Water Drops on Surfaces

Huamin Wang Peter J. Mucha Greg Turk
Georgia Institute of Technology *

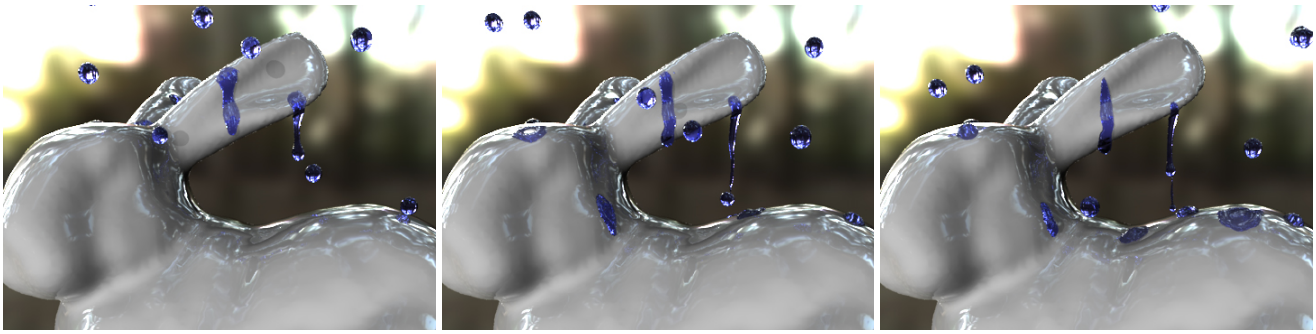


Figure 1: Water dripping off a bunny's ear.

Abstract

We present a physically-based method to enforce contact angles at the intersection of fluid free surfaces and solid objects, allowing us to simulate a variety of small-scale fluid phenomena including water drops on surfaces. The heart of this technique is a *virtual surface* method, which modifies the level set distance field representing the fluid surface in order to maintain an appropriate contact angle. The surface tension that is calculated on the contact line between the solid surface and liquid surface can then capture all interfacial tensions, including liquid-solid, liquid-air and solid-air tensions. We use a simple dynamic contact angle model to select contact angles according to the solid material property, water history, and the fluid front's motion. Our algorithm robustly and accurately treats various drop shape deformations, and handles both flat and curved solid surfaces. Our results show that our algorithm is capable of realistically simulating several small-scale liquid phenomena such as beading and flattened drops, stretched and separating drops, suspended drops on curved surfaces, and capillary action.

CR Categories: I.3.7 [COMPUTER GRAPHICS]: Three-Dimensional Graphics and Realism—Animation;

Keywords: physically based animation, liquid-solid interaction, contact line/angle, virtual surface, water drop

1 Introduction

Although simulating fluids has been an active research topic in graphics for a decade, much of the previous research has been mainly concerned with the liquid free surface motion. Solid surfaces have usually been treated as impermeable boundary conditions and surface tensions between the liquid and a solid are usually ignored. This is quite reasonable for large-scale liquid simulations. However if one wishes to synthesize small-scale liquid motions such as water drops flowing on a glass window, surface tension effects become too strong to be neglected. Our work concentrates on modeling the capillary solid coupling, with surface tensions exerted by the solid object influencing the liquid motions according to different physical parameters such as the force of gravity, the surface tension coefficient, the viscosity coefficient and the affinity between the liquid and the solid material.

Real fluids that come into contact with a solid object form a characteristic angle with the surface of the object known as the *contact angle* [de Gennes 1985]. The contact angle for so-called *hydrophobic* surfaces causes water to bead up, while a *hydrophilic* surface allows a drop of water to spread out. We will use the term *affinity* to describe the hydrophobicity or hydrophilicity of a surface. The affinity between water and a surface affects not only the behavior of a static drop of water, but also greatly influences the motion of a moving drop. The fluid/solid interaction can also be seen to affect the behavior of drop merging and splitting and the motion of water rivulets.

The core of our algorithm is a *virtual surface* method, which allows us to simulate small-scale behaviors of fluids with important contact angle effects. Given a stable contact angle, this method estimates the appropriate surface tension at the contact line between the solid surface and liquid surface in a fluid solver. It implicitly constructs a virtual surface penetrating into the solid surface, replacing the original liquid-solid surface by the virtual surface, and estimates surface tension using this newly created surface. Unlike some other models that are focused on modeling axisymmetric water drops, this method can handle arbitrary 3D liquid shapes by using implicit signed distance functions to represent all surfaces. When the solid surface is sufficiently smooth, our virtual surface method accurately approximates the true surface tensions.

Water (or any other liquid) is effectively defined here by its viscosity and its surface tension against the air. The dominant factor distinguishing the water motion for drops of the same size is the affinity between water and the solid material, as can be quantified by the stable contact angle between the liquid-air and liquid-surface interfaces. Here we use a simple dynamic contact angle model for

*e-mail:{whmin@cc, mucha@math, turk@cc}.gatech.edu

Copyright © 2005 by the Association for Computing Machinery, Inc.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail permissions@acm.org.
© 2005 ACM 0730-0301/05/0700-0921 $5.00

capillary solid coupling in terms of three contact angles: the receding contact angle, the wet advancing contact angle and the dry advancing contact angle. Our results indicate this model is sufficient for simulating many small-scale fluid motions.

Our fluid solver represents and updates the liquid surface using the particle level set method. Compared with other approaches such as molecular particle dynamics or adaptive Lagrangian meshing, the level set distance function can efficiently simulate a drop's internal fluid dynamics and can easily handle drop breakup and coalescence. Since the liquid volume only occupies a small portion of the whole domain in most small-scale liquid simulations, we use a sparse, piecewise representation of the grid in the fluid solver to save both computation time and memory.

2 Related Work

The related work that we cover comes from three different areas: previous methods for synthesizing drop motions in graphics, computational fluid dynamics and its application in liquid simulations, and research on surface tension in physics.

In graphics, most previous water drop systems provide various ways to model water drops efficiently but remain incapable of capturing some of the physical drop motions that we observe in the real world. Dorsey et al. [1996] used a particle system to synthesize drops and their effects on weathering appearance textures for large solid models, assuming that each drop's deformation is too small to be noticeable. Kaneda et al. [1993; 1996; 1999] used a particle system to simulate water drops flowing on a flat surface. Flowing water drops are modeled in Fournier et al. [1998] by a mass-spring system with surface tension and volume conservation constraints. Though the mass-spring system allowed various efficient simulations, it has difficulty handling the drop separating and merging processes, especially when many drops interact in a large scene. Yu et al. [1999] successfully modeled static droplet shapes on flat surfaces using a metaball concept, and Tong et al. [2002] later presented a volume-preserving approach to model water flows using metaballs, but neither considered any surface tension effects on the moving interface. Generally speaking, the above methods do not consider interaction between the fluid dynamics internal to the water drops and the surface tension at the liquid interfaces, making it relatively difficult to simulate a wide range of drop deformation and motion realistically and accurately.

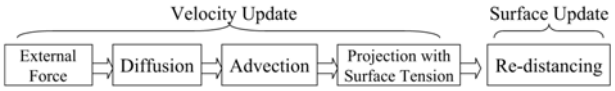
Computational fluid dynamics has been successfully and practically applied to simulate fluid animation in graphics since Foster and Metaxas [1996]. Shortly after that, the stable fluid method was introduced by Stam [1999], in which the semi-Lagrangian method is used to handle liquid velocity advection. In a series of papers, Enright, Fedkiw and Foster [2001; 2002] used the level set method to evolve liquid surfaces so that more complex liquid motions can be simulated. They further showed how to combine the level set method with particles (the so-called *particle level set*) to reduce volume loss and increase the surface accuracy. For a large viscosity, the time step must be extremely small according to the CFL condition when one solves the viscosity term using explicit schemes. Stam [1999] showed the viscosity term can be solved with larger time steps using the implicit Euler method, assuming a uniform viscosity distribution. Recently Losasso et al. [2004] demonstrated the use of an octree structure for surface evolution instead of a regular grid so that more surface details can be maintained. Surface tensions in [Losasso et al. 2004] are used as first order Dirichlet pressure boundary conditions on air boundary cells by estimating mean curvatures from the surface's signed distance function. The second order pressure boundary condition scheme was presented in [Enright et al. 2003].

In physics, chemistry and material science, researchers have performed numerous experiments to understand the liquid-solid interfacial tension and developed various simulation techniques for treating the liquid-solid interactions. Korlie [1997] simulated a liquid drop on a flat solid surface using quasi-molecular particles. Feng et al. [2002] studied the drop impact and flattening process using Lagrangian meshing by the finite element method. Bussman et al. [1999] developed a volume tracking algorithm for the volume-of-fluid method, and they successfully simulated single drop splashing and impact on curved shapes in their later work, treating the contact angle as an immediate boundary condition. Healy [1999] used the 2D level set method and enforced the contact angle by modifying the liquid-air surface immediately to simulate an axisymmetric drop impact on a flat surface. Zhao et al. [1998] demonstrated drop falling and depositing effect using a variational level set evolution equation obtained by minimizing the surface tension energy. Sussman et al. [1998] first proposed the virtual surface idea for flat solid surfaces in 2D and in axisymmetric geometries to constrain contact angles. Renardy et al. [2001] later implemented the same idea for the volume-of-fluid method, and their algorithm was also limited to flat solid surfaces in 2D. To our knowledge, there are still no previously published methods to model 3D interfacial tensions for arbitrarily curved solid surfaces.

3 Algorithm Overview

Before we examine the capillary solid coupling problem, we first describe our fluid solver and discuss how the virtual surface method will be incorporated into it. Our fluid solver uses a finite difference formulation on a rectilinear 3D grid. Fluid velocities are stored at the faces of the grid cells. The fluid-air interface is represented by a signed distance field which is moved using the particle level set technique. The signed distance field ϕ for the fluid-air interface is central to our approach because it is this field that is modified to create the virtual surface.

Like many fluid simulation programs, we split the simulation procedure into a few main steps. In our case, the first four update the velocity field while the last step updates the liquid surface using the velocity field.



External forces such as gravity are first applied to update the velocity field. We then use the implicit Euler method to solve for viscous momentum diffusion and the semi-Lagrangian method to calculate the velocity advection. The final projection step solves a Poisson equation to make the velocity field divergent free. In this step, we use the calculated surface tension as the first-order boundary pressure condition at air boundary cell for the Poisson equation. After updating the velocity field, we extrapolate the velocity field using a fast algorithm [Enright et al. 2002], we evolve the liquid surface using the particle level set method with the HJ-WENO scheme [Jiang and Peng 2000], and we complete the signed distance function by the fast marching algorithm [Tsitsiklis 1995; Sethian 1996].

The remainder of the paper is organized as follows. We first review the background on surface tension in Section 4. In Section

5 we show how to construct the virtual surface and how to estimate mean curvatures for curved solid surfaces in 3D. After this, we present a dynamic contact angle model in Section 6 for choosing the contact angle, which can be used to realistically synthesize various small-scale liquid phenomena. Results and conclusions are given in Sections 8 and 9.

4 Physical Background

Surface tension (*interfacial tension*) is an important factor in small-scale liquid simulations. It is caused by unbalanced molecular cohesive forces in the interfacial region where two phases meet (liquid-air, liquid-solid or solid-air). There are two ways to analyze the surface tension's influence on the liquid motion. One is to use the surface tension force imposed onto the liquid surface directly in the incompressible Navier-Stokes equation (Eq. 1),

$$\mathbf{u_t} = -(\mathbf{u}\cdot\nabla)\mathbf{u} + \nu\nabla(\nabla\mathbf{u})/\rho - \nabla P/\rho + (\mathbf{F} - \gamma\kappa\cdot\mathbf{N})/\rho\,, \quad \nabla\cdot\mathbf{u} = 0\,, \tag{1}$$

where $\mathbf{u}$ is the velocity field, ν is the viscosity coefficient, ρ is the liquid density, κ is the surface mean curvature, $\mathbf{N}$ is the liquid surface normal vector, $\mathbf{F}$ is the external force and γ is the surface tension coefficient. Surface tension can be similarly represented in terms of the pressure difference across the surface, according to Laplace's Law,

$$\Delta P_{surf} = \gamma\cdot\kappa\,. \tag{2}$$

where ΔP_{surf} is the pressure difference across the liquid surface. Both representations describe surface tension as being linearly dependent with respect to the surface mean curvature κ.

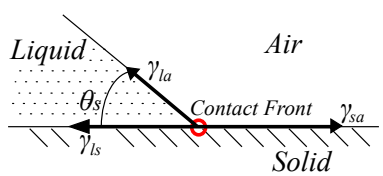


Figure 2: Contact front in equilibrium

There are three different interfaces upon which surface tensions act on the contact front where a liquid surface meets a solid object: liquid-air, liquid-solid, and solid-air. According to Young's relation (Fig. 2) [de Gennes 1985], if the contact line at the intersection of these three interfaces reaches equilibrium with no external forces, then the surface tensions satisfy

$$\gamma_{sa} - (\gamma_{la}\cos\theta_s + \gamma_{ls}) = 0 \tag{3}$$

where $\theta_s (0 < \theta_s < \pi)$ is the stable contact angle, and γ_{ls}, γ_{sa} and $\gamma = \gamma_{la}$ are interfacial tension coefficients for the liquid-solid, solid-air and liquid-air surfaces, respectively. Since it is difficult to measure surface tension directly, the stable contact angle is a common term used to quantify the affinity between a liquid and solid material. When θ_s is small (say, close to zero), the solid surface is said to be *hydrophilic*, and the liquid surface tends to spread flat. The solid surface is called *hydrophobic* if θ_s is large, and the liquid tends to bead up on the surface.

When external body forces act on the fluid, the actual equilibrium contact angle between the solid surface and liquid surface may slightly differ from θ_s. For example, the observed stable contact angle can be smaller than θ_s when a drop sits on a table, as the curvature at the contact line becomes slightly positive to hold the pressure due to gravity.

Symbols	Definition
γ	The surface tension
κ	The mean curvature $\nabla(\nabla\phi/\lvert\nabla\phi\rvert)$
ϕ	The signed distance function to some surface
Ω_s	The solid surface
Ω_l	The liquid surface with ϕ_l, or simply ϕ
Ω_{la}	The liquid air surface with ϕ_{la}
Ω_{ls}	The liquid solid surface with ϕ_{ls}
Ω_v	The virtual surface with ϕ_v
Ω_{new}	The new surface with ϕ_{new}

Table 1: Some commonly used symbols

The characteristics of the liquid motion depend greatly on the liquid scale. Water moving in a large tank will behave in an entirely different manner than a water droplet that is flowing on a table. A number of different dimensionless numbers are commonly used to characterize the relative scales of different forces. For instance, the Bond number is defined as the ratio between typical gravitational and surface tension forces, the Weber number describes the ratio between inertial and surface tension forces, and the Capillary number describes the ratio of viscous and surface tension forces. For our purposes here, it is sufficient to note that the capillary length of the liquid-air interface for water under gravitational acceleration is $\sqrt{\gamma/(\rho g)} \approx 4\,\text{mm}$. At scales orders of magnitude larger than this, surface tension effects are difficult to discern; but flows on the scales of a few capillary lengths (centimeters) typically have important surface tension and contact angle effects.

5 Virtual Surface Extrapolation

Our virtual surface approach makes use of the signed distance field ϕ that represents the liquid-air interface in order to simulate contact angle effects. The virtual surface can accurately capture the effects of each surface tension force on the contact line through a series of steps. The virtual surface is extended into the solid at the desired stable contact angle that would balance the surface tension forces on the contact line as in Young's relation (Eq. 3). If the actual surface is not at the desired contact angle, the resulting kink has a non-zero curvature. The appearance of this curvature in the pressure boundary condition of the projection step yields the desired forces.

Our method modifies the original liquid surface Ω_l around each contact front cell independently so that the curvatures calculated from the modified surface Ω_{new} correctly take all interfacial tensions into account. The full liquid surface Ω_l is the union of the liquid-air surface Ω_{la} and liquid-solid surface Ω_{ls}, with the full surface defined implicitly in the distance field ϕ. By modifying ϕ, we replace Ω_{ls} by a virtual surface Ω_v. We then estimate the mean curvature on this new surface and use the curvature as the surface tension pressure (Eq. 2). Using original liquid surfaces to estimate surface tensions directly without the virtual surface method is the same as using π as the stable contact angle with the virtual surface method, which would greatly limit the range of phenomena that could be simulated.

We will now define the virtual surface. Let $L(t)$ be the curved contact line between the solid surface Ω_s and the liquid surface Ω_l (Fig. 3). Let $N_s(t)$ be the surface normal of the solid and let $N_l(t)$ be the liquid surface normal on $L(t)$. The value t is a unit arc length parameter ($|L'(t)| = 1$), and t is chosen so that as t increases, the position $L(t)$ rotates counter-clockwise around the normal $N_s(t)$. By

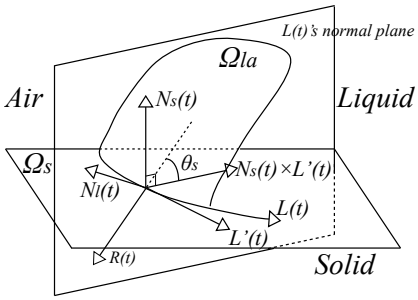


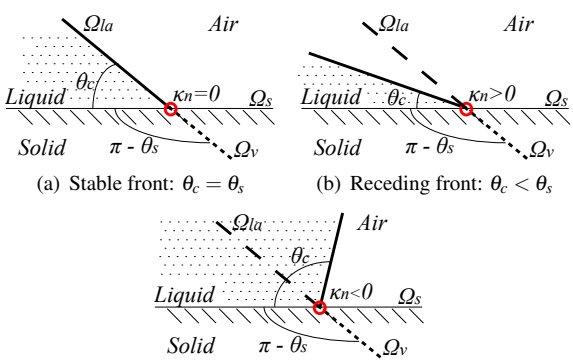
Figure 3: The solid surface and liquid air surface

definition, $N_s(t)$ and $N_l(t)$ define a plane that is normal to the contact line $L(t)$. The angle between $N_s(t)$ and $N_l(t)$ defines the contact angle between Ω_s and Ω_l. Our virtual surface begins along $L(t)$ and extends down into the solid at a specific angle dictated by the solid-fluid affinity. Specifically, the virtual surface $V(s,t)$ is defined to be

$$\begin{aligned} V(s,t) = &\quad L(t) + sR(t) \quad (for\ \ s > 0)\,, \\ R(t) = &\quad -\sin\theta_s \cdot N_s(t) - \cos\theta_s \cdot (N_s(t) \times L'(t))\,. \end{aligned} \tag{4}$$

For a given t_0, $V(s,t_0)$ is geometrically a ray shot from $L(t_0)$ in the direction $R(t_0)$, and it has an angle of $\frac{\pi}{2} + \theta_s$ with $N_s(t)$ in $L(t)$'s normal plane.

Let us examine the virtual surface in $L(t)$'s normal plane (Fig. 4a). If the current angle θ_c equals the stable contact angle θ_s, then the contact line should be stable in the normal plane by definition. This is justified in the virtual surface method because the curvature κ_n in the normal plane is zero when N_s and N_l coincide. When θ_c is not equal to θ_s, using the mean curvature calculated by the virtual surface method provides a new way to estimate dynamic surface tensions on the contact line. Fig. 4b shows the receding case ($\theta_c < \theta_s$) and Fig. 4c shows the advancing case ($\theta_c > \theta_s$).



(a) Stable front: $\theta_c = \theta_s$ (b) Receding front: $\theta_c < \theta_s$

(c) Advancing front: $\theta_c > \theta_s$

Figure 4: The contact front in 2D.

The input ϕ_l (or simply ϕ) to the virtual surface method is the original distance function of the liquid surface Ω_l, including Ω_{la} and Ω_{ls}. For open surfaces, such as Ω_{la} and Ω_{ls}, we define Ω in a region if for any point P, its closest point is not on Ω's boundary. We then define the full closed liquid surface through ϕ as the minimum defined value of ϕ_{la} (the distance to Ω_{la}) and ϕ_{ls} (the distance to Ω_{ls}). This signed distance will then be modified to create a new distance function ϕ_{new} in which Ω_{ls} is replaced by the virtual surface Ω_v. All surfaces are represented by signed distance functions with no explicit formulations here.

In the remainder of this section, we first show how the virtual surface method works on contact front cells ($\phi = 0$) in 2D and 3D for a flat solid surface. We then describe how the method is naturally extended to general boundary cells and curved solid surfaces. We also assume the stable contact angle θ_s is unique in this section. In Section 6, we will discuss how to choose the contact angle according to different situations, including moving fluid surfaces and pre-wetted solid surfaces.

5.1 Modifying Surfaces in 2D

In 2D, the contact front is a single point, and the virtual surface is simply a ray extending from this contact point into the solid. We create the virtual surface by modifying the distance field values of ϕ to accurately reflect the distance between locations inside the solid and the closest point on the virtual surface. We then merge the virtual surface with the liquid-air surface, which is obtained from the original liquid surface.

For a given contact point, we locally modify the liquid surface in a small stencil of grid cells of ϕ, and these updated values will later be used to estimate curvatures. Our first-order curvature estimation scheme only requires the stencil size to be three nodes in each dimension. Our method can be easily extended, however, to handle larger stencil boxes when higher order curvatures are demanded.

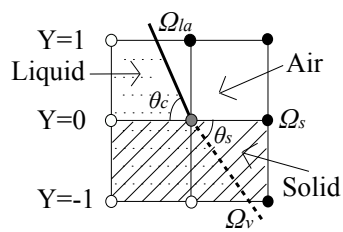


Figure 5: The 2D stencil box. The empty dots are water nodes, and the solid dots are air nodes. The Y=0 plane is the solid surface and the Y direction is the solid surface normal. The solid line is the liquid-air surface and the dashed line is the virtual surface.

Let C be a 3×3 stencil box centered at the contact front as shown in Fig. 5. Without loss of generality, we assume the center $C_{0,0}$ is at the contact front ($\phi(0,0) = 0$), $C_{-1,0}$ is in water ($\phi(-1,0) < 0$) and $C_{1,0}$ is in air ($\phi(1,0) > 0$). In order to compute the new distance function ϕ_{new}, the first step is to calculate the virtual surface's distance function ϕ_v from each node on the Y=0 and Y=-1 planes inside of the solid. Let ψ be the distance to the contact point from each node on the Y=0 plane, then by definition we know $\psi(0,0) = 0$, $\psi(-1,0) = -h$ and $\psi(1,0) = h$ where h is the node size. For nodes on the Y=0 plane, ϕ_v is:

$$\phi_v(x,0) = \begin{cases} \psi(x,0)\sin\theta_s & x\cos\theta_s > 0 \\ \frac{x}{|x|}h & otherwise \end{cases} \tag{5}$$

For nodes on Y=-1 plane, ϕ_v is:

$$\phi_v(x,-1) = \begin{cases} [\psi(x,0) - h\cos\theta_s]\sin\theta_s & x\cos\theta_s + \sin\theta_s > 0 \\ \frac{x}{|x|}(\psi^2(x,0) + h^2)^{1/2} & otherwise \end{cases} \tag{6}$$

Given the virtual surface's distance function ϕ_v defined above for all nodes Y≤ 0, we combine ϕ_v with the liquid-air distance function ϕ_{la} defined for all nodes Y≥ 0 to form a new distance function ϕ_{new}. We first determine ϕ_{new} for Ω_{new}'s boundary nodes on each side, then estimate ϕ_{new} for the rest of the nodes using the fast marching

algorithm from the boundary nodes. For ϕ_v's boundary nodes on the Y=-1 plane, $|\phi_v|$ is less than h, which means Ω_v is definitely closer than Ω_{la}. Therefore, for those nodes, the virtual surface's boundary nodes on the Y=-1 plane are also the new surface's boundary nodes, and $\phi_{new} = \phi_v$. Similarly, the liquid surface's boundary nodes on the Y=1 plane are also the new surface's boundary nodes: $\phi_{new} = \phi_{la} = \phi_l$, since the liquid-solid surface is beneath the solid surface Ω_s : Y=0. We finally determine the new surface's boundary nodes on the Y=0 plane. $C_{0,0}$ is definitely a boundary node for the new surface, with $\phi_{new} = 0$ by definition. For other two nodes, if they are air boundary nodes, we determine their values as:

$$\phi_{new} = \begin{cases} \phi_{la} \\ \phi_v \end{cases} = \begin{cases} \phi_l & |\phi_{la}| < |\phi_v| \\ \phi_v & otherwise \end{cases} \tag{7}$$

We can determine the values at air boundary nodes because the liquid-solid surface is a thin surface beneath the solid plane, so ϕ_{la} is still equal to ϕ_l for air-side nodes. Meanwhile, water boundary nodes are so close to the liquid-solid surface that their ϕ_l may in fact be equal to ϕ_{ls}.

ϕ_v and ϕ_l can be further used to fix any numerical errors that occur when estimating the new distances by the fast marching method. If a node is on the Y=0 or the Y=-1 plane, we bound its value ϕ_{new} by ϕ_v. If an air node is on the Y=0 or the Y=1 plane, we bound its value ϕ_{new} by ϕ_l. Again, we do not consider ϕ_l for any water nodes for the same reason described before: ϕ_l may not be equal to ϕ_{la}, but equal to ϕ_{ls}. Fig. 6 shows a simulation of 2D capillary action using the 2D virtual surface method. The small contact angle causes a column of water to be drawn up into the thin tube. Also note the bending of the lower water surface.

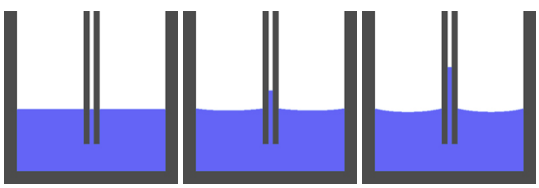


Figure 6: 2D capillary action. Solid surfaces are all hydrophilic.

5.2 Modifying Surfaces in 3D

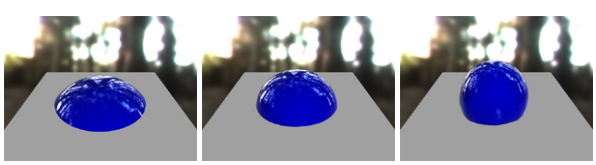


Figure 7: Stable drops sitting on the ground with different stable contact angles θ_s. Due to gravity, the actual angle between the drop surface and the ground is slightly smaller than θ_s.

The virtual surface method in 3D is similar to that in 2D: first calculate the distance function ϕ_v to the virtual surface, then merge it with the liquid-air surface by calculating ϕ_{new} on the new surface's boundary nodes. The stencil box C is now a $3 \times 3 \times 3$ cube that is centered on the contact line ($\phi_l(0,0,0) = 0$), and the Y-axis is the constant solid normal direction.

The major difference for 3D is that the contact front in 3D is a curve on the solid surface plane, which causes new difficulty in determining the virtual surface's distance function ϕ_v. Fortunately we can show that if the contact line is sufficiently smooth and if the stable contact angle θ_s is not extreme (small $|\cos\theta_s|$), this 3D virtual surface's shortest distance problem can be reduced to the 2D ϕ_v's shortest distance problem in $L(t)$'s normal plane. The solution to this 2D case has already been given in Eq. 5 and 6. We include the proof to a supporting claim in the appendix for the interested reader. According to the 2D solution, it is not necessary to know the exact position of the closest point $L(t_0)$ on $L(t)$. We do, however, need to know the shortest distance ψ from any node on the Y=0 plane to the contact line $L(t)$. Here we will show how to recover ψ from the original liquid surface's distance function ϕ_l.

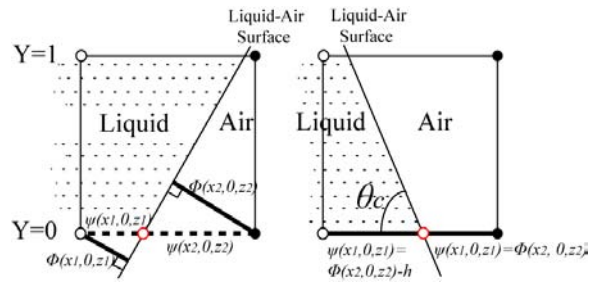


Figure 8: Two possible cases in estimating the distance ψ to the contact line. The red dot is the contact front.

Let $C_{x1,0,z1}$ and $C_{x2,0,z2}$ be two neighboring boundary nodes of the liquid surface such that $\phi_l(x1,0,z1) < 0$ and $\phi_l(x2,0,z2) > 0$. Also let $C_{x1,1,z1}$ and $C_{x2,1,z2}$ be two nodes above them on the Y=1 plane. There are two possible cases for estimating ψ according to different liquid contact fronts (Fig. 8). In the left case, when the current angle $\theta_c > \pi/2$, assuming that the liquid surface is sufficiently smooth, then $\frac{\psi(x1,0,z1)}{\psi(x2,0,z2)} = \frac{\phi_l(x1,0,z1)}{\phi_l(x2,0,z2)}$, and therefore,

$$\begin{aligned} \psi(x1,0,z1) &= \frac{\phi_l(x1,0,z1)}{|\phi_l(x1,0,z1)-\phi_l(x2,0,z2)|}h \\ \psi(x2,0,z2) &= \frac{\phi_l(x2,0,z2)}{|\phi_l(x1,0,z1)-\phi_l(x2,0,z2)|}h \end{aligned} \tag{8}$$

where h is the node size. In the right case, when $\theta_c < \pi/2$, we have $\psi(x1,0,z1) = \phi_l(x2,0,z2) - h$ and $\psi(x2,0,z2) = \phi_l(x2,0,z2)$. Here we classify the boundary pair to be the left case if $\phi_l(x2,0,z2) > \phi_l(x2,1,z2)$; otherwise the right case applies.

For each boundary pair, ψ is calculated as given above. If one node is included in more than one boundary pair, the distance is chosen to be the one with the smallest absolute value. After determining ψ for all boundary nodes, we use the 2D fast marching method to estimate ψ for the rest of the nodes on the Y=0 plane.

Once we have calculated the virtual surface distance function ϕ_v, we determine the new surface's boundary nodes in a similar manner to the 2D case. Boundary nodes on the Y=$\pm$ 1 planes can be immediately determined by the virtual surface and liquid surface's boundary nodes. For air boundary nodes on the Y=0 plane, the new distance is chosen using Eq. 7. The water boundary nodes are ignored for the same reason as before. After that, ϕ_{new} for non-boundary nodes are estimated using the 3D fast marching method, and they are further corrected using the known shortest distances to the virtual surface and to the liquid surface respectively, again similar to the 2D case. Fig. 7 shows the shapes of drops with different stable contact angles.

5.3 Virtual Surfaces on Curved Solid Shapes

So far we have described how to modify the surface near the contact front so that the surface tension can be estimated on a flat solid surface. This results in maintaining a characteristic stable contact angle. We will now discuss contact angles on curved solid surfaces.

For each air boundary cell in a level set grid, surface tension is calculated as the boundary condition for the pressure projection step. Since those air boundary cells may not be exactly on the contact front, we cannot apply the virtual surface modification directly. One straightforward way to resolve this would be to first find the closest contact point to the air boundary cell, then use the surface tension estimated at this position as a Dirichlet boundary condition. Unfortunately, it is quite difficult to form a second order or higher boundary condition scheme for those closest points not aligned with grid cells at all. Further, finding the closest contact point depends heavily on estimating the surface's signed distance gradient correctly, which is relatively unstable especially on highly curved surfaces.

Here we use a simple first order boundary condition scheme in our method. We choose the stencil box to be centered at the air boundary cell and the stencil node size to be the same as the grid cell size. We determine the stencil box's coordinate systems by surface normals: The stencil's Y axis is the solid normal direction $Y = N_s$ and the stencil's X axis is the orthogonalized liquid normal direction $X = N_l - (N_s \cdot N_f)N_s$. Then we sample the liquid distance function for each stencil node using linear interpolation, and subtract $\phi(0,0,0)$ from each stencil node's distance value. The actual liquid surface represented by the stencil box is the liquid surface's iso-contour where the distance equals $\phi_l(0,0,0)$. We modify this iso-surface represented in the stencil box and estimate the surface tension for that air boundary cell even if it is not immediately on the contact front. We then calculate the surface mean curvature for the center node $C_{0,0,0}$ in the stencil box by Eq. 9 from [Osher and Fedkiw 2002]:

$$\kappa = (\phi_x^2\phi_{yy} - 2\phi_x\phi_y\phi_{xy} + \phi_y^2\phi_{xx} + \phi_x^2\phi_{zz} - 2\phi_x\phi_z\phi_{xz} + \phi_z^2\phi_{xx} + \phi_y^2\phi_{zz} - 2\phi_y\phi_z\phi_{yz} + \phi_z^2\phi_{yy})/|\nabla\phi|^3 \tag{9}$$

where the first and second derivatives of ϕ are estimated using second order finite difference formulae.

Since the stencil's X direction is the orthogonalized liquid normal direction and we have $\phi_l(0,0,0) = 0$, node $C_{0,-1,0}$ should be inside of the liquid surface iso-contour. However, this may not be true in some cases, such as when the liquid normal directions have not been accurately estimated. In that case, we bound $\phi_l(0,-1,0)$ to be always less than some maximum value $\varepsilon = -10^{-6}$ so that the virtual surface construction will not fail because of missing the contact line when no nodes on the Y=0 plane satisfies $\phi_l < 0$.

Although the distance to the virtual surface that is calculated by this method is not exact, the value is still a good approximation to the actual shortest distance, given a sufficiently smooth solid surface or a small stencil size. Our calculations show that the virtual surface method can estimate surface tensions robustly and accurately.

6 Dynamic Contact Angle Model

In the real world, a unique stable contact angle is not sufficient to model fluid drop movements on solid surfaces. For instance, the phenomenon called *contact angle hysteresis* [Schwartz and Garoff 1985], where a tiny drop is suspended on a vertical plane, cannot be modeled with a single stable contact angle. This phenomenon requires a dynamic contact angle model, and can indeed be captured using two stable contact angles: a receding (minimum) stable contact angle θ_s^r and an advancing (maximum) stable contact angle θ_s^a. Any angle between θ_s^r and θ_s^a can be a valid stable contact angle before the contact front starts to move.

Because it is sufficient to capture the effect of hysteresis, we use a simple dynamic model with two stable contact angles set by the contact front velocity in the liquid surface's normal direction. In the advancing case when the velocity is moving into previously dry regions, we use the advancing contact angle θ_s^a; otherwise, we use the receding contact angle θ_s^r. If the contact line is static (the velocity is below some threshold), we first calculate both boundary pressures P_r and P_a using θ_s^r and θ_s^a, respectively. Since we assume $\theta_s^r \le \theta_s^a$, $P_r \le P_a$ and we then choose the actual pressure to be:

$$P = \begin{cases} 0, & if \quad P_r \cdot P_a < 0 \\ P_a, & if \quad P_a < 0 \\ P_r, & if \quad P_r > 0 \end{cases} \tag{10}$$

The actual values of the receding and advancing stable contact angles depend on the properties of both the liquid and the solid. The contact angles can also depend on the wetness of the solid surface. If the solid surface has already been wet, liquid remaining on the surface can help subsequent drops move more freely on the surface. For such wettable solid surfaces, we maintain a *wetting history map* for the grid in order to indicate which regions have already been wetted. We then use a wetted advancing contact angle θ_{s-w}^a smaller than the dry advancing contact angle θ_{s-d}^a. We do not discriminate between the wet and the dry cases for the receding contact angle since the receding angle moves into a wetted region in both cases.

7 The Sparse Grid Representations

In order to simulate complex drop interactions, the grid domains that we use can become significantly larger than those in other liquid simulations. A typical grid domain in our experiments can contain $400 \times 400 \times 400$ grid cells. Fortunately, the liquid volume only occupies a small portion of the whole domain space. We use a sparse grid representation in which the domain is first subdivided into $8 \times 8 \times 8$ box regions. If the region contains any liquid or if it is close to the liquid surface, we activate this region and allocate memory for it. Otherwise, the region is inactive and no computation time or memory is used for the region.

8 Applications and Results

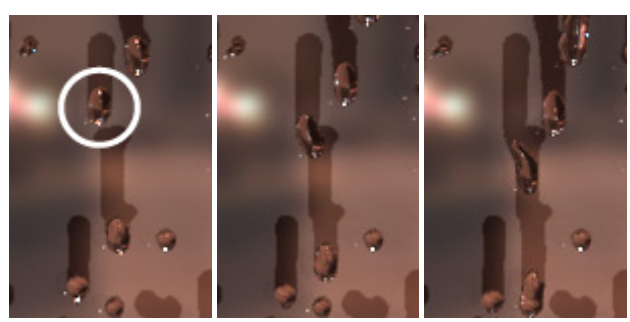


Figure 9: The circled drop follows a previous drop's path.

We have integrated the virtual surface method into our fluid solver and we have simulated several different small-scale liquid motion

scenarios. Typically each simulation takes 5-8 days to simulate on one Pentium Xeon 2.8GHZ Workstation. Since the computation domain space is huge, our simulations are still relatively time consuming even though our algorithm works efficiently.

For completeness, we discuss the simulation parameters used for our simulations. For simplicity, we take constant time steps updating velocities every $2 \cdot 10^{-4}$ second. The fluid in each of our simulations here is taken to be water, as defined by its physical properties: the surface tension between the liquid-air interface is $\gamma = 73\,\mathrm{g/s^2}$ (at room temperature) and the viscosity is $\nu = 0.01\,\mathrm{cm^2/s}$. We apply no-slip conditions on the solid surface. The only external force used here is the acceleration due to gravity $g = 980\,\mathrm{cm/s^2}$. The typical drop size in our simulation is from 2mm to 6mm. We use a second order Runge-Kutta scheme to trace particles for both the semi-Lagrangian method in the velocity advection step and for the particle level set method. Our Poisson solver uses the preconditioned conjugate gradient method with a modified incomplete Cholesky decomposition preconditioner. Since the water drop's velocity varies greatly and the volume loss is severe only on high-velocity surfaces, in the particle level set method, we choose the particle number for each grid cell according its velocity magnitude with a maximum of 32 particles per cell. Using the particle level set method dramatically reduces volume loss during simulations.

When the grid cell size is not sufficiently small, surface tension estimations for small drops are less accurate and may cause instability in drop motions. Fortunately, surface tensions on small drops can be ignored since their visual effects are hardly noticeable. In our experiment, we did not use surface tension if the water drop only contains 27 grid cells or less.

We have created several animated scenes based on our method of simulating fluid with interfacial tension. The first simulation considers flat window panes with varying surface properties, showing how the solid surface property and randomly added drops can influence the water drop's flowing paths. In the beginning, water drops are identically distributed on each pane. The left pane has $\theta_s^a = 90^\circ$ and $\theta_s^r = 60^\circ$. The middle pane is more hydrophilic in the wake of the falling drop, with $\theta_s^a = 90^\circ$ and $\theta_s^r = 30^\circ$. In addition, we use a maximum receding surface tension bound to enhance the hysteresis effect. The right pane has similar contact angles as the left pane except that it also uses the wetting history and $\theta_{s-w}^a = 60^\circ$. Compared with those on the left pane, the drops on the middle pane leave longer trails because the receding contact angle is small and the receding surface tension is limited. On the right pane, the solid surface becomes wet after water drops flow on it, so that a water drop is likely to follow the previous drop's path (Fig. 9). These three panes' affinities to water are similar to those of plastic, glass and marble, respectively.

The second and third simulations show a pipe and a bunny, with water dripping onto these surfaces from a height of roughly 0.1 meter. The surface of the pipe is represented analytically, and a distance field is used to represent the bunny. For these solid surfaces, $\theta_s^a = 90^\circ$ and $\theta_s^r = 60^\circ$. The tube is tilted at an angle of 10° from horizontal. Notice the behavior of drops on the tube's bottom and the bunny's ears. Surface tension holds a drop from leaving the solid surface until enough water accumulates so that the drop becomes sufficiently large (Fig. 11 and 12). Often when the drop leaves the surface, tiny satellite drops are formed when the thin connecting strand of water snaps.

The leaf in the fourth simulation is comprised of two planes spanning an angle of 120°, and the leaf axis is tilted at an angle of 15°. Fig. 10 shows a sequence of a drop hitting the leaf and merging with its neighbor. Notice that drops flatten when they first hit the leaf, but then bead up due to the hydrophobic nature of the leaf. Also note the manner in which separate water drops flow to the middle of the leaf and join to form larger and longer rivulets (Fig. 12).

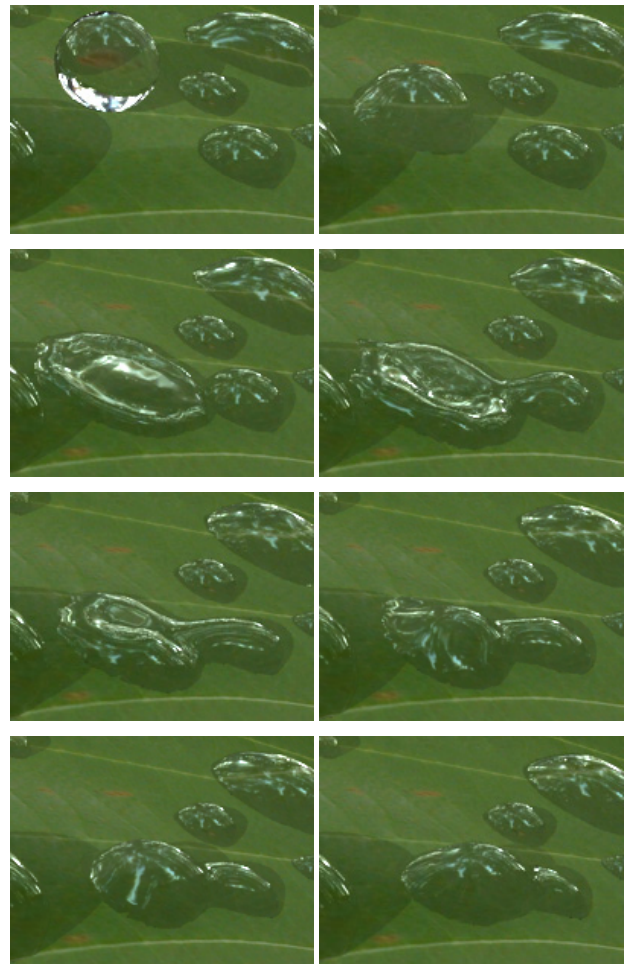


Figure 10: Drop impacts on a leaf. A flattened drop appears in the upper-right image. Time advances left to right, then top to down.

To generate the rendered images, we construct triangle meshes for the liquid surface using the marching cubes algorithm. Images were synthesized using our rendering program based on the physically-based ray tracer (pbrt) [Pharr and Humphreys 2004]. The environment maps are high dynamic range images from Paul Debevec's Light Probe Image Gallery. The leaf texture image is from Mayang's free texture library (http://www.mayang.com/textures).

9 Conclusion and Future Work

In this paper, we have presented an algorithm to solve the capillary solid coupling problem by modeling surface tensions between the liquid and solid object. The virtual surface method replaces the liquid-solid surface by a virtual surface beneath the solid surface so that the estimated boundary pressure can represent all surface tensions on the contact front. We use a dynamic contact angle model to choose different stable contact angles according to the contact front's velocity and surface wetness. Our results show that the algorithm is robust, accurate, and ready to be incorporated into level set fluid solvers.

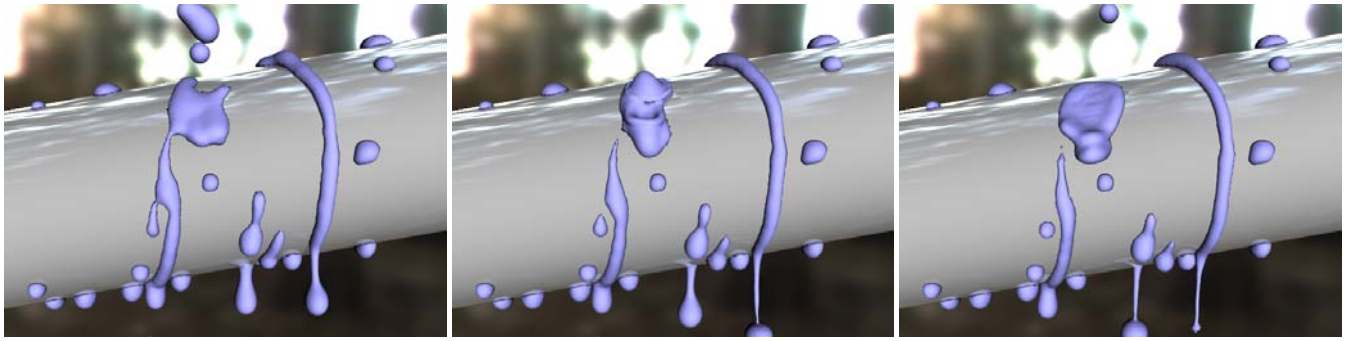


Figure 11: Water drops on a pipe.

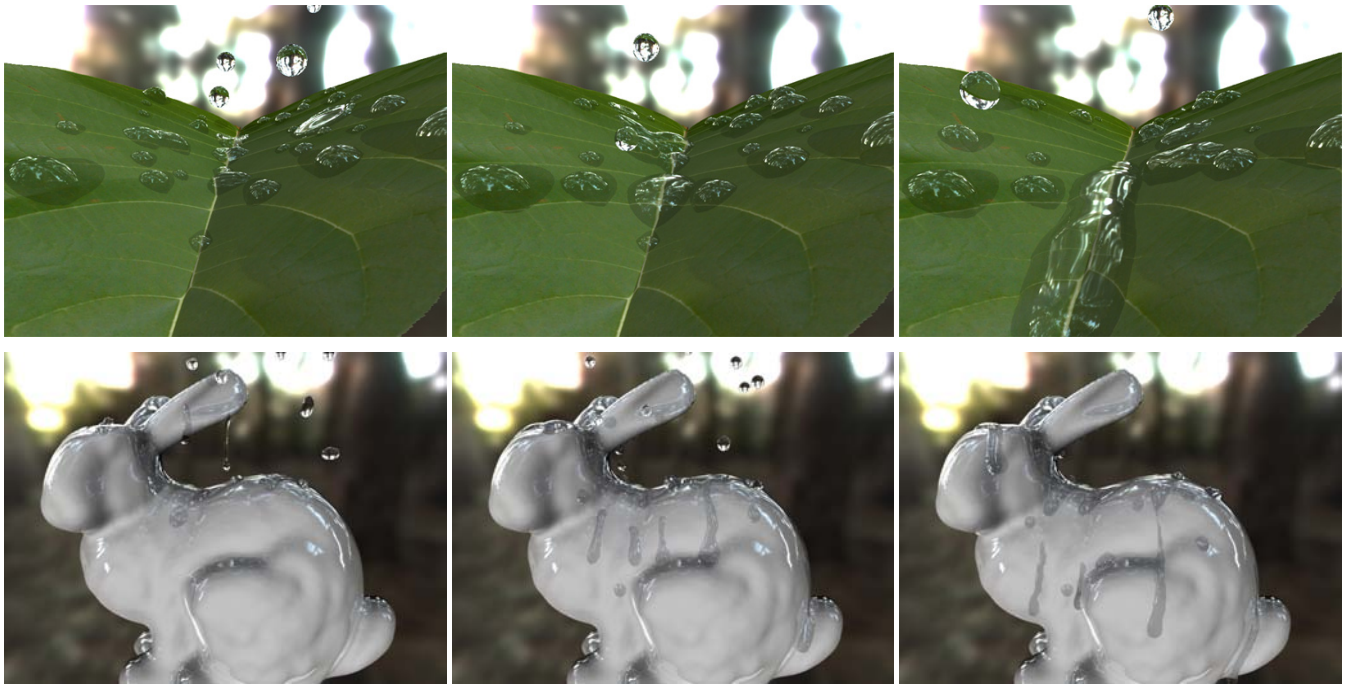


Figure 12: Drops on a leaf (top) and on a bunny (bottom). These simulations show the formation of long rivulets.

Although we use a sparse grid representation for our simulations, the memory and the computing times used for simulating small-scale fluid phenomena are still large. We plan to concentrate much of our future effort on further reducing these computational costs. Related to this is the issue of maintaining fine details of the fluid surface. Since the virtual surface method depends on sampling the signed distance function represented by the grid cells, the grid domain needs to be sufficiently refined in order to keep surface details. The octree structure [Losasso et al. 2004] might be useful for representing cells on the contact front. Another possible way to recover the surface details might be to use the particles from the particle level set method, possibly by reconstructing a point set surface.

10 Acknowledgements

We would especially like to thank Mark Carlson, Chris Wojtan and Howard Zhou for various useful suggestions, Spencer Reynolds for creating video clips and Nathan Sisterson for making sound effects in our animations. Also, we would like to thank everyone who spent time on reading early versions of this paper, including the anonymous reviewers. This research is supported in part by NSF grant DMS 0204309.

References

BUSSMANN, M., MOSTAGHIMI, J., AND CHANDRA, S. 1999. On a three-dimensional volume tracking model of droplet impact. *Phys. Fluids 11*, 1406.

DE GENNES, P. 1985. Wetting: Statics and dynamics. *Rev. Mod. Phys. 57*, 3, 827–863.

DORSEY, J., PEDERSEN, H. K., AND HANRAHAN, P. 1996. Flow and changes in appearance. In *Proc. of ACM SIGGRAPH '96*, 411–420.

ENRIGHT, D., MARSCHNER, S., AND FEDKIW, R. 2002. Animation and rendering of complex water surfaces. In *Proc. of ACM SIGGRAPH '02*, 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 *Proceeding of the 4th ASME-JSME Joint Fluids Eng. Conf., FEDSM2003-45144*, 1–6.

FENG, Z. G., DOMASZEWSKI, M., MONTAVON, G., AND CODDET, C. 2002. Finite element analysis of effect of substrate surface roughtness on liquid droplet impact and flattening process. *Journal of Thermal Spray Technology 11*, 1, 62–68.

FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. In *Proc. of ACM SIGGRAPH '01*, 23–30.

FOSTER, N., AND METAXAS, D. 1996. Realistic animation of liquids. *Graph. Models Image Process. 58*, 5, 471–483.

FOURNIER, P., HABIBI, A., AND POULIN, P. 1998. Simulating the flow of liquid droplets. In *Graphics Interface*, 133–142.

HEALY, W. M. 1999. *Modeling the Impact of a Liquid droplet on a Solid Surface*. PhD thesis, Georgia Institute of Technology.

JIANG, G.-S., AND PENG, D. 2000. Weighted eno schemes for hamilton jacobi equations. *SIAM J. Sci. Comput. 21*, 2126–2143.

KANEDA, K., KAGAWA, T., AND YAMASHITA, H. 1993. Animation of water droplets on a glass plate. In *Proc. Computer Animation '93*, 177–189.

KANEDA, K., ZUYAMA, Y., YAMASHITA, H., AND NISHITA, T. 1996. Animation of water droplet flow on curved surfaces. In *Proc. PACIFIC GRAPHICS '96*, 50–65.

KANEDA, K., IKEDA, S., AND YAMASHITA, H. 1999. Animation of water droplets moving down a surface. *The Journal of Visualization and Computer Animation 10*, 1, 15–26.

KORLIE, M. 1997. Particle modeling of liquid drop formation on a solid surface in 3-d. *Compute. and Math. with Appl. 33*, 9, 97–114.

LOSASSO, F., GIBOU, F., AND FEDKIW, R. 2004. Simulating water and smoke with an octree data structure. In *Proc. of ACM SIGGRAPH '04*, vol. 23, 457–462.

OSHER, S., AND FEDKIW, R. 2002. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag.

PHARR, M., AND HUMPHREYS, G. 2004. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann.

RENARDY, M., RENARDY, Y., AND LI, J. 2001. Numerical simulation of moving contact line problems using a volume-of-fluid method. *Journal of Computational Physics 171*, 243–263.

SCHWARTZ, L. W., AND GAROFF, S. 1985. Contact angle hyesteresis on heterogeneous surfaces. *Langmuir 1*, 2, 219–230.

SETHIAN, J. 1996. A fast marching level set method for monotonically advancing fronts. In *Proc. Natl. Acad. Sci.*, vol. 93, 1591–1595.

STAM, J. 1999. Stable fluids. In *Proc. of ACM SIGGRAPH '99*, 121–128.

SUSSMAN, M., AND UTO, S. 1998. A computational study of the spreading. of oil underneath a sheet of ice. *CAM Report 98-32, University of California, Dept. of Math, Los Angeles*.

TONG, R., KANEDA, K., AND YAMASHITA, H. 2002. A volume-preserving approach for modeling and animating water flows generated by metaballs. *The Visual Computer 18*, 8, 469–480.

TSITSIKLIS, J. 1995. Efficient algorithms for globally optimal trajectories. *IEEE Trans. on Automatic Control 40*, 1528–1538.

YU, Y.-J., JUNG, H.-Y., AND CHO, H.-G. 1999. A new water droplet model using metaball in the gravitational field. *Computer and Graphics 23*, 213–222.

ZHAO, H.-K., MERRIMAN, B., OSHER, S., AND WANG, L. 1998. Capturing the behavior of bubbles and drops using the variational level set method. *Computational Physics 143*, 495–518.

APPENDIX

CLAIM 1: Given any point P and its closest point $L(t_0)$ on $L(t)$, let $V(s_0,t_0)$ be a point on Ω_v so that $P-V(s_0,t_0)$ is perpendicular to $R(t)$. If $1+s\cos\theta_s|L''(t)|>0$ and $N_s(t)$ is constant, $V(s_0,t_0)$ is a closest point to P.

PROOF: Let $SD(t)$ and $SD(s,t)$ be squared Euclidean distance functions from P to $L(t)$ and $V(s,t)$ respectively. $L(t_0)$ is the closest point to P, so: $SD'(t0) = -2[P-L(t_0)]\cdot L'(t_0) = 0$ and $SD''(t0) = -2[P-L(t_0)]\cdot L''(t_0)+2>0$. Let $k(t)$ be a scalar function $1+s\cos\theta_s|L''(t)|$, we see:

$$\partial V(s,t)/\partial t = L'(t) - s\cos\theta_s(N_s\times L''(t)) = k(t)L'(t)$$

$k(t)>0$ for all t is satisfied if $L(t)$ is sufficiently smooth ($|L''(t)|$ is small) and if $|\cos\theta_s|$ is small enough. Geometrically, this means the virtual surface has no self-intersections so that $V(s_0,t_0)$ and $L(t_0)$ share the same tangent direction $L'(t)$. For $SD(s,t)$, we get:

$$\begin{aligned}
\frac{\partial SD(s_0,t_0)}{\partial s} &= -2[P-V(s_0,t_0)]\cdot R(t_0) &&= 0\\
\frac{\partial SD(s_0,t_0)}{\partial t} &= -2k(t_0)[P-V(s_0,t_0)]\cdot L'(t_0) &&= 0\\
\frac{\partial^2 SD(s_0,t_0)}{\partial s^2} &= 2|R(t_0)|^2 &&> 0\\
\frac{\partial^2 SD(s_0,t_0)}{\partial s\partial t} &= 2k(t_0)R(t_0)\cdot L'(t_0) &&= 0
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 SD(s_0,t_0)}{\partial t^2} &= -2[k(t_0)(P-V(s_0,t_0))\cdot L'(t_0)]'\\
&= -2k(t_0)[(P-V(s_0,t_0))\cdot L'(t_0)]'\\
&= k(t_0)(SD''(t_0) - 2s\sin\theta_s N_s\cdot L''(t_0)\\
&\quad -2s\cos\theta_s((N_s\times L'(t_0))\cdot L''(t_0) - |L''(t_0)|))\\
&= k(t_0)SD''(t_0) > 0
\end{aligned}$$

since $(N_s\times L'(t))\cdot L''(t) = |L''(t)|$. So $V(s_0,t_0)$ is a local minimum of $SD(s,t)$.

CLAIM 2: If $L(t_0)$ is the closest contact point to $C_{x,0,z}$, then $L(t_0)$ is also the closest contact point to $C_{x,-1,z}$.

PROOF: The node $C_{x,-1,z}$'s position is: $C_{x,-1,z} = C_{x,0,z} - hN_s$, where N_s is the Y direction in this case. Let $SD_0(t)$ and $SD_{-1}(t)$ be the squared distances to two nodes respectively, $SD_{-1}(t)$ satisfies:

$$\begin{aligned}
SD'_{-1}(t) &= SD'_0(t) + hN_s\cdot L'(t_0) = 0\\
SD''_{-1}(t) &= -2(C_{x,-1,z} - L(t_0))\cdot L''(t_0) + 2(L'(t_0))\cdot L'(t_0)\\
&= SD''_0(t_0) + 2hN_s\cdot L''(t_0) = SD''_0(t_0) > 0
\end{aligned}$$

Therefore, $L(t_0)$ is also the closest contact point to $C_{x,-1,z}$.