Video-Guided Real-to-Virtual Parameter Transfer for Viscous Fluids

TETSUYA TAKAHASHI, University of North Carolina at Chapel Hill
MING C. LIN, University of Maryland at College Park, University of North Carolina at Chapel Hill

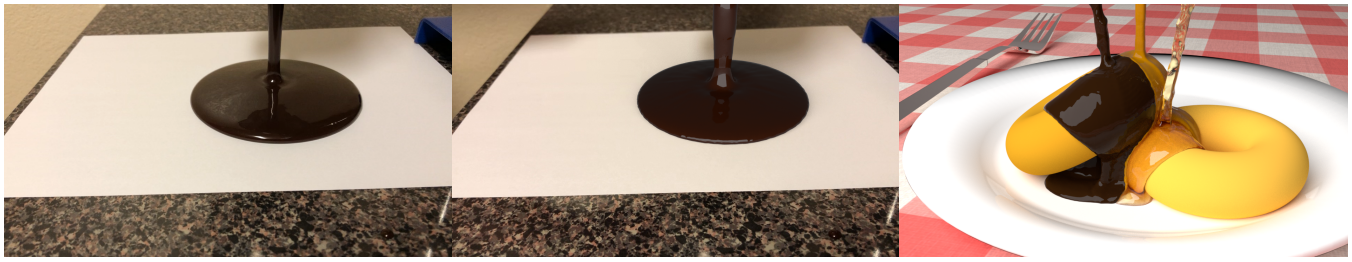

Fig. 1. Our framework identifies the set of physical variables and viscosity parameters from example videos capturing fluid flows in the real world (left) by approximating the flows with viscous fluid simulation (middle). The identified physical values and parameters can then be used to simulate viscous fluids in a new scenario, preserving the style of the fluid flows in the example videos (right).

In physically-based simulation, it is essential to choose appropriate material parameters to generate desirable simulation results. In many cases, however, choosing appropriate material parameters is very challenging, and often tedious trial-and-error parameter tuning steps are inevitable. In this paper, we propose a real-to-virtual parameter transfer framework that identifies material parameters of viscous fluids with example video data captured from real-world phenomena. Our method first extracts positional data of fluids and then uses the extracted data as a reference to identify the viscosity parameters, combining forward viscous fluid simulations and parameter optimization in an iterative process. We evaluate our method with a range of synthetic and real-world example data, and demonstrate that our method can identify the hidden physical variables and viscosity parameters. This set of recovered physical variables and parameters can then be effectively used in novel scenarios to generate viscous fluid behaviors visually consistent with the example videos.

CCS Concepts: • **Computing methodologies** → **Physically-based simulation**; **Material parameter optimization**.

Additional Key Words and Phrases: Fluid simulation, data-driven simulation, viscosity, optimization

ACM Reference Format:
Tetsuya Takahashi and Ming C. Lin. 2019. Video-Guided Real-to-Virtual Parameter Transfer for Viscous Fluids. *ACM Trans. Graph.* 38, 6, Article 237 (November 2019), 12 pages. https://doi.org/10.1145/3355089.3356551

1 INTRODUCTION

Fluids are ubiquitous and common – encountered in various aspects of our everyday lives. Examples of these materials include water, milk, honey, machinery oil, molten chocolate, paint, and shampoo. These liquids have different properties and exhibit distinct behaviors. One key factor that determines fluid properties and behaviors is viscosity, as this can be realized from the Reynolds number, which consists of viscosity parameters and characterizes flow patterns of fluids. For example, fluids with low viscosity values flow vividly generating turbulence and splashes, whereas highly viscous fluids exhibit damped motions and characteristic rotational behaviors, such as buckling phenomena. While many of previous works have focused on inviscid fluids, several researchers have attempted to more accurately simulate the dynamics of highly viscous fluids and improved the visual fidelity with physically-based viscosity models [Batty and Bridson 2008; Batty et al. 2012; Bergou et al. 2010; Carlson et al. 2002; Larionov et al. 2017; Zhu et al. 2015].

While physically-based approaches can effectively simulate viscous fluids based on the physical properties of fluids, one known challenge is that it can be very difficult, time-consuming, and tedious to choose appropriate parameters to generate desirable fluid behaviors, e.g., approximating behaviors of viscous fluids observed in the real world. If physical parameters are not ideal, these approaches would generate visually disconcerting results, which negatively impact our sense and recognition of the fluid materials and dampen our experience in various applications, such as video games, movies, and virtual reality. Even worse, such parameters would cause simulation failure leading to unpredictable results. Consequently, it is necessary to manually tune parameters through laborious trial-and-error processes until satisfactory visual results are obtained. In practice, fluid simulation can take several hours or more, requiring many hours of waiting time to check intermediate results. Thus, such manual parameter-tuning is beyond practical.

One possibility to select appropriate parameters for fluid simulation is to adopt material parameters listed in a book or measured in the real world (e.g., using a viscometer and a rheometer). In general, however, viscosity values of most of fluid materials are not available at hand, and such instruments are not widely available for personal use, as mentioned in [Nagasawa et al. 2019]. In addition, fluid simulation is one way to approximate the behaviors of real complex fluid

Authors' addresses: Tetsuya Takahashi, Computer Science, University of North Carolina at Chapel Hill, tetsuya@cs.unc.edu; Ming C. Lin, Computer Science, University of Maryland at College Park, University of North Carolina at Chapel Hill, lin@cs.umd.edu.

0730-0301/2019/11-ART237 $15.00
https://doi.org/10.1145/3355089.3356551

flows using a simplified mathematical model of physics to make the simulation tractable, and it is known that different fluid simulation methods often lead to distinct simulation results even though the same governing equations are solved under the same simulation setting and physical parameters [Um et al. 2017]. As such, there is no guarantee that fluid simulation with the viscosity values found in a handbook or measured with a viscometer can generate fluid behaviors similar to those observed in the real world, except for few limited, ideal situations. Furthermore, since some fluid simulation methods are devised to improve the efficiency, robustness, and capability based on heuristics, these methods may not have physical parameters corresponding to their counterparts in the real world. From a viewpoint of artists, physical parameters are not necessarily intuitive enough to generate their conceived fluid effects because changes in physical parameters modify fluid flows in a complex and unpredictable way, and the same material parameters can lead to different behaviors depending on simulation scales. Instead, one possible approach is to use examples of desired fluid behaviors to infer appropriate parameters based on the given observed examples.

In this paper, we propose a new material parameter optimization framework for facilitating parameter identification for fluids with example videos captured in the real world. Our framework takes as input a video capturing real fluid flows and extracts positional information of fluids from the example video for a reference. Then, we identify the set of physical values and viscosity parameters by minimizing the differences between the example video and fluids simulated with our viscous fluid solver in an iterative process. Since it is challenging to accurately reconstruct 3D fluid information from 2D videos, we measure the differences of the example data and our simulation results in the 2D screen space by projecting our 3D simulation results onto the screen space. Because of the 3D simulation analysis in the iterative process, the results of the forward simulation with the identified parameters allow us to infer hidden physical quantities of fluids in the videos. Furthermore, the identified parameters can be used in completely new scenarios while preserving the styles of the fluid behaviors in the example videos. To show the effectiveness of our framework, we validate the identified parameters in various scenarios, infer hidden physical quantities, and demonstrate the parameter transfer from the real world to virtual environments.

In summary, our main contributions and key results include:

- **A parameter optimization framework** that identifies the viscosity parameters for fluids based on example videos captured from real-world fluid phenomena, inferring hidden physical quantities of fluids.
- **Screen-space evaluation** that allows for measuring differences between the example data and simulation results without reconstructing 3D data.
- **Parameter transfer from real to virtual environments**. It introduces a new data-driven approach for fluid animation and enables us to reproduce fluid behaviors in the virtual environment, preserving the observed fluid properties in the real world.

To the best of our knowledge, our framework is the first method for identifying material parameters of Newtonian fluids with a single-view video, and Figure 1 demonstrates the effectiveness of our framework.

2 RELATED WORK

Fluid simulation has been a major research topic of significant interest in computer graphics, and various techniques have been proposed. In this section, we focus our discussion on previous works closely related to our method. Later, we also discuss several works on material parameter estimation.

2.1 Viscous Fluid Simulation

Viscous fluids exhibit behaviors different from inviscid fluids, and reproducing their characteristic behaviors has been required over years for various applications. In the Eulerian approach, Stam [1999] developed a stable fluid method using implicit integration with the Laplacian form of viscosity for fluids without free surfaces. Later, Carlson et al. [2002] extended the method with implicit viscosity integration for fluids with free surfaces, compromising the accurate handling of rotational motions at the free surfaces. Rasmussen et al. [2004] augmented the implicit Laplacian-based formulation with explicitly integrated off-diagonal components to account for the rotational behaviors while sacrificing the robustness of their solver. Batty and Bridson [2008] proposed a fully implicit viscosity integration scheme for the full form of viscosity to improve the simulation accuracy in the free surface handling, and this approach was extended for adaptive tetrahedral meshes [Batty and Houston 2011] and octree data structures [Goldade et al. 2019], and for two-way solid-fluid coupling [Takahashi and Lin 2019]. Larionov et al. [2017] proposed a pressure-viscosity coupled solver to further improve the accuracy in the free surface handling. Recently, Kim et al. [2019] proposed an efficient deep-learning-based framework to interpolate simulation results using different viscosity values. Unlike these approaches for 3D volumes, Vantzos et al. [2018] proposed an efficient two-dimensional approach to simulating viscous thin films.

To simulate more general fluids, e.g., viscoelastic fluids and non-Newtonian fluids, various approaches have been also proposed. Goktekin et al. [2004] presented a method for simulating viscoelastic fluids with extra elastic forces. To handle fluids with a variety of properties in a unified way, material point methods have been widely adopted with some specialized extensions for snow [Stomakhin et al. 2013], foams [Yue et al. 2015], melting solids [Stomakhin et al. 2014], elastoplastic solids [Fang et al. 2019; Gao et al. 2017], and granular materials [Daviet and Bertails-Descoubes 2016; Klár et al. 2016; Yue et al. 2018]. While these approaches allow us to simulate a wider range of materials, the computational cost is generally higher than those for purely Newtonian fluids, and the optimization requires more iterations due to the larger number of optimizable variables in the constitutive laws. Although these simulation methods can be adopted in our framework, in this paper, we focus on purely Newtonian viscous fluids.

In the Lagrangian setting, one commonly used approach to simulating viscous fluids is based on Smoothed Particle Hydrodynamics (SPH), and various approaches have been proposed to improve the efficiency and robustness. Takahashi et al. [2015] proposed an implicit viscosity integration to improve the robustness compared to explicit

integration, adopting the method of [Batty and Bridson 2008]. To further improve the efficiency, Peer et al. [2015] presented a different implicit viscosity integration model with prescribed gradient, compromising the physical consistency, and this approach was extended to improve the diffusivity of the vorticity [Peer and Teschner 2017] and to support a wider range of viscous fluid behaviors [Bender and Koschier 2016]. Recently, Weiler et al. [2018] presented a robust and efficient implicit viscosity formulation while achieving physical consistency. To handle fluids with various properties in a unified way, Barreiro et al. [2017] proposed using conformation constraints within the position-based dynamics framework.

Unlike these approaches based on SPH, some works simulated viscous fluids by formulating particle interactions using spring forces between particles [Clavet et al. 2005; Takahashi et al. 2014]. Taking advantages of Lagrangian discretization, several specialized techniques based on simplicial complexes have been proposed to simulate viscous threads and sheets [Batty et al. 2012; Bergou et al. 2010; Zhu et al. 2015].

In the fluid simulation literature, a variety of simulation methods have been proposed to simulate viscous fluids. However, few research has been conducted to select appropriate parameters for fluid simulation. Recently, Nagasawa et al. [2019] proposed a parameter blending scheme for the method of [Yue et al. 2015]. Their method optimized coefficients based on the data measured from the real fluids using a viscometer to better approximate the behaviors of blended fluid materials. Although their work and ours share the similar goal of finding appropriate parameters to generate visually plausible results, these approaches are orthogonal. In their work, material parameters are determined based on the mixture ratio of fluid materials and the blending model whose coefficients are precomputed with the measured viscosity values. In contrast, our framework identifies material parameters using a single-view example video captured from real-world fluid phenomena, through the iterative inversion.

2.2 Fluid Capturing

Capturing fluids has been a challenging problem over the decades. One reason is that fluids do not have their rest shapes, and this fact makes it unreasonable to assume predefined shapes or deformations from specific shapes, which can be effectively used in capturing the dynamics of rigid bodies [Monszpart et al. 2016] and deformable objects [Wang et al. 2015]. In addition, the appearance of fluids can be easily and significantly affected by surrounding environments, e.g., due to light scattering, absorption, reflection, and refraction, making fluid capturing even more challenging.

An early work to model fluid volumes was proposed by Ihrke and Magnor [2004] and Hasinoff and Kutulakos [2007]. They reconstructed fluid volumes by solving a least squares problem, penalizing differences between numerically computed pixel intensity and observed intensity. These approaches were extended to avoid blurry, reconstructed volumes by transferring the appearance of fluid volumes [Okabe et al. 2015]. For the dynamic 3D volume reconstruction, several researchers made use of volume representations, similar to tomography. Atcheson et al. [2008] modeled dynamic gaseous volumes based on information captured with multiple cameras. Gregson et al. [2012] focused on fluid mixing based on dye concentrations. A similar minimization approach was employed to reconstruct 3D liquid surfaces, but not volumes, with submerged checker board patterns [Morris and Kutulakos 2011]. The main focus of these works are on modeling fluid geometry, and velocities of the fluids are not inferred or roughly estimated with an assumption on the rotational symmetry.

To compute more accurate velocity fields based on data captured from real-world phenomena, e.g., videos, several researchers have proposed methods that combine fluid simulation with iterative inversion. Wang et al. [2009] reconstructed not only fluid volumes but fluid velocities from fluid videos, which were captured using synchronized stereo cameras with dyed fluids. Li et al. [2013] recovered water surfaces and their velocities by combining the shallow water simulation with water surfaces reconstructed using a shape reconstruction method based on shading. Gregson et al. [2014] proposed a velocity reconstruction framework based on an optical flow method with physics regularizer terms similar to [Chen et al. 2016; Corpetti et al. 2002], combining tomographic 3D volume information captured with the method of [Gregson et al. 2012], and this framework was augmented to achieve the velocity reconstruction from a single-view video [Eckert et al. 2018]. Recently, Zang et al. [2019] proposed a tomographic reconstruction algorithm for time-varying deforming objects, capturing both of the volumes and deformation fields.

In the physics literature, researchers often utilized sophisticated hardware to directly capture the fluid volumes or velocity fields. One popular approach is Particle Image Velocimetry (PIV), and a good overview for PIV is given in [Grant 1997]. PIV injects tiny particles into fluids, illuminates the particles with a sheet of laser light, and then estimates the particle movements and fluid velocities. In the graphics literature, Xiong et al. [2017] proposed a new PIV algorithm that colors particles based on their depth to track 3D velocity fields with a single camera.

While various algorithms have been proposed for fluid capturing, these algorithms typically require a sophisticated setup, such as synchronized cameras, dyed fluids, and laser device. Thus, in our framework, we avoid such setup and use a commonly available device, smartphone, for fluid capturing. However, we note that these capturing techniques are orthogonal to our goal and can be easily incorporated into our framework.

2.3 Material Parameter Estimation

In physically-based simulations, choosing simulation parameters is one of the most critical steps to generate visually plausible results or even to perform stable simulations. Because of the importance and difficulty in tuning physical parameters, various researchers have attempted to facilitate this process.

One commonly used approach for material parameter estimation is to find optimal parameters that generate behaviors close to example data, e.g., captured in the real world, and this approach has been extensively adopted in the literature, especially for deformable solids [Gerlach and Matzenmiller 2007]. Pai et al. [2001] proposed a method for acquiring material parameters from interactions with

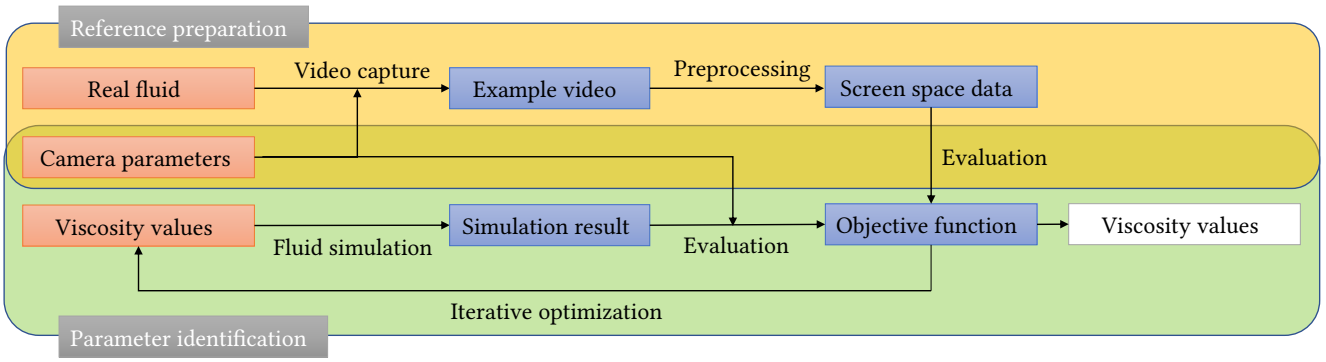

Fig. 2. Overview of our parameter identification framework. Our framework consists of two stages: reference preparation and parameter identification. In the reference preparation stage, we capture a video of real fluid flows and preprocess the video to extract positional information of the fluid. In the parameter identification stage, we iteratively perform fluid simulation, project simulated fluids onto the screen space, and evaluate objective functions with the extracted fluid data. Finally, our framework outputs identified viscosity values.

deformable objects via robotic measurement facility. Becker and Teschner [2007] proposed a framework to optimize elasticity parameters with linear Finite Element Method based on the relation between the initial undeformed geometry and applied forces. Lee and Lin [2012] also presented a framework to identify material parameters using FEM simulation by minimizing the distances between surface nodes from the simulation and reference. Bickel et al. [2009] proposed a method for optimizing the material properties of deformable objects with deformation measurements taken from real-world experiments. Later, Bickel et al. [2010] used their techniques to fabricate deformable objects that have their desirable properties. These material parameter optimization techniques were further extended using model reduction to improve the efficiency [Xu et al. 2015]. Xu and Barbič [2017] presented an optimization framework for damping coefficients to improve the behaviors of deformable objects. Yan et al. [2018] presented an inexact descent approach to accelerating the parameter optimization of elastic materials. Deformation measured in the real world was also used for the parameter identification for clothing [Clyde et al. 2017; Miguel et al. 2012; Wang et al. 2011] and human body [Pai et al. 2018]. In sound rendering, sound captured from various materials was also used as reference data to optimize the audio material parameters [Ren et al. 2013].

While some approaches take example data from the real world for references, these references can be prepared by users. Twigg et al. [2011] proposed an optimization method that finds a user-specified shape under gravity. A similar optimization approach was employed and extended to handle frictional contact for hair [Derouet-Jourdan et al. 2013] and shells [Ly et al. 2018]. Li et al. [2014] presented a space-time optimization framework that simultaneously optimizes the dynamics and material parameters of subspace deformable objects.

Several researchers also proposed material parameter identification methods based on images and videos to avoid using specialized equipment to estimate deformations and forces. Wang et al. [2015] proposed a material parameter optimization approach by combining expectation maximization method and Nelder-Mead method. Yang and Lin [2016] identified material properties for deformable objects from a few images with the particle swarm method. These material parameter optimization techniques with videos taken in the real world are also applied to cloth [Bhat et al. 2003; Yang et al. 2016, 2017], hair [Hu et al. 2017] and rigid bodies [Bhat et al. 2002; Monszpart et al. 2016].

Although various attempts have been made to facilitate the parameter tuning and selection, little research has been conducted for fluids. In this paper, we address this problem, and propose perhaps the first method for identifying the material parameters for fluids using captured video data.

3 OVERVIEW

Our goal is to identify material parameters of fluids, with which a viscous fluid simulator can generate fluid behaviors as close as possible to the example data captured from real-world phenomena. Figure 2 illustrates our material parameter identification framework. Our framework consists of two stages: reference preparation stage and parameter identification stage. In the reference preparation stage, our framework takes as input example videos captured from real-world fluid phenomena. Then, we preprocess the videos and extract positional data of the fluid so that these data are amenable in the following parameter identification stage. The parameter identification stage is an iterative process and takes initial or refined material parameters as input. In this stage, we first perform forward fluid simulations with the material parameters to obtain simulation results. Next, we project the simulation results onto the screen space with the camera parameters used to capture the example videos, and then evaluate our objective functions, which measure the differences between the example data and projected simulation results. Our framework iteratively refines the material parameters and finally outputs identified material parameters.

In our framework, a viscous fluid solver is iteratively used within the parameter identification stage, and the identified material parameters (which can be used in different scenarios) are for the viscous fluid solver. While our framework is not restricted to a specific fluid solver, for self-containedness, we first briefly review our viscous fluid solver in § 4. The details of our material parameter identification framework are described in § 5.

4 VISCOUS FLUID SOLVER

The dynamics of viscous fluids can be described by the incompressible Navier-Stokes equations given by

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \mathbf{s} + \frac{1}{\rho}\mathbf{f}, \tag{1}$$

$$\mathbf{s} = \eta\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right), \tag{2}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{3}$$

where t denotes time, $\frac{D}{Dt}$ material derivative, $\mathbf{u}$ velocity, ρ density, p pressure, $\mathbf{s}$ symmetric viscous stress tensor, $\mathbf{f}$ external force, and η dynamic viscosity. We include the gravity force and surface tension force based on the standard ghost fluid method [Bridson 2015] (we set the surface tension coefficient as 0.1 $\mathrm{kg/s^2}$ based on our experiments) as external forces. To advance the simulation step, we first address the advection term with the affine particle-in-cell (APIC) approach [Jiang et al. 2015], add external forces, and then handle pressure and viscosity terms simultaneously.

We address the pressure and viscosity terms in a unified and implicit manner as

$$\frac{\mathbf{u}^{t+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \mathbf{s}^{t+1}, \tag{4}$$

$$\mathbf{s}^{t+1} = \eta\left(\nabla\mathbf{u}^{t+1} + \left(\nabla\mathbf{u}^{t+1}\right)^T\right), \tag{5}$$

$$\nabla \cdot \mathbf{u}^{t+1} = 0, \tag{6}$$

where $\mathbf{u}^*$ denotes intermediate velocity after advection and external force steps, and Δt time step size. To solve the unified pressure-viscosity problem, we discretize it based on the variational principle [Larionov et al. 2017] using the volume computation method described in [Takahashi and Lin 2019].

While the dynamic viscosity η can be spatially and temporally varying, in this paper, we focus on the single viscosity value η as most of real Newtonian fluids hold a uniform property over the fluid volume, and this makes the parameter identification and validation of the identification results tractable. In the next section, we aim to identify the viscosity parameter η based on a given example video.

5 VISCOSITY PARAMETER IDENTIFICATION

Our method identifies the viscosity parameters of fluids by minimizing the differences between example data captured from real-world phenomena and fluids simulated with our viscous fluid solver. While multiple formulations can be considered for this parameter identification problem, e.g., with soft constraints, to invalidate physics violations and undesirable local minima [Yan et al. 2018], we formulate the problem with hard constraints as the following constrained space-time optimization problem:

$$\eta = \underset{0 \le \eta}{\arg\min}\, E, \tag{7}$$

$$E = \sum_f \omega_f E_f \quad \text{subject to } \mathbf{Q}_{f+1} = F(\mathbf{Q}_f), \tag{8}$$

where E denotes an objective function, which measures the differences between example data and the simulated fluids, ω weighting coefficients for each frame (we set ω 0 or 1 to exclude some specific frames), $\mathbf{Q}$ a state variable for fluids, F a function for the forward simulation, and frame index $f = 0 \ldots N-1$, where N denotes the number of frames considered in the optimization.

5.1 Objective Function

To use videos as a reference for the parameter identification, it is necessary to extract some information on fluids, such as fluid geometry, that can be compared with results of 3D fluid simulations. In the literature, some works attempted to reconstruct 3D fluid geometry and velocity from videos, e.g., [Li et al. 2013; Okabe et al. 2015; Wang et al. 2009]. However, these approaches typically require a complex equipment setup, such as synchronized multiple cameras, depth sensors, and/or dyed liquid; or they need to restrict fluid motions because it is very challenging to reconstruct 3D fluid data from videos which include 2D information only (i.e., 3D information is already lost). Since there are multiple 3D fluid configurations, which lead to similar 2D fluid configurations on the screen, the 3D data reconstruction from 2D videos is ambiguous, i.e., this problem is under-determined. Additionally, fluids generally have no preferred shape, and thus it is not reasonable to consider the rest shape or deformations from the rest shape, making it difficult to capture the fluid geometry, unlike rigid and deformable bodies. Furthermore, the appearance of fluids can be easily and significantly changed due to the optical properties of fluid surfaces, e.g., with light scattering, absorption, reflection, and refraction, and thus it is very difficult to obtain the reliable 3D fluid data from videos.

Therefore, we eschew reconstructing 3D fluid data and instead measure the differences between the example data and results of 3D simulation on the 2D screen space (with the same size as example videos). In our framework, we evaluate the differences in terms of the fluid geometry in the 2D space (i.e., as a silhouette) and define our objective function for frame index f as

$$E_f = \frac{1}{2M}(\mathbf{g}_f - \hat{\mathbf{g}_f})^T \mathbf{C}_f (\mathbf{g}_f - \hat{\mathbf{g}_f}) \tag{9}$$

where M denotes the total count of pixels in the screen space, $\mathbf{C}$ a diagonal coefficient matrix (we set entries in C as 0 or 1 to exclude specific domains in the screen space), $\mathbf{g}$ and $\hat{\mathbf{g}}$ denote silhouette obtained from simulation results and extracted from the example videos, respectively. We use binary values for $\mathbf{g}$ and $\hat{\mathbf{g}}$, and define them at each pixel in the 2D screen space.

5.2 Fluid Video Capturing

For the reference silhouette $\hat{\mathbf{g}}$, we first capture example fluid videos. While there are various ways to capture the fluid videos, it is important to adopt a setup, which can be easily prepared and used to capture different viscous fluid materials while minimizing sources of errors (e.g., human interventions and gaps between the simulation and real fluid flows) as much as possible. Although one intuitive setup would be to pour liquids from a container, we found that liquid pouring is not ideal because it requires some human interventions (i.e., manipulations of the container) and forms very thin fluid sheets near the edge of the container, causing too strong surface tension forces which dominate viscosity forces. Given these, we prepare a simple setup, where viscous fluids flow from the hole at the bottom

Fig. 3. Setup for capturing a single-view video for behaviors of viscous fluids. Viscous fluids flow out from the hole at the bottom of the container due to the gravity, and the fluid flow is captured with a smartphone fixed using a stand.

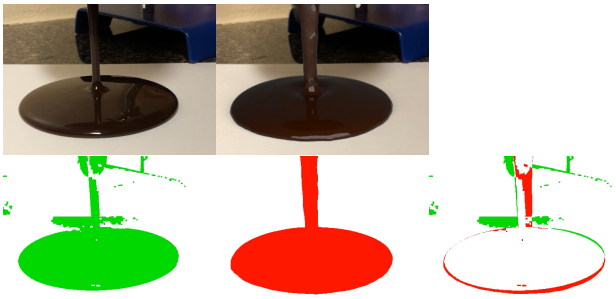

Fig. 4. (Top) from left to right, example fluid video, and simulation result. (Bottom) Extracted silhouette from the example video, projections of fluid surfaces from the simulation, and the differences between the silhouette from the example data and simulation result.

of the container due to the gravity, as shown in Figure 3 (for more details, please see the supplementary video).

In our work, we use a normal smartphone, iPhone 8, and capture the fluid flows with the resolution of 1280×720 at 30 fps from a single view. We fix the camera positions, calibrate the camera to obtain intrinsic parameters in advance, and use these parameters to obtain extrinsic parameters. An image of a captured fluid video is given in Figure 4 (top left).

5.3 Fluid Data Extraction from Video

To compute the silhouette for the reference, we extract positional data from the 2D example fluid videos. To this end, we use the standard background subtraction method based on Gaussian mixture modeling [Zivkovic 2004]. Then, we separate the extracted silhouette, i.e., foreground from the background with a thresholding method. Finally, we perform the morphological, closing and opening operations for the extracted foreground at the pixel level to remove some noisy estimates. We define the foreground as 1 and background as 0 for $\mathbf{g}$. The extracted silhouette is shown in Figure 4 (bottom left).

5.4 Screen Space Evaluation

To evaluate the objective function, we compute $\mathbf{g}$, silhouette of simulated fluids on the 2D screen space at each frame through the forward simulations. Since geometry of fluids is represented by a set of particles, we first construct fluid surfaces to approximate the surfaces of the real fluid flows, and then project the surfaces onto the 2D screen space using the camera parameters which are used to capture the example videos.

To construct fluid surfaces, we take a standard approach. First, we generate implicit functions from the set of particles, construct surfaces using the marching cubes algorithm, and then perform several smoothing operations to better approximate the real fluid surfaces. In our work, we represent the surfaces with a set of triangles for the ease of projections onto the screen space.

Next, in the projection step, we form the silhouette of the fluid surfaces as a union of all the projected triangles on the screen space. To project each triangle, first, we independently project the three vertices of the triangle in the same way as the camera does. The projection operation can be written as

$$\mathbf{x} = \mathbf{KAX}, \tag{10}$$

where $\mathbf{X}$ and $\mathbf{x}$ denote the homogeneous coordinates of the vertex before and after projection, respectively, $\mathbf{K}$ and $\mathbf{A}$ intrinsic and extrinsic parameters, respectively, which can be computed with a camera calibration technique. After the projection of the three vertices, we can form a new triangle on the screen space. To compute $\mathbf{g}_t$, silhouette formed by a triangle t, we perform the inside/outside check for the center of each pixel, and we assign 1 to $\mathbf{g}_{t,i}$ if the center of pixel i is inside of the silhouette, otherwise 0. Finally, we assemble all the silhouettes from the triangle to form the silhouette of the fluid surfaces, i.e.,

$$\mathbf{g} = \bigcup_t \mathbf{g}_t. \tag{11}$$

Figure 4 (bottom, middle) shows a computed silhouette from the simulation (top, middle), and the silhouette difference is given in Figure 4 (bottom, right). After the projections of all the triangles, we can straightforwardly compute the objective function E.

5.5 Parameter Optimization

Our objective function is formulated with example data captured in the real world, which generally include some noise, and involves multiple discontinuous operations, such as background subtraction and morphological operations, liquid domain computation and surface reconstruction from a set of particles, and projections of the 3D fluid surfaces onto the 2D screen space over multiple steps. Consequently, our objective function is discontinuous and nonlinear with many unacceptable local minima. Since evaluating analytical gradient is not practical for such discontinuous functions [McNamara et al. 2004], it is preferable to employ optimization methods based on sampling which can be used without evaluating the gradient analytically, as done in [Hu et al. 2017; Wang et al. 2015; Yang and Lin 2016]. In addition, sampling-based approaches can naturally satisfy the hard constraint for the physics in the constrained optimization problem (8) by performing forward simulations.

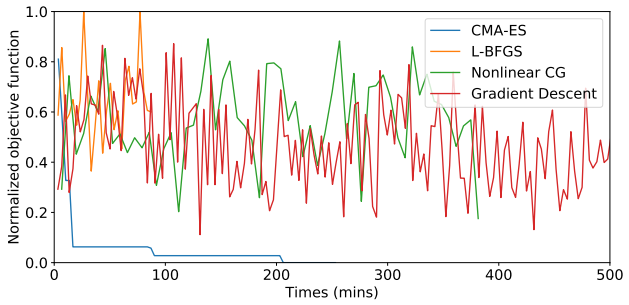

Fig. 5. Plots of the objective functions with different optimizers for the honey shown in Figure 8.

Table 1. Viscosity parameter identification results with synthetic videos. $\hat{\eta}$ denotes reference fluid viscosity (kg/(s · m)), Re Reynolds number, η identified viscosity value (kg/(s · m)), ϵ_η, ϵ_v, and ϵ_p relative errors for the viscosity (%), pressure (%), and velocity (%), respectively. The error for the viscosity is relatively small and up to around 5%.

$\hat{\eta}$	Re	η	ϵ_η	ϵ_p	ϵ_v
1.0×10^0	1.25×10^2	1.06×10^0	5.85	14.1	1.21
3.0×10^0	3.67×10^1	2.94×10^0	1.93	6.60	0.57
1.0×10^1	9.50×10^0	0.98×10^1	1.52	8.28	0.56
3.0×10^1	2.67×10^0	3.02×10^1	0.80	4.34	0.32
1.0×10^2	6.01×10^{-1}	1.01×10^2	1.35	3.34	0.74
3.0×10^2	1.52×10^{-1}	3.10×10^2	3.23	15.6	0.87

In our framework, we use a derivative-free optimizer, CMA-ES [Hansen and Kern 2004], which is known as robust to noise and efficient compared to other derivative-free optimization methods, such as the particle swarm method and Nelder-Mead method. In the optimization, to enforce $0 \leq \eta$, we resample viscosity values if sampled viscosity values are smaller than 0.

While we have tested multiple gradient-based optimizers using finite difference approximations, such as L-BFGS, nonlinear conjugate gradient, and gradient descent with momentum, we found that these approaches almost always got stuck at suboptimal local minima because of the inaccurate estimates of the gradient for the noisy objective functions, and the computational cost for the convergence was higher than CMA-ES in most of our experiments. Figure 5 compares the convergence behaviors for CMA-ES, L-BFGS, nonlinear conjugate gradient, and gradient descent with momentum optimizers.

6 VALIDATIONS AND DISCUSSIONS

We implemented our framework in C++. For the parameter identification, we typically formulate the objective function with up to 100 video frames of high resolutions to make the optimization tractable (i.e., $N \leq 100$). We usually perform up to 80 iterations for CMA-ES optimization with an initial value between 0.0 and 3.0×10^2 kg/(s · m) and standard deviation between 1.0×10^1 and 3.0×10^2 kg/(s · m). The overall computation time varies and depends on the video resolution, the number of video frames, the number of optimization iterations, the scale of fluid simulation, and the computational complexity of the (viscous) fluid solver. We used blender cycles renderer for Figure 4 and Figure 8 (first and second rows) and mitsuba renderer for the others.

We tested our framework in a wide range of scenarios. First, we validate the reliability of our algorithm with synthetic examples, and then we evaluate our framework with example videos captured in the real world.

6.1 Validation with Synthetic Videos

To test our framework, we generated several videos using our viscous fluid solver, and used the videos as input for our framework. The purpose of this experiment is to validate that our algorithm can identify viscosity parameters which are used to generate the synthetic videos, only with 2D data in the screen space.

We chose a scenario, where a viscous fluid flows from the hole at the bottom of a container, as shown in Figure 6 (top). In this scene, we tested with multiple viscosity values, 1.0×10^0, 3.0×10^0, 1.0×10^1, 3.0×10^1, 1.0×10^2, and 3.0×10^2 kg/(s · m), and simulations are executed with the grid resolution of 256^3 and up to 913.6k particles. The simulation parameters and identification results are summarized in Table 1, and we note that this experiment covers a sufficiently wide range of Reynolds numbers for viscous fluids and thus fluid behaviors. A plot for the objective function is given in Figure 7. The second row of Figure 6 demonstrates simulation results in the same scenario with the identified parameters, and in general, visual differences between the reference and the simulated videos are indiscernible.

One advantage of our framework with iterative inversion using the full 3D simulation is that we can infer hidden physical variables which are not available from the video data, e.g., velocity of fluid flows and pressure distributions. Figure 6 visualizes the pressure and velocity distributions for the input example (third and fifth rows) and simulation results with the identified parameters (fourth and sixth rows). Similar to the comparison with the surface rendering, differences for the pressure distributions and velocity fields between real and virtual fluids are generally indiscernible.

In these experiments, we used the same solver for synthetic example generation and parameter identification, and thus resulting fluid behaviors are same if the same viscosity values are used. However, we note that the example data include only rendered fluid surfaces generated by 3D fluid simulations, i.e., projected onto the 2D screen space (losing full 3D information which can be perfectly matched), and positional data are extracted with image processing algorithms, introducing some errors. Consequently, it is not guaranteed that our algorithm finds the ground truth, and the value of the objective function is 0. Nonetheless, our framework can identify viscosity parameters with *up to around 5%* relative errors, only with the 2D information, and the inferred pressure and velocity values are within 20% and 2% of relative errors, respectively (see Table 1). The plot in Figure 7 also demonstrates that the good local minimum is very close to the ground truth viscosity values while the objective function increases as viscosity values deviate from the ground truth.

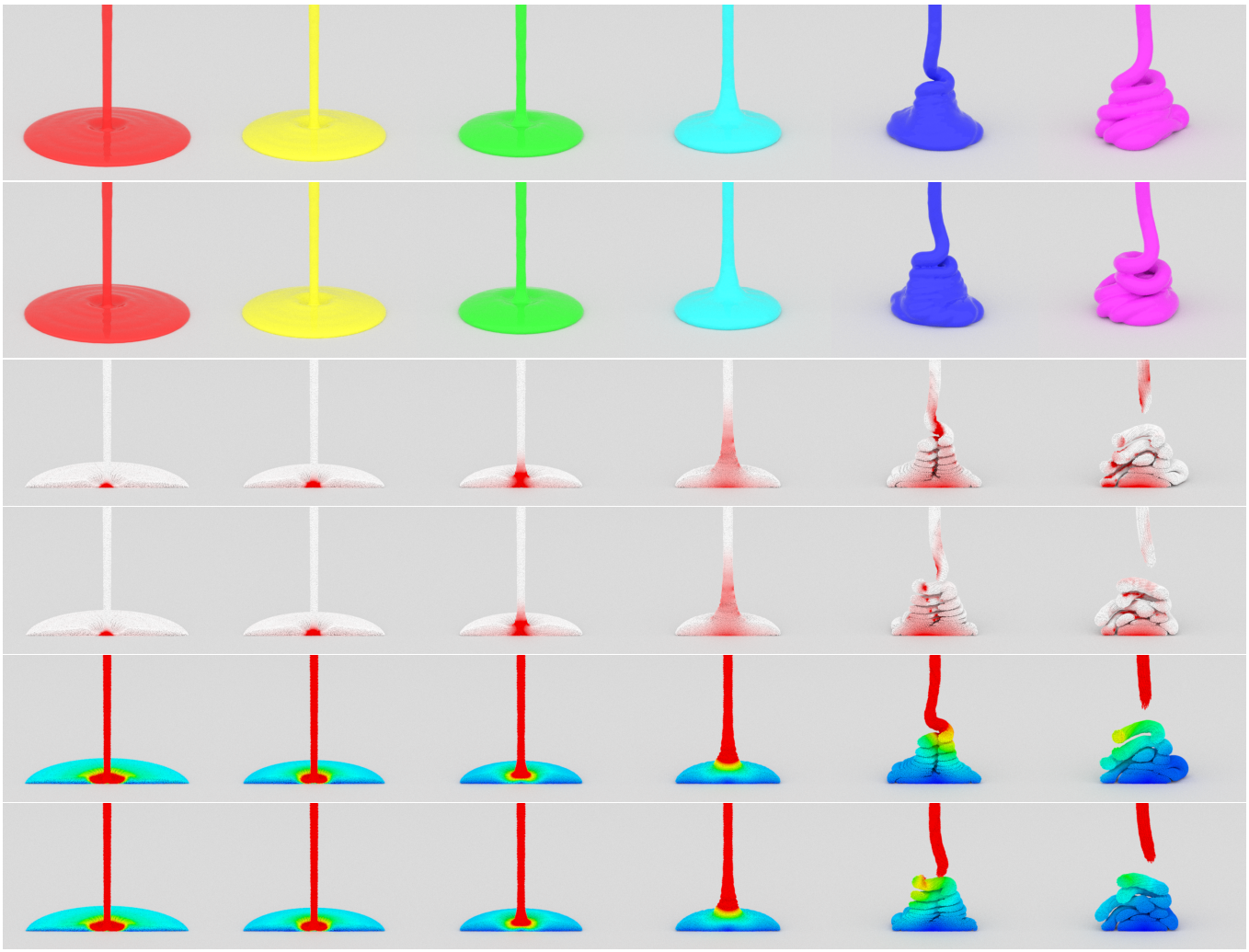

Fig. 6. Validation results with synthetic videos for the scenario of flowing fluids. (First row) from left to right, simulated video as input, with viscosity parameters $\eta = 1.0 \times 10^0$, 3.0×10^0, 1.0×10^1, 3.0×10^1, 1.0×10^2, and 3.0×10^2 kg/(s · m). (Second row) recovered results using our framework with identified viscosity parameters, $\eta = 1.06 \times 10^0$, 2.94×10^0, 0.98×10^1, 3.02×10^1, 1.01×10^2, and 3.10×10^2 kg/(s · m). The relative errors are 5.85%, 1.93%, 1.52%, 0.80%, 1.35%, and 3.23%, respectively. Cutaway particle visualization for pressure profiles for the input (third row) and simulation with the identified parameters (fourth row). Cutaway particle visualization using rainbow colors for velocity profiles as the input (fifth row) and simulation with the identified parameters (sixth row).

6.2 Identification with Real World Captured Data

We also tested our framework with example videos captured from real world fluid phenomena. In this study, we experimented with (HERSHEY's) Caramel Syrup, red (Equaline Antibacterial) Hand Soap, (HERSHEY's) Chocolate Syrup, purple (Softsoap) Liquid Hand Soap, blue (Dove Men+Care) Body and Face Wash, and (Gunter's) Pure Clover Honey under the room temperature of 22.2 °C, and used a cylindrical container (radius is 4.3 cm) with a hole at the bottom (hole radius is 0.9 cm), and fluid volumes of 170 cm^3 for each experiment. To perform the parameter optimization, we setup the simulation scenarios as close as possible to the scene for the real experiments, and simulations are executed with the grid resolution of 160^3 and up to 1,168.7k particles. The captured videos and simulation results with identified viscosity parameters are shown in Figure 8. A plot for the objective function with different viscosity value (we normalized objective functions such that the minimum and maximum values are 0 and 1, respectively, for visualization purposes), and a plot for the convergence behaviors are given in Figure 9. Statistics and performance are summarized in Table 2.

Since the ground truth of viscosity parameters are not available for viscous fluids in our examples, it is not possible to validate the accuracy of the identified parameters. However, Figure 8 demonstrates that the behaviors of the simulated viscous materials with the identified parameters are visually in close agreement with the fluids in the example videos. In addition, we note that the range of viscosity

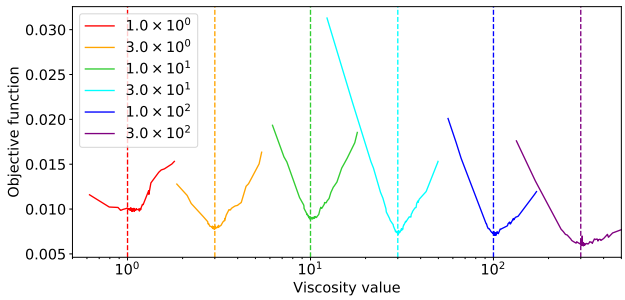

Fig. 7. Plots of the objective functions with different viscosity values for Figures 6. Good local minima are located close to the ground truth (vertical dot lines).

Table 2. Viscosity parameter identification results with example videos captured from real world fluid flows. Re denotes Reynolds number, η identified viscosity value (kg/(s · m)), t average time in minutes for each iteration, T total time in hours for the parameter identification, and $\hat{v}$ and v (cm/s) average flow speed of the fluids estimated from the video and computed from the simulation, respectively.

Materials	Re	η	t	T	$\hat{v}$	v
Caramel	3.16×10^1	0.19	26.3	10.8	6.1	6.7
Red hand soap	2.72×10^1	0.22	14.3	13.9	5.2	5.8
Chocolate syrup	4.80×10^0	1.25	15.2	7.7	4.5	5.3
Purple body soap	6.33×10^{-1}	4.74	19.4	11.7	1.7	1.5
Blue liquid soap	5.15×10^{-1}	5.82	8.6	8.0	1.1	1.3
Honey	2.67×10^{-1}	7.86	7.5	9.0	0.6	0.6

values for honey is known as between 2.0 and 10.0 kg/(s · m), and our identified viscosity value for the honey is 7.86 kg/(s · m) and is within the range, which further validates the reliability of our framework. Furthermore, we note that our framework can identify the viscosity parameters for fluids exhibiting the coiling behaviors, reproducing the buckling phenomena for the blue soap and honey.

Similar to the case for the synthetic videos, one advantage of the iterative inversion using the 3D simulation is that we can estimate hidden variables for the real fluid flows, e.g., pressure and velocity profiles (which are not available in the example videos), as shown in Figure 8 (third and fourth rows). To validate the accuracy of the simulation with the identified parameters, we compare the flow speed of the fluids on the ground, which can be estimated from the example videos. Results are summarized in Table 2, and the average relative errors are up to around 10% in our experiments.

6.3 Real-to-Virtual Parameter Transfer

The identified viscosity parameters can be used in novel scenarios. Figure 10 demonstrates a chocolate coating for a cake with the identified viscosity parameter for the ganache, simulated with the grid resolution of 128^3 and up to 351.0k particles. Figure 11 shows a honey pouring onto a honey dipper with the identified parameter of the honey, simulated with the grid resolution of 256^3 and up to 1,175.0k particles. Figure 12 demonstrates a pouring of magenta hand soap, light-lavender body soap, and aqua-green shampoo onto a hand with the identified parameters, simulated with the grid resolution of 256^3 and up to 1,600.0k particles. Note that the differences in fluid behaviors in the example videos are sufficiently reflected in this scene, generating distinct fluid flows. Thus, we believe that it is undesirable to randomly choose viscosity parameters from the range of typical viscosity values for soap materials, even if such data are available. Figure 1 (right) demonstrates a scene with simulated donuts covered by chocolate syrup, caramel, and honey with the identified parameters, simulated with the grid resolution of 256^3 and up to 1,840.0k particles. In this scene, we also clearly observe that caramel, chocolate syrup, and honey behave very differently according to their material properties.

6.4 Discussions

Our framework can identify material parameters effectively as demonstrated. However, it is not guaranteed that the resulting parameters are close to the measured parameters unless the experiments are conducted under relatively ideal, controlled conditions. There are some factors for this discrepancy. First, fluid simulation is a numerical approximation of the complex fluid flows with a simplified model derived based on various assumptions (e.g., no slip boundary condition and uniformly distributed fluid particles), which might not hold in some cases. In addition, while our focus is on purely Newtonian fluids, some real-world materials exhibit non-Newtonian properties as well, and thus simulation results would deviate from the real fluid behaviors. Given the relatively coarse simulation resolution, it is not possible to accurately capture the small scale details of fluids and solid boundaries, and their resulting influence to the simulation (e.g., neglected boundary details and strong surface tension due to thin fluid sheets).

One factor that affects our optimization results is numerical viscosity. When viscosity values are low, the numerical viscosity can dominate the effect of real viscosity, and the optimizer could erroneously identify the viscosity parameter based on the numerical viscosity.

While our framework can benefit common scenarios as demonstrated, some liquids may not be easily accessible even for video capturing, e.g., lava, blood, molten gold, because of the danger, ethics, and cost.

7 CONCLUSIONS AND FUTURE WORK

We proposed perhaps the first parameter identification framework to facilitate the parameter-tuning for fluid simulation with a single-view example video captured from real-world fluid phenomena. Our framework takes example fluid videos as a reference and minimizes the differences between the reference and simulated fluids (using our solver) to identify material parameters. For the difference measurement with example videos, we presented a screen space evaluation method, which compares the reference and simulation results on the 2D screen space, avoiding erroneous and ambiguous 3D reconstruction of fluid data. We validated our parameter identification framework with a range of synthetic and real-world data and demonstrated that identified material parameters can be effectively used to infer hidden physical variables of real fluids and to simulate

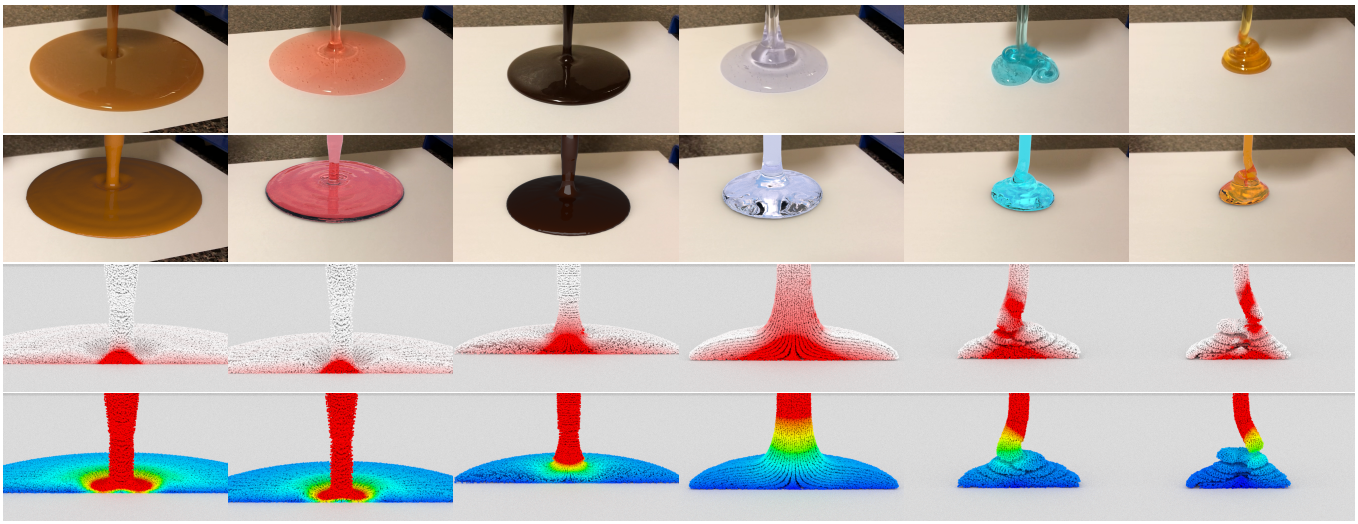

Fig. 8. Parameter identification results with example videos from the real-world fluid phenomena. (Top) from left to right, caramel, red hand soap, chocolate syrup, purple soap, honey, and blue body soap. (Bottom) simulation results with identified viscosity parameters, $\eta = 0.19, 0.22, 1.25, 4.74, 5.82$, and 7.86 kg/(s · m).

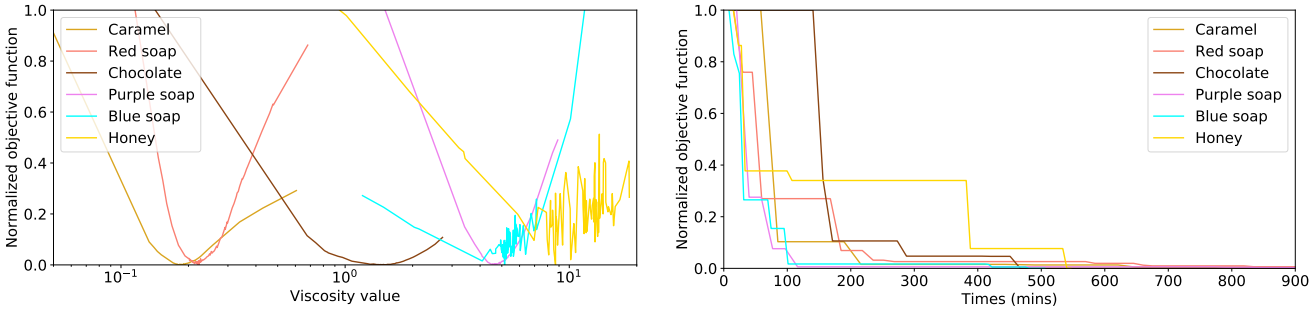

Fig. 9. Plots of the objective functions with different viscosity values (left) and convergence plot for the parameter identification (right), for Figure 8.

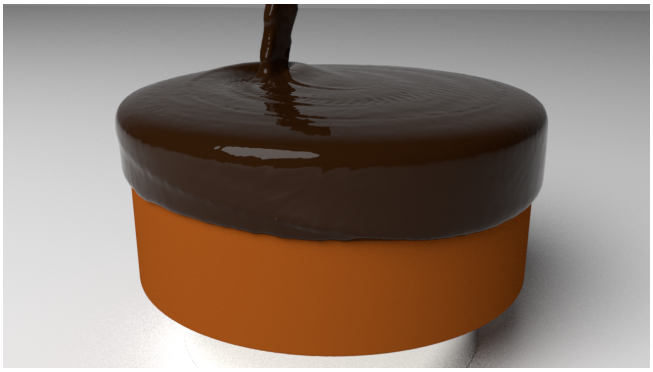

Fig. 10. Simulated chocolate ganache poured onto a cake with the identified viscosity parameter $\eta = 1.25$ kg/(s · m).

Fig. 11. Virtual honey dripped onto a honey dipper with the identified viscosity parameter $\eta = 7.86$ kg/(s · m).

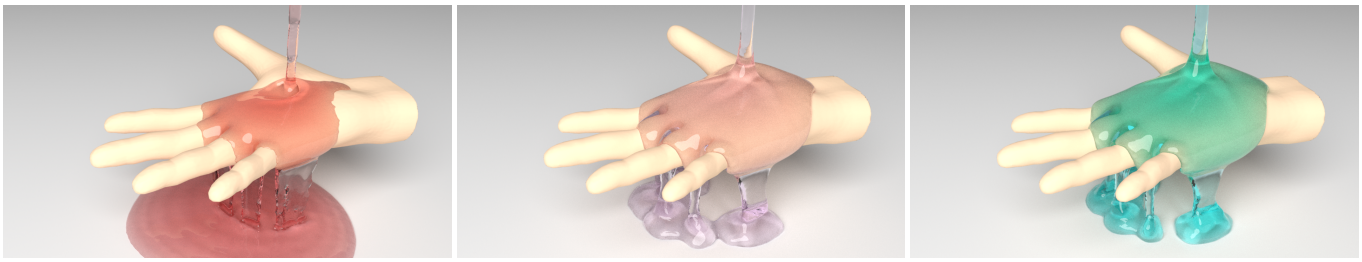

Fig. 12. From left to right, magenta hand soap, light-lavender body soap, and aqua-green shampoo poured onto a hand with the identified viscosity parameters $\eta = 0.22$, 4.74, and 5.82 kg/(s · m), respectively.

viscous fluids in novel scenarios, generating fluid behaviors visually consistent to the example data.

There are several promising future research directions. In the real world, there are many different types of fluid-like materials, such as non-Newtonian fluids and granular materials, which require more complex constitutive laws to simulate. In addition, material parameters for fluids can spatially and temporally vary, e.g., due to heat and stress. It would be interesting to develop a parameter identification framework that can take into account more sophisticated physics and property change. Another important work is to simultaneously identify multiple physical parameters, such as the dynamic viscosity and surface tension coefficient, and this is a challenging problem because the influences of the physical parameters to the fluid behaviors are not necessarily orthogonal, thus making it difficult to individually identify these parameters. While we found that the identification is still possible with the increased computational cost according to our early experiments, it is difficult to quantitatively evaluate the efficacy, and we leave this issue as our future work.

In general, it is difficult to obtain meaningful and reliable fluid data with simple computer vision techniques, such as background subtraction, from normal videos available in public. Thus, it would be necessary to explore some descriptors for fluids which can be reliably used for the difference measurements. We would also like to investigate advanced computer vision techniques and deep learning, to extract fluid information. Along this direction, we believe that learning-based approaches for video analysis and processing are promising.

ACKNOWLEDGMENTS

This work is supported in part by National Science Foundation and Elizabeth Stevinson Iribe Chair Professorship. We would like to thank Christopher Batty and Ryoichi Ando for discussions on the viscous fluid solver and anonymous reviewers for their valuable suggestions and comments.

REFERENCES

Bradley Atcheson, Ivo Ihrke, Wolfgang Heidrich, Art Tevs, Derek Bradley, Marcus Magnor, and Hans-Peter Seidel. 2008. Time-resolved 3D Capture of Non-stationary Gas Flows. *ACM Trans. Graph.* 27, 5, Article 132 (Dec. 2008), 9 pages.

Héctor Barreiro, Ignacio García-Fernández, Iván Alduán, and Miguel A. Otaduy. 2017. Conformation Constraints for Efficient Viscoelastic Fluid Simulation. *ACM Trans. Graph.* 36, 6, Article 221 (2017), 221:1–221:11 pages.

Christopher Batty and Robert Bridson. 2008. Accurate Viscous Free Surfaces for Buckling, Coiling, and Rotating Liquids. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 219–228.

Christopher Batty and Ben Houston. 2011. A Simple Finite Volume Method for Adaptive Viscous Liquids. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 111–118.

Christopher Batty, Andres Uribe, Basile Audoly, and Eitan Grinspun. 2012. Discrete Viscous Sheets. *ACM Transactions on Graphics* 31, 4, Article 113 (2012), 7 pages.

Markus Becker and Matthias Teschner. 2007. Robust and Efficient Estimation of Elasticity Parameters using the linear Finite Element Method.. In *Simulation and Visualization*. 15–28.

J. Bender and D. Koschier. 2016. Divergence-Free SPH for Incompressible and Viscous Fluids. *IEEE Transactions on Visualization and Computer Graphics* PP, 99 (2016), 1–1.

Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete Viscous Threads. *ACM Transactions on Graphics* 29, 4, Article 116 (2010), 10 pages.

Kiran S. Bhat, Steven M. Seitz, and Jovan Popovic. 2002. Computing the Physical Parameters of Rigid-Body Motion from Video. In *Proceedings of the 7th European Conference on Computer Vision-Part I*. 551–565.

Kiran S. Bhat, Christopher D. Twigg, Jessica K. Hodgins, Pradeep K. Khosla, Zoran Popović, and Steven M. Seitz. 2003. Estimating Cloth Simulation Parameters from Video. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 37–51.

Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. 2010. Design and Fabrication of Materials with Desired Deformation Behavior. *ACM Trans. Graph.* 29, 4, Article 63 (July 2010), 10 pages.

Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Wojciech Matusik, Hanspeter Pfister, and Markus Gross. 2009. Capture and Modeling of Non-linear Heterogeneous Soft Tissue. *ACM Trans. Graph.* 28, 3, Article 89 (July 2009), 9 pages.

Robert Bridson. 2015. *Fluid Simulation for Computer Graphics*. A K Peters/CRC Press.

Mark Carlson, Peter J. Mucha, R. Brooks Van Horn, III, and Greg Turk. 2002. Melting and Flowing. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 167–174.

Da Chen, Wenbin Li, and Peter Hall. 2016. Dense Motion Estimation for Smoke. In *Asian Conference on Computer Vision (Asian Conference on Computer Vision)*.

Simon Clavet, Philippe Beaudoin, and Pierre Poulin. 2005. Particle-based Viscoelastic Fluid Simulation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 219–228.

David Clyde, Joseph Teran, and Rasmus Tamstorf. 2017. Modeling and Data-driven Parameter Estimation for Woven Fabrics. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation (SCA '17)*. Article 17, 11 pages.

T. Corpetti, E. Memin, and P. Perez. 2002. Dense estimation of fluid flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 3 (March 2002), 365–380.

Gilles Daviet and Florence Bertails-Descoubes. 2016. A Semi-implicit Material Point Method for the Continuum Simulation of Granular Materials. *ACM Trans. Graph.* 35, 4, Article 102 (2016), 102:1–102:13 pages.

Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, Gilles Daviet, and Joëlle Thollot. 2013. Inverse Dynamic Hair Modeling with Frictional Contact. *ACM Trans. Graph.* 32, 6, Article 159 (Nov. 2013), 159:1–159:10 pages.

Marie-Lena Eckert, Wolfgang Heidrich, and Nils Thürey. 2018. Coupled Fluid Density and Motion from Single Views. *Comput. Graph. Forum* 37, 8 (2018), 47–58.

Yu Fang, Minchen Li, Ming Gao, and Chenfanfu Jiang. 2019. Silly Rubber: An Implicit Material Point Method for Simulating Non-equilibrated Viscoelastic and Elastoplastic Solids. *ACM Trans. Graph.* (2019).

Ming Gao, Andre Pradhana Tampubolon, Chenfanfu Jiang, and Eftychios Sifakis. 2017. An Adaptive Generalized Interpolation Material Point Method for Simulating Elastoplastic Materials. *ACM Trans. Graph.* 36, 6, Article 223 (2017), 223:1–223:12 pages.

S. Gerlach and A. Matzenmiller. 2007. On parameter identification for material and microstructural properties. *GAMM-Mitteilungen* 30, 2 (2007), 481–505.

Tolga G. Goktekin, Adam W. Bargteil, and James F. O'Brien. 2004. A Method for Animating Viscoelastic Fluids. *ACM Trans. Graph.* 23, 3 (2004), 463–468.

Ryan Goldade, Yipeng Wang, Mridul Aanjaneya, and Christopher Batty. 2019. An Adaptive Variational Finite Difference Framework for Efficient Symmetric Octree Viscosity. *ACM Trans. Graph.* (2019).

Ian Grant. 1997. Particle image velocimetry: a review. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 211 (1997), 55–76.

James Gregson, Ivo Ihrke, Nils Thuerey, and Wolfgang Heidrich. 2014. From Capture to Simulation: Connecting Forward and Inverse Problems in Fluids. *ACM Trans. Graph.* 33, 4, Article 139 (July 2014), 139:1–139:11 pages.

James Gregson, Michael Krimerman, Matthias B. Hullin, and Wolfgang Heidrich. 2012. Stochastic Tomography and Its Applications in 3D Imaging of Mixing Fluids. *ACM Trans. Graph.* 31, 4, Article 52 (July 2012), 52:1–52:10 pages.

Nikolaus Hansen and Stefan Kern. 2004. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In *PPSN*.

S. W. Hasinoff and K. N. Kutulakos. 2007. Photo-Consistent Reconstruction of Semi-transparent Scenes by Density-Sheet Decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, 5 (2007), 870–885.

Liwen Hu, Derek Bradley, Hao Li, and Thabo Beeler. 2017. Simulation-Ready Hair Capture. *Comput. Graph. Forum* 36 (2017), 281–294.

Ivo Ihrke and Marcus Magnor. 2004. Image-based Tomographic Reconstruction of Flames. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 365–373.

Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran, and Alexey Stomakhin. 2015. The Affine Particle-in-cell Method. *ACM Trans. Graph.* 34, 4, Article 51 (July 2015), 10 pages.

Byungsoo Kim, Vinicius C. Azevedo, Nils Thuerey, Theodore Kim, Markus H. Gross, and Barbara Solenthaler. 2019. Deep Fluids: A Generative Network for Parameterized Fluid Simulations. *Computer Graphics Forum* (2019).

Gergely Klár, Theodore Gast, Andre Pradhana, Chuyuan Fu, Craig Schroeder, Chenfanfu Jiang, and Joseph Teran. 2016. Drucker-prager Elastoplasticity for Sand Animation. *ACM Trans. Graph.* 35, 4, Article 103 (2016), 103:1–103:12 pages.

Egor Larionov, Christopher Batty, and Robert Bridson. 2017. Variational Stokes: A Unified Pressure-viscosity Solver for Accurate Viscous Liquids. *ACM Trans. Graph.* 36, 4, Article 101 (July 2017), 11 pages.

Huai-Ping Lee and Ming C. Lin. 2012. Fast optimization-based elasticity parameter estimation using reduced models. *The Visual Computer* 28, 6 (2012), 553–562.

Chuan Li, David Pickup, Thomas Saunders, Darren Cosker, David Marshall, Peter Hall, and Philip Willis. 2013. Water Surface Modeling from a Single Viewpoint Video. *IEEE Transactions on Visualization and Computer Graphics* 19, 7 (July 2013), 1242–1251.

Siwang Li, Jin Huang, Fernando de Goes, Xiaogang Jin, Hujun Bao, and Mathieu Desbrun. 2014. Space-time Editing of Elastic Motion Through Material Optimization and Reduction. *ACM Trans. Graph.* 33, 4, Article 108 (2014), 10 pages.

Mickaël Ly, Romain Casati, Florence Bertails-Descoubes, Mélina Skouras, and Laurence Boissieux. 2018. Inverse Elastic Shell Design with Contact and Friction. In *SIGGRAPH Asia 2018 Technical Papers (SIGGRAPH Asia '18)*. Article 201, 16 pages.

Antoine McNamara, Adrien Treuille, Zoran Popović, and Jos Stam. 2004. Fluid Control Using the Adjoint Method. *ACM Trans. Graph.* 23, 3 (2004), 449–456.

E. Miguel, D. Bradley, B. Thomaszewski, B. Bickel, W. Matusik, M. A. Otaduy, and S. Marschner. 2012. Data-Driven Estimation of Cloth Simulation Models. *Comput. Graph. Forum* 31, 2pt2 (2012), 519–528.

Aron Monszpart, Nils Thuerey, and Niloy J. Mitra. 2016. SMASH: Physics-guided Reconstruction of Collisions from Videos. *ACM Trans. Graph.* 35, 6, Article 199 (Nov. 2016), 14 pages.

Nigel J. W. Morris and Kiriakos N. Kutulakos. 2011. Dynamic Refraction Stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 8 (Aug. 2011), 1518–1531.

Kentaro Nagasawa, Takayuki Suzuki, Ryohei Seto, Masato Okada, and Yonghao Yue. 2019. Mixing Sauces: A Viscosity Blending Model for Shear Thinning Fluids. *ACM Trans. Graph.* 38, 4, Article 95 (July 2019), 17 pages.

Makoto Okabe, Yoshinori Dobashi, Ken Anjyo, and Rikio Onai. 2015. Fluid Volume Modeling from Sparse Multi-view Images by Appearance Transfer. *ACM Trans. Graph.* 34, 4, Article 93 (July 2015), 93:1–93:10 pages.

Dinesh K. Pai, Kees van den Doel, Doug L. James, Jochen Lang, John E. Lloyd, Joshua L. Richmond, and Som H. Yau. 2001. Scanning Physical Interaction Behavior of 3D Objects. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. 87–96.

Dinesh K. Pai, Austin Rothwell, Pearson Wyder-Hodge, Alistair Wick, Ye Fan, Egor Larionov, Darcy Harrison, Debanga Raj Neog, and Cole Shing. 2018. The Human Touch: Measuring Contact with Real Human Soft Tissues. *ACM Trans. Graph.* 37, 4, Article 58 (July 2018), 12 pages.

Andreas Peer, Markus Ihmsen, Jens Cornelis, and Matthias Teschner. 2015. An Implicit Viscosity Formulation for SPH Fluids. *ACM Trans. Graph.* 34, 4, Article 114 (2015), 10 pages.

A. Peer and M. Teschner. 2017. Prescribed Velocity Gradients for Highly Viscous SPH Fluids with Vorticity Diffusion. *IEEE Transactions on Visualization and Computer Graphics* 23, 12 (2017), 2656–2662.

N. Rasmussen, D. Enright, D. Nguyen, S. Marino, N. Sumner, W. Geiger, S. Hoon, and R. Fedkiw. 2004. Directable Photorealistic Liquids. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 193–202.

Zhimin Ren, Hengchin Yeh, and Ming C. Lin. 2013. Example-guided Physically Based Modal Sound Synthesis. *ACM Trans. Graph.* 32, 1, Article 1 (2013), 16 pages.

Jos Stam. 1999. Stable Fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. 121–128.

Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. 2013. A Material Point Method for Snow Simulation. *ACM Trans. Graph.* 32, 4, Article 102 (2013), 102:1–102:10 pages.

Alexey Stomakhin, Craig Schroeder, Chenfanfu Jiang, Lawrence Chai, Joseph Teran, and Andrew Selle. 2014. Augmented MPM for Phase-change and Varied Materials. *ACM Transactions on Graphics* 33, 4, Article 138 (2014), 11 pages.

Tetsuya Takahashi, Yoshinori Dobashi, Issei Fujishiro, Tomoyuki Nishita, and Ming C. Lin. 2015. Implicit Formulation for SPH-based Viscous Fluids. *Computer Graphics Forum* 34, 2 (2015), 493–502.

Tetsuya Takahashi and Ming C. Lin. 2019. A Geometrically Consistent Viscous Fluid Solver with Two-Way Fluid-Solid Coupling. *Computer Graphics Forum* 38, 2 (2019), 49–58.

Tetsuya Takahashi, Tomoyuki Nishita, and Issei Fujishiro. 2014. Fast simulation of viscous fluids with elasticity and thermal conductivity using position-based dynamics. *Computers & Graphics* 43 (2014), 21–30.

Christopher D. Twigg and Zoran Kačić-Alesić. 2011. Optimization for Sag-free Simulations. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '11)*. 225–236.

Kiwon Um, Xiangyu Hu, and Nils Thuerey. 2017. Perceptual Evaluation of Liquid Simulation Methods. *ACM Trans. Graph.* 36, 4, Article 143 (2017), 12 pages.

Orestis Vantzos, Saar Raz, and Mirela Ben-Chen. 2018. Real-time Viscous Thin Films. In *SIGGRAPH Asia 2018 Technical Papers (SIGGRAPH Asia '18)*. Article 281, 10 pages.

Bin Wang, Longhua Wu, KangKang Yin, Uri Ascher, Libin Liu, and Hui Huang. 2015. Deformation Capture and Modeling of Soft Objects. *ACM Trans. Graph.* 34, 4, Article 94 (July 2015), 12 pages.

Huamin Wang, Miao Liao, Qing Zhang, Ruigang Yang, and Greg Turk. 2009. Physically Guided Liquid Surface Modeling from Videos. *ACM Trans. Graph.* 28, 3, Article 90 (July 2009), 90:1–90:11 pages.

Huamin Wang, James F. O'Brien, and Ravi Ramamoorthi. 2011. Data-driven Elastic Models for Cloth: Modeling and Measurement. *ACM Trans. Graph.* 30, 4, Article 71 (2011), 12 pages.

Marcel Weiler, Dan Koschier, Magnus Brand, and Jan Bender. 2018. A Physically Consistent Implicit Viscosity Solver for SPH Fluids. *Computer Graphics Forum* 37, 2 (2018), 145–155.

Jinhui Xiong, Ramzi Idoughi, Andres A. Aguirre-Pablo, Abdulrahman B. Aljedaani, Xiong Dun, Qiang Fu, Sigurdur T. Thoroddsen, and Wolfgang Heidrich. 2017. Rainbow Particle Imaging Velocimetry for Dense 3D Fluid Velocity Imaging. *ACM Trans. Graph.* 36, 4, Article 36 (July 2017), 14 pages.

Hongyi Xu and Jernej Barbič. 2017. Example-based Damping Design. *ACM Trans. Graph.* 36, 4, Article 53 (July 2017), 14 pages.

Hongyi Xu, Yijing Li, Yong Chen, and Jernej Barbič. 2015. Interactive Material Design Using Model Reduction. *ACM Trans. Graph.* 34, 2, Article 18 (March 2015), 14 pages.

Guowei Yan, Wei Li, Ruigang Yang, and Huamin Wang. 2018. Inexact Descent Methods for Elastic Parameter Optimization. In *SIGGRAPH Asia 2018 Technical Papers (SIGGRAPH Asia '18)*. Article 253, 14 pages.

Shan Yang, Tanya Ambert, Zherong Pan, Ke Wang, Licheng Yu, Tamara Berg, and Ming C. Lin. 2016. Detailed Garment Recovery from a Single-View Image.

Shan Yang, Junbang Liang, and Ming C. Lin. 2017. Learning-based Cloth Material Recovery from Video. In *2017 IEEE International Conference on Computer Vision*.

S. Yang and M. C. Lin. 2016. MaterialCloning: Acquiring Elasticity Parameters from Images for Medical Applications. *IEEE Transactions on Visualization and Computer Graphics* 22, 9 (Sept 2016), 2122–2135.

Yonghao Yue, Breannan Smith, Christopher Batty, Changxi Zheng, and Eitan Grinspun. 2015. Continuum Foam: A Material Point Method for Shear-Dependent Flows. *ACM Trans. Graph.* 34, 5, Article 160 (2015), 160:1–160:20 pages.

Yonghao Yue, Breannan Smith, Peter Yichen Chen, Maytee Chantharayukhonthorn, Ken Kamrin, and Eitan Grinspun. 2018. Hybrid Grains: Adaptive Coupling of Discrete and Continuum Simulations of Granular Media. In *SIGGRAPH Asia 2018 Technical Papers (SIGGRAPH Asia '18)*. Article 283, 19 pages.

Guangming Zang, Ramzi Idoughi, Ran Tao, Gilles Lubineau, Peter Wonka, and Wolfgang Heidrich. 2019. Warp-and-Project Tomography for Rapidly Deforming Objects. *ACM Trans. Graph.* (2019).

Bo Zhu, Minjae Lee, Ed Quigley, and Ronald Fedkiw. 2015. Codimensional non-Newtonian Fluids. *ACM Trans. Graph.* 34, 4, Article 115 (2015), 9 pages.

Z. Zivkovic. 2004. Improved adaptive Gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 2. 28–31 Vol.2. https://doi.org/10.1109/ICPR.2004.1333992