

Video-based fluid reconstruction and its coupling with SPH simulation

Chao Wang¹ · Changbo Wang^{1,2}  · Hong Qin³ · Tai-you Zhang¹

© Springer-Verlag Berlin Heidelberg 2016

Abstract Conventional methods to create fluid animation primarily resort to physically based simulation via numerical integration, whose performance is dominantly hindered by large amount of numerical calculation and low efficiency. Alternatively, video-based methods could easily reconstruct fluid surfaces from videos, yet they are not able to realize two-way dynamic interaction with their surrounding environment in a physically correct manner. In this paper, we propose a hybrid method that combines video-based fluid surface reconstruction and popular fluid animation models to compute and re-animate fluid surface. First, the fluid surface's height field corresponding to each video frame is estimated by using the shape-from-shading method. After denoising, hole-filling, and smoothing operations, the height field is utilized to calculate the velocity field, where the shallow water model is adopted. Then we treat the height field and velocity field as real data to drive the simulation. Still, only one layer of surface particles is not capable of driving the smoothed particle hydrodynamics (SPH) system. The surface particles (including 3D position and its velocity) are then employed to guide the spatial sampling of the entire volume underneath. Second, the volume particles corresponding to each video frame are imported into the SPH system to couple with other

possible types of particles (used to define interacting objects), whose movement is dictated by the direct forcing method, and fluid particles' geometry information is then corrected by both physical models and real video data. The resulting animation approximates the reconstruction surface from the input video, and new physically based coupling behaviors are also appended. We document our system's detailed implementation and showcase visual performance across a wide range of scenes.

Keywords Video-based reconstruction · Height field · Shallow water · SPH · Two-way coupling · Physically based simulation

1 Introduction and motivation

Liquids are ubiquitous in our everyday life and fluid simulation is indispensable not only in engineering applications, but also in graphics field such as special visual effects in movies or video games. During the last two decades, modeling and simulating fluid behaviors remains a difficult problem and continues to attract significant attention with growing interest and progresses in graphics. Although computational fluid dynamics (CFD) is a well-established research area with a long history, there are still many open research problems about natural phenomena that we would like to model and simulate realistically in interactive graphics applications. In this paper, we focus on the fluid–solid coupling using recovered data from recorded videos.

The complex movement of liquid and all possible rules to characterize it make it impossible to formulate the whole equations of motion without sacrificing accuracy to a certain extent. Numerous numerical methods have been proposed to approximate fluid motion, and in the simulation approaches, there are several competing techniques for liquid simula-

Electronic supplementary material The online version of this article (doi:[10.1007/s00371-016-1284-2](https://doi.org/10.1007/s00371-016-1284-2)) contains supplementary material, which is available to authorized users.

✉ Changbo Wang
cbwangcg@gmail.com

- ¹ School of Computer Science and Software Engineering, East China Normal University, Putuo District, Shanghai, China
- ² MOE International Joint Lab of Trustworthy Software, East China Normal University, Shanghai, China
- ³ Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY 11794-4400, USA

tion with a variety of trade-offs. These physically based methods are based on simulating fluid dynamics from the initial state of a fluid scene. When we examine different forms of discretization that approximate numerical solutions resulted from the Navier–Stokes (N-S) equation, the physically based methods are roughly divided into three categories: lattice Boltzmann method (LBM) [4, 5, 20], Eulerian method [6, 7, 29, 34], and Lagrangian method such as SPH [1, 9, 11, 18, 19, 28]. Although to date, physically based methods are now becoming the mainstream for generating realistic fluid animations, and the three methods documented above all have their own limitations. The main limitations of LBM are the poor scalability and small time-steps. The Eulerian method suffers from lengthy computational time, aliasing boundary discretization, and poor scalability. Carefully designed smoothing kernel, compressibility, and blobby surface effects have also confined the application scopes of Lagrangian method. The three types of methods heavily rely on the initial state, which is tedious to set if we wish to obtain a specific fluid scene animation. Besides, these approaches can suffer from numerical errors that accumulate over time, including volume loss and loss of surface details. The computational cost is another issue in physically based fluid simulation, since the governing partial differential equations are expensive to solve and the time-steps need to be sufficiently small to maintain stability and accuracy.

Alternatively, another type of fluid animation could be enabled by the video-based reconstruction method. In order to acquire the details of fluid dynamic scenes, the video data can be captured by a single hand-held camera, binocular stereo cameras, or camera array from different views, and the 3D shape information of the scene could be estimated from videos. With the help of video sequences as input, higher quality can be obtained and both geometric accuracy and visual quality can be improved by exploiting the redundancy of data. Although the reconstruction techniques have been developed for many years, the reconstruction of fluid also gives rise to special challenges [32]. Fluid’s complex dynamics, complex topological change over time and frequent occlusions make it extremely difficult to match and track features when using stereo-matching reconstruction methods. It is also difficult to approximate the realistic water due to generally non-Lambertian appearance inaccuracy and the loss of small-scale details. In the video-based methods, user interaction and user-assisted model refinement are oftentimes required to refine the initial 3D shape to obtain high-quality models. Moreover, the reconstructed results are merely one layer of surface. It lacks volumetric information and cannot be practically utilized for physical interaction with their immediate surroundings. For example, the work in [16] only uses the velocity field to simulate virtual object floating and ignores the object’s weight, which is not a real physical interaction at all.

In this paper, we present the idea of combining video-based reconstruction with physically based simulation to model dynamic water surface geometry information from video, and then use the geometry information to realize physically based simulation and fluid–solid interaction. That is, we adapt video-based reconstruction methods as a correction tool to constrain the water surface that couples with other rigid bodies. The shape-from-shading (SFS) method is first applied to reconstruct the surface’s height field. In the height field optimization step, we remove redundant errors, by applying the hole-filling techniques and smoothness constraints. Then the shallow water equation is employed to estimate velocity field between two reconstructed surfaces of adjacent frames. The reconstructed surface information spreads across the entire 3D volume and is imported into the SPH model to interact with rigid bodies. Visually plausible fluid–solid coupling animations are devised as our experimental results. The entire process is automatic and efficient. It may be note that, it is also possible to accelerate such synthesized results using graphics hardware. To the best of our knowledge, our work presented in this paper represents the initial attempt that aims to reuse reconstructed fluid surface information in its physically plausible coupling with fluid’s numerical simulation and our method has the following advantages:

- *Video-based data correction* Our system’s input involves a recorded video with fluid content, together with virtually synthesized objects. The height field and velocity field are estimated from input video with high fidelity. The surface geometry is discretized and expanded into 3D volume, which is regarded as real data to guide fluid particles’ movement. We allow the recovered fluid surface to serve as boundary condition towards the recover of the entire 3D volume information.
- *Two-way coupling* We intend to seek the possible combination of video-based reconstruction techniques and physically based simulation. The surface geometry recovered from the input video is employed to serve as boundary condition, which helps discretize the 3D volume. A synthesized solid can be tightly coupled with the fluid volume, and such coupling is implemented in our experimental results.
- *Detail preservation* The movement of fluid particles in the simulation loop is constrained by the recovered real data and the SPH model. The direct force method is adopted to analyze force effect on particles of non-fluid objects. In this way, the final dynamic water model matches with the high fidelity of real video data and the synthesized results are physically sound. The surface details could be preserved and the system performs consistently well across a wide range of water movement manifested in various videos.

2 Related work

In the field of physically based fluid simulation, the most commonly used methods for numerically approximating fluid motion are Lagrangian method, Eulerian method [15, 34], and LBM [20, 25]. Specifically, Lagrangian approaches consider the continuum as a particle system, such as the SPH method. WCSPH method is proposed in [2] that utilizes the Tait equation with a high speed of sound resulting in a weakly compressible formulation with very low density fluctuations. Solenthaler et al. [28] solve the shallow water equations using 2D SPH particles and simulate height field fluids. Nadir Akinci et al. [1] propose a momentum-conserving two-way coupling method for SPH fluids and rigid bodies that is completely applied to hydrodynamic forces. A detailed survey is introduced in [11]. Besides, Ladický et al. [14] propose a novel machine learning-based approach, which formulates physics-based fluid simulation as a regression problem by estimating the acceleration of every particle for each frame. Nevertheless, physically based simulation methods heavily rely on the initial state, and it is difficult to obtain a specific fluid scene animation subject to the user-specific design.

In contrast, video-based methods for modeling and simulating fluid adopt a completely different strategy by analyzing surface data to obtain fluid surface geometry from video inputs. The most important advantage of using the input of video sequences is the higher quality of fluid motion one could obtain. The best frames in the video could be selected to get better visualization quality. Related techniques, generally called shape-from-X methods, have been proposed to recover surface geometry information from images or videos, where X could be stereo [8], distortion [12], shading, and so on. Wang et al. [32] present a hybrid framework to efficiently reconstruct realistic water surface geometry from real water scenes by using video-based reconstruction techniques, together with physically based surface optimization. Nevertheless, their experiments are under laboratory conditions and not suitable for outdoor scenes. In contrast, SFS [33] is a critical shape recovery technique based on single image or video recording the ordinary scenes. Tan et al. [30] introduce a linear approach for shape from shading, and this approach is first applied to the discrete approximations for surface normal based on finite difference. Pickup et al. [23] make the first attempt to show the simultaneous reconstruction of 3D surface geometry and velocity from single input video by combining SFS and optical flow to derive a water model that is incompressible in real world. Li et al. [16] introduce a video-based approach for producing water surface models, where SFS is used to capture the water surface geometry first, and then the strategy is to apply the shallow water model to estimate the 3D velocity that is missing from the SFS raw data. Briefly summarizing the above work, these reconstruction methods are capable of acquiring more

surface detail information, but the acquired and computed results are merely one layer of the surface (i.e., thin surface sheet), which in principle is lack of volumetric information and could not be practically utilized in any type of two-way dynamic interaction with its surrounding environments.

There are also other techniques that combine video data and physically based simulation together to generate fluid animations. Kwatra et al. [13] reconstruct 3D geometry of real scenes from binocular video sequences and then simulate one-way coupling from the video to the fluid. Their rendered output is constrained by the original viewpoints of the cameras. Comparing to their work, we reconstruct 3D geometry of fluid from monocular video and focus on new interaction phenomena corrected by video data and SPH model. The output animations can be generated at any angles.

Our current work in this paper makes new efforts towards significant improvement: the reconstructed surface information is expanded into 3D volume, which is treated as boundary constraints in the data acquisition process as well as in the volumetric data initialization process, and the physically based simulation method (i.e., the SPH model) is applied to achieve the fluid–solid coupling in order to obtain the new animation that is exhibiting real dynamic interaction with two-way coupling and physically plausible behaviors.

3 System overview

The input of our system is single fluid video stream and the output is a new animation video that contains new coupling behaviors between fluid and virtual solid. As shown in Fig. 1, it contains two main components, including the reconstruction part for volumetric data recovery and the simulation part for physically coupling.

The first three steps of the system process are designed to acquire the water surface geometry, including the height field and velocity field. Tsai's linear approximation method [24] is implemented to recover the height field. After obtaining the height field, the post-processing stage comprises the denoising, hole-filling, and smoothing operations in order to eliminate noises and consolidate the acquired data into a clean and reusable state. The underlying physics model utilized to estimate the velocity field is the equation of shallow water, which is a special case of water simulation that affords fast computation and high-precision water geometry [21]. Then the surface geometry is further expanded into 3D volume. This way, the fluid surface guides the formation of a volumetric data set and one layer of surface expands to many layers. Here the volumetric data are capable of coupling with (virtual) solids in the subsequent processes.

The second key component employs the SPH model to simulate the coupling between solid and reconstructed volumetric data set. The rigid object is also decomposed

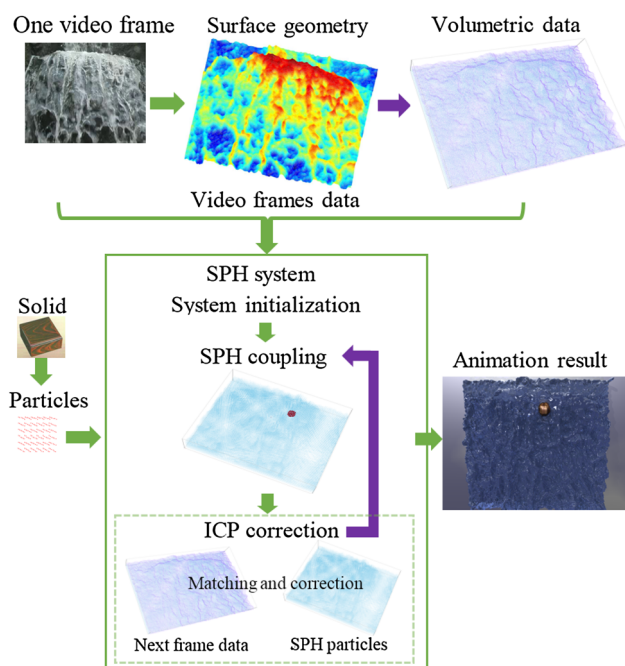


Fig. 1 The system accepts a single input video (a set of frames) as input and outputs a 3D water animation. The whole process contains two main steps: the first step applies the SFS technology, shallow water equation and surface discretization method to handle each input frame and acquire initial water surface geometry and volume information; the second step adopts the SPH model to simulate the fluid–solid coupling and the ICP method to correct the SPH particles using volumetric data in the next frame. Finally, the animation result is rendered. In the output animation, most of the boundary conditions (based on the input video) are preserved and new coupling behaviors are appended

into particles at the same time. The direct force method is employed to control the object and preserve the solid shape. Here the movement of the volumetric particles is restricted by the SPH model and data reconstructed from video frames. After a few iterations in the SPH system, the iterative closest point method [3, 17] (ICP for short) is adopted to match and modify the SPH particles using the volumetric data, recovered from the next frame. At the end of such processes, the marching cubes method is used to extract surface, which is rendered as visually plausible animation results eventually. In the following sections, we shall detail our system’s technical elements step by step.

4 Fluid geometry reconstruction

This section introduces our single video-based approach that acquires the height field and the velocity field, as well as real-time surface discretization. With video input, we can obtain the basic fluid structure, then add splash details or fluid–solid coupling with the basic fluid structure preserved. The velocity field satisfies the shallow water model property. The color

remapping technique is employed to illustrate the comparison between the video frame and the recovered height field.

4.1 Height field reconstruction using SFS

The first step concentrates on recovering some initial geometry information from the input video. For the input video we shall process each frame independently (where the resolution of the input video is $m \times n$). Tsai’s method [24] is a very simple and fast algorithm for computing the depth map from a single monocular image. The reflectance function for the Lambertian surface is modeled as follows:

$$E(x, y) = R(p, q) = \frac{1 + pp_s + qq_s}{\sqrt{1 + p^2 + q^2} \sqrt{1 + p_s^2 + q_s^2}}, \quad (1)$$

where $E(x, y)$ is normalized gray level at the pixel (x, y) , $p = \frac{\partial z}{\partial x}$, $q = \frac{\partial z}{\partial y}$, $p_s = \frac{\cos \tau \sin \sigma}{\cos \sigma}$, $q_s = \frac{\sin \tau \sin \sigma}{\cos \sigma}$, τ is the tilt and σ is the slant of the illuminant. The finite difference is used to compute p and q . Using the Taylor series expansion up through the first-order term, the SFS problem turns into a linear approximation [24]. After obtaining the depth $z(x, y, t)$ at each pixel in the video frame, the actual height field $h(x, y)$ is calculated as $h(x, y) = z(x, y) - z_b$, where z_b is the minimum height of all pixels. In practice, the surfaces tend to have vertical drifts that are caused by the global luminance changes in the video recording stage. The height field is recalculated to remove the effect of global luminance change: $h(x, y, t) = h(x, y, t) - \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n h(i, j, t)$. Figure 2 shows several outdoor water examples with shape from shading recovered surfaces using Tsai method. These examples are taken from the Dyntex Dataset [22] (including gentle and breaking waves, fountain and waterfall). The resolution effects the density of points in the height field (each pixel has a height value). The resolution of the videos is 352×288 and the number of surface particles in the height field is 101,376. Interpolation and resampling can be applied to change the points density in each frame if needed. The frame rate affects the correction step interval in the simulation step (detailed in Sect. 5.3).

4.2 Height field processing and color remapping

Due to the highlight, splatter or spray in the video frame, large amount of noise exists in the acquired height field, which will influence accuracy in a negative way. The statistical outlier removal method is employed to denoise and its equation is $\mu_{\text{mean}} + \varepsilon_d \sigma_d$, where μ_{mean} is the mean distance of neighbors, ε_d is the standard deviation multiplier and σ_d is the standard distance deviation. For each pixel, the mean distance to its neighbors (the adjacent pixels in the 3D space) is calculated. Octree data structure is utilized in the neighbor searching process, which is the most commonly used to partition a

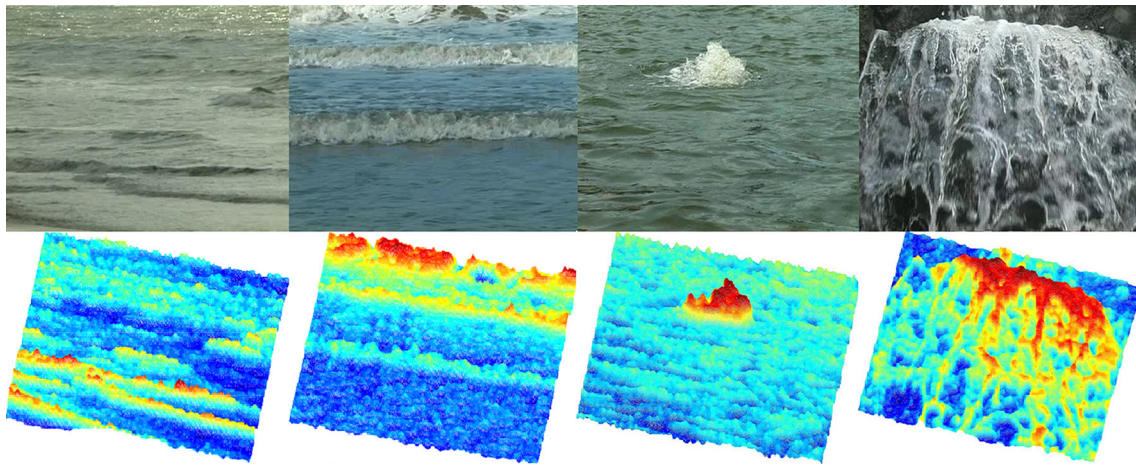


Fig. 2 Some example input videos and their SFS surface height field. The surfaces are rendered from a different view point to reveal the 3D information. The red color means the larger height value, while the blue

color means the smaller value. The sequence number of the video are 649h310, 649db10, 6486410 and 64ad110 in the Dyntex database

3D space by recursively subdividing it into eight octants. Assuming that the distribution is Gaussian with a mean and a standard deviation, all points whose mean distance is outside an interval defined by the global distances mean and standard deviation will be considered as outliers.

For each removed outlier's pixel position, its k^2 neighbors (neighbors in a square window) data are taken into account to fill these holes. Then the height field is smoothed. The last step is to smooth the result height. For each pixel, sum up its neighbors' height and divide it by k^2 and set it as its final height. Figure 3 gives an example of this process. Finally the height field of each frame is normalized by subtracting per-frame means, avoiding dithering in the result animations. In order to illustrate the accuracy of the reconstructed result, the mapping between the intensity of pixels in the input video frame and the height field is defined. The height field is turned into an RGB picture, as shown in Fig. 4.

According to RGB color information and the reconstructed height in each frame, a hash table is created and the mapping relationship between the height value and the RGB color is built. For the height field of each frame, the

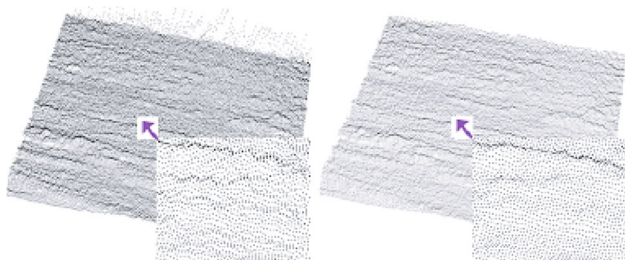


Fig. 3 After the denoising, hole filling and smoothing, we obtain better height field

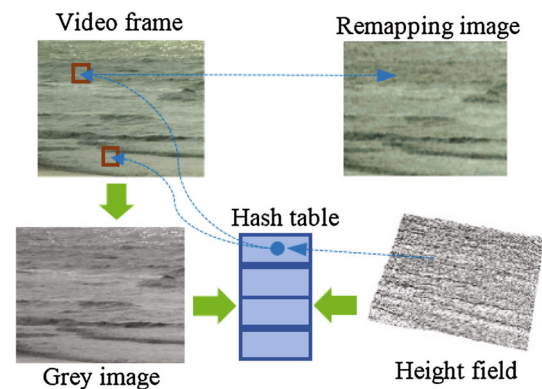


Fig. 4 The remapping process is adopted to transform the height field into an RGB image. The relationship between the RGB information and height value is created using a hash table and the height field is turned into an RGB image

maximum height is labeled as h_{max} , the minimum is h_{min} , and the grey value $G(x, y)$ for each pixel's height $h(x, y)$ in one frame is:

$$G(x, y) = \frac{(h - h_{min})(G_{max} - G_{min})}{h_{max} - h_{min}} + G_{min}, \quad (2)$$

where h denotes $h(x, y)$ in one video frame, G_{max} and G_{min} are the maximum and minimum grey values of the video frame (in YUV space), respectively. When a grey value G corresponds to multiple color values, we compute the similarity of neighbors to select the most matching color to match with the grey value. For example, some grey value G_k corresponds to multiple colors $c_1, c_2 \dots c_m$ in the hash table, the similarity is computed in Eq. 3:



Fig. 5 The comparison between video frame and the height field is shown in the remapping image. Notice the *right* remapping image approximates the *left* video frame, revealing a high accuracy

$$s_i = \sum_{j=0}^7 (c_{ir} - c_{jr})^2 + (c_{ig} - c_{jg})^2 + (c_{ib} - c_{jb})^2, \quad (3)$$

where $1 \leq i \leq m$ and j is one of the eight neighbors of pixel i and c_{ib}, c_{ig}, c_{ir} or c_{jb}, c_{jg}, c_{jr} are the RGB components. We choose c_i as G_k 's color value where s_i is the minimum, as illustrated in Fig. 4. Figure 5 shows the comparison. Our system is capable of producing visually plausible results and achieves very high percentage of accuracy.

4.3 Velocity field estimation

The 3D velocity missing from the SFS data are estimated in this subsection. SFS surfaces is considered as a prior to constrain the velocity estimation. Shallow water is a special case of water simulation that allows fast computation [26, 31] that describes the flow below a pressure surface in a fluid. In this paper, we choose shallow water to simplify the deep water model. The equations are derived from the strategy of depth-integrating the Navier–Stokes equations, in the case where the horizontal length scale is much greater than the vertical length scale. Under this condition, the conservation of mass implies that the vertical velocity of the fluid is small. The governing equation is:

$$h_t + h_x \mathbf{u} + h_y \mathbf{v} + h(\mathbf{u}_x + \mathbf{v}_y) = 0. \quad (4)$$

In this paper, $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ represents the velocity field along (x, y, z) directions, x and y axes are in the image domain and z axis is perpendicular to the x – y plane. In other words, each pixel in a frame has a height value along z axis.

As for the additional smoothness constraint [33], it ensures a smooth surface in order to stabilize the convergence to a unique solution, and is given by

$$\int \int (p_x^2 + p_y^2 + q_x^2 + q_y^2) dx dy, \quad (5)$$

where p and q are surface gradients along the x and y directions. At this point, our goal is to estimate a relatively steady

velocity field and the smoothness constraint is selected as that in [16]:

$$\int \int |\nabla \mathbf{u}|^2 + |\nabla \mathbf{v}|^2 dx dy. \quad (6)$$

The objective energy function for velocity estimation is a weighted combination of the shallow water equation and the smoothness constraint:

$$E = \int \int [(h_t + h_x \mathbf{u} + h_y \mathbf{v} + h(\mathbf{u}_x + \mathbf{v}_y))^2 + \alpha^2 (|\nabla \mathbf{u}|^2 + |\nabla \mathbf{v}|^2)] dx dy. \quad (7)$$

Here z_t can be directly estimated by subtracting the two consecutive shape from shading surfaces. Equation 4 can be minimized by solving the associated Euler–Lagrange equations. The solving process in [16] is applied, but it differs in that the result z_t in each frame remains unchanged to keep its consistency with the real data. After obtaining the velocity field, we calculate the acceleration field between two consecutive frames.

Here the height field is recovered from the input video with high accuracy and velocity field satisfies shallow water equation. However, only one layer of surface geometry is not enough to drive the SPH simulation system; it is necessary to obtain 3D volume data.

4.4 Sampling of volumetric particles

In the natural fluid scenes, the same water surface corresponds to different types of volume information, such as vortex, undercurrent, rocks and mud. To solve the non-unique volume problem, we simplify the deep water model. The surface geometry acquired from video frames is regarded as a physical guidance towards the practical goal of producing new fluid animation. A lot of efforts have been made to approximate the same appearance and the most parts in the result animation are closely similar to the input video. In addition, new coupling behaviors are appended into the result video.

We add a river bed, discretize the surface particles, and add volumetric particles in order to fill into the entire volume (the length is 352, the width is 288, and the height is max value of $h(x, y, t)$) as shown in Fig. 6. Here the river bed is defined and the volume particles are added from bottom to top. The value of the river bed on the z axis determines the thickness of the SPH water.

When the vertical internal particles' max height is less than the surface particle's height, new particles are continuously produced. The distance R between two particles controls the sampling resolution, which is also crucial to density estimation in SPH system. So the radius of particles R is computed

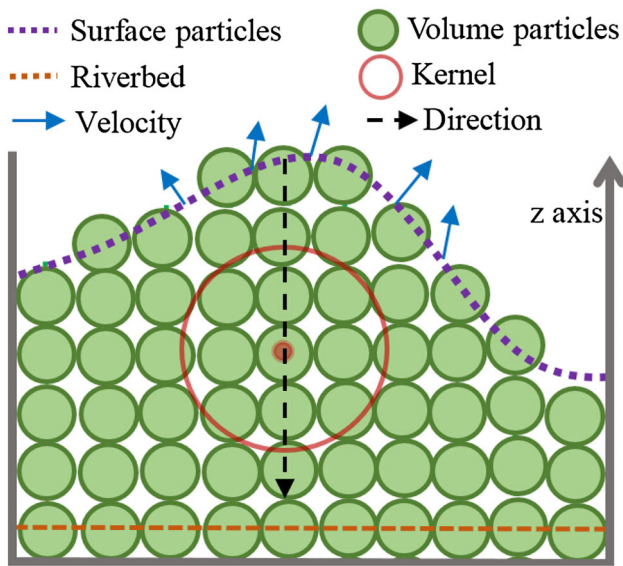


Fig. 6 This image demonstrates how the surface is discretized into the 3D volume. First, according to the surface points (purple) recovered from video, the volumetric particles are generated from bottom to top. You can consider this process as a resampling process. Then we spread the velocity field (blue arrow) from surface to the river bed (orange dotted line). Finally, a kernel function (orange circle) is applied on each volumetric particle to recalculate its velocity. At last, one layer of surface geometry is expanded to the entire volume

based on the parameters in SPH to avoid explosion. As shown in Fig. 6, the black dashed arrow indicates the velocity field decreases in proportion from top to bottom and inner particle’s velocity is recalculated and smoothed by the kernel function (in brown circle). First, the velocity and acceleration of the inner particles is initialized to zero and interpolated based on surface neighbors’ velocities. That is, the velocity field spreads from the surface to the boundary. The value of particles’ velocity (including acceleration) between the surface and the river bed decreases in proportion (from 1 to 0 gradually). Then, a kernel function (in the brown circle in Fig. 6), such as spherical kernel function:

$$W(d, h) = \begin{cases} 1 - \frac{3d}{2h} + \frac{d^3}{2h^3}, & d \leq \sigma \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

is applied to this new velocity field to enhance neighbors’ relationship (each particle’s velocity is recalculated based on its neighbors). This sampling process on the 2D planar configuration is illustrated in Fig. 7.

After volumetric sampling of the video 649h310, the number of particles jumps from 101,376 to 223,468 (with the assumption of particle radius R in the SPH system being set to 1.5). The discretization result is illustrated in Fig. 8. At this stage, we have completed our reconstruction process. In fact, water surface is moving horizontally, so particles (containing height field and velocity field) are generated from the video

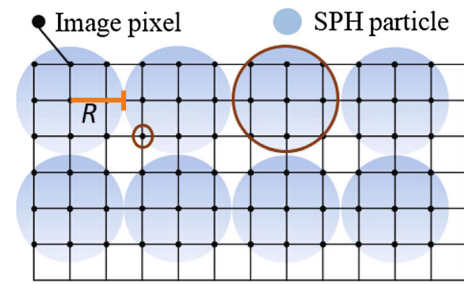


Fig. 7 In the image coordinate system, each pixel has a height value. This picture illustrates the proposed sampling process on the 2D planar configuration. Several pixels are sampled into a bigger SPH particle, including position and velocity. At last, SPH particles fill the 3D volume

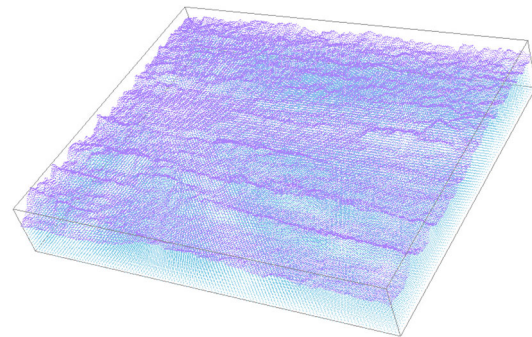


Fig. 8 This picture shows the discretization result. One layer of surface particles (purple color) is discretized into the whole 3D volume (light blue color). The value of the river bed on the z axis determines the thickness of the SPH water

frame-by-frame. It is worth noting that the surface geometry information is preserved. The volumetric particles recovered from video frames is marked as $\text{Volume}_{\text{video}}^k$, where $k \subseteq [1, \text{the number of frames}]$. The $\text{Volume}_{\text{video}}^1$ is used to initialize the SPH system.

5 Fluid–rigid coupling using SPH model

The SPH method works by dividing the fluid into a set of discrete elements, referred to as particles. The fluid particles are driven by the N-S governing equation employing the SPH model. As for solid particles, the direct force method is implemented to realize the two-way coupling. Here $\text{Volume}_{\text{video}}^1$ data are selected to initialize the SPH system and the SPH particles are divided into two categories : volumetric particles in the SPH system (marked as $\text{Volume}_{\text{sph}}^k$) and the solid particles (marked as $\text{Volume}_{\text{solid}}$).

5.1 Fluid movement with particles

The force of each fluid particle is composed of three parts: pressure, viscosity force, and external force. External force is gravity in general and pressure force is caused by pressure differences in the fluid. For instance, consider the flow of

liquid in the pipe, the pressure of the inlet area will be larger than the outlet area, so the liquid moves. The force value is equal to the pressure gradient, from areas of high pressure to low pressure areas. Viscosity force is caused by the velocity difference between particles and the value of the force is associated with fluid viscosity coefficient and the velocity difference.

The contributions of particles to one certain position are weighted according to its distance from the particle of interest. Mathematically speaking, this is governed by the kernel function W . A quantity A_i at an arbitrary position x_i is approximately computed with a set of known quantities A_j at neighboring particles position x_j :

$$A_i = \sum_j \frac{m_j}{\rho_j} A_j W(d, h), \tag{9}$$

where d equals $\|x_i - x_j\|$ and h is the kernel radius.

When searching neighbors, we adopt the uniform grid. The 3D space is subdivided into cubic cells and each particle is associated to one cell in the construction stage. To find all relevant neighbors, the cell of a particle and all adjacent cells are queried. If the cell size is equal to the kernel support, 27 cells have to be queried in 3D, which is optimal according to Ihmsen et al. [10]. The number of particles associated with a cell and the number of neighboring particles depends on the initial particle distance. It may be noted that, this query step is easy to parallelize.

When computing density, A is substituted by ρ . Adopting the formulation in [19], We choose Poly6 function

$$W_{\text{poly6}}(d, h) = \frac{315}{64\pi h^4} \begin{cases} (h^2 - d^2)^3, & 0 \leq d \leq h \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

as kernel function for density interpolation and Spiky function

$$W_{\text{spiky}}(d, h) = \frac{15}{\pi h^6} \begin{cases} (h - d)^3, & 0 \leq d \leq h \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

for pressure computation. So the final acceleration \mathbf{a}_i is formulated as follows:

$$\mathbf{a}_i = \mathbf{g} - m \frac{45}{\pi h^6} \sum_j \left(\frac{p_i + p_j}{2\rho_i\rho_j} (h - d)^2 \frac{x_i - x_j}{d} \right) + m\mu \frac{45}{\pi h^6} \sum_j \left(\frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_i\rho_j} (h - d) \right), \tag{12}$$

where ρ is the density, \mathbf{g} is the gravity acceleration, p is the pressure, μ is the coefficient of viscosity and \mathbf{v} is the velocity. At each time step we recalculate density, pressure, acceleration, velocity of each fluid particle and update its

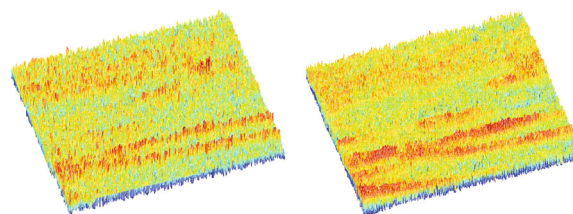


Fig. 9 After 40 iterations, the particles in the SPH system (left) and the particles loaded from the next frame (right) are shown together in the picture. SPH particles approximate the recovered data, where fluid surface geometry is preserved. In the following correction process, we use the recovered data in the next frame (right) to correct the SPH particles (left)

position. The SPH's iteration time step is set to be 1 ms and the interval time between two consecutive video frames is 40 ms (i.e., 25 fps). After every 40 iterations, the next frame particles' data are loaded to correct the current particles in the SPH simulation. Figure 9 shows the comparison of two groups of particles.

According to the input video resolution, we define the boundary to restrict the whole scene range. In order to reduce the computational burden of the collision between the liquid particles and the boundary, the particles are pushed back by the boundary and the boundary is set to be stationary. The force \mathbf{f}_b is computed as $k^s \mathbf{n} + k^d (\mathbf{v}_r \cdot \mathbf{n}) \mathbf{n}$, where k^s is the spring coefficient, k^d is the damping constant, \mathbf{n} is the collision point's normal vector, \mathbf{v}_r is the relative velocity.

5.2 Fluid–solid interaction

In order to simulate the interaction between fluid–solid, the rigid body is sampled with particles. In fluid volume, suppose the collision point (normal vector $\mathbf{n} = \frac{x_f - x_s}{\|x_f - x_s\|}$, position $x_c = x_s + \mathbf{n}R$ and velocity \mathbf{v}_c) between a fluid particle A (with position x_f , velocity \mathbf{v}_f , mass m_f) and a solid particle B (with position x_s , velocity \mathbf{v}_s , mass m_s) can be detected if the distance of two centers is smaller than $2R$. The solid centroid position x_o and velocity \mathbf{v}_o are computed by

$$x_o = \frac{1}{n} \sum_i^n x_{si}, \quad \mathbf{v}_o = \frac{1}{n} \sum_i^n \mathbf{v}_{si}. \tag{13}$$

Here x_{si} is the i th solid particle position and n is the number of particles. Then an offset vector $\mathbf{q}_o (\mathbf{q}_{oi} = x_{si} - x_o)$ is stored, representing each solid particle's relative position to the centroid. The solid's moment of inertia is computed as

$$I = m_s \sum_i^n \mathbf{q}_{oi}^T \mathbf{q}_{oi}. \tag{14}$$

The collision force between A and B is

$$\mathbf{F}_i = \frac{1}{\Delta t} \left[\left(\frac{1}{m_s} + \frac{1}{m_f} \right) \mathbf{E} + \mathbf{r}_i^T \mathbf{I}^{-1} \mathbf{r}_i \right]^{-1} \mathbf{v}_i, \tag{15}$$

where Δt is the iteration time in the SPH system, \mathbf{E} is a 3×3 identity matrix and \mathbf{r}_i is the relative position of collision position with respect to the centroid:

$$\mathbf{r}_i = \mathbf{w}^t \times (x_c - x_o), \tag{16}$$

and the relationship between \mathbf{v}_i and relative velocity \mathbf{v}_r is

$$\mathbf{v}_i = \beta[\mathbf{v}_r - (\mathbf{v}_r \cdot \mathbf{n})\mathbf{n}] - \mathbf{v}_r, \mathbf{v}_r = \mathbf{v}_f - \mathbf{v}_c. \tag{17}$$

β is a coefficient. Here $\mathbf{v}_c = \mathbf{v}_o + \mathbf{w}^t \times (x_c - x_o)$ is the velocity of collision position and \mathbf{w} is a parameter associated with solid rotation.

The velocity and position of each solid particle is recalculated as

$$\begin{aligned} \mathbf{v}_{si}^{t+1} &= \mathbf{v}_{si}^t + \frac{\mathbf{F}_i}{m_s} + \mathbf{w}^{t+1} \times \mathbf{q}_{oi} - \mathbf{w}^t \times \mathbf{q}_{oi}, \\ x_{si}^{t+1} &= x_{si}^t + \frac{\mathbf{v}_{si}^t + \mathbf{v}_{si}^{t+1}}{2} \Delta t. \end{aligned} \tag{18}$$

In each iteration, the state of every solid particle is updated independently (including both position and velocity). We recalculate x_o and \mathbf{v}_o , then \mathbf{q}_o is used to rearrange the particles and keep the solid shape unchanged. Finally, \mathbf{q}_o is updated. The solid is capable of rotating and translating. During the entire process, the solid shape is preserved. If the deformation of the solid is beyond the threshold, the solid particles will be rearranged based on the centroid parameters and variable w .

5.3 Video data correction

The brute-force change of the particle positions in the SPH system will certainly introduce discontinuity. In this step, data correction is adopted to make the SPH particles as close to the real data as possible. The frame rate of the input video affects the correction step interval in the simulation step. Here the frame rate is 25 and the time interval between two adjacent frames (transformed into volumetric particles) is 40 ms. The time step in the SPH system is Δt , which is set to be 1 ms in practice. After $40/\Delta t$ iterations, the volumetric particles $\text{Volume}_{\text{video}}^{k+1}$ in the next video frame is applied to correct the SPH particles $\text{Volume}_{\text{sph}}^k$. The volumetric particles $\text{Volume}_{\text{sph}}^k$ and solid particles $\text{Volume}_{\text{solid}}$ in the SPH system are processed separately and only the volumetric particles $\text{Volume}_{\text{sph}}^k$ are corrected in this stage. The solid particles data remains unchanged. As illustrated in Fig. 10(1), the ICP [3,17] method is employed to match two group of particles : the SPH particles $\text{Volume}_{\text{sph}}^k$ (red particles) and new volumetric particles $\text{Volume}_{\text{video}}^{k+1}$ (green particles). The algorithm iteratively revises the transformation (combination of translation and rotation) needed to minimize the distance error between two groups of point cloud. Here the point-to-plane error is adopted.

$$\begin{aligned} \mathbf{M}_{\text{opt}} &= \arg \min_{\mathbf{M}} \sum_i \left((\mathbf{M} \cdot s_i^{k+1} - d_i^k) \cdot \mathbf{n}_i \right)^2 \\ s_i^{k+1} &= (\text{Volume}_{\text{video}_{ix}}^{k+1}, \text{Volume}_{\text{video}_{iy}}^{k+1}, \text{Volume}_{\text{video}_{iz}}^{k+1}, 1)^T \\ d_i^k &= (\text{Volume}_{\text{sph}_{ix}}^k, \text{Volume}_{\text{sph}_{iy}}^k, \text{Volume}_{\text{sph}_{iz}}^k, 1)^T \end{aligned} \tag{19}$$

where \mathbf{M} and \mathbf{M}_{opt} are 4×4 3D rigid-body transformation matrices. $\mathbf{n}_i = (n_{ix}, n_{iy}, n_{iz}, 0)^T$ is the unit normal vector at $\text{Volume}_{\text{sph}_i}^k$. The relative orientation between the two

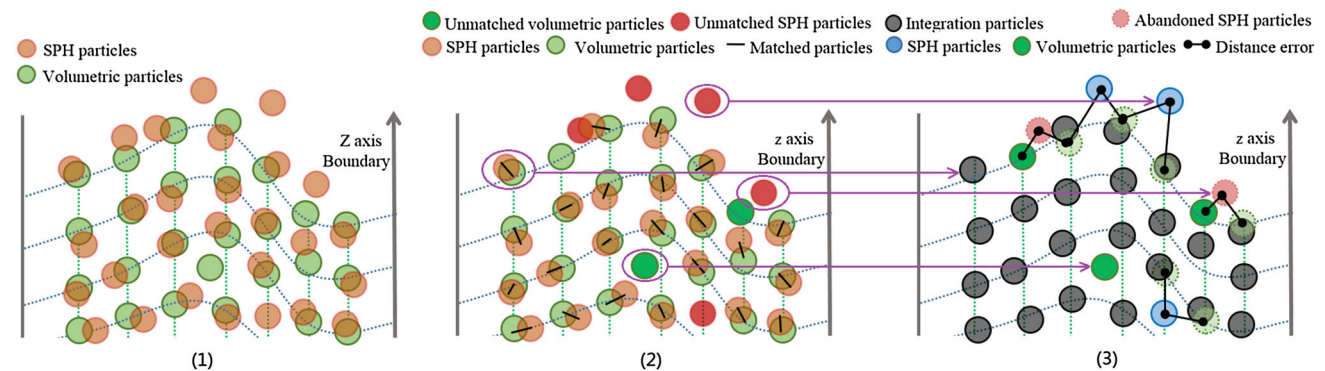


Fig. 10 In the correction process, ICP method is applied to match two group of particles, as shown in subgraph 1. Then particles are divided into three categories: matched particles, unmatched volumetric particles (dark green ones) and unmatched SPH particles (dark red particles) in subgraph 2. Each pair of matched particles are integrated into one particle (grey particles in subgraph 3). The distance error of each unpaired SPH particle (red color) to its nearest volumetric data neighbors (green

color) is calculated. We abandon unmatched SPH particles (purple) whose distance error below a threshold value. Under the same condition, unmatched volumetric particles of the next frame are preserved (dark green ones). As for these unmatched particles whose distance error above a threshold value (light blue ones), we treat them as new splatter and spray caused by interaction and retain them in the local area

group of particles is small, the nonlinear optimization problem is approximated with a linear least-squares one that can be solved more efficiently.

After ICP process, two group of particles are divided into three categories: matched particles, unmatched volumetric particles (dark green particles) and unmatched SPH particles (dark red particles), as shown in Fig. 10(2). Each pair of matched particles are integrated into one particle (grey particles in Fig. 10(3)). The average of both velocity and acceleration is calculated as new velocity and acceleration. As illustrated in Fig. 10(3), the distance error of each unpaired SPH particle (colored in red) to its nearest real data neighbors (colored in green) is calculated. As for these unmatched particles whose distance error below a threshold value, the SPH particles are abandoned (purple particles) and new volumetric particles of the next frame are selected (dark green ones). As for these unmatched particles whose distance error above a threshold value (light blue ones), we treat them as new splatter and spray caused by interaction and retain them in the local area. Finally, these retained particles $Volume_{sph}^{k+1}$ (including grey, dark green and light blue particles in Fig. 10(3) combining with solid particles $Volume_{solid}$ are imported into the SPH system again and new iterations are carried out. The fluid particles geometry information is corrected by both SPH models and real video data. Finally, new splatter and spray caused by interaction are retained in the local area and the other parts of the surface particles are correspondingly modified by the video data. The procedure of the whole process is shown in Algorithm 1.

Algorithm 1 Video-based Fluid Simulation

```

1: Initialize SPH using volumetric data  $Volume_{video}^1$ 
2: Add solid particles  $Volume_{solid}$ 
3:  $totalFrames \leftarrow N$  //N is the number of frames
4:  $\Delta t \leftarrow 1ms$  //time step in SPH system
5:  $k \leftarrow 1$ 
6:  $FPS \leftarrow 25$ 
7:  $Volume_{sph}^k = Volume_{video}^1$ 
8: while  $k \leq totalFrames$  do
9:   while  $iterations \leq (1000ms / (FPS * \Delta t))$  do
10:    SPH advection( $Volume_{sph}^k, Volume_{solid}$ )
11:   end while
12:   Apply ICP to match  $Volume_{video}^{k+1}$  and  $Volume_{sph}^k$ 
13:   Integrate matched particles
14:   Compute distance error for unmatched particles
15:    $Volume_{sph}^{k+1} \leftarrow$  retained particles
16:    $k \leftarrow k + 1$ 
17: end while

```

5.4 Surface extraction and rendering

Here the marching cubes method is adopted to obtain the iso-surface. In a grid space the cells that contain the surface are first identified. After the data correction process, the surface

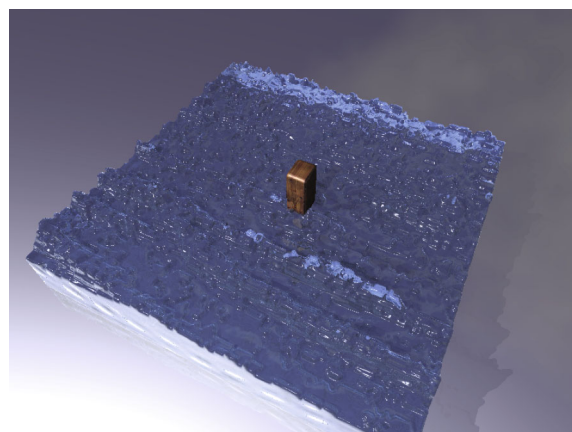


Fig. 11 Particles are extracted to form a surface using the marching cubes method and the textures between fluid and solids are different

is extracted. Due to different textures, fluid and solid particles are processed, respectively, including surface extraction and texturing. Then, as illustrated in Fig. 11, we integrate them together and render the result. Different from Kwatra's work in [13], the whole 3D simulation scene can be rendered at any angle. Finally, an angle of the camera's view is selected and the simulation results including details are rendered. We place the camera at $(176, -50, 50)$ in the 3D coordinate system, pointing to $(176, 144, 0)$.

6 Results and analysis

6.1 Experiments and evaluation

To validate the proposed method, we first choose 649h310 video in the Dyntex dataset [22] and then apply our approach to estimate the height field and compare them with the recorded video frame. In the surface reconstruction process, the P_s and Q_s in Tsai's method [24] are both set to be 0.01. The statistical outlier removal method is applied and the number of points to use for mean distance estimation is 30 and the standard deviation multiplier for the distance threshold calculation is 2. The window size for hole filling and smoothing is 3×3 . The volume particles' radius is 1.5. For example, a volume particle's position is $(x, y, h(x, y, t))$ so another adjacent volume particle's position is $(x + 3, y + 3, h(x + 3, y + 3, t))$. When estimating velocity, we set $\alpha = 1$ and $\beta = 0.5$ (in the fluid–solid interaction stage). The comparison between the video frame and reconstructed height field image is shown in Fig. 5. In the SPH simulation, the kernel h is twice the radius and Δt is 1 ms.

Due to the data independence property of the SPH system, it is suitable for GPU parallelization. Particle properties are related to its neighbors' properties. Neighbors' properties are collected to calculate the particle property, avoiding serial-

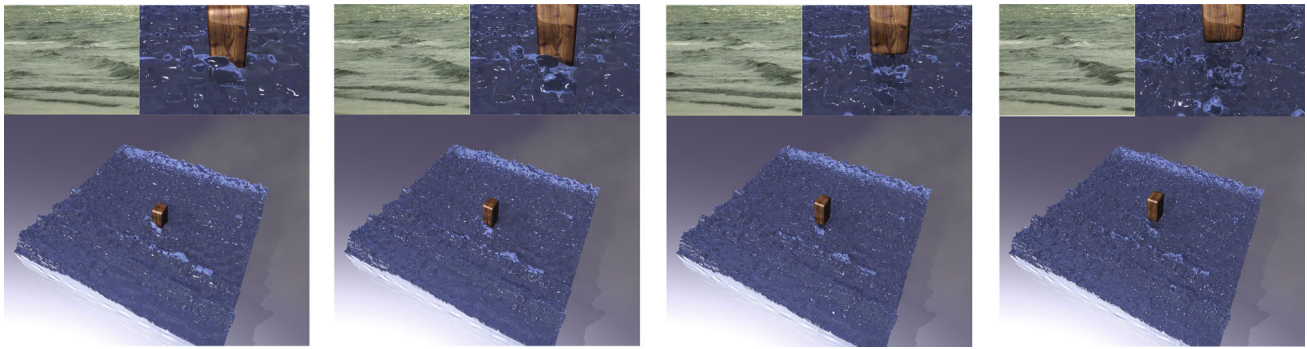


Fig. 12 Here we show the output surfaces (the *bottom row*) of four different video frames (frame 90, 98, 106, 114 at the *top left row*). The video is 649h310 in the Dyntex database. The pictures (*top right row*) illustrate splash details. Notice that even the underlying physics in

not perfectly accurate, our method is able to preserve boundary condition based on the input video and produce new visually plausible two-coupling results

Table 1 Average particles number and average time cost

Video	Reconstruction step				Simulation step		
	Surface particles	Volumetric particles	Time (ms)	FPS	SPH particles	Time (ms)	FPS
649h310	101,376	223,468	23.9	41.6	223,768	94.7	10.6
649db10	101,376	254,920	27.2	37.0	255,220	118.7	8.4
6486410	101,376	200,461	22.7	43.5	200,761	85.3	11.7
64ad110	101,376	285,813	30.1	33.3	286,113	126.7	7.9

Table 2 The comparisons of our work with alternative approaches in both the video-based reconstruction method and the physical simulation method

	Reality	Input data independence	Surface details	Volumetric data	Computation complexity	Physical interaction
Video-based reconstruction [16]	+++	---	++	---	++	---
Physical simulation [27]	++	+++	+++	+++	-	+++
Video-based fluid re-simulation [25]	+++	+	++	---	+++	--
Our hybrid method	+++	++	+++	+++	+++	+++

Here '+' denotes the higher quality and '-' denotes the lower quality

ization which is caused by write-after-write fetch operations. Moreover, the uniform grid is employed for the neighbors' search. Taking advantage of coalescing and broadcasting mechanism in CUDA, faster computation could be achieved. The coupling results are shown in Fig. 12. The similar appearance of the input video fluid scene is reproduced, furthermore, new coupling behaviors are integrated.

Table 1 demonstrates the efficiency of our method when the number of particles is huge. Due to the same resolution of the input video, the number of particles in the reconstruction step is equal for each frame. The cost times of the reconstruction step are slightly different on account of diverse

water surface in videos. Volumetric sampling process multiplies the number of particles and the ICP matching algorithm increases the complexity in SPH simulation and consumes more time.

We also provide an analysis of our method, using comparisons with alternative approaches in terms of perspectives of the video-based reconstruction, the SPH simulation and video-based fluid re-simulation, as shown in Table 2. The SPH system intends to model the equations of hydrodynamics towards the goal of producing fluid animation, which visually approximates to the real fluid behaviors. Meanwhile, the video-based reconstruction methods recover the surface

geometry from the recorded video. Higher quality of the surface details could be obtained, but the volumetric information is missing. Our approach combines the advantages of two diverse types of methods, because this new method could not only preserve the surface details, but also utilize volumetric data to realize the fluid–solid coupling in a physically plausible sense. The proposed method performs consistently well across different types of large-scale outdoor water scenes (including gentle and breaking wave, fountain and waterfall, etc.).

6.2 Limitation and application domain

In the reconstruction step, the SFS method is first applied to recover height field from the input video. Then the reconstructed height field is utilized for velocity field estimation, which is governed by shallow water equation. The reconstructed data are treated as ‘real data’ and applied to estimate volumetric data and correct the dynamics of SPH model. Obviously, the original reconstructed height field is crucial in the entire estimation and re-simulation process. So our method is inevitably subject to possible artifacts of SFS, which is not yet capable of coping with shadow and reflection of water surface and might result in possibly serious geometric distortion. Besides, in the simulation scene, fluid and solid particles are processed, respectively, including surface extraction and texturing. The resulting animations cannot be real-time rendered. The algorithm described here could be applied into video enhancement and the simulation/animation could be utilized for the rapid production of special visual effects in movies or video games.

7 Conclusion and future work

In this paper, we have presented an efficient method combining Lagrangian method (i.e., SPH) and the SFS method. SFS is employed to acquire water surface geometry from single input video. We improve the height field by applying denoising, hole-filling, and smoothing operations. When estimating the velocity field, the shallow water equation is applied as a physical model. The visual comparison with real pictures illustrates accuracy of the reconstruction process. Nonetheless, only one layer of surface model is not possible to drive the entire SPH simulation, so the surface geometry from the acquired dataset is employed to serve as boundary condition to help discretize the entire fluid volume and couple with additional (yet virtual) rigid bodies. In the SPH simulation, the direct force method is utilized to control the movement of objects. The fluid particles are constrained by both the N-S equation and the real, reconstructed data. Two-way coupling is demonstrated as experimental results. Our method has the attractive characteristics:

- The entire system’s input is only a video sequence recorded by an ordinary camera, together with (virtual) objects via synthesis. The surface is expanded into 3D volume. It works fully automatically and the fluid animation is closely approximating the input video.
- The real data and physical SPH model are tightly integrated to realize the two-way coupling. The system works consistently across a wide range of fluid movement/scene acquired by videos.
- The physical fluid–solid coupling is realized, which is guided by the reconstructed data and SPH model. We can reproduce a new fluid animation, and the new synthesis is guided by the input video while the new coupling behaviors are appended.

There are many interesting future extensions based on our current research efforts. First, The reconstructed surface geometry information is treated as real data, so the system is solely limited by the artifacts of SFS. Second, in the discretization process, we simplify the deep water model to obtain the volumetric dataset. In the near future, more complicated fluid models such as vortex or undercurrent could be added. As a possible improvement, completely different coupling effects could be achieved according to the same source of fluid video. Boundary condition handling, and more complex objects could also be considered in order to improve the entire system’s functionalities.

Acknowledgments This paper is partially supported by Natural Science Foundation of China under Grant Nos. 61532002, 61272199, National High-tech R&D Program of China (863 Program) under Grant 2015AA016404, the Specialized Research Fund for Doctoral Program of Higher Education under Grant 20130076110008, and Open Funding Project of State Key Laboratory of Virtual Reality Technology and Systems of Beihang University under Grant BUAA-VR-15KF-14. The authors would like to thank all reviewers for their very helpful and constructive comments and suggestions. We also thank Dyntex Dataset for the support of rich fluid videos for our study.

References

1. Akinci, N., Ihmsen, M., Akinci, G., Solenthaler, B., Teschner, M.: Versatile rigid-fluid coupling for incompressible sph. *ACM Trans. Graph.* **31**(4), 62:1–62:8 (2012)
2. Becker, M., Teschner, M.: Weakly compressible sph for free surface flows. In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 209–217 (2007)
3. Besl, P.J., McKay, N.D.: Method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 239–256 (1992)
4. Chao, J., Mei, R., Singh, R., Shyy, W.: A filter-based, mass-conserving lattice boltzmann method for immiscible multiphase flows. *Int. J. Numer. Methods Fluids* **66**(5), 622–647 (2011)
5. Chen, S., Doolen, G.D.: Lattice Boltzmann method for fluid flows. *Annu. Rev. Fluid Mech.* **30**(1), 329–364 (1998)
6. Chentanez, N., Müller, M.: Real-time eulerian water simulation using a restricted tall cell grid. *ACM Trans. Graph.* **30**(4), 82:1–82:10 (2011)

7. Foster, N., Metaxas, D.: Realistic animation of liquids. *Graph. Models Image Process.* **58**(5), 471–483 (1996)
8. Gregson, J., Ihrke, I., Thurey, N., Heidrich, W.: From capture to simulation-connecting forward and inverse problems in fluids. *ACM Trans. Graph.* **33**(4), 139:1–139:11 (2014)
9. He, X., Wang, H., Zhang, F., Wang, H., Wang, G., Zhou, K.: Robust simulation of sparsely sampled thin features in sph-based free surface flows. *ACM Trans. Graph.* **34**(1), 7:1–7:9 (2014)
10. Ihmsen, M., Akinci, N., Becker, M., Teschner, M.: A parallel sph implementation on multi-core cpus. *Comput. Graph. Forum* **30**(1), 99–112 (2011)
11. Ihmsen, M., Orthmann, J., Solenthaler, B., Kolb, A., Teschner, M.: Sph fluids in computer graphics. In: *Eurographics*, pp. 21–42 (2014)
12. Kutulakos, K., Steger, E.: A theory of refractive and specular 3d shape by light-path triangulation. *Int. J. Comput. Vis.* **76**(1), 13–29 (2007)
13. Kwatra, V., Mordohai, P., Narain, R., Penta, S.K., Carlson, M., Pollefeys, M., Lin, M.C.: Fluid in video: augmenting real video with simulated fluids. *Comput. Graph. Forum* **27**(2), 487–496 (2008)
14. Ladický, L., Jeong, S., Solenthaler, B., Pollefeys, M., Gross, M.: Data-driven fluid simulations using regression forests. *ACM Trans. Graph.* **34**(6), 199:1–199:9 (2015)
15. Lentine, M., Zheng, W., Fedkiw, R.: A novel algorithm for incompressible flow using only a coarse grid projection. *ACM Trans. Graph.* **29**(4), 114:1–114:9 (2010)
16. Li, C., Pickup, D., Saunders, T., Cosker, D., Marshall, D., Hall, P., Willis, P.: Water surface modeling from a single viewpoint video. *IEEE Trans. Vis. Comput. Graphics* **19**(7), 1242–1251 (2013)
17. Low, K.L.: *Linear Least-squares Optimization for Point-to-plane ICP Surface Registration*. University of North Carolina, Chapel Hill (2004)
18. Monaghan, J.J.: Smoothed particle hydrodynamics. *Annu. Rev. Astron. Astrophys.* **30**, 543–574 (1992)
19. Müller, M., Charypar, D., Gross, M.: Particle-based fluid simulation for interactive applications. In: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 154–159 (2003)
20. Ojeda, J., Susín, A.: Enhanced lattice boltzmann shallow waters for real-time fluid simulations. *Eurographics* **2013**, 25–28 (2013)
21. Pan, Z., Huang, J., Tong, Y., Bao, H.: Wake synthesis for shallow water equation. *Comput. Graph. Forum* **31**, 2029–2036 (2012)
22. Péteri, R., Fazekas, S., Huiskes, M.J.: Dyntex: a comprehensive database of dynamic textures. *Pattern Recognit. Lett.* **31**(12), 1627–1632 (2010)
23. Pickup, D., Li, C., Cosker, D., Hall, P., Willis, P.: Reconstructing mass-conserved water surfaces using shape from shading and optical flow. In: *Proceedings of the 10th Asian Conference on Computer Vision, Volume Part IV*, pp. 189–201 (2011)
24. Ping-Sing, T., Shah, M.: Shape from shading using linear approximation. *Image Vis. Comput.* **12**(8), 487–498 (1994)
25. Quan, H., Wang, C., Song, Y.: Fluid re-simulation based on physically driven model from video. *The Visual Computer*, pp. 1–14 (2015)
26. Robert Bridson, A.K.P.: *Fluid simulation for computer animation* (2008)
27. Shao, X., Zhou, Z., Magnenat-Thalmann, N., Wu, W.: Stable and fast fluid-solid coupling for incompressible sph. *Comput. Graph. Forum* **34**(1), 191–204 (2015)
28. Solenthaler, B., Miller, M., Gross, M.H., Bucher, P., Chentanez, N.: Sph based shallow water simulation. *Virtual Reality Interactions and Physical Simulations*, pp. 39–46 (2011)
29. Stam, J.: Stable fluids. In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 121–128 (1999)
30. Tan, P., Zeng, G., Wang, J., Kang, S.B., Quan, L.: Image-based tree modeling. *ACM Trans. Graph.* **26**(3) (2007)
31. Thurey, N., Müller-Fischer, M., Schirm, S., Gross, M.: Real-time breaking waves for shallow water simulations. In: *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pp. 39–46 (2007)
32. Wang, H., Liao, M., Zhang, Q., Yang, R., Turk, G.: Physically guided liquid surface modeling from videos. *ACM Trans. Graph.* **28**(3), 341–352 (2009)
33. Zhang, R., Tsai, P.S., Cryer, J.E., Shah, M.: Shape-from-shading: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(8), 690–706 (1999)
34. Zhu, B., Lu, W., Cong, M., Kim, B., Fedkiw, R.: A new grid structure for domain extension. *ACM Trans. Graph.* **32**(4), 63:1–63:12 (2013)



Chao Wang is a graduate student of Software Engineering Institute, East China Normal University, P.R. China. He received his B.E. degree in Software Engineering from Tongji University in 2009. His research interests are video-based reconstruction, fluid simulation and computer rendering. He has completed projects including 3D printer project, kinect-based point cloud reconstruction and character recognition project.



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



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

modeling, computer-aided design, computer-aided geometric design, computer animation and simulation, virtual environments and virtual engineering.



Tai-you Zhang is a graduate student of Software Engineering Institute, East China Normal University, P.R. China. He received his B.E. degree in Mathematics from East China Normal University. His research interest is fluid simulation based on physics