Robust Simulation of Small-Scale Thin Features in SPH-based Free Surface Flows

Xiaowei He[1], Huamin Wang[2],Fengjun Zhang[1], Hongan Wang[1], Guoping Wang[3], Kun Zhou[4]
[1]Institute of Software, Chinese Academy of Sciences
[2]The Ohio State University, USA
[3]Peking University, China
[4]Zhejiang University, China

Smoothed particle hydrodynamics (SPH) is efficient, mass preserving, and flexible in handling topological changes. However, small-scale thin features are difficult to simulate in SPH-based free surface flows, due to a number of robustness and stability issues. In this paper, we address this problem from two perspectives: the robustness of surface forces and the numerical instability of thin features. We present a new surface tension force scheme based on a free surface energy functional, under the diffuse interface model. We develop an efficient way to calculate the air pressure force for free surface flows, without using air particles. Compared with previous surface force formulae, our formulae are more robust against particle sparsity in thin feature cases. To avoid numerical instability on thin features, we propose to adjust the internal pressure force by estimating the internal pressure at two scales and filtering the force using a geometry-aware anisotropic kernel. Our result demonstrates the effectiveness of our algorithms in handling a variety of small-scale thin liquid features, including thin sheets, thin jets, and water splashes.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Animation*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*Physically based modeling*

General Terms: Fluid Dynamics

Additional Key Words and Phrases: smoothed particle hydrodynamics (SPH), surface tension, diffuse interface model, liquid animation, thin feature

ACM Reference Format:

1. INTRODUCTION AND BACKGROUND

Small-scale thin features, such as water streamlets and sheets, provide interesting details in physically based liquid animation. But how to prevent them from being destroyed by resolution limit and numerical instability is challenging in computer graphics. While most research efforts have been spent on solving this problem for grid-based Eulerian simulators [Losasso et al. 2004; Irving et al. 2006; Kim et al. 2007; Sussman and Ohta 2009] and mesh-based Lagrangian simulators [Thürey et al. 2010; Wojtan et al. 2010; Brochu et al. 2010; Zhang et al. 2011; Clausen et al. 2013], little has been done to smoothed particle hydrodynamics (SPH) and its simulators. In fact, SPH is highly sensitive to the lack of particles around liquid surfaces in free surface flows, which makes small-scale thin features even harder to simulate. Since SPH simulators are welcomed in many applications for its efficiency, mass preservation, and flexibility in handling topological changes, we think it is necessary to robustly simulate thin features in SPH-based free surface flows as well.

Different from the recent work on the resolution limit of particle-based simulation [Adams et al. 2007; Solenthaler and Gross 2011; Ando et al. 2012; Ando et al. 2013], our work is focused on the numerical aspect of small-scale thin features. Specifically, we are interested in knowing how to improve their robustness, even when there are not sufficient particles. From our experience, we found two main factors related to this problem.

The first factor is the surface forces, especially surface tension. Surface tension plays an important role in both maintaining and destroying small-scale thin features in the real world. There are two typical ways to calculate surface tension under the SPH framework: the continuum surface force (CSF) method [Morris 2000; Müller et al. 2003; Hu and Adams 2006] and the inter-particle interaction force (IIF) method [Nugent and Posch 2000; Tartakovsky and Meakin 2005; Becker and Teschner 2007]. By defining surface tension as a mean curvature flow at the macroscopic level, the CSF method calculates the surface normal at each particle and then uses a smoothing kernel to estimate the divergence of the surface normal. Alternatively, the IIF method calculates surface tension at the microscopic level as an inter-molecular force between two particles. While both methods are capable enough for large water bodies, they become less accurate and robust with fewer particles, making thin features difficult to survive over time regardless of surface tension coefficients.

The second factor is the numerical instability inherent in both attraction forces and repulsion forces. Unlike linear spring forces, SPH-based attraction forces, including the surface tension force and the air pressure force, are stronger when particles move clos-

Authors' addresses: .

DOI 10.1145/XXXXXXX.YYYYYYY
http://doi.acm.org/10.1145/XXXXXXX.YYYYYYY

er and weaker when particles are more separated. If they are the only forces, they will separate particles into a number of clusters. This problem is commonly known as *tensile instability*. Previous research on tensile instability was mainly focused on large deformation in elastic solids [Chen et al. 1999; Monaghan 2000], and the proposed techniques are not directly applicable to liquid simulation. Meanwhile, SPH-based repulsion forces, such as the internal pressure force, tend to push particles out, when they are located slightly off the same line or plane due to numerical errors. Both attraction forces and repulsion forces can cause thin liquid features to rupture, such as the thin sheet example shows in Fig. 6. We note that numerical instability is different from real world surface tension instability. Its existence in free surface flows is largely due to the fact that particles are defined on the liquid side of free surfaces only. So adding ghost particles on the air side can help reduce numerical instability as Schechter and Bridson [2012] suggested, but that requires more implementation effort and computational cost.

Based on these two observations, we made the following contributions to robustly simulate small-scale thin features:

—A surface tension scheme derived from the surface energy functional under the diffuse interface model. This scheme can robustly reflect the local geometry of liquid surfaces, even without a sufficient number of particles.

—An air pressure formula solely based on liquid particles in the liquid phase. It can produce a variety of air pressure effects with little computational overhead.

—An internal pressure force algorithm based on two-scale pressure estimation and geometry-aware anisotropic filtering. It effectively reduces numerical instability, without affecting the impressibility of water bodies.

We implemented our new methods and integrated them into a standard SPH/WCSPH solver. The whole system is efficient and compatible with graphics hardware acceleration. Our experiment shows that it can realistically and robustly simulate a variety of small-scale thin features, such as thin jets (Fig. 9), thin films (Fig. 10), and water splashes (Fig. 8).

2. PREVIOUS WORK

Smoothed particle hydrodynamics (SPH) has been widely used in computational physics and computer graphics to simulate dynamic liquid behaviors. Previous research has been focused on a number of problems, including artificial viscosity [Monaghan 1989; 1994], incompressibility [Becker and Teschner 2007; Solenthaler and Pajarola 2009], boundary conditions [Müller et al. 2003; Schechter and Bridson 2012], coupling with other fluids and solids [Monaghan 1994; Müller et al. 2005; Solenthaler and Pajarola 2008; Ihmsen et al. 2010; Akinci et al. 2012], and particles with variable sizes [Adams et al. 2007; Solenthaler and Gross 2011; Ando et al. 2012; Ando et al. 2013].

Among these problems, surface tension and its influence on small-scale thin features is a much less studied one. Initially developed for multiphase flows [Morris 2000], the continuum surface force (CSF) method was extended by Müller and colleagues [2003] to handle free surface cases as well. Hu and Adams [2006] improved the robustness of the CSF method by formulating surface tension as the divergence of a stress tensor, rather than the surface normal. Sirotkin and Yoh [2011] presented a new smoothing kernel and gradient correction terms to avoid compressional instability in the CSF method. The particle-based surface tension flow can also be calculated by the inter-particle interaction force (IIF) method, as Nugent and Posch [2000] showed. Using a combination of repulsion and attraction forces, Tartakovsky and Meakin [2005] used the IIF method to simulate both surface tension and fluid-solid coupling effects. Becker and Teschner [2007] applied the IIF method to calculate surface tension in free surface flows.

Unfortunately, the accuracy of CSF and IIF depends on a sufficient number of particles. Their results become less reliable and more noisy, when they handle thin features with fewer particles. Alternatively, Zhang [2010] and Andersson and collaborators [2010] proposed to reconstruct liquid surfaces for surface tension calculation. Yu and colleagues [2012] maintained liquid surfaces over time explicitly as triangle meshs. Both methods are more robust than particle-based surface tension methods, but they require additional computational cost. Since many issues in free surface flows do not occur in multiphase flows, a straightforward idea is to create ghost particles on the air side of free surfaces, as Schechter and Bridson [2012] showed. The computational overhead of processing these new particles can be large, when a scene contains many thin liquid features.

The existence of thin features in liquid animation also relies on the liquid surface reconstruction process. The blobby sphere approach proposed by Blinn [1982] extracts an isosurface from a scalar field using a sum of radial basis functions. Zhu and Bridson [2005] later improved this method to reduce artificial bumps and indentations, by adding compensations for local particle density variations. Adams and collaborators [2007] proposed to track particle-surface distances over time, so that liquid surfaces can be smoothly reconstructed for non-uniform particles. Instead of using an isotropic smoothing kernel, Yu and Turk [2010] used an anisotropic smoothing kernel based on local particle distributions, in order to reduce surface bumps without destroying thin features. Bhatacharya and colleagues [2011] formulated liquid surface reconstruction as a constrained optimization problem and used the level set approach to minimize the thin plate energy of liquid surfaces. Akinci and collaborators [2012] applies mesh operations to reduce bumps and improve the reconstruction quality efficiently. While our work is focused on numerical simulation, our system can benefit from the use of these liquid surface reconstruction techniques for more robust thin feature effects as well.

3. SURFACE FORCES

In this section, we propose new techniques to handle surface tension and air pressure for SPH-based free surface flows. Both techniques are based on the diffuse interface model, whose history can be traced back to van der Waals' early work [1893]. The basic idea behind the diffuse interface model is to assume that a liquid surface has a small but finite thickness, across which physical quantities can change rapidly but smoothly from one phase to another. The surface energy in a diffuse interface can be defined as a Helmholtz free energy functional [Cahn and Hilliard 1958]:

$$\mathcal{E} = \int_V \left[f(c) + \frac{\kappa}{2} \left| \nabla c \right|^2 \right] dV, \qquad (1)$$

in which V is the liquid volume, and κ is a squared gradient energy coefficient, $f(c)$ is the bulk free energy density, and c is the condensation field. Typically, the condensation value c is 1, if a point is within the volume; or 0, if a point is outside of the volume; and changes smoothly from 1 to 0, when a point moves across the interface. Intuitively, $|\nabla c|$ indicates where the interface is and how fast c changes. The squared gradient energy term in Eq. 1 is the surface tension energy,

$$\mathcal{E}^{\mathrm{s}} = \int_V \frac{\kappa}{2} |\nabla c|^2 dV, \tag{2}$$

which is proportional to the surface area. The gradient of this energy can then be formulated as the surface tension force, which minimizes the surface area. The diffuse interface model is naturally compatible with particle-based representations, since it does not require liquid surfaces to be explicit.

3.1 Surface Tension Force

To calculate surface tension using the diffuse interface model under the SPH framework, we simply define $c = 1$ at each particle, known as the color field [Morris 2000; Müller et al. 2003]. We then calculate $\nabla_i c$ as:

$$\nabla_i c = \frac{\sum\limits_j V_j c_j \nabla_i W_{ij}^h}{\sum\limits_j V_j W_{ij}^h}, \tag{3}$$

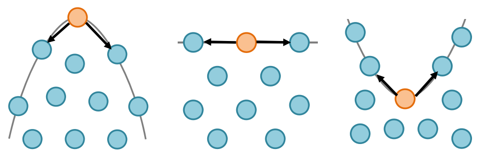
in which V_j is particle j's volume, $W_{ij}^h = W(r_{ij}, h)$ is a smoothing kernel function with a radius h, and r_{ij} is the distance between particle i and j. The denominator $\sum\limits_j V_j W_{ij}^h$ is used here to compensate for missing air particles in free surface flows. Using Eq. 3, we can then obtain $\frac{\kappa}{2}|\nabla_i c|^2$ at each particle i. By treating it as a smoothed energy density at each particle and ignoring the influence of other particles on it, we assume that $\mathcal{E}^{\mathrm{S}}$ can be minimized by minimizing each energy density separately. This assumption allows us to define the surface tension force using the gradient of the energy density:

$$\mathbf{F}_i^{\mathrm{s}} = V_i \nabla_i \left(\frac{\kappa}{2} |\nabla_i c|^2 \right) = \frac{\kappa}{2} \sum_j V_i V_j |\nabla c_j|^2 \nabla_i W_{ij}^h. \tag{4}$$

Intuitively, the surface tension energy density $\frac{\kappa}{2}|\nabla_i c|^2$ can be considered as an approximation to the local surface area of particle i. The surface tension force tries to minimize it, by summing up a set of attraction forces. In ideal cases where interior particles have zero surface areas and surface particles are explicitly specified, we can simply treat the surface tension force as the sum of attraction forces caused by neighboring surface particles, as Fig. 1 shows. To ensure momentum conservation in practice, we calculate the average of two surface tension energy densities and use it in the following surface tension force formula:

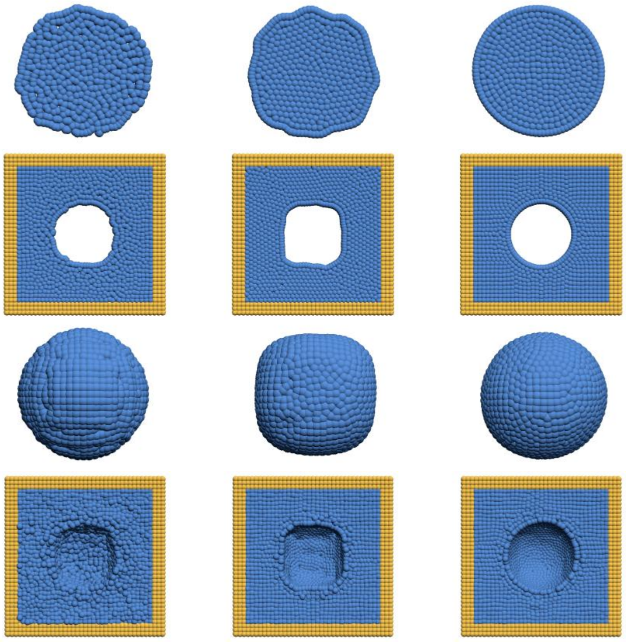
$$\mathbf{F}_i^{\mathrm{s}} = \frac{\kappa}{4} \sum_j V_i V_j \left(|\nabla c_i|^2 + |\nabla c_j|^2 \right) \nabla_i W_{ij}^h. \tag{5}$$

The main advantage of our method is its robustness against particle sparsity, which is commonly noticed on thin features. Unlike the CSF method that relies on ∇c to determine the normal direction and the mean curvature, our method uses $|\nabla c|^2$ to estimate the local surface area only. So when normal estimation becomes problematic, such as a liquid sheet made of a single particle layer, our method can still calculate surface tension forces accurately. Fig. 2 compares our method with the CSF method (in [Müller et al. 2003]) and the IIF method (in [Becker and Teschner 2007]), when the support domain of the smoothing kernel contains less than 20 particles. It shows that our method is more robust in both 2D and 3D.



(a) Convex (b) Flat (c) Concave

Fig. 1. The surface tension force in three surface cases. In these cases, we can model the surface tension force as the sum of attraction forces between surface particles. It tries to deform convex and concave surfaces into flat surfaces, where the surface tension energy gets minimized.



(a) The CSF method (b) The IIF method (c) Our method

Fig. 2. 2D and 3D comparison examples of the three surface tension methods. We simulate the concave examples by restraining liquid particles in a closed container and applying both surface tension forces and air pressure forces (to be discussed in Subsection 3.2).

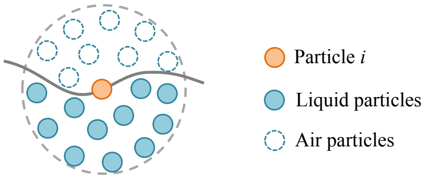
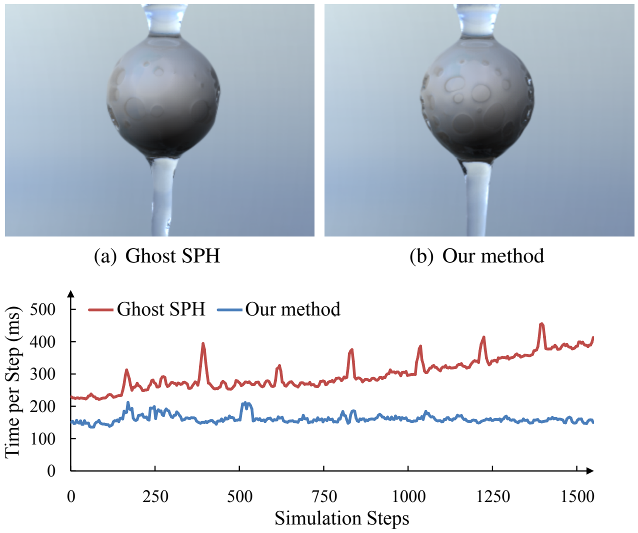


Fig. 3. A surface particle with both air particles and liquid particles in its neighborhood. By defining a negative air pressure at each neighboring liquid particle, we can calculate the air pressure force without explicitly defining air particles.

3.2 Air Pressure Force

Because of the air pressure force, water cannot leave solid surfaces freely nor occupy air bubble volumes in the real world. The air pressure force is straightforward to simulate in multiphase flows using both liquid and air particles. For single-phase free surface



(a) Ghost SPH (b) Our method

Fig. 4. Flowing water. While both the ghost SPH method and our method can be used to simulate flowing water on a solid sphere, our method requires no air particles and runs faster.

flows, Schechter and Bridson [2012] proposed to calculate the air pressure force in a similar way, by adding ghost air particles around liquid surfaces. Since the use of air particles requires more memory and computational cost, an interesting question is: can we simulate the air pressure force without air particles?

To answer this question, let us first assume that air particles still exist. A liquid particle i should be surrounded by both air particles and liquid particles as Fig. 3 shows. Let p_{atm} be the air pressure at each air particle k. The air pressure force at particle i can be calculated as:

$$\mathbf{F}_i^{\mathrm{a}} = -V_i p_{\mathrm{atm}} \sum_k V_k \nabla_i W_{ik}^h. \tag{6}$$

Assuming that air particles and liquid particles are smoothly and uniformly distributed, we have:

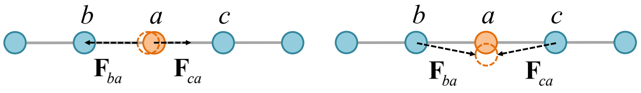
$$\sum_j V_j \nabla_i W_{ij}^h + \sum_k V_k \nabla_i W_{ik}^h = \nabla 1 = \mathbf{0}, \tag{7}$$

in which the first term is summed over liquid particles and the second term is summed over air particles. Using Eq. 7, we can replace the sum in Eq. 6 and get:

$$\mathbf{F}_i^{\mathrm{a}} = V_i p_{\mathrm{atm}} \sum_j V_j \nabla_i W_{ij}^h. \tag{8}$$

Intuitively, Eq. 8 formulates the air pressure force as the sum of attraction forces, by assigning neighboring liquid particles with negative air pressures.

To compare the performance of our method with the ghost SPH method by Schechter and Bridson [2012], we create a solid sphere example as Fig. 4 shows. This example indicates that both methods can produce the flowing effect (together with our surface tension formula in Subsection 3.1), in which water flows on the solid surface and merges at the bottom of the sphere. But since our method does not need air particles, it is faster than the ghost SPH method and its computational cost is independent of thin features. Furthermore, our method does not require any extra memory cost in calculating the air pressure force while the ghost method added approximately 25% memory overhead for this example.



(a) An attraction case (b) A repulsion case

Fig. 5. 1D examples that demonstrate numerical instability issues in SPH-based attraction and repulsion forces.

4. NUMERICAL INSTABILITY

Although the formulae proposed in Section 3 can robustly calculate surface forces, we may still see thin liquid features being affected by numerical instability. This issue is related to both attraction forces and repulsions forces under the SPH framework.

The instability issue related to SPH-based attraction forces, including our surface tension force and our air pressure force, is known as *tensile instability*. To understand this problem, let us consider a 1D case containing one movable particle a and two fixed particles b and c, as Fig. 5a shows. Assuming that the particles have the same size and they are affected by attraction forces only. If a is exactly in the middle of b and c, it receives zero total force and it can stay static. However, if a is positioned slightly closer to b due to numerical errors, then the attraction force exerted on a by b will be larger and the attraction force exerted on a by c will be smaller. So the total force gets unbalanced and pushes a even closer to b. In simulation, this will cause particles to form a set of clusters. According to Swegle and collaborators [1995], the existence of tensile instability can be mathematically identified as a sufficient condition: $\sigma W'' > 0$, where σ is the stress state and W'' is the second derivative of the smoothing kernel to the particle distance.

While SPH-based repulsion forces do not have the tensile instability issue, they have their own instability issue as Fig. 5b shows. In this example, particle a can stay at rest between b and c, when only repulsion forces exist. But if a is positioned slightly off the line, repulsion forces will push it out even further. As a result, repulsion forces cannot maintain thin liquid features in free surface flows, including thin jets and thin sheets. We note that this problem do not appear in multiphase flows, because surrounding air particles will prevent liquid particles from escaping thin features easily.

The instability caused by attraction forces (or called tensile instability) can occur everywhere, while the instability caused by repulsion forces can occur on thin features only. In practice, we do not notice tensile instability in water bodies, since repulsion forces avoid particles from being arbitrarily close. However, tensile instability becomes problematic on thin features, where particles are sparser and internal pressure forces get underestimated by most of the existing simulators.

Based on the previous analysis, we derive our solution against numerical instability as follows. To reduce tensile instability, we first increase internal pressure forces on thin features using two smoothing kernels. Once repulsive internal pressure forces become larger, their instability issue gets noticeable and needs to be addressed as well. We then apply an anisotropic filter on internal pressure forces, so that their influence is limited within thin features. By calculating internal pressure forces in this way, our system can robustly maintain thin features, such as the thin sheet in Fig.6.

4.1 Two-Scale Pressure Estimation

When using a large smoothing kernel, most algorithms cannot distinguish particle sparsity on thin features from particle sparsity in low internal pressure regions. As a result, internal pressure forces are often underestimated on thin features and cannot prevent tensile instability from happening. Using a small kernel can reduce tensile instability, but since fewer particles are involved in calculation, the estimated pressure will be more noisy and less reliable.

To robustly estimate internal pressures for both water bodies and thin features, our idea is to use small and large smoothing kernels together. Let $h = R$ be the radius of the large kernel and $h = r$ be the radius of the small kernel. We first calculate the density using the multiphase method proposed by Solenthaler and Pajarola [2008]:

$$\rho_i^r = \alpha m \sum\nolimits_j W_{ij}^r \quad \text{and} \quad \rho_i^R = m \sum\nolimits_j W_{ij}^R, \tag{9}$$

in which m is the particle mass, α is a scaling factor which corresponds to $(r/R)^3$ in three-dimensional space and $(r/R)^2$ in two-dimensional space. We typically set $R = 2.5d$ and $r = d$, where d is the expected reference distance between two particles. Then we use the local Poisson method [He et al. 2012] to convert the two densities into two pressures p_i^r and p_i^R, respectively. For efficiency, we only apply the special formula with the radius of the integration domain degrading to zero. The pressure p_i^R is more accurate for particles in water bodies, but it is underestimated for particles on thin features. Meanwhile, the pressure p_i^r is less reliable, but it does not have the underestimation issue on thin features. To provide a smooth transition from one to another, we calculate the final internal pressure at particle i as:

$$p_i = \max\left(p_i^R, \beta p_i^r\right) + p_{atm}. \tag{10}$$

The coefficient β serves two purposes in Eq. 10. Firstly, it ensures that the pressure for particles in water bodies does not affected by p_i^r, which might contain noise. Secondly, it controls the magnitude of repulsive internal pressure forces on thin features, so that they do not suppress attractive surface forces, especially surface tension. Since the surface tension force is related to the surface tension energy density, we use an empirical equation to define β:

$$\beta = \gamma \max_i(\kappa \left|\nabla_i c\right|^2) / \max_i \left(p_i^r\right). \tag{11}$$

where γ needs to be located in the range of [0, 0.5] so as to get a plausible simulation result. The use of β can be considered as a balance between surface tension effects and thin features. A smaller β makes surface tension effects more salient but destroys thin features, while a greater β preserves thin features but weakens surface tension effects. We note that β is introduced completely from the algorithmic perspective and it has no physical meanings, but it provides us a convenient way to adjust the visual effects. We will demonstrate how β can affect the fluid behavior by setting γ to different values in the results section.

4.2 Anisotropic Pressure Filtering

After we fix tensile instability using the two-scale pressure estimation method in Subsection 4.1, we now have to face the instability caused by the increased internal pressure forces. This instability problem is often exaggerated by the noise in p_i, even after the use of a small β in Eq. 10.

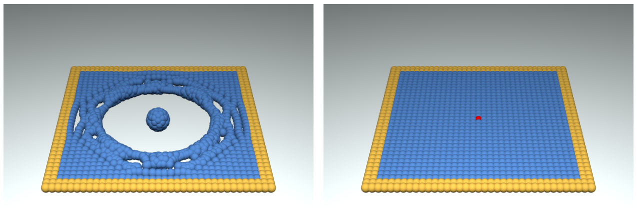

(a) Without our method (b) With our method

Fig. 6. A thin sheet. Without calculating internal pressure forces by our method, the thin sheet ruptures into water drops after small perturbation as (a) shows. Using our method, the sheet stays at rest as shown in (b).

Inspired by the anisotropic surface reconstruction method proposed by Yu and Turk [2010], we solve this instability by applying an anisotropic filter on internal pressure forces. Let $\mathbf{C}_i = \sum_j (\mathbf{x}_j - \mathbf{x}_i)(\mathbf{x}_j - \mathbf{x}_i)^{\mathsf{T}} W_{ij}^R$ be the anisotropic covariance matrix defined at particle i. We propose a tensor matrix $\mathbf{T}_i$ as:

$$\mathbf{T}_i = \frac{p_i^R}{p_i}\mathbf{I} + \left(1 - \frac{p_i^R}{p_i}\right)\frac{\mathbf{C}_i}{\|\mathbf{C}_i\|_2}. \tag{12}$$

For particles in water bodies, $p_i = p_i^R$ and $\mathbf{T}_i$ is the identity matrix. For particles on thin features, p_i is larger than p_i^R and $\mathbf{T}_i$ becomes more anisotropic. Using this tensor matrix, we then formulate the internal pressure force as:

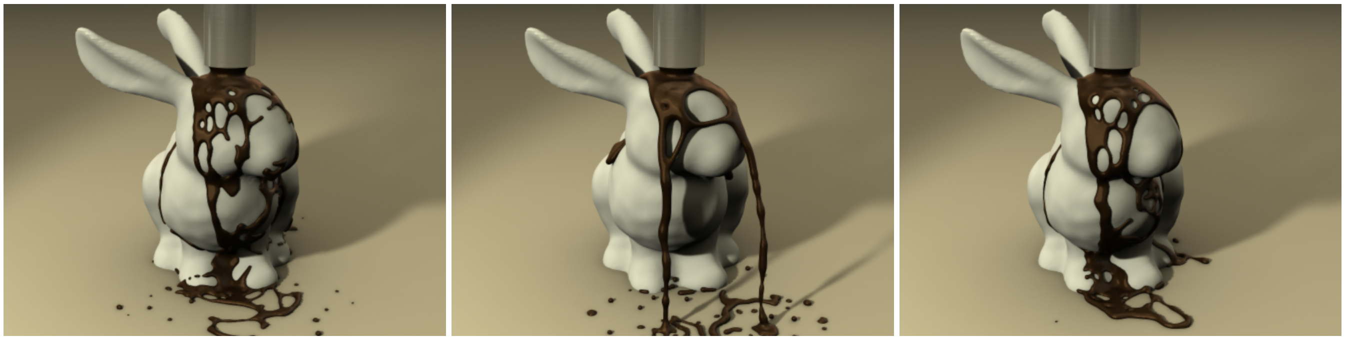
$$\mathbf{F}_i^{\mathrm{p}} = -\frac{1}{2}\sum_j V_i V_j \left(p_i \mathbf{T}_i + p_j \mathbf{T}_j\right) \cdot \nabla_i W_{ij}^h. \tag{13}$$

Intuitively, Eq. 13 diminishes the internal pressure effect in the direction perpendicular to thin features. In this way, the internal pressure force removes tensile instability, without destroying thin features by its own instability issue. For particles in water bodies, Eq. 13 is simply reduced to a standard formula with no anisotropic filtering.

5. RESULTS

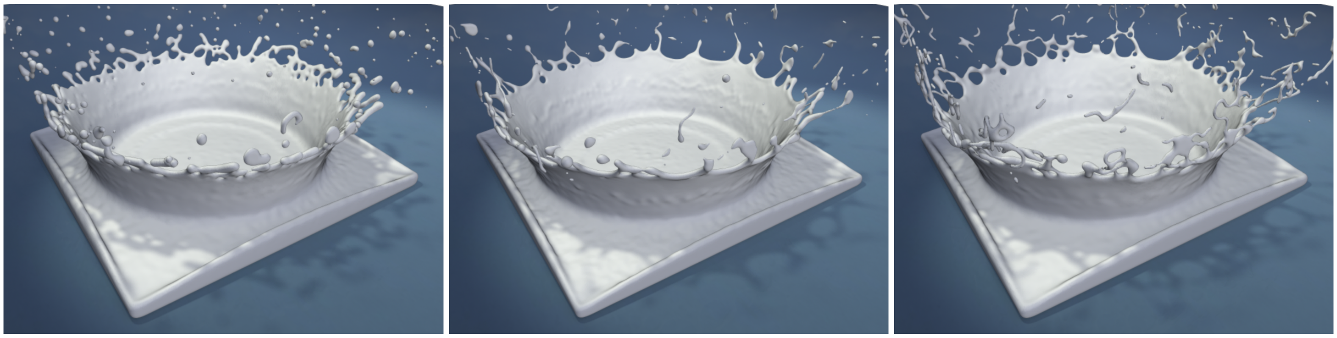
(Please refer to the supplemental video for the animation results.) We implement our methods and we integrate them into a standard SPH/WCSPH solver and all the smoothing kernels are chosen according to the work by Müller and colleagues [2003]. We use the parallel index sorting algorithm presented by Ihmsen and colleagues [2011] to construct a uniform grid and accelerate the neighborhood search. We use the level set approach proposed by Bhatacharya and collaborators [2011] for liquid surface reconstruction. To model solid objects, we uniformly create solid particles close to solid surfaces and treat them as ghost liquid particles. In this case, we ignore the solid-liquid surface tension and treat the solid-air surface tension as the liquid-air surface tension. We test our system on a quad-core Intel Xeon W3550 3.07GHz workstation with 6GB memory. We set the time step as $\Delta t = 0.001$s in simulation. Table I shows the coefficients and timings (per time step) of our examples, including the number of particles N, the surface tension efficient κ, the average computational time for surface forces t^{s}, the average computational time for internal pressures t^{p}, and the total computational time t^{tot}.

Water jet. Fig. 9 compares the simulation effects of using different surface tension coefficients. In this example, a water jet breaks up into small water drops with the same volume but less surface area. Commonly known as *Plateau-Rayleigh instability*, this effect



(a) Without the surface tension force (b) Without the air pressure force (c) With both surface forces

Fig. 7. Chocolate on bunny. This example demonstrates the effects of using different surface forces. We use both the surface tension force and the air pressure force in most of the examples.



(a) $\gamma = 0$ (b) $\gamma = 0.2$ (c) $\gamma = 1.0$

Fig. 8. Water crown. This example shows how the pressure correction can affect the fluid behavior by setting γ to different values.

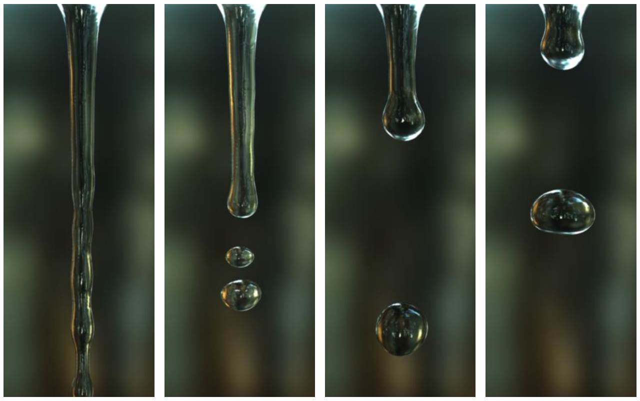
Table I. Simulation coefficients and timings

Name	N	κ (N/m)	t^{s} (ms)	t^{p} (ms)	t^{tot} (ms)
Chocolate on Bunny	191K	0.035	49	41	281
Water Jet	5.2K	0.015 – 0.09	1	2	20
Bounced Drop	235K	0.1	92	90	532
Water on Sphere	71K	0.015	26	39	174
Water Crown	348K	0.015	124	133	752

becomes more likely to happen, when the surface tension coefficient increases from left to right.

Chocolate on bunny. Fig. 7 demonstrates the simulation effects of different surface forces. Without using the surface tension force, liquid particles can leave the streamlets freely and form a number of drops, as shown in Fig. 7a. Without using the air pressure force, the streamlets do not flow on solid surfaces, as show in Fig. 7b. We obtain more physically plausible effects by using both surface forces as Fig. 7c shows.

Water crown. Fig. 8 compares the fluid behaviors of using different values of γ. In the extreme case of $\gamma = 0$, a bunch of small droplets will pinch off due to the numerical instability caused by the surface tension. In the other extreme case of $\gamma = 1$, which means the pressure will be overcorrected, the fluid fails to reveal the plausible surface tension behavior. To balance between surface tension effects and thin features, an appropriate choice is to set $\gamma = 0.2$ and we are able to get the plausible simulation result.



(a) $\kappa = 0.015$ (b) $\kappa = 0.03$ (c) $\kappa = 0.06$ (d) $\kappa = 0.09$

Fig. 9. Water jet. This example shows the Plateau-Rayleigh instability effects of using different surface tension coefficients (in N/m).

Water on sphere. In this example, we present another example to demonstrate the importance of our internal pressure force algorithm for preserving thin features as Fig. 10 shows. Here we directly set $\gamma = 0.2$ and ignore the air pressure force to make the visual difference more noticeable. Without adjusting internal pressure forces, the water film quickly ruptures into pieces. After using our technique, the water film stays in simulation until it gets destroyed by resolution limits.

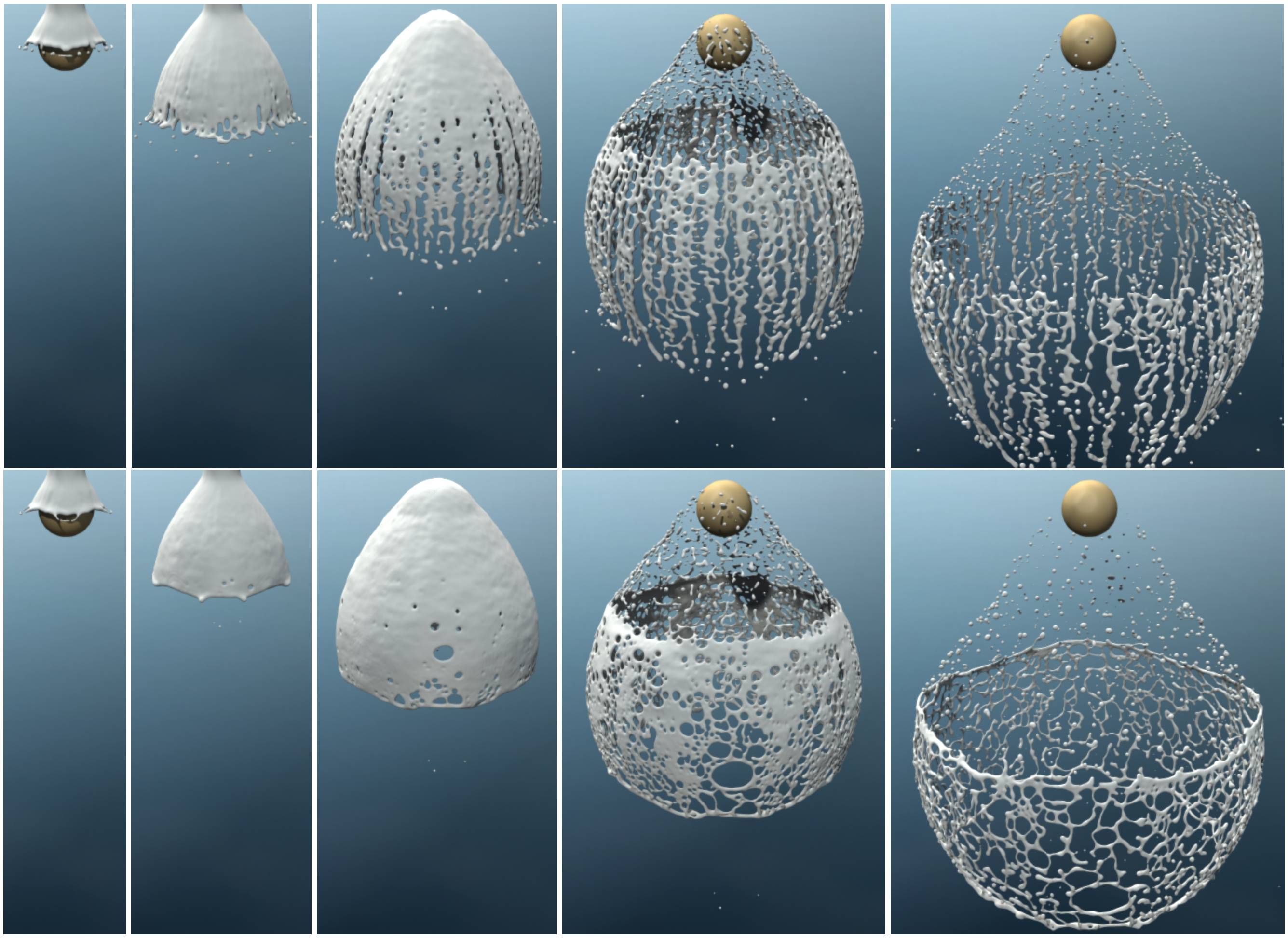


Fig. 10. Water on sphere. In this example, we compare the difference between not using (top) and using (bottom) our internal pressure force algorithm. This algorithm effectively avoid numerical instability issues, but it cannot prevent thin features from being destroyed by resolution limits.

6. LIMITATIONS

Our system does not address the resolution limit issue, so it cannot handle thin features whose sizes are less than a single particle. It cannot be used to animate air bubbles in large water bodies, since it does consider air compressibility. It does not consider solid surface properties or wetting effects either, so it cannot model complex solid influence on liquid, such as hydrophobic effects. Although our surface tension formula is more robust than the CSF and IIF methods, it is still an approximation and it is not so accurate as the ghost SPH method [Schechter and Bridson 2012]. Some of the coefficients in our algorithms (including κ and β) are not based on physics, and they need to be tuned for different examples. Finally, how to preserve thin features during the liquid surface reconstruction process is still a difficult problem, since thin features may be erroneously identified as noises. Using a smaller smoothing kernel can preserve thin features, but it may not be sufficient for removing actual noises, as we found in some examples.

7. CONCLUSION AND FUTURE WORK

In this paper, we identified that surface forces and numerical instability are the two main factors affecting small-scale thin features in SPH-based free surface flows. We demonstrated the use of the free surface energy functional in formulating surface tension forces, and we studied the possibility to handle air pressure effects without using air particles. We proposed a novel algorithm to calculate internal pressure forces on thin features, which effectively reduces numerical instability issues.

Our immediate plan next is to test the compatibility of our system with graphics hardware acceleration. We are also interested in combining it with the mesh-based tracking method [Yu et al. 2012], to simulate the effects that our current system cannot handle. In the long term, we would like to explore the possibility of using free surface flows to animate complex liquid-solid interactions and air bubbles in large water bodies, most of which can be simulated only by multiphase flows in the past.

ACKNOWLEDGMENTS

REFERENCES

Adams, B., Pauly, M., Keiser, R., and Guibas, L. J. 2007. Adaptively sampled particle fluids. *ACM Trans. Graph. (SIGGRAPH) 26,* 3 (July).

Akinci, G., Akinci, N., Ihmsen, M., and Teschner, M. 2012. An efficient surface reconstruction pipeline for particle-based fluids. In *Proc. of VRIPHYS*. 61–68.

Akinci, N., Ihmsen, M., Akinci, G., Solenthaler, B., and Teschner, M. 2012. Versatile rigid-fluid coupling for incompressible SPH. *ACM Trans. Graph. (SIGGRAPH) 31,* 4 (July), 62:1–62:8.

Ando, R., Thürey, N., and Tsuruno, R. 2012. Preserving fluid sheets with adaptively sampled anisotropic particles. *IEEE Trans. Vis. Comp. Graph. 18,* 8, 1202–1214.

Ando, R., Thürey, N., and Wojtan, C. 2013. Highly adaptive liquid simulations on tetrahedral meshes. *ACM Trans. Graph. (SIGGRAPH) 32,* 4 (July).

B., A., S., J., A., M., F., E., and L., D. 2010. Modeling surface tension in SPH by interface reconstruction using radial basis functions. In *Proc. of the 5th International SPHERIC Workshop*.

Becker, M. and Teschner, M. 2007. Weakly compressible SPH for free surface flows. In *Proc. of SCA*. 209–217.

Bhatacharya, H., Gao, Y., and Bargteil, A. 2011. A level-set method for skinning animated particle data. In *Proc. of SCA*. 17–24.

Blinn, J. 1982. A generalization of algebraic surface drawing. *ACM Trans. Graph. 1,* 3, 235–256.

Brochu, T., Batty, C., and Bridson, R. 2010. Matching fluid simulation elements to surface geometry and topology. *ACM Trans. Graph. 29,* 4, 47.

Cahn, J. and Hilliard, J. 1958. Free energy of a nonuniform system. I. interfacial free energy. *The Journal of Chemical Physics 28*, 258.

Chen, J. K., Beraun, J. E., and Jih, C. J. 1999. An improvement for tensile instability in smoothed particle hydrodynamics. *Computational Mechanics 23,* 4, 279–287.

Clausen, P., Wicke, M., Shewchuk, J. R., and O'brien, J. F. 2013. Simulating liquids and solid-liquid interactions with lagrangian meshes. *ACM Trans. Graph. 32,* 2 (Apr.), 17:1–17:15.

He, X., Liu, N., Li, S., Wang, H., and Wang, G. 2012. Local poisson SPH for viscous incompressible fluids. *Comp. Graph. Forum 31,* 6, 1948–1958.

Hu, X. and Adams, N. 2006. A multi-phase SPH method for macroscopic and mesoscopic flows. *Journal of Computational Physics 213,* 2, 844–861.

Ihmsen, M., Akinci, N., Becker, M., and Teschner, M. 2011. A parallel SPH implementation on multi-core CPUs. *Comp. Graph. Forum 30,* 1, 99–112.

Ihmsen, M., Akinci, N., Gissler, M., and Teschner, M. 2010. Boundary handling and adaptive time-stepping for PCISPH. In *Proc. of VRIPHYS*. 79–88.

Irving, G., Guendelman, E., Losasso, F., and Fedkiw, R. 2006. Efficient simulation of large bodies of water by coupling two and three dimensional techniques. *ACM Trans. Graph. (SIGGRAPH) 25,* 3 (July), 805–811.

Kim, B., Liu, Y., Llamas, I., Jiao, X., and Rossignac, J. 2007. Simulation of bubbles in foam with the volume control method. *ACM Trans. Graph. (SIGGRAPH) 26,* 3 (July).

Losasso, F., Gibou, F., and Fedkiw, R. 2004. Simulating water and smoke with an octree data structure. *ACM Trans. Graph. (SIGGRAPH) 23,* 3 (Aug.), 457–462.

Monaghan, J. 2000. Sph without a tensile instability. *Journal of Computational Physics 159,* 2, 290C–311.

Monaghan, J. J. 1989. On the problem of penetration in particle methods. *Journal of Compututational Physics 82,* 1 (May), 1–15.

Monaghan, J. J. 1994. Simulating free surface flows with SPH. *Journal of Computational Physics 110,* 2 (Feb.), 399–406.

Morris, J. 2000. Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids 33,* 3, 333–353.

Müller, M., Charypar, D., and Gross, M. 2003. Particle-based fluid simulation for interactive applications. In *Proc. of SCA*. 154–159.

Müller, M., Solenthaler, B., Keiser, R., and Gross, M. 2005. Particle-based fluid-fluid interaction. In *Proc. of SCA*. 237–244.

Nugent, S. and Posch, H. 2000. Liquid drops and surface tension with smoothed particle applied mechanics. *Physical Review E 62,* 4, 4968.

Schechter, H. and Bridson, R. 2012. Ghost SPH for animating water. *ACM Trans. Graph. (SIGGRAPH) 31,* 4 (July), 61:1–61:8.

Sirotkin, F. and Yoh, J. 2011. A new particle method for simulating breakup of liquid jets. *Journal of Computational Physics*.

Solenthaler, B. and Gross, M. 2011. Two-scale particle simulation. *ACM Trans. Graph. (SIGGRAPH) 30,* 4 (July), 81:1–81:8.

Solenthaler, B. and Pajarola, R. 2008. Density contrast SPH interfaces. In *Proc. of SCA*. 211–218.

Solenthaler, B. and Pajarola, R. 2009. Predictive-corrective incompressible SPH. *ACM Trans. Graph. (SIGGRAPH) 28,* 3 (July), 40:1–40:6.

Sussman, M. and Ohta, M. 2009. A stable and efficient method for treating surface tension in incompressible two-phase flow. *SIAM J. Sci. Comput. 31,* 4 (June), 2447–2471.

Swegle, J., Hicks, D., and Attaway, S. 1995. Smoothed particle hydrodynamics stability analysis. *Journal of computational physics 116,* 1, 123–134.

Tartakovsky, A. and Meakin, P. 2005. Modeling of surface tension and contact angles with smoothed particle hydrodynamics. *Physical Review E 72,* 2, 026301.

Thürey, N., Wojtan, C., Gross, M., and Turk, G. 2010. A multiscale approach to mesh-based surface tension flows. *ACM Trans. Graph. (SIGGRAPH) 29,* 4 (July), 48:1–48:10.

van der Waals, J. 1893. The thermodynamic theory of capillarity under the hypothesis of a continuous variation of density. *Z. Phys. Chem 13*, 657.

Wojtan, C., Thürey, N., Gross, M., and Turk, G. 2010. Physics-inspired topology changes for thin fluid features. *ACM Trans. Graph. (SIGGRAPH) 29*, 50:1–50:8.

Yu, J. and Turk, G. 2010. Reconstructing surfaces of particle-based fluids using anisotropic kernels. In *Proc. of SCA*. 217–225.

Yu, J., Wojtan, C., Turk, G., and Yap, C. 2012. Explicit mesh surfaces for particle based fluids. *Comp. Graph. Forum (Eurographics) 31,* 4 (May), 815–824.

Zhang, M. 2010. Simulation of surface tension in 2d and 3d with smoothed particle hydrodynamics method. *J. Comput. Phys. 229,* 19 (Sept.), 7238–7259.

Zhang, Y., Wang, H., Wang, S., Tong, Y., and Zhou, K. 2011. A deformable surface model for real-time water drop animation. *IEEE Trans. Vis. Comp. Graph. 99*.

Zhu, Y. and Bridson, R. 2005. Animating sand as a fluid. *ACM Trans. Graph. (SIGGRAPH) 24,* 3 (July), 965–972.

Received Month YYYY; accepted Month YYYY