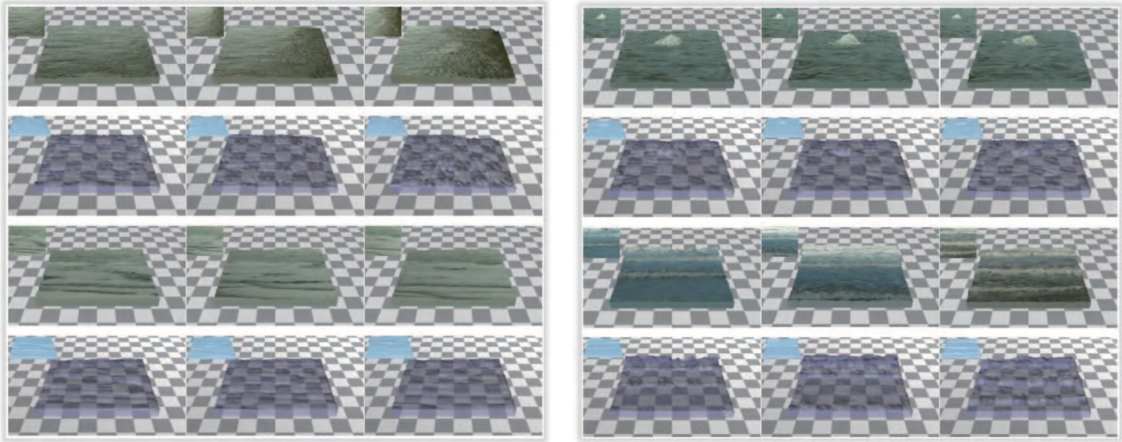
Figure 12 (Color online) The qualitative comparsion between different methods. (a) Input video frame; (b) Wang et al. [51]; (c) Nie et al. [11]; (d) Hu et al. [50]; (e) our method.

particles has not changed, and their horizontal positions have changed a little. The second row shows our results. The number of ball particles has an apparent decrease, and these particles have a distinct horizontal motion. Our result vividly reproduces the inflow and the outflow of the video scene. This significant promotion owes to our generation and extinction model and the guidance of target particles' 3D velocity field.

Here, we provide qualitative analysis of our model, using comparisons with alternative approaches. Figure 12 visually compares the output models from different methods. It can be seen that Nie et al. [11] reconstructed oversharpened 3D fluid volumes while Hu et al. [50] reconstructed oversmoothed 3D fluid volumes. And Wang et al.'s [51] reconstructed result is very different from the original frame. In contrast, our model appears sharper than [50] and smoother than [11], which indicates that a more complete solution is found.

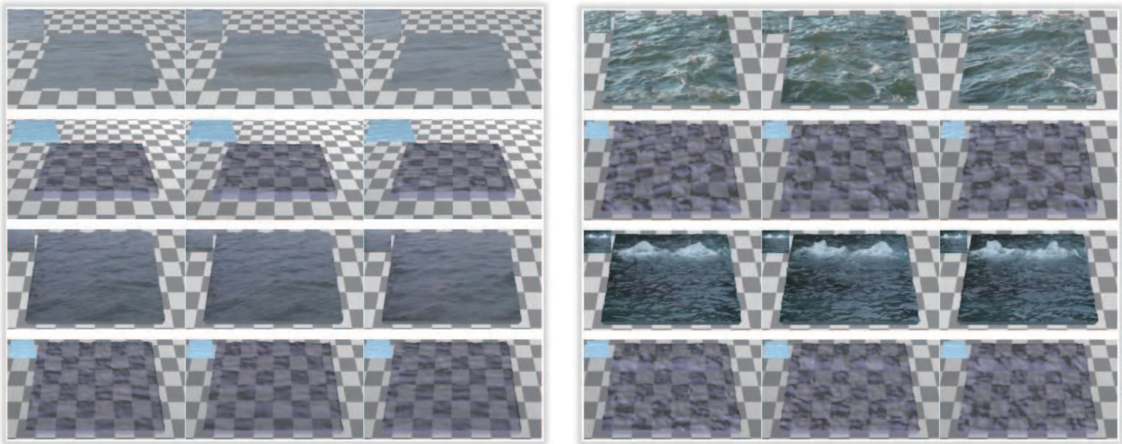
7.4 Reconstruction from real video

We selected real-world videos from the Dyntex dataset (13) and the Internet (14). We design two sets of experiments: gentle motions (Figures 13(a) and 14(a)) and complex motions (Figures 13(b) and 14(b)). First, we use pure SPH simulation to initialize the fluid particles. We set the time step to 0.01 ($\Delta t = 0.01$). Based on the generation and extinction model, it takes about 150 iterations to obtain a relatively still 3D fluid volume. The time step of the reconstruction process is 0.002 ($\Delta t = 0.002$). We adopt an adaptive time step selection strategy. A large time step is used to accelerate the simulation of the initialization phase, which does not require high precision, and a small-time step is used to improve the accuracy of reconstruction. For each frame, target particles guide 20 iterations of the fluid particles. This can maintain time consistency with the frame rate of the video. For all videos, we adopt $w_a = 0.018/h$, $w_V = 0.005/h$ to define the weight of $F_X(i)$ and $F_V(i)$, respectively. Our results (Figures 13 and 14) show that our algorithm is capable of realistically reconstructing several large-scale liquid phenomena.



(a) (b)

Figure 13 (Color online) Our reconstructed results from the Dyntex dataset. Given three frames of each monocular video (on the top-left corner of each picture in the first and third rows), we render them with the same texture (the first and third rows) as the input video to show the 3D motion field. Our method generates a series of opaque and transparent 3D fluid volumes (the second and fourth rows) to illustrate the surface details better. (a) Gentle motions; (b) complex motions.



(a) (b)

Figure 14 (Color online) Our reconstructed results from the Internet. The four videos are the Yangtze River, the Weizhou Island, the Yalu River, and the Baotu Spring from left to right and from top to bottom, respectively. (a) Gentle motions; (b) complex motions.

7.5 Fluid editing

In Figure 15, we illustrate the capability of our method to extract features based on off-screen rendering and combine them to create desired target particles based on dynamic weight. We develop a practical edit system to deform target particles in real time, which integrates feature selection and feature fusion as shown in Figure 15. Users can import target particles' files, select, extract and save their interest features, and import new scenes to merge features with these scenes seamlessly. The edit system can automatically implement batch processing.

We use the edited area after feature fusion $\widehat{\mathbb{M}}$ to realize the fluid edit based on the adaptive multilayer external force guiding model. We extract 'Fountain' and 'Surge' features in batches. Then, we apply 'Fountain' features to the original scene (Figure 16(a)) and new scene (Figure 16(b)). In addition, we edit 'Fountain' scene (Figure 16(c)) by fusing 'Surge' features.

7.6 Application scenarios

It is worth mentioning that the reconstructed 3D fluid volume can be flexibly utilized, e.g., for re-simulation, domain modification, or guiding purposes, and has wide applications in films, cartoons, computer games, AR, and so on. Figure 17 shows some application examples such as the fabulous indoor

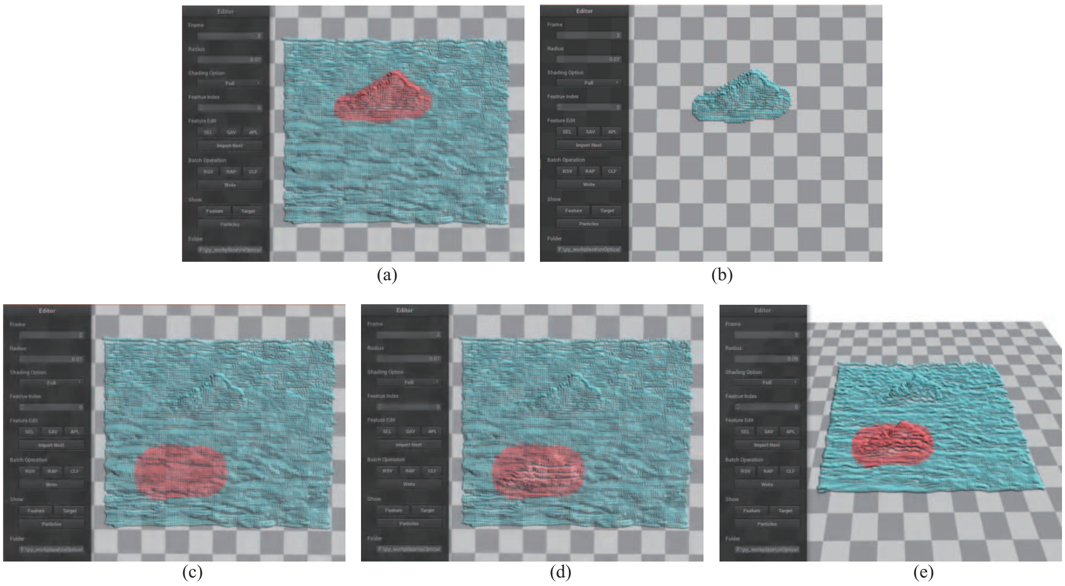


Figure 15 (Color online) Deformation of target particles. (a) and (b) detail the feature extraction. (c) and (d) show the feature fusion. First, users select the 'Fountain' feature (a) and obtain the result of the feature extraction $\mathbb{S}$ (b). Then, users choose the edited area $\mathbb{M}$ from a new scene or a new position (c). Finally, the selected feature area $\mathbb{S}$ and the edited area $\mathbb{M}$ are fused and transformed into the edited area after feature fusion $\widehat{\mathbb{M}}$ (d). (e) shows the fusion result $\widehat{\mathbb{M}}$ from a fresh perspective.

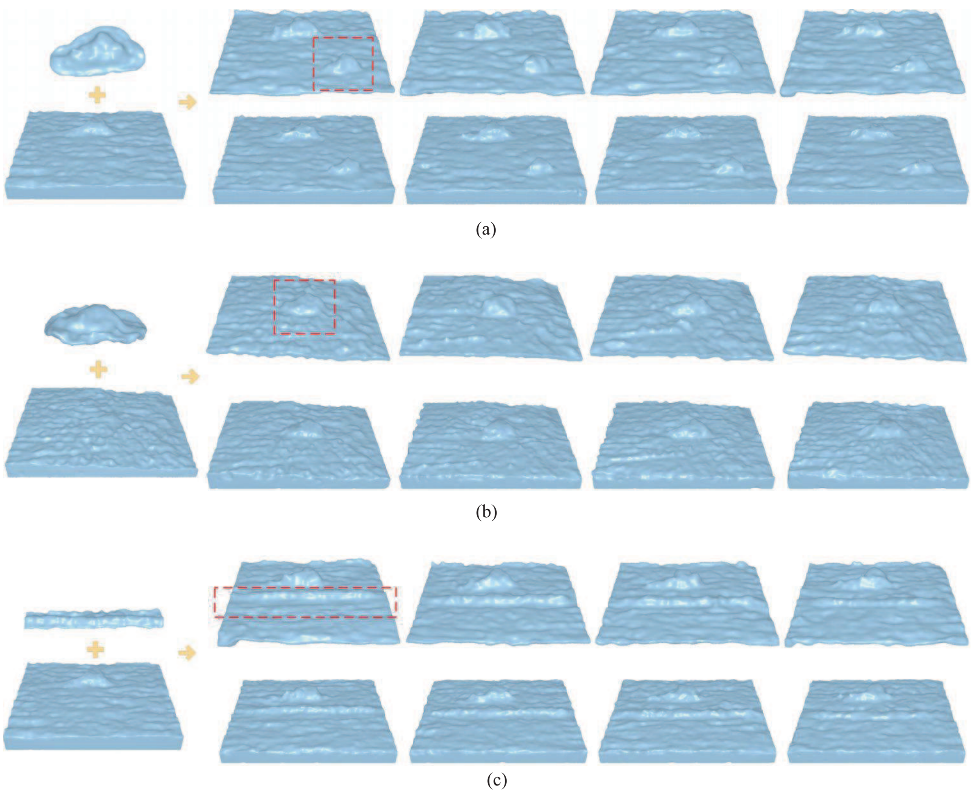


Figure 16 (Color online) Fluid editing results. Each set of results shows features (the first column of the first row), the scene to be edited (the first column of the second row), the deformed target particle (the second to fifth columns of the first row), and the edited three-dimensional fluid (the second to fifth columns of the second row). (a) Applying 'Fountain' features to the Fountain scene; (b) applying 'Fountain' features to the Gentlewaves scene; (c) applying 'Surge' features to the Fountain scene.

or square Fountain (Figure 17(a)), the stunning shopping mall Gentlewaves (Figure 17(b)). Slightwaves of the beach (Figure 17(c)) in the summer let us feel comfortable and cozy. The 3D fluid volume of the Surge blends and harmonizes with the bridge (Figure 17(d)) and presents us with a fast-flowing and

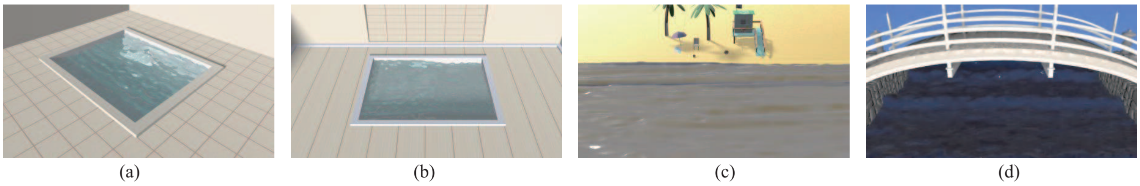


Figure 17 (Color online) Application scenarios of the reconstructed results. (a) Fountain; (b) Gentlewaves; (c) Slightwaves; (d) Surge.

treacherous river.

8 Conclusion

Our approach efficiently reconstructs 3D fluid volumes with rich surface details and no loopholes and it conveniently edits user-specified fluid behaviors. Our framework considers the monocular fluid video as a reference to guide the 3D fluid volume simulation. We proposed an adaptive multilayer external force guiding model to promote the guidance, which appends the position traction and velocity guidance of the two external forces to SPH, avoiding the trial-and-error parameter tuning step. Simultaneously, we simulated the inflow and outflow behaviors of the 3D fluid volume using the 3D velocity field of target particles as well as the generation and extinction model of fluid particles. Moreover, we confirmed that our model could efficiently generate fluid behaviors visually, which was consistent with the requirements of the videos or the user.

However, our approach has a few limitations, which can be addressed in future work. We subtly set up the generation and extinction of fluid particles, but the addition and deletion of fluid particles inevitably led to the 3D fluid volume boundary jitters. Further refinement of the generation and extinction model would be a promising future research direction. Moreover, the input videos have no background, which weakens the universality of our reconstruction method. Thus, exploring some techniques to obtain meaningful and reliable fluid information from general videos available in public would be necessary. Furthermore, we would like to advocate seeking novel strategies for the water surface reconstruction, such as investigating the machine learning approach [52, 53] and advanced computer vision techniques, to replace the SfS method that is limited by several assumptions.

Acknowledgements This work was supported in part by National Key R&D Program of China (Grant No. 2019YFF0301305) and National Natural Science Foundation of China (Grant No. U19A2063). The authors sincerely thank the anonymous reviewers for their kind suggestions.

References

1 Wang H, Liao M, Zhang Q, et al. Physically guided liquid surface modeling from videos. ACM Trans Graph, 2009, 28: 1–11

2 Tewari A, Zollhöfer M, Garrido P, et al. Self-supervised multi-level face model learning for monocular reconstruction at over 250 Hz. In: Proceedings of 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, 2018. 2549–2559

3 Thapa S, Li N, Ye J. Dynamic fluid surface reconstruction using deep neural network. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, 2020. 21–30

4 Müller M, Charypar D, Gross M H. Particle-based fluid simulation for interactive applications. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, 2003. 154–159

5 Foster N, Metaxas D. Realistic animation of liquids. Graph Model Image Process, 1996, 58: 471–483

6 Stam J. Stable fluids. In: Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, 1999. 121–128

7 Takahashi T, Lin M C. Video-guided real-to-virtual parameter transfer for viscous fluids. ACM Trans Graph, 2019, 38: 1–12

8 Zhang R, Tsai P-S, Cryer J E, et al. Shape-from-shading: a survey. IEEE Trans Pattern Anal Machine Intell, 1999, 21: 690–706

9 Tsai P-S, Shah M. Shape from shading using linear approximation. Image Vision Comput, 1994, 12: 487–498

10 Dou P, Wu Y, Shah S K, et al. Monocular 3D facial shape reconstruction from a single 2D image with coupled-dictionary learning and sparse coding. Pattern Recogn, 2018, 81: 515–527

11 Nie X, Hu Y, Su Z, et al. External forces guided fluid surface and volume reconstruction from monocular video. In: Proceedings of Pacific Graphics Short Papers, Seoul, 2019

12 Eckert M L, Heidrich W, Thürey N. Coupled fluid density and motion from single views. Comput Graph Forum, 2018, 37: 47–58

13 Foster N, Fedkiw R. Practical animation of liquids. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, 2001. 23–30

14 Rasmussen N, Enright D, Nguyen D Q, et al. Directable photorealistic liquids. In: Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Grenoble, 2004. 193–202

15 Angelidis A, Neyret F, Singh K, et al. A controllable, fast and stable basis for vortex based smoke simulation. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Vienna, 2006. 25–32

16 Kim Y, Machiraju R, Thompson D S. Path-based control of smoke simulations. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Vienna, 2006. 33–42

17 Treuille A, McNamara A, Popović Z, et al. Keyframe control of smoke simulations. ACM Trans Graph, 2003, 22: 716–723

18 Pan Z, Huang J, Tong Y, et al. Interactive localized liquid motion editing. ACM Trans Graph, 2013, 32: 1–10

19 Thürey N, Keiser R, Pauly M, et al. Detail-preserving fluid control. Graph Model, 2009, 71: 221–228

20 Zhang X, Liu S. SPH fluid control with self-adaptive turbulent details. Comp Anim Virtual Worlds, 2015, 26: 357–366

21 Zhang X, Liu S. Parallel SPH fluid control with dynamic details. Comput Anim Virtual Worlds, 2018, 29: e1801

22 Nielsen M B, Christensen B B, Zafar N B, et al. Guiding of smoke animations through variational coupling of simulations at different resolutions. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, New Orleans, 2009. 217–226

23 Nielsen M B, Bridson R. Guide shapes for high resolution naturalistic liquid simulation. ACM Trans Graph, 2011, 30: 1–8

24 Huang R, Melek Z, Keyser J. Preview-based sampling for controlling gaseous simulations. In: Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Computer Animation, Vancouver, 2011. 177–186

25 Feng G, Liu S. Detail-preserving SPH fluid control with deformation constraints. Comput Anim Virtual Worlds, 2018, 29: e1781

26 Zhang G, Lu D, Zhu D, et al. Rigid-motion-inspired liquid character animation. Comput Anim Virtual Worlds, 2013, 24: 205–213

27 Lu J M, Chen X S, Yan X, et al. A rigging-skinning scheme to control fluid simulation. Comput Graph Forum, 2019, 38: 501–512

28 Ma P, Tian Y, Pan Z, et al. Fluid directed rigid body control using deep reinforcement learning. ACM Trans Graph, 2018, 37: 1–11

29 Zhang G, Zhu D, Qiu X, et al. Skeleton-based control of fluid animation. Vis Comput, 2011, 27: 199–210

30 Ihmsen M, Orthmann J, Solenthaler B, et al. SPH fluids in computer graphics. In: Proceedings of the 35th Annual Conference of the European Association for Computer Graphics, Strasbourg, 2014

31 Yang T, Martin R R, Lin M C, et al. Pairwise force SPH model for real-time multi-interaction applications. IEEE Trans Visual Comput Graph, 2017, 23: 2235–2247

32 Snape P, Zafeiriou S. Kernel-PCA analysis of surface normals for shape-from-shading. In: Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, 2014. 1059–1066

33 Murase H. Surface shape reconstruction of an undulating transparent object. In: Proceedings of Third International Conference on Computer Vision, Osaka, 1990. 313–317

34 Pickup D, Li C, Cosker D, et al. Reconstructing mass-conserved water surfaces using shape from shading and optical flow. In: Proceedings of the 10th Asian Conference on Computer Vision, Queenstown, 2010. 189–201

35 Li C, Pickup D, Saunders T, et al. Water surface modeling from a single viewpoint video. IEEE Trans Visual Comput Graph, 2013, 19: 1242–1251

36 Péteri R, Fazekas S, Huiskes M J. DynTex: a comprehensive database of dynamic textures. Pattern Recogn Lett, 2010, 31: 1627–1632

37 Madill J, Mould D. Target particle control of smoke simulation. In: Proceedings of Graphics Interface 2013, Regina, 2013. 125–132

38 Becker M, Teschner M. Weakly compressible SPH for free surface flows. In: Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, San Diego, 2007. 209–217

39 Raveendran K, Wojtan C, Thürey N, et al. Blending liquids. ACM Trans Graph, 2014, 33: 1–10

40 Manteaux P, Vimont U, Wojtan C, et al. Space-time sculpting of liquid animation. In: Proceedings of the 9th International Conference on Motion in Games, Burlingame, 2016. 61–71

41 Jian M, Guo F, Yin C, et al. Automatic correction of non-uniform illumination for 3D surface heightmap reconstruction. In: Proceedings of the IEEE International Conference on Multimedia and Expo, New York City, 2009. 1402–1405

42 Liu Z C, Wang D W, Liu Y, et al. Adaptive adjustment algorithm for non-uniform illumination images based on 2D Gamma function (in Chinese). Trans Beijing Inst Tech, 2016, 36: 191–196, 214

43 Rahman S, Rahman M M, Abdullah-Al-Wadud M, et al. An adaptive gamma correction for image enhancement. J Image Video Proc, 2016, 2016: 35

44 Wang D, Yan W, Zhu T, et al. An adaptive correction algorithm for non-uniform illumination panoramic images based on the improved bilateral gamma function. In: Proceedings of 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2017. 1–6

45 Han P, Wang D, Yang X, et al. An improved adaptive correction algorithm for non-uniform illumination panoramic image. In: Proceedings of IEEE 2nd International Conference on Electronic Information and Communication Technology (ICEICT), 2019. 258–262

46 Karmakar S, Ghosh K, Sarkar S, et al. Design of a low-pass filter by multi-scale even order Gaussian derivatives. Signal Process, 2006, 86: 3923–3933

47 Ruoho-Airola T, Salmi T, Amnel T. Visualization of air quality time series by low-pass Gaussian filtering. In: Proceedings of the 20th International Conference for Environmental Protection, Graz, 2006. 291–294

48 Qu F, Ren D, Liu X, et al. A face image illumination quality evaluation method based on Gaussian low-pass filter. In: Proceedings of the 2nd IEEE International Conference on Cloud Computing and Intelligence Systems, Hangzhou, 2012. 176–180

49 Guo X, Li Y, Ma J, et al. Mutually guided image filtering. IEEE Trans Pattern Anal Mach Intell, 2020, 42: 694–707

50 Nie X, Hu Y, Shen X. Physics-preserving fluid reconstruction from monocular video coupling with SFS and SPH. Vis Comput, 2020, 36: 1247–1257

51 Wang C, Wang C, Qin H, et al. Video-based fluid reconstruction and its coupling with SPH simulation. Vis Comput, 2017, 33: 1211–1224

52 Li Z, Murez Z, Kriegman D J, et al. Learning to see through turbulent water. In: Proceedings of IEEE Winter Conference on Applications of Computer Vision, Lake Tahoe, 2018. 512–520

53 Stets J D, Li Z, Frisvad J R, et al. Single-shot analysis of refractive shape using convolutional neural networks. In: Proceedings of IEEE Winter Conference on Applications of Computer Vision, Waikoloa Village, 2019. 995–1003