Simulation of bubbles

Wen Zheng, Jun-Hai yong, Jean-Claude Paul

Simulation of bubbles

Wen Zheng [a,b,*], Jun-Hai Yong [a], Jean-Claude Paul [a,c]

[a] School of Software, Tsinghua University, Beijing, China
[b] Department of Computer Science and Technology, Tsinghua University, Beijing, China
[c] CNRS, France

ARTICLE INFO

Article history:
Received 23 February 2007
Received in revised form 26 June 2009
Accepted 23 August 2009
Available online 27 August 2009

Keywords:
Bubble simulation
Fluid simulation
Physically-based modeling
Navier–Stokes equations
Regional level set method
Semi-implicit surface tension

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 a very thin liquid film of bubbles, we have developed a regional level set method allowing multi-manifold interface tracking. Based on the definitions of regional distance and its five operators, the implementation of the regional level set method is very easy. An implicit surface of liquid film with arbitrary thickness can be reconstructed from the regional level set function. To overcome the numerical instability problem, 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.

1. Introduction

Bubble clusters have a complex but characteristic structure, which can change both in geometry and in topology due to the surrounding flow and film rupture. This change in the structure results in a very interesting and impressive visual impact, but also presents a great challenge for computer graphics to model the bubble structures and handle their dynamics. Although much research has been carried out to achieve a realistic dynamic bubble effect, the resulting models are either too simplified or too limited. A more 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 the bubbles: a surface tension-dominated mixture of liquid and gas, which suggests that a grid-based fluid simulator can be used 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 efficiently captured since the magnitude of the thinnest film can be only several nanometers. The second is that, by using the explicit surface tension model, the large surface tension may result in tremendously high computation costs, without which a 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 bubble structure 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.

* Corresponding author. Address: School of Software, Tsinghua University, Beijing, China. Fax: +86 10 62795460.
E-mail addresses: zhen-w@mails.tsinghua.edu.cn, moskuito.zw@gmail.com (W. Zheng), yongjh@mail.tsinghua.edu.cn (J.-H. Yong), paul@tsinghua.edu.cn (J.-C. Paul).

In this paper, we focus on the geometry and dynamics to distinguish our work from the research that focuses on optical effects and the interference colors 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 gives an overview of the fluid simulator our work is based on. Section 4 proposes a new interface tracking method, Regional Level Set method, which allows multi-manifold surfaces, to represent our approximated bubble structure. Section 5 proposes a new semi-implicit surface tension model which is much more stable. Section 6 presents a simple control scheme of bubble's lifecycle. Several animation results produced by our method are explained in Section 7, while Section 8 offers a brief survey and discussion of recent papers related to the simulation of bubbles after our paper was published. We conclude our work and give some ideas for future work in Section 9.

2. Previous work

2.1. Fluid simulator

Foster and Metaxes [1] first introduced the full Navier–Stokes equation into the computer graphics field to simulate complex liquid motion. Stam [2] 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 time step. Foster and Fedkiw [3] further improved the efficiency by using a conjugate gradient method to solve incompressibility, 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. [4] enhanced this hybrid surface representation by using particles on both sides of surface to maintain some more details. These previous studies have contributed to a framework of fluid simulators for computer simulation and animation.

Some variations on this fluid simulation framework have also been developed for different specific purposes. Goktekin et al. [5] proposed a viscoelastic fluid model by including additional elastic terms in Navier–Stokes equations. Losasso [6] proposed an octree-based technique to improve the simulation of small-scale fluid phenomena. Song et al. [7] proposed a constrained interpolation profile (CIP)-based advection method to reduce numerical dissipation and diffusion, using a continuous fluid model in their work. Hong and Kim [8] 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 [9] used particles connected by springs to simulate single bubble formation, deformation, collision with a planar surface and the collision of two bubbles; however, his method cannot handle the complicated situation of clustered bubbles due to the nontrivial reconnecting problem caused by topology changes, nor did he deal with the interaction between a bubble and its surrounding fluid. Kück et al. [10] simulated dynamic foams by simplifying bubbles into spherical particles, thereby avoiding the topology changing problem, and focused on the interaction between bubbles and cluster behaviors. At the same time, however, they also ignored the deformation of individual bubbles, which is very important for visual realism. Greenwood and House [11] 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. Despite this improvement, bubbles are still unable to deform using their method.

Hong and Kim [12] 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 [8]. However, this method cannot handle clustered bubbles where thin film exists. Although they utilized Losasso's octree-based technique [6] to achieve a higher resolution, the thickness of liquid films rapidly reduces to a magnitude smaller than the smallest grid size and unnaturally disappears. Additionally, their explicit surface tension model greatly slows down the whole simulator.

2.3. Our contribution

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 multi-phase 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 five self-defined operators, 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, thereby greatly improving the efficiency of the entire simulator.

3. Overview of fluid simulator

Our method is mainly based on the Navier–Stokes fluid simulator. The Navier–Stokes equation for incompressible viscous fluid is

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

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

where $\mathbf{u}$ is the velocity, p is the pressure, v 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 uses the signed-distance function φ to represent the interface, that is $\varphi > 0$ if in gas region, $\varphi < 0$ if in liquid region, $\varphi = 0$ if on the interface, and the magnitude of φ equals to the distance to the interface. We can then evolve φ through level set equation:

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

For bubbles simulation, the dynamic effects from liquid and gas are both important; thus we take both of the two different materials into account. In such a case, viscosity v and density ρ in Eq. (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:

$$\rho(\phi) = (\rho_l - \rho_g)H(\phi) + \rho_g$$
$$v(\phi) = (v_l - v_g)H(\phi) + v_g$$

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

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

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

The surface tension effect plays a very important role in bubbles simulation. It can be simulated by adding the surface tension force $\mathbf{f}_{st}$ into the external force term $\mathbf{f}$ in Eq. (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 $\delta(\varphi)$ is the Delta function which translates a surface force into a body force. We can compute κ and $\mathbf{n}$ easily from the signed-distance function φ,

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

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

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

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

We compute the numerical solution by splitting the whole system described above into several steps:

1. Evolve the interface by solving Eq. (3).
2. Evolve the velocity field by solving Eqs. (1) and (2).
   2.1. Solve the advection term, the external force term, and the diffusion term in Eq. (1), and obtain an intermediate velocity $\mathbf{u}^*$:

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

   2.2. Use $\mathbf{u}^*$ and the Preconditioned Conjugate Gradient method to compute pressure p by solving a Poisson equation deduced from Eq. (2):

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

Then 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 bubbles simulator. However two major problems stop us from directly extending this framework into bubbles simulation. One problem comes from the extremely small thickness of liquid film, while the other comes from the oscillation problem due to large surface tension. We will describe the two problems in detail below; our solutions to these problems will be proposed in Sections 4 and 5 respectively.

4. Regional level set method

Liquid film is a crucial element of bubble clusters. A liquid film is usually very thin, with the thickness of the thinnest film even reaching the magnitude of several nanometers [13]. Unfortunately, the traditional level set method can only represent the features whose length scale is larger than the size of a grid cell. As seen in Fig. 1, when the length scale

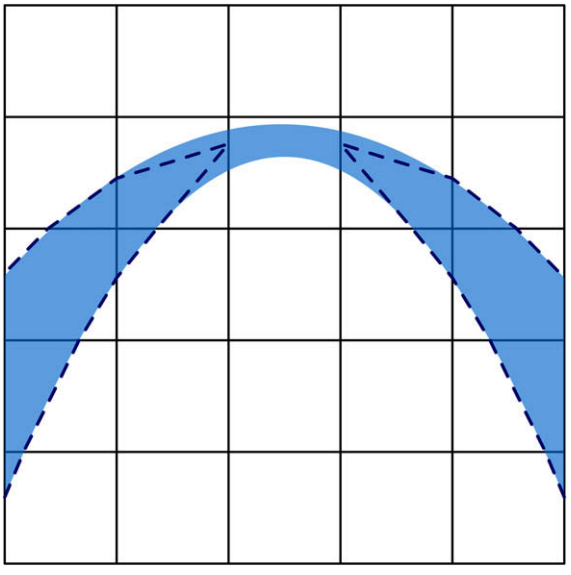

Fig. 1. Liquid film represented by level set method. The shadowed part is the actual film region, and the dashed line is the level set representation. The thinnest film is lost.

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 the level set method, even using the octree-based adaptive technique [6].

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

4.1. Regional distance function

The traditional level set method usually utilizes the signed-distance function φ to represent the interface; also, the sign of φ is responsible for indicating different spatial regions. Since the sign of a real value has only two choices (positive and negative), the signed-distance function φ is only capable of binary regionalization (say liquid and gas regions in two-phase fluid simulation), which limits the traditional level set method within two-manifold interfaces.

Some researchers [14,15] in the geometry modeling field have proposed a new idea to break the limit of the signed-distance function. Instead of using the sign of a real-valued function for binary regionalization, they use a *region code* to allow multiple regionalization. The *region code* is an integer value which has unlimited number of choices; thus, it can indicate as many regions as needed. We extend this idea to the representation of our approximated bubble geometry which is multi-manifold interface.

Regional scalar function can be defined as a two-tuple spatial function. The first element is an integer-valued region code that indicates in which region the given position is located, while the second element is a real-valued scalar function which 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 the distance function to the interface as the scalar function $S(\mathbf{X})$, we get a multi-regional extension of the signed-distance function φ: the *regional distance function* φ_R.

Now with the regional distance function, we are able to represent the topology of the bubble structure by dividing space into several regions: all positions immersed in liquid are defined as one region called the *liquid region*; a bubble is a pocket of gas; thus, 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 can be renamed the *gas region*. As seen in Fig. 2, the liquid film separating different bubbles becomes the interface between different bubble regions.

4.2. Redefinition of basic operators

To complete our new interface tracking technique, we replace the signed-distance function φ with the regional distance function φ_R wherever necessary.

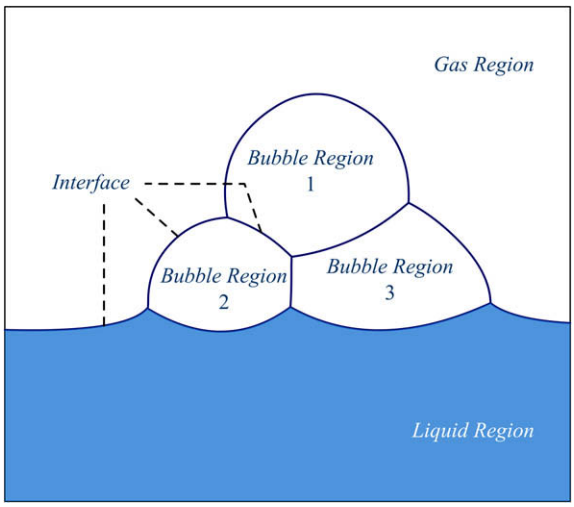

Fig. 2. Topology of bubble cluster by region code.

In numerical computation, φ_R will be stored at discrete grids, thus interpolation is necessary for attaining φ_R at arbitrary location. Furthermore, since we use the semi-Lagrangian scheme to solve the level set Eq. (3), interpolation is also indispensable. It can be seen as a weighted sum of several grid values. Take the one-dimensional linear interpolation for some kind of function q as an example: $q = (1 - x)\, q_i + x\text{-}q_{i+1}$, where q_i and q_{i+1} are defined at grid i and $i + 1$ respectively, and x is the distance from the interpolation location to grid i. As seen in the above formula, the q functions are first multiplied by a positive real scalar, and then summed up into a new q function. The situations are similar for bi-linear, bi-cubic, tri-linear and tri-cubic interpolations. Hence, we summarize the interpolation procedure into two basic operators: scalar multiplication and addition.

The gradient and the curvature information are also required by the surface tension formula (5). As seen in (6) and (7), the gradient and the curvature are both obtained by exerting the partial differential operators on φ. In numerical computation, the partial differential operators usually change into difference operators, (for instance, the central difference, $\frac{\phi_{i+1} - \phi_{i-1}}{2\Delta i}$), which are composed of subtraction of distance functions and division by a positive real scalar. Hence, we also summarize the gradient and curvature computation into two basic operators: scalar division and subtraction.

In addition, when we change the designation of which side is the inside of the interface, the gradient and the curvature will have different signs, which means the gradient and the curvature are related to a specified inside region. Thus, we use the regional scalar function to represent the two quantities. This treatment leads to another basic operator in the surface tension formula (5): multiplication of two regional scalar functions.

To sum up, we have summarized all computations concerning φ_R into five basic operators: addition, subtraction, scalar multiplication, scalar division, and multiplication of two regional scalar functions. Given $f_{R1} = \langle r_1, d_1 \rangle$ and $f_{R2} = \langle r_2, d_2 \rangle$, we can now redefine the five operators for regional scalar function as follows:

Addition operator

$$f_{R1} + f_{R2} = \langle r_{center}, \mathrm{sgn}(r_1, r_{center})d_1 + \mathrm{sgn}(r_2, r_{center})d_2 \rangle \quad (9)$$

Subtraction operator

$$f_{R1} - f_{R2} = \langle r_{center}, \mathrm{sgn}(r_1, r_{center})d_1 - \mathrm{sgn}(r_2, r_{center})d_2 \rangle \quad (10)$$

Scalar multiplication operator

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

Scalar division operator

$$\frac{f_{R1}}{a} = \left\langle r_1, \frac{d_1}{a} \right\rangle, \quad a \in \mathbf{R}$$

Multiplication operator

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

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

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

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

$$\phi_{R1} + \phi_{R2} = \begin{cases} \langle r_1, d_1 + d_2 \rangle, & \text{if } r_1 = r_2 \\ \langle r_1, d_1 - d_2 \rangle, & \text{if } r_1 \neq r_2,\ d_1 \geqslant d_2 \\ \langle r_2, d_2 - d_1 \rangle, & \text{if } r_1 \neq r_2,\ d_1 < d_2 \end{cases}$$

Based on the new-defined operators, the implementation of the new interface tracking technique becomes very easy. Without modifying any formula formally, the only job is simply replacing all ϕ with ϕ_R. We call this new interface tracking method the *Regional Level Set* method.

4.3. Reconstructing the liquid surface

To use the regional distance function, we assume that liquid films separating bubbles have infinitesimal thickness; however, in reality the thickness does not equal zero, thus we still need to recover the thickness of the thin liquid film. Given a spatial scalar function $l(\mathbf{X})$ indicating the distribution of film thickness, we can reconstruct a signed-distance function $\varphi^*(\mathbf{X})$ from $\varphi_R(\mathbf{X}) = \langle r(\mathbf{X}), d(\mathbf{X}) \rangle$ by

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

where r_{liquid} is the region code of a liquid region. Then as seen in Fig. 3, $\varphi^*(\mathbf{X})$ represents a real structure of bubble clusters whose liquid films have non-zero thickness. Since the scalar function $l(\mathbf{X})$ can be any value, the thickness of the reconstructed liquid film can be arbitrarily small, and at the same time the topology of bubble clusters can still be preserved without unexpected film rupture.

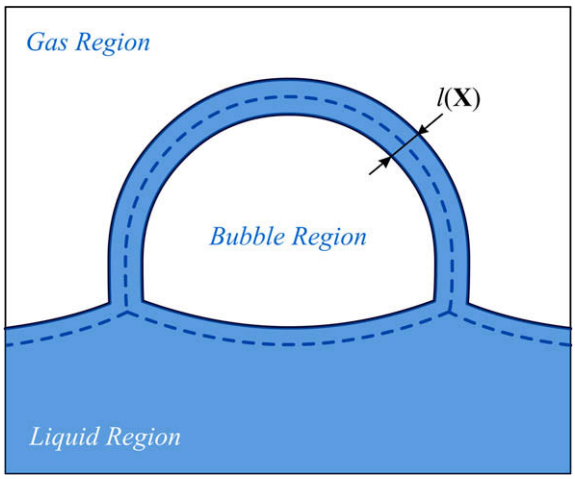

Fig. 3. The reconstructed bubble structure from ϕ^*. The dashed line is the interface of ϕ_R and the solid line is the interface of ϕ^*.

In the Heaviside function (4) and the Delta function (8) we replace ϕ by ϕ^*, because both functions requires an actual distance field. However, a thin film has insufficient thickness for the Heaviside function to work properly. Since the liquid volume of a thin film has changed into interface, the volume properties on thin film can be treated as surface properties. Thus we can use the Delta function to smooth out physical properties, such as the density and viscosity, instead of using the Heaviside function.

We also use φ^* for rendering, for both the direct ray-tracing and the extraction of triangular meshes. The implementation for the direct ray-tracing of φ^* is the same with the ray-tracing of the traditional level set function φ, we simply replace φ by φ^*. The extraction of triangular meshes is slightly different. We respectively extract the inside surface of each individual region except the liquid region. For instance, when extracting region i, we modify the formula of φ^* as follows:

$$\phi^*(\mathbf{X}) = \begin{cases} d(\mathbf{X}) - \frac{l(\mathbf{X})}{2}, & \text{if } r(\mathbf{X}) = r_i \\ -d(\mathbf{X}) - \frac{l(\mathbf{X})}{2}, & \text{if } r(\mathbf{X}) \neq r_i \end{cases}$$

where r_i is the region code of region i. This modification avoids the vanishing of the features within the same grid cell, such as the inside and outside surfaces of a thin film.

4.4. Volume preserving

To resist the volume loss problem and maintain more surface details, we can easily replant the particle correction scheme proposed by Foster [3] and Enright [4] 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 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 [4] to correct the interface of Region r.

The randomly placed particles bring noises onto the surface. This is acceptable in animations of splashing liquids, since the noises render a wavy appearance which is visually realistic. However, due to domination of the surface tension effect, the bubble surface should be much smoother than the liquid surface. Thus, the noises caused by particles will become visually disturbing. A direct way to solve this problem is to use sufficiently dense particles. This may become very time consuming and memory consuming. Some other ways of volume preserving or improvement of the particles correction method may be worth a try in the future.

5. 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 far as our knowledge, all previous work concerning surface tension in compute graphics field use explicit time integration schemes. Explicit schemes suffer from a numerical instability problem called the numerical oscillation if the time step is large, and thus 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 cost. Previous work concerning surface tension effect, such as the simulation of water drops [16] and the simulation of small-scale fluid motion [8], costs an extremely large amount of computation to stabilize the simulation. To overcome this drawback, we propose here a novel semi-implicit surface tension model, motivated by [20], 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 the interface position is given by

$$I_\Gamma^{new} = I_\Gamma + \mathbf{u}^{new} \Delta t \tag{13}$$

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

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

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

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

As seen in (15), the additional term compared to Eq. (5) 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 is stronger if the surface tension force exerts longer, and when Δt approaches 0, Eq. (15) will approach Eq. (5).

Eq. (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 the Hessian matrix.

And then we can translate (15) into

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

where we use the implicit scheme only on the standard Laplacian and use the explicit scheme on all other terms. The work in [17] shows that this scheme is unconditionally stable. In such a way, Eq. (16) becomes a Poisson equation that we can solve efficiently by a Preconditioned Conjugate Gradient method.

When a bubble touches a solid surface, its motion will be affected by the wall adhesion force. It can be directly computed at grid cells adjacent to the solid wall, but a more natural way is the virtual surface method proposed by Wang [16]. The mean curvature and the unit normal computed from the virtual surface then can be used in the surface tension force formula (5) to include the wall adhesion effect.

A comparison between our semi-implicit scheme and the explicit scheme can be seen in the surface tension test of Fig. 4. With a small surface tension, both schemes work well; in addition, the result of the semi-implicit scheme has little difference from the one of the explicit scheme, because the small surface tension coefficient scales down the implicit term. With large surface tension, the semi-implicit scheme can still attain correct results due to the stabilization of the implicit term, while the explicit scheme immediately becomes highly instable.

6. 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 bubble's lifecycle 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.

6.1. 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 new generated closed pockets of gas. Greenwood and House [11] 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.

6.2. Merging of bubbles

Bubbles persist until liquid films rupture. If a liquid film separating two bubbles ruptures, the two will merge into one. If a 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 question that how to merge, the question that when to merge is more complicated. A 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 the drainage which means that liquid flows down along the film under the gravity; 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 the *region thickness*, and we assume that the drainage procedure linearly correlates to time. Consequently we attain 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 a bubble will break, and $t_{persist}$ is the user-specified time length that a bubble can persist for. The drainage starts only when a bubble touches another bubble or the atmosphere gas. The *region thickness* is set to be the initial thickness l_{init}

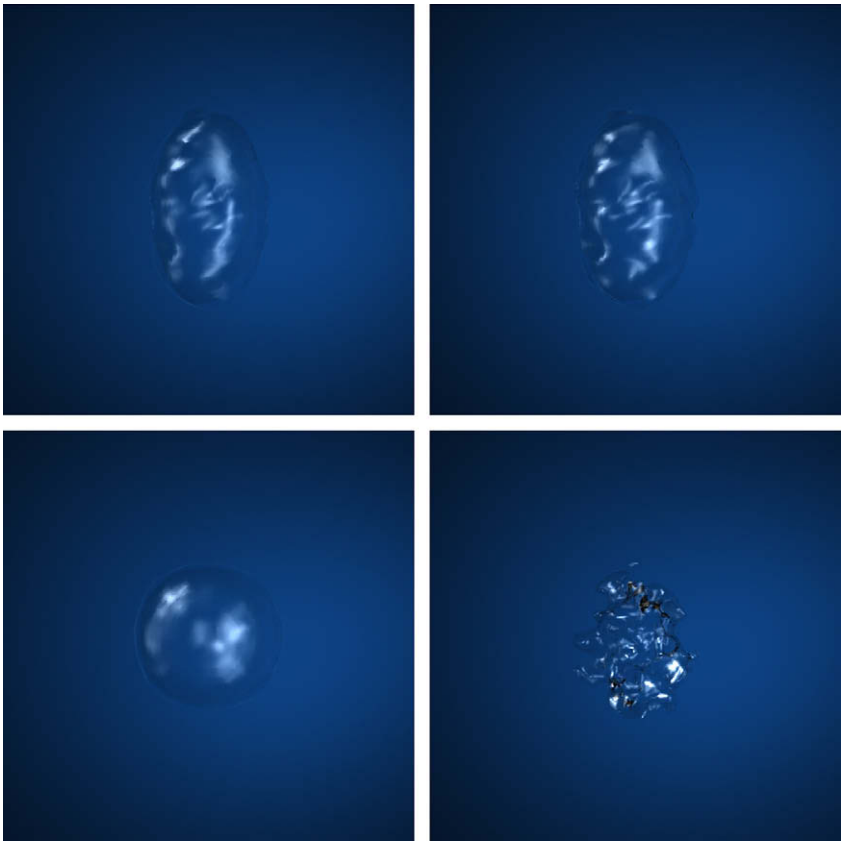

Fig. 4. Surface tension test of a single bubble, snapshot at $t = 0.5$ s. Top left: the semi-implicit method with small surface tension; bottom left: the semi-implicit method with large surface tension; top right: the explicit method with small surface tension; bottom right: the explicit method with large surface tension.

once a bubble is surrounded by liquid, while the *region thickness* of the atmosphere gas is set to be the minimum thickness l_{min}.

We also assume that the thickness change due to the 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 $\varphi_R(\mathbf{X}) = \langle d(\mathbf{X}), r(\mathbf{X}) \rangle$ with the Delta function by

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

Finally, we combine (17) and (18) into an equation governing the evolution of the *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 the *region thickness* for every individual bubble through (19), and we merge adjacent regions whose *film thickness* is less than the minimum thickness l_{min}. 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.

7. Results

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

A comparison of surface tension tests between our semi-implicit scheme and the explicit scheme is represented in Fig. 4. A single elliptic bubble originally stays still in a noisy background wind field. The bubble then deforms under the surface tension force towards a spherical shape with disturbance from the wind field. With a surface tension coefficient, $\sigma = 10$, the explicit method works stably, while the semi-implicit method has a neglectable damping due to the scaled-down implicit term. With a large surface tension coefficient, $\sigma = 500$, the explicit method becomes highly instable, while the semi-implicit method still works well as a result of the strong stabilization effect of the scaled-up implicit term.

Fig. 5 shows a pan of boiling water, in which we can see the whole lifecycle of bubbles. The left sequence is the top

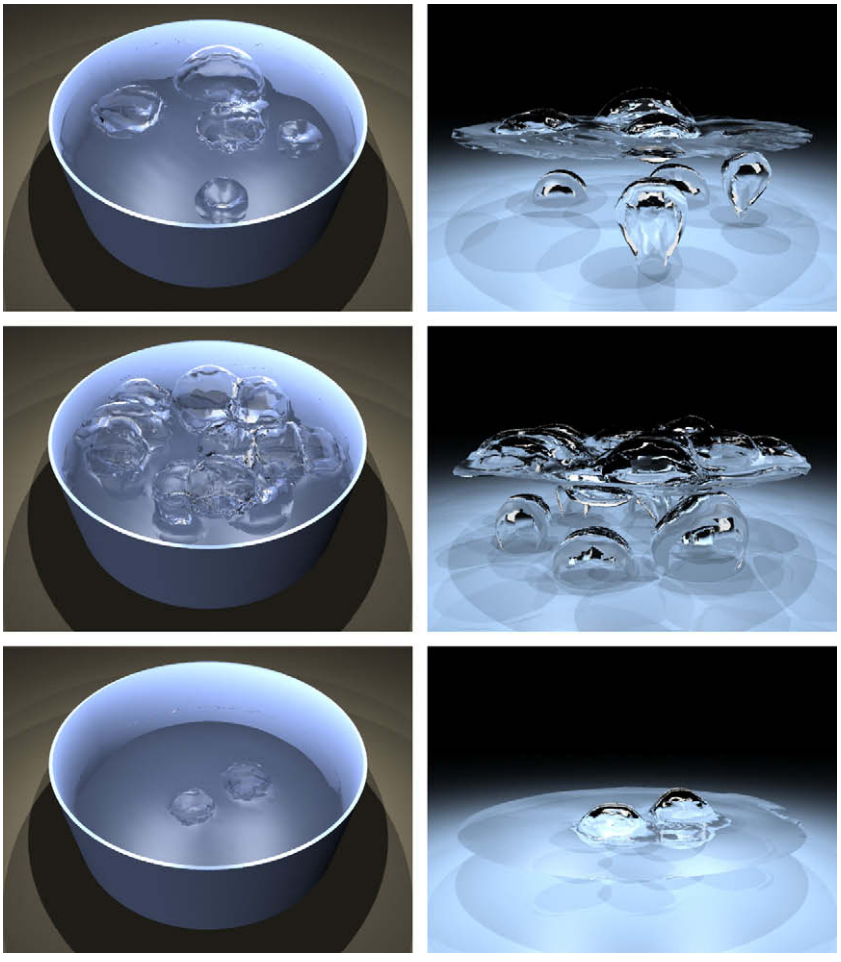

Fig. 5. Boiling water.

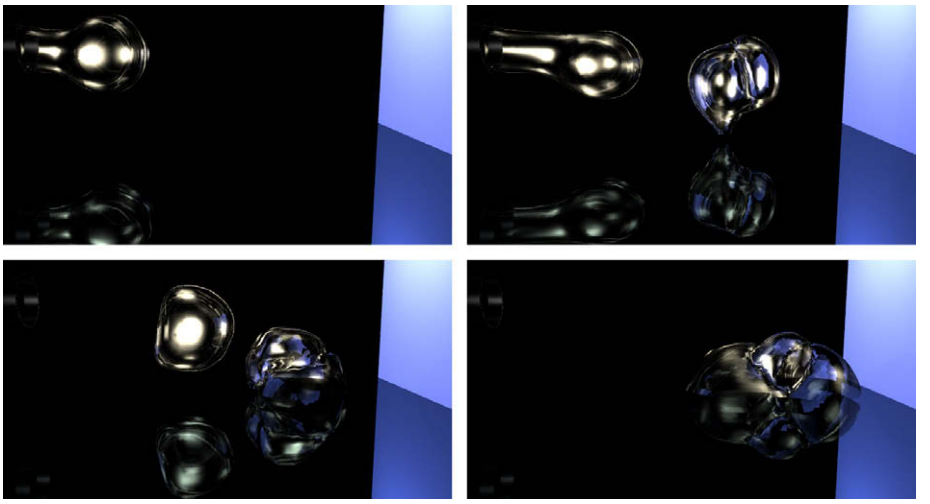

Fig. 6. Blowing bubbles.

view, and the right sequence is the side view without rendering the pan in order to show more details inside water. Bubbles are first generated at the bottom of the pan because of vaporization. Due to the buoyancy, they are detached from the bottom and rise up until touching the 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 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.

Fig. 6 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 effects 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.

Fig. 7 demonstrates a pile of stacking bubbles, in which we can see the complicated structure of many clustering bubbles. The generation of bubbles and the physical environment are the same with the example of Fig. 5, except for that we stop bubbles from bursting to ensure their stacking. When the bubbles rise to the top of water and pile up, water between bubbles is quickly drained by the gravity, and the bubbles quickly form a dry foam structure: many individual gas pockets separated by thin liquid film which contains very little water. This example illustrates the ability of our method to represent the complex foam structure with layers of bubbles.

We performed all the 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 [5]. The speed of our simulation method is approximately the same with the original fluid simulator we based on. For simulation, the first example of Fig. 5 on a $40 \times 20 \times 40$ grid costs about 20 s per frame, the second example of Fig. 6 on a $30 \times 12 \times 15$ grid costs about 2 s per frame, and the third example of Fig. 7 on a $30 \times 20 \times 10$ grid costs about 1 s per frame. Benefit from the unconditional stability of our semi-implicit surface tension model, we took only one time step 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 [18] to extract triangle mesh from implicit surface, and rendered it by an open source renderer PIXIE [19]. Because the focus of this paper is not on the bubbles' rendering, we used a glass shader for all the 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 and third examples, 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 a more adaptive surface extraction method or increasing the grid resolution.

8. Recent related work

New progresses in the simulation of bubbles have been published since our conference paper. Cleary et al. [21] re-

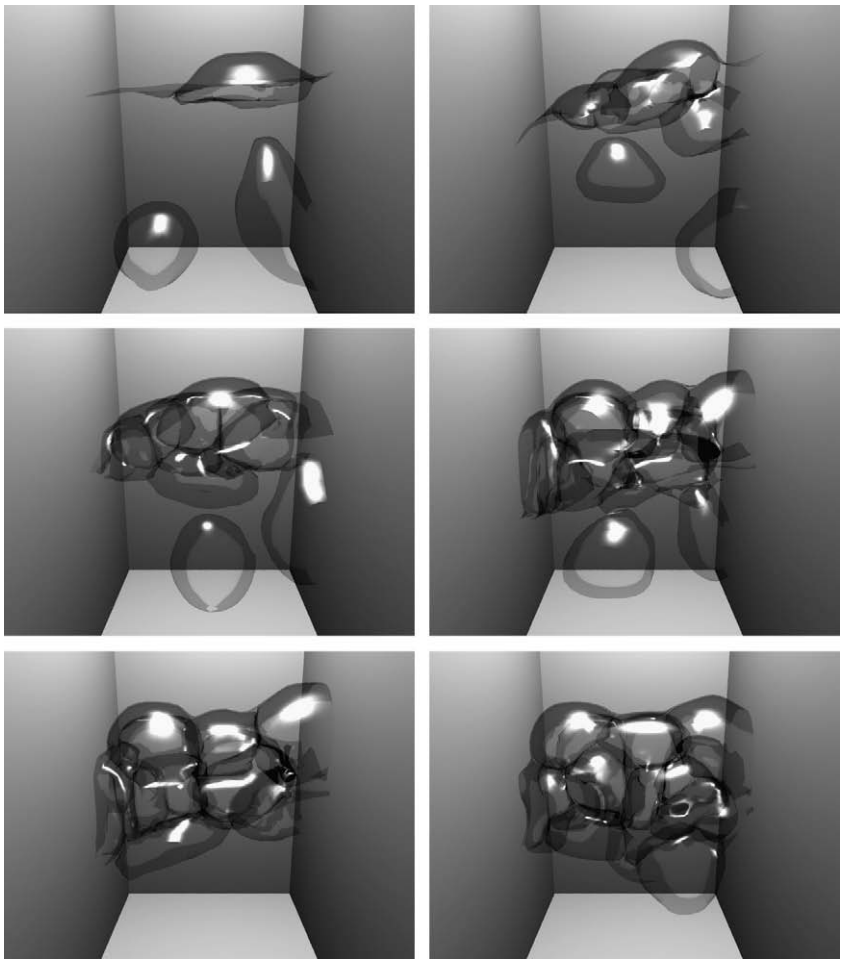

Fig. 7. Stacking bubbles.

searched on a particle-based method focusing on the large number of small volume bubbles, and simulated the bubble formation procedure due to dissolved gas. Since the targeted phenomena are small bubbles, deformation and surface tension are not taken into account. A similar effort has been done in the paper of Hong et al. [22], where they utilized the SPH method for small bubbles, the grid-based method for large bubbles and the background flow, and then coupled these two. Small bubbles are still under the assumption of no deformation, while large bubbles are simulated by the particle level set method, which does not allow multiple regions. Thus this work is limited in the underwater scenario without existence of thin films. Another similar achievement was attained by Mihalef et al. [23]. In their work, particles, generated from the marker level set method, are used to handle both small volume bubbles and droplets, which are coupled with a grid-based simulation. Particle-based method was also adopted in the paper of Thürey et al. [24], but they aimed at real-time applications. A shallow water method was utilized to handle the background fluid, while particles with a procedural vortex flow were used for underwater bubbles, and the SPH method was used for foams on water surface. Bubbles are allowed to deform in a limited way, but cannot merge or split. Furthermore, the assumption of low velocity and strong surface tension also bounded the applicable range. Kim and Carlson [25] focused on the bubble formation procedure due to phase changes, and proposed a simple boiling model. Their work is based on the particle level set method, and thus limited to binary regions, i.e. underwater bubbles. To capture full effects of clustering large bubbles, our regional level set method was adopted by Kim et al. [26] who remedied the volume loss problem by a PI controller.

Note that most of recent papers focused on small volume bubbles which are difficult to capture by a grid-based method and typically have little deformation. Thus particle-based methods were adopted in these papers to approximate small bubbles and save computation cost. However, as stated in Section 2.2, particle-based methods are lack of or with limited deformation. For large bubbles whose deformation has significant impact on visual effects, these methods are too simplified to attain full effects of bubbles, including geometry and topology changing. In contrast, our method offers a solution to the above problem. Our method was adopted in the paper of Kim et al.

[26], which also aimed at large bubbles. The results in their paper demonstrated again the capability of our method to easily capture the geometry and motion of thin films between bubbles. Users can choose between our method and a particle-based method according to their specific requirements.

9. Conclusion

We propose a novel 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 very thin liquid films, 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 the surface tension, we exploit a novel semi-implicit surface tension model which is unconditionally stable, and can easily be solved by a 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.

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 [8], 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 [17].

Acknowledgments

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).

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.gmod.2009.08.001.

References

[1] N. Foster, D. Metaxes, Realistic animation of liquids, Graphical Models and Image Processing 58 (1996) 471–483.
[2] J. Stam, Stable fluids, in: Proceedings of SIGGRAPH '99, 1999, pp. 121–128.
[3] N. Foster, R. Fedkiw, Practical animation of liquids, in: Proceedings of SIGGRAPH '01, 2001, pp. 23–30.
[4] D. Enright, S. Marschner, R. Fedkiw, Animation and rendering of complex water surfaces, in: ACM Transactions on Graphics, Proceedings of SIGGRAPH '02 21 (2002) 736–744.
[5] T.G. Goktekin, A.W. Bargteil, J.F. O'Brien, A method for animating viscoelastic fluids, in: Proceedings of ACM SIGGRAPH '04, 2004, pp.463–468.
[6] F. Losasso, F. Gibou, R. Fedkiw, Simulating water and smoke with an octree data structure, in: Proceedings of SIGGRAPH '04, 2004, pp. 457–462.
[7] O. Song, H. Shin, H. Ko, Stable but non-dissipative water, ACM Transactions on Graphics 24 (2005) 81–97.
[8] J. Hong, C. Kim, Discontinuous fluids, in: Proceedings of SIGGRAPH '05, 2005, pp.915–920.
[9] D. Roman, Animation of soap bubble dynamics, cluster formation and collision, in: Computer Graphics Forum, Proceedings of Eurographics'2001 20 (2001) C67–C75.
[10] H. Kück, C. Vogelgsang, G. Greiner, Simulation and rendering of liquid foams, in: Proceedings of Graphics Interface '02, 2002, pp. 81–88.
[11] S. Greenwood, D. House, Better with bubbles: enhancing the visual realism of simulated fluid, in: Proceedings of 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2004, pp. 287–296.
[12] J. Hong, C. Kim, Animation of bubbles in liquid, in: Proceedings of Eurographics '03, 2003, pp. 253–262.
[13] D. Weaire, S. Hutzler, The Physics of Foams, Oxford University Press, Oxford, 1999.
[14] J. Bloomenthal, K. Ferguson, Polygonization of non-manifold implicit surfaces, in: Proceedings of SIGGRAPH '95, 1995, pp. 309–316.
[15] S. Yamazaki, K. Kase, K. Ikeuchi, Non-manifold implicit surfaces based on discontinuous implicitization and polygonization, in: Proceedings of Geometric Modeling and Processing, 2002, pp. 138–146.
[16] H. Wang, P.J. Mucha, G. Turk, Water drops on surfaces, in: Proceedings of SIGGRAPH '05, 2005, pp. 921–929.
[17] J. Xu, H. Zhao, An Eulerian formulation for solving partial differential equations along a moving interface, Journal of Scientific Computing 19 (2003) 573–594.
[18] W.E. Lorensen, H.E. Cline, Marching cubes: a high resolution 3D surface construction algorithm, in: International Conference on Computer Graphics and Interactive Techniques, 1987, pp. 163–169.
[19] O. Arikan, Pixie: photorealistic renderer, 2005. <http://sourceforge.net/projects/pixie>.
[20] S. Hysing, A new implicit surface tension implementation for interfacial flows, International Journal for Numerical Methods in Fluids 51 (2006) 659–672.
[21] P.W. Cleary, S.H. Pyo, M. Prakash, B.K. Koo, Bubbling and frothing liquids, in: ACM Transactions on Graphics, Proc. of SIGGRAPH'2007 26 (2007).
[22] J.-M. Hong, H.-Y. Lee, J.-C. Yoon, C.-H. Kim, Bubbles alive, in: Proceedings of ACM SIGGRAPH'2008, 2008.
[23] V. Mihalef, D. Metaxas, M. Sussman, Simulation of two-phase flow with sub-scale droplet and bubble effects, in: Proceedings of Eurographics 2009, 2009.
[24] N. Thürey, F. Sadlo, S. Schirm, M. Müller-Fischer, M. Gross, Real-time simulations of bubbles and foam within a shallow water framework, in: Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2007, pp. 191–198.
[25] T. Kim, M. Carlson, A simple boiling module, in: Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer animation, 2007, pp. 27–34.
[26] B. Kim, Y. Liu, I. Llamas, X. Jiao, J. Rossignac, Simulation of bubbles in foam with volume control, in: ACM Transactions on Graphics, Proceedings of SIGGRAPH'2007 26 (2007).