Real-time High-fidelity Surface Flow Simulation

Bo Ren, Tailing Yuan, Chenfeng Li, Kun Xu, and Shi-Min Hu,

Abstract—Surface flow phenomena, such as rain water flowing down a tree trunk and progressive water front in a shower room, are common in real life. However, compared with the 3D spatial fluid flow, these surface flow problems have been much less studied in the graphics community. To tackle this research gap, we present an efficient, robust and high-fidelity simulation approach based on the shallow-water equations. Specifically, the standard shallow-water flow model is extended to general triangle meshes with a feature-based bottom friction model, and a series of coherent mathematical formulations are derived to represent the full range of physical effects that are important for real-world surface flow phenomena. In addition, by achieving compatibility with existing 3D fluid simulators and by supporting physically realistic interactions with multiple fluids and solid surfaces, the new model is flexible and readily extensible for coupled phenomena. A wide range of simulation examples are presented to demonstrate the performance of the new approach.

Index Terms—shallow-water equation, flow on curved surfaces, finite volume method, coupled simulation

1 Introduction

Fluid phenomena play a major role in the visual appearance of the real world. Over the past decade, a number of fluid simulators have been developed in the graphics community, which can produce a range of visually appealing scenes driven by fluid flow in the space, e.g. rising smoke from a cigarette, wavering blaze of candle light, splashy motion of water pouring into a cup, etc. Despite the great success in capturing the spatial fluid flow, much less attention has been paid to the surface flow phenomena. The most common fluids in daily life are air and water, both of which are transparent. In numerous cases, the fluid motion becomes visible or more interesting only when it is constrained by a solid surface. Contrary to volumetric fluid motions that feature bulk motions and diverse free-surface details, surface flows are usually thin and their visual variations rely on the flowing patterns that are strongly influenced by the interaction with the underlying solid surface. Besides such obvious examples as rain water on windscreen and road flooding, the surface flow is almost always observed whenever and wherever water interacts with a surface, e.g. taking a shower, painting on a wall, and frying with oil etc. The common occurrence of surface flow phenomena highlights the demand of systematic research in surface flow modelling. In this paper, we propose a real-time high-fidelity simulation scheme that models surface flows directly on general triangle meshes, taking into account a full spectrum of physics relevant to the visual appearance of surface flow.

The problem of surface flow can be solved, at least in principle, by the 3D Navier-Stokes equations with boundary conditions appropriately enforced on the surface geometry. But it is often difficult, if not impossible, for a 3D simulator to deal with detailed mesh surfaces since the resolution required to capture fine geometry features and thin-layer flow motions can be very high, regardless of whether a grid-based or a particle-based simulator is employed. Meanwhile, under certain conditions and assumptions, the 3D Navier-Stokes equation can be transformed into some dedicated 2D equations without losing the essential 3D flow information. Examples of this include the shallow-water flow and the lubrication models, both of which assume the fluid domain has one dimension that is significantly smaller than the others. In these models, the surface flow can be represented by a height field associated to the surface mesh and subsequently the problem can be solved in a two-dimensional space, significantly reducing the computational complexity.

Despite the potential benefit, it poses a number of new challenges to truthfully capture complex surface flow on general 3D geometries with these simplified 2D flow models. First, the standard shallow-water flow and the lubrication models both have strict assumptions, e.g. a smooth flat/spherical surface or a high fluid viscosity, while real-world surface flows often happens on arbitrarily curved surfaces and can be non-viscous. Secondly, the visual effects of real-world surface flow are complex and sensitive to the surface geometry and the flow status. Liquids tend to move along wrinkles instead of moving across them, and they flow slower at rough regions than at smooth regions. Liquids also flow more easily if they are with a larger amount or at a larger velocity. These complex flow effects require additional mathematical and numerical treatment to faithfully reproduce and achieve plausible and stable simulation. Finally, the surface flow and other fluid phenomena, such as the 3D spatial flow and multiple fluids, are often coupled involving various interactions with the solid surface, which further complicates the surface flow simulation.

- *Bo Ren is with the College of Computer and Control Engineering, Nankai University, Tianjin.*
  E-mail: rb@nankai.edu.cn
- *Chenfeng Li is with the College of Engineering of Swansea University, UK.*
  E-mail: c.f.li@swansea.ac.uk
- *Tailing Yuan, Kun Xu are with the Department of Computer Science and Technology, Tsinghua University, Beijing.*
  Email: xukun@tsinghua.edu.cn
- *Shi-Min Hu, is with the Department of Computer Science and Technology, Tsinghua University, Beijing and School of Computer Science and Informatics, Cardiff University, UK. Shi-Min Hu is the corresponding author.*
  Email: shimin@tsinghua.edu.cn

There have been some good research works in surface flow simulation (e.g. [1], [2], [3]), but the above challenges are not fully resolved. Shallow Water Equations (SWE) have been solved with an implicit scheme on triangle meshes in [1], but the simulation only considered the hight field construction with varying bottom heights, hence insufficient to recover natural liquid flowing along ridges and creases. Recognizing this weakness, a viscous thin-film approach [3] was recently proposed for surface flow on curved surfaces, but the method is constrained to low Reynolds numbers and is unsuitable for finite-speed non-viscous flow.

In this paper we systematically address the aforementioned challenges to reproduce realistic surface flow phenomena over arbitrary triangle meshes. The surface flow is represented as a 2D height field on the associated 3D triangle mesh and is solved using an extended SWE model. In order to capture non-viscous flow motion along edges and creases over detailed 3D meshes, a novel feature-based friction model compatible with finite liquid speed is derived. Supporting the friction model, an efficient geometric computing scheme is derived to transform the triangle mesh information into a data structure that suits for simulation procedures, and it also supports liquid advection along triangle meshes and uniform surface tension calculation over arbitrary surfaces. Enhanced simulation performance on downward surfaces is achieved by taking advantage of the source term, which in turn provides a straightforward integration with existing 3D simulators. Wind blowing effect and a novel strategy recovering surface eroding and dissolving are also readily captured within our algorithm scheme, with flexible support for artistic design. An explicit scheme to solve the SWE on a triangle mesh is chosen for simplicity and better parallel computing performance, allowing our simulation to run at real-time speed on a consumer GPU platform. The main contributions can be described as follows:

- A novel feature-based friction model that is of critical importance for capturing realistic surface-flow patterns on sophisticated solid surfaces, along with an efficient geometric computing scheme for surface-flow simulation on arbitrary curved surfaces.
- A systematic, real-time solution for non-viscous surface-flow phenomena on detailed meshes using the extended SWE, with support for various physical effects including convective advection, surface tension, wind blowing, and two-phase dissolving etc.
- An extensible algorithm framework that is flexible for artistic design and able to perform coupled simulation with existing 3D simulators.

2 Previous Work

Various previous researches have involved fluid motion within surfaces. The two-dimensional flow within surfaces of arbitrary topology can be solved using a 2D texture space on the 3D surface [4]. Triangle meshes are also used for 2D simulation with a local flattening technique [5]. Later, a well-parallelizable method adopting the Lattice Boltzmann Model (LBM) combined with the finite-volume method is proposed for efficient calculation of 2D fluid motion in a triangle mesh [6]. Different aspects of 2D fluid simulation have been studied and enhanced since the early works, such as preserving discrete circulation avoiding numerical diffusion [7] and extending the 2D fluid simulation to deformable surfaces [8]. Using the Closest Point Method, a 2D flow field is extended into 3D space near a small neighborhood of the surface and solved by standard 3D numerical schemes [9]. GPU friendly algorithms for 2D fluid simulation also begin to receive attention [9], [10]. Recently, fluid equations are solved on the surface of a sphere under spherical coordinates [11]. These techniques produce 2D flow effect on 2D manifolds and are able to generate fluid patterns visually similar to that of a 3D fluid simulation projected on a 2D surface.

On the other hand, the on-surface flow, where the liquid has finite thickness over the surface, is much less studied by the graphics community. Shallow-water equations (SWE) are simplified, by ignoring the convective term, and solved with implicit schemes to obtain surface flow and wave propagation effects over curved surfaces [1], [2]. Recently, based on the lubrication theory and the gradient flow model, the motion of a thin viscous film on a curved surface is simulated [3] using an optimization approach, producing realistic thin film appearance. None of the aforementioned methods works for general surface flow with unconstrained velocity profile and full surface interaction between liquid and detailed surface features.

The SWE method has been combined with 3D simulation methods using grids or particles [12], [13]. However these previous works assumed stable SWE simulation on upward surfaces and did not consider downward surfaces. In this paper we derive an integration scheme of SWE simulator with 3D Smoothed Particle Hydrodynamics (SPH) [14] simulator and it provides a stable solution over arbitrarily oriented 3D triangle mesh surface, where the conservation laws are satisfied in the transition between simulators.

There are other researches that are relevant to our work. SWE is introduced to the graphics community for animating water waves [15], [16] with implicit solvers. An explicit scheme can also be adopted subject to time step constraints [17]. The most popular particle-based 3D simulators for graphics applications are based on the SPH method. Better incompressibility under larger time steps can be achieved following later researches over Weakly Compressible SPH (WCSPH) [18], Position Based Dynamics methods [19] or Divergence Free SPH (DFSPH) method [20]. Reproducing multi-fluid phenomena in explicit 3D simulators has been studied in recent works [21], [22], [23], [24]. Scalar and vector function interpolation over unstructured data sets such as triangle meshes and point sets has been performed under local coordinate with assistance of affine mappings and local flattening, where semi-Lagragian advection strategies are also provided [5], [25]. Surface tension and contact angle problems have been studied for water drops and for free surface flows, based on the local surface curvature [26], [27] or the surface tension energy [28]. Recently, a surface-only simulation technique is proposed to handle 3D liquid with surface tension [29]. Some existing works on the solid-fluid phase change (e.g. [30], [31]) are also relevant to our two-phase coupled simulation scheme to some extent.

3 Shallow-water Equations on a Triangle Mesh

The SWE model was previously adopted to triangle meshes in the absence of the convection, the source and the friction terms [1], [2]. In this section we briefly summarize the extended SWE employed in the proposed approach.

In the SWE model, the liquid motion is described in 2D space using the planar velocity $\mathbf{u}$, the water surface height h, the bottom height position b, and the water column depth $d = h - b$. The conservation of mass and momentum is described as [32]:

$$\frac{\partial d}{\partial t} + \mathbf{u} \cdot \nabla d = -d(\nabla \cdot \mathbf{u}) + Qd \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{\nabla P}{\rho} + \frac{\mathbf{\Gamma}}{\rho} + \mathbf{a}_{ext} \tag{2}$$

where Q is the volumetric source added per unit volume per unit time, ρ is the density of fluid, P is the total pressure measured as $P = \rho g h + P_s$, in which P_s is the pressure due to surface tension force. $\mathbf{\Gamma} = \mathbf{\Gamma}_w - \mathbf{\Gamma}_b$ is the friction force exerted on the liquid, where $\mathbf{\Gamma}_w$ refer to the driving force due to wind blowing and $\mathbf{\Gamma}_b$ the drag force at the solid surface. $\mathbf{a}_{ext}$ is the acceleration due to external body forces or motion of reference frame.

For engineering applications, the SWE model holds for flow on a planar bed with mild slope, such as river, sea and atmosphere where the depth dimension is much smaller than the horizontal dimensions. The above equations can be readily extended to represent liquid flow on triangle meshes, where the liquid is defined as liquid depth d in the outer normal direction of each mesh vertex, and the flow velocity is defined on the tangential plane of the vertex. Due to mesh surface curvature, the gravity acceleration is now split into a normal part g_n and a tangential part g_t, contributing to the pressure term $P = \rho g_n h + P_s$ and the external acceleration term $a_{ext} = g_t$ respectively. To adapt the SWE model on curved surfaces, the underlying assumptions are 1) the flow velocity does not vary along the normal direction of the surface and 2) the pressure is "hydrostatic" subject to the normal component of the gravity.These two conditions hold approximately for real-world surface flow, depending on the liquid and surface properties.

The extended SWE (1-2) consists of a set of mathematical terms, each corresponding to a different physical effect of real world surface flow. Largely omitted by previous studies, the friction term $\mathbf{\Gamma}$ represents bottom friction and wind blowing effects, which recovers such commonly observed surface flow effects as pushing by winds, following local seams or edges, moving more freely at smooth regions, and slowing down on rough surfaces. The convection term recovers spatial transport due to motion of flow, which is only neglectable in viscous low-speed models. The hydrostatic pressure and the pressure due to surface tension are captured in the pressure term P. The source term Qd captures source/sink effects, such as rain drops falling continuously onto the surface and liquid leaving the surface from the downward faces. The slope of non-horizontal bottom can be captured in the external body force term $\mathbf{a}_{ext}$, where the tangential part of gravity acceleration is considered.

The following sections are arranged as follows. In §4, a feature-based friction model is derived along with other drag forces for the friction term, which can respond to detailed mesh features of general 3D models. In §5, we explain the numerical schemes for the convective term, surface tension and source term on triangle meshes, all of which are designed to be suitable for parallel acceleration. In §6, the proposed SWE method is further extended to support coupled simulation on triangle meshes and two-phase simulation for artistic effects. In §7 we explain the implementation of the proposed method and summarize the algorithm framework. A number of examples are presented in §8 to demonstrate the performance of the new method. Finally, concluding remarks including limitations are discussed in §9.

4 Friction Effect on Surface Flow

Due to the thin nature of surface flow, friction forces have a dominant effect on its flow pattern and significantly affect its visual appearance, especially when the underlying solid surface has fine features. In this section we discuss in detail the friction term, where a novel feature-based friction model is proposed for high-fidelity surface flow simulation.

4.1 Feature-based Friction Model

To use height fields on triangle meshes, it is often assumed that the mesh is locally planar and the local normals have a relatively good alignment, otherwise the original mesh is smoothed (e.g. by Laplacian smoothing) before simulation [1], [2], [3]. This is to achieve simulation stability and to avoid self-intersection of the height-field-represented liquid surface at concave features, which not only causes artifacts in visual appearance but also violates the conservation law of physics. However, the mesh smoothing essentially eliminates the fine surface features which have direct and significant effects on surface flow pattern in the real world. Even if the difference between the original surface and the smoothed surface can be represented as a non-constant terrain height field [1], it is still insufficient to reproduce realistic surface flow effects, especially along creases and edges [3], and the information loss of fine surface details is almost inevitable when a low-resolution mesh is used as the smoothed mesh. Unfortunately, as long as the mesh geometry has features of the same level of curvature, this dilemma between simulation feasibility and visual plausibility cannot be easily resolved simply by increasing mesh resolution.

To overcome the above challenge, we propose a feature-based friction model that can respond to detailed local surface features of the original mesh during the simulation on a smoothed mesh. Intuitively the friction force should relate to high frequency variations of local features: a more significant friction effect is expected along the direction with larger variance, and vice versa. In the light of this insight, we link the bottom friction to the local roughness, such that the friction coefficients differ in different directions depending on the local anisotropic roughness.

For simplicity, it is assumed in the following discussions that a smoothed mesh has been obtained, and the scalar (height, etc.) and vector (velocity, etc.) functions in Eqns. (1-2) are defined on vertices of the smoothed surface. Also,

the "tangential plane" is associated to the smoothed mesh unless explicitly specified. The effects of detailed mesh features on the original mesh are modelled by a variable anisotropic bottom friction on the smoothed mesh, which is defined via two computational steps: the shape operator on a semi-flattened original mesh and the bottom friction based on a friction tensor.

4.1.1 Shape operator on a semi-flattened original mesh

The smoothed mesh contains only the low-frequency curvature information, but the friction effect mostly relates to local high-frequency variations. For example, there is no frictional effect on a perfectly spherical surface, although its curvature is nonzero. Thus, to describe the friction effect, it is desirable to eliminate the low-frequency curvature information from the original mesh and build a semi-flattened original mesh (SOM) that contains only the high-frequency geometry information. A 1D illustration is given in Fig. 1, where the solid blue line represents the original mesh, the solid red line the smoothed mesh, the dashed blue line the SOM, and the dashed red line the flattening of the smoothed mesh. The associated low-frequency curvature information is removed by a flattening operation on the smoothed mesh, which we will discuss in detail in §5.1, then the same amount of adjustment is applied to the original mesh and the SOM is constructed without the low-frequency information. Specifically, for a point v on the smoothed mesh and along the normal direction of v, its counterpart v_p can be located on the original mesh. Then, for a nearby point v'_p on the original mesh and along the same normal direction of v, we can trace back the corresponding point v' on the smoothed mesh. The parallel lines vv_p and $v'v'_p$ form a plane, and v' can be considered to have been rotated by an angle θ around v within this plane during the flattening step. Finally, we rotate v'_p around v_p within this plane by the same angle θ. For a given point v on the smoothed mesh, its SOM patch is formed by repeating the same procedure for all nearby vertices v'_p on the original mesh.

The shape operator tensor S_i on each triangle surface i of the SOM can then be obtained by calculating the tangential gradient of the local normals [3]. Alternatively, it can be calculated by averaging the triangle's three vertices, whose principal curvature values and directions can be calculated from least-squares fitting on the mesh [33]. In our experiments the former method gives more robust estimates on triangle meshes.

4.1.2 Bottom friction based on a friction tensor

The bottom friction force $\mathbf{\Gamma_b}$ is a viscous drag effect acting on the liquid at the solid surface, which is related to the local roughness, viscous friction intensity and liquid velocity [32]. The local roughness has a direct influence on the friction force, such that a higher friction effect is expected along the direction with higher variation, and vice versa. This requires an anisotropic friction model. Physically, in the isotropic case where bottom friction force direction aligns with velocity direction, one can model bottom friction force as product of a scalar friction coefficient and relative velocity: $\mathbf{\Gamma_b} = \mu\mathbf{u}$, where μ is the friction intensity. In the anisotropic case where the strength of frictional effect differs along different directions, the direction of bottom friction force can differ

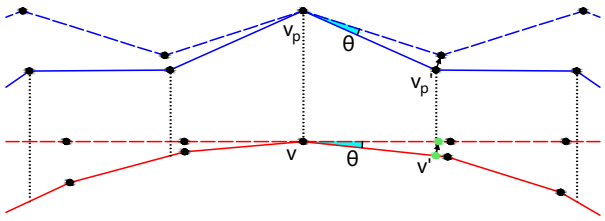


Fig. 1. Generating semi-flattened original mesh (SOM) in 1D space. The solid red line represents the smoothed mesh, the dashed red line represents its flattening, the solid blue line represents the original mesh, and the dashed blue line represents the SOM. The SOM, which represents the high-frequency mesh variation, is obtained by rotating the nearby vertices on the original mesh in the same way as flattening the corresponding vertices on the smoothed mesh.

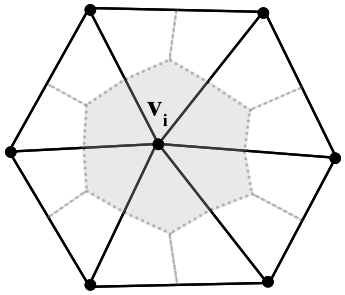


Fig. 2. Control area of vertex v_i on a triangle mesh. The light grey region surrounded by dashed lines is v_i's control area.

from the velocity direction. In such case a natural extension is to replace the scalar friction intensity in the isotropic case by a friction tensor, i.e. $\mathbf{\Gamma_b} = \mathbf{T} \cdot \mathbf{u}$.

The simplest friction tensor takes the form of a diagonal matrix. In such case, the fluid flow aligned with x-axis or y-axis (these directions are the friction tensor's principal directions) experiences a friction effect proportional to the first or second diagonal element (principal value) of the matrix. For the general case, given a triangle mesh, for each vertex in its local coordinate system, we assume that the direction with largest local geometry variation has the largest friction effect along it, and take this direction and its orthogonal direction as the principal directions of the friction tensor. The average roughnesses along these two directions determine the principal values, and the detailed formulation is explained below.

Given a vertex v on the smoothed mesh, the maximum average roughness for v's neighbourhood is defined as

$$\tau_{max} = \max_{\forall \mathbf{x}} \{ \frac{\sum_{i \in I_v} |\mathbf{x}^T \mathbf{S}_i \mathbf{x}| A_i}{\sum_{i \in I_v} A_i} \} \tag{3}$$

where $\mathbf{x}$ denotes any direction vector within the tangential plane of v (i.e. $|\mathbf{x}| = 1$ and $\mathbf{x} \perp \mathbf{n}_v$), I_v consists of all triangles on the SOM that has overlap with the projected control area of v (i.e. the control area of v projected along v's normal onto SOM), and A_i and $\mathbf{S}_i$ are the area and shape operator of triangle i. Let $\bar{\mathbf{x}}$ denote the direction along which τ_{max} is achieved, and $\bar{\mathbf{x}}_\perp$ denote its perpendicular direction. Similarly, the average roughness along the perpendicular direction $\bar{\mathbf{x}}_\perp$ can be defined as

$$\tau_\perp = \frac{\sum_{i \in I_v} |\bar{\mathbf{x}}_\perp^T \mathbf{S}_i \bar{\mathbf{x}}_\perp| A_i}{\sum_{i \in I_v} A_i}. \tag{4}$$

In Eqn. (3), $\mathbf{x}^T\mathbf{S}\mathbf{x}$ is the curvature value along the direction $\mathbf{x}$ at a location with a shape operator $\mathbf{S}$. Thus $\bar{\mathbf{x}}$ is the direction with the largest local high-frequency variation, along which the average roughness is τ_{max}; $\bar{\mathbf{x}}_\perp$ is the perpendicular direction, along which the average roughness value is $\tau_\perp$.

Suppose $(\bar{\mathbf{x}}, \bar{\mathbf{x}}_\perp) = (\mathbf{e}_s, \mathbf{e}_t)\mathbf{M}$, where $\mathbf{e}_s, \mathbf{e}_t$ are the local coordinate base vectors of the tangential plane of vertex v and $\mathbf{M}$ is an affine matrix. The friction tensor can be defined as

$$\mathbf{T} = \mu\mathbf{M}diag\{\tau_{max}, \tau_\perp\}\mathbf{M}^{-1} \tag{5}$$

where μ is the friction intensity. Using the above friction tensor, the anisotropic bottom friction force for liquid flowing at velocity $\mathbf{u}$ can be formally expressed as

$$\mathbf{\Gamma}_b = \mathbf{T} \cdot \mathbf{u}. \tag{6}$$

For convenience in Eqn. (2), we define $\gamma = \mu/\rho$ as the intensity coefficient instead of separately using μ and ρ values.

It is worth noting that if $\tau_{max} = \tau_\perp$, Eqn.(6) is equivalent to $\mathbf{\Gamma}_b = \mu\mathbf{I} \cdot \mathbf{u} = \mu\mathbf{u}$, degenerating to the isotropic case [32].

Beside responding to fine details of the solid surface, the proposed feature-based friction model also provides flexibility for artistic design. For example, the friction tensor can be alternatively defined by the color gradient of a texture map applied on a smooth surface. One can also directly alter the friction tensor for mesh surfaces. By tailoring the friction tensor, artists can control the direction and motion of surface flow.

Intuitively, following the above friction model, the high resolution curvature information can be computed and stored, serving as a "texture map" of the frictional influence on a smoothed surface. The smoothed mesh is in turn much less likely to cause self-intersection issues.

4.2 Other Drag Forces

The wind drag effect is modeled following physical convention by a quadratic drag term as [32], [34]:

$$\frac{\mathbf{\Gamma}_w}{\rho} = \gamma_w \mathbf{u}_w |\mathbf{u}_w| \tag{7}$$

where γ_w is the wind drag coefficient, $\mathbf{u}_w$ is the wind speed.

It is also possible to introduce other kind of drag forces in the friction term. For example, if the liquid carries small dust with it, a quadratic term determined by relative velocity between dust and liquid can be added to the equation, we will discuss this in detail in §6.

5 Convective, Surface Tension and Source Terms

In this section we explain in detail how to evaluate on a triangle mesh the convective, surface tension and source terms in Eqns. (1-2). Compared to the friction term, the influence from these terms onto the surface flow pattern is indirect and more subtle, but all these terms must be properly treated to achieve stable and efficient simulation for the full range of surface flow phenomena.

5.1 Convective Term and Differential Operator

The feature-based bottom friction responds to finite velocity of the surface flow and does not make low-velocity assumption as in previous methods (e.g. [1]). Thus, the nonlinear convective terms in Eqns.(1-2), which reflect the influence from inertia due to local fluid motion, cannot be ignored as in those methods. In 3D fluid simulators these terms can be solved by a semi-Lagrangian advection step, or by finite-volume techniques using the integral form of the equations [17], [35]. Although it seems a natural option to adopt the standard finite-volume techniques for our purpose, due to the thin nature of surface flow this strategy does not yield stable simulation with large time steps. To enhance the stability for surface flow simulation under large time steps, we adopt the semi-Lagrangian approach on triangle meshes.

The challenges here are to estimate the start position on an un-structured triangle mesh during the semi-Lagragian advection and to interpolate accurately the associated vector field values. A brute-force trial is to sequentially trace the triangles on the velocity path, but doing so is computationally slow. Inspired by the recent work [25], we propose an efficient approach to directly compute the start position values based on an exponential map and a local affine matrix.

When calculating the advected value of a vertex i, we flatten an n-ring neighborhood of this vertex onto its tangential plane similar to the exponential map proposed in [25]. This preserves the geodesic distance of nearby surface points to i, and its effectiveness in scalar advection has been analyzed in the original paper. Then, on the flattened n-ring neighborhood in the tangential plane, we directly identify the specific triangle in which the start position falls in and interpolate the function values there. This approach also allows Runge-Kutta methods and other higher-order advection schemes to be straightforwardly performed during the advection step.

The interpolation of scalar functions such as pressure and liquid height etc. can be carried out using the standard triangular interpolation. For the 2D velocity vector, since a triangle mesh does not have a consistent coordinate system on the surface, we extend the vertex-based affine matrix [36] to the whole n-ring neighborhood. First, a 3D local coordinate system is defined on each vertex, combining a local tangential 2D coordinate in the tangential plane and the normal direction. We then calculate a 2×2 affine matrix M_{ij} from the vertex j's local tangential coordinate to the vertex i's local tangential coordinate in each i-neighborhood using the following steps:

- Suppose vertex i and vertex j have outer normals $\mathbf{n}_i$ and $\mathbf{n}_j$, rotate the 3D coordinate of vertex j around the direction $\mathbf{n}_i \times \mathbf{n}_j$ so that $\mathbf{n}_j$ aligns with $\mathbf{n}_i$.
- Now the tangential planes of the two vertices are parallel, and their current affine matrix from vertex j's rotated tangential coordinate to vertex i's tangential coordinate is M_{ij}.

For velocity interpolation in the flattened neighborhood of vertex i, we can directly use the $M_{ij}\mathbf{v}_j$ value for calculation. This strategy effectively extends the semi-Lagrangian advection on discrete exponential maps [25] to vector functions, and it avoids the discontinuity problem near mesh edges

which often requires special attention [5]. It also provides a small runtime performance benefit compared to a naive route tracing across the traingles. Namely after finding the triangle containing the original point of advection (say using the naive tracing in [5] or parallely traverse the neighboring triangles), a simple direct interpolation of affined velocity will be sufficient instead of sequentially calculating each affine transformation and assuring velocity continuity across mesh edges on a traced route.

It is critical to evaluate the differential operators in Eqns. (1-2) in order to solve the SWE model on a triangle mesh. In a finite volume approach [37], the gradient and divergence operators at a given vertex i are evaluated as follows:

$$(\nabla f)_i = \frac{1}{A_i}\sum_{\triangle_j}(\nabla f)_{\triangle_j} A_{\triangle_j} \tag{8}$$

$$(\nabla \cdot \mathbf{F})_i = \frac{1}{A_i}\sum_{\triangle_j}(\nabla \cdot \mathbf{F})_{\triangle_j} A_{\triangle_j} \tag{9}$$

where the summation is over all adjacent triangles $\triangle_j$ that share vertex i, A_i is the barycentric control area of vertex i (Fig.2), $A_{\triangle_j}$ is its part in the triangle $\triangle_j$. The gradient $(\nabla f)_{\triangle_j}$ and the divergence $(\nabla \cdot \mathbf{F})_{\triangle_j}$ are calculated on the triangle face $\triangle_j$ using the finite volume approach, where vector values are properly modified by the above affine method.

However, for the $d(\nabla \cdot \mathbf{u})$ term on the right-hand side of Eqn. (1), Eqn. (9) does not provide a symmetrical formulation if simply multiplied by d_i. Instead we use the following modification:

$$(d(\nabla \cdot \mathbf{u}))_i = \frac{1}{A_i}\sum_{\triangle_j}\bar{d}_{\triangle_j}(\nabla \cdot \mathbf{u})_{\triangle_j} A_{\triangle_j} \tag{10}$$

where $\bar{d}_{\triangle_j}$ is the average d on triangle $\triangle_j$, i.e. the average of its three vertices.

5.2 Surface Tension

The surface tension pressure P_s can be obtained according to either force equilibrium or energy conservation, and the two strategies are equivalent in physics. For data structure consistency and implementation convenience, we take an energy approach similar to [28] and the detailed formulation is explained below. It is noted that alternative schemes based on force equilibrium can also be adopted, where the surface tension pressure is proportional to the local mean curvature (see e.g. [1], [26]).

The surface energy E_s of a liquid surface is simply given as

$$E_s = \sigma A \tag{11}$$

where σ is the coefficient of surface tension, A is the total surface area. On a triangulated liquid surface, Eqn. (11) becomes

$$E_s = \sum \sigma A_{\triangle} \tag{12}$$

where $A_{\triangle}$ is any triangle face of the liquid surface. As the liquid flow is represented as a height function on the triangle mesh, the liquid surface mesh can be recovered with the vertices off-set by its height value along the normal direction.

The surface tension at a given point v on the liquid surface is:

$$\mathcal{F}_{s,v} = -\nabla_v E_s = -\sum \sigma \nabla_v A_{\triangle}. \tag{13}$$

Specifically, if v is at a vertex position, then Eqn. (13) becomes

$$\mathcal{F}_{s,v} = -\sum_{\triangle_j} \sigma \nabla_v A_{\triangle_j}. \tag{14}$$

For a triangle with vertices v, v_1, v_2 in counter-clockwise order, the three components of the $\nabla_v A$ term are:

$$\frac{\partial A}{\partial x} = \frac{1}{2}[(y_1 - y_2)cos\theta_{xy} + (z_2 - z_1)cos\theta_{zx}] \tag{15}$$

$$\frac{\partial A}{\partial y} = \frac{1}{2}[(z_1 - z_2)cos\theta_{yz} + (x_2 - x_1)cos\theta_{xy}] \tag{16}$$

$$\frac{\partial A}{\partial z} = \frac{1}{2}[(x_1 - x_2)cos\theta_{zx} + (y_2 - y_1)cos\theta_{yz}] \tag{17}$$

where x, y, z are the coordinate components of the vertices, θ_{xy}, θ_{yz}, θ_{zx} are the plane angle between the triangle and the xy, yz, zx planes.

The surface tension pressure can then be calculated as

$$|P_s| = \frac{|\mathcal{F}_{s,v}|}{\sum A_{\perp}} \tag{18}$$

where $A_{\perp}$ is the area of $A_{\triangle_j}$ projected onto the plane perpendicular to the mesh normal at vertex v. Eqn. (18) is used to calculate the surface tension pressure at inner vertices (i.e. $d > 0$ vertices). For the liquid front, i.e. $d = 0$ or $d < 0$ (due to depth extrapolation discussed in detail in §7.3) vertices that directly link to a $d > 0$ vertex by a mesh edge, the scheme needs to be extended as follows.

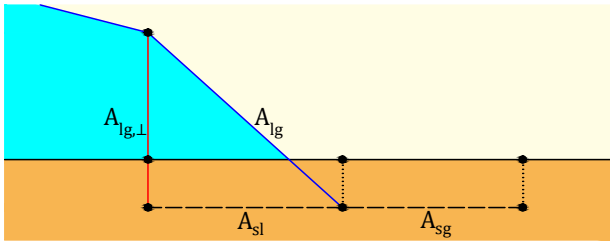


Fig. 3. A 1D illustrative diagram for Eqn. (20). Black line indicates the surface mesh flattened to the tangential plane; blue line is the liquid surface; red line indicates the area which is perpendicular to the normal of liquid front line in the tangential plane. Small black dots are vertices on the solid-surface and liquid-surface meshes.

On the liquid front the shape of the interface is described by the Young's equation [38]:

$$\sigma_{lg}cos\theta_c + \sigma_{sl} = \sigma_{sg} \tag{19}$$

where θ_c is the contact angle, σ_{lg}, σ_{sl}, σ_{sg} are coefficients of surface tension on the liquid and air interface, the liquid and solid interface, and the solid and air interface.

Based on Young's discussion [38], the following relations can be derived: $\sigma_{lg} = \sigma$, $\sigma_{sl} = \frac{1-cos\theta_c}{2}\sigma$, $\sigma_{sg} = \frac{1+cos\theta_c}{2}\sigma$. They also relate to interface energy in the neighborhood of the contact point. As the water front can only move in the tangential plane, the surface tension at a vertex v can be represented as:

$$\mathcal{F}_{bs} = -\nabla_{xy}(\sigma_{lg}A_{lg} + \sigma_{sl}A_{sl} + \sigma_{sg}A_{sg}) \tag{20}$$

where ∇_{xy} denotes derivative in the tangential xy plane, A_{lg} is the triangle area on the liquid surface having vertex v, A_{sl} is the triangle area on the liquid-solid interface having vertex v, and A_{sg} is the triangle area on the solid-air

interface having vertex v. It can be verified that $\mathcal{F}_{bs}$ vanishes in the ideal case that the actual plane angle between A_{lg} and A_{sl} is θ_c. A 1D diagram is shown in Fig. 3.

Then the pressure due to $\mathcal{F}_{bs}$ is

$$|P_{bs}| = \frac{|\mathcal{F}_{bs}|}{\sum A_{lg,\perp}} \tag{21}$$

where $A_{lg,\perp}$ is the area of A_{lg} projected to the plane which is perpendicular to the normal of liquid front line in the tangential plane. The sum consists of all A_{lg} triangles containing the vertex v at which P_{bs} is calculated.

Eqn. (21) is used to calculate the surface tension pressure at the liquid front. We calculate Eqns. (20-21) with the surface mesh flattened to the tangential plane of a vertex (see §5.1) and the liquid depth d kept unchanged on each nearby vertex. To maintain stability in large time steps, we also bound P_{bs} in the rare situation where $A_{lg,\perp}$ is close to zero.

The signs of surface pressure values P_s and P_{bs} are determined by whether the surface is concave or convex. This can be easily achieved by examining the sign of dot product of the surface tension force and the outer normal. Alignment of the surface tension force and the outer normal implies a concave surface and the pressure values should be negative, and vice versa.

5.3 The Source Term

The second term on the right-hand side of Eqn. (1) contains a volumetric source Q. When $Q > 0$, the term corresponds to a source, i.e. liquid added to the surface flow; when $Q < 0$, it corresponds to a sink, i.e. liquid removed from the surface flow. Suppose the volume change within a time step due to the source or sink is V, then $Q = \frac{V}{Ad\Delta t}$, where A is the control area of the mesh vertex. So the source term can be rewritten as $\frac{V}{A\Delta t}$. In other words, if liquid volume changes by V due to source or sink effect within a time step Δt, the influence to liquid depth can be physically computed by the source term. Through the source term, we are able to improve the stability of SWE simulation on triangle meshes, as well as enable integration with existing 3D simulators, which will be discussed in detail in §6.1.

6 Coupled Phenomena

6.1 Discharging and Charging of Surface Flow

In Eqn. (2), when the local outer normal direction is pointing downwards, g_n will be negative and cause instability to the SWE calculation. The fact is that on upward surfaces (positive g_n), larger depth leads to larger positive pressure, driving liquid away and reducing depth. On downward surfaces (negative g_n), larger depth leads to larger negative pressure, attracting liquid and increasing depth, hence causing instability. This is not a numerical issue on divergence. The surface tension pressure can alleviate the problem but is unable to completely remove the instability. From a physical point of view, the above "instability" corresponds to the fact that liquid will form drops and discharges from the solid surface if enough amount is accumulated in these downward-facing areas, which is out of the capacity of a pure SWE simulator.

We propose to combine the SWE simulation on triangle meshes with a WCSPH simulator. When a liquid depth value exceeds a user-defined threshold, we allow a certain amount of liquid to detach from the surface. First, the volume and position of the discharge are calculated. Then, after removing the discharged liquid from the SWE simulator, new liquid particles are correspondingly added to the particle simulator. To satisfy the physics, the changes of liquid volume in both simulators are computed using the source term described in §5.3. More details are given below.

In the SWE simulator, after updating the depth at each vertex i, if the depth value is over a user-defined threshold ϵ_1, we set its depth value to ϵ_2. This effectively means a volume of $(d - \epsilon_2)A_i$ is removed from vertex i, where A_i is i's control area. An SPH particle with the same volume is created and added to the particle simulator. The initial position of the particle is $\mathbf{v}_i + (b + \frac{1}{2}(\epsilon_1 + \epsilon_2))\mathbf{n}_i$, where $\mathbf{v}_i$ is vertex i's global position, $\mathbf{n}_i$ is its outer normal direction. The initial speed is set to $\frac{\Delta d}{\Delta t}\mathbf{n}_i$, where Δd is the depth change computed from the current time step.

At the same time we also examine the 1-ring neighbor vertices of vertex i. For a neighbor vertex j, if $d_j < \epsilon_1$ but $d_j > \eta_1$, where η_1 is another user-defined threshold, we set the depth value of j to η_2 and add an SPH particle in the same way as above. This is based on the observation that detaching liquid will drag a small amount of liquid around the detaching point with it, and so doing also provides a smoother liquid surface.

On the other hand, an SPH particle can be absorbed by the SWE simulator when it hits the surface. We detect SPH particles that comes within a certain distance of the mesh vertices in a time step. These SPH particles are then removed from the particle simulator. The removed particle in turn act as a source term in Eqn. (1), with their volume contributing to the source term of the nearest vertex on the mesh as in §5.3. In the above scheme, the volume of added SPH particles equals the volume of liquid removed within the control areas of the vertices; the volume of SPH particles removed from the 3D simulator is added into SWE simulator using the source term.

We label all vertices in the SWE simulator that have liquid detached from them and label all particles that should be absorbed in the particle simulator. The former are then be updated to create new particles, and the latter contribute to the source term of the nearest vertex for next time step's calculation.

Due to the SWE assumptions and the fact that the surface tension effect is negligible above certain length scale (e.g. for water its several millimeters), the above hybrid scheme works best in the situation where the surface flow is interacting with a small amount of free liquid. The mesh vertex positions can also serve as boundary particles in the SPH simulation [39].

6.2 Dissolving of Solid Surface

Erosion and dissolution of solid surfaces are interesting phenomena in the real world, where the surface flow dissolves another phase during its motion. The proposed approach can be readily extended to reproduce this kind of coupled phenomena. Inspired by [21] and based on the theory of multiphase flow [40], both the liquid phase and the dissolved phase are simulated using the SWE model (1-2).

For the liquid phase, denoted by subscript l, the governing equations are:

$$\frac{\partial d_l}{\partial t} + \mathbf{u}_l \cdot \nabla d_l = -d_l(\nabla \cdot \mathbf{u}_l) \tag{22}$$

$$\frac{\partial \mathbf{u}_l}{\partial t} + (\mathbf{u}_l \cdot \nabla)\mathbf{u}_l = -\lambda \frac{\nabla P_l}{\rho_l} - \mathbf{\Gamma}_l + \lambda \mathbf{a}_{ext} - \mathbf{M}_l \tag{23}$$

where $\lambda = \frac{d_l + d_s}{d_l}$ is the ratio between total depth and the effective liquid depth, the hydrostatic pressure that contribute to P_l is calculated as $\rho_l g_n h_l = \rho_l g_n (b + d_l + d_s)$. $\mathbf{M}_l$ is the momentum loss due to new dissolved phase, which will be given later. The friction term $\mathbf{\Gamma}_l = \mathbf{\Gamma}_{bl} + \mathbf{\Gamma}_{drag}$ contains bottom friction and drag from dissolved phase. The former is calculated as the total bottom friction subtracting the bottom friction on dissolved phase:

$$\mathbf{\Gamma}_{bl} = \lambda \gamma_l \mathbf{T} \cdot \mathbf{u}_l - \kappa(\lambda - 1)\gamma_s \mathbf{T} \cdot \mathbf{u}_s \tag{24}$$

where $\kappa = \frac{\rho_s - \rho_l}{\rho_l}$ is the coefficient of specific gravity. Drag from dissolved phase is calculated quadratically:

$$\mathbf{\Gamma}_{drag} = \kappa(\lambda - 1) C_d (\mathbf{u}_l - \mathbf{u}_s)|\mathbf{u}_l - \mathbf{u}_s| \tag{25}$$

where C_d is drag coefficient between two phases.

For the dissolved phase, denoted by subscript s, the governing equations are:

$$\frac{\partial d_s}{\partial t} + \mathbf{u}_s \cdot \nabla d_s = -d_s(\nabla \cdot \mathbf{u}_s) + Q_s \tag{26}$$

$$\frac{\partial \mathbf{u}_s}{\partial t} + (\mathbf{u}_s \cdot \nabla)\mathbf{u}_s = -\kappa \frac{\nabla P_s}{\rho_s} - \mathbf{\Gamma}_s + \kappa \mathbf{a}_{ext} \tag{27}$$

The hydrostatic pressure that contributes to P_s is calculated as $\rho_s g_n h_s = \rho_s g_n (b + k_s d_s)$, here $0 < k_s \leq 1$ and $k_s d_s$ is the average thickness of the region where solid transport occurs. No surface tension pressure is considered for the dissolved phase. Q_s is the dissolving source contribution. Referring to Eqn. (24-25), the friction term is:

$$\mathbf{\Gamma}_s = \frac{\kappa}{\kappa + 1}\gamma_s \mathbf{T} \cdot \mathbf{u}_s - \frac{\kappa}{\kappa + 1} C_d (\mathbf{u}_l - \mathbf{u}_s)|\mathbf{u}_l - \mathbf{u}_s|. \tag{28}$$

Assuming the dissolved phase is carried and driven by the liquid phase, we only calculate Eqns. (26-27) in the "wet" region. We also consider there is a maximum dissolution rate, i.e. $d_s \leq \alpha d_l$, α is positive and usually less than 1.

In the simulation, the solid surface is considered to have a variable layer of dissolvable thickness $b_s \geq 0$. In each time step the dissolved amount is first calculated by $Q_s = e_p(d_l + d_s)|\mathbf{u}_l|$, where e_p is the dissolving rate. Then we make sure $Q_s \Delta t$ does not exceed the current $min(b_s, \alpha d_l - d_s)$ value. With the modified Q_s, the momentum carried over to newly dissolved phase $\mathbf{M}_l$ is calculated as:

$$\mathbf{M}_l = (\kappa + 1)\frac{Q_s}{d_l}\mathbf{u}_s. \tag{29}$$

We ensure the relation $d_s \leq \alpha d_l$ when updating d_s and after the advection steps of both phases. The exceeded amount of dissolved phase is considered to stick back to the surface layer as an addition to b_s value. The bottom height b is thus calculated as $b = b_0 + b_s$, where b_0 is the "real" solid bottom height. Similar to §7.2, the bottom friction tensor $\mathbf{T}$ can be updated following the change of b as well.

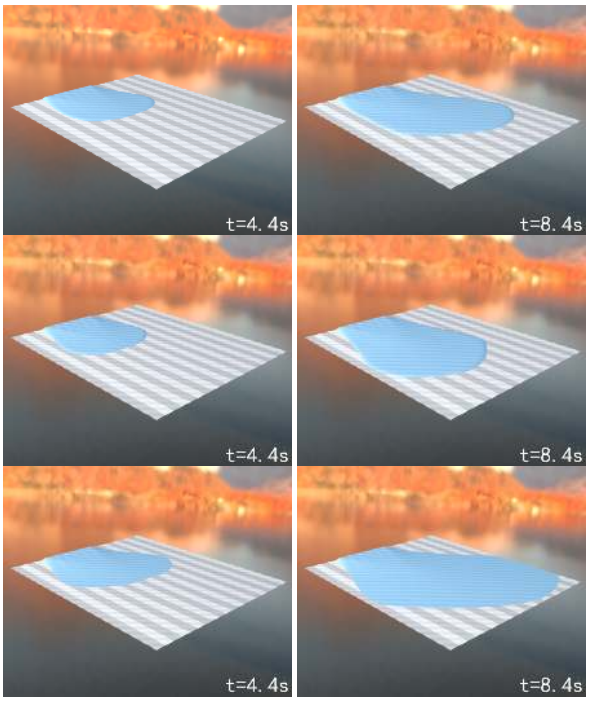


Fig. 4. Top: Our approach. The feature-based friction ensures the liquid flow to follow the wrinkle direction, and the contact angle is maintained on the liquid front. Middle: Replacing the feature-based friction with an isotropic friction, liquid fails to flow along the wrinkle direction. Bottom: Removing the surface tension calculation in Eqn. (21), the contact angle is no longer maintained and liquid moves faster across the surface.

7 Implementation

In this section we further explain the implementation of the proposed real-time high-fidelity surface flow simulation framework, and a work flow of the new scheme is also provided.

7.1 Pre-computation from the Triangle Mesh

The formulas given in §4 and §5 require certain knowledge of the neighborhood mesh structure of a vertex on the triangle mesh. Many of the calculations can be done in a pre-computation step and stored for later use in the simulation.

The pre-computation includes several steps. Firstly, a smoothed mesh is computed from the original mesh using the Laplacian smoothing method. One can also perform mesh simplification before smoothing to reduce the number of vertices in the smoothed mesh. Then for each vertex v on the smoothed mesh, we construct a data structure that contains its n-ring neighborhood information. Specifically, nearby vertex and triangle face structure flattened in the tangential plane and the 2×2 affine matrices M_{ij} for all vertices in the n-ring neighborhood (see §5.1) are stored. Next, we also pre-compute the friction tensor for each vertex on the smoothed mesh (see §4.1). Finally, other geometry information such as the control area of a vertex and the bottom height position b etc. are also pre-computed and stored, where b is set equal to the distance of v to its projected point v_p on the original mesh along the normal direction of the smoothed mesh.

Note that the above pre-computation result does not change if the original mesh is not deformed. In such cases

the pre-computation can be done only once for a given triangle mesh and the data structure obtained can be later applied to multiple simulations, ensuring efficiency during the simulation step. Since in one time step the CFL condition puts a limit to fluid displacements, which scales with mesh resolution, a 3-ring neighborhood structure is found sufficient for simulations in practice.

7.2 Deformable Meshes

The proposed approach readily supports surface flow on deformable meshes. When the original mesh deforms, the smoothed mesh needs to be recomputed, and the liquid depth and velocity values on the new vertices can be obtained from interpolation on their projected points over the former smoothed mesh. However, computational cost can be much reduced if the mesh deformation follows some restrictions. If the detailed original mesh vertices always lie above the smoothed mesh vertices along the smoothed mesh normal (this can be achieved by moving the smoothed mesh vertices to the projected position of original mesh vertices after the smoothing algorithm), we can edit the vertices' positions of the smoothed mesh (e.g. for stretching or bending) and the bottom height (e.g. for mesh detail changing) on each vertex to indirectly control mesh deformation. In such cases we only need to re-calculate the normal of the smoothed mesh, the affine matrix and the friction tensor.

7.3 Extrapolation Operations

Similar to [1], we extrapolate the depth d from $d > 0$ positions to those vertices on the liquid front that directly link to them by a mesh edge. If the extrapolated value is above zero, we set the value as 0. This extrapolated value provides a better estimation of the liquid front position, and benefits in pressure gradient computation, in the advection step and in calculation of A_{lg} in Eqns. (20-21). It also helps as an anti-aliasing method in the output liquid surface mesh for rendering. It is to be noted that in Eqn. (10), when averaging d on a triangle containing liquid front line vertices, 0 is used instead of negative extrapolation depths. This is because mathematically the averaged depth in Eqn. (10) should be the average height of the "wet" region part with liquid.

The velocity of surface flow is well-defined in the "wet" regions but not in the "dry" regions, which is needed for advection of liquid on the mesh. Similar to SWE simulation on a planar surface [17], we use a fast-marching step to define the velocity for the "dry" regions in the n-ring neighborhood of vertices with positive liquid depth. The velocity of a "dry" (i.e. $d = 0$) vertex is set to that of the nearest $d > 0$ vertex.

7.4 Algorithm Framework

The work flow of the proposed real-time high-fidelity surface flow simulation framework includes four main steps:

1) For a triangle mesh, the information needed in the simulation is extracted in the pre-computation step described in §7.1. For coupling with the particle similator, a particle boundary is also generated from the triangle mesh.
2) At the beginning of the simulation, the pre-computed data are read into the simulators. For deformable meshes, normal of the smoothed mesh, the affine matrix and the friction tensor are re-computed each step instead.
3) In each time step of the SWE simulator, we sequentially perform advection, velocity updating, fast marching extrapolation and height updating following the formulation described in §4 and §5. A dissolving phase can also be simulated using a slightly modified set of governing equations §6.2, providing artistic effect of surface dissolving.
4) For coupling with 3D particle simulators, a time step is simultaneously simulated in the particle simulator. At the end of each time step of the particle simulator, whose calculation is performed in the same way as the standard WCSPH approach, we handle interactions between the SWE simulator and the particle simulator following the schemes described in §6.1.

8 Results

In this section we show the effectiveness of our approach through various simulation results, and a supplemental video is also provided to demonstrate the liquid motions. The algorithm is implemented on an Nvidia GeForce GTX 980 GPU. Detailed performance data are listed in Table 1, where all video clips are played with 30 frames per second. We typically set $\gamma = 1.3, \sigma = 1.0, \kappa = 2.0, C_d = 10.0, e_p = 0.005$, and set $\epsilon_1, \epsilon_2, \eta_1, \eta_2$ separately about $3.0, 1.6, 2.4, 2.1$ times the average mesh edge length.

Example 1, Fig. 4 demonstrates the bottom friction and the surface tension models, including the effect of contact angle. Liquid flows into the scene from one side and the bottom height is artificially set with sinusoidal wrinkles. In the top row is the result using our method. It can be observed that the liquid naturally maintains its contact angle at the liquid front while the flow largely follows the wrinkle direction. In the middle row the feature-based friction is replaced with an isotropic friction. It is clear that without the feature-based friction model, the liquid fails to flow according to surface features. In the bottom row, the P_{bs} calculation is removed. As a result, the contact angle is no longer maintained and the liquid also flows faster across the surface.

Example 2, Fig. 5 shows a comparison study, with water flowing down from the upper side of a highly-curved tree trunk surface with many fine wrinkles. The simulation result from our method is shown in the top row, where the flow pattern naturally follows the veins on the bark and temporary submerging of the local vein ridge can also be observed as the water volume passes the bark surface. These effects from the fine surface features are hard to capture using previous approaches. For comparison, the recent method [3] is implemented, and the result is shown in the middle row. It is worth noting that the mass density value has to be bounded in each step to enable the lubrication-theory-based simulation to run on the trunk model. It can be observed that the liquid has very viscous appearance with residual liquid remained at the top of the trunk even near the end of the simulation. Non-smooth liquid surface is also observed early in the simulation due

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2017.2720672, IEEE Transactions on Visualization and Computer Graphics

Fig. 5. Water flowing down a highly-curved tree trunk. Top row: simulated using our method at 3.72ms/step, the flow pattern naturally follows the veins on the bark forming realistic streams down the surface. Middle row: simulated using a latest lubrication model [3] at 821.28ms/step. Bottom row: simulated using previous SWE method [1] at 65.35ms/step. The flow motions obtained with the previous approaches appear very viscous and fail to respond to local surface features.

Fig. 6. Liquid flowing on a deforming surface, where the flow motion naturally follows the wrinkles during non-rigid mesh movement.

TABLE 1
Performance

Example	Vertex Number	Average Time (ms/step)	Step per Frame
1	15,800	2.58	3
2	44,000	3.72	2
3	23,200	3.11 + 7.23*	4
4	52,200	4.97 / 7.62†	2
5	23,200 + 129,000‡	6.84 + 8.91‡	3
6	45,100	3.52	2

* Re-calculation time for non-rigid mesh.
† Two-phase coupled simulation.
‡ Particle number and simulation time of SPH simulation.

to unaligned surface normals. Showing in the bottom row is result of the previous method [1] which also uses the SWE model. Without modeling of feature-based friction due to solid surface variations, the liquid flows unnaturally on the highly curved surface and does not respond to the surface features. It can also be observed that the liquid front moves fast down the trunk, but the main volume of water has a highly viscous appearance, possibly because the low-speed assumption and the artificial velocity damping do not work well with thin surface flow with finite speed. Our approach is able to run at 3.72ms per step (7.44ms per frame) on GPU, achieving real-time performance. The methods [1] and [3] run at 65.35ms per step (130.70ms per frame) and 821.28ms per step (1.64s per frame) separately on an 2.40GHz Intel Xeon processor.

Example 3, in Fig. 6, liquid flows on a deforming mesh. Besides non-rigid mesh movement, the detailed mesh features are also varied during the simulation. Visually realistic surface flow is obtained on the deformable mesh, where the liquid flows more easily along the direction of wrinkles.

In Example 4 we demonstrate the dissolution process of a solid surface and the direct control of friction tensor. Fig. 7 uses a smooth mesh surface as input. In the top and middle cases, liquid is flowing down a truncated cone placed on a horizontal plane. The bottom friction is calculated using the texture map on the surface, where the color gradient direction is artificially taken as $\bar{\mathbf{x}}$, the gradient value as τ_{max}, and set $\tau_{\perp} = \beta\tau_{max}$, where $\beta < 1$ is a positive scaling factor. Under different friction tensor settings, the liquid motion follows the texture pattern to different extents, which will provide flexibility for artistic design. Shown in the bottom row is a two-phase coupled simulation: the frictional surface texture is gradually dissolved by the transparent liquid, which in turn changes the bottom friction, generating an

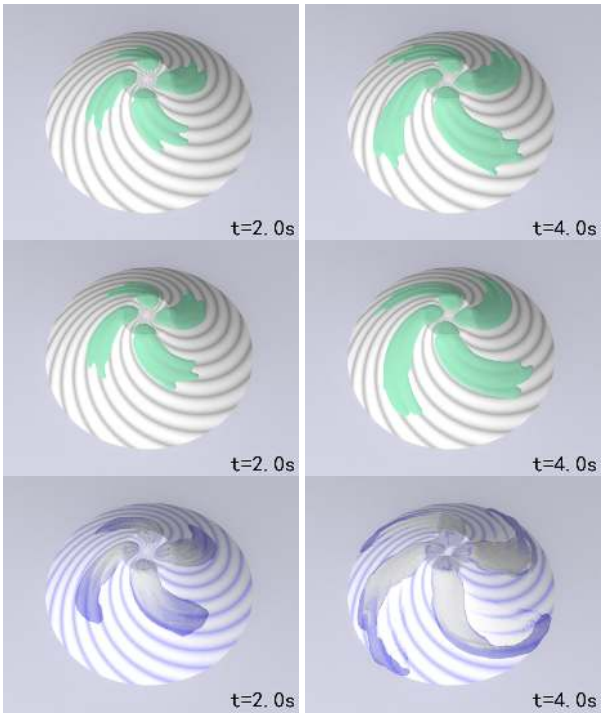


Fig. 7. Top view of liquid flowing down a truncated cone using an artificially generated friction tensor. Top: $\gamma = 30$, $\beta = 0.2$; Middle: $\gamma = 80$, $\beta = 0.1$. Liquid has more tendency to follow the texture direction under higher γ value. Bottom: A two-phase coupled simulation, where the paint on the surface is eroded by the transparent liquid and dissolves in the liquid flow.

eroded appearance. The ratio between d_s and $d_s + d_l$ is used as the dissolved phase concentration, controlling liquid color in the rendering. The concentration variance of the two phases and the water-eroded surface pattern are both realistically reproduced.

Example 5, in Fig. 8 water flows down the armadillo model. The proposed approach naturally captures liquid motions, including in the high-curvature regions, such as water dripping from the downside of the model (e.g. under the arm) and falling onto the upside of the model (e.g. on the leg).

Example 6, in Fig. 9 liquid flows down a man's face. Different wind velocity fields, previously generated by other 3D simulations, are applied to the scene. The top row shows result with no wind, where liquid flows down the face. In the middle row wind blows from the front of the face, pushing the liquid around. In the bottom row wind blows from the left, pushing the liquid to the right. Plausible surface flow following face features in both low and high curvature areas are obtained with existence of wind blowing effect.

9 Conclusion and Discussion

A novel systematic solution is proposed for reproducing sophisticated surface flow phenomena in graphics applications, achieving real-time performance with broad range of high-fidelity results. The proposed approach features an original friction model and flexible coupling with 3D fluid flows and multi-phase interactions. It also provides a consistent surface tension formulation and a stable numerical scheme on general triangle meshes with efficient geometric computation.

With respect to the numerical solver, an explicit time integration approach is adopted for the benefit of GPU parallelization. A real-time simulation performance is achieved under the appropriate CFL condition. An added benefit from adopting the explicit simulation scheme lies in the convenience of simulating multi-fluid and fluid-surface interactions. We also adopt the semi-Lagrange method for convective advection, which is widely used in graphics applications both in implicit and explicit simulations but is not exactly conservative. There are some recent techniques (e.g. [41]) on MAC grids to enhance the conservation property of semi-Lagrange methods, and it may be beneficial to extend such techniques to arbitrary 3D triangle meshes.

On the limitation side, the CFL condition limits further performance gain, beyond benefits of parallelization, from large time steps, and the advection algorithm described in §5.1 also requires the surface flow must not travel out of the neighborhood n-ring structure in one time step. Having larger friction forces (linear bottom friction or quadratic wind friction) will also require smaller time steps for simulation stability using the explicit schemes. The mesh deformation scheme in §7.2 works well for small mesh deformation, but can lose stability in the presence of large mesh distortion.

Besides the aforementioned limitations which will benefit from further research, we would also like to extend the current approach to cover phenomena involving surface flow in gaps. One possibility for this is to incorporate the lubrication theory in the simulation framework. Viscous muddy fluid behavior on 2D surface following recent works in 3D (e.g. [42]) is another possible future direction.

Acknowledgments

This work is supported by the Natural Science Foundation of China (Project Number 61602265, 61521002), Research Grant of Beijing Higher Institution Engineering Research Center, European Community's Seventh Framework Programme (Marie Curie International Research Staff Exchange Scheme, Grant No. 612607) and the Sêr Cymru National Research Network in Advanced Engineering and Materials. We would also like to thank Huamin Wang for his help.

References

[1] H. Wang, G. Miller, and G. Turk, "Solving general shallow wave equations on surfaces," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '07. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007, pp. 229–238. [Online]. Available: http://dl.acm.org/citation.cfm?id=1272690.1272721

[2] R. Angst, N. Thuerey, M. Botsch, and M. Gross, "Robust and efficient wave simulations on deforming meshes," *Computer Graphics Forum*, vol. 27, no. 7, pp. 1895–1900, 2008. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2008.01337.x

[3] O. Azencot, O. Vantzos, M. Wardetzky, M. Rumpf, and M. Ben-Chen, "Functional thin films on surfaces," in *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, ser. SCA '15. New York, NY, USA: ACM, 2015, pp. 137–146. [Online]. Available: http://doi.acm.org/10.1145/2786784.2786793

[4] J. Stam, "Flows on surfaces of arbitrary topology," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 724–731, Jul. 2003. [Online]. Available: http://doi.acm.org/10.1145/882262.882338

Fig. 8. Water flowing down the armadillo model. Water drops are formed under the arm and drip onto the leg.

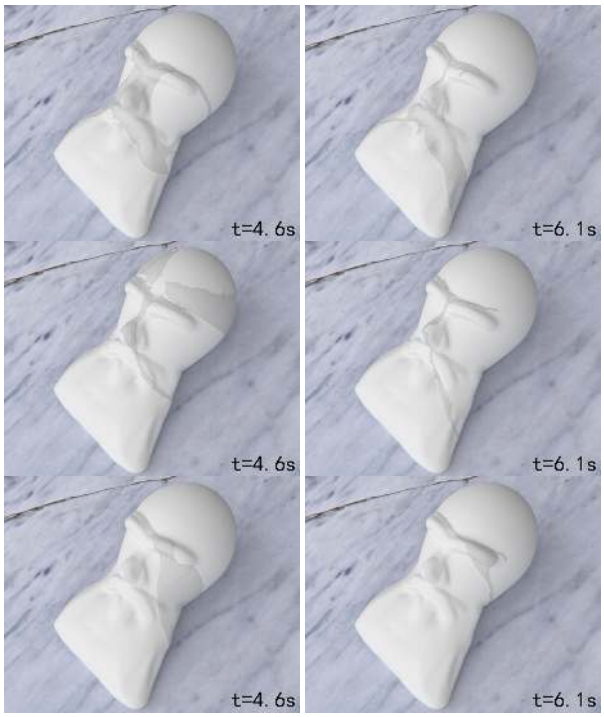


Fig. 9. Liquid flows down a man's face. Different wind is applied to the scene. Top: result with no wind. Middle:wind blows from the front of the face, pushing the liquid around. Bottom: wind blows from the left, pushing the liquid to the right.

[5] L. Shi and Y. Yu, "Inviscid and incompressible fluid simulation on triangle meshes: Research articles," *Comput. Animat. Virtual Worlds*, vol. 15, no. 3-4, pp. 173–181, Jul. 2004. [Online]. Available: http://dx.doi.org/10.1002/cav.v15:3/4

[6] Z. Fan, Y. Zhao, A. Kaufman, and Y. He, "Adapted unstructured lbm for flow simulation on curved surfaces," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '05. New York, NY, USA: ACM, 2005, pp. 245–254. [Online]. Available: http://doi.acm.org/10.1145/1073368.1073404

[7] S. Elcott, Y. Tong, E. Kanso, P. Schröder, and M. Desbrun, "Stable, circulation-preserving, simplicial fluids," *ACM Trans. Graph.*, vol. 26, no. 1, Jan. 2007. [Online]. Available: http://doi.acm.org/10.1145/1189762.1189766

[8] P. Neill, R. Metoyer, and E. Zhang, "Fluid flow on interacting deformable surfaces," in *ACM SIGGRAPH 2007 Posters*, ser. SIGGRAPH '07. New York, NY, USA: ACM, 2007. [Online]. Available: http://doi.acm.org/10.1145/1280720.1280783

[9] S. Auer, C. B. Macdonald, M. Treib, J. Schneider, and R. Westermann, "Real-time fluid effects on surfaces using the closest point method," *Computer Graphics Forum*, vol. 31, no. 6, pp. 1909–1923, 2012. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2012.03071.x

[10] L. Carvalho, M. Andrade, and L. Velho, "Fluid simulation on surfaces in the gpu," in *Graphics, Patterns and Images (SIBGRAPI), 2012 25th SIBGRAPI Conference on*, Aug 2012, pp. 205–212.

[11] D. J. Hill and R. D. Henderson, "Efficient fluid simulation on the surface of a sphere," *ACM Trans. Graph.*, vol. 35, no. 2, pp. 16:1–16:9, Apr. 2016. [Online]. Available: http://doi.acm.org/10.1145/2879177

[12] N. Thürey, U. Rüde, and M. Stamminger, "Animation of open water phenomena with coupled shallow water and free surface simulations," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '06. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2006, pp. 157–164. [Online]. Available: http://dl.acm.org/citation.cfm?id=1218064.1218086

[13] N. Chentanez, M. Muller, and T.-Y. Kim, "Coupling 3d eulerian, heightfield and particle methods for interactive simulation of large scale liquid phenomena," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 10, pp. 1116–1128, Oct. 2015. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2015.2449303

[14] M. Müller, D. Charypar, and M. Gross, "Particle-based fluid simulation for interactive applications," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '03. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2003, pp. 154–159. [Online]. Available: http://dl.acm.org/citation.cfm?id=846276.846298

[15] M. Kass and G. Miller, "Rapid, stable fluid dynamics for computer graphics," *SIGGRAPH Comput. Graph.*, vol. 24, no. 4, pp. 49–57, Sep. 1990. [Online]. Available: http://doi.acm.org/10.1145/97880.97884

[16] A. T. Layton and M. van de Panne, "A numerically efficient and stable algorithm for animating water waves." *The Visual Computer*, vol. 18, no. 1, pp. 41–53, 2002. [Online]. Available: http://dblp.uni-trier.de/db/journals/vc/vc18.html

[17] R. Bridson, *Fluid Simulation for Computer Graphics*. A K Peters/CRC Press, Sep. 2008. [Online]. Available: http://www.worldcat.org/isbn/1568813260

[18] M. Becker and M. Teschner, "Weakly compressible sph for free surface flows," in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '07. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007, pp. 209–217. [Online]. Available: http://dl.acm.org/citation.cfm?id=1272690.1272719

[19] M. Macklin and M. Müller, "Position based fluids," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 104:1–104:12, Jul. 2013. [Online]. Available: http://doi.acm.org/10.1145/2461912.2461984

[20] J. Bender and D. Koschier, "Divergence-free sph for incompressible and viscous fluids," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 3, pp. 1193–1206, March 2017.

[21] B. Ren, C. Li, X. Yan, M. C. Lin, J. Bonet, and S.-M. Hu, "Multiple-fluid sph simulation using a mixture model," *ACM Trans. Graph.*, vol. 33, no. 5, pp. 171:1–171:11, Sep. 2014. [Online]. Available: http://doi.acm.org/10.1145/2645703

[22] T. Yang, J. Chang, B. Ren, M. C. Lin, J. J. Zhang, and S.-M. Hu, "Fast multiple-fluid simulation using helmholtz free energy," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 201:1–201:11, Oct. 2015. [Online]. Available: http://doi.acm.org/10.1145/2816795.2818117

[23] X. Yan, Y.-T. Jiang, C.-F. Li, R. R. Martin, and S.-M. Hu, "Multiphase sph simulation for interactive fluids and solids," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 79:1–79:11, Jul. 2016. [Online]. Available: http://doi.acm.org/10.1145/2897824.2925897

[24] B. Ren, Y. Jiang, C. Li, and M. C. Lin, "A simple approach for bubble modelling from multiphase fluid simulation," *Computational Visual Media*, vol. 1, no. 2, pp. 171–181, 2015. [Online]. Available: http://dx.doi.org/10.1007/s41095-015-0020-6

[25] S. Jeong and C.-H. Kim, "Combustion waves on the point set surface," *Computer Graphics Forum*, vol. 32, no. 7, pp. 225–234, 2013. [Online]. Available: http://dx.doi.org/10.1111/cgf.12230

[26] H. Wang, P. J. Mucha, and G. Turk, "Water drops on surfaces," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 921–929, Jul. 2005. [Online]. Available: http://doi.acm.org/10.1145/1073204.1073284

[27] Y. Zhang, H. Wang, S. Wang, Y. Tong, and K. Zhou, "A deformable surface model for real-time water drop animation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 8, pp. 1281–1289, Aug. 2012. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2011.141

[28] X. He, H. Wang, F. Zhang, H. Wang, G. Wang, and K. Zhou, "Robust simulation of sparsely sampled thin features in sph-based free surface flows," *ACM Trans. Graph.*, vol. 34, no. 1, pp. 7:1–7:9, Dec. 2014. [Online]. Available: http://doi.acm.org/10.1145/2682630

[29] F. Da, D. Hahn, C. Batty, C. Wojtan, and E. Grinspun, "Surface-only liquids," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 78:1–78:12, Jul. 2016. [Online]. Available: http://doi.acm.org/10.1145/2897824.2925899

[30] F. Losasso, G. Irving, E. Guendelman, and R. Fedkiw, "Melting and burning solids into liquids and gases," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 3, pp. 343–352, 2006.

[31] C. Wojtan, M. Carlson, P. J. Mucha, and G. Turk, "Animating corrosion and erosion." in *NPH*. Citeseer, 2007, pp. 15–22.

[32] C. B. Vreugdenhil, *Numerical Methods for Shallow-Water Flow*. Springer, Dec. 2010.

[33] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds." Springer-Verlag, 2002, pp. 35–57.

[34] T. H. A. Duhaut and D. N. Straub, "Wind stress dependence on ocean surface velocity: Implications for mechanical energy input to ocean circulation," *Journal of Physical Oceanography*, vol. 36, no. 2, pp. 202–211, 2006.

[35] R. J. LeVeque, *Finite volume methods for hyperbolic problems*, ser. Cambridge texts in applied mathematics. Cambridge, New York: Cambridge University Press, 2002, autres tirages : 2003, 2005, 2006. [Online]. Available: http://opac.inria.fr/record=b1100566

[36] F. de Goes, M. Desbrun, and Y. Tong, "Vector field processing on triangle meshes," in *SIGGRAPH Asia 2015 Courses, Kobe, Japan, November 2-6, 2015*, 2015, pp. 17:1–17:48. [Online]. Available: http://doi.acm.org/10.1145/2818143.2818167

[37] V. R. Voller, *Basic control volume finite element methods for fluids and solids*. World Scientific Publishing Company Incorporated, 2009, vol. 1.

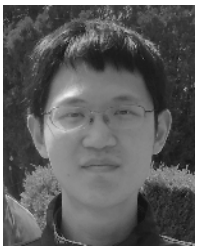
[38] T. Young, "An essay on the cohesion of fluids," *Philosophical Transactions of the Royal Society of London*, vol. 95, pp. 65–87, Jan. 1805. [Online]. Available: http://dx.doi.org/10.1098/rstl.1805.0005

[39] N. Akinci, M. Ihmsen, G. Akinci, B. Solenthaler, and M. Teschner, "Versatile rigid-fluid coupling for incompressible sph," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 62:1–62:8, Jul. 2012. [Online]. Available: http://doi.acm.org/10.1145/2185520.2185558

[40] G. H. Yeoh and J. Tu, *Computational Techniques for Multiphase Flows*. Butterworth-Heinemann, 2009.

[41] M. Lentine, J. T. Grétarsson, and R. Fedkiw, "An unconditionally stable fully conservative semi-lagrangian method," *J. Comput. Phys.*, vol. 230, no. 8, pp. 2857–2879, Apr. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.jcp.2010.12.036

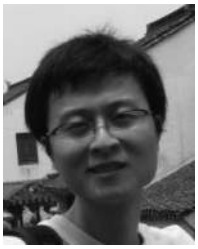
[42] A. Peer and M. Teschner, "Prescribed velocity gradients for highly viscous sph fluids with vorticity diffusion," *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, no. 99, pp. 1–1, 2016.



Bo Ren received the PhD degree from Tsinghua University in 2015. He is currently a lecturer in the College of Computer and Control Engineering, Nankai University, Tianjin. His research interests include physically-based simulation and rendering, scene geometry reconstruction and analysis.

Tailing Yuan is a master student in Tsinghua University. He recieved his bachelor degree from the same university in 2016. His research interests include physically-based simulation and scene text recognition.

Chenfeng Li is currently an Associate Professor at the College of Engineering of Swansea University, UK. He received his BEng and MSc degrees from Tsinghua University, Beijing in 1999 and 2002, respectively, and received his Ph.D. degree from Swansea University in 2006. His research interests include computational solid mechanics, computational fluid dynamics, and computational graphics. He is the Editor-in-Chief of Engineering Computations.



Kun Xu is an associate professor in the Department of Computer Science and Technology, Tsinghua University. Before that, he received his bachelor and doctors degrees from the same university in 2005 and 2009, respectively. His research interests include realistic rendering and image/video editing.

Shi-Min Hu is currently a professor in the department of Computer Science and Technology, Tsinghua University, Beijing. He received the PhD degree from Zhejiang University in 1996. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He has published more than 100 papers in journals and refereed conference. He is Editorin-Chief of Computational Visual media, and on editorial board of several journals, including IEEE Transactions on Visualization and Computer Graphics, Computer Aided Design and Computer & Graphics. He is a senior member of IEEE and ACM.