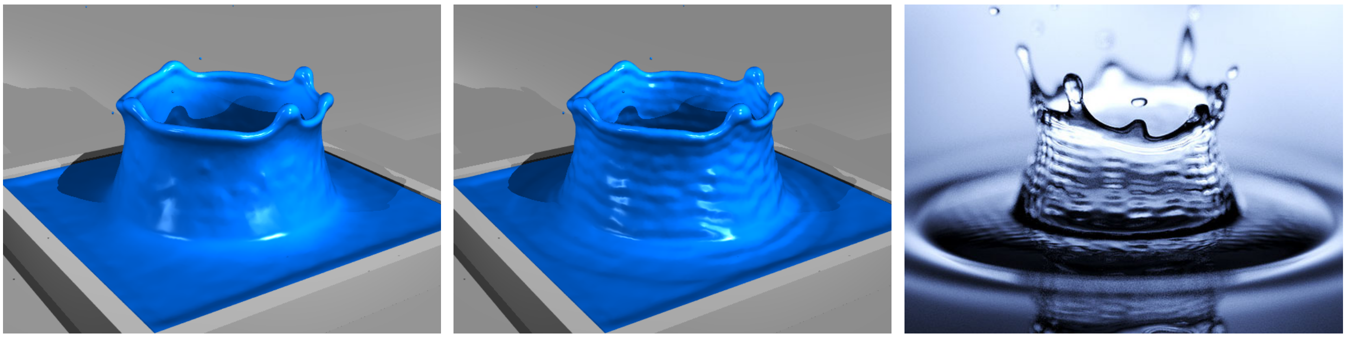
Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2016)
Ladislav Kavan and Chris Wojtan (Editors)

Enriching SPH Simulation by Approximate Capillary Waves

Sheng Yang†[1], Xiaowei He†[2,5], Huamin Wang[3], Sheng Li‡[1], Guoping Wang[1], Enhua Wu[2,6], and Kun Zhou[4]

[1]Peking University [2]State Key Lab. of CS, ISCAS [3]The Ohio State University [4]Zhejiang University [5]UCAS [6]University of Macau



(a) Without enrichment (b) With enrichment (c) Photograph

Figure 1: *Water crown. SPH simulation has difficulty in generating and maintaining capillary waves, as shown in (a). By approximating capillary waves using surface compression waves, our enrichment algorithm efficiently generates realistic capillary waves in (b), similar to the real waves in a photograph (c).*

Abstract

Capillary waves are difficult to simulate due to their fast traveling speed and high frequency. In this paper, we propose to approximate capillary wave effects by surface compression waves under the SPH framework. To achieve this goal, we present a method to convert surface tension energy changes measured from SPH simulation into high-frequency density variations. Based on the compression wave propagation model, we present an approximate technique to simulate capillary wave propagation in a high-frequency particle density field. To address noise issues in wave simulation, we develop a simple way to apply the zero pressure condition on free surfaces in projection-based incompressible SPH. Our experiment shows that the developed algorithm can produce realistic capillary wave effects on both thin liquid features and large liquid bodies. Its computational overhead is also small.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Capillary waves, commonly known as ripples, are ubiquitous in small-scale liquid behaviors. Different from gravity waves, capillary waves are caused by surface tension, which is especially large in water. Because of that, they have high speed and frequency in the real world, making them difficult to simulate in physically based liquid simulation. Previously, graphics researchers [WMT07, ATBG08, TWGT10, YWTY12, JW15] have successfully addressed this problem using mesh-based surface representations. Unfortunately, tracking mesh-based liquid surfaces is a challenging problem, due to complex topological changes.

Meshless techniques, such as smoothed particle hydrodynamics (SPH) and Eulerian simulation, are naturally free of topological issues. Since they have difficulty in handling capillary waves due to numerical instability or dissipation, an interesting question is: *can we enrich simulated results with wave details?* This detail enrichment idea is not new and it has been explored before for several natural phenomena, including liquid animation [JKB*10, NSCL08]. But accurately incorporating dynamic waves into simulated low-frequency results is not easy, since the motion of these waves should be simulated independently, as Kim and collaborators [KTT13] pointed out. To solve this problem, they built a 3D

† Sheng Yang and Xiaowei He are joint first authors.

‡ e-mail: lisheng@pku.edu.cn

DOI: 10.2312/sca.20161220

extension field around liquid surfaces and used the closest point method to simulate surface waves. Recently, Solenthaler and colleagues [SBC*11] proposed to separate 2D horizontal liquid motion from 1D vertical motion, so they can enrich 2D SPH simulation by height-field-based waves. Although these techniques have demonstrated their effectiveness in handling gravity waves, they cannot be easily extended to animate capillary waves. It is not only because these two types of waves have different wave equations, but also because they occur in different situations. Gravity waves are typically noticeable on the free surface of a large water body. In contrast, capillary waves can occur anywhere, from ripples in a coffee mug to a splashing water crown in Figure 1. So many data structures used in previous methods are not easily applicable to capillary waves.

In this paper, we study how to enrich SPH with capillary wave details. We choose SPH rather than others as our basic framework, because SPH can effectively represent small liquid features with a low memory cost. Our general strategy is to run SPH simulation first, and then simulate dynamic capillary waves on top of it. The challenge is: *how can we approximately model and simulate capillary waves?* Since SPH is sensitive to particle distribution, if we still model capillary waves on particle positions, we must use small time steps and a significant number of particles inevitably, to prevent waves from being quickly smeared out. Fortunately, besides gravity waves and capillary waves, there are a third type of fluid waves: *compression waves*, or commonly known as *sound waves*. Graphics researchers [RSM*10, MRA*13] have investigated the simulation of compression waves to model sound propagation in virtual environment. Similar to capillary waves, compression waves are fast. Because they travel within the medium rather than on the surface, they can be reliably simulated without numerical issues on surface.

The basic idea behind this work is to generate capillary wave effects by simulating compression wave propagation instead. Specifically, we model compression waves by a *high-frequency density field* defined over SPH particles. This field accounts for high-frequency density variation caused by wave propagation, which cannot be captured by the original density field in SPH simulation. Based on this representation, we make a series of technical contributions toward the development of our algorithm:

- **Wave propagation**. Based on the compression wave propagation model, we develop an implicit surface wave simulation approach to approximate capillary wave propagation in the high-frequency density field.
- **Seeding**. We present a new method to create high-frequency density variations as wave seeds, using the surface tension energy change calculated in SPH simulation.
- **Zero pressures**. We propose a simple way to handle free surface boundary condition in projection-based incompressible SPH, which provides better performances in both robustness and accuracy.

Our experiment shows that the resulting algorithm can generate realistic capillary wave effects on a variety of liquid surfaces, such as water crown (Figure 1), streamlet (Figure 9), thin water sheet (Figure 10), and gravity-free water spheres (Figure 4). It has a small computational overhead, thanks to the reusability of the neighborhood search provided by SPH.

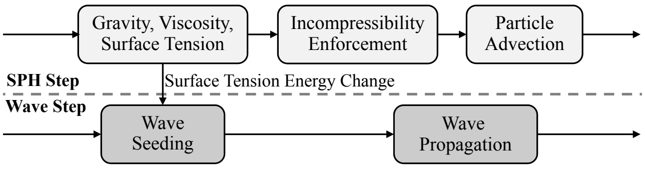

Gravity, Viscosity, Surface Tension
Incompressibility Enforcement
Particle Advection
SPH Step
Surface Tension Energy Change
Wave Step
Wave Seeding
Wave Propagation

Figure 2: *Algorithm pipeline. Our algorithm contains two steps: an SPH simulation step and a capillary wave simulation step. The algorithm transfers the surface tension energy change in SPH simulation into wave simulation as wave seeds.*

2. Other Related Work

Fluid simulation is a large graphics research topic, and we will focus our discussions on smoothed particle hydrodynamics (SPH) only. Researchers have explored the use of SPH in simulating various fluid effects, such as water bodies [MCG03, SP09], multiphase fluids [MSKG05, SP08], and solid-fluid coupling [AIA*12, HLW*12].

SPH-based surface tension. Researchers are interested in modeling surface tension under the SPH framework to simulate small-scale liquid behaviors. Most early work [Mor00, MCG03, HA06] was focused on the continuum surface force (CSF) approach. The problem with CSF is that it relies heavily on particle samples in free surface cases. If there are not enough particles, numerical instability and inaccuracy will become problematic in simulation results. Alternatively, researchers modeled surface tension at the microscopic level as inter-molecular forces among particles [TM05, BT07, AAT13]. Recently, He and colleagues [HWZ*14] developed a new surface tension model based on the diffuse interface method, to handle sparsely sampled free surfaces. Compared with other surface tension effects, capillary waves are especially difficult to simulate, since they require consistent and accurate surface tension estimation. No existing SPH technique can generate detailed capillary wave effects, as far as we know.

SPH-based incompressibility. Early work on SPH simulation used explicit time integration to handle incompressibility of fluid by gas equation [MFZ97, MCG03, CBP05] or Tait's equation [Mon94, BT07]. Since explicit time integration requires stringent time-stepping, researchers [SP09, HLL*12, MM13] proposed to enforce incompressibility by adjusting particle positions iteratively. Cummins and Rudman [CR99] and Shao and Lo [SL03] applied the pressure projection method on SPH particles by solving a Poisson equation. Recently, Ihmsen and colleagues [ICS*14] developed an implicit scheme to more accurately solve particle pressures for incompressibility. For density-based methods, we refer to Ihmsen and colleagues [IOS*14]'s for a thorough review.

When using SPH to simulate free surface flows, an interesting question is how can we specify the zero pressure condition. A typical strategy is to detect liquid particles near free surfaces and assign them with zero pressures. When free surfaces are not densely sampled, this strategy can cause small particle oscillations, which may not be obvious in SPH simulation but can trigger noisy wave patterns in wave simulation. Although this issue has been addressed by the ghost SPH method developed by Schechter and Bridson [SB12], we prefer to use a simpler and faster method in our system.

```
Algorithm 1 Enriched_SPH($\mathbf{X}, \mathbf{V}, \Delta t$)
  for every particle $i$ do
      $\hat{\rho}_i \leftarrow \rho_0$;
      $\dot{\hat{\rho}}_i \leftarrow 0$;
      Construct its neighborhood set $J_i$;
  for every time step do
      for every particle $i$ do                                        ▷ The SPH step
          Compute its gravity, viscosity, and surface tension forces;
          Update its velocity $\mathbf{v}_i$ by the forces;
      Compute the pressure $p$ by pressure projection;                 (Eq. 13)
      for every particle $i$ do
          Compute its pressure force $\mathbf{F}_i^{\mathrm{p}}$;                          (Eq. 17)
          Update its velocity $\mathbf{v}_i$ by $\mathbf{F}_i^{\mathrm{p}}$, and its position $\mathbf{x}_i$;
      for every particle $i$ do                                        ▷ The wave step
          Update its neighborhood set $J_i$;
          Add new wave seeds into $\hat{\rho}_i$;                           (Eqs. 12 and 11)
      Update $\hat{\rho}$ by solving wave propagation;                   (Eq. 6)
      for every particle $i$ do
          Update $\dot{\hat{\rho}}_i$;                                         (Eq. 8)
```

3. Algorithm Description

The pipeline of our enriched SPH simulation algorithm is shown in Figure 2 and the pseudocode is provided in Algorithm 1. Given a set of particles with initial positions $\mathbf{X}$ and velocities $\mathbf{V}$, the algorithm first assigns $\hat{\rho}_i$ and $\dot{\hat{\rho}}_i$ to each particle i, representing the high-frequency particle density and density changing rate, respectively. We note that $\hat{\rho}$ is different from the particle density obtained in SPH simulation, which will be denoted as ρ in this paper. Our algorithm runs simulation in two steps. In the first step, it runs incompressible SPH with pressure projection to simulate low-frequency particle motions. Here we present a novel method to apply zero pressures on free surface, to improve stability due to particle deficiency. Since the first step is unable to handle capillary waves well, the algorithm uses the second step to re-simulate capillary wave effects in the high-frequency density field. Specifically, it calculates the surface tension energy change at each particle, and uses it to adjust the high-frequency particle density as a capillary wave seed. It then applies our surface compression wave model to simulate wave propagation by updating $\hat{\rho}$ and $\dot{\hat{\rho}}$. Finally, we incorporate $\hat{\rho}$ into the surface reconstruction process to demonstrate wave effects in the reconstructed liquid surface meshes.

The neighborhood search is often the computational bottleneck in many SPH algorithms, including ours. Fortunately, since the wave step does not update particle positions, we can use the same neighborhood search result for both the SPH step and the wave step. So the wave step has a small computational overhead.

3.1. Capillary Wave Propagation

In this subsection, we will describe the time evolution of the high-frequency density field $\hat{\rho}$ by an implicit wave propagation model. Later in Subsection 3.2, we will discuss how new density variations can be introduced from SPH simulation into $\hat{\rho}$ as wave seeds.

Since we do not modify particle positions in wave simulation, we assume that particles are static samples and we use the acoustic wave equation to update density variations caused by compression or decompression:

$$\frac{D^2\hat{\rho}}{Dt^2} = c_0^2\nabla^2\hat{\rho}, \tag{1}$$

in which c_0 is the wave speed. Although capillary waves travel faster than gravity waves, they are still slower than sound waves, so we do not use the speed of sound as c_0. Instead, we use the dispersion relation of capillary waves to calculate the phase velocity:

$$c_0 = \omega/k = \sqrt{\gamma k/\rho_0}, \tag{2}$$

in which ω is the angular frequency, k is the wavenumber, γ is the surface tension, and ρ_0 is the reference density. For water, we use $\rho_0 = 1.0 \times 10^3$ kg/m^3, so we typically use c_0=0.5 to 1.0m/s according to the magnitude of surface tension.

Although capillary waves and sound waves have similar behaviors, they are still different: capillary waves travel on surface only, while sounds waves can travel within liquid volume. If we simply solve Equation 1, we will see unnatural or faster capillary wave propagation, especially when we handle large water bodies. A simple solution to this problem is to explicitly detect surface particles and run wave simulation on them only. Unfortunately, this can lead to noisy artifacts due to error and discontinuity in particle detection. Instead, we introduce an additional Laplacian smoothing process to model energy dissipation at particle i:

$$\frac{D\hat{\rho}(\mathbf{x}_i)}{Dt} = (\nu_{\max} + b_i(\nu_{\min} - \nu_{\max}))\nabla^2\hat{\rho}(\mathbf{x}_i), \tag{3}$$

in which $\nu_{\max}$ and $\nu_{\min}$ are two positive coefficients and b_i is a surface particle indicator. If i is a surface particle (for $b_i = 1$), we use $\nu_{\min}$ to account for small capillary wave energy loss occurred over surface. When i is an interior particle (for $b_i = 0$), we use $\nu_{\max}$ to quickly diminish high-frequency density variations. The bunny example in Figure 3b shows that this method can effectively prevent compression waves from penetrating though the body. Since the method does not directly modify density values, its result is much less sensitive to accuracy of surface particle detection. Here we define b_i as:

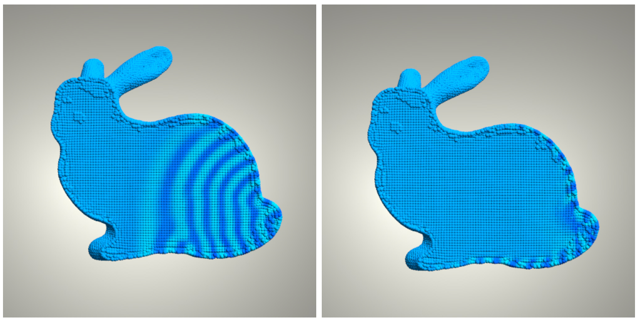
$$b_i = \begin{cases} 0, & \|\nabla\rho(\mathbf{x}_i)\|^2 < B_0 \text{ and } \rho_i > B_1, \\ 1, & \text{otherwise.} \end{cases} \tag{4}$$

in which B_0 and B_1 are two threshold variables and,

$$\nabla\rho(\mathbf{x}_i) = \frac{\rho_0 \sum_j V_j \nabla_i W_{ij}}{\sum_j V_j W_{ij}}, \tag{5}$$

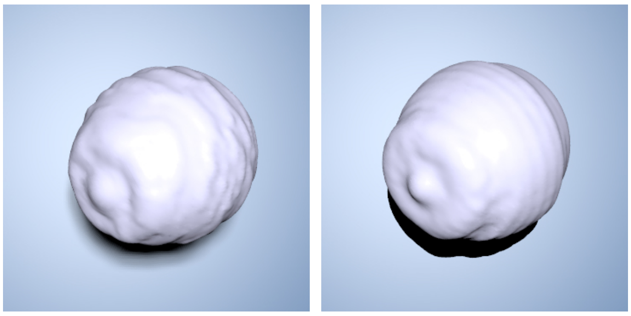
where V_i is the particle volume and W_{ij} is the smoothing kernel function with a smoothing length h: $W_{ij} = W(\|\mathbf{x}_i - \mathbf{x}_j\|, h)$. Equation 5 is evaluated using SPH particle positions, not the high-frequency density field $\hat{\rho}$. We note that we use two conditions in Equation 4 to ensure the detection of a surface particle. Although some interior particles may be falsely detected as surface particles, they are unlikely to cause artifacts, since density variations will be dissipated even before reaching them.

Discretization. Equation 1 is a second-order partial differential equation. To discretize it, we define an additional field $\dot{\hat{\rho}}$ over particles to model high-frequency density changing rate. Similar



(a) Compression wave (b) Surface compression wave

Figure 3: *Wave propagation over a static bunny. Instead of limiting wave propagation to surface particles only, our energy dissipation method diminishes high-frequency density variations within the body. Its result is less sensitive to surface particle detection.*



(a) Zero pressures on particles (b) Our zero-pressure approach

Figure 4: *Merging spheres. (a) Obvious noise issues can be noticed in the wave simulation when we apply the position-based method as an SPH solver. By using our zero-pressure approach in projection-based incompressible SPH, we can effectively avoid noise issues in wave simulation result, as shown in (b).*

to [TWGT10], we use the implicit Newmark-β method with $\beta = \frac{1}{4}$. By incorporating Equation 3 into Equation 1, we obtain the following implicit time integration formula:

$$\hat{\rho}_i^{t+1} = \hat{\rho}_i^t + \Delta t \dot{\hat{\rho}}_i^t + C_i\left(\nabla^2\hat{\rho}_i^t + \nabla^2\hat{\rho}_i^{t+1}\right), \tag{6}$$

where Δt is the time step and $C_i = \frac{1}{4}\Delta t^2 c_0^2 + \frac{1}{2}\Delta t(\nu_{\max} + b_i(\nu_{\min} - \nu_{\max}))$. Using the approximate rule proposed by Shao and Lo [SL03], we discretize the Laplacian operator with its coefficient in a mass-preserving way:

$$C_i\nabla^2\hat{\rho}_i = \sum_j 4m_j\frac{C_i+C_j}{(\rho_i+\rho_j)^2}\left(\hat{\rho}_j - \hat{\rho}_i\right)W'_{ij}, \tag{7}$$

in which m_j is the mass of particle j and $W' = \partial W/\partial d$ is the derivative of the smoothing kernel with respect to the distance. In our implementation, we treat all particles as having equal mass. Therefore, the linear system constructed from Equations 6 and 7 is symmetric positive definite. After we update $\hat{\rho}_i$, we renew $\dot{\hat{\rho}}_i$ by:

$$\dot{\hat{\rho}}_i^{t+1} = \frac{2}{\Delta t}\left(\hat{\rho}_i^{t+1} - \hat{\rho}_i^t\right) - \dot{\hat{\rho}}_i^t. \tag{8}$$

3.2. Capillary Wave Seeding

Given the approximate capillary wave propagation model in Subsection 3.1, we now would like to study how to generate capillary wave sources. Our basic idea is to measure surface tension energy variations in SPH simulation, and adjust $\hat{\rho}$ accordingly. The surface tension energy of a diffuse interface can be defined as a Helmholtz free energy functional [CH58]:

$$E_{\mathrm{s}} = \int_\Omega \rho_0 \frac{1}{2}\kappa\|\nabla\rho\|^2 dV, \tag{9}$$

in which Ω is the liquid domain and κ is the squared gradient energy coefficient. Under the SPH framework, Equation 9 can be discretized into $E_{\mathrm{s}} = \sum_i e_{\mathrm{s},i}$:

$$e_{\mathrm{s},i} = \frac{1}{2}\kappa m_i\|\nabla\rho(\mathbf{x}_i)\|^2, \tag{10}$$

in which $\nabla\rho(\mathbf{x}_i)$ is calculated using Equation 5.

In our system, the surface tension energy are essentially used twice: once by SPH in computing the surface tension force, and once by wave equations in seedling. When the SPH-based surface tension energy at particle i is changed by $\Delta e_{\mathrm{s},i} = e_{\mathrm{s},i}^{t+1} - e_{\mathrm{s},i}^t$ from t to $t+1$, we expect to see the same change at particle i in wave simulation as well. Since the wave propagation model described in Subsection 3.1 is based on sound waves rather than capillary waves, we need to convert the surface tension energy change into the high-frequency density change. Jamet and colleagues [JTB02] pointed out that $\Delta\hat{p}_i = -\Delta e_{\mathrm{s},i}$ for van der Waals fluids. According to [MFZ97], we have $\hat{p} = c_0^2\hat{\rho}$ and $c_0^2\Delta\hat{\rho}_i = -\Delta e_{\mathrm{s},i}$. Although we can use this equation to update $\hat{\rho}_i$ directly, we prefer not to do so since it is not mass-preserving. The resulting artifact can be noticed as wave amplitudes become greater and greater over time. Instead, we assume that the density change is immediately propagated once by Equation 1. So we update $\hat{\rho}_i$ as:

$$\hat{\rho}_i^{\mathtt{new}} = \hat{\rho}_i^t + \frac{1}{2}\Delta t^2 c_0^2\nabla^2\left(\hat{\rho}_i^t - \frac{1}{c_0^2}\Delta e_{\mathrm{s},i}\right). \tag{11}$$

Since the Laplacian operator guarantees the mass exchanges between neighboring particles are equal, Equation 11 must also be mass-preserving.

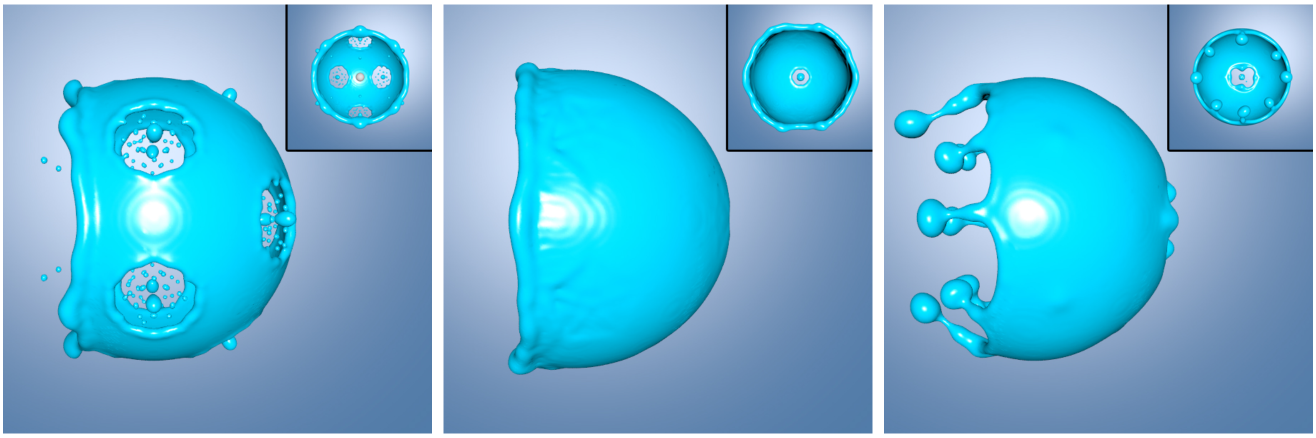
Noise reduction. To eliminate noisy artifacts, we propose to replace $\Delta e_{\mathrm{s},i}$ in Equation 11 by:

$$\Delta e_{\mathrm{s},i}^{\mathtt{imp}} = b_i\mathcal{H}\left(\left|\Delta e_{\mathrm{s},i}\right| - e_0\right)\Delta e_{\mathrm{s},i}, \tag{12}$$

in which e_0 is a threshold on the surface tension energy change, b_i is given by Equation 4 and $\mathcal{H}(\cdot)$ is the Heaviside step function. Equation 12 serves two purposes. First, it uses b_i to prevent capillary wave seeds from being generated at interior liquid surfaces. Second, it adjusts the magnitudes of capillary wave seeds, so that they are generated only when $\left|\Delta e_{\mathrm{s},i}\right| > e_0$.

3.3. Zero Pressure Condition in Incompressible SPH

Since surface tension energy estimation is sensitive to particle distribution near free surfaces, we find most density-based SPH solvers cannot be straightforwardly applied as the basic SPH solver for wave simulations due to the noise problem (see Figure 4a as an example). Another problem of the density-based SPH solvers is that they can not hold attractive pressure forces because the pressure is always clamped to be nonnegative to avoid stability issues [ICS*14, MM13]. This treatment will not cause any problem



(a) Position-based method (b) ISPH with zero pressures on particles (c) ISPH with our zero-pressure approach

Figure 5: *A bursting bubble. This example shows a liquid bubble, which is discretized into a single layer of particles, being hit by a bullet. The surface tension coefficient is set to* 0.001 *for all the three cases.*

for splashy scenarios, but can fail to create the "fingering" effect as Figure 5(c) and 6(b) demonstrate. Therefore, we choose the divergence-based pressure projection scheme [CR99] to solve fluid incompressibility. To avoid the ambiguous detection for surface particles, we borrow the idea from [NT14] to compensate the pressure coefficient for missing air particles. Besides, we apply the virtual air particle method developed by He and colleagues [HWZ*14] to calculate the pressure force, which proves to be both efficient and accurate.

Pressure projection. The first step in pressure projection is to find the pressure field p by solving [CR99]:

$$\nabla \cdot \left(\frac{1}{\rho}\nabla p\right) = \frac{1}{\Delta t}\nabla \cdot \mathbf{v}. \tag{13}$$

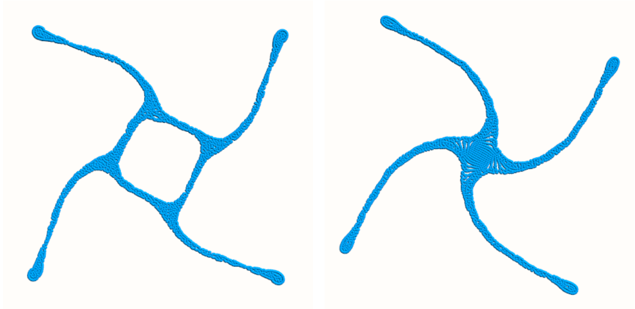
We discretize the left side into: $\sum_j \frac{8m_j}{(\rho_i+\rho_j)^2}\left(p_i - p_j\right)W'_{ij}$. If the liquid surface is surrounded by ghost air particles, the neighborhood of a surface particle must contain both liquid particles J and ghost air particles J^{a}. Since an air particle has zero pressure, we formulate the left side at surface particle i into:

$$\mathbf{LHS}_i = p_i \sum_{j\in J\bigcup J^{\mathrm{a}}} \frac{8m_j}{(\rho_i+\rho_j)^2} W'_{ij} - \sum_{j\in J} \frac{8m_j}{(\rho_i+\rho_j)^2} p_j W'_{ij}. \tag{14}$$

To avoid ghost air particles, we set the first term as $p_i\alpha_0$ if $\sum_{j\in J} \frac{8m_j}{(\rho_i+\rho_j)^2} W'_{ij} < \alpha_0$ [NT14], in which α_0 is constant estimated from a typical liquid particle fully surrounded by uniformly distributed particles. It means if a fluid particle is near the surface and lack some neighbors, we need to compensate p_i's coefficient for missing neighbors if no ghost air particles are included during simulation. Note that J is the set of surrounding liquid particles that are found by neighborhood search. So Equation 14 is applicable to both interior particles and surface particles. We can apply the same idea to the right side of Equation 13. Assuming that ghost particles have the same velocity as the surface particle, we discretize the right side into:

$$\mathbf{RHS}_i = \frac{1}{\Delta t} \sum_{j\in J} \frac{m_j}{\rho_j}\left(\mathbf{u}_i - \mathbf{u}_j\right)\cdot \nabla_i W_{ij}. \tag{15}$$

Now both equation 14 and 15 require no special treatment for free surface particles and we can use surrounding liquid particles to evaluate Equation 13 immediately.



(a) Zero pressures on particles (b) Our zero-pressure approach

Figure 6: *A rotating square. This example shows that our approach can preserve the liquid boundary better than a standard approach which detects surface particles and applies zero pressures on them.*

Pressure Force calculation. The second step is to obtain the particle pressure force. SPH simulators can calculate the pressure force as [MCG03]: $\mathbf{F}^{\mathrm{p}}_i = \sum_j \frac{m_j}{\rho_j}\left(p_i - p_j\right)\nabla_i W_{ij}$. Assuming that a liquid surface particle is surrounded by both liquid particles and ghost air particles, and they are ideally distributed, we have:

$$\sum_{j\in J\bigcup J^{\mathrm{a}}} \frac{m_j}{\rho_j}(p_i)\nabla_i W_{ij} = 0. \tag{16}$$

Using Equation 16 and the fact that $p_j = 0$ for any $j \in J^{\mathrm{a}}$, we can formulate the pressure force at particle i as:

$$\mathbf{F}^{\mathrm{p}}_i = -\sum_{j\in J} \frac{m_j}{\rho_j} p_j \nabla_i W_{ij}. \tag{17}$$

However, for particles with $\sum_{j\in J} \frac{8m_j}{(\rho_i+\rho_j)^2} W'_{ij} \geq \alpha_0$, we would still apply the momentum-preserving formulation to calculate the pressure force.

One issue remaining unsolved is the density drifting problem. There are two typical ways to solve this problem. One is to add a

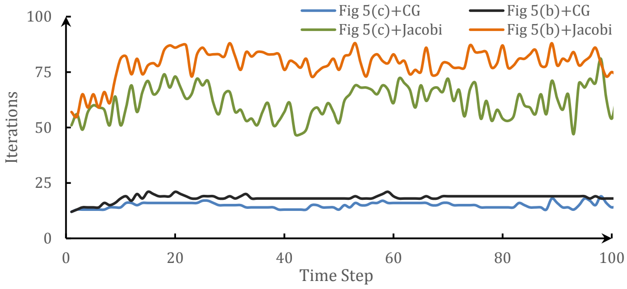


Figure 7: *Performance comparison in solving the pressure Poisson equation 13. Our zero pressure approach can significantly accelerate the convergence rate for both the conjugate gradient (CG) method and the Jacobi method, compared to ISPH with zero pressures on particles. The residual error is set to* 10^{-6}.

term $\beta(\rho_0 - \rho_i^*)$ to right hand side of Equation 13 to those particles whose temporal densities ρ^* are greater than the reference density [LTKF08]. The other is to correct the density with some density based methods [RWT11]. In our implementation, we use the latter strategy and apply the position based method [MM13] to correct the density. Since the incompressibility has already been guaranteed in equation 13, we only apply 3 iterations at each timestep.

Comparisons. Figure 6 compares our approach with a standard approach, which applies zero pressures on detected surface particles directly. This rotating square example shows that our method handles free surface flows more robustly. Besides, Figure 7 shows that our zero-pressure treatment can significantly accelerate the convergence rate in solving the pressure Poisson equation. Figure 5 shows another comparison to demonstrate the robustness and accuracy of our our zero-pressure approach, compared to both a density-based SPH solver [MM13] and ISPH with zero pressures on particles. After enriched with capillary waves, our method can effectively reduce the noise issues in wave simulation results as shown in Figure 4b.

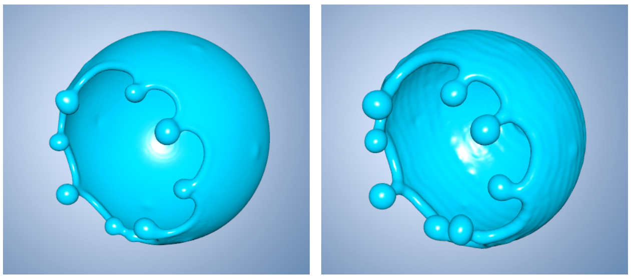
4. Surface Reconstruction

To make this paper self-contained, we would like to describe how to reconstruct liquid surfaces with capillary wave effects for final animation production. Our basic strategy is to reconstruct liquid surface meshes by a standard particle skinning algorithm (such as [BGB11]) first, and then use the high-frequency density field to modify vertex positions for capillary wave effects.

Assuming that a particle is in spherical shape, we define its radius as a function of its pressure: $r_i(\rho_i) = (3m_i/(4\pi\rho_i))^{\frac{1}{3}}$, and we calculate the radius change caused by the use of high-frequency density as: $\Delta r_i = r_i(\hat{\rho}_i) - r_i(\rho_0)$. Intuitively, Δr_i indicates how thinner or thicker the surface should be due to the high-frequency particle density. Let I be a reconstructed mesh vertex with normal $\mathbf{n}_I$, we apply an offset to its position $\mathbf{x}_I$:

$$\Delta \mathbf{x}_I = \delta \left(\frac{\sum_i \Delta r_i W_{Ii}}{\sum_i W_{Ii}} \right)^{\gamma} \mathbf{n}_I, \tag{18}$$

in which i is a SPH particle close to I. We typically use $\delta = 0.5$ and $\gamma = 1.0$ in our experiment. We note that the wave simulation step is independent of surface reconstruction.



(a) Without enrichment (b) With enrichment

Figure 8: *A bursting bubble enriched with capillary waves.*

Although our final result quality relies on the quality of surface reconstruction, it dose not mean that our wave enriching method requires the surface reconstruction method to be able to create the perfect smooth surface. As we can see from the video, the bumpy artifacts moves much slower than the wave, so we can easily distinguish them from the capillary waves. Therefore, other particle skinning methods (e.g. [AIAT12, YT13]) will also work in our method.

5. Results and Discussions

(Please see the supplemental video for more animation examples.) We test the performance of our system on a quad-core Intel Core 3.3GHz PC with 16GB main memory. We use the Z-index sort method [IABT11] to accelerate the neighborhood query and each neighborhood contains approximately 30 particles. Since we use an implicit integrator to solve wave propagation, wave simulation is numerically stable even when using a large time step. However, a large time step can cause large errors in surface tension energy estimation. In practice, we set the time step to Δt =0.0016s for reasonable effects. Besides, we seed ghost particles near the solid boundary to impose free-slip boundary condition for the pressure Poisson equation [HLW*12]. For the wave propagation equation, we ensure the wave density gradient is zero at the solid boundary to reflect the incoming waves. Table 1 shows the timings where we apply the conjugate gradient method to solve both the wave propagation equation and the pressure Poisson equation.

Examples. Our experiment evaluates the capillary waves generated by our system in a wide range of small-scale water examples, including water crown formed by a water drop (in Figure 1), gravity-free water balls (in Figure 4), surface waves caused by pouring water (in Figure 9), a bubble hit by a bullet (in Figure 8), and thin water sheets formed by colliding water streams (in Figure 10). These examples demonstrate that the generated wave effects are visually appealing and similar to the capillary waves in the real world.

Controllability. The wave simulation step in our algorithm can be done either simultaneously with SPH simulation, or offline as post-processing. The system offers three options for users to adjust capillary wave effects: the wave speed c_0 (in Equation 1), the wave energy dissipation rate $\nu_{\min}$ (in Equation 3), and the surface tension energy coefficient κ controlling the wave amplitude (in Equation 10). Although they are currently defined as constants, they can be adjusted for realistic spatio-temporally varying wave effects. Besides, the wave sources can also be added or deleted manually for regions of interest. For example, we have prevented

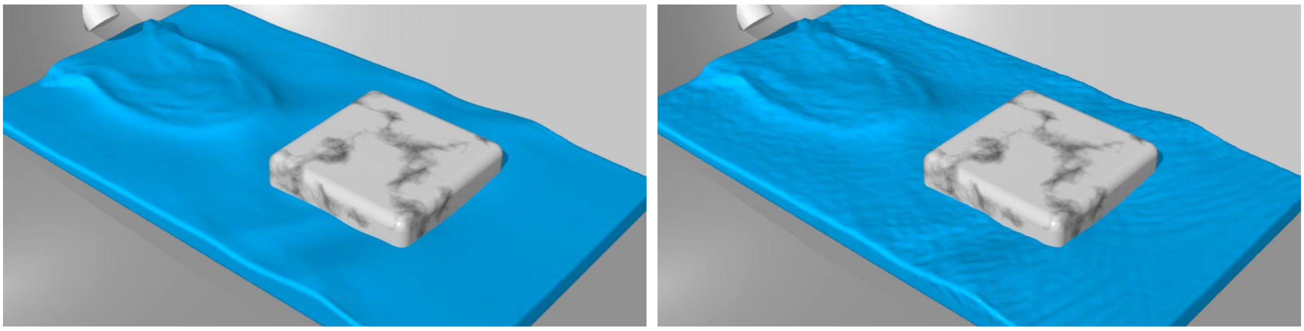

(a) Without enrichment (b) With enrichment

Figure 9: *Surface waves formed by a stream of water. Small ripples are created by water splashes and drops in this example.*

	N	$t_{neighbor}$	t_{pos}	t_{water}	t_{wave}	t_{total}
Fig. 1	807K	0.67	0.35	0.72	0.75	2.74
Fig. 8(b)	25K	0.05	0.02	0.06	0.06	0.18
Fig. 9	1809K	1.41	0.87	1.64	1.73	5.94
Fig. 10	393K	0.39	0.18	0.40	0.45	1.43

Table 1: *Timings in seconds. This table lists the number of particles N, the neighbor querying cost $t_{neighbor}$, the density correction cost t_{pos}, the incompressibility solver cost t_{water}, the wave solver cost t_{wave}, and the total cost t_{total} per time step.*

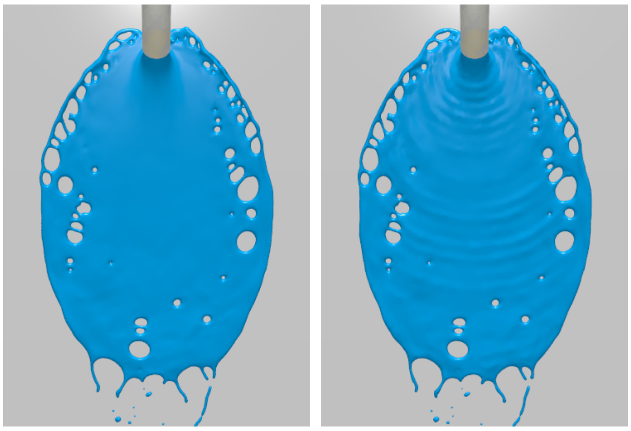

(a) Without enrichment (b) With enrichment

Figure 10: *A thin sheet. Our method generates capillary waves on a thin sheet, formed by two water streams colliding into each other.*

the capillary waves from being created near the boundary of container in Figure 1 and 9 by explicitly setting $\Delta e_{s,i} = 0$ to the fluid particles near the container.

Comparison to [MBT*15]. Our capillary-wave-enriching method differs from the surface-turbulence-enriching method in two aspects. First, the surface-turbulence-enriching method requires to construct a dense point set to represent the liquid's surface and explicitly maintain its smoothness and regularity, while our method directly simulates the waves on original fluid particle and does not require extra data structures to deal with the topology changes in fluid. Therefore, the implementation of our method is much easier. Second, our method aims to capture the capillary waves for particle methods, so we seed the waves according to the surface energy instead of curvature, which is proved to be more robust and accurate for SPH solvers to produce surface tension effects [HWZ*14](Please see the video for a comparison).

Limitations. The wave simulation step in our algorithm is fundamentally based on compression waves rather than capillary waves, so it is not meant to be physically accurate. The simulated compression wave is limited by the basic particle resolution because no additional structures are introduced. Our algorithm does not control the wavelength and it requires user to specify the wave propagation speed as a constant. In reality, the propagation speed of gravity-capillary waves can vary, depending on the wavelength and the orientation of the liquid surface. Finally, the quality of our result is somewhat built upon the quality of surface reconstruction.

6. Conclusions and Future Work

We present a new approach to efficiently and realistically enrich SPH simulation with capillary wave effects. We formulate the whole approach based on the assumption that capillary waves can be approximated by surface compression waves, and our experiment reveals it is a reasonable assumption. Our research further shows the surface tension energy change measured under the SPH framework can provide a good clue on the source of capillary waves, although the waves cannot be maintained by SPH simulation.

Looking into the future, we would like to perform a theoretical study to improve the accuracy of our capillary wave propagation model. We also would like to investigate solid boundary conditions for more accurate wave reflection. We will improve the performance of our algorithm by GPU, and study the compatibility of our wave simulation method with position-based dynamics. Finally, we are interested in exploring the use of our approximation idea under other liquid simulation frameworks.

Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive comments and Mingming He for modeling and rendering. This research was supported by the National Key Technology R&D Program of China (No. 2015BAK01B06) and the National Natural Science Foundation of China (No. 61421062, 61232014, 61472010, 61170205, 61272326, 6140051239). Enhua Wu was also supported by Macao SAR FDCT (068/2015/A2).

References

[AAT13] Akinci N., Akinci G., Teschner M.: Versatile surface tension and adhesion for SPH fluids. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 182:1–182:8. 2

[AIA*12] Akinci N., Ihmsen M., Akinci G., Solenthaler B., Teschner M.: Versatile rigid-fluid coupling for incompressible SPH. *ACM Trans. Graph. (SIGGRAPH) 31*, 4 (July 2012), 62:1–62:8. 2

[AIAT12] Akinci G., Ihmsen M., Akinci N., Teschner M.: Parallel surface reconstruction for particle-based fluids. *Comp. Graph. Forum 31*, 6 (2012), 1797–1809. 6

[ATBG08] Angst R., Thuerey N., Botsch M., Gross M.: Robust and efficient wave simulations on deforming meshes. *Comp. Graph. Forum 27*, 7 (2008), 1895–1900. 1

[BGB11] Bhatacharya H., Gao Y., Bargteil A.: A level-set method for skinning animated particle data. In *Proc. of SCA* (2011), pp. 17–24. 6

[BT07] Becker M., Teschner M.: Weakly compressible SPH for free surface flows. In *Proc. of SCA* (2007), pp. 209–217. 2

[CBP05] Clavet S., Beaudoin P., Poulin P.: Particle-based viscoelastic fluid simulation. In *Proc. of SCA* (July 2005), pp. 219–228. 2

[CH58] Cahn J., Hilliard J.: Free energy of a nonuniform system. I. interfacial free energy. *The Journal of Chemical Physics 28* (1958), 258. 4

[CR99] Cummins S. J., Rudman M.: An SPH projection method. *Journal of computational physics 152*, 2 (1999), 584–607. 2, 5

[HA06] Hu X., Adams N.: A multi-phase SPH method for macroscopic and mesoscopic flows. *Journal of Computational Physics 213*, 2 (2006), 844–861. 2

[HLL*12] He X., Liu N., Li S., Wang H., Wang G.: Local Poisson SPH for viscous incompressible fluids. *Comp. Graph. Forum (Pacific Graphics) 31*, 6 (2012), 1948–1958. 2

[HLW*12] He X., Liu N., Wang G., Zhang F., Li S., Shao S., Wang H.: Staggered meshless solid-fluid coupling. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 149:1–149:12. 2, 6

[HWZ*14] He X., Wang H., Zhang F., Wang H., Wang G., Zhou K.: Robust simulation of sparsely sampled thin features in SPH-based free surface flows. *ACM Trans. Graph. 34*, 1 (Dec. 2014), 7:1–7:9. 2, 5, 7

[IABT11] Ihmsen M., Akinci N., Becker M., Teschner M.: A parallel SPH implementation on multi-core CPUs. *Comp. Graph. Forum 30*, 1 (2011), 99–112. 6

[ICS*14] Ihmsen M., Cornelis J., Solenthaler B., Horvath C., Teschner M.: Implicit incompressible SPH. *IEEE Trans. Vis. Comp. Graph 20*, 3 (March 2014), 426–435. 2, 4

[IOS*14] Ihmsen M., Orthmann J., Solenthaler B., Kolb A., Teschner M.: SPH fluids in computer graphics. In *Eurographics 2014-State of the Art Reports* (2014), The Eurographics Association, pp. 21–42. 2

[JKB*10] Jang T., Kim H., Bae J., Seo J., Noh J.: Multilevel vorticity confinement for water turbulence simulation. *Vis. Comput. 26*, 6-8 (June 2010), 873–881. 1

[JTB02] Jamet D., Torres D., Brackbill J.: On the theory and computation of surface tension: The elimination of parasitic currents through energy conservation in the second-gradient method. *Journal of Computational Physics 182*, 1 (2002), 262–276. 4

[JW15] Jeschke S., Wojtan C.: Water wave animation via wavefront parameter interpolation. *ACM Trans. Graph. (SIGGRAPH) 34*, 3 (2015), 27. 1

[KTT13] Kim T., Tessendorf J., Thürey N.: Closest point turbulence for liquid surfaces. *ACM Trans. Graph. 32*, 2 (Apr. 2013), 15:1–15:13. 1

[LTKF08] Losasso F., Talton J., Kwatra N., Fedkiw R.: Two-way coupled SPH and particle level set fluid simulation. *IEEE Trans. Vis. Comp. Graph. 14*, 4 (2008), 797–804. 6

[MBT*15] Mercier O., Beauchemin C., Thuerey N., Kim T., Nowrouzezahrai D.: Surface turbulence for particle-based liquid simulations. *ACM Trans. Graph. (SIGGRAPH Asia) 34*, 6 (Nov. 2015). 7

[MCG03] Müller M., Charypar D., Gross M.: Particle-based fluid simulation for interactive applications. In *Proc. of SCA* (2003), pp. 154–159. 2, 5

[MFZ97] Morris J. P., Fox P. J., Zhu Y.: Modeling low Reynolds number incompressible flows using SPH. *Journal of computational physics 136*, 1 (1997), 214–226. 2, 4

[MM13] Macklin M., Müller M.: Position based fluids. *ACM Trans. Graph. 32*, 4 (July 2013), 104:1–104:12. 2, 4, 6

[Mon94] Monaghan J. J.: Simulating free surface flows with SPH. *Journal of Computational Physics 110*, 2 (Feb. 1994), 399–406. 2

[Mor00] Morris J.: Simulating surface tension with smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids 33*, 3 (2000), 333–353. 2

[MRA*13] Mehra R., Raghuvanshi N., Antani L., Chandak A., Curtis S., Manocha D.: Wave-based sound propagation in large open scenes using an equivalent source formulation. *ACM Trans. Graph. 32*, 2 (Apr. 2013), 19:1–19:13. 2

[MSKG05] Müller M., Solenthaler B., Keiser R., Gross M.: Particle-based fluid-fluid interaction. In *Proc. of SCA* (2005), pp. 237–244. 2

[NSCL08] Narain R., Sewall J., Carlson M., Lin M. C.: Fast animation of turbulence using energy transport and procedural synthesis. *ACM Trans. Graph. (SIGGRAPH Asia) 27*, 5 (Dec. 2008), 166:1–166:8. 1

[NT14] Nair P., Tomar G.: An improved free surface modeling for incompressible sph. *Computers & Fluids* (2014), 304ÍC314. 5

[RSM*10] Raghuvanshi N., Snyder J., Mehra R., Lin M., Govindaraju N.: Precomputed wave simulation for real-time sound propagation of dynamic sources in complex scenes. *ACM Trans. Graph. (SIGGRAPH) 29*, 4 (July 2010), 68:1–68:11. 2

[RWT11] Raveendran K., Wojtan C., Turk G.: Hybrid smoothed particle hydrodynamics. In *In Proc. of SCA* (2011), ACM, pp. 33–42. 6

[SB12] Schechter H., Bridson R.: Ghost SPH for animating water. *ACM Trans. Graph. (SIGGRAPH) 31*, 4 (July 2012), 61:1–61:8. 2

[SBC*11] Solenthaler B., Bucher P., Chentanez N., Müller M., Gross M.: SPH based shallow water simulation. In *Virtual Reality Interactions and Physical Simulations (VRIPHYS)* (2011), pp. 39–46. 2

[SL03] Shao S., Lo E. Y.: Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Advances in Water Resources 26*, 7 (2003), 787–800. 2, 4

[SP08] Solenthaler B., Pajarola R.: Density contrast SPH interfaces. In *Proc. of SCA* (2008), pp. 211–218. 2

[SP09] Solenthaler B., Pajarola R.: Predictive-corrective incompressible SPH. *ACM Trans. Graph. (SIGGRAPH) 28*, 3 (July 2009), 40:1–40:6. 2

[TM05] Tartakovsky A., Meakin P.: Modeling of surface tension and contact angles with smoothed particle hydrodynamics. *Physical Review E 72*, 2 (2005), 026301. 2

[TWGT10] Thürey N., Wojtan C., Gross M., Turk G.: A multiscale approach to mesh-based surface tension flows. *ACM Trans. Graph. (SIGGRAPH)* (2010), 48:1–48:10. 1, 4

[WMT07] Wang H., Miller G., Turk G.: Solving general shallow wave equations on surfaces. In *Proc. of SCA* (2007), pp. 229–238. 1

[YT13] Yu J., Turk G.: Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Trans. Graph. 32*, 1 (2013), 5. 6

[YWTY12] Yu J., Wojtan C., Turk G., Yap C.: Explicit mesh surfaces for particle based fluids. *Comp. Graph. Forum (Eurographics) 31*, 4 (May 2012), 815–824. 1