Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2006)
M.-P. Cani, J. O'Brien (Editors)

Simulation of Bubbles

Wen Zheng[1, 2], Jun-Hai Yong[1], and Jean-Claude Paul[1, 3]

[1] School of Software, Tsinghua University, China
[2] Department of Computer Science and Technology, Tsinghua University, China
[3] CNRS, France

Abstract

We present a novel framework based on a continuous fluid simulator for general simulation of realistic bubbles, with which we can handle as many significant dynamic bubble effects as possible. To capture nature of the very thin liquid film of bubbles, we have developed a regional level set method allowing multi-manifold interface tracking. The regional level set method is based on the definitions of regional distance and its five operators, which makes it very easy to implement. We can reconstruct an implicit surface of liquid film with arbitrary thickness from the representation of regional level set functions. To overcome the numerical instability caused by surface tension, we exploit a new semi-implicit surface tension model which is unconditionally stable and makes the simulation of surface tension dominated phenomena much more efficient. An approximated film thickness evolution model is proposed to control the bubble's lifecycle. All these new techniques combine into a general framework that can produce various realistic dynamic effects of bubbles.

Categories and Subject Descriptions (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically Based Modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.6.8 [Simulation and Modeling]: Type of Simulation—Animation

1. Introduction

Bubble cluster has a complex but characteristic structure, which can change both in geometry and in topology due to the surrounding flow and film rupture. This results in a very interesting and impressive visual impact, and also is a great challenge for computer graphics to model the structure and dynamics of bubbles. Many researchers in computer graphics carried out studies to achieve a realistic dynamic bubble effect, but a general model that can handle the full effect of dynamic bubbles is not yet available.

The purpose of this paper is to propose a novel method that is general enough to include as many dynamic effects of realistic bubbles as possible. Therefore, we must understand fully the physics of bubbles: surface tension dominated mixture of liquid and gas, which suggests that we can use a grid-based fluid simulator to handle the bubble dynamics. However, a typical grid-based fluid simulator used in computer graphics has two major problems when extended to the simulation of bubbles. The first is that the thickness of liquid film is too small to be captured. The magnitude of the thinnest film could be only several nanometers. The second is that, by using the current explicit surface tension model, the large surface tension may result in tremendously high computation costs. Otherwise, without sufficient computation, the numerical oscillation problem will occur. In our novel method, we have successfully explored a new surface tracking technique called the Regional Level Set method and a new semi-implicit surface tension model to solve the two major problems respectively. We have also developed algorithms to manage the creation and merging of bubbles and algorithms to reconstruct the liquid surface from the Regional Level Set function for rendering. All these techniques are combined into a general framework that can produce a large variety of highly realistic simulations and animations of bubbles.

We focus on the geometry and dynamics in this paper, so that it is distinguished from the research that focuses on the optical effect and interference phenomenon.

The paper is organized as follows. Section 2 describes

the related work both on simulation of bubbles and fluid simulation in computer graphics. Section 3 describes our new methods for simulation of bubbles. Several results produced by our method are explained in Section 4. We conclude our work and give some ideas for future work in section 5.

2. Related Work

2.1. Fluid Simulator

Foster and Metaxes [FM96] first introduced the full Navier-Stokes equation into the computer graphics field to simulate complex liquid motion. Stam [Sta99] significantly improve the computing efficiency and stability by introducing a semi-Lagrangian scheme to solve the advection term of the Navier-Stokes equation allowing for a large timestep. Foster and Fedkiw [FF01] further improved the efficiency by using a conjugate gradient method to solve impressibility, and proposed a hybrid method combining marker particles and level set method to improve both the smoothness and accuracy of surface tracking. Enright et. al. [EMF02] enhanced this hybrid surface representation by using particles on both sides of surface to maintain some more details. These previous studies build up a framework of fluid simulator for computer simulation and animation.

Researchers have also developed some variations on this fluid simulation framework for different specific purposes. Goktekin et. al. [GBO04] proposed a viscoelastic fluid model by including additional elastic terms in Navier-Stokes equations. Losasso [LGF04] proposed an octree-based technique to improve the simulation of small-scale fluid phenomena. Song et. al. [SSK05] proposed a constrained interpolation profile (CIP)-based advection method to reduce numerical dissipation and diffusion. They utilized a continuous fluid model in their work. Hong and Kim [HK05] proposed a discontinuous fluid model to handle small-scale details of multi-phase fluids.

2.2. Bubbles

In the computer graphics field, several studies have been carried out to simulate the dynamic effect of bubbles. Roman [Rom01] used particles connected by springs to simulate single bubble formation, deformation, collision with a planar surface and the collision of two bubbles. His method cannot handle the complicated situation of clustered bubbles due to the nontrivial reconnecting problem caused by topology changes, and he did not deal with the interaction between a bubble and its surrounding fluid. Kück et al. [KVG02] simulated dynamic foams by simplifying bubbles into spherical particles, so they avoided the topology changing problem, and focused on the interaction between bubbles and cluster behaviors. At the same time, they also ignored the deformation of individual bubbles, which is very important for visual realism. Greenwood and House [GH04] combined Kück's method with a fluid simulator to enable the interaction between a bubble and its surrounding fluid in order to enhance the visual effect of fluid animation. However, bubbles are still unable to deform using their method.

Hong and Kim [HK03] have used a fluid simulator to attain the dynamic effect of bubbles in liquid. They used VOF (Volume of Fluid) method to handle the fluid dynamics and a continuous body force surface tension model to handle the surface tension effect. This method can be further improved by the discontinuous fluid model they developed in [HK05]. However, this method cannot handle clustered bubbles where thin film exists. Although they utilized an octree-based technique proposed by Losasso [LGF04] to achieve a higher resolution, the thickness of liquid film rapidly reduces to a magnitude smaller than the smallest grid size and unnaturally disappears. Additionally, the explicit surface tension model they used slows down the whole simulator enormously.

2.3. Our Contributions

To overcome some limitations of these previous studies on simulation of bubbles, our work aims at a dynamic bubbles model that is general enough to handle as many interesting bubble effects as possible, including formation and deformation of individual bubbles, film rupture and merging of bubbles, interaction between adjacent bubbles, interaction between bubble and solid surface, and interaction between a bubble and its surrounding fluid.

We used a continuous multiphase fluid simulator to handle the bubble dynamics, and some new techniques have been developed to make the fluid simulator more suitable for bubbles simulation:

- Regional Level Set Method: an extension of the traditional level set method, it is able to track multi-manifold surface, and allows us to naturally represent thin film with arbitrary thickness. Based on the five operators we defined, this method is very easy to implement.
- Semi-Implicit Surface Tension Model: compared with the explicit surface tension model commonly used before, it achieves an unconditional stability and then greatly improves the efficiency of the whole simulator.

3. Simulation of Bubbles

3.1. Overview of Fluid Simulator

Our method is mainly based on the Navier-Stokes fluid

simulator. The Navier-Stokes equations for incompressible viscous fluid are

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla\cdot(\nu\nabla\mathbf{u}) - \frac{\nabla p}{\rho} + \mathbf{f}\,, \tag{1}$$

and

$$\nabla\cdot\mathbf{u} = 0\,, \tag{2}$$

where $\mathbf{u}$ is the velocity, p is the pressure, ν is the viscosity, ρ is the density, and $\mathbf{f}$ is the external force, such as gravity, buoyancy, surface tension, and other user-defined forces.

Besides the velocity field and pressure field, we also need to represent and evolve the interface between liquid and gas. A typical fluid simulator usually use a signed distance function ϕ to represent the interface, that is $\phi > 0$ if in gas region, $\phi < 0$ if in liquid region, $\phi = 0$ if on the interface, and the magnitude of ϕ equals to the distance to the interface. Then we can evolve ϕ through the level set equation:

$$\frac{\partial \phi}{\partial t} + \mathbf{u}\cdot\nabla\phi = 0\,. \tag{3}$$

For simulation of bubbles, the dynamic effects from liquid and gas are both important, so we take both of the two different materials into account. In such case, the viscosity ν and the density ρ in Equation (1) are discontinuous across the interface. We choose to use a continuous formula based on the Heaviside function, in order to smooth out the discontinuity:

$$\rho(\phi) = (\rho_l - \rho_g)H(\phi) + \rho_g\,,$$

and

$$\nu(\phi) = (\nu_l - \nu_g)H(\phi) + \nu_g\,,$$

where ϕ is the signed distance to the interface, the subscript l denotes liquid, g denotes gas, and $H(\phi)$ is the Heaviside function,

$$H(\phi) = \begin{cases} 0, & \text{if } \phi < -\varepsilon, \\ \frac{1}{2} + \frac{\phi}{2\varepsilon} + \frac{1}{2\pi}\sin(\frac{\pi\phi}{\varepsilon}), & \text{if } |\phi| \le \varepsilon, \\ 1, & \text{if } \phi > \varepsilon, \end{cases} \tag{4}$$

where ε is the "thickness" of the interface. So the property jump across the interface is smoothed out within the distance ε to the interface.

The surface tension effect is a very important role in simulation of bubbles. It can be simulated by adding the surface tension force $\mathbf{f}_{st}$ into the external force term $\mathbf{f}$ in Equation (1) as a body force. The surface tension force on interface is defined as

$$\mathbf{f}_{st} = -\sigma\kappa\mathbf{n}\delta(\phi)\,, \tag{5}$$

where σ is the surface tension coefficient, κ is the mean curvature, $\mathbf{n}$ is the unit surface normal, and $\delta(\phi)$ is the Delta function which translate the force from surface force into body force. We can compute κ and $\mathbf{n}$ easily from the signed distance function ϕ,

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}\,, \tag{6}$$

and

$$\kappa = \nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}\,, \tag{7}$$

and $\delta(\phi)$ is defined as

$$\delta(\phi) = \begin{cases} \frac{1}{2\varepsilon}\left(1 + \cos\left(\frac{\phi}{\varepsilon}\pi\right)\right), & \text{if } |\phi| \le \varepsilon, \\ 0, & \text{if } |\phi| > \varepsilon. \end{cases} \tag{8}$$

We compute the numerical solution by splitting the whole system described above into two steps. The first step is to evolve the interface by solving Equation (3). The second step is to evolve the velocity field by solving Equation (1) and (2).

At the second step, the advection term, the external force term and the diffusion term in Equation (1) are firstly solved to get an intermediate velocity $\mathbf{u}^*$:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -(\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla\cdot(\nu\nabla\mathbf{u}) + \mathbf{f}\,.$$

And then we use $\mathbf{u}^*$ to compute the pressure p by solving a Poisson equation deduced from Equation (2):

$$\nabla^2 p = \frac{\rho}{\Delta t}\nabla\cdot\mathbf{u}^*\,.$$

Finally, we use the pressure to compute the velocity in the next time step:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{\nabla p}{\rho}\,.$$

The fluid simulator described above is the foundation of our bubble simulation. However there are two major problems stop us from directly extending this framework into the simulation of bubbles. One problem comes from the extremely small thickness of liquid film, and the other comes from the oscillation problem due to the large surface tension. We will describe the two problems in detail and propose our solutions to these two problems in Section 3.2 and Section 3.3 respectively.

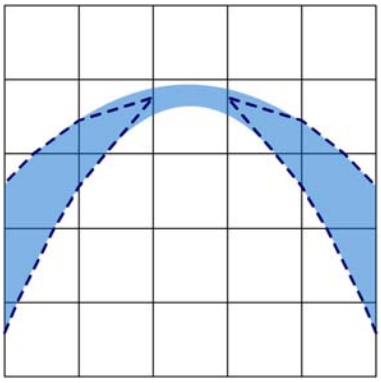

Figure 1: *Liquid film represented by level set method. The shadowed part is the actual film region, and the dashed line is the level set representation. The thinnest film is lost.*

3.2. Regional Level Set Method

Liquid film is a crucial element of bubble cluster. In most cases, liquid film is very thin, and the thickness of thinnest film can reach the magnitude of several nanometers [WH99]. Unfortunately, the traditional level set method can only represent the features whose smallest length scale is larger than the size of grid cell. When the length scale decreases to a degree that is smaller than the grid cell, the feature will automatically disappear. Thus it will be impossible to maintain the liquid film by level set method, even using the octree-based adaptive technique [LGF04]. See Figure 1 as an example.

Since the thickness of liquid film is so small that compared to the length scale of bubbles its thickness can be ignored, we assume that the liquid film thinner than the size of grid cell has an infinitesimal thickness. This assumption makes it possible that the interface will become non-two-manifold, and we have to modify the traditional level set method to allow multi-manifold interface.

Some researchers [BF95, YKI02] in geometry modeling field have developed a new idea to represent non-manifold implicit surface. Instead of using the sign of a real-valued function to represent a binary regionalization space (say liquid and gas regions in fluid simulation), they use a region code to allow multiple regionalization. We extend this idea to the representation of bubble geometry, and explore a new interface tracking technique allowing multi-manifold surface called Regional Level Set method.

First we give the definition of *regional scalar function.* It's a two-tuples spatial function. The first element is an integer-valued region code that indicates which region the given position locates in, and the second element is a real-valued scalar function. It can be expressed as

$$f_R(\mathbf{X}) = \langle R(\mathbf{X}), S(\mathbf{X}) \rangle ,$$

where $\mathbf{X}$ is the given position, $R(\mathbf{X})$ is the region code, $S(\mathbf{X})$ is the scalar function, and the angular bracket indicates an ordered set.

When we use distance to interface as the scalar function $S(\mathbf{X})$, we get a *regional distance function* ϕ_R, and then we can represent the topology of bubble cluster by dividing space into several regions: all positions immersing in liquid is defined as one region called *liquid region*; a bubble is a pocket of gas, so we define each closure of gas volume as an independent region called *bubble regions*; one of these bubble regions may have an infinite volume which means this region includes atmospheric gas, and we can rename it as *gas region*. As seen in Figure 2, the liquid film separating different bubbles becomes the interface between different bubble regions.

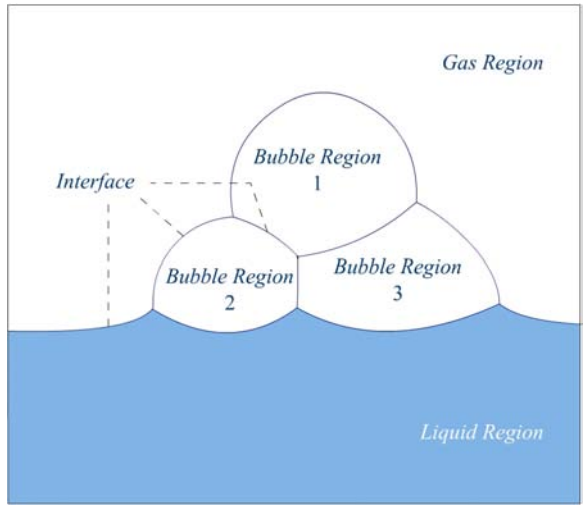

Figure 2: *Topology of bubble cluster by region code.*

Now we replace the signed distance function ϕ with the regional distance function ϕ_R to develop a new interface tracking technique. To achieve this goal, we summarized all computations concerning ϕ_R into five operators: addition, subtraction, scalar multiplication, scalar division, and multiplication of two regional scalar functions. Given $f_{R1}=<r_1, d_1>$ and $f_{R2}=<r_2, d_2>$, we redefine the five operators as follows:

Addition operator

$$\begin{aligned} & f_{R1} + f_{R2} \\ =< & r_{center}, \mathrm{sgn}(r_1, r_{center})d_1 + \mathrm{sgn}(r_2, r_{center})d_2 >, \end{aligned} \tag{9}$$

Subtraction operator

$$\begin{aligned} & f_{R1} - f_{R2} \\ =< & r_{center}, \mathrm{sgn}(r_1, r_{center})d_1 - \mathrm{sgn}(r_2, r_{center})d_2 >, \end{aligned} \tag{10}$$

Scalar multiplication operator

$$af_{R1} = f_{R1}a =< r_1, ad_1 >, \quad a \in \mathbf{R},$$

Scalar division operator

$$\frac{f_{R1}}{a} =< r_1, \frac{d_1}{a} >, \quad a \in \mathbf{R},$$

Multiplication operator

$$f_{R1} f_{R2} = f_{R2} f_{R1} = \begin{cases} d_1 d_2, & \text{if} \quad r_1 = r_2, \\ -d_1 d_2, & \text{if} \quad r_1 \neq r_2, \end{cases}$$

where the sgn(r, r_{center}) in (9) and (10) is the sign function defined as

$$\text{sgn}(r, r_{center}) = \begin{cases} -1, & \text{if} \quad r \neq r_{center}, \\ +1, & \text{if} \quad r = r_{center}. \end{cases} \tag{11}$$

It is noticeable that we have to specify the r_{center} for operators (9) (10) and sign function (11). This r_{center} is the region code of position $\mathbf{X}_{\text{center}}$ where the computation takes place. The r_{center} can be decided from a regional distance function $\phi_R(\mathbf{X})$ by tri-linear interpolation, but interpolation depends on the addition operator (9). This leads to a cyclic definition. So we particularly define the addition operator for the interpolation of regional distance function $\phi_R(\mathbf{X})$. Given ϕ_{R1}=<r_1, d_1> and ϕ_{R2}=<r_2, d_2>, we have

$$\phi_{R1} + \phi_{R2} = \begin{cases} < r_1, d_1 + d_2 >, & if \quad r_1 = r_2, \\ < r_1, d_1 - d_2 >, & if \quad r_1 \neq r_2, d_1 \geq d_2, \\ < r_2, d_2 - d_1 >, & if \quad r_1 \neq r_2, d_1 < d_2. \end{cases}$$

Based on the new-defined operators, we attain the new interface tracking technique by simply replacing all ϕ with ϕ_R without modifying any formulas formally. We call this new method the *regional level set* method.

To use regional distance function, we suppose that liquid film separating bubbles has infinitesimal thickness, but in reality the thickness does not equal to zero. Given a spatial scalar function $l(\mathbf{X})$ indicating the distribution of film thickness, we can compute a signed distance function $\phi^*(\mathbf{X})$ from $\phi_R(\mathbf{X})$ = <$r(\mathbf{X})$, $d(\mathbf{X})$> by

$$\phi^*(\mathbf{X}) = \begin{cases} d(\mathbf{X}) - \dfrac{l(\mathbf{X})}{2}, & if \quad r(\mathbf{X}) \neq r_{liquid}, \\ -d(\mathbf{X}) - \dfrac{l(\mathbf{X})}{2}, & if \quad r(\mathbf{X}) = r_{liquid}, \end{cases}$$

where r_{liquid} is the region code of liquid region. Then as seen in Figure 3, $\phi^*(\mathbf{X})$ represents a real structure of bubble cluster whose liquid film has non-zero thickness. We use ϕ^* to replace ϕ in Heaviside function (4) and Delta function (8), and we also use ϕ^* for rendering.

To resist the volume loss problem and maintain more surface details, we can easily replant the particle correction scheme proposed by Foster [FF01] and Enright [EMF02] into the regional level set method. First, we extend two sets of particles to N sets of particles, where N is the number of regions. Each set of particles corresponds to one region. We use *Pr* to denote the particles of region *r*. Then, we randomly place *Pr* inside region *r* within a specified distance to the interface. Particles will move along with the undergoing velocity by the second-order Runge-Kutta method, while evolving the implicit surface. At every time step, for every region *r*, we treat *Pr* as negative particles and all other particles as positive particles, and we use the same rule in [EMF02] to correct the interface of region *r*.

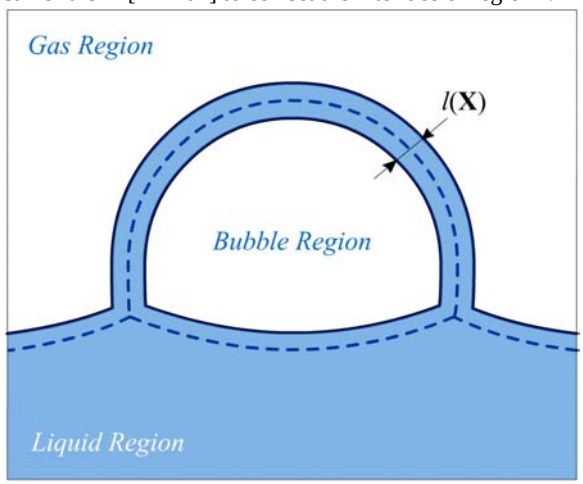

Figure 3: *The reconstructed bubble structure from ϕ^*. The dashed line is the interface of ϕ_R and the solid line is the interface of ϕ^*.*

3.3. Semi-Implicit Surface Tension

Unlike large scale phenomena, bubbles are dominated by surface tension. Thus the surface tension effect becomes a key to the realism of bubble motion. As our knowledge, all previous work concerning surface tension in compute graphics field use explicit time integration scheme. This has a drawback that it suffers from a numerical instability problem called numerical oscillation when time step is large, and we have to limit the time step size both by the grid size and by the surface tension coefficient, which potentially leads to prohibitive computation costs. Recent work concerning surface tension effect, such as the simulation of water drops [WMT05] and the simulation of small-scale fluid motion [HK05], costs extremely large amount of computation to make simulation stable. To overcome this drawback, we propose here a new semi-implicit surface tension model which is unconditionally stable.

First we give a theorem of differential geometry, which states that

$$\Delta_s \mathrm{I}_\Gamma = -\kappa \mathbf{n}, \tag{12}$$

where I_Γ is the identity mapping on interface Γ which represents the interface position, κ is the mean curvature defined by (7), and $\mathbf{n}$ is the unit normal defined by (6). Here Δ_s is the surface Laplacian defined by

$$\Delta_s = \nabla_s \cdot \nabla_s,$$

where ∇_s is the surface gradient operator defined by

$$\nabla_s = \nabla - (\mathbf{n} \cdot \nabla)\mathbf{n}.$$

Then an implicit time integration of interface position is given by

$$\mathrm{I}_\Gamma^{new} = \mathrm{I}_\Gamma + \mathbf{u}^{new} \Delta t. \tag{13}$$

We apply Δ_s on both sides of (13), and we combine it with (12) to get

$$-(\kappa \mathbf{n})^{new} = -\kappa \mathbf{n} + \Delta t \left(\Delta_s \mathbf{u}^{new} \right). \tag{14}$$

Then we combine (14) with Formula (5), and we get the semi-implicit version of surface tension force:

$$\mathbf{f}_{st} = \frac{\mathbf{u}^{new} - \mathbf{u}}{\Delta t} = -\sigma \delta(\phi) \kappa \mathbf{n} + \sigma \delta(\phi) \Delta t \left(\Delta_s \mathbf{u}^{new} \right). \tag{15}$$

Compared to (5), Formula (15) has an additional term, which introduces a diffusion procedure in the tangential direction of surface, which will propagate the surface tension force along surface. The coefficient σ implies that larger surface tension will generate more surface diffusion and promise the system to be stable. The coefficient Δt implies that the diffusion will be stronger if the surface tension force exerts longer, and when Δt approaches 0, Equation (15) will approach (5).

Equation (15) leads to a linear system, and we can directly solve it by a linear system solver. But we can further simplify the computation by decomposing the surface Laplacian into a standard Laplacian:

$$\Delta_s = \nabla_s^2 = \nabla^2 - \frac{\partial^2}{\partial \mathbf{n}^2} - \kappa \frac{\partial}{\partial \mathbf{n}},$$

where $\frac{\partial}{\partial \mathbf{n}} = \mathbf{n} \cdot \nabla$ and $\frac{\partial^2}{\partial \mathbf{n}^2} = \mathbf{n} \cdot D^2 \cdot \mathbf{n}$. Here D^2 is a Hessian matrix.

And then we can translate (15) into

$$\begin{aligned} \mathbf{f}_{st} &= \frac{\mathbf{u}^{new} - \mathbf{u}}{\Delta t} \\ &= -\sigma \delta(\phi) \left(\kappa \mathbf{n} + \Delta t \left(\frac{\partial^2 \mathbf{u}}{\partial \mathbf{n}^2} + \kappa \frac{\partial \mathbf{u}}{\partial \mathbf{n}} \right) \right) \\ &\quad + \sigma \delta(\phi) \Delta t \left(\nabla^2 \mathbf{u}^{new} \right), \end{aligned} \tag{16}$$

where we use an implicit scheme only on the standard Laplacian and use the explicit scheme on all other terms. The work in [XZ03] shows that this scheme is unconditionally stable. In such a way, Equation (16) becomes a Poisson equation that we can solve efficiently by the Preconditioned Conjugate Gradient Method.

When bubble touches solid surface, its motion will be affected by the wall adhesion force. It can be directly computed at grid cells adjacent to solid wall. An alternative way is the virtual surface method proposed by Wang [WMT05]. The mean curvature and unit normal computed from the virtual surface then can be used in the surface tension force formula (5) to include the wall adhesion effect.

3.4. Lifecycle of Bubbles

By using the Regional Level Set method, we can naturally represent the geometry and topology of bubbles. However, as we use different region codes for different pockets of gas, the emerging of new bubbles and the merging of different bubbles are not automatically included in the Regional Level Set method. This returns the controllability of the lifecycle of bubbles to users, and users can develop their own control methods for various specific purposes. Here we provide a simple control method that approximates the lifecycle of bubbles in the real world.

Creation of Bubbles:
New bubbles are created for both Chemical and Physical reasons. For bubbles transited from other materials, such as beer bubbles and boiling water, we directly insert the bubble with a new region code. For bubbles split from existing bubbles, the only evidence of bubble formation is the new generated close pocket of gas. Greenwood and House [GH04] has proposed a modified Flood Fill algorithm to detect gas pockets. We adopt their method to mark every close region of gas. If two different pockets with the same region code are detected, which means they are new bubbles split from the same old bubble, we treat one of the two bubbles as a new one, and change the region code of all cells contained in the new bubble into a new region code.

Merging of Bubbles:

Bubbles persist until liquid film rupture. If liquid film separating two bubbles ruptures, the two will merge into

one. If liquid film separating a bubble from atmosphere gas ruptures, the bubble will merge into the atmosphere gas and break up. In both cases, we only need to change the region codes of cells contained in the two regions separated by the rupturing liquid film into the same region code.

Compared to the problem of how to merge, the problem of when to merge is more complicated. Liquid film ruptures when its thickness is too thin to resist the surface tension force, and there are several reasons for the changing of thickness. We consider only two major reasons in our control method: the first is drainage which means that liquid flows down along the film; the second is the deformation of the film which means that the stretch of film causes thickness decrease and the shrinking of film causes thickness increase.

For simplicity, we assume that each bubble has a unified thickness named *region thickness*, and we assume that the drainage procedure linearly depends on time. Consequently we get an approximated drainage equation:

$$\frac{dl_r}{dt} = \frac{l_{init} - l_{min}}{t_{persist}}, \tag{17}$$

where l_r is the *region thickness* of bubble r, l_{init} is the initial thickness, l_{min} is the minimum thickness where bubble will break, and $t_{persist}$ is the user-specified time length that a bubble can persist for. The drainage starts only when bubble touches another bubble or atmosphere gas. The *region thickness* is set to be the initial thickness l_{init} once bubble is surrounded by liquid. And the *region thickness* of atmosphere gas is set to be the minimum thickness l_{min}.

We also assume that the thickness change due to film deformation is unified for each bubble. Then we get an approximated deformation equation:

$$dl_r = \frac{l_r}{A_r} dA_r, \tag{18}$$

where A_r is the total surface area of bubble r, which can be computed from the regional distance function $\phi_R(\mathbf{X}) = \langle d(\mathbf{X}), r(\mathbf{X}) \rangle$ with the Delta function by

$$A_r = \sum \delta\left(\phi_r^*(\mathbf{X})\right) \left|\nabla \phi_r^*(\mathbf{X})\right| \Delta x^3, \text{ and}$$
$$\phi_r^*(\mathbf{X}) = \begin{cases} d(\mathbf{X}), & \text{if } r(\mathbf{X}) \neq r, \\ -d(\mathbf{X}), & \text{if } r(\mathbf{X}) = r. \end{cases}$$

Finally, we combine (17) and (18) to get the equation governing the evolution of *region thickness* of bubble r:

$$\frac{dl_r}{dt} = \frac{l_{init} - l_{min}}{t_{persist}} + \frac{l_r}{A_r}\frac{dA_r}{dt}. \tag{19}$$

At every time step, we update *region thickness* for every individual bubble through (19), and we make adjacent regions, whose *film thickness* is less than the minimum thickness l_{min}, merge into one. The *film thickness* between two adjacent regions is approximated by averaging the *region thickness* of the two regions or taking the maximum *region thickness* of the two.

4. Results

We have implemented our new method for simulation of bubbles and used it to produce some example animations to demonstrate various bubble effects.

Figure 4 shows a pan of boiling water, in which we can see the whole lifecycle of bubbles. The left sequence is the top view, and the right sequence is the side view without rendering pan in order to show more details inside water. Bubbles are first generated at the bottom of the pan because of vaporization. Due to buoyancy, they are detached from the bottom and rise up until touching water surface. Then bubbles stable down and form a flattened spherical shape. During rising and floating, the attraction and clustering behavior of bubbles can be detected. After exposed to the atmosphere air for a certain time, the film of bubbles thins down and finally ruptures, which leads to the break up of bubbles. To focus on bubbles themselves, we approximated the bubble generation by randomly changing a volume of water into gas at the bottom. We use a very large surface tension coefficient to make bubbles quickly achieve equilibrium. Such a large surface tension can be stably solved by our semi-implicit surface tension model with a large time step.

Figure 5 shows a snapshot sequence of blowing bubbles, in which we can see many different kinds of dynamic effects of bubbles. The formation of bubbles is carried out by blowing liquid film away from a cylinder straw on the left. When the bubble grows large enough, it leaves the straw and deforms into a spherical shape. Bubbles interact with each other when they collide, and gradually form an equilibrium clustered structure. Due to gravity, bubbles fall onto the ground, and they interact with the solid surface. Bubbles also interact with the solid wall on the right when they touch it. After a certain time, different bubbles begin to merge. In this example, almost every kind of dynamic bubble effect is included. We used an inflow boundary condition to generate the blowing wind force. A solid wall is placed on the right side to stop bubbles being blown out of the computation domain. For demonstration's sake, we stopped bubbles merging into atmosphere gas.

We performed both examples on a PC with 3 GHz CPU and 3 GB RAM. To overcome the volume loss problem, we doubled the grid resolution for surface tracking. This technique was reported by [GBO04]. The speed of our simulation method is approximately the same with the

original fluid simulator we based on. The first example of Figure 4 on a 40×20×40 grid costs about 20 seconds per frame for simulation. And the second example of Figure 5 on a 30×12×15 grid costs about 2 seconds per frame for simulation. Benefit from the unconditional stability of our semi-implicit surface tension model, we took only one timestep for each frame, which dramatically reduces the computation cost. And due to the regional level set method we used, we can handle thin film with arbitrary thickness even on a very coarse grid.

We used the Marching Cubes method [LC87] to extract a triangle mesh from the implicit surface, and rendered it by an open source renderer PIXIE [Ari05]. Because the focus of this paper is not on the rendering, we used a glass shader for both examples. For the first example, we reconstruct a signed-distance function from the regional level set representation with a constant film thickness. For the second example, we ignored the thickness and rendered the liquid film as single layer interface, so the interference color was not taken into account. Some rendered images suffer from insufficient smoothness at places where surface curvature is high. This can be ameliorated by adopting an adaptive surface extraction method or increasing the grid resolution.

5. Conclusions

We propose a new general framework based on a fluid simulator for realistic simulation and animation of bubbles, through which we can effectively handle formation and deformation of bubbles, film rupture and merging of bubbles, interaction between adjacent bubbles, interaction between bubbles and a solid surface, and interaction between bubbles and their surrounding fluid. To capture the very thin liquid film, we develop a regional level set method by defining a regional distance function and its five operators. With these definitions, we can easily extend the traditional level set method to a new version that allows multi-manifold interface tracking by simply replacing the signed distance function with the regional distance function. Note that the regional level set method is suitable for not only simulation of bubbles but also all other multi-manifold surface tracking problems, such as interaction of multiple liquids. To overcome the numerical oscillation problem arose by surface tension, we exploit a novel semi-implicit surface tension model which is unconditionally stable, and can easily be solved by the Preconditioned Conjugate Gradient method, so that it dramatically speeds up the simulation. We also provide an approximated control scheme for the lifecycle of bubbles to handle bubble creation, bubble splitting, film thickness decrease, film rupture, and merging of bubbles. Using the proposed methods, we produce several results that demonstrate the ability of our framework to make realistic animations of bubbles. Although in the examples presented the number of bubbles is limited, we believe that our method is also capable of simulating a complex foam structure containing a large number of bubbles with various sizes.

Because of the continuous fluid model we adopted, the interface is “thickened” into several grid sizes and the discontinuity of physical properties is smoothed out. We can improve the reality by adopting a discontinuous fluid model, such as proposed by [HK05], to attain sub-grid accuracy. In the control scheme for bubble’s lifecycle, we used an approximated thickness model, within which the thickness of one bubble is unified. This can be further improved by utilizing a more physics-based thickness evolution model. For example, we can use a thickness distribution function extrapolated from interface to space and solving the thickness advection and thickness diffusion equations along the interface by methods proposed by [XZ03]. We will do these improvements in future works.

6. Acknowledgements

The research was supported by Chinese 973 Program (2002CB312106), and the National Science Foundation of China (60403047, 60533070). The second author was supported by the project sponsored by a Foundation for the Author of National Excellent Doctoral Dissertation of PR China (200342), and a Program for New Century Excellent Talents in University (NCET-04-0088). We would like to thank INRIA (France).

References

[Ari05] ARIKAN O.: Pixie: Photorealistic renderer (2005). http://sourceforge.net/projects/pixie.

[BF95] BLOOMENTHAL J., FERGUSON K.: Polygonization of non-manifold implicit surfaces. In *Proc. SIGGRAPH ’95* (1995), pp. 309-316.

[EMF02] ENRIGHT D., MARSCHNER S., FEDKIW R.: Animation and rendering of complex water surfaces. *ACM Trans. on Graphics (Proc. SIGGRAPH ’ 02) 21*, 3 (2002), 736–744.

[FF01] FOSTER N., FEDKIW R.: Practical animation of liquids. In *Proc. SIGGRAPH ’01* (2001), pp. 23–30.

[FM96] FOSTER N., METAXES D.: Realistic animation of liquids. *Graphical Models and Image Processing 58* (1996), pp. 471-483.

[GBO04] GOKTEKIN T. G., BARGTEIL A. W., O’BRIEN J. F.: A method for animating viscoelastic fluids. In *Proceedings of ACM SIGGRAPH ‘04* (2004), pp.463–468.

[GH04] GREENWOOD S., HOUSE D.: Better with bubbles:

Enhancing the visual realism of simulated fluid. In *Proc. 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aug. 2004), pp. 287-296.

[HK03] HONG J., KIM C.: Animation of bubbles in liquid. In *Proc. Eurographics '03* (2003), pp. 253-262.

[HK05] HONG J., KIM C.: Discontinuous fluids. In *Proc.* SIGGRAPH *'05* (2005), pp.915-920.

[Hys05] HYSING S.: A new implicit surface tension implementation for interfacial flows. *International Journal for Numerical Methods in Fluids 2005*, (in press).

[KVG02] KÜCK H., VOGELGSANG C., GREINER G.: Simulation and rendering of liquid foams. In *Proc. Graphics Interface '02* (2002), pp. 81–88.

[LC87] LORENSEN W. E., CLINE H. E.: Marching Cubes: A high resolution 3D surface construction algorithm. In *International Conference on Computer Graph-ics and Interactive Techniques* (1987), pp. 163-169.

[LGF04] LOSASSO F., GIBOU F., FEDKIW R.: Simulating water and smoke with an octree data structure. In *Proc. SIGGRAPH '04* (2004), pp. 457-462.

[Rom01] ROMAN D.: Animation of soap bubble dynamics, cluster formation and collision. *Computer* Graphics *Forum 20*, 3 (Sept. 2001), C67-C75. (Proc. Eurographics'2001).

[SSK05] SONG O., SHIN H., KO H.: Stable but Non-Dissipative Water. *ACM Transactions on Graphics 11*, 2 (2005), 1-16.

[Sta99] STAM J.: Stable fluids. In *Proc. SIGGRAPH* '99 (1999), pp. 121-128.

[WH99] WEAIRE D., HUTZLER S.: *The physics of foams*. Oxford, 1999.

[WMT05] WANG H., MUCHA P.J., TURK G.: Water Drops on Surfaces. In *Proc. SIGGRAPH '05* (2005), pp. 921-929.

[XZ03] XU J., ZHAO H.: An Eulerian Formulation for Solving Partial Differential Equations along a Moving Interface. *Journal of Scientific Computing 19*, n 1-3 (Dec. 2003), 573-594.

[YKI02] YAMAZAKI S., KASE K., IKEUCHI K.: Non-manifold Implicit Surfaces Based on Discontinuous Implicitization and Polygonization. In *Proc. Geometric Modeling and Processing 2002* (July 2002), pp. 138-146.

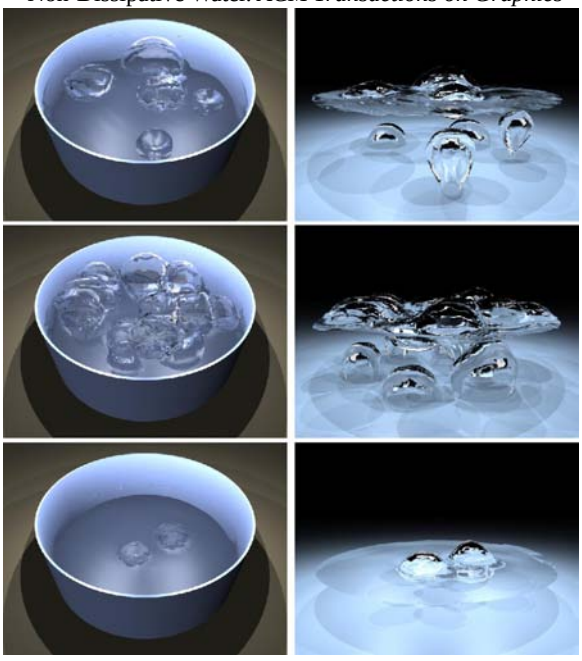

Figure 4: *Boiling water.*

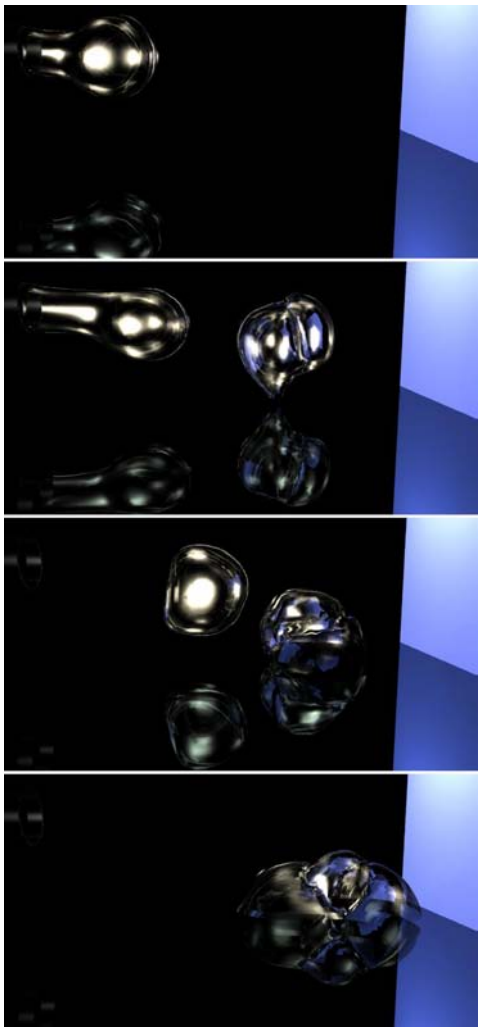

Figure 5: *Blowing bubbles.*