Simulation and Rendering of Liquid Foams

Hendrik Kück[a] Christian Vogelgsang[b] Günther Greiner[b]

[a] Department of Computer Science
University of British Columbia
Vancouver, Canada
kueck@cs.ubc.ca

[b] Institut für Informatik
Friedrich Alexander Universität Erlangen-Nürnberg
Erlangen, Germany
{vogelgsang,greiner}@cs.fau.de

Abstract

In this paper we present a technique for simulating and rendering liquid foams. We are aiming at a functional realism that allows our simulation to be consistent with the physical effects in real liquid foam while avoiding the prohibitive computational cost of a physically accurate simulation. To this end, we have to recreate two important attributes of foam. The dynamic behaviour of the simulated foam must be based on the physics of real foam, and the characteristic interior structures of foam and their optical properties must be reproduced. We tackle these requirements by introducing a two part hybrid rendering approach. The first stage is geometric and determines the dynamic behaviour of the foam by simulating structural forces on a set of spheres, which represent the foam bubbles. In the second stage we render these spheres using a special surface shader that implicitly reconstructs the foam surfaces and performs the shading calculations. This two step approach allows us to easily integrate our technique into existing ray-tracing systems. We include images of an example animation to demonstrate the visual quality.

Key words: liquid foam, simulation, rendering, natural phenomena, shading techniques

1 Introduction

The simulation of natural phenomena is an important and fascinating research field in computer graphics. Convincing simulations and renderings of these phenomena add greatly to the realism of computer generated imagery.

Liquid foams are natural phenomena that are prominent in everyday life. On most carbonated beverages, for example, there will be at least a small amount of froth. While techniques have been developed to simulate and render liquids and their dynamics very realistically, the addition of foam would significantly increase the realism of animations.

Another point motivating the need for realistic computer generated foam is that foam is quite difficult to handle, as it tends to disintegrate over time. Also, its internal structure changes due to effects such as film rupture and diffusion. Such changes alter the visual appearance of the foam. Thus, filming scenes featuring foam is a difficult task and using computer animations is an effective alternative. Due to the complex dynamical and optical properties of foams, it is very hard or even impossible to achieve sufficiently believable animations of foam at various scales using existing modelling and rendering techniques such as particle systems or texturing. Thus, a specialized technique for simulation and rendering of foam is needed.

The rest of the paper is structured as follows. The next section describes previous work in the field of foam simulation and rendering. Section 3 gives an overview of the physical and geometric properties of liquid foams. In Section 4 we describe our model for the simulation of foam structure and dynamics. Our rendering method is explained in Section 5. Section 6 contains implementation details and presents some results. We end with concluding remarks and ideas for future work.

2 Previous Work

Soap films and bubbles have fascinated children and researchers from many different disciplines for centuries. While children are fascinated by the fragility and beauty of the iridescent soap bubbles, mathematician's interest in soap film structures is mainly due to the surface minimizing property of these films [11]. Although this theoretic research on minimal surfaces has produced various insights in the geometry of foams [23, 2], many problems still remain unsolved [24].

Analytical solutions for the geometry of soap bubble clusters exist for clusters of up to three bubbles. The liquid films are spherical in these cases. Glassner models such clusters using CSG operations and renders them using a special shader that takes Fresnel reflection and interference effects at the soap films into account [13].

As the geometry of larger bubble clusters cannot be computed analytically, numerical methods have to be used. The Surface Evolver program by Brakke evolves

triangular meshes towards a state of minimal energy, taking surface tension and other forces into account [5]. It can be used to numerically compute the geometry of bubble clusters in equilibrium [19]. However, due to its computational expense, this approach is not feasible for large numbers of bubbles. Almgren and Sullivan [1] render bubble cluster geometries produced by Surface Evolver simulations using a shading model very similar to the one used by Glassner.

Icart and Arquès restrict their simulation of soap froth to two-dimensional foam, i.e. bubbles in a common plane [14]. They use spherical and planar films to approximate the foam structure. As for Almgren, Sullivan and Glassner, the main focus of their work is on the simulation of the interference effects at the liquid films.

In contrast, Durikovic does not address optical properties at all and focuses instead on the simulation of the geometry and the dynamic behaviour of soap bubbles [10]. He uses particles connected by springs to approximate the bubble surface and its deformation in response to forces caused by wind or contact with other bubbles or objects.

None of the published research dealing with the rendering of soap bubbles or foam takes the junctions of the liquid films into account, although these are of great importance for the visual appearance of foams.

Physicists have been studying foams for a long time to investigate their dynamic properties. Although there is a great deal of published research in soft condensed matter physics on many aspects of foams [26, 3, 15, 25, 21], foam dynamics is still far from completely understood. Due to the instability of liquid foam, many foam experiments tend to be very difficult to perform. This is one reason why physicists developed and used computer simulations of foam structure and dynamics. Most of the approaches restrict themselves to the simulation of two-dimensional foams. This reduction in dimensionality significantly decreases the complexity of the problem and most insights gained from these 2D simulations can be transferred to the 3D case. There are two different types of models for the foam structure that researchers have used in their simulations. Kermode and Bolton both developed computer programs for the simulation of two-dimensional foam. They model the foam as a network of curved liquid films. Kermode's simulation program 2D-Froth (described in [26]) assumes the films to be infinitely thin circular segments joined at vertices. For given pressures in the enclosed cells, the structure is equilibrated by moving the vertices. Bolton [4] extends this model by allowing the junctions of films to have spatial extents, which allows for the simulation of foam with higher liquid fractions.

The other type of model, used by Durian [7, 8] represents the foam as a collection of interacting gas bubbles. In comparison with network based approaches, this greatly simplifies the simulation of the dynamic behaviour of foam since one does not have to deal with topological changes as bubbles move.

3 Foam Physics

3.1 Foam Structure

Liquid foams are two phase systems consisting of a liquid enclosing bubbles of gas. Depending on the liquid content, the structure of the foam can vary greatly. In an extremely wet foam, the gas bubbles are spherical and separated by large amounts of liquid. At the other extreme, a very dry foam consists of extremely thin films of liquid separating the gas bubbles. Foams encountered in everyday life such as beer foam or dish washing foam are quite dry, featuring liquid films with thickness ranging from a few to several tens of microns.

Where the films meet, small tubes with a triangular cross-section are formed (see Figure 3). Known as *Plateau borders*, these tubes are where most of the water in a foam is contained. They are responsible for some of the characteristic visual properties of liquid foams.

The surface tension in liquids causes contracting forces along the surface of the liquid. As a result, liquids always try to minimize their surface area. The reason why a bubble remains stable and does not collapse into a drop of liquid is that there is an excess pressure inside the bubble. This produces forces acting on the bubble surface that exactly cancel the forces owing to the surface tension.

3.2 The Laplace-Young Law

Laplace and Young derived an equation relating the radius of curvature R of a liquid film with surface tension γ to the pressure difference Δp between the gas cells it separates [3]:

$$R = 4\frac{\gamma}{\Delta p}.$$

This law can be used to compute the radius of individual soap bubbles, as well as the radius of curvature of films separating two gas cells. According to this law, such a separating liquid film will be curved toward the cell with the smaller pressure. It also follows from this law that smaller bubbles have higher pressure than larger bubbles and thus the separating film between two different size bubbles will be curved towards the larger bubble.

3.3 Law of Plateau

In the 19th century, Plateau performed experiments with soap films and frameworks [20]. He experimentally established rules about the geometric properties of soap films that were later theoretically proven. These rules

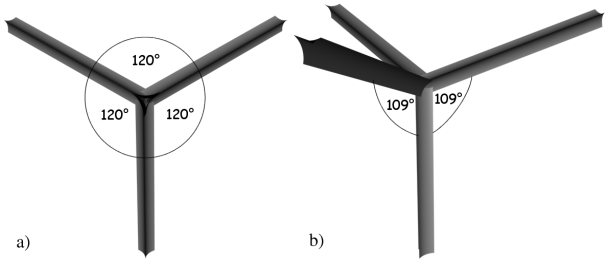

Figure 1: Angles formed by films and Plateau borders

state that the liquid films in a foam always meet in groups of three. At these junctions (the Plateau borders) the films always form angles of $120°$ (see Figure 1a). The Plateau borders themselves meet in groups of four in the tetrahedral angle of $109.5°$ (Figure 1b).

3.4 Dynamic Effects in Liquid Foams

The structure of foam changes over time for several reasons. Gravity exerts forces on the liquid, resulting in a drainage of the foam. Thus, the films get thinner over time, and the probability of film rupture increases. As more and more films break, the foam gets coarser and finally disintegrates. Additionally, in the case of froth on a liquid, new bubbles could rise from the liquid and add to the foam from below.

3.5 Optical Effects

Liquid foam has fascinating optical properties. For example, it is intriguing, that a transparent liquid can form a bright white foam that seems to be completely opaque. These properties are caused by the effects of refraction and reflection at the liquid/gas boundaries of the complex foam structures. In this section we take a closer look at these interactions of light with the foam structures.

If we follow a ray of light into the foam, we have to distinguish between collision with a film and a Plateau border. As the liquid films are extremely thin, their two

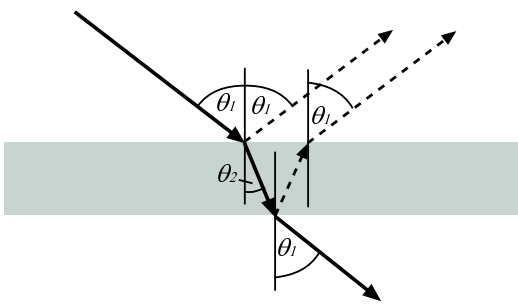

Figure 2: Light ray refracted and reflected at liquid film

boundary surfaces are almost perfectly parallel to each other. As illustrated in Figure 2, a ray of light passing through a film is refracted according to Snell's law at the first boundary. However, because the boundaries are parallel, the change in direction will be reversed at the second interface. Thus the ray leaves in the same direction as it entered but with a small positional offset.

Part of the light is reflected at the interfaces. The fraction of the light that is reflected is given by the Fresnel reflection coefficient [12], a value which depends on the indices of refraction of liquid and gas and also on the incident angle of the light ray. A fraction of the light entering the liquid will also be reflected inside the liquid one or more times. This results in light leaving the liquid in the same direction as the directly reflected light (see figure 2) but with a phase shift due to the different path lengths and the additional phase shift occurring when the light enters the liquid. Due to the interference of the light waves taking these different paths, some wavelengths are reinforced, while others are cancelled [6]. If the thickness of the films is sufficiently close to the wavelength of visible light (0.5 to about 2 microns), these interferences cause only a few maxima and minima in the visible spectrum and thus produce the characteristic iridescent colours of soap bubbles. However, for all but extremely dry foams, the liquid films are thicker and thus do not exhibit visible interference effects [3].

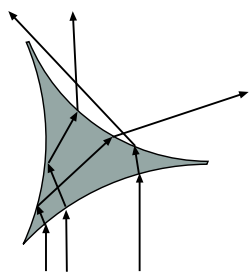

Figure 3: Refraction and total refraction at Plateau border

While the effect of refraction on light rays passing through liquid films is negligible, the effect is significant at Plateau borders. Due to refraction and total reflection at the interfaces, the direction of the rays leaving the Plateau border can vary greatly for the same incident angle and only a small positional offset (see Figure 3). Due to these refraction effects and the additional Fresnel reflections at films and Plateau borders, the light is scattered inside the foam and its direction is randomized. To what extent the direction of light entering the foam is randomized, depends on the 'optical thickness' of the foam and on the distance that the light travels through the foam. The optical thickness determines the frequency of scattering events in a medium and it can be characterized in terms of the transport mean free path, i.e. the average distance a photon will travel before it is scattered into a random direction [16]. Measurements show that in the

case of liquid foams this value is proportional to the average bubble diameter [9]. The randomization of light directions due to multiple scattering results in the nearly uniform white appearance of dense liquid foams.

4 Simulation

The model we use for the simulation is very similar to the one proposed by Durian [7]. However, there are important differences. Durian's model is restricted to the simulation of two-dimensional foams in the wet regime. We extended it to three dimensions and also generalized it to simulate foams with arbitrary liquid content by allowing attractive forces between bubbles. The only external influence on the simulated foam in Durian's work is exerted by moving the topmost bubbles with constant velocity. We allow for interaction of the simulated foam with the environment by taking forces due to contact with external objects into account.

During the simulation, we do not compute the exact geometry of the foam structure or how the bubbles are deformed as they come into contact with other bubbles or external objects. Instead, we represent the bubbles by spheres with fixed radii and move them according to assumed forces.

4.1 Bubble - Bubble Forces

If we take a look at two bubbles in an extremely wet foam, we see that the bubbles are spherical gas cells surrounded by water. When these bubbles touch each other, they deform into non-spherical shapes. As a sphere has minimal surface area, this deformation increases the surface of the bubbles and thus gives rise to a repulsive force due to surface tension. According to Durian [7], these forces are close to harmonic and can thus be approximated by spring forces between touching bubbles.

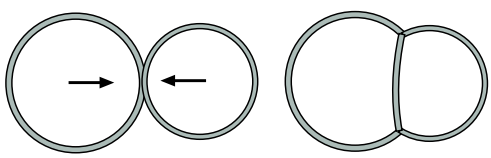

Figure 4: Attracting force acting on touching bubbles

The situation is more complicated in a dry foam. If two bubbles surrounded by air touch each other as shown in Figure 4 they can share the separating film and thus decrease the overall surface by moving closer together. This causes an attractive force which drags the bubbles toward each other. The surface area is minimized when the Plateau condition is met, i.e. when the films meet at 120° angles as on the right side of Figure 4.

We use a combination of attractive and repulsive spring forces acting on touching bubbles to model this behaviour. Ideally, these forces will cancel each other exactly when the Plateau conditions are met.

The repulsive force $\mathbf{F}^r_{ij}(= -\mathbf{F}^r_{ji})$ acting on bubble i due to contact with bubble j in our model is given by

$$\mathbf{F}^r_{ij} = k_r \left(\frac{1}{||\mathbf{p}_i - \mathbf{p}_j||} - \frac{1}{rad_i + rad_j} \right) (\mathbf{p}_i - \mathbf{p}_j),$$

where k_r is a user-defined coefficient that models the surface tension in the liquid. $\mathbf{p}_i$ and $\mathbf{p}_j$ are the positions of the sphere centres for the spheres representing bubbles i and j and rad_i and rad_j are the radii.

The attractive force between two bubbles depends on the surface area that these bubbles can share when moving closer together. Obviously, if additional other bubbles touch both bubbles, the possible common surface area will be smaller, resulting in a smaller pairwise attractive force. To approximate this, we take the number of neighbours of the spheres into account when computing the attractive force $\mathbf{F}^a_{ij}$ acting on bubble i due to contact with bubble j. More specifically,

$$\begin{aligned} \mathbf{F}^a_{ij} &= k_a \, c_{nb} \, c_{dist} \, \frac{\mathbf{p}_j - \mathbf{p}_i}{\|\mathbf{p}_j - \mathbf{p}_i\|} \\ \text{with} & \\ c_{nb} &= \frac{\frac{1}{|NB_i|} + \frac{1}{|NB_j|}}{2} \\ c_{dist} &= \frac{\|\mathbf{p}_j - \mathbf{p}_i\| - max(rad_i, rad_j)}{min(rad_i, rad_j)}. \end{aligned}$$

Again, k_a is a user-defined coefficient. The ratio $\frac{k_a}{k_r}$ models the liquid content (i.e. the wetness) of the simulated foam. $|NB_i|$ is the cardinality of the set of neighbours of sphere i, i.e. all the spheres that are overlapping with that sphere in the current configuration.

For gas bubbles to change their position relative to each other, liquid must flow through the films and Plateau borders. As the bubbles move with respect to their neighbours, they experience a force opposed to this movement due to viscosity of the liquid. We assume the force F^v_i to be proportional to the velocity at which bubble i moves relative to the bubbles touching it. That is,

$$\begin{aligned} \mathbf{F}^v_i &= k_v(\overline{\mathbf{v}_i} - \mathbf{v}_i) \\ &= k_v \left(\left[\sum_{j \in NB_i} \frac{\mathbf{v}_j}{|NB_i|} \right] - \mathbf{v}_i \right), \end{aligned}$$

where k_v is a user-defined parameter modelling the viscosity of the liquid, $\mathbf{v}_i$ is the velocity of bubble i and $\overline{\mathbf{v}_i}$ is the mean velocity of the bubbles in contact with i, i.e. the bubbles in NB_i.

4.2 Object - Bubble Forces

We want the foam to react to objects in the scene. We must therefore compute the forces acting on the bubbles due to contact with those objects. The forces we take into account here are very similar to the ones between bubbles. Again, we assume attractive and repulsive spring forces for bubbles touching the objects. Additionally, we also consider forces due to friction at the object surface:

$$\mathbf{F}_i^{of} = k_{of}(\overline{\mathbf{v}^o}_i - \mathbf{v}_i), \tag{1}$$

where $\overline{\mathbf{v}^o}_i$ is the mean velocity of the scene objects touching bubble i and k_{of} is the friction coefficient, which can be adjusted by the user on a per-object basis.

4.3 Other Forces

An additional constant force $\mathbf{F}_g$ acts on all forces and models the effect of gravity. Another frictional force in our model is the force $\mathbf{F}_i^{air}$ acting on bubble i due to air resistance. We assume the air to be static, so that this force is proportional to the velocity of the bubble:

$$\mathbf{F}_i^{air} = -k_{air}\,\mathbf{v}_i,$$

where the user-defined parameter k_{air} models the air resistance.

4.4 Deriving an ODE System

The total force $\mathbf{F}_i^{total}$ acting on bubble i is given by

$$\begin{aligned}
\mathbf{F}_i^{total} &= \mathbf{F}_i^{ra} + \mathbf{F}_i^{fr} + \mathbf{F}^g, \\
\text{with} & \\
\mathbf{F}_i^{ra} &= \sum_{j \in NB_i} (\mathbf{F}_{ij}^{r} + \mathbf{F}_{ij}^{a}) \; + \sum_{k \in NO_i} (\mathbf{F}_{ik}^{or} + \mathbf{F}_{ik}^{oa}); \\
\mathbf{F}_i^{fr} &= \mathbf{F}_i^{v} + \mathbf{F}_i^{of} + \mathbf{F}_i^{air},
\end{aligned}$$

where $\mathbf{F}_i^{ra}$ represents all the repulsive and attractive forces acting on bubble i due to contact with adjacent bubbles (NB_i) and objects (NO_i). $\mathbf{F}_i^{fr}$ represents the frictional forces acting on bubble i due to movement relative to its neighbour bubbles, objects and the surrounding air.

Like Durian [7], we assume that the bubbles have no mass. A consequence of this assumption is, that the forces acting on a bubble have to sum up to $\mathbf{0}$, that is,

$$\mathbf{F}_i^{total} = \mathbf{0}.$$

Using this, we can relate the velocity of a bubble to the position and the velocities of adjacent bubbles and objects. Solving for $\mathbf{v}_i$ gives

$$\mathbf{v}_i = \frac{1}{k_v + k_{of} + k_{air}} \left(k_v \overline{\mathbf{v}}_i + k_{of} \overline{\mathbf{v}^o}_i + \mathbf{F}_i^{ra} + \mathbf{F}^g\right).$$

Since $\mathbf{v}_i = \dot{\mathbf{p}}_i$, we have a first-order ODE system. We use numerical integration to solve for the positions of the bubbles at the current animation frame. The result of this simulation is then used as start configuration for the simulation for the next frame.

Unlike most physical models involving spring forces, the springs in our model are not statically assigned, but depend instead upon the position of the bubbles. Thus, for each step of the numerical integration scheme, it has to be determined which spheres overlap with which other spheres and which external objects, as this information is needed for the computation of the forces acting on the bubbles. We exploit the temporal coherence of sphere overlaps to accelerate these computations using a scheduling approach similar to the one described in [17].

4.5 Other Dynamic Effects

Our simulation also models the rupture of liquid films and the creation of bubbles rising from a supporting liquid (or appearing from any other source) by simply removing random bubbles during the simulation and adding new random bubbles according to a set of user-defined parameters specifying the distribution of positions and radii of new bubbles as well as the rate of additions and film rupture.

4.6 Generating Geometry

Using the result of the simulation, we create simple geometry consisting of overlapping spheres with the computed centres and radii. The spheres are flattened at the intersection with scene objects. The resulting geometry is passed to the ray-tracer for rendering using the shading approach described in the next section.

5 Shading

The main purpose of our shading algorithm is to transform a set of spheres into the visual appearance of a contiguous foam structure. The existing geometry is a sphere structure like the one on the left side of Figure 5. Our shader then tries to mimic the look of real foam, as depicted on the right side.

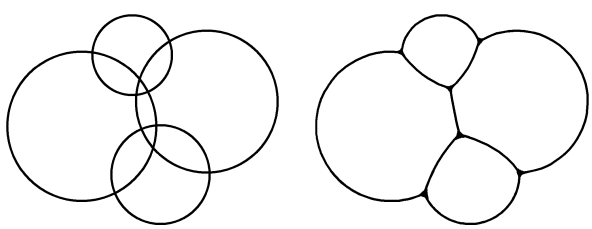

Figure 5: Sphere representation (left) and corresponding foam structure (right)

We use a special surface shader, which is called at every intersection of a viewing ray with one of the gener-

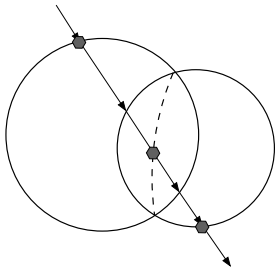

Figure 6: Ray intersecting overlapping spheres

ated spheres. For each such intersection, we must decide if there is a liquid film or Plateau border at this position in the corresponding foam structure, for which local shading has to be performed. Otherwise, no shading calculations are carried out at this intersection. In the example shown in Figure 6 we only want to perform local shading at the three marked positions. As can also be seen in this example, the surface of an assumed liquid film separating two or more bubbles does not necessarily coincide with the surfaces of the spheres. Thus, in such cases, we have to somehow approximate the intersection with this separating film from the information we get for the ray's intersections with the spheres. The basic idea is to make all these decisions depending on the order in which a ray enters and leaves the overlapping foam spheres. To do this, we associate a data structure with a ray that keeps track of which spheres the ray currently is inside, as well as the order in which it entered these spheres.

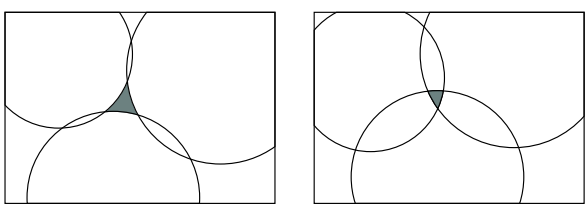

Figure 7: Two sphere configurations in which a Plateau border is assumed

Given this information at a ray/sphere intersection, we decide to assume a Plateau border in the foam structure in both cases shown in Figure 7. The grey areas indicate where Plateau borders would be assumed by our algorithm. In the left case, a ray passing through the grey area travels through a small amount of free space between spheres. We detect this case when the ray enters a sphere from free space and the distance to the last intersection with another sphere is smaller than a user-defined upper bound for the width of Plateau borders. As shown in the right case in Figure 7, we also assume a Plateau border when the ray traverses the intersection of at least three overlapping spheres. The condition to detect this case at an intersection is that the ray was in exactly three spheres and at this intersection leaves the one sphere out of the three that it entered the first. If a plateau border is detected using one of these conditions, we perform the local shading as described in Section 5.2.

If the current intersection does not meet the conditions for a Plateau border, we check if it represents an intersection with a liquid film. This will be the case if *a)* a sphere is entered from empty space or *b)* the ray leaves a sphere into empty space and neither at this nor at the next intersection along the ray are the conditions for a Plateau border met. The local shading calculations for the intersection are then performed using the shading model for liquid films described in Section 5.1. There is one other case when local shading for liquid films must be performed. It deals with the assumed liquid films separating touching bubbles such as at the second marked point in Figure 6. Such a separating liquid film is assumed when the ray was in exactly two spheres and leaves the sphere it entered first. Since the assumed separating film does not coincide

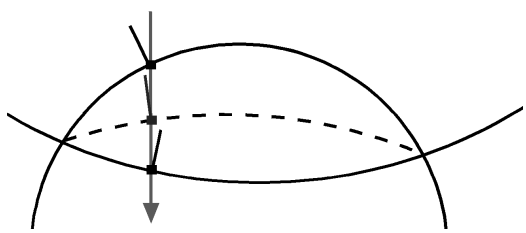

Figure 8: Approximation of intersection with separating liquid film

with the sphere surfaces, we must approximate the intersection with that film. We do this by simply averaging the positions and normal vectors of the intersections with the two overlapping spheres as shown in Figure 8. As the smaller bubble has a higher curvature, the averaged normal vectors will give the visual impression of a surface curved towards the larger bubble, which is consistent with the shape of liquid films in real foams (see Section 3.2). It has to be mentioned however, that this does not necessarily hold for grazing viewing angles. Thus, this simple and somewhat crude approximation manages to qualitatively recreate the curvature of the liquid films quite well and at low computational cost for most viewing angles.

5.1 Local Shading for Liquid Films

As described in Section 3.5 refraction does not change the direction of rays passing through the liquid films. The mentioned positional offset of the ray is so small (a few microns at most) that it can be safely ignored. So when a ray hits a liquid film, we perform the local shading and continue to follow the ray further into the scene. For the local shading of the films we use a simple shader that

approximates Fresnel reflection [12]. Currently we do not take interference effects into accounts. However, a shader that simulates these effects using the techniques described in [6, 22] could be used instead for the rendering of extremely dry foams.

5.2 Local Shading for Plateau Borders

As mentioned in section 3.5, the high curvature shapes of the Plateau borders cause incident light to be scattered in many different directions through Fresnel reflection, refraction and total reflection. In our shading approach for the Plateau border we model both the scattering of direct incident light towards the camera at a single Plateau border, as well as the diffusion of light due to multiple scattering in the foam.

We approximate the diffusion as the sum of the intensities of all light sources illuminating the foam, weighted by a user-defined parameter. This parameter represents the diffusive properties of the foam. To determine the amount of direct incident light, that is scattered towards the camera at a Plateau border, we first approximate the Plateau border's principal direction $\mathbf{d}$. We construct $\mathbf{d}$ as being maximally close to perpendicular to the normal vectors of the adjacent sphere surfaces. The direction $\mathbf{r}$ of an incoming light ray can be split up into a component parallel to $\mathbf{d}$ and a perpendicular one. We then make the simplifying assumption that Fresnel reflection, refraction and total reflection have the effect of completely randomizing the component perpendicular to $\mathbf{d}$ while leaving the parallel component almost unchanged. The only change to the parallel component is due to refraction because the Plateau border surfaces are perpendicular to $\mathbf{d}$ and thus reflections have no effect on this component. As the change that occurs due to refraction when the ray enters the liquid is counteracted by the change occurring when it exits, the assumption of a small change is valid.

The contribution of a single light source according to this model is then taken to be the intensity of that light source weighted by $cos^n\theta$, where n is a user-defined parameter and θ is the minimum angle between any light ray from this light source scattered at the Plateau border according to the assumptions stated above and the direction from the Plateau border to the camera. The cos^n distribution models the small possible change of the component parallel to $\mathbf{d}$ due to refraction.

6 Implementation

We implemented our approach as a combination of a geometry and material shader for Mental Ray from Mental Images, a commercial ray-tracing system. The implementation includes a geometry shader responsible for the simulation calculation and sphere creation. A material shader was implemented to perform the shading in the rendering stage.

In our implementation, we support two different types of objects that can interact with the simulated foam. The first object type is convex polygonal objects. The foam is restricted to the interior of these objects. The other supported shapes are spheres. They can be placed as foam obstacles. These types of objects were chosen based on the ease and efficiency of intersection tests with the foam spheres. The objects used to affect the foam need not appear in the rendering. Using invisible spheres and convex objects, arbitrary object shapes can be approximated.

6.1 Results

Figure 9 contains three frames of a simple animation. In this animation, a small amount of foam consisting of about 700 bubbles slides down a slant and around a spherical obstacle. The resulting animation is quite convincing, featuring plausible foam dynamics. The interaction with external objects greatly increase the realism.

The simulation computations for this scene took about 4 seconds per frame. The time spent in the rendering itself was 40 seconds per frame (for 800x630 images on a dual PII/400). For larger numbers of bubbles, the simulation and rendering times stay reasonable. A scene with 3000 foam bubbles resulted in a per-frame simulation time of 28 seconds and a rendering time of 90 seconds.

7 Conclusion and Future Work

We have developed a rendering technique that allows high quality foam simulation and rendering in reasonable time. The simulation stage uses a physically-based 3D simulation with foam bubbles represented by fixed-size spheres. With this approach, we avoid the explicit construction of the complex interior surfaces of foam. To achieve visually convincing results, we use a custom surface shader to approximate these structures in the rendering stage. Although not physically accurate, this combination leads to a rendering method that is consistent with the physical effects in real foam and reasonably fast for production use.

Previous work on the rendering of bubbles and liquid foams in computer graphics has mainly focused on the modelling and rendering of single bubbles or small clusters of bubbles. The other extreme, a dense foam viewed from a large distance, could be rendered using multiple scattering techniques as described in [18], thus ignoring the internal structure of the foam completely. Our work lies somewhere between these extreme cases. The compromise between physics and heuristics we take in our model works quite well for foams with several thousand bubbles, viewed from moderate distance.

A promising area for future research could be the extension of our technique to allow arbitrary numbers of

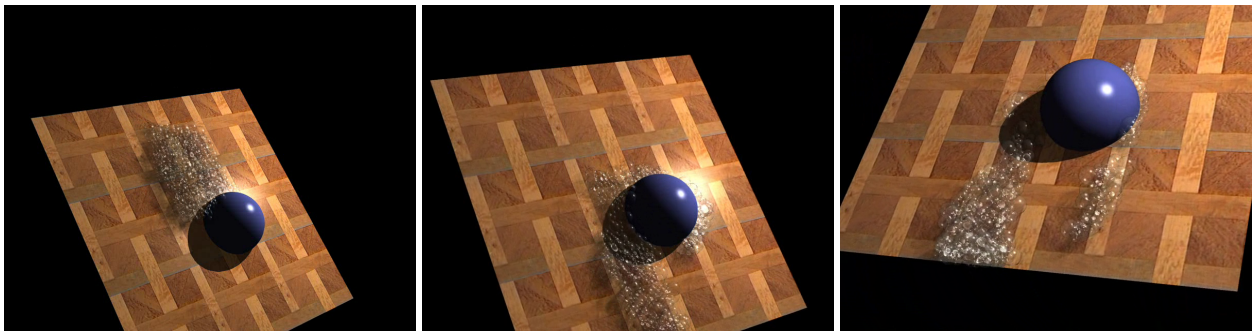

Figure 9: Foam of approximately 700 bubbles sliding past a hemisphere on an inclined plane

bubbles viewed from arbitrary distances. This could be achieved using a level of detail approach. For close-ups, the internal structure of the foam should be explicitly modelled and rendered with more suitable shading models. For dense foams, computation of bubble positions in the interior may be avoidable. At the same time, our approximation of light transport inside dense foam needs to be replaced by an actual simulation of the multiple scattering effects. Such an integrated technique would allow efficient simulation and rendering of convincing animations of a wide variety of liquid foams.

Acknowledgements

We want to thank the Animation/VFX department at SZM Studios, Munich, Germany, who sponsored this project by providing both hardware and software. Special thanks must go to Horst Hadler and Michael Kellner at SZM Studios for their support in many ways and especially for answering numerous technical questions. We would also like to thank the anonymous reviewers for their detailed and valuable comments.

References

[1] F. Almgren and J. Sullivan. Visualization of soap bubble geometries. In M. Emmer, editor, *The Visual Mind*, pages 79–83. MIT Press, 1993.

[2] F. J. Almgren, Jr. and J. E. Taylor. The geometry of soap films and soap bubbles. *Scientific American*, 235(1):82–93, 1976.

[3] J. J. Bikerman. *Foams*. Springer-Verlag, Berlin, 1973.

[4] F. Bolton. A computer program for the simulation of two-dimensional foam, 1990. www.tcd.ie/Physics/Foams/plat.html.

[5] K. Brakke. The surface evolver. *Experimental Mathematics*, 1:141–165, 1992.

[6] M. L. Dias. Ray tracing interference color. *IEEE Computer Graphics and Applications*, pages 54–60, 1991.

[7] D. J. Durian. Foam mechanics at the bubble scale. *Physical Review Letters*, 75(26):4780–4783, 1995.

[8] D. J. Durian. Bubble-scale model of foam mechanics: Melting, nonlinear behaviour and avalanches. *Physical Review*, E(55):1739–1751, 1997.

[9] D. J. Durian, D. A. Weitz, and D. J. Pine. Multiple light-scattering probes of foam structure and dynamics. *Science*, 252(5006):686–688, 1991.

[10] R. Durikovic. Animation of soap bubble dynamics, cluster formation and collision. *Computer Graphics Forum*, 20(3), 2001.

[11] M. Emmer. Soap bubbles in art and science: From the past to the future of math art. In M. Emmer, editor, *The Visual Mind*, pages 135–142. MIT Press, 1993.

[12] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer graphics - Principles and Practice*. Addison-Wesley, 2nd edition, 1990.

[13] A. Glassner. Soap bubbles. *IEEE Computer Graphics and Applications*, 20(6):99–109, 2000.

[14] I. Icart and D. Arquès. An approach to geometrical and optical simulation of soap froth. *Computers and Graphics*, 23(3):405–418, 1999.

[15] C. Isenberg. *The Science of Soap Films and Soap Bubbles*. Dover Publications, 1987.

[16] A. Ishimaru. *Wave Propagation and Scattering in Random Media*. Wiley-IEEE Press, 1999.

[17] D. J. Kim, L. J. Guibas, and S. Y. Shin. Fast collision detection among multiple moving spheres. *IEEE Transactions on Visualization and Computer Graphics*, 4(3):230–242, 1998.

[18] F. Pérez, X. Pueyo, and F. X. Sillion. Global illumination techniques for the simulation of participating media. In Julie Dorsey and Philipp Slusallek, editors, *Eurographics Rendering Workshop 1997*, pages 309–320, New York City, NY, 1997. Eurographics, Springer Wien.

[19] R. Phelan, D. Weaire, and K. Brakke. Computation of equilibrium foam structures using the surface evolver. *Experimental Mathematics*, 4:181–192, 1995.

[20] J. A. F. Plateau. *Statique Expérimentale et Théorique des Liquides soumis aux seules Forces Moléculaires*. Gauthier-Villars, Paris, 1873.

[21] R. K. Prud'homme and S. A. Khan, editors. *Foams: Theory, Measurements and Applications*. Marcel Decker, Inc., 1995.

[22] B. E. Smits and G. Meyer. Newton's Colors: Simulating Interference Phenomena in Realistic Image Synthesis. In K. Bouatouch and C. Bouville, editors, *Photorealism in Computer Graphics*, pages 185–194, 1992.

[23] J. M. Sullivan. The geometry of bubbles and foams. In J.-F. Sadoc, editor, *Foams, Emulsions, and Cellular Materials*, pages 379–402. NATO ASI, 1998.

[24] J. M. Sullivan and F. Morgan. Open problems in soap bubble geometry. *Int. J. Math.*, 7(6):833–842, 1996.

[25] D. Weaire and J. Banhart. *Foams and Films*. MIT Verlag Bremen, 1999.

[26] D. Weaire and S. Hutzler. *The Physics of Foams*. Oxford University Press, 1999.