

Stretching and Wiggling Liquids

Doyub Kim
Seoul National University

Oh-young Song
Sejong University

Hyeong-Seok Ko
Seoul National University

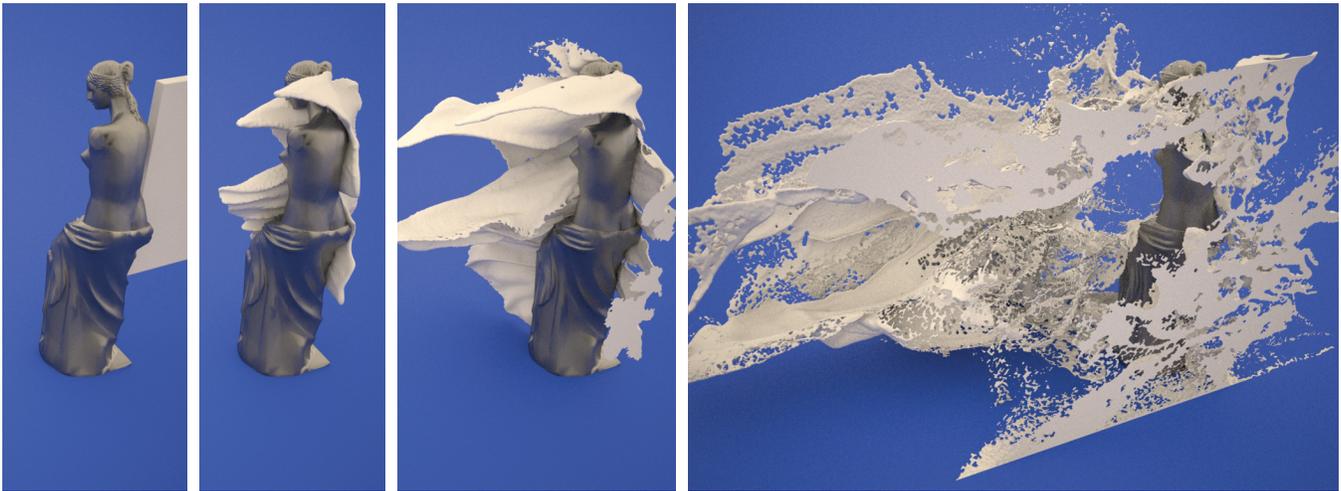


Figure 1: Liquid injected toward a statue of Venus generates a complex splash. Simulation was carried out with a $192 \times 96 \times 64$ regular grid, and surface tracking was performed with a $768 \times 384 \times 256$ octree grid.

Abstract

This paper presents a novel framework for simulating the stretching and wiggling of liquids. We demonstrate that complex phase-interface dynamics can be effectively simulated by introducing the Eulerian vortex sheet method, which focuses on the vorticity at the interface (rather than the whole domain). We extend this model to provide user control for the production of visual effects. Then, the generated fluid flow creates complex surface details, such as thin and wiggling fluid sheets. To capture such high-frequency features efficiently, this work employs a denser grid for surface tracking in addition to the (coarser) simulation grid. In this context, the paper proposes a filter, called the liquid-biased filter, which is able to downsample the surface in the high-resolution grid into the coarse grid without unrealistic volume loss resulting from aliasing error. The proposed method, which runs on a single PC, realistically reproduces complex fluid scenes.

CR Categories: I.3.7 [Computing Methodologies]: Computer Graphics—Three-Dimensional Graphics and Realism;

Keywords: fluid animation, level set method, Eulerian vortex sheet method, surface tracking

1 Introduction

We are frequently exposed to snapshots on billboards and slow-motion pictures in TV commercials showing volumes of liquid that stretch into a sheet, wiggle, and then break into droplets. This beautiful, complex phenomenon results from liquid–air interfacial effects such as force instabilities, and other inter-medium interactions. This paper is concerned with reproducing such phenomena in the context of visual effects. To this end, the paper adopts physically-based approaches, but aims to develop a controllable, viable method.

Physics-based simulation of three-dimensional liquids essentially samples physical quantities using discrete grids [Foster and Fedkiw 2001; Enright et al. 2002], particles [Müller et al. 2003; Premoze et al. 2003], or a hybrid of both [Song et al. 2007; Losasso et al. 2008]. While the above solvers produce plausible results when they are applied to large bodies of liquid (compared to the grid size and/or inter-particle distance), they often produce overly damped or dissipated results when they are used to generate wiggling/tearing of liquid sheets. We believe that this limitation can be alleviated by tailoring the solvers such that liquid–air interfacial effects are accounted for through dynamic simulation and capturing of the resulting surface to a satisfactory extent.

In this paper, we propose a novel method for simulating liquid–air interfacial effects. Our method consists of two major components: (1) a controllable Eulerian vortex sheet model, and (2) a liquid-biased filter. The Eulerian vortex sheet model of Herrmann [2003; 2005] focuses on the interface (rather than the whole volume) in the simulation of complex interface dynamics. We extend this model so that the vortex around the liquid surface can be controlled as desired. With the above method, we are able to generate detailed fluid motion around the liquid surface without expensive computational cost. Also, when the wiggling and stretching feature of the liquid surface appears which contains high-frequency modes, denser mesh should be applied according to the Nyquist limit. However,

denser meshing implies higher cost, especially for solving linear system during the computation of incompressible flow. This work uses denser grids only for the purpose of capturing interfacial surfaces, in addition to the (coarser) simulation grids. In this setup, we develop a new filter that shows superior capturing performance for thin liquid volumes, so that unrealistic aliasing error can be removed.

Figure 1 demonstrates that the proposed method can realistically reproduce violent, complicated break-up events. We note that, on a single PC, the simulation took 8 to 120 seconds to advance a single frame; advection was performed using linear semi-Lagrangian interpolation in all of the experiments in this paper.

2 Related Work

The stable fluids framework introduced by Stam [1999] is an important innovation in the field of fluid animation. The implicit pressure projection and semi-Lagrangian advection used in this framework enable us to take large time-steps without blowing up the simulation. Foster and Fedkiw [2001] demonstrated that the stable fluids framework, when combined with the level set method, can be used for liquid animation. In other work, some researchers resorted to multi-phase dynamics for liquid animation [Song et al. 2005; Hong and Kim 2005; Losasso et al. 2006].

Realism has been a constant issue in fluid simulation. Noting that vorticity is an important element in realistic fluid movements, Fedkiw et al. [2001] introduced the vorticity confinement method, which detects and explicitly models the curly components in fluid flows. Selle et al. [2005] reproduced turbulent smoke/liquid movements by introducing the vortex particle method. A procedural approach (called curl noise) to generate vorticity was proposed by Bridson et al. [2007]. Recently, several turbulence-based methods have been presented to resolve sub-cell vorticities [Kim et al. 2008b; Schechter and Bridson 2008; Narain et al. 2008]. Accurate advection is critical for realistic fluid simulation. Several high-order advection schemes also have been proposed, including the back and forth error compensation and correction (BFECC) method [Kim et al. 2007], MacCormack scheme [Selle et al. 2008], and constrained interpolation profile (CIP) methods [Song et al. 2005; Kim et al. 2008a].

In the simulation of a liquid, surface tracking performance is another major factor in obtaining realistic results. After the introduction of the level set method by Foster and Fedkiw [2001], Enright et al. [2002] proposed the particle level set method to improve the surface tracking accuracy. The particle level set method was extended to derivative particles [Song et al. 2007] and the marker level set [Mihalef et al. 2007]. Surface tracking quality can also be enhanced by using grids of higher resolution; this idea formed the basis for various adaptive data structures, including the octree [Losasso et al. 2004], semi-Lagrangian contouring [Bargteil et al. 2006], lattice-based tetrahedral meshes [Chentanez et al. 2007], and the hierarchical RLE grid [Houston et al. 2006].

In addition to the above grid-based methods, particle-based fluid solvers have also been actively studied. Müller et al. [2003] introduced smoothed particles hydrodynamics (SPH), and Premoze et al. [2003] introduced the moving particles semi-implicit method (MPS) to the graphics community. Zhu and Bridson [2005] introduced the fluid implicit particle (FLIP) method to reduce numerical dissipation of a grid-based advection solver. Recently, Adams et al. [2007] presented the adaptive SPH method, and Solenthaler and Pajarola [2009] introduced Predictive-Corrective Incompressible SPH scheme to increase the stability of the particle solver. To exploit the advantages of both grid-based and particle-based methods, Losasso et al. [2008] and Hong et al. [2008] coupled the level

set and SPH frameworks.

3 Basic Fluid Solver

The method proposed in this paper is based on the stable fluids solver and the level set method. This section briefly describes this basic solver and lists the equations that form the groundwork for deriving the proposed method.

3.1 Level Set Method

The level set method is a popular technique for tracking liquid surfaces. Level set ϕ is a signed-distance function such that $|\nabla\phi| = 1$ for all domains and the interface is defined at $\phi = 0$. Due to this property, we can easily obtain the surface normal $\mathbf{n} = \nabla\phi/|\nabla\phi|$, and mean curvature $\kappa = \nabla \cdot \mathbf{n}$. In the level set framework, a surface integral can be transformed to a volume integral with

$$\int_{\Gamma} f(s) ds = \int_V f(\mathbf{x}') \delta(\phi(\mathbf{x}')) |\nabla\phi(\mathbf{x}')| d\mathbf{x}', \quad (1)$$

where δ is the smeared delta function

$$\delta(x) = \begin{cases} \frac{1}{2e} + \frac{1}{2e} \cos\left(\frac{\pi x}{e}\right) & \text{if } |x| \leq e \\ 0 & \text{if } |x| > e. \end{cases} \quad (2)$$

We can generate a certain quantity ψ extrapolated along the direction normal to the liquid surface by imposing the condition [Foster and Fedkiw 2001]

$$\nabla\psi \cdot \nabla\phi = 0, \quad (3)$$

which is one of the Eikonal equations and can be solved efficiently using the fast marching method (FMM) [Sethian 1999].

Now we explain how to evolve the interfacial surface under the flow field. When the velocity field is given by \mathbf{u} , the level set field can be updated by solving

$$\phi_t + \mathbf{u} \cdot \nabla\phi = 0. \quad (4)$$

The above level set advection equation can be solved using the semi-Lagrangian method [Stam 1999; Foster and Fedkiw 2001]. Since numerical diffusion can cause a significant loss of mass, in this work we employed the particle level set method (PLS) of Enright et al. [2002]. After evolving the surface, we perform FMM once again to ensure the signed-distance property.

3.2 Incompressible Flow

In this paper, we assume inviscid incompressible free-surface flow when modeling the liquid dynamics. Incompressible Euler equations are given by the momentum conservation equation

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{f} \quad (5)$$

and the mass conservation equation

$$\nabla \cdot \mathbf{u} = 0, \quad (6)$$

where \mathbf{u} and p represent the velocity and pressure, respectively. The term \mathbf{f} represents external forces, such as gravity or the vorticity confinement force [Fedkiw et al. 2001]. We solve the above equations using the fractional step method of Stam [1999], in which a Poisson equation is solved under divergence-free conditions to obtain the pressure, which is then used to project the intermediate velocity field into a divergence-free field.

For simulating the liquid, we use a free-surface model in which atmospheric pressure is assumed to be constant ($p|_{\phi>0} = p_{atm}$). This condition is reflected in the simulator as a Dirichlet boundary condition at the liquid–air interface. Under this condition, we solve the free-surface flow using the ghost fluid method [Gibou et al. 2002]. Since the free-surface model does not simulate the fluid flow of air, the velocity at the interface should be extrapolated to the air region using equation (3) [Foster and Fedkiw 2001].

In the following two sections, we describe the two main methods proposed in this paper. First, we introduce the Eulerian vortex sheet method and extend it to make the technique controllable. Second, we present a liquid-biased filter in the context of super-sampled surface tracking that is targeted to not missing thin liquid structures.

4 Controllable Eulerian Vortex Sheet Model

As a liquid evolves, motion of the surface is affected by the surrounding air. When a liquid stretches into a thin sheet, this interaction force becomes more dominant and often causes the liquid to wiggle or break into droplets. One approach that could potentially be used to simulate such situations is the multi-phase method, which can reflect the presence of air by assigning jump conditions to the density and viscosity at the air–liquid interface [Hong and Kim 2005]. Except for the surface tension, however, the multi-phase method does not handle the phase interface explicitly. Although the phase-interaction is highly concentrated at the interface, the multi-phase method treats the entire domain uniformly. Only density and viscosity are the related variables, but they are not directly involved in the deformation of the surface. In this context, it is worth considering the Eulerian vortex sheet method, which explicitly focuses on the liquid surface rather than the whole domain.

The vortex sheet method assumes that vorticity $\omega = \nabla \times \mathbf{u}$ is concentrated at the liquid–air interface. The phase interface itself constitutes a *vortex sheet* with varying vortex sheet strength. For the three-dimensional case, vortex sheet strength η can be approximated at the interface Γ by [Herrmann 2003]

$$\eta = \mathbf{n} \times (\mathbf{u}^+ - \mathbf{u}^-)|_{\Gamma}, \quad (7)$$

where \mathbf{u}^{\pm} is the velocity at ϕ^{\pm} . Note that η is a three-dimensional vector quantity. The above equation implies that the vortex sheet strength is a jump condition for the tangential component of the velocity across the interface. Combining the above equation with the Euler equations, the transport of the vortex sheet can be expressed as

$$\eta_t + \mathbf{u} \cdot \nabla \eta = -\mathbf{n} \times \{(\eta \times \mathbf{n}) \cdot \nabla \mathbf{u}\} + \mathbf{n} \{(\nabla \mathbf{u} \cdot \mathbf{n}) \cdot \eta\} + \mathbf{S}. \quad (8)$$

The derivation and a detailed explanation of this equation can be found in the works of Tryggvason [1988] and Pozrikidis [2000]. Here, the left-hand side of the equation is related to the advection of the vortex sheet strength η , and the terms in the right-hand side represent the effects of stretching, dilatation, and extra sources. This equation is valid only at the interface since η is defined on the surface. However, for numerical simulation, in the vicinity of the surface, η is extrapolated along the surface normal direction using equation (3).

In the original vortex sheet equation of Pozrikidis [2000], \mathbf{S} consists of the surface tension and baroclinity terms. Since we compute surface tension more accurately [Enright et al. 2003], we ignore the surface tension for now. Pozrikidis [2000] uses the baroclinity term $2A\mathbf{n} \times (\mathbf{a} - \mathbf{g})$, where $A = (\rho^- - \rho^+)/(\rho^- + \rho^+)$ is the Atwood number, \mathbf{a} is the average convective acceleration at the interface, and $\mathbf{g} = -9.8\hat{\mathbf{j}}$ is the acceleration due to gravity. The baroclinity

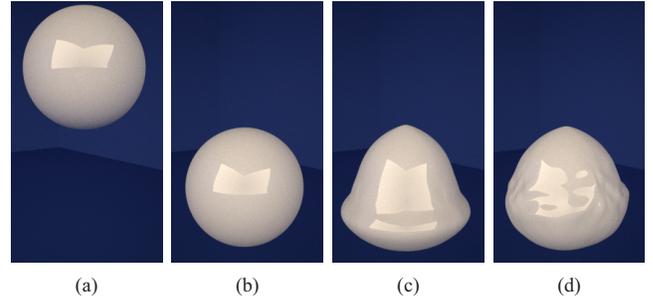


Figure 2: Liquid drop simulation demonstrating the effect of the vortex sheet method. (a) the initial setup. (b) the result of the free-surface simulation without the vortex sheet method. (c) the baroclinity effect generated by the vortex sheet method. (d) the result of the vortex sheet simulation with ambient noise.

term accounts for interfacial effects such as the Rayleigh–Taylor instability [Tryggvason 1988], which arise due to the density difference between liquid and air. In adopting the vortex sheet method, we use the source term

$$\mathbf{S} = bA\mathbf{n} \times (\mathbf{a} - \mathbf{g}), \quad (9)$$

where b is a parameter used to control the magnitude of this effect. We set b to a value between 1 and 2 in this work.

Solving equation (8) gives the vortex sheet strength η over the liquid surface. We now show how to calculate the velocity with this information. First, we can calculate vorticity ω from η with

$$\omega(\mathbf{x}) = \int_{\Gamma} \eta(s) \delta(\mathbf{x} - \mathbf{x}(s)) ds. \quad (10)$$

The above equation can be rewritten as a volume integral (see equation (1))

$$\omega(\mathbf{x}) = \int_V \eta(\mathbf{x}') \delta(\mathbf{x} - \mathbf{x}') \delta(\phi(\mathbf{x}')) |\nabla \phi(\mathbf{x}')| d\mathbf{x}'. \quad (11)$$

The conventional approach to computing the velocity from the vorticity ω is to use the stream function $\nabla^2 \psi = \omega$, and then compute $\mathbf{u} = \nabla \times \psi$. For controllability, however, the present work employs a local correction method that is similar to the method of Selle et al. [2005].

With the vorticity computed from equation (11), we apply vorticity confinement to the velocity field. Letting $\mathbf{N} = \nabla|\omega|/|\nabla|\omega||$, the vorticity confinement force can be written as

$$\mathbf{f}_v(\mathbf{x}) = \alpha h(\mathbf{N} \times \omega), \quad (12)$$

where h is the size of the grid cell, and α is the control parameter for the vorticity confinement force. Simulation of a single time step is completed by augmenting the external force term of equation (5) with the confinement force. We use the semi-Lagrangian scheme for the advection term in equation (8). All other terms are discretized with second-order central difference, and first-order Euler integration is used for time-marching. In implementing the vorticity confinement, we blend the interface-concentrated vorticity ω of equation (11) using a truncated Gaussian kernel [Selle et al. 2005]. We used a kernel width of $10h$.

A nice feature of the vortex sheet formulation, which we take advantage of in the context of visual effects production, is that the interface dynamics are explicitly expressed as a source term. Inspired by Herrmann [2003] and Bridson et al. [2007], we exploit

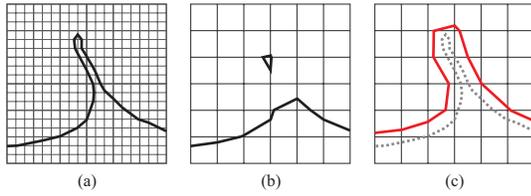


Figure 3: Aliasing in the simulation grid. (a) the surface sampled with a dense grid. (b) the same surface downsampled with a coarse grid. (c) the surface downsampled with the liquid-biased filter (red line) overlaid with the surface in (a) (dotted line).

this feature to generate turbulence effects. To do so, we replace η in equation (11) with $\hat{\eta} = \eta(1 + \mathbf{T}\Phi)$. Here, \mathbf{T} is the projection operator and Φ is the ambient vector noise, which is generated using the Perlin noise algorithm [Perlin 1985]. Operator \mathbf{T} projects this noise to the tangential plane of the surface, since vortex sheet strength should be orthogonal to the surface normal.

Figure 2 demonstrates the effects of the vortex sheet method through the example of dropping a single large water ball. When the dropping ball is simulated using the free-surface model without the vortex sheet method, the ball is simply translated in the direction of gravity (Figure 2 (b)). When the baroclinity effect is added, the interface of the ball winds up due to the density difference between the two different media. Figure 2 (c) shows the result when $b = 1$, and Figure 2 (d) shows the result when ambient noise is added to (c). The introduction of ambient noise adds visual complexity to the liquid surface.

In this section we adopted the vortex sheet model for effective handling of the interfacial dynamics. In the process, several extensions were made for visual effects production. The extended vortex sheet model now has three control variables, namely b , α , and Φ , which can be used to control the extent of liquid–air interaction, the strength of the vortices, and the ambient noise, respectively. In the following section, we describe how to capture the rich details of liquid surface generated by the vortex sheet model efficiently.

5 Surface Tracking with Super-sampling

Performing surface tracking on a higher-resolution grid can increase visual realism of the simulation without introducing too much extra cost. Goktekin et al. [2004] found that using a higher resolution grid for surface tracking can reduce volume loss significantly. Bargteil et al. [2006] pioneered the use of higher resolution octree grids for surface tracking. Recently, Wojtan and Turk [2008] embedded a high-resolution grid into a coarser grid in finite element method (FEM) simulations, and used the resulting formalism to simulate thin structures of a viscoelastic fluid.

As noted by Lossaso et al. [2004] and Bargteil et al. [2006], however, using a denser grid for surface tracking can lead to artifacts. For example, suppose that we have a thin water structure tracked via a dense grid (Figure 3 (a)). The problem occurs when the simulator tries to solve the projection step for maintaining incompressibility. As shown in Figure 3 (b), the simulation grid cannot resolve such high-frequency signals. The liquid regions that are not captured will vanish because they are no longer incompressible. To resolve the above problems, we present the liquid-biased filter, which can effectively conserve incompressibility of small liquid fractions.

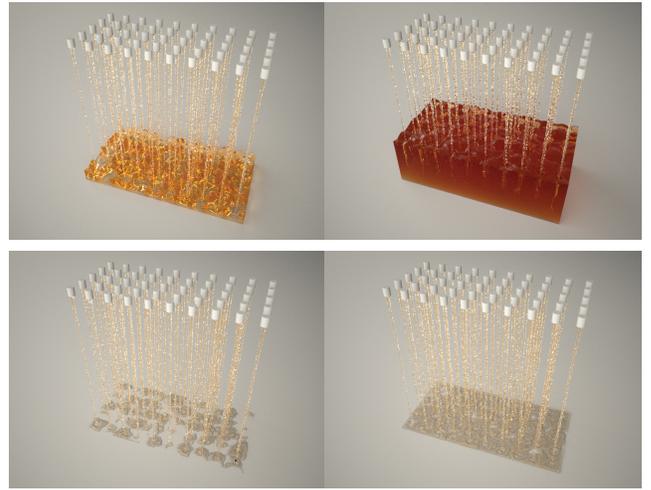


Figure 4: Shower simulations generated with the liquid-biased filter (top row), and without the liquid-biased filter (bottom row). Both simulations were performed with a $128 \times 128 \times 64$ grid, and a $512 \times 512 \times 256$ octree grid was used for surface tracking.

5.1 Liquid-Biased Filtering

When sampling needs to be done, a filter can be designed to reduce the aliasing error. A sophisticated filtering method is required when downsampling the surface from a dense grid to a coarse grid. Several filters have been proposed for reconstructing implicit surfaces [Marschner and Lobb 1994]. The immersed boundary approach [Song et al. 2005], which uses a smeared delta function as a filter, is also an anti-aliasing method. Although the reconstructed distance value may be accurate in the above methods, however, the sign may be flipped. Therefore, even when using sophisticated downsampling filters, some parts of the liquid volume may still be interpreted as air regions.

In this context, we design a new filter, which we call the *liquid-biased filter* since it is targeted to not missing liquid regions in the process of downsampling. Our filter makes the liquid regions of the coarse grid enclose all the liquid regions of the dense grid by simply shifting the level set threshold value. Let us denote the level sets defined on the coarse and dense grids as ϕ_C and ϕ_D , respectively. The liquid surface in the dense grid is defined as the zero-level set $\phi_D = 0$. From the definition of the signed-distance function, the region $\phi_D < \epsilon$ (where ϵ is a small positive number) includes the region $\phi_D < 0$. We define ϵ to be $\epsilon' \cdot h_C$, where h_C is the grid size of the coarse grid. In transferring the liquid surface in the dense grid to the coarse grid, here we take the incompressible region according to $\phi_C < \epsilon$ (the regions enclosed by the red line in Figure 3 (c)), instead of $\phi_C < 0$ (the regions enclosed by the dotted line in Figure 3 (c)). We set $\epsilon' = 0.5$ to capture even the thinnest structures in the dense grid.

To solve free-surface incompressible flow, we apply the ghost fluid method [Gibou et al. 2002] with a Dirichlet boundary condition on the surface given by the atmospheric pressure $p = p_{atm}$. Since we define the incompressible region as $\phi_C < \epsilon$, the pressure p at $\phi_C = \epsilon$ should be p_{atm} . Let us consider solving the one dimensional pressure Poisson equation, where the interface $\phi_C = \epsilon$ is located between grid points i and $i + 1$ (of the coarse grid), and the above Dirichlet boundary condition is applied at that interface. The

discretized Poisson equation can be written as,

$$\frac{p_{i+1}^G - 2p_i + p_{i-1}}{h_C^2} = \frac{u_{i+1/2} - u_{i-1/2}}{h_C}, \quad (13)$$

where p_{i+1}^G is the ghost pressure value extrapolated from the internal side of the liquid. To obtain p_{i+1}^G , linear extrapolation can be performed with

$$p_{i+1}^G = \frac{p_{atm} + (\theta - 1)p_i}{\theta}, \quad (14)$$

where θ is the normalized distance from grid point i to interface $\phi_C = \epsilon$. Substituting equation (14) into (13) produces a linear system that can be solved using the preconditioned conjugate gradient method. The computed pressure contains an error, since we shifted the boundary about 0.5ϵ . However, this liquid-biased filtering successfully preserves sub-cell scale liquid fractions.

The effect of the liquid-biased filter against aliasing is shown in Figure 4. In this simulation, liquid sprays are emitted by nozzles and drop into a rectangular container. When the liquid-biased filter is used, the water level rises as expected. Without the liquid-biased filter, however, the water level does not rise because the thin water layer at the bottom of the container is not captured via the coarse simulation grid. During the liquid-biased filtering, however, we do not accumulate the down-sampled volume over time. Thus, we don't apply an opposite bias, and there is no accumulation of volume. In the same context, we would like to make it clear that the liquid-biased filtering is different from the *thickening method* by Chentanez et al. [2007], which adds more thickness where the volume loss occurs.

Due to its Eulerian representation of the surface, the level set method can automatically handle topological changes. Numerical diffusion has a catalytic role in generating the topological changes. When the grid resolution increases, the amount of diffusion decreases, which can lead to prevention of topological changes. In particular, when liquid-biased filtering is used on a high-resolution grid, two adjoining liquid volumes may remain disjoint rather than combining.

To properly handle topological changes, artificial diffusion can be applied to force the liquid to merge. The amount of diffusion should be as small as possible, since it might blur out desirable surface details. Also, the diffusion should be biased to the merging of liquid, since merging of air means deletion of the liquid sheet. To meet these requirements, we simply reinitialize the level set from $\phi_D = d$ (instead of $\phi_D = 0$), where d is a small positive number. Due to the property of the level set, merging takes place if the distance between the liquid volumes is less than $2d$. We found that setting $d = 0.1h_C$ gives plausible results. However, parameter d is actually not that sensitive. Value nearby $0.1h_c$ will suffice.

6 Results

All experiments reported in this paper were performed on an Intel Core2 Quad Q6600 2.4 GHz processor with 4 GB of memory without parallel execution. We used uniform regular staggered grid for the coarse mesh, and an adaptive octree grid for the dense mesh. Every physical quantity except the level set was stored in the coarse grid. In all simulations, the advection step was performed using the first-order semi-Lagrangian method. The particle level set method [Enright et al. 2002] was used to correct the numerical dissipation error during the level set advection. We used $1 \sim 2$ for b , $10 \sim 50$ for α , and $0 \sim 1$ for the magnitude of the noise. In addition, we used $1.5 \times h_C$ for e in equation 2, and $799/801$ for A in equation 9. We restricted CFL number to be under 3 for stability.



Figure 5: Liquid sprays injected toward a dragon model. The simulation was performed using a $160 \times 80 \times 80$ regular grid, and surface tracking was performed using a $640 \times 320 \times 320$ octree grid.

Figure 1 shows the result obtained when the proposed simulator was applied to the reproduction of a liquid interacting with a static solid obstacle. The simulation took about 8 to 120 seconds to advance a single animation frame, depending on the scene complexity. Most of the computational time was spent performing the octree refinement. The simulator successfully reproduced the fine details of liquid sheets and droplets that form as the liquid volume hits the statue.

In Figure 5, a large number of massless particles are emitted from the nozzles, and are soon converted into a grid-based surface by the particle level set method. Due to our vortex sheet model and high-resolution surface tracking grid, complex details of the liquid can be simulated and visualized, even with a relatively coarse simulation grid. About 10 to 30 seconds were required to simulate a single animation frame.

To compare the simulation quality afforded by the proposed method with that achieved using conventional free-surface and multiphase simulations, we performed another experiment in which the various methods were used to simulate a large water ball dropping into a rectangular body of water under gravity. In this experiment, we did not utilize level-set particles. We simulated the scene with the proposed method (Figure 7 (a)), with the conventional free-surface flow (Figure 7 (b)), with a multi-phase solver (Figure 7 (c)), and with the conventional free-surface flow but with a higher-resolution simulation grid such that the computation time is the same as in Figure 7 (a) (Figure 7 (d)). Figure 7 demonstrates that, compared to the other approaches, the proposed method generates a more realistic result that retains complex details and thin structures.

We note that the liquid-biased filter is designed to precisely capture subcell-sized interfaces, not to compensate volume error. However, the liquid-biased filter is helpful in preserving volume. Complex scene like Figure 4 shows some volume errors over time, as shown in the blue-solid line in Figure 6. Thin structures like shower stems generate mass losses, and it gains volume slightly more than the input flow, due to the topology handling step. However, compared to the setup without liquid-biased filtering (red-solid line), our solver shows better performance in volume conservation. When we measured volume variation for the water-ball drop experiment (Figure 7), it had $+0.56\%$ volume errors between the first frame and the last frame, and $+0.67\%$ for the maximum error at 97^{th} frame. These small errors were occurred because tangling sheet of of air

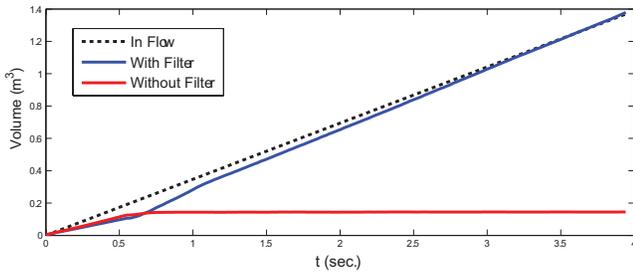


Figure 6: The volume change rates of the shower experiments in Figure 4. Dashed line, blue solid line, and red solid line indicates the input flow, experiment with the liquid-biased filter, and experiment without the liquid-biased filter, respectively.

or isolated bubbles are deleted when topologies was changed.

To simulate micro-scale surface tension effects, we applied surface tension force [Hong and Kim 2005] with geometrical diffusion [Wojtan and Turk 2008]. Even if the process is not entirely physically-correct, we’ve found that the additional diffusion process is quite essential for reducing the artifacts caused by super-sampling method and particle level set correction.

In our experiments, we did not employ high-order schemes, such as BFECC or CIP. We note that such decision was not to replace high-order schemes with the proposed method. In fact, if the proposed method had combined with a high-order scheme, it would have produced richer details. What this paper tries to demonstrate is that the proposed method can produce complex liquid scenes without increasing the overall accuracy of the simulation but via an effective use of computation to more important regions (in the production of visual effects) and explicit user control.

7 Conclusions

In this paper, we have presented a novel framework for simulating complex liquid motion, introduced by phase interface dynamics. We noted that when thin liquid structures make fast movements in air, the interfacial dynamics becomes a dominant component of the liquid motion. We demonstrated that complex phase-interface dynamics can be effectively simulated using the free-surface model by introducing the Eulerian vortex sheet method. By making several extensions to the original vortex sheet method, we devised an interfacial dynamics solver capable of reproducing a wide range of liquid scenes with artistic control.

To track the surface under the complex fluid flow, we employed an extra high resolution grid for surface tracking, in addition to the simulation grid. In this setup, the mismatch caused by the down-sampling from the dense to the coarse grids could result in aliasing errors. In simulating thin liquid structures, loss of liquid volume is more noticeable than that of air volume. We proposed a new filter targeted to this situation, called the liquid-biased filter. This filter was able to downsample the surface without unrealistic volume loss.

Our method has several limitations. By simulating only water with a free surface boundary, it can cause dissipation of air. For example, air bubbles are easily smeared out inside the water volume, rather than rising to the water surface. This is partially due to the use of the liquid-biased filter which expands the water regions. In situations where accurately capturing both air movement and water dynamics is essential, a multiphase solver [Song et al. 2005; Hong and Kim 2005; Losasso et al. 2006] could be used as a basic Navier-Stokes

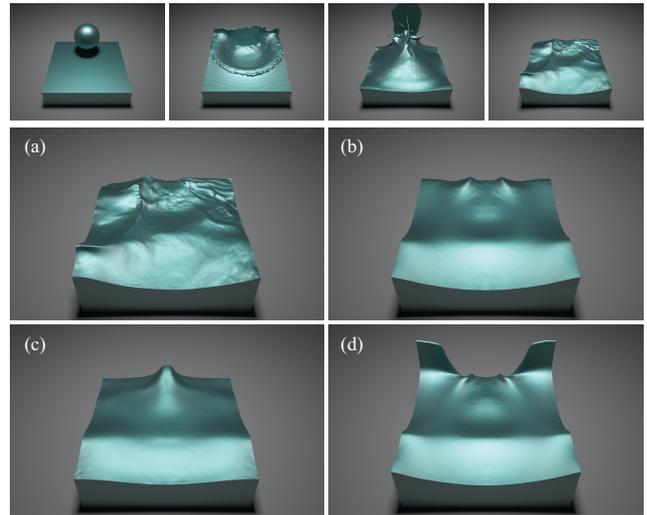


Figure 7: Water-ball drop experiment. The top row shows snapshots generated using the proposed method (vortex sheet with surface super-sampling). The simulation resolution was 96^3 , and the surface tracking resolution was 384^3 . (a) The close-up version of the fourth snapshot in the top row. The rest of the images in the middle and bottom rows were taken during the same simulation as (a). (b) The result of the conventional free-surface flow simulation with the same simulation resolution (96^3). (c) The result of a multiphase simulation with the same simulation resolution (96^3). (d) The result of a conventional free-surface flow simulation but with a higher simulation resolution (128^3).

solver. Our super-sampling model cannot capture very tiny droplets below the super-sampling resolution because it is still based on Eulerian sampling.

Acknowledgements

This work was supported by the Brain Korea 21 Project in 2009, Ministry of Culture, Sports and Tourism(MCST) and Korea Culture Content Agency(KOCCA) in the Culture Technology(CT) Research & Development Program 2009, Seoul Research and Business Development Program (10557), Seoul Metropolitan Government R&D Program (10581), the IT R&D program of MKE/MCST/IITA (ITAA1100090201260001000100100, Development of Digital Clothing Software Technology), ASRI (Automation and Systems Research Institute at Seoul National University), and Ministry of Science and Technology under National Research Laboratory (NRL) grant M10600000232-06J0000-23210.

References

- ADAMS, B., PAULY, M., KEISER, R., AND GUIBAS, L. J. 2007. Adaptively sampled particle fluids. *ACM Trans. Graph.* 26, 3, 48.
- BARGTEIL, A. W., GOKTEKIN, T. G., O’BRIEN, J. F., AND STRAIN, J. A. 2006. A semi-lagrangian contouring method for fluid simulation. *ACM Trans. Graph.* 25, 1, 19–38.
- BRIDSON, R., HOURIHAM, J., AND NORDENSTAM, M. 2007. Curl-noise for procedural fluid flow. *ACM Trans. Graph.* 26, 3, 46.

- CHENTANEZ, N., FELDMAN, B. E., LABELLE, F., O'BRIEN, J. F., AND SHEWCHUK, J. R. 2007. Liquid simulation on lattice-based tetrahedral meshes. In *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 219–228.
- ENRIGHT, D., MARSCHNER, S., AND FEDKIW, R. 2002. Animation and rendering of complex water surfaces. *ACM Trans. Graph.* 21, 3, 736–744.
- ENRIGHT, D., NGUYEN, D., GIBOU, F., AND FEDKIW, R. 2003. Using the particle level set method and a second order accurate pressure boundary condition for free surface flows. In *In Proc. 4th ASME-JSME Joint Fluids Eng. Conf., number FEDSM2003-45144*. ASME, 1–6.
- FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual simulation of smoke. *Computer Graphics (Proc. ACM SIGGRAPH 2001)* 35, 15–22.
- FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. *Computer Graphics (Proc. ACM SIGGRAPH 2001)* 35, 23–30.
- GIBOU, F., FEDKIW, R. P., CHENG, L.-T., AND KANG, M. 2002. A second-order-accurate symmetric discretization of the poisson equation on irregular domains. *J. Comp. Phys.* 176, 1, 205–227.
- GOKTEKIN, T. G., BARGTEIL, A. W., AND O'BRIEN, J. F. 2004. A method for animating viscoelastic fluids. *ACM Trans. Graph.* 23, 3, 463–468.
- HERRMANN, M. 2003. Modeling primary breakup: A three-dimensional eulerian level set/vortex sheet method for two-phase interface dynamics. *Annual Research Briefs, Center for Turbulence Research*.
- HERRMANN, M. 2005. A eulerian level set/vortex sheet method for two-phase interface dynamics. *J. Comp. Phys.* 203, 2, 539–571.
- HONG, J.-M., AND KIM, C.-H. 2005. Discontinuous fluids. *ACM Trans. Graph.* 24, 3, 915–920.
- HONG, J.-M., LEE, H.-Y., YOON, J.-C., AND KIM, C.-H. 2008. Bubbles alive. *ACM Trans. Graph.* 27, 3, 48.
- HOUSTON, B., NIELSEN, M. B., BATTY, C., NILSSON, O., AND MUSETH, K. 2006. Hierarchical rle level set: A compact and versatile deformable surface representation. *ACM Trans. Graph.* 25, 1, 151–175.
- KIM, B., LIU, Y., LLAMAS, I., AND ROSSIGNAC, J. 2007. Advections with significantly reduced dissipation and diffusion. *IEEE Transactions on Visualization and Computer Graphics* 13, 1, 135–144.
- KIM, D., SONG, O.-Y., AND KO, H.-S. 2008. A semi-lagrangian cip fluid solver without dimensional splitting. *Computer Graphics Forum* 27, 2, 467–475.
- KIM, T., THÜREY, N., JAMES, D., AND GROSS, M. 2008. Wavelet turbulence for fluid simulation. *ACM Trans. Graph.* 27, 3, 50.
- LOSASSO, F., GIBOU, F., AND FEDKIW, R. 2004. Simulating water and smoke with an octree data structure. *ACM Trans. Graph.* 23, 3, 457–462.
- LOSASSO, F., SHINAR, T., SELLE, A., AND FEDKIW, R. 2006. Multiple interacting liquids. *ACM Trans. Graph.* 25, 3, 812–819.
- LOSASSO, F., TALTON, J., KWATRA, N., AND FEDKIW, R. 2008. Two-way coupled sph and particle level set fluid simulation. *IEEE Transactions on Visualization and Computer Graphics* 14, 4, 797–804.
- MARSCHNER, S. R., AND LOBB, R. J. 1994. An evaluation of reconstruction filters for volume rendering. In *Proceedings of Visualization '94*, 100–107.
- MIHALEF, V., METAXAS, D., AND SUSSMAN, M. 2007. Textured liquids based on the marker level set. *Comput. Graph. Forum* 26, 3, 457–466.
- MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 154–159.
- NARAIN, R., SEWALL, J., CARLSON, M., AND LIN, M. C. 2008. Fast animation of turbulence using energy transport and procedural synthesis. *ACM Trans. Graph.* 27, 5, 166.
- PERLIN, K. 1985. An image synthesizer. *SIGGRAPH Comput. Graph.* 19, 3, 287–296.
- POZRIKIDIS, C. 2000. Theoretical and computational aspects of the self-induced motion of three-dimensional vortex sheets. *J. Fluid Mech.* 425, 335–366.
- PREMOŽE, S., TASDIZEN, T., BIGLER, J., LEFOHN, A., AND WHITAKER, R. T. 2003. Particle-based simulation of fluids. *Computer Graphics Forum* 22, 3, 401–410.
- SCHECHTER, H., AND BRIDSON, R. 2008. Evolving sub-grid turbulence for smoke animation. In *Proceedings of the 2008 ACM/Eurographics Symposium on Computer Animation*.
- SELLE, A., RASMUSSEN, N., AND FEDKIW, R. 2005. A vortex particle method for smoke, water and explosions. *ACM Trans. Graph.* 24, 3, 910–914.
- SELLE, A., FEDKIW, R., KIM, B., LIU, Y., AND ROSSIGNAC, J. 2008. An unconditionally stable maccormack method. *J. Sci. Comput.* 35, 2-3, 350–371.
- SETHIAN, J. 1999. *Level Set Methods and Fast Marching Methods*. Cambridge University Press.
- SOLENTHALER, B., AND PAJAROLA, R. 2009. Predictive-corrective incompressible sph. *ACM Trans. Graph.* 28, 3, 40.
- SONG, O.-Y., SHIN, H., AND KO, H.-S. 2005. Stable but non-dissipative water. *ACM Trans. Graph.* 24, 1, 81–97.
- SONG, O.-Y., KIM, D., AND KO, H.-S. 2007. Derivative particles for simulating detailed movements of fluids. *IEEE Transactions on Visualization and Computer Graphics* 13, 4, 711–719.
- STAM, J. 1999. Stable fluids. *Computer Graphics (Proc. ACM SIGGRAPH '99)* 33, Annual Conference Series, 121–128.
- TRYGGVASON, G. 1988. Numerical simulations of the rayleigh-taylor instability. *J. Comp. Phys.* 75, 2, 253–282.
- WOJTAN, C., AND TURK, G. 2008. Fast viscoelastic behavior with thin features. *ACM Trans. Graph.* 27, 3, 47.
- ZHU, Y., AND BRIDSON, R. 2005. Animating sand as a fluid. *ACM Trans. Graph.* 24, 3, 965–972.