Li FY, Wang CB, Qin H *et al.* Augmented flow simulation based on tight coupling between video reconstruction and Eulerian models. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 33(3): 452–462 May 2018. DOI 10.1007/s11390-018-1830-7

Augmented Flow Simulation Based on Tight Coupling Between Video Reconstruction and Eulerian Models

Feng-Yu Li[1], Chang-Bo Wang[1,*], *Member*, *CCF*, Hong Qin[2], *Member*, *IEEE*, and Hong-Yan Quan[1]

[1] *School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, China*
[2] *Department of Computer Science, State University of New York at Stony Brook, NY 11794-4400, U.S.A.*

E-mail: li_fengyu@foxmail.com; cbwang@sei.ecnu.edu.cn; qin@cs.sunysb.edu; hyquan@sei.ecnu.edu.cn

Received January 8, 2018; revised March 20, 2018.

Abstract Hybrid approaches such as combining video data with pure physics-based simulation have been popular in the recent decade for computer graphics. The key motivation is to clearly retain salient advantages from both data-driven method and model-centric numerical simulation, while overcoming certain difficulties of both. The Eulerian method, which has been widely employed in flow simulation, stores variables such as velocity and density on regular Cartesian grids, thereby it could be associated with (volumetric) video data on the same domain. This paper proposes a novel method for flow simulation, which is tightly coupling video-based reconstruction with physically-based simulation and making use of meaningful physical attributes during re-simulation. First, we reconstruct the density field from a single-view video. Second, we estimate the velocity field using the reconstructed density field as prior. In the iterative process, the pressure projection can be treated as a physical constraint and the results of each step are corrected by obtained velocity field in the Eulerian framework. Third, we use the reconstructed density field and velocity field to guide the Eulerian simulation with anticipated new results. Through the guidance of video data, we can produce new flows that closely match with the real scene exhibited in data acquisition. Moreover, in the multigrid Eulerian simulation, we can generate new visual effects which cannot be created from raw video acquisition, with a goal of easily producing many more visually interesting results and respecting true physical attributes at the same time. We demonstrate salient advantages of our hybrid method with a variety of animation examples.

Keywords video reconstruction, velocity estimation, fluid simulation, volume modeling and re-simulation

1 Introduction

Flow/fluid simulation has received considerable attention in recent decades in computer graphics, thanks to its ubiquitous and powerful capability towards modeling a wide variety of natural phenomena with details such as vortex and splash. The model-centric numerical simulation is necessary in flow simulation, and the research on this subject has resulted in widespread applications, including special effects in movies and realistic environment in computer games. Especially in the most recent years, the uprising trend of virtual reality technologies in our everyday life gives a strong prospect for the broader application of flow/fluid simulation. Physically-based flow simulation mainly has offered two types of approaches: the Lagrangian[1] method and the Eulerian method[2]. Both approaches have their advantages and drawbacks. Since it is convenient to numerically approximate spatial derivatives on a fixed grid, the Eulerian method receives a lot of researchers' attention[3-5] with increasing popularity. In our paper, we mainly consider the Eulerian method. However, in order to obtain better results, the time consumption is enormous as resolution increases. Moreover, it is some-

Regular Paper

Special Section of CVM 2018

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61532002, 61672237, 61672077 and 61672149, the Natural Science Foundation of USA under Grant Nos. IIS-1715985, IIS-0949467, IIS-1047715, and IIS-1049448, and the National High Technology Research and Development 863 Program of China under Grant No. 2015AA016404.

[*] Corresponding Author

times difficult to achieve a visual effect of a particular scene, due to the requirement for appropriate initial values and strict boundary conditions.

In contrast, as the inverse problem of flow simulation, the purpose of flow capture is to measure the flow state and its physical properties in the real world, such as velocity and density. With the rapid development of data acquisition hardware, it is convenient to obtain a sequence of flow in video data. However, some details of the flow may be lost due to the equipment and environmental constraints. Moreover, it is hard to obtain a particular phenomenon with varying boundaries and/or initial value conditions. Despite this, we can obtain a rough volume of the flow and its plausible density field and velocity field, which are what we wish to utilize towards possibly enhancing the physical simulation with correct and meaningful physical quantities.

The key motivation of this paper is to clearly retain prominent advantages from both video-based method and model-centric numerical simulation, while overcoming certain difficulties of both. With the guidance of the video data, including density and velocity, the simulation results shall match with the real scene. In addition, we can inject the lost yet interesting details and change the boundary condition through physical approaches, which are difficult in video reconstruction, with a goal of creating many more vivid effects. Since the Eulerian method relies on regular grids, it is a natural choice for us to explore the Eulerian method, which could be associated with volumetric video data on the same domain. Our goal is to seek realistic simulation results that are matching with natural scenes.

In the past few years, researchers have explored the connection of flow simulation and the inverse problems, providing a few novel ideas. Okabe *et al.*[6] used the modeled volume from videos as a guide and Gregson *et al.*[7] used the reconstructed velocity field to restart fluid simulations. As a follow-up, Zhai *et al.*[8] further estimated the physical parameters of the fluid and applied physical editing on existing flows. Nonetheless, the above methods lack a tight coupling. Quan *et al.*[9] combined image analysis and physical method to reconstruct fluid in real time but lacking follow-up interactions. Combining with multi-fluid simulation, Ren *et al.*[10] modeled real-world bubble phenomena. In addition, Wang *et al.*[11] combined video-based fluid surface reconstruction and SPH (smoothed particle hydrodynamics) simulation to re-animate fluid surface. But the connection between video reconstruction and SPH is not natural. Inspired by the prior research work, we wish to use the reconstructed velocity field and density field to guide the physical simulation with stronger connection and tighter coupling.

We propose a novel method for flow simulation that is tightly coupling physical simulation with fluid capture and making use of meaningful physical attributes during re-simulation. Given a single-view video, we reconstruct the density field first. Then we estimate the velocity field using the reconstructed density field as prior. During iterations, the pressure projection can be treated as a physical constraint and the results of each step are corrected by obtained velocity field in the Eulerian framework. Finally, we use the reconstructed density field and velocity field to guide the Eulerian simulation with anticipated new results. Within the framework, we conduct experiments for certain graphics applications, such as detail enhancement and solid-fluid coupling under different boundaries and/or initial value conditions. The contributions of this paper are as follows.

• *Fluid Density Field Reconstruction from a Single Video.* We make a reasonable assumption to reconstruct the approximate density field and obtain plausible volume from a single-view video.

• *Tight Coupling Between Video Reconstruction and Eulerian Models.* In our method, pressure projection can be treated as a physical constraint of the reconstruction of velocity fields, and the reconstruction results of each step are corrected by physical simulation. Then the final reconstruction results can be used to guide the simulation. In such a way, we are capable of coupling the two tightly towards hybrid modeling over pure numerical simulation and video data acquisition.

• *Augmented Flow Simulation with Controllable Details.* Matching with the real scene, we can also add a lot of interesting details, such as complicated solid-liquid coupling subject to new boundaries and/or initial value conditions.

In the following sections, we briefly discuss some related work. Section 3 gives an overview of our approach. Section 4 discusses the density field reconstruction, and Section 5 details the coupling of multigrid Eulerian simulation and reconstruction. We discuss some controllable details and demonstrate graphics results of our work in Section 6, and draw the conclusion and outline future work in Section 7.

2 Related Work

Our approach is mainly relevant to fluid simulation, volume modeling, and velocity estimation. We introduce them in the subsequent parts.

- Fluid simulation aims to obtain the states of the fluid by solving the complex incompressible Navier-Stokes equation governing the motion. There have been lots of fantastic studies. Foster and Metaxas[12] introduced fluid simulation for computer graphics application. The SPH method[13] and its improvements[14-17], vortex methods[18], and position-based fluid[19] are exciting researches using the Lagrangian method. Since it is convenient to numerically approximate spatial derivatives on a fixed mesh, the Eulerian method also attracts much attention. Bridson's book[2] provides a brilliant overview. Stam[20] proposed an unconditionally stable model which can obtain complex fluid phenomena, leading to a large number of applications of fluid simulation. Zhu and Bridson[21] proposed a hybrid particle-in-cell solver to get a more accurate solution. Zhang *et al.*[4] proposed an IVOCK (integrated vorticity of convective kinematics) method which cheaply captures much of what is lost in self-advection to enhance the details. Focusing on solid-liquid interaction, researchers have proposed a lot of interesting algorithms. Batty *et al.*[22] presented a tricky variational approach that obtains reliable and accurate solutions, focusing on the classical pressure projection step and employing the idea of energy. Teng *et al.*[5] proposed a method to achieve a two-way coupling in which the underlying geometric representation is exactly Eulerian. In this paper, we use the Eulerian method to establish the connection with the inverse problem.

Acceleration of the solver is critical in fluid simulation, especially in the applications which demand high efficiency such as computer games. Many methods were proposed to speed up the algorithm. Losasso and Fedkiw[23] presented a method for simulating fluid on an unrestricted octree data structure and a technique for discretizing the Poisson equation on octree grid. Chentanez and Mueller-Fischer[24] presented a multigrid method for pressure projection and McAdams *et al.*[25] employed a multigrid cycle as a preconditioner for the conjugate gradient method. Hardware-based acceleration is also very attractive. Chentanez[26] performed real-time Eulerian simulation on a GPU optimizing the grid representation. The proposed solutions by Ihmsen *et al.*[27] focused on systems with multiple CPUs. In this paper, we adopt the multigrid method and parallel method to accelerate the whole process.

- Volume modeling aims at the measurement of the fluid density field through a series of fluid image sequences. Ihrke and Magnor[28] reconstructed a volumetric model from a number of pictures of flame through a tomographic method. Atcheson *et al.*[29] captured full 3D, non-stationary fluid on a Cartesian grid using a Schlieren tomography system. Gregson *et al.*[30] presented a method for tomographic reconstruction of 3D volumes by a new stochastic tomographic reconstruction algorithm based on random walks. However, these methods involve sophisticated hardware setups. Some algorithms were proposed to solve this problem. Liu *et al.*[31] proposed an approach of modeling of smoke from a single view, which does not model volumetric smoke. Stephan *et al.*[32] proposed an algorithm for the single-view modeling of approximately symmetric volumetric phenomena. Okabe *et al.*[6] proposed an approach of 3D modeling of fluid phenomena formulating the problem as an energy minimization problem. The method just needs sparse multi-view images, such as only a single-view input. In this paper, we mainly consider single-view videos, which do not require sophisticated equipment. Although it is impossible to reconstruct the volume from a single-view video precisely, we can get the rough shape of the fluid and plausible density field. As a guide of subsequent fluid simulation, the method can be applied in many scenarios.

- Velocity estimation for 3D fluid flows has been a challenging problem. Particle Image Velocimetry (PIV) was used in the early work [33]. However, the estimated fields are too coarse for the application. Liu and Shen[34] explored the connection between fluid flow and optical flow. The method provides a rational foundation for the application of optical flow method to the estimation of fluid velocity based on images. Kadri-Harouna *et al.*[35] estimated fluid velocity based on divergence-free wavelet in the incompressible constraint and a biorthogonal wavelet expansion of optical flow. Zuo and Qi[36] proposed a spatial-temporal optical flow method to estimate the velocity fields of fluid flows, taking physical principles into account. However, most of these methods are only applicable for 2D flows.

Gregson *et al.*[7] focused on tracking 3D fluid velocities based on density fields modeled by prior fluid imaging methods. They interpreted the pressure projection solver as a proximal operator. Inspired by this, we not only guide the physical simulation by reconstructed data from the video but also impact velocity estimation through the physical process. In such a way, we are capable of coupling the pure physical simulation with video data acquisition tightly.

3 Overview of Our Method

The input of our approach is a single video of fluid, which contains much valuable information, such as density and velocity. The information is stored in pixels constructed by grids, just like voxels in two dimensions. Therefore we connect it with the Eulerian method, which divides the space into a lot of small grids, just like pixels in three dimensions. Thereby, the reconstruction problem could be associated with physically-based flow simulation naturally. A pipeline of our system can be found in Fig.1.

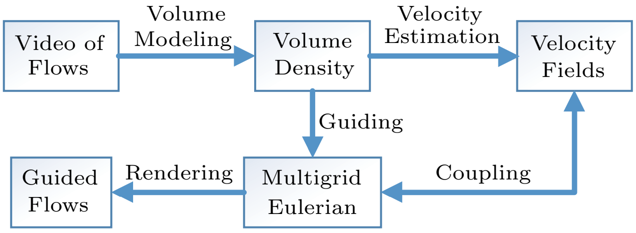


Fig.1. Pipeline of our method.

First, we model the 3D density field of the fluid from the input video and store it in voxels. Although there are many methods to reconstruct the density field, a similar system in [6] which can model volumetric fluid phenomena from only a single-view input is adopted in our framework.

Then we use the reconstructed density fields as prior to estimate physically plausible 3D dense velocity fields. The state-of-the-art fluid capturing process is from [7], connecting forward and inverse problems in fluids. In this manner, the pressure projection term in the Eulerian method is regarded as a proximal operator. In our method, the results of each step are corrected by physical simulation to improve the accuracy of matching physical property. In the following simulation, the estimated velocity fields are used to guide the physical results. Thus we can achieve a tight coupling of physical model and reconstructed data.

The following step is a physical-based flow simulation. Due to the reasons mentioned above, the Eulerian method fits our framework. The velocity fields and density fields recovered from the first frame image of the video are regarded as initial settings. In the evolution, the velocity fields guided by the reconstructed velocity fields are regarded as the last results and advect the density. The density fields are also guided by that reconstructed from the video. Then we can generate sequences that match the captured flows with simulation. We can also make use of this method to generate new visual effects, such as details enhancement and augmented fluid-rigid coupling. In every step, there is one time-consuming part. In order to improve the efficiency, we adopt a general multigrid method and implement the algorithm in parallel.

4 Density Field Reconstruction

The density field reconstruction is essentially volume modeling, which can be regarded as a minimization problem, formulated as follows:

$$\boldsymbol{d} = \mathtt{arg}\ \underset{\boldsymbol{d}}{\mathtt{min}} |\boldsymbol{Bd} - \boldsymbol{p}|^2,$$

where matrix $\boldsymbol{d}$ represents a 3D volume that belongs to the flow space, $\boldsymbol{B}$ represents the volume rendering operation under the space, and $\boldsymbol{p}$ represents the input image. Using the emission model, each voxel can be seen as a source of emission and $\boldsymbol{B}$ describes the weight of each voxel along the ray. In order to make the system as simple as possible, we omit the nonlinear optical effects, such as scattering, and form a linear operation $\boldsymbol{Bd}$, which renders the image of the 3D volume.

The entire system can be seen in Fig.2, which is similar to [6]. First, the least squares method (LSM) is used to solve the problem and we transform it into a linear system described as

$$\boldsymbol{Bd} = \boldsymbol{p}.$$

To solve the linear system, we would like to generate the sparse matrix $\boldsymbol{B}$ by choosing basis functions with local support. To simplify matters, we use the box basis function and arrive at

$$B_{ij} = \|x_1 - x_2\|,$$

where x_1 and x_2 are the points of intersection of the i-th casting ray and the j-th voxel respectively, and B_{ij} is the corresponding entry that is the distance between the intersections of the ray and the corresponding voxel. After getting the matrix, we compute a least squares solution:

$$\boldsymbol{d} = (\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{p}.$$

The conjugate gradient method is a good choice to solve this equation because of the following good properties: low cost, simplicity of coding, and parallel implementation. Since we have only one direction of the video information, in order to recover the 3D density field, we assume that there is the same image in the

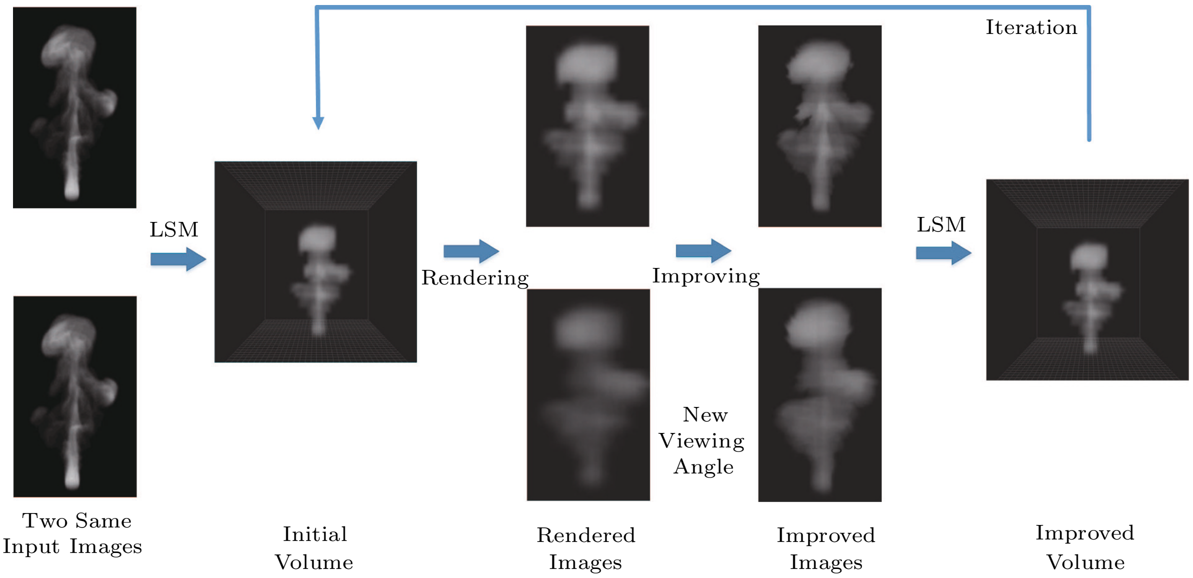


Fig.2. System of density field reconstruction.

orthogonal direction and get an approximate symmetrical result. Although it is not very accurate, for many situations, we can get reasonable results.

The pixel number of each frame in the video is fixed. However, we want to get the density fields in different scales. To solve this question, we conduct a sampling operation at first and then have a unified framework. Initial volume has obvious grid artifacts. Especially, the rendered image is very blurry when we take a new viewing angle. Then we choose the expectation maximization (EM) like iterative approaches to improve the appearance of the volume. During each iteration, we alternately minimize the energy fixing $\boldsymbol{p}$ and $\boldsymbol{d}$.

First, the volume is rendered by a simple matrix multiplication $\boldsymbol{p} = \boldsymbol{Bd}$ with the original and several new viewing angles. At this time, the images are blurry and unnatural. Therefore the next operation is to improve the images to eliminate the artifacts, which includes preprocessing and histogram-matching operation. The preprocessing can be divided into three steps. In the first step, we apply a morphological dilation operation to the image. In the second step, a Gaussian blur is applied to eliminate the artifacts. And in the third step, we apply a morphological erosion operation which is opposite to the first step. After preprocessing, the grid artifact and some other interference in the image can be eliminated. One problem is that the image becomes blurred. To solve the problem, we apply histogram-matching operation to modify the rendered images to have the same histogram as the input image.

Second, we model the volume from all the improved images using LSM again. The input at this time includes not only the original view, but also the rendered new views after improvement. When the system converges or the number of iterations reaches the maximum (in our work the maximum is set to 10), the iteration stops.

We can get a smooth result with only a single-view video and the blur operation in improving the image. Actually, it is impossible to reconstruct the density field precisely in such a situation. However, it can be used to guide the result of Eulerian simulation. It can also be treated as prior of the following velocity estimation. In addition, we can recover the lost details and generate new visual effects in the simulation. The density field is stored in discrete grids, which could be easily accessible by the Eulerian method.

5 Coupling of Simulation and Reconstruction

5.1 Multigrid Eulerian Simulations

In the Eulerian simulations, a time splitting method is usually adopted to solve the incompressible Euler equation:

$$\begin{aligned}&\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \boldsymbol{f},\\&\nabla\cdot\boldsymbol{u} = 0,\end{aligned}$$

where $\boldsymbol{u}$ denotes the velocity of the fluid, ρ is the density, p is the pressure, and $\boldsymbol{f}$ is a body force.

Fluid states are updated from an initial state $\boldsymbol{u}_0$ by marching through time with a time step Δt. First, we solve the momentum equation to obtain an intermediate velocity field $\boldsymbol{u}^*$ resulting from self-advection and body force $\boldsymbol{f}$ such as gravity and buoyancy:

$$\boldsymbol{u}^{**} = \texttt{advect}(\boldsymbol{u}^n, \Delta t),$$
$$\boldsymbol{u}^* = \boldsymbol{u}^{**} + \Delta t \boldsymbol{f}.$$

In the first step, advection is very important. Semi-Lagrangian advection[20] is a mature method to solve it. Other methods are also available, such as BFECC[37] and FLIP model[21]. Then we project it to be divergence-free by subtracting ∇p, and p is derived by a Poisson solver from $\boldsymbol{u}^*$:

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \Delta t \frac{1}{\rho} \nabla p,$$
$$\nabla^2 p = \rho \nabla \cdot \boldsymbol{u}^* / \Delta t,$$

where ∇^2 is the Laplacian operator.

After getting the divergence-free velocity field, scalar field ϕ such as density is advected using

$$\frac{\partial \phi}{\partial t} = -\boldsymbol{u} \cdot \nabla \phi. \tag{1}$$

Moreover, in one typical step, the most time-consuming part is the updating of pressure p. Incomplete Cholesky preconditioned conjugate gradient has been widely used to solve this linear system, and it is effective for moderate grid resolutions. However, its performance deteriorates as grid resolution increases. For the projection step, we apply a multigrid method because of advantages including less storage, more convergence, and more parallelization.

A multigrid solver operates on hierarchical grids from fine to coarse. On each level, we have to solve a linear system of $A^l p^l = b^l$. To downsample from coarse to fine, a standard 8-to-1 average can be used. For the restriction and the prolongation operator, tri-linear interpolation is a good choice. For smoothing operator, the Red-Black Gauss-Seidel (RBGS) method is used.

5.2 Velocity Estimation and Coupling

We can estimate the 3D velocity fields using the density fields obtained in Section 4 as prior. The reconstruction of velocity can be seen as a minimization problem similar to the optical flow:

$$E(\boldsymbol{u}) = E_{\mathrm{T}}(\boldsymbol{u}) + \alpha E_{\mathrm{SM}}(\boldsymbol{u}) + \beta E_{\mathrm{KE}}(\boldsymbol{u}) + E_{\mathrm{DF}}(\boldsymbol{u}),$$

where $E_{\mathrm{T}}(\boldsymbol{u})$ is the fluid transport term, $E_{\mathrm{SM}}(\boldsymbol{u})$ is a smoothness term, $E_{\mathrm{KE}}(\boldsymbol{u})$ is a kinetic energy penalty, and $E_{\mathrm{DF}}(\boldsymbol{u})$ is a physical constraint to ensure the velocity divergence-free. The optimization parameters α and β can be adjusted (we use $\alpha = 1\mathrm{e}{-4}, \beta = 1.0$).

Gregson *et al.*[7] found that transport equation is equivalent to the brightness constancy of optical flow, which is considered as a data item. According to (1), the transport term is defined as:

$$E_{\mathrm{T}}(\boldsymbol{u}) = \int_{\Omega} (\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi)^2 \mathtt{d}\Omega,$$

where ϕ is the reconstructed density at each voxel and $\nabla \phi$ is its spatial gradient. The equation compares the change in density due to advection to the change in captured density, which is actually a fitting term.

In optical flow, since the first term only is ill-posed, a smooth prior is usually added. As a 3D extension of the prior based on the gradient of the flow, which is used by Horn and Schunck[38], the smoothness term is defined as:

$$E_{\mathrm{SM}}(\boldsymbol{u}) = \int_{\Omega} |\nabla u_i|^2 + |\nabla u_j|^2 + |\nabla u_k|^2 \mathtt{d}\Omega,$$

where u_i, u_j and u_k are the components of velocity in each direction.

To avoid the smoothest solution and spurious velocities, a kinetic energy penalty is important:

$$E_{\mathrm{KE}}(\boldsymbol{u}) = \boldsymbol{u}^2.$$

To solve this problem, we divide it into two parts and define $F(\boldsymbol{u}) = E_{\mathrm{T}}(\boldsymbol{u}) + \alpha E_{\mathrm{SM}}(\boldsymbol{u}) + \beta E_{\mathrm{KE}}(\boldsymbol{u})$ and $G(\boldsymbol{u}) = E_{\mathrm{DF}}(\boldsymbol{u})$. Given the proximal operators, we choose the alternating direction method of multipliers (ADMM)[39] method which is most popular and easy to implement. Algorithm 1 summarizes the variational ADMM method in our work. In the algorithm, k is the iteration index, $\boldsymbol{v}$ is a slack variable, and $\boldsymbol{q}$ is the Lagrange multiplier. We initialize $\boldsymbol{v}$ and $\boldsymbol{q}$ to zero. In line 6, $\boldsymbol{u}^{n+1}$ denotes the velocity solved by the Eulerian method, which is used to correct the result, and `Ga()` denotes a 3D Gaussian blur operation. In this process, $prox_{\lambda F}$ and $prox_{\lambda G}$ denote the proximal operators for F and G, which are defined by

$$prox_{\lambda f}(\boldsymbol{x}) = \mathop{\mathtt{arg\,min}}_{\boldsymbol{w}} f(\boldsymbol{w}) + \frac{1}{2\lambda} \|\boldsymbol{w} - \boldsymbol{x}\|_2^2,$$

where f is a function, which can be F or G, and $\boldsymbol{w}$ represents $\boldsymbol{u}$ or $\boldsymbol{v}$.

For the first subproblem ($prox_{\lambda F}$), after discretization, we turn it to a simple linear least-squares problem. Conjugate gradient is a feasible solver for this linear system. Also, this process is very time-consuming, thereby we adopt a multigrid method to achieve it. For the second subproblem ($prox_{\lambda G}$), it is the pressure projection exactly. We solve it in the Eulerian simulation.

Algorithm 1 . Variational ADMM Method

```
1: procedure ADMM($prox_{\lambda F}$, $prox_{\lambda G}$)
2: while $k < maxIters$ do
3:     $\boldsymbol{u}^{k+1} = prox_{\lambda F}(\boldsymbol{v}^k - \boldsymbol{q}^k)$
4:     $\boldsymbol{v}_P^{k+1} = prox_{\lambda G}(\boldsymbol{u}^{k+1} + \boldsymbol{q}^k)$
5:     $\boldsymbol{v}_r^{k+1} = \boldsymbol{v}_P^{k+1} - \mathtt{Ga}(\boldsymbol{v}_P^{k+1})$
6:     $\boldsymbol{v}^{k+1} = \boldsymbol{v}_r^{k+1} + \mathtt{Ga}(\boldsymbol{u}^{n+1})$
7:     $\boldsymbol{q}^{k+1} = \boldsymbol{q}^k + \boldsymbol{u}^{k+1} - \boldsymbol{v}^{k+1}$
8: end while
9: return $\boldsymbol{u}^k$
10: end procedure
```

Since our input video has only one view, the reconstructed density field will be roughly symmetrical and smooth with the necessary assumption and blur operation to eliminate artifact. Especially in the interior of the flow space, the error will cause little change in the density field between the adjacent frames, and the velocity information of this part may be lost. Therefore, in line 5 and line 6 of the algorithm, we add a physical correction. Through the Gaussian blur operation, the detail components can be extracted and the lost information can be filled with physical results.

The entire interaction process of velocity can be seen in Fig.3. The projection is just the proximal operator for $G(\boldsymbol{u})$ as one part of velocity reconstruction. At the end of each iteration, the result is corrected by physical simulation. In the process, we set a threshold as the termination criterion. If the average variation between two iterations is less than the threshold or the number of iterations is larger than the maximum (here we set the maximum to 5), the iteration terminates.

After recovering the velocity field, it can be used to guide the velocity field in the Eulerian simulation using the same approach in line 5 and line 6 of Algorithm 1:

$$\boldsymbol{r} = \boldsymbol{u}^n - \mathtt{Ga}(\boldsymbol{u}^n),$$
$$\boldsymbol{u}^n = \boldsymbol{r} + \mathtt{Ga}(\boldsymbol{u}^k),$$

where $\boldsymbol{u}^k$ is the result in the ADMM iteration, $\boldsymbol{u}^n$ is the velocity in the Eulerian simulation, and residual $\boldsymbol{r}$ stores the extracted high frequency components. Through this operation, the information of video and physical simulation can be retained.

In the Eulerian simulation, a new density field is obtained by the advection. We can also use the density field modeling in the previous part to correct it:

$$\boldsymbol{s} = \boldsymbol{s} \cdot \eta + \boldsymbol{s}^r \cdot (1 - \eta),$$

where $\boldsymbol{s}$ is the density field in the Eulerian simulation, $\boldsymbol{s}^r$ is the density field recovered from the video, and η is a weight denoting how much the density is dominated by the simulation (here $\eta = 0.8$). Through coupling of data from video and physical model, the results not only match with the real scene exhibited in data acquisition, but also have new visual effects which cannot be created from raw video acquisition at the same time. Based on this, we can have many applications.

6 Controllable Details and Experimental Results

In this section, we discuss several controllable details and show the graphics results of our approach, with the characteristic of maintaining the information from both video data and physical model.

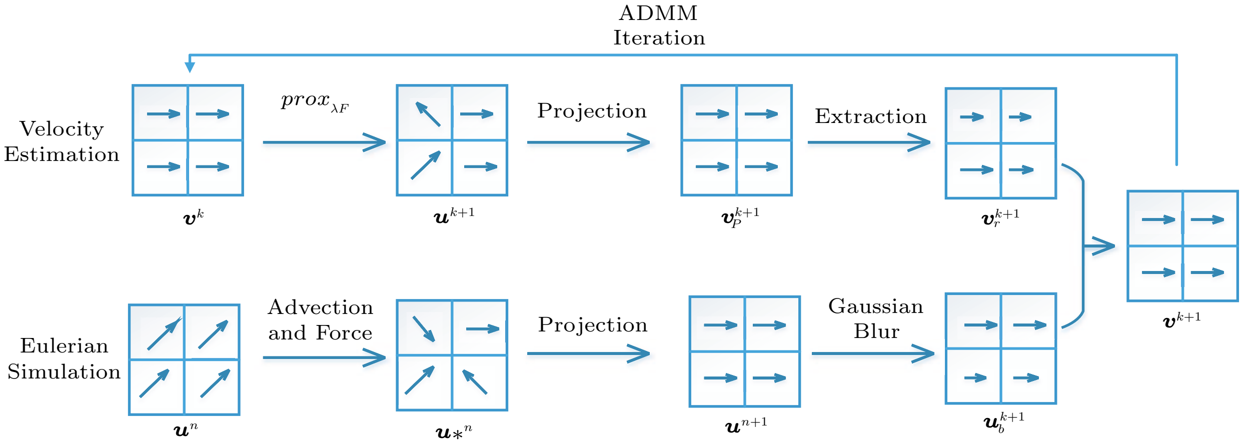


Fig.3. Details of coupling.

Considering the input video, tracking the fluid phenomenon with the camera is a great way. The captured data from Gregson *et al.*[30] is representative. We choose the "captured smoke" dataset as the input of our experiments. However, this approach also requires hardware setups to eliminate the unwanted effects of the real world. Moreover, the subsequent processing such as camera calibration is complex. To avoid these problems, we also generate realistic fluid videos instead of real acquisition using Maya2015 with FumeFX as the input of more experiments. We implement our algorithm on a PC with an Intel® Core™ i5-4570 CPU (3.20 GHz), 8.0 GB of memory, and a NVIDIA GeForce GTX 750 Ti GPU.

Given a flow video, we first recover the density field and model the volume. Fig.4 shows the results of different methods. We can find that the volume of the smoke has been recovered compared with the ground truth in Fig.4(a). Compared with LSM, the iterative method has a better appearance. The size of the volume is 128 × 256 × 128 voxels, and the average time consumption per frame is 24.6 seconds. As a priori, this part can be pre-processed. In addition, since temporal coherence is not taken into account, frames can be processed independently. Fig.5 shows one slice of velocity fields visualized by the color map. The result of the original ADMM method has a rough shape. However, some information is missing in the middle of the volume. By comparison, our variational ADMM has a more accurate result. On the resolution of $64\times128\times64$, the average time consumption is 44.7 seconds. Then after the coupling of reconstruction data and physical simulation, we obtain the animation matching realistic scene, which can be seen in Fig.6(d). Compared with the results of pure physical simulation in Fig.6(c), our results are more realistic, and match with the ground truth.

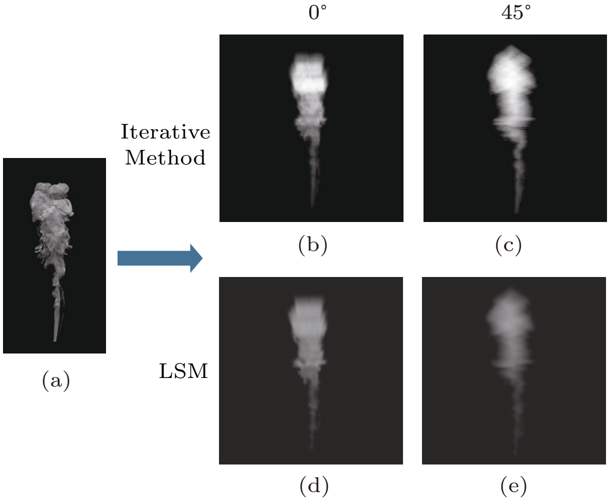


Fig.4. Reconstructed density field of "captured smoke". The input is the 70th frame of the dataset. Fig.4(d) and Fig.4(e) are the results of LSM. (b) (c) Results of iterative method. (b) (d) Results rendered in original viewing angle. (c) (e) Results rendered in a new viewing angle (45°).

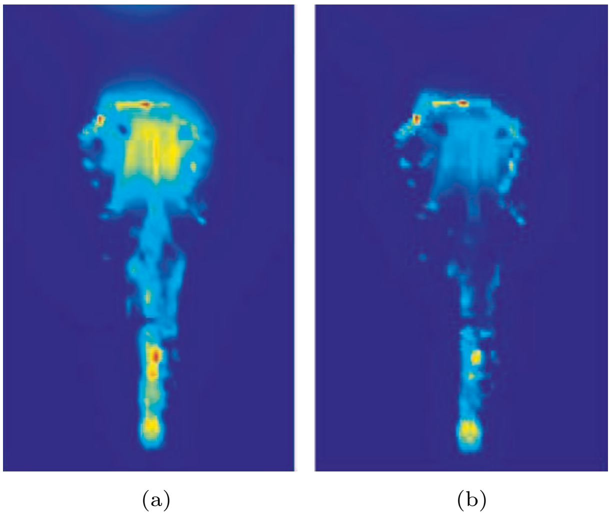


Fig.5. Central slice of reconstructed velocity fields visualized by color map. (a) Our variational ADMM. (b) Original ADMM.

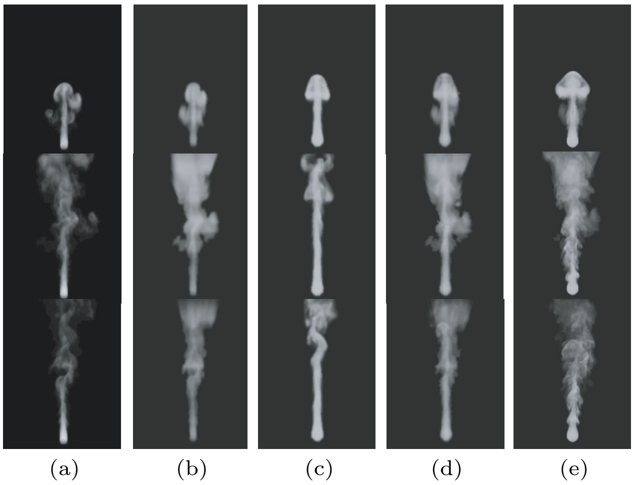


Fig.6. Results of smoke animation in five cases. (a) Input ground truth of "generated smoke". (b) Volume modeling. (c) Pure physical simulation. (d) Our method without additional details. (e) Our method with additional details. From top to bottom: frame 50, frame 100 and frame 150.

6.1 Details Enhancement

Because of the limitation of the devices, the input video may be blurred, losing a lot of interesting features, such as whirlpool. Therefore in the Eulerian simulation, we wish to restore the missing information. We mainly have two approaches to achieve it.

First, we simply increase the resolution of the grids. With the growth of resolution, the numerical calculation result will be more accurate and some details will appear. However, the time consumption will be incredible when the grids are very dense. Second, some physical schemes of recovering details, such as vorticity confinement, can be utilized. IVOCK[4] is the modern method to enhance details and we add it to our system.

In Fig.6(e), we can see the smoke pictures which have added the details. The results have much improvement compared with Fig.6(d), without additional details.

6.2 Solid-Flow Coupling

The solid-flow coupling is a very common phenomenon attracting much attention. With the input flows, we want to get the phenomenon of coupling with the solid in similar scenes.

In order to achieve this effect, we make some changes to our approach. In the Eulerian simulation, we enforce the solid boundary conditions via the variational approach[22] in the pressure solving phase. During velocity estimation, we replace the kinetic energy penalty term $E_{\rm KE}$ with the total kinetic energy of the system:

$$E_{\rm KE} = \int_{\Omega} \rho \|\boldsymbol{u}\|^2 + \boldsymbol{V}^* \boldsymbol{M}_S \boldsymbol{V},$$

where ρ and $\boldsymbol{u}$ are the density and the velocity of the fluid respectively, $\boldsymbol{V}$ is the generalized velocity of the solid, $\boldsymbol{V}^*$ is the adjoint of $\boldsymbol{V}$, and $\boldsymbol{M}_S$ is the mass linear operator. Then we solve the similar equation to the previous part. And in the projection term, we process the boundary condition again.

The simulation results of solid-flow coupling can be seen in Fig.7. The animation not only retains the volume matching with the real world but also creates a new scene of collision with solid.

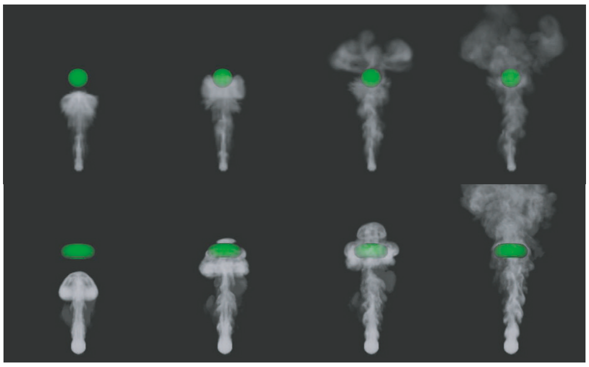


Fig.7. Results of solid-fluid coupling.

6.3 Artistic Control

We wish to provide an easy way for artists to control the fluid. Assume that there is a wind in the environment, we can add an external force to the N-S equation in the Eulerian simulation. Besides, we should make a correction in the velocity estimation problem via

$$\boldsymbol{u} = \boldsymbol{u} + \Delta t \boldsymbol{f},$$

where $\boldsymbol{f}$ is the force of wind. Moreover, we can control the fluid to a particular shape in the local position. $\boldsymbol{f}$ in the above formula can be a control force. Fig.8 shows an example of the simulation result of adding a wind. There is a wind that blows to the right and later turns to the left.

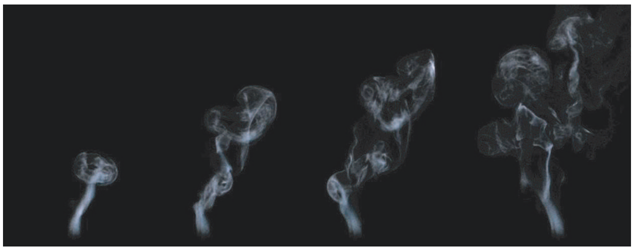


Fig.8. Results of adding a wind.

There are several similar methods, such as "Volume Modeling"[6], "Video-Based Reconstruction"[11] and "From Capture to Simulation"[7]. Compared with these methods, our method has advantages in some aspects. Only one single-view video is needed, thereby our approach does not require sophisticated hardware setups. Based on reasonable assumptions, plausible density fields can be reconstructed. Corrected by physical simulation, the reconstructed velocity fields can be more accurate. In addition, we can add controllable details with a tight coupling.

Our work still has several limitations. Since we have only a single video available, the volume of flow should be approximately symmetrical. Otherwise, the derived physical attributes might not be very meaningful towards re-simulation, animation control, and detail augmentation. In addition, our method is limited to simulation over uniform grids at present.

7 Conclusions

In this paper, we presented a novel technique coupling the video-based reconstruction and physically-based simulation tightly, and our goal is to make some contributions to possible hybrid modeling for flow simulation. The density and velocity fields have been reconstructed from a single-view video, and are then used to guide the Eulerian simulation. Through various graphics examples, we found that our method can not only generate physically-meaningful animation matching with realistic scenes and retaining properly-reconstructed physical quantities, but also enhance missing yet interesting visual details.

In the future, we wish to extend this approach to other flow types, such as fountain and flame. We believe that the methodology is also compatible in these situations. In terms of algorithm framework, we want to achieve a stronger two-way coupling between video data and physical simulation. In this way, reconstructed data can be improved through simulation and then be used to edit flows. In addition, we wish to conduct more graphics experiments and seek possible extensions based on our current framework.

References

[1] Ihmsen M, Orthmann J, Solenthaler B, Kolb A, Teschner M. SPH fluids in computer graphics. In *Proc. Eurographics 2014 — State of the Art Reports*, Apr. 2014.

[2] Bridson R. Fluid Simulation for Computer Graphics. CRC Press, 2008.

[3] Zuo Q, Qi Y, Qin H. A novel, integrated smoke simulation design method supporting local projection and guiding control over adaptive grids. *The Visual Computer*, 2013, 29(9): 883-892.

[4] Zhang X, Bridson R, Greif C. Restoring the missing vorticity in advection-projection fluid solvers. *ACM Transactions on Graphics*, 2015, 34(4): 52:1-52:8.

[5] Teng Y, Levin D I W, Kim T. Eulerian solid-fluid coupling. *ACM Transactions on Graphics*, 2016, 35(6): 1-8.

[6] Okabe M, Dobashi Y, Anjyo K, Onai R. Fluid volume modeling from sparse multi-view images by appearance transfer. *ACM Transactions on Graphics*, 2015, 34(4): 93:1-93:10.

[7] Gregson J, Ihrke I, Thuerey N, Heidrich W. From capture to simulation: Connecting forward and inverse problems in fluids. *ACM Transactions on Graphics*, 2014, 33(4): 70-79.

[8] Zhai X, Hou F, Qin H, Hao A. Inverse modelling of incompressible gas flow in subspace. *Computer Graphics Forum*, 2017, 36(6): 100-111.

[9] Quan H, Yu M, Song X, Gao Y. Real time reconstruction of fluid in video. *International Journal of Modeling, Simulation, and Scientific Computing*, 2013, 4(04): 1342001.

[10] Ren B, Jiang Y, Li C, Lin M C. A simple approach for bubble modelling from multiphase fluid simulation. *Computational Visual Media*, 2015, 1(2): 171-181.

[11] Wang C, Wang C, Qin H, Zhang T. Video-based fluid reconstruction and its coupling with SPH simulation. *The Visual Computer*, 2017, 33(9): 1211-1224.

[12] Foster N, Metaxas D. Realistic animation of liquids. *Graphical Models & Image Processing*, 1996, 58(5): 471-483.

[13] Müller M, Charypar D, Gross M. Particle-based fluid simulation for interactive applications. In *Proc. the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, July 2003, pp.154-159.

[14] Ihmsen M, Cornelis J, Solenthaler B, Horvath C, Teschner M. Implicit incompressible SPH. *IEEE Transactions on Visualization & Computer Graphics*, 2014, 20(3): 426-435.

[15] Solenthaler B, Pajarola R. Predictive-corrective incompressible SPH. *ACM Transactions on Graphics*, 2009, 28(3): 1-6.

[16] Schechter H, Bridson R. Ghost SPH for animating water. *ACM Transactions on Graphics*, 2012, 31(4): Article No. 61.

[17] Wang X K, Ban X J, Zhang Y L, Liu S N, Ye P F. Surface tension model based on implicit incompressible smoothed particle hydrodynamics for fluid simulation. *Journal of Computer Science and Technology*, 2017, 32(6): 1186-1197.

[18] Park S I, Kim M J. Vortex fluid for gaseous phenomena. In *Proc. the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, July 2005, pp.261-270.

[19] Macklin M. Position based fluids. *ACM Transactions on Graphics*, 2013, 32(32): 104:1-104:12.

[20] Stam J. Stable fluids. In *Proc. the 26th Annual Conference on Computer Graphics and Interactive Techniques*, Aug. 1999, pp.121-128.

[21] Zhu Y, Bridson R. Animating sand as a fluid. *ACM Transactions on Graphics*, 2005, 24(3): 965-972.

[22] Batty C, Bertails F, Bridson R. A fast variational framework for accurate solid-fluid coupling. *ACM Transactions on Graphics*, 2007, 26(3): Article No. 100.

[23] Losasso F, Fedkiw R. Simulating water and smoke with an octree data structure. *ACM Transactions on Graphics*, 2004, 23(3): 457-462.

[24] Chentanez N, Mueller-Fischer M. A multigrid fluid pressure solver handling separating solid boundary conditions. *IEEE Transactions on Visualization & Computer Graphics*, 2012, 18(8): 1191-1201.

[25] McAdams A, Sifakis E, Teran J. A parallel multigrid poisson solver for fluids simulation on large grids. In *Proc. the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, July 2010, pp.65-74.

[26] Chentanez N. Real-time Eulerian water simulation using a restricted tall cell grid. *ACM Transactions on Graphics*, 2011, 30(4): Article No. 82.

[27] Ihmsen M, Akinci N, Becker M, Teschner M. A parallel SPH implementation on multi-core CPUs. *Computer Graphics Forum*, 2011, 30(1): 99-112.

[28] Ihrke I, Magnor M. Image-based tomographic reconstruction of flames. In *Proc. the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, August 2004, pp.365-373.

[29] Atcheson B, Ihrke I, HeidrichW, Tevs A, Bradley D, Magnor M, Seidel H P. Time-resolved 3D capture of non-stationary gas flows. *ACM Transactions on Graphics*, 2008, 27(5): 1-9.

[30] Gregson J, Krimerman M, Hullin M B, Heidrich W. Stochastic tomography and its applications in 3D imaging of mixing fluids. *ACM Transactions on Graphics*, 2012, 31(4): 1-10.

[31] Liu Z, Hu Y, Qi Y. Modeling of smoke from a single view. In *Proc. International Conference on Virtual Reality and Visualization*, December 2011, pp.291-294.

[32] Stephan W, Dirk L, Marcus M. Fast image-based modeling of astronomical nebulae. *Computer Graphics Forum*, 2013, 32(32): 93-100.

[33] Keane R D, Adrian R J. Theory of crosscorrelation analysis of PIV images. *Flow, Turbulence and Combustion*, 1992, 49(3): 191-215.

[34] Liu T, Shen L. Fluid flow and optical flow. *Journal of Fluid Mechanics*, 2008, 614(614): 253-291.

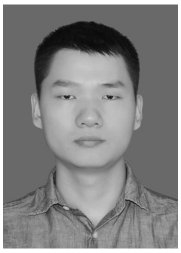
[35] Kadri-Harouna S, Dérian P, Héas P, Mémin E. Divergence-free wavelets and high order regularization. *International Journal of Computer Vision*, 2013, 103(1): 80-99.

[36] Zuo Q, Qi Y. A novel spatial-temporal optical flow method for estimating the velocity fields of a fluid sequence. *The Visual Computer*, 2017, 33(3): 293-302.

[37] Kim B M, Liu Y, Llamas I, Rossignac J. FlowFixer: Using BFECC for fluid simulation. In *Proc. Eurographics Conference on Natural Phenomena*, Aug. 2005, pp.51-56.

[38] Horn B K, Schunck B G. Determining optical flow. *Artificial Intelligence*, 1981, 17(1/2/3): 185-203.

[39] Parikh N, Boyd S. Proximal algorithms. *Foundations & Trends in Optimization*, 2014, 1(3): 127-239.



Feng-Yu Li is a graduate student of School of Computer Science and Software Engineering, East China Normal University, Shanghai. He received his B.E. degree in communication engineering from Zhengzhou University, Zhengzhou. His research interest is fluid simulation based on physics and video-based reconstruction.

Chang-Bo Wang is a professor of School of Computer Science and Software Engineering, East China Normal University, Shanghai. He received his Ph.D. degree in applied mathematics at the State Key Laboratory of Computer Aided Design and Computer Graphics, Zhejiang University, Hangzhou, in 2006, and his B.E. degree in 1998 and M.E. degree in civil engineering in 2002, both from Wuhan University of Technology, Wuhan. His research interests include physically based modeling and rendering, computer animation and realistic image synthesis and information visualization.

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

Hong-Yan Quan is a teacher in the School of Computer Science and Software Engineering, East China Normal University, Shanghai. She received her B.S. and M.S. degrees in computer science from Harbin University of Science and Technology, Harbin, and her Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, in 2006. Her research interests include computer graphics, computer vision, and virtual reality. She has published research papers in video-based modeling for fluid phenomena. She is a member of the IEEE Computer Society.