Animating Bubble Interactions in a Liquid Foam

Oleksiy Busaryev*
Ohio State University

Tamal K. Dey*
Ohio State University

Huamin Wang*
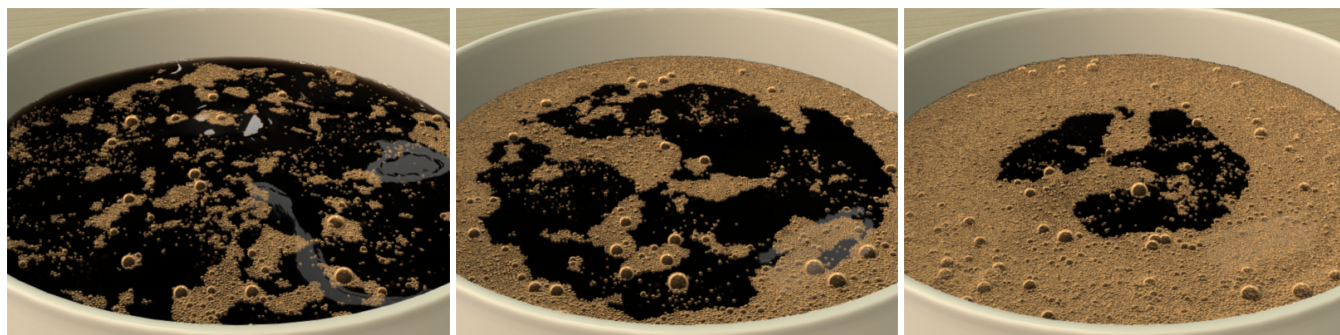Ohio State University

Zhong Ren†
Zhejiang University

Figure 1: *Coke foam. By representing foam geometry using a weighted Voronoi diagram, our particle-based algorithm can efficiently provide bubble features in existing liquid animation. This example contains up to 100K bubbles and each frame takes less than 20 seconds to simulate.*

Abstract

Bubbles and foams are important features of liquid surface phenomena, but they are difficult to animate due to their thin films and complex interactions in the real world. In particular, small bubbles (having diameter <1cm) in a dense foam are highly affected by surface tension, so their shapes are much less deformable compared with larger bubbles. Under this small bubble assumption, we propose a more accurate and efficient particle-based algorithm to simulate bubble dynamics and interactions. The key component of this algorithm is an approximation of foam geometry, by treating bubble particles as the sites of a weighted Voronoi diagram. The connectivity information provided by the Voronoi diagram allows us to accurately model various interaction effects among bubbles. Using Voronoi cells and weights, we can also explicitly address the volume loss issue in foam simulation, which is a common problem in previous approaches. Under this framework, we present a set of bubble interaction forces to handle miscellaneous foam behaviors, including foam structure under Plateau's laws, clusters formed by liquid surface bubbles, bubble-liquid and bubble-solid coupling, bursting and coalescing. Our experiment shows that this method can be straightforwardly incorporated into existing liquid simulators, and it can efficiently generate realistic foam animations, some of which have never been produced in graphics before.

Keywords: Liquid, foam, bubble interaction, weighted Voronoi diagram, natural phenomena, surface tension.

Links: ◈DL ⬥PDF ⬥WEB ⬥VIDEO

*e-mail: {busaryev,tamaldey,whmin}@cse.ohio-state.edu
†e-mail: zhongren@siggraph.org

1 Introduction

Enriching liquid animation with bubbles and foams can significantly improve its realism. However, bubbles and foams are difficult to simulate due to their different physical properties in the real world, such as surface tension of the liquid, liquid volume percentage, and bubble sizes. For example, small bubbles in a microfoam of coffee latte are creamy and liquid-like, while bubbles in a soap foam are larger and more transparent. Bubbles also have complex interaction behaviors, including clustering, coalescing, deforming, and interacting with liquids or solids. Physical nature of these phenomena has attracted attention of mathematicians, physicists and computer scientists. A considerable amount of research [Brakke 1992; Gardiner et al. 2000; Weaire and Hutzler 2001; Kim et al. 2007] was done to form a mathematical description of foam geometry and to model dynamic foam properties, such as surface evolution and topological changes in the foam structure. Being physically accurate, these models are often not directly applicable to handle a large number of bubbles, because of their computational cost.

Alternatively, particle-based approaches [Kück et al. 2002; Greenwood and House 2004; Cleary et al. 2007] have been proposed in graphics to efficiently simulate liquid bubbles and foams, assuming that bubble surface deformation is less noticeable in a dense foam. This is a valid assumption for small bubbles, not only because of their small scales, but also because of the larger surface tension effect that can quickly restore bubble shapes from deformation. In order to model inter-bubble dynamics, these methods typically treat each bubble as a sphere, and then they apply interaction forces whenever two spheres intersect. While this approach is sufficient for standalone bubbles, it fails to properly capture connectivity when bubbles form clusters and is of limited use in modeling surface-tension-based interaction among bubbles on a liquid surface. How to maintain volume conservation for each individual bubble is another challenging problem for these techniques, since foam geometry and bubble shapes are not explicitly represented.

To solve these problems, we propose a simple and efficient particle-based algorithm that can simulate realistic bubble interactions in complex foam scenarios, as shown in Figure 1. The basic idea behind this method is to represent bubbles using weighted points and gather them into a weighted Voronoi diagram. Our work shows that the algorithm can benefit from using this diagram in three ways.

First, it explicitly reveals bubble connectivity. From the weighted Voronoi diagram, we can directly infer which two bubbles share the liquid film, and which bubbles are potential candidates for tension-based interaction. The Voronoi diagram allows us to accurately model the instability of multiple-bubble intersections that is quickly resolved into triple (in 2D) or quadruple (in 3D) intersections in the real world. Secondly, it provides a simple and natural approximation to the actual foam geometry, and it can be directly used for rendering. Finally, the weighted Voronoi diagram can be used to explicitly calculate bubble volumes, which is a crucial component in the volume correction process.

Our system pipeline starts each time step with formulating bubble interaction forces, based on the bubble connectivity information provided by the weighted Voronoi diagram. The forces are then used in an implicit integrator to evolve bubble positions and velocities over time. After that, it calculates the volume of each bubble using the Voronoi cell, and compensates volume changes by adjusting the Voronoi weight. Finally, the system handles bursting and coalescing effects by performing topological changes, and it reconstructs the weighted Voronoi diagram for the next time step.

In summary, we propose a novel particle-based algorithm to simulate bubble interactions in a liquid foam, by making the following contributions: 1) a weighted Voronoi representation that models bubble connectivity and foam geometry; 2) a set of bubble interaction forces that can produce various interaction behaviors; and 3) a volume correction method for particle-based bubbles. We illustrate our method with animations of typical real-world foam scenarios, including clustering, stacking, sticking to liquid or solid surfaces, bursting and coalescing. We also incorporate this method into a particle level set liquid simulator, and test its capability of adding realistic bubbles and foams in liquid animation.

2 Previous Work

Foam Physics. Real-world foams exhibit significantly different structures and dynamic behaviors due to their physical properties. A dry foam, in which the liquid volume is typically less than 1 percent of the whole volume, can form a specific structure under laws discovered by the Belgian physicist Joseph Plateau. Taylor [1976] later proved that this structure minimizes the bubble surface area under volume constraints. Different from the dry foam, a wet foam has more complex structures due to its liquid volume, and it has been studied in multiple ways [Weaire et al. 1993; Herzhafta et al. 2005; Piazza et al. 2008]. Foam structures can also be classified according to their bubble sizes, such as monodisperse foam [Kraynik et al. 2003] that contains uniformly sized bubbles and polydisperse foam [Kraynik et al. 2004], in which the bubble size can vary. Compared with foam structure, foam dynamics is a less studied problem due to the difficulty in observing dynamic behaviors, including drainage, rheology, coarsening, and merging. A comprehensive study on both foam structure and foam dynamics can be found in Weaire and Hutzler's book [2001]. In this paper, we focus on simulating rheological and merging behaviors of polydisperse foams, in both dry and wet cases for graphics applications.

Due to the similarity between the weighted Voronoi diagram and foam geometry, physicists studied using the diagram to model static foam structures [Redenbach et al. 2012] in the past. Although they also used the diagram to initialize foams in dynamic simulation, they chose to solve bubble dynamics using more sophisticated models [Kraynik et al. 2004; Brakke 1992] instead, since only a small set of bubbles were considered in their typical examples. In contrast, we found the weighted Voronoi diagram can be used directly in dynamic simulation as well, and it provides a good approximation to foam structure for a great number of bubbles.

Foam Simulation. The simulation of bubbles and foams involves two aspects: the deformation of individual bubble surfaces, and the interactions among bubbles, liquids and solids. An early example of deformable bubbles is proposed by Brakke [1992]. Durikovic [2001] used a spring mesh to represent bubble surfaces and approximated interaction forces using an intermolecular Van der Waals force model. To facilitate topological changes, Kim et al. [2011] developed a 2D algorithm using the immersed boundary method. Kelager [2009] introduced the ghost bubble technique to the vertex-based dry foam simulation. Implicit representations, such as Volume-of-Fluid (VOF), can also be used to simulate deformable bubbles, as Hong et al. [2003] and Mihalef et al. [2006] showed. Zheng et al. [2006] proposed a regional level set method to implicitly model liquid foams as multi-manifold surfaces. Kim et al. [2007] addressed the volume loss in the regional level set method using a volume control technique. While these methods can handle surface deformation of individual bubbles, they need considerable computational time to handle a large number of bubbles.

By ignoring surface deformation, particle-based techniques are specifically developed for handling bubble interactions with the environment. Durian [1995] first proposed to use a mass-spring model to animate bubble interactions in 2D. Kück et al. [2002] extended this idea to 3D, and also developed a way to render Plateau borders and curved films between contacting bubbles. Greenwood and House [2004] incorporated the Kück model into a particle-level-set-based liquid simulator. Bubble interactions may also be approximated by Smoothed Particle Hydrodynamics (SPH) as in [Cleary et al. 2007; Thürey et al. 2007; Hong et al. 2008], though they typically do not consider internal foam geometry. While we also represent bubbles as particles in our system, we improve simulation accuracy by the use of weighted Voronoi diagrams, and we can handle more bubble interaction behaviors.

Fluid Simulation. Fluid simulation is an important research topic in computer graphics, and various techniques were proposed for animation purposes, including Eulerian approaches [Foster and Metaxas 1996; Stam 1999; Enright et al. 2002; Chentanez and Müller 2011], Smoothed Particle Hydrodynamics (SPH) [Müller et al. 2003; Adams et al. 2007], Lagrangian-based methods using meshes [Bargteil et al. 2007; Thürey et al. 2010; Wicke et al. 2010], and simplified or hybrid algorithms [Losasso et al. 2004; Bargteil et al. 2006; Wang et al. 2007; Losasso et al. 2008]. Some of these approaches [Brochu et al. 2010; Sin et al. 2009] also leveraged Voronoi diagrams for simulation. Since our system provides additional foam features to liquid animation, it can be straightforwardly incorporated into any liquid simulator.

3 Bubbles and Foams

A *spherical bubble* is a three-dimensional ball $B(x, r)$ of radius r centered at the point x. These spherical bubbles come together to form a *foam*. Therefore, the space of the foam is the pointwise union of a set of balls: $\cup B_i$. Each spherical bubble B_i in a foam takes a share of this space, which is called the *bubble* of B_i. The center and the radius of a bubble are simply taken as those of its spherical counterpart. The boundary of a bubble is a set of curved surface patches called *films*. These films are formed in a way so that the total surface area over all bubbles is minimized under volume constraints, as Taylor [1976] and Sullivan [1998] pointed out. In particular, the geometry of dry foam, whose liquid volume is typically less than 1 percent, can be described by Plateau's laws:

1. Films meet in threes along *Plateau borders* at angles of $2\pi/3$.
2. Plateau borders meet in fours at angles of $\cos^{-1}(-1/3)$.
3. Each bubble film has constant mean curvature.
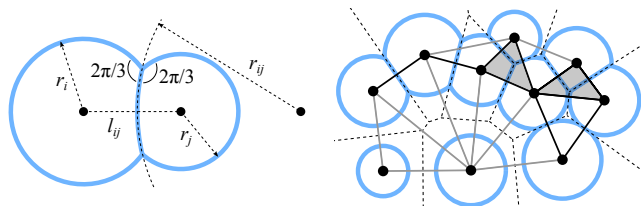
(a) Two bubbles resting in the Plateau equilibrium. The interfacial film between bubbles is a spherical patch with radius r_{ij}.

(b) A weighted Voronoi diagram (dotted), Delaunay complex (solid edges), space-filling diagram (blue curves), and alpha complex (dark edges and triangles).

Figure 2: *Bubble and foam configuration.*

Any dry foam geometry not following Plateau laws is unstable, and the surface tension force will rearrange those bubbles back into the Plateau equilibrium. Given a pair of contacting bubbles as shown in Figure 2a, the interfacial film between them is not necessarily planar and it can be modeled as a patch of sphere whose radius r_{ij} is calculated as: $\frac{1}{r_{ij}} = \frac{1}{r_j} - \frac{1}{r_i}$, in which r_i and r_j are the radii of the two bubbles respectively and $r_i \geq r_j$. We can verify that this configuration corresponds to the Plateau equilibrium. We also notice that r_{ij} becomes larger when the two radii get closer. In particular, the film is planar when the two radii are identical. For multiple intersections, films between bubbles may not be spherical. Instead they can be approximated by a mean curvature proportional to the pressure difference across the film. When pressure difference is nonzero, the film is curved towards the bubble with a lower pressure.

3.1 Foam Representation

We define a foam as a tessellation of the space occupied by a union of balls. A well known data structure called *space-filling diagram* exists in computational geometry, which also tessellates such a space. Observing this analogy, we represent a foam as the space filling diagram. Simply stated, a space filling diagram is the intersection of a union of balls with the *weighted Voronoi diagram* (also called *power diagram*) which is a generalization of the well known Voronoi diagram [Aurenhammer 1987]. Although it does not exactly model curved interfacial films in the real world, it provides a sound approximation to bubble structures and foam geometry. A Voronoi diagram is sufficient in representing monodisperse foams, in which bubbles have the same size. To further handle polydisperse foams with different bubble sizes, we need the weighted Voronoi diagram. Specifically, an original spherical bubble $B_i = B(x_i, r_i)$ is represented as the intersection between B_i and the weighted Voronoi cell of x_i with weight r_i. A detailed definition of these concepts is given below.

Voronoi Diagram and Delaunay Triangulation. Let P be a point set in $\mathbb{R}^3$. For any point $p \in P$, the *Voronoi cell* V_p of p is defined as the locus of points in $\mathbb{R}^3$ having p as their nearest neighbor in P: $V_p = \{x \in \mathbb{R}^3, \text{ for } \forall q \in P : |x - p| \leq |x - q|\}$. Each Voronoi cell is convex and its boundary consists of lower-dimensional convex faces. The collection of Voronoi cells and their faces forms a cell complex tessellating $\mathbb{R}^3$, called the *Voronoi diagram* of P. Its dual complex is called the *Delaunay triangulation* of P.

Weighted Counterparts. Voronoi diagrams and Delaunay triangulations can also be constructed using non-Euclidean metrics. Specifically, we can associate each point $p \in P$ with a real-valued weight r ($r \geq 0$). This is equivalent to representing each weighted point p as a ball with radius r centered at p. The squared

weighted distance of any point $x \in \mathbb{R}^3$ from $p \in P$ is given by $d^2(x, p) = |x - p|^2 - r^2$. The *weighted Voronoi diagram* and its dual *weighted Delaunay triangulation* are then defined in the same way as the original ones, except that they replace Euclidean distances with squared weighted distances. Figure 2b shows a weighted Voronoi example with its Delaunay triangulation.

Space-Filling Diagrams. After we construct the Voronoi diagram by treating each spherical bubble as a weighted point, we represent the bubble of $B_i = B(x_i, r_i)$ in a foam as the intersection of B_i with the Voronoi cell for x_i. Films between two bubbles are defined as the intersection of balls with Voronoi polygons. Such decomposition of the union of balls into convex cells is known as a *space-filling diagram*. It should be noted that we do not need an explicit representation of the foam geometry during simulation. Instead, we use the connectivity information among bubbles, which is captured by the dual of the space filling diagram called the *alpha complex* [Edelsbrunner 2001]. For example, Figure 2b shows the dark edges and shaded triangles in a 2D alpha complex. The edges are dual to Voronoi edges which correspond to the common intersection of two bubbles and the triangles are dual to the Voronoi vertices which correspond to the common intersection of three bubbles. The edges of the alpha complex provide information about which bubbles are in contact.

4 Foam Dynamics

Various factors contribute to the motion of bubbles. For example, two bubbles in contact can experience both repulsion and attraction forces, which try to bring them back to the Plateau equilibrium. Meanwhile, bubbles on a liquid surface are likely to form clusters, caused by weak interaction forces even before they contact. In this section, we present a set of bubble forces to model various foam effects in our system. These forces can be roughly grouped into three categories based on the type of their interactions: bubble-bubble interaction, bubble-liquid interaction, and bubble-solid interaction.

The interactions between two bubbles can be triggered either by a strong or a weak interaction force as we will see later. The strong interaction happens only between bubbles that are in contact while the weak interaction happens only between bubbles that are Voronoi neighbors but are not in contact. We distinguish the latter from the former by testing Delaunay edges that are not alpha complex edges.

4.1 Bubble-Bubble Interaction

Bubbles in a dry foam (such as soap foam), whose liquid volume is typically less than 1 percent, form a specific geometric structure that minimizes the total surface area under volume constraints. Formulated in the 19th century by the Belgian physicist Joseph Plateau, Plateau's laws describe this structure using three conditions as we described before. Soap bubbles not in this configuration are unstable, and the surface tension force quickly rearranges them back into the Plateau equilibrium. In particular, when two separate bubbles touch each other, a strong attraction force tries to move them closer into the Plateau configuration. Meanwhile, if bubbles are closer than necessary, the force will become repulsive due to both surface minimization and volume constraints. Given two spherical bubbles with radii r_i and r_j respectively, we can reach the Plateau equilibrium when the distance l_{ij} between two bubble centers satisfies: $l_{ij}^2 = r_i^2 + r_j^2 - r_i r_j$. Figure 2a shows such an example, in which bubble films form $2\pi/3$ angles according to Plateau's laws. This condition also provides a reasonable approximation for multiple bubble cases. However, it becomes less valid for wet foams, in which the extra liquid in interfacial films can affect the surface tension. Unfortunately, the geometric structure of wet foam is more
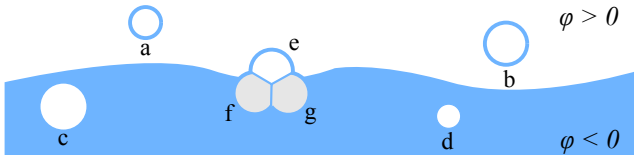
Figure 3: *Liquid bubbles (c and d), surface bubbles (f and g), and air bubbles (a, b, and e). They are classified according to their signed distances to the liquid surface.*

complex, and no clear conclusions have been made to this problem yet. Here we introduce a wetness coefficient λ ($1 \leq \lambda \leq 1$) to account for different wetness conditions:

$$l_{ij}^2 = r_i^2 + r_j^2 + (3\lambda - 1)r_i r_j. \tag{1}$$

For example, setting $\lambda = 0$ allows us to mimic a completely dry foam, where bubbles form a polyhedral structure. Making λ close to 1 models a wet foam, where bubbles have almost spherical shape and merely touch each other in the equilibrium state.

Compared to the static foam geometry, bubble-bubble interaction in dynamic environment is more difficult to measure and study. Similar to [Kück et al. 2002; Greenwood and House 2004], we use a simple spring model to handle the force between two contacting bubbles. Let x_i and x_j be the centers of two intersecting bubbles neighboring in the weighted Voronoi diagram, we define a *strong interaction force* applied on x_i using the resting length l_{ij}:

$$f_i^{\mathrm{sint}} = -k \sum_j \left(x_i - x_j - l_{ij} \frac{x_i - x_j}{|x_i - x_j|} \right), \tag{2}$$

in which k is a constant stiffness coefficient.

4.2 Bubble-Liquid Interaction

To simulate bubble-liquid interaction, we classify bubbles into three types based on their positions with respect to a liquid volume. Let φ be the signed distance function to a liquid surface, a bubble centered at x_i with radius r_i is named as a liquid bubble, if $\varphi(x_i) < -r_i$. When $-r_i \leq \varphi(x_i) \leq r_i$, we define the bubble as a surface bubble, if and only if it is not isolated from the liquid surface by other bubbles. The rest of bubbles are called air bubbles. Figure 3 shows an example, in which bubbles e, f, and g all satisfy the condition: $-r_i \leq \varphi(x_i) \leq r_i$. However, bubble e is still an air bubble, since it is separated from the liquid by bubble f and g. We detect this separation by sampling φ over the bubble's spherical boundary. If there exists a negative sample that is not covered by other bubbles, it signifies a surface bubble, otherwise it is an air bubble. Since air bubbles do not directly interact with the liquid, we will only consider liquid bubbles and surface bubbles next.

Liquid Bubble. A liquid bubble within a liquid volume is subject to two forces in our system. Similar to [Cleary et al. 2007] and [Hong et al. 2008], we first define a liquid drag force depending on the bubble velocity relative to the fluid:

$$f_i^{\mathrm{drag}} = c_{\mathrm{drag}} r_i^2 (u(x_i) - v_i)|u(x_i) - v_i|, \tag{3}$$

in which c_{drag} is a drag coefficient, $u(x_i)$ is the liquid velocity at the bubble center x_i, and v_i is the bubble velocity. The other force is the buoyant force that lifts the bubble up to the surface. Assuming liquid bubbles are incompressible in the liquid volume, we have:

$$f_i^{\mathrm{buoy}} = -V_i \rho g = -\tfrac{4}{3} \pi r_i^3 \rho g. \tag{4}$$

where g is the gravity acceleration, ρ is the liquid density, and V_i is the bubble volume.

(a) A surface bubble in equilibrium
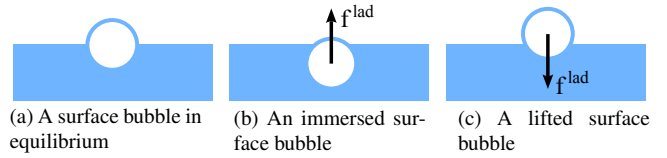
(b) An immersed surface bubble

(c) A lifted surface bubble

Figure 4: *A surface bubble in different states. Its equilibrium state can be reached by the use of a liquid adhesion force.*

(a) Bubbles before being attracted
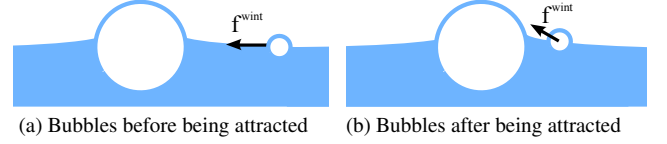
(b) Bubbles after being attracted

Figure 5: *Bubble clustering caused by surface tension. We use a weak interaction force to obtain this effect.*

Surface Bubble. Surface bubble motion is highly influenced by surface tension. While modeling surface tension directly on bubbles are difficult and computationally expensive, we define two interaction forces here, both of which are formulated to achieve specific surface bubble behaviors. Like other surface tension effects, surface bubbles move in a way such that the liquid surface area can be minimized. Therefore, a surface bubble is able to reach an equilibrium on the liquid surface, under both the buoyant force and the surface tension force, as Figure 4a shows. When the bubble gets immersed (in Figure 4b), the buoyant force becomes larger while the surface tension force becomes weaker, so the total force will push the bubble to its equilibrium state. Meanwhile, a lifted surface bubble (in Figure 4c) receives a larger surface tension force than the buoyant force, so it will be pulled back. Since it is difficult to accurately calculate both forces in this case, we define a liquid adhesion force f^{lad} to help the surface bubble stay on the liquid surface:

$$f_i^{\mathrm{lad}} = -\sigma_{\mathrm{lad}} \varphi(x_i) \nabla \varphi(x_i), \tag{5}$$

in which σ_{lad} is an adhesion coefficient, and $\nabla \varphi(x_i)$ is the liquid surface normal at x_i. Surface tension can also cause two separated bubbles to move toward each other, in order to minimize the overall liquid surface area as Figure 5 shows. So for any two non-contacting surface bubbles that are neighbors in the weighted Voronoi diagram, we define a *weak interaction force* using the approximated sine value of the surface inclination angle θ_{ij}:

$$f_{ij}^{\mathrm{wint}} = \alpha \sin \theta_{ij} \frac{x_j - x_i}{|x_j - x_i|}, \quad \sin \theta_{ij} = \frac{r_j}{|x_i - x_j|}, \tag{6}$$

in which α is a force magnitude parameter. It should be noted that inter-bubble interactions, including both strong and weak interaction, rely on the topology of the weighted Delaunay triangulation (or its dual weighted Voronoi diagram). Delaunay edges that belong to the alpha complex indicate strong interaction forces for contacting bubbles, while the remaining Delaunay edges help us formulate weak interaction forces for surface bubbles.

4.3 Bubble-Solid Interaction

In order to model the hydrophilicity of certain solid objects, we use an adhesion force to prevent bubbles from leaving solid surfaces:

$$f_i^{\mathrm{sad}} = -\sigma_{\mathrm{sad}} \psi(x_i) \nabla \psi(x_i), \tag{7}$$

in which σ_{sad} is an adhesion coefficient and $\psi(x_i)$ gives the signed distance from the bubble center x_i to the solid. The adhesion force acts on each bubble that has ever touched the solid. In addition, we

define a solid attraction force to move surface bubbles closer to the solid, due to a similar surface minimization reason we discussed before in Subsection 4.2:

$$f_i^{\text{sat}} = -\beta \nabla \psi / |\psi|, \qquad (8)$$

in which β is a solid attraction coefficient.

5 Foam Simulation

Based on foam representation and foam dynamics presented in Section 3 and 4 respectively, we develop a foam simulator to update bubbles and foams in each time step. It first uses an implicit integrator to evolve bubble positions and velocities, according to foam dynamics. It then applies a volume conservation constraint to compensate the bubble volume loss during simulation, especially when bubbles are heavily squeezed. Finally, we perform topological updates on the foam structure and reconstruct the weighted Voronoi diagram, to ensure that it remains valid for further simulation.

5.1 Time Integrator

We use the implicit solver proposed by Baraff and Witkin [1998] to evolve bubble particles over time. Given bubble positions $\{x_i^t\}$ and velocities $\{v_i^t\}$ at time t, the solver uses the backward Euler method to compute bubble velocities $\{v_i^{t+1}\}$ at the next time step:

$$\left(\mathbf{M} - \Delta t \partial f^t / \partial v - \Delta t^2 \partial f^t / \partial x \right) v^{t+1} = \mathbf{M} v^t + \Delta t f^t, \qquad (9)$$

in which $\mathbf{M}$ is a mass matrix, Δt is the time step, and f, x and v are vectors of bubble forces, positions, and velocities respectively. The force vector f is made of the gravity force, the air damping force, and the total interaction force. The gravity force $f_i^{\text{grav}} = m_i g$ on bubble i is independent of bubble position and velocity, so its Jacobian matrices $\partial f^{\text{grav}} / \partial x$ and $\partial f^{\text{grav}} / \partial v$ are both zero matrices. We define the damping force using two damping coefficients (c_{vis} and c_{lap}) and a normalized Laplacian matrix $\mathbf{L}$: $f^{\text{damp}} = -c_{\text{vis}} v - c_{\text{lap}} \mathbf{L} v$. Its Jacobian matrix can be written as:

$$\mathbf{c}_{ij} = \begin{cases} -(c_{\text{vis}} + c_{\text{lap}}) \mathbf{I}, & \text{for } i = j \\ c_{\text{lap}} \left(|N_i| |N_j| \right)^{-1/2} \mathbf{I}, & \text{for } i \neq j \text{ and } i \in N_j \\ \mathbf{0}, & \text{otherwise} \end{cases} \qquad (10)$$

in which $\mathbf{c}_{ij}$ is a 3×3 sub-matrix, and N_i is the 1-ring contact neighborhood of vertex i, given by the weighted Voronoi diagram. The overall interaction force is the sum of three interaction forces discussed in Section 4. Most interaction forces are comparably small, so they can be treated explicitly by simply ignoring their Jacobian matrices. The only exception is the strong bubble interaction force modeled by a stiff spring in Equation 2. Similar to the formula proposed by Choi and Ko [2002], we use Taylor expansion to find the Jacobian matrix of f^{sint} respect to x. We also drop the geometric term when the spring is compressed, to ensure numerical stability.

The linear system in Equation 9 is guaranteed to be symmetric positive definite. We solve it using the Preconditioned Conjugate Gradient (PCG) method with an incomplete Cholesky preconditioner. Since interaction forces in our method depend on bubble connectivity that may vary over time, the whole system is not unconditionally stable. Fortunately, our experiment shows that the system can robustly handle large time steps ($\Delta t \in [0.005s, 0.02s]$) for most examples without any oscillation artifacts.

5.2 Volume Correction

The weighted Voronoi diagram changes when bubble move. As a result, some bubbles represented by the cells of the space-filling
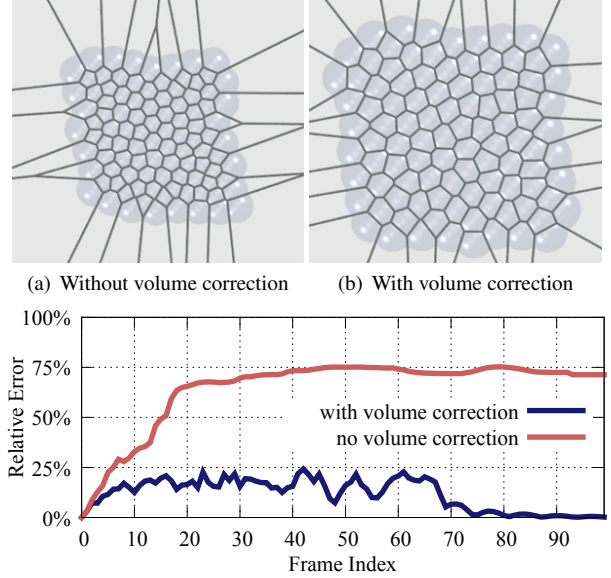

(a) Without volume correction   (b) With volume correction

Figure 6: *A 2D cluster. This example visualizes a cluster of uniform bubbles with (a) and without (b) volume correction. The plot shows that the maximum relative error of bubble volumes can be greatly reduced, after using our volume correction method.*

diagram may experience considerable volume changes. To preserve the bubble volume, we perform a volume correction step.

We consider two ways to adjust the bubble volume: to change the bubble position x_i, or to change its weight r_i. By modifying the weight alone we can implicitly increase or decrease the volume of an individual bubble. However, it may affect the volume of its neighbors. One may try to iterate the process to rectify the volumes of all bubbles. But, this does not work because the total volume of the union of all bubbles changes only by the modifications of the bubbles on the boundary of the union. The intended changes for interior bubbles are not reflected in this sum. We need position changes along with weight changes to overcome this constraint. Unfortunately, explicit adjustment of positions without causing artifacts in bubble dynamics is difficult. We solve this dilemma by adjusting weights alone and then letting the simulation adjust the positions. The resting length l_{ij} between two neighboring bubbles depends on their radii. If r_i increases relative to r_j, the bubble at x_j is pushed away from x_i decreasing the overlap and thus increasing the bubble's volume. Figure 6 shows the effect of this volume correction. The maximum relative error is 25 percent after volume correction in contrast to a 75 percent error without it. Smaller time steps and multiple iterations can be used to reduce this error further, if necessary. Another potential method to perform volume correction is to introduce it as a force; however, this approach is complicated by necessity of computing the volume derivatives.

In order to know how much volume needs to be compensated for each bubble, we store its initial volume V_i^0. At each time step, we calculate the volume V_i of its Voronoi cell in the space-filling diagram, using the method proposed by Cazals et al. [2011]. We then multiply the current bubble radius r_i by a factor of $1 + \gamma((V_i^0 / V_i)^{-1/3} - 1)\Delta t$, where γ is a parameter to control the correction amount. Since we do not consider bubble surface deformation caused by liquid, we do not differentiate liquid bubbles, surface bubbles, and air bubbles in volume calculation. However, bubbles attached to a solid boundary have their volumes occupied by a solid, so we compensate their calculated volume V_i by a factor of 2. Fig-
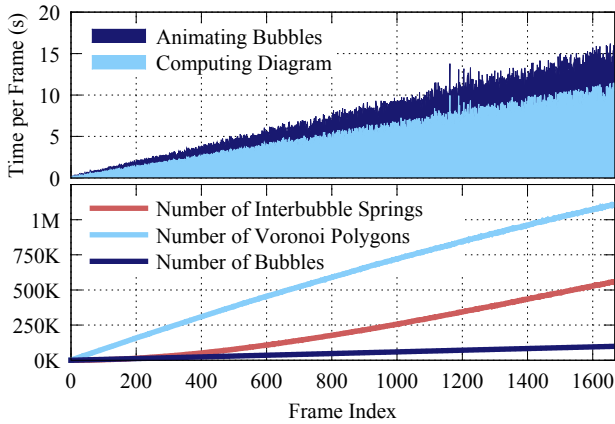
Figure 7: *Timing results and scene complexity over 1700 simulation frames in the Coke foam example for 100K bubbles.*

ure 6 plots out maximum relative volume errors over 100 simulated frames, with and without using our volume correction method.

5.3 Bursting and Coalescing

Liquid drainage may cause foam topology to change over time. In particular, bubbles on the boundary of a foam cluster may burst, while bubbles inside the foam may merge with other bubbles. To model this process, we maintain an age variable e_i for each bubble. At each time step, we randomly remove bubbles based on their ages and radii. The probability for bubble i to get removed is defined as:

$$P_i = 1 - e^{-\tau r_i e_i}, \tag{11}$$

in which τ is the bursting speed coefficient. Intuitively, Equation 11 means younger and smaller bubbles have better chances to survive, while older and larger bubbles are likely to disappear. When deleting a bubble inside of a foam, we distribute its volume to its closest neighbor and we set the neighbor's position as the weighted average of their original positions before deletion.

6 Results

The supplementary video submitted along with this paper demonstrates our animation results. We implemented our foam simulation model and tested it on an Intel Core i5-2500K 3.3GHz CPU. To demonstrate interaction with liquid flows and solid surfaces, we coupled the foam model with a particle level set fluid simulator [Enright et al. 2002], running on a 64^3 uniform Cartesian grid. We used adaptive time steps in bubble simulation, which can vary from 0.02s to 0.005s, depending on fluid flows and bubble velocities. Our system uses standard units. The gravity acceleration is 9.8m/s^2, and the liquid density is 10^3kg/m^3. A list of parameters and their ranges is summarized in Table 1 for convenience.

We use the Coke foam example to study the complexity and computational cost of our system as Figure 1 and Figure 7 show. In this example, 100K bubbles are added into the scene, and >500K bubble-bubble springs are created to handle strong interaction forces. Bubble sizes vary from 0.15mm to 3mm. The total number of polygons in the weighted Voronoi diagram is more than 1M. This example shows that the computational cost increases with the scene complexity as expected. The simulation time spent on animating bubbles is approximately one third of the time spent on constructing the Voronoi diagram in each time step. For 100K bubbles, our system simulates each rendered frame in 13 to 17 seconds (excluding fluid simulation time). It may seem that a dynamic Voronoi diagram
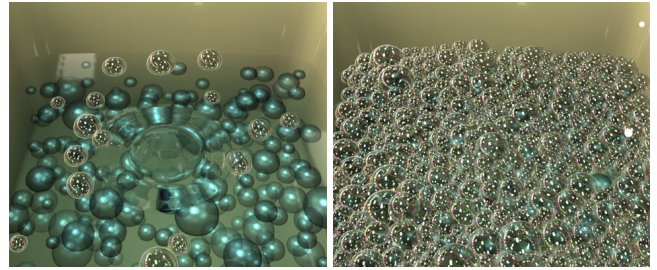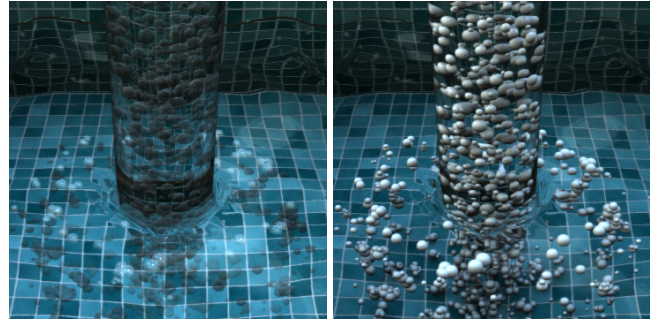

Figure 8: *Soap foam. Soap bubbles pile up in a water container.*

(a) Transparent bubbles   (b) Opaque bubbles

Figure 9: *Pouring water. Water bubbles emerge on the liquid surface, but quickly burst due to their instability in the real world. We model this effect by using a larger bursting speed coefficient.*

construction would be more efficient, but a good practical solution for updating a 3D Voronoi diagram dynamically is still elusive unfortunately. In practice, it may suffice to compute a local Voronoi diagram for a sufficiently large neighborhood of each bubble.

Our animations were rendered at 30Hz, using the GPU-based micro-polygon ray tracing method [Hou and Zhou 2011]. We modeled bubbles as semi-transparent objects with thin films separately from the scene, in order to avoid a large number of reflection and refraction rays if doing ray tracing on bubbles directly. We then incorporated transparent appearances of bubbles into the scene, by modifying corresponding rays. To speed up the rendering process, we also maintained a separate bounding volume hierarchy for bubbles, since their intersections with rays should be determined first.

Soap Foam. Bubbles in a soapy liquid made by dish detergent or hand soap tend to pile up, creating dense and complicated foam structures. In this example, we created such a scene with up to 4K bubbles, as Figure 8 shows. The bubble size varies from 0.25 to 1cm, and the water container is 16cm×16cm×16cm. At the end of the animation, bubbles can form at least ten layers in the foam. To prevent foam piles from sudden motion, we reduced the bursting coefficient τ, especially for those close to solid walls. We also varied the wetness coefficient λ according to the distance from the liquid surface, so bubbles on the top are drier than liquid surface bubbles. Each frame of this example took 0.6 to 2.0s to simulate.

Coke Foam. We simulated another Coke foam example that contains up to 16K bubbles with varying sizes, as shown in Figure 1 and 10. The largest bubble in this example has a size of 1.25mm, while the smallest bubble is 0.375mm big. It shows how the bubble size can affect bubble behaviors, such as bubble-bubble attraction and bursting. We use the same wetness coefficient $\lambda = 0.4$ for all bubbles, since they do not pile up. For testing purpose, we also animate the same scene by only using strong interaction forces for

Notation	Name	Usage	Unit	Range
λ	Wetness coeff.	To determine the resting distance between two bubbles (in Equation 1)	1	$[0, 1]$
k	Stiffness coeff.	To model the strong interaction force (in Equation 2)	N/m	$[1.0, 8.0]$
c_{drag}	Drag coeff.	To model the drag force for a liquid bubble (in Equation 3)	kg/m^3	$[0.05, 0.5]$
σ_{lad}	Liquid adhesion coeff.	To model the liquid adhesion force for a surface bubble (in Equation 5)	N/m	$[10.0, 20.0]$
α	Bubble attraction coeff.	To model the weak interaction force caused by tension (in Equation 6)	$kg \cdot m/s^2$	$[0.2, 0.6]$
σ_{sad}	Solid adhesion coeff.	To model the solid adhesion force for any bubble (in Equation 7)	N/m	$[15.0, 30.0]$
β	Solid attraction coeff.	To model the solid attraction force caused by solid (in Equation 8)	$kg \cdot m^2/s^2$	$[2.0, 6.0]$
$c_{\text{vis}}, c_{\text{lap}}$	Damping coeff.	To dissipate the kinetic energy over time (in Equation 10)	1	$[10^{-5}, 10^{-4}]$
γ	Volume correction coeff.	To determine the amount of volume compensation in each time step	1	1
τ	Bursting speed coeff.	To specify how fast bubbles burst (in Equation 11)	$m^{-1}s^{-1}$	$[0.0125, 0.2]$

Table 1: *Parameters used in our system.*

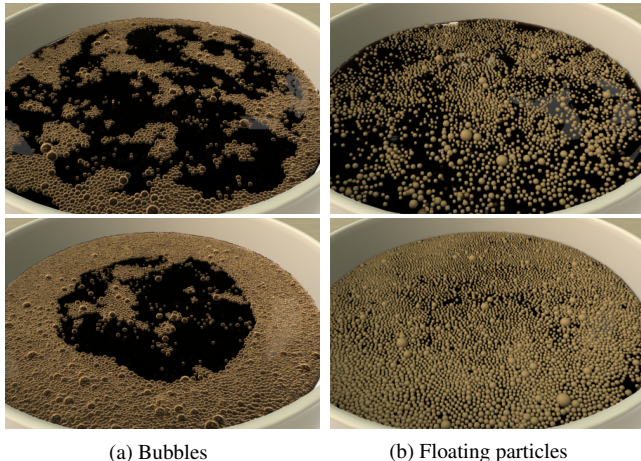(a) Bubbles   (b) Floating particles

Figure 10: *Coke foam. Images in (a) show that with our interaction forces, coke bubbles form clusters and are attracted to the mug walls. With interaction forces limited to basic collision responses, we can also simulate them as floating plastic particles in (b).*

collision responses. This allows us to simulate bubbles as if they were floating plastic particles. The simulation time of each frame varies from 0.5 to 2.8s.

Pouring Water. Our system is also able to simulate bubbles in a complex liquid scene, such as the pouring water example shown in Figure 9. There were 3.4K bubbles (with sizes of 0.75mm to 2.5mm) generated from the escaped air particles in the particle level set fluid simulator. Bubbles traveled with the water flow and emerged on the liquid surface due to the buoyant force. We increased the bursting speed coefficient τ in this example to account for the instability of water bubbles in the real world. Each frame in this example took 0.3 to 5.2s to simulate.

7 Limitations

Like other particle-based approaches, our method does not handle bubble surface deformation and it is not suitable for large bubbles (whose diameters are greater than 1cm). For numerical stability, we formulate bubble interaction forces using a linear stiffness model. However, interaction forces in the real world can be highly nonlinear, and they can be affected by many other conditions that our system does not model so far, such as drainage and coarsening effects. When dealing with bubble-liquid interactions, we do not consider how bubble motions can affect the liquid flow. Accurately preserving volumes in our system requires a smaller time step, which increases the computational cost. Finally, our implicit integrator is not unconditionally stable due to its dependency on bubble connectivity, although it barely affects the system performance.

8 Conclusion and Future Work

In this paper, we showed that a weighted Voronoi diagram can be used to approximate the actual foam geometry in a sound manner, for both dry and wet foams containing small bubbles. Based on this representation, we demonstrated that a particle-based system can efficiently and realistically handle bubble interactions, even without explicitly modeling surface deformation or surface tension. Our experiment further tested its compatibility with existing liquid simulators, and revealed its capability of generating natural bubble effects, such as clustering, stacking, bursting, coalescing, and bubble-liquid and build-solid interactions.

Besides solving those limitations listed in Section 7, we plan to accelerate our system by the use of GPU-based algorithms. Since we define most parameters in our system based on effects rather than physics, finding optimal values for them becomes a challenging and time-consuming task in practice. We are interested in carrying out both experimental and numerical study on this issue in the future.

Acknowledgments

References

ADAMS, B., PAULY, M., KEISER, R., AND GUIBAS, L. J. 2007. Adaptively sampled particle fluids. *ACM Transactions on Graphics (SIGGRAPH) 26* (July).

AURENHAMMER, F. 1987. Power diagrams: properties, algorithms and applications. *SIAM Journal on Computing 16*, 78–96.

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *Proc. of SIGGRAPH '98*, E. Fiume, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 43–54.

BARGTEIL, A. W., GOKTEKIN, T. G., O'BRIEN, J. F., AND STRAIN, J. A. 2006. A semi-Lagrangian contouring method for fluid simulation. *ACM Transactions on Graphics 25* (January), 19–38.

BARGTEIL, A. W., WOJTAN, C., HODGINS, J. K., AND TURK, G. 2007. A finite element method for animating large viscoplastic flow. *ACM Transactions on Graphics (SIGGRAPH) 26* (July).

BRAKKE, K. A. 1992. The Surface Evolver. *Experimental Mathematics 1*, 2, 141–165.

BROCHU, T., BATTY, C., AND BRIDSON, R. 2010. Matching fluid simulation elements to surface geometry and topology. *ACM Transactions on Graphics (SIGGRAPH) 29* (July), 47:1–47:9.

CAZALS, F., KANHERE, H., AND LORIOT, S. 2011. Computing the volume of a union of balls: a certified algorithm. *INRIA 2009 no 7013 Technical report*, August, 1–19.

CHENTANEZ, N., AND MÜLLER, M. 2011. Real-time Eulerian water simulation using a restricted tall cell grid. *ACM Transactions on Graphics (SIGGRAPH) 30* (Aug.), 82:1–82:10.

CHOI, K.-J., AND KO, H.-S. 2002. Stable but responsive cloth. *ACM Transactions on Graphics (SIGGRAPH) 21* (July), 604–611.

CLEARY, P. W., PYO, S. H., PRAKASH, M., AND KOO, B. K. 2007. Bubbling and frothing liquids. *ACM Transactions on Graphics (SIGGRAPH) 26* (July).

DURIAN, D. J. 1995. Foam mechanics at the bubble scale. *Physical Review Letters 75* (Dec), 4780–4783.

DURIKOVIC, R. 2001. Animation of soap bubble dynamics, cluster formation and collision. *Computer Graphics Forum 20*, 3, 67–.

EDELSBRUNNER, H. 2001. *Geometry and Topology for Mesh Generation*. Cambridge University Press, England.

ENRIGHT, D., MARSCHNER, S., AND FEDKIW, R. 2002. Animation and rendering of complex water surfaces. *ACM Transactions on Graphics (SIGGRAPH 2002) 21* (July), 736–744.

FOSTER, N., AND METAXAS, D. 1996. Realistic animation of liquids. *Graphical Models and Image Processing 58* (September), 471–.

GARDINER, B., DLUGOGORSKI, B., AND JAMESON, G. 2000. The steady shear of three-dimensional wet polydisperse foams. *Journal of Non-Newtonian Fluid Mechanics 92*, 151–166.

GREENWOOD, S. T., AND HOUSE, D. H. 2004. Better with bubbles: enhancing the visual realism of simulated fluid. In *Proc. of SCA*.

HERZHAFTA, B., KAKADJIANB, S., AND MOANC, M. 2005. Measurement and modeling of the flow behavior of aqueous foams using a recirculating pipe rheometer. *Colloids and Surfaces A: Physicochem. Eng. Aspects 263*, 153–164.

HONG, J.-M., AND KIM, C.-H. 2003. Animation of bubbles in liquid. *Computer Graphics Forum 22*, 3, 253–262.

HONG, J.-M., LEE, H.-Y., YOON, J.-C., AND KIM, C.-H. 2008. Bubbles alive. *ACM Transactions on Graphics 27* (August), 48:1–48:4.

HOU, Q., AND ZHOU, K. 2011. A shading reuse method for efficient micropolygon ray tracing. *ACM Transactions on Graphics (SIGGRAPH Asia) 30* (Dec.), 151:1–151:8.

KELAGER, M. 2009. Vertex-based simulation of dry foam. In *Master Thesis, University of Copenhagen, Denmark*.

KIM, B., LIU, Y., LLAMAS, I., JIAO, X., AND ROSSIGNAC, J. 2007. Simulation of bubbles in foam with the volume control method. *ACM Transactions on Graphics (SIGGRAH) 26* (July).

KIM, Y., SEOL, Y., LAI, M.-C., AND PESKIN, C. S. 2011. The immersed boundary method for two-dimensional foam with topological changes. *Communications in Computational Physics*.

KRAYNIK, A. M., REINELT, D. A., AND VAN SWOL, F. 2003. Structure of random monodisperse foam. *Physical Review E 67*.

KRAYNIK, A. M., REINELT, D. A., AND VAN SWOL, F. 2004. Structure of random foam. *Physical Review Letters 93*, 20.

KÜCK, H., VOGELGSANG, C., AND GREINER, G. 2002. Simulation and rendering of liquid foams. In *Proc. of Graphics Interface*, 81–88.

LOSASSO, F., GIBOU, F., AND FEDKIW, R. 2004. Simulating water and smoke with an octree data structure. *ACM Transactions on Graphics (SIGGRAPH) 23* (Aug.), 457–462.

LOSASSO, F., TALTON, J., KWATRA, N., AND FEDKIW, R. 2008. Two-way coupled SPH and particle level set fluid simulation. *IEEE Transactions on Visualization and Computer Graphics 14* (July).

MIHALEF, V., UNLUSU, B., METAXAS, D., SUSSMAN, M., AND HUSSAINI, M. Y. 2006. Physics based boiling simulation. In *Proc. of SCA*.

MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. In *Proc. of SCA*.

PIAZZA, L., GIGLI, J., AND BULBARELLO, A. 2008. Interfacial rheology study of espresso coffee foam structure and properties. *Journal of Food Engineering 1984*, 3, 420–429.

REDENBACH, C., SHKLYAR, I., AND ANDR, H. 2012. Laguerre tessellations for elastic stiffness simulations of closed foams with strongly varying cell sizes. *International Journal of Engineering Science 50*, 1, 70 – 78.

SIN, F., BARGTEIL, A. W., AND HODGINS, J. K. 2009. A point-based method for animating incompressible flow. In *Proc. of SCA*, ACM, New York, NY, USA, SCA '09, 247–255.

STAM, J. 1999. Stable fluids. In *Proc. of SIGGRAPH '99*, Computer Graphics Proceedings, Annual Conference Series, 121–128.

SULLIVAN, J. M. 1998. The geometry of bubbles and foams. *Foams and Emulsions (NATO ASI volume E) 354*, 379–402.

TAYLOR, J. E. 1976. The structure of singularities in soap-bubble-like and soap-film-like minimal surfaces. *Annals of Mathematics 103*, 3 (May), 489–539.

THÜREY, N., SADLO, F., SCHIRM, S., MÜLLER-FISCHER, M., AND GROSS, M. 2007. Real-time simulations of bubbles and foam within a shallow water framework. In *Proc. of SCA*, 191–198.

THÜREY, N., WOJTAN, C., GROSS, M., AND TURK, G. 2010. A multi-scale approach to mesh-based surface tension flows. *ACM Transactions on Graphics (SIGGRAPH) 29* (July), 48:1–48:10.

WANG, H., MILLER, G., AND TURK, G. 2007. Solving general shallow wave equations on surfaces. In *Proc. of SCA*, 229–238.

WEAIRE, D., AND HUTZLER, S. 2001. *The Physics of Foams*. Oxford University Press.

WEAIRE, D., PITTET, N., HUTZLER, S., AND PARDAL, D. 1993. Steady-state drainage of an aqueous foam. *Physical Review Letters 71*, 6, 2670–2673.

WICKE, M., RITCHIE, D., KLINGNER, B. M., BURKE, S., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2010. Dynamic local remeshing for elastoplastic simulation. *ACM Transactions on Graphics (SIGGRAPH) 29* (July), 49:1–49:11.

ZHENG, W., YONG, J.-H., AND PAUL, J.-C. 2006. Simulation of bubbles. In *Proc. of SCA*, 325–333.