Sounding Liquids:
Automatic Sound Synthesis from Fluid Simulation

William Moss Hengchin Yeh Jeong-Mo Hong* Ming C. Lin
Dinesh Manocha†
University of North Carolina at Chapel Hill and Dongguk University

Abstract

We present a novel approach for synthesizing liquid sounds directly from visual simulations of fluid dynamics. The sound generated by liquid is mainly due to the vibration of resonating bubbles in the medium. Our approach couples physically-based equations for bubble resonance with a real-time shallow-water fluid simulator as well as an hybrid SPH-grid-based simulator to perform automatic sound synthesis. Our system has been effectively demonstrated on several benchmarks.

1 Introduction

Auditory display provides a natural, intuitive human-computer interface for many desktop applications including video games, training systems, computer aided design, and scientific visualization. Audio interfaces can also be used to develop assistive technologies for the visually impaired. Similar to digital image synthesis, automatic sound synthesis is central to creating a compelling, realistic virtual world.

Most existing sound synthesis approaches have focused on sound generation due to colliding solid or deformable objects in air. Complementing prior work, we investigate new methods for sound synthesis in a liquid medium. Our formulation is based on prior work in physics and engineering which shows that sound is generated by the resonance of bubbles within the fluid [Rayleigh 1917]. We couple physics-based fluid simulation with the automatic generation of liquid sound based on Minneart's formula [Minnaert 1933] for spherical bubbles and spherical harmonics [Leighton 1994] for non-spherical bubbles. We also present a fast, general method for tracking the bubble formations and a simple technique to handle a large number of bubbles within a given time budget.

Our synthesis algorithm offers the following advantages: (1) it renders both liquid sounds and visual animation simultaneously using the same fluid simulator; (2) it introduces minimal computational overhead on top of the fluid simulator; (3) for fluid simulators that generates bubbles, no additional physical quantities, such as force, velocity, or pressure are required – only the geometry of bubbles; (4) for fluid simulators without bubble generation, a physically-inspired bubble generation scheme provides plausible audio; (5) it can adapt the computations to balance between costs and quality.

We also decouple sound rendering rates (44 kHz) from graphical updates (30-60 fps) by distributing the bubble events and processing per simulation step over multiple audio frames. Our sound synthesis system has been coupled with two types of fluid simulators: one based on the shallow water equations and the other using a hybrid grid-SPH method. We demonstrate the integrated system on a variety of scenarios involving liquid-liquid (see Fig. 1) and liquid-object interaction.

*jmhong@dongguk.edu
†e-mail: wmoss, hyeh, lin, dm@cs.unc.edu

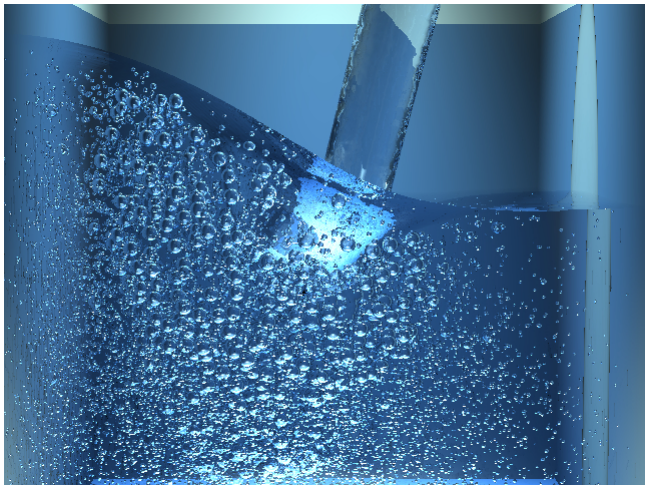

Figure 1: *Liquid sounds generated using bubble resonance automatically from visual simulations of pouring water (see the video).*

2 Related Work

There is extensive literature on fluid simulation and sound synthesis. We limit our discussion to prior work closely related to ours.

Fluid Simulation: Since the seminal works of Foster and Metaxas [1996], Stam [1999] and Foster and Fedkiw [2001] there has been tremendous interest and research on simulating fluids in computer graphics. Generally speaking, current algorithms for visual simulation of fluids can be classified into three broad categories; grid-based methods, smoothed particle hydrodynamics (SPH) and shallow-water approximations. We refer the reader to a recent survey [Bridson and Müller-Fischer 2007] for more details.

Sound Synthesis: Most of the prior work on sound synthesis in computer graphics has focused on simulating sounds from rigid and deformable bodies [O'Brien et al. 2001; van den Doel et al. 2001; O'Brien et al. 2002; Raghuvanshi and Lin 2006; James et al. 2006; Bonneel et al. 2008], the sound resulting from objects moving rapidly through air [Dobashi et al. 2003; Dobashi et al. 2004] and the sound of woodwinds and other instruments [Florens and Cadoz 1991; Scavone and Cook 1998].

Our work is inspired by van den Doel [2005] that introduced the first method for generating liquid sounds using Minneart's formula, which defines the resonant frequency of a spherical bubble in an infinite volume of water in terms of the bubble's radius. This method provides a simple technique to generate fluid sounds through manual control of the users by adjusting various parameters. Our work generalizes this approach by introducing efficient methods to handle non-spherical bubbles which occur frequently in nature. We also offer mechanisms to enable visual simulations of fluid dynamics to determine these parameters automatically, making it possible to synthesize liquid sounds directly from fluid animation. Other liquid sound synthesis methods provide limited physical basis for the generated sounds [Imura et al. 2007].

The physics literature presents extensive research on the acoustics of bubbles, dating back to the work of Lord Rayleigh [1917]. There have been many subsequent efforts, including works on bubble formation due to drop impact [Pumphrey and Elmore 1990; Prosperetti and Oguz 1993] and cavitation [Plesset and Prosperetti 1977], the acoustics of a bubble popping [Ding et al. 2007], as well as multiple works by Longuet-Higgins presenting mathematical formulations for monopole bubble oscillations [1989a; 1989b] and non-linear oscillations [1991]. T. G. Leighton's [1994] excellent text also covers the broad field of bubble acoustics and provides many of the foundational theories for our work.

3 Generation of Liquid Sound

Sound is produced by surface vibrations of an object under external force(s). These vibrations travel through the surrounding medium to the human ear and the changes in pressure are perceived as sound.

3.1 Spherical Bubble Principles

Sound in liquid is primarily generated by bubble formation and resonance. Although an impact between a solid and a liquid will generate some sound directly, the amplitude is far lower than the sound generated from the created bubbles. We refer the reader to Leighton's [1994] excellent text on bubble acoustics for more detail.

Minneart's formula, which derives the resonant frequency of a perfectly spherical bubble in an infinite volume of water from the radius, provides a physical basis for generating sound in liquids. Since other sound sources rarely exist in fluids, we assume that the bubble is given an initial excitation and subsequently oscillates, but is not continuously forced. The sound generated by the bubble will, therefore, be dominated by the resonant frequency, as other frequencies will rapidly die out after the bubble is created. Therefore, a resonating bubble acts like a simple harmonic oscillator, making the resonant frequency dependent on the stiffness of the restoring force and the effective mass of the gas trapped within the bubble. The stiffness of the restoring force is the result of the pressure within the bubble and the effective mass is dependent on the volume of the bubble and the density of the medium. If we approximate the bubble as a sphere with radius, r_0, then for cases where $r_0 > 1\mu m$, the force depends predominantly on the ambient pressure, p_0, and the resonant frequency is given by Minneart's formula,

$$f_0 = \frac{1}{2\pi}\sqrt{\frac{3\gamma p_0}{\rho r_0^2}}, \tag{1}$$

where γ is the ratio of specific heats of the gas (≈ 1.4 for air), p_0 is the gas pressure inside the bubble at equilibrium and ρ the density of the fluid. For air bubbles in water under one atmosphere, Eqn. 1 reduces to a simple form: $f_0 r_0 \approx 3m/s$. The human audible range is 20 Hz to 20 kHz, so we will restrict our model to the corresponding bubble of radii, 0.15 mm to 15 cm.

An oscillating bubble, just like a simple harmonic oscillator, is subject to viscous, radiative and thermal damping. Viscous damping rapidly goes to zero for bubbles of radius greater than 0.1 mm, so we will only consider thermal and radiative damping. We refer the reader to section 3.4 of [Leighton 1994] for a full derivation, and simply present the results. Thermal damping is the result of energy lost due to conduction between the bubble and the surrounding liquid, whereas radiative damping results from energy radiated away in the form of acoustic waves. These two can be approximated as,

$$\delta_{th} = \sqrt{\frac{9(\gamma-1)^2}{4G_{\text{th}}} f_0} \qquad \delta_{rad} = \sqrt{\frac{3\gamma p_0}{\rho c^2}} \tag{2}$$

where c is the speed of sound and G_{th} is a dimensionless constant associated with thermal damping. The total damping is simply the sum, $\delta_{tot} = \delta_{th} + \delta_{rad}$.

Modeling the bubble as a damped harmonic oscillator, oscillating at Minneart's frequency, the impulse response is given by

$$\tau(t) = A_0 sin(2\pi f(t) t) e^{-\beta_0 t}, \tag{3}$$

where A_0 is determined by the initial excitation of the bubble and $\beta_0 = \pi f_0 \delta_{tot}$ is the rate of decay due to the damping term δ_{tot} given above. We also replace f_0 in the standard harmonic oscillator equation with $f(t)$, where $f(t) = f_0(1 + \xi\beta_0 t)$. This helps mitigate the approximation of the bubble being in an infinite volume of water by adjusting the frequency as it rises over time and nears the surface. van den Doel [2005] conducted a user study and determined $\xi \approx 0.1$ to be the optimal value for a realistic rise in pitch.

The only outstanding question is the initial amplitude, A_0, in Eqn. 3, which we address is section 3.3.

3.2 Generalization to Non-Spherical Bubbles

The approximations given above assume that the shape of the bubble is spherical and given that an isolated bubble converges to a spherical shape, the previous method is a passable approximation. Even simple laboratory tests involving a single nozzle releasing air into a stationary fluid result in non-spherical bubbles, so we expect non-spherical bubbles to arise frequently in more complex and turbulent scenarios. Studies of bubble entrapment by ocean waves have also shown that the entrapment creates long, tube-like bubbles which we illustrate in one of our benchmarks (the "dam break" scenario in section 5), demonstrating the need to handle these types of bubbles. Longuet-Higgins performed a study showing that an initial distortion of the bubble surface of only $\frac{r_0}{2}$ results in a pressure fluctuation as large as $\frac{1}{8}$ atmosphere [Longuet-Higgins 1989a]. Therefore, the shape distortion of bubbles is a very significant mechanism for generating underwater sound. The generated audio also creates a more complete sound, since a single non-spherical bubble will generate multiple frequencies (as can be seen in the accompanying video).

In order to develop a more exact solution for non-spherical bubbles, we consider the deviations from the perfect sphere in the form of spherical harmonics, i.e.

$$r(\theta, \phi) = r_0 + \sum c_n^m Y_n^m(\theta, \phi) \tag{4}$$

We refer the reader to section 3.6 of [Leighton 1994] for a full derivation, but by solving for the motion of the bubble wall under the influence of the inward pressure, outward pressure and surface tension on the bubble (which depends on the curvature), it can be shown that each zonal spherical harmonic Y_n^0 oscillates at

$$f_n^2 \approx \frac{1}{4\pi^2}(n-1)(n+1)(n+2)\frac{\sigma}{\rho r_0^3} \tag{5}$$

where σ is the surface tension. Longuet-Higgins [1992] notes that unlike spherical bubbles, the higher order harmonics decay predominantly due to viscous damping, and not thermal or radiative damping. The amplitude of the n^{th} mode thus decays with $e^{-\beta_n t}$, and

$$\beta_n = (n+2)(2n+1)\frac{\nu}{\rho r_0^2} \tag{6}$$

where ν is the kinematic viscosity of the liquid. Given the frequency and damping coefficient for each spherical harmonic, we can again use Eqn. (3) to find the time evolution for each mode. Fig.2 gives several examples of oscillation modes corresponding to different spherical harmonics.

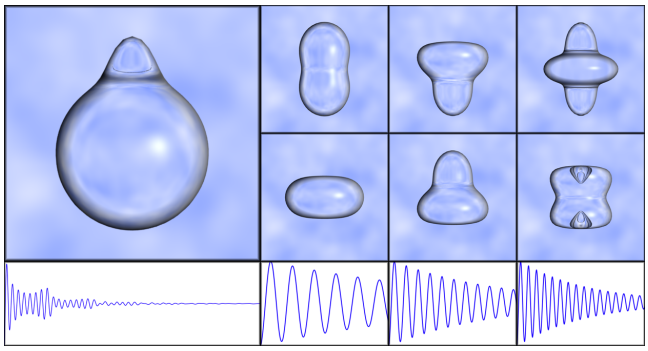

Figure 2: *Here we show a bubble decomposed into spherical harmonics. The upper left shows the original bubble. The two rows on the upper right show the two octaves of the harmonic deviations from the sphere for three harmonics. Along the bottom is the final sound generated by the bubble and the components for each harmonic.*

3.3 Amplitude of Bubble Excitation

For a bubble with mean radius r_0 that oscillates with a displacement ϵr_0, the pressure p at distance l is given by

$$p = -\frac{\epsilon r_0^3 w_0^2}{l} sin(2\pi f_0 t) \tag{7}$$

[Longuet-Higgins 1992]. Simplifying using Eqn. (1), we see that $|p| \propto \epsilon r_0/l$. Longuet-Higgins plugs in empirically observed values for $|p|$ and suggests that the initial displacement is of order 1% to 10% of the mean bubble radius r_0. Therefore, we can set $A_0 = \epsilon r_0$ in Eqn. (3), where $\epsilon \in [0.01, 0.1]$ is a tunable parameter that determines the initial excitation of the bubbles. We found that using a power law to select ϵ was effective and ensured that most of our values were below 10%.

For non-spherical bubbles, we have a separate instance of Eqn. (3) for each harmonic mode, and must determine the amplitude for each harmonic mode. The time-varying shape of the bubble can be described by the following formula,

$$r(\theta, \varphi; t) \sim r_0 + \sum_n c_n^0(t) Y_n^0(\theta, \varphi) \cos(2\pi f_n t + \vartheta), \tag{8}$$

and as with a spherical bubble, each n^{th} harmonic mode radiates pressure waves p_n as it oscillates. The first-order term of the radiated pressure p_n, when observed at a distance l from the source, depends on $(r_0/l)^{n+1}$ [Longuet-Higgins 1989a; Longuet-Higgins 1989b], which dies out rapidly and can be safely ignored. The second-order term of the radiated pressure decays as l^{-1} and oscillates at a frequency of $2f_n$, twice fast as the shape oscillation. Leighton proposes the following equation for p_n

$$\begin{aligned} p_n \quad &= -\tfrac{1}{l}\left(\tfrac{(n-1)(n+2)(4n-1)}{2n+1}\tfrac{\sigma c_n^2}{r_0^2}\right) \\ &\left(\frac{f_n}{\sqrt{(4f_n^2 - f_b^2)^2 + (4\beta_n f_n)^2}}\right) cos(4\pi f_n t) \end{aligned} \tag{9}$$

where c_n is the shorthand for c_n^0, the coefficient of the n^{th} zonal spherical harmonic from Eqn. (8), $f_b = (f_0^2 - \beta_0^2)^{\frac{1}{2}}$ is the frequency of the radial (0^{th}) mode (shifted due to damping), and β_n is the damping factor whose value is determined by Eqn. (6) and whose effect is such that $c_n(t) \propto e^{-\beta_n t}$. Using Eqns. (9) and (6), we can determine the time evolution of each of the n spherical harmonic modes. It is worth noting here, that the second term in Eqn. (9) depends on $\frac{1}{4f_n^2 - f_b^2}$, which means that as $2f_n$ approaches f_b, the n^{th} mode resonates with the 0^{th} mode, and the value of $|p_n|$ increases dramatically, as shown in Fig. ???. Therefore we select the most important modes in the spherical harmonic decomposition (described in section 4.4), by choosing values of n with frequencies that are close to $\frac{1}{2}f_b$. The number of spherical harmonics used is determined by the user, cutting off the modes (the left end and the right end of Fig. ???) whose contributions are below a certain threshold. In our benchmarks, using 10 modes worked well.

4 Integration with Fluid Dynamics

Given these basic principles for generating sound from liquid, we now present our approach to couple these formulations with a fluid simulator that computes liquid dynamics and properly handles bubble formation and interactions.

4.1 Types of Fluid Simulators

There are many challenging computational issues in this coupling, the first is selecting a fluid simulation framework. As mentioned earlier, the three commonly used categories for fluid dynamics in visual simulation are grid-based methods, SPH and shallow-water approximations. We consider two fluid simulators that utilize all three of these methods. Our shallow water formulation is an integrated adaptation of the work of Thürey et al. [2007a; 2007b] and Hess [2007]. We present a brief overview of our shallow-water approximation in the supplementary document (Appendix A). The other is a hybrid grid-based and SPH approach, taken heavily from the work of Hong et al. [2008]. We present a brief overview of the mathematical formulation in the supplementary document (Appendix B) and refer the readers to [Hong et al. 2008] for details.

4.2 Overall Algorithm

Bubble Generation Model: In the case of the shallow water equations, where bubbles are not handled by the fluid simulator, we present a simple heuristic for bubble generation. Recent works in visual simulation by Narain et al. [2007] and Mihalef et al. [2009] use curvature and Weber number, respectively, to determine regions where bubbles should be created. We use Weber number, $We = \frac{\rho \Delta U^2 L}{\sigma} \propto u^2 \kappa$, where u is the fluid velocity and κ is the curvature of the fluid surface. When the Weber number is above a threshold, we generate a bubble in that location. Works on bubble entrapment by rain [Pumphrey and Elmore 1990] and ocean waves [Deane and Stokes 2002] suggest that bubbles are created in a $r^{-\alpha}$ distribution, where α determines the ratio of small to large bubbles. In nature the $alpha$ takes value from 1.5 to 3.3 for breaking ocean waves [Deane and Stokes 2002] and ≈ 2.9 for rains [Pumphrey and Elmore 1990], thus in simulation it can be tuned according to the scenario. Therefore, once we have determined a location for a new bubble using the Weber number, we select a bubble at random from the distribution and generate sound from it.

Adaptive Sound Synthesis: For the grid-based fluid simulator, once it has taken a time step, we need to identify the bubbles and synthesize sounds from them. Specifically, we need to handle two types of bubbles, those formed by the level sets and those formed by the SPH particles. The level-set bubbles can be separated from the rest of the mesh returned by the level set method because they lie completely beneath the water surface and form fully connected components. Once we have meshes representing the surface of the bubbles, we decompose each mesh into spherical harmonics that approximate the shape, using the algorithm presented in Section 4.4. The spherical harmonic decomposition and the subsequent sound synthesis is linear in the number of harmonic modes calculated. Therefore, the number of spherical harmonics calculated can be adjusted depending on desired accuracy and available computation time. Once we have the desired number of spherical harmonics, we determine the resonant frequencies using Eqn. (5).

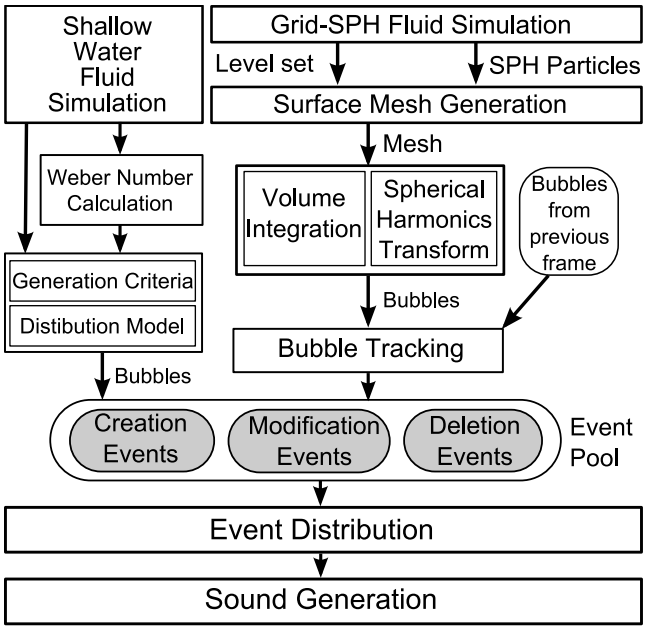

Figure 3: *Overview of Our Liquid Sound Synthesis System*

For SPH bubble particles, there are two cases–when a bubble is represented by a single particle and when it is represented by multiple particles. In the case of a single particle bubble, as with the shallow water equations, we simply use the radius and Eqn. (1). When multiple SPH particles form one bubble, we need to determine the surface formed by the bubble. We first cluster the particles into groups that form a single bubble and then use the classic marching cubes algorithm [Lorensen and Cline 1987] within each cluster to compute the surface of the bubble. Once we have the surface of the bubble, we use the same method as the level set bubble to find the spherical harmonics and generate audio. Our general method is summarized in Fig. 3 with pseudo code given in the supplementary document (Appendix C).

Decoupling Sound Update from Graphical Rendering: Since computing the fluid dynamics at 44,000 Hz, the standard frequency for good quality audio, would add an enormous computation burden, we need to reconcile the difference between the fluid simulator time step, T_{sim} (30-60 Hz), and the audio generation time step, T_{audio} (44 kHz). We can use Eqn. (1) and Eqn. (5) to calculate the resonant frequency at each T_{sim} and then use Eqn. (3) to generate the impulse response for all the T_{audio}'s until the subsequent T_{sim}. Naively computing the impulse response based on Eqn. (3) at each T_{audio} can create complications due to a large number of events that take place in phase at each T_{sim}. In order to resolve this problem, we randomly distribute each creation, merge and deletion event from T_{sim} onto one of the $\sim$733 T_{audio}'s between the current and last T_{sim}. The final issue is how to track when to create new bubbles, delete old bubbles, merge multiple bubbles or split one bubble into many. We discuss these events in the next section.

4.3 Bubble Tracking and Merging

At each time step the fluid simulator returns a list of level set bubble meshes and SPH particles which we convert into a set of meshes, each representing a single bubble. At each subsequent time step we collect a new set of meshes and compare it to the set of meshes from the previous time step with the goal of identifying which bubbles are new, which are preexisting and which have disappeared. For each mesh, M, we attempt to pair it with another mesh, M_{prev}, from the previous time step such that they represent the same bubble after moving and deforming within the time step. We first choose a distance, $l \geq v_{max}\Delta t$, where v_{max} is the maximum possible speed of a bubble. We then define $\texttt{neighbor}(M, l)$ as the set of meshes from the previous time step whose center of masses lie within l of M. For each mesh in $\texttt{neighbor}(M, l)$, we compute its *similarity score* based on the proximity of its center of mass to M and the closeness of the two volumes, choosing the mesh with the highest similarity score. Once we have created all possible pairs of meshes between the new and the old time steps, we are left with a set of bubbles from the old time step with no pair–the bubbles to remove– and a set of bubbles in the new time step–the bubbles to create. Although it may be possible to create slightly more accurate algorithm by tracking the particles that define an SPH or level-set bubble, these methods would also present nontrivial challenges. For example, in the case of tracking the level set bubbles, the level set particles are not guaranteed to be spaced in any particular manner and are constantly added and deleted, making this information difficult to use. In the case of tracking bubbles formed by SPH particles, there would still be issues related to bubbles formed by multiple SPH particles. The shape could remain primarily unchanged with the addition or removal of a single particle and therefore the audio should remain unchanged as well, even though the IDs of the particles change. We chose this simple approach because of its generality and its ability to handle both level-set and SPH bubbles, as well as other fluid simulators, using the same framework.

4.4 Spherical Harmonic Decomposition

In order to decompose a mesh, M, into a set of the spherical harmonics that approximate it, we must first guarantee that M is a closed triangulated surface mesh and that it is *star-shaped*. A mesh is *star-shaped* if there is a point o such that for every point p on the surface of M, segment $\overline{op}$ lies entirely within M. The length of the segment $\overline{op}$ can be described as a function $|\overline{op}| = r(\theta, \varphi)$ where θ and φ are the polar and azimuthal angles of p in a spherical coordinate system originating at o. The function $r(\theta, \varphi)$ can be expanded as a linear combination of spherical harmonic functions as in Eqn.(4).

The coefficient c_n^m can be computed through an inverse transform $c_n^m = \int_\Omega P(\theta, \varphi)\overline{Y}_n^m(\theta, \varphi)d\Omega$, where the integration is taken over Ω, the solid angle corresponding to the entire space. Furthermore, if T is a triangle in M and we define the solid angle spanned by T as Ω_T, then we have $\Omega = \bigcup_{T\in M}\Omega_T$ and $c_n^m = \sum_{T\in M}\int_{\Omega_T} P(\theta, \varphi)\overline{Y}_n^m(\theta, \varphi)d\Omega$. The integration can be calculated numerically by sampling the integrand at a number of points on each triangle. For sound generation, we only need the zonal coefficients c_n^0, with n up to a user defined bandwidth, B. The spherical harmonic transform runs in $O(BN_p)$ where N_p is the total number of sampled points.

If the bubble mesh is not star-shaped, then it cannot be decomposed into spherical harmonics using Eqn. (4). To ensure that we generate sound for all scenarios, if our algorithm cannot find a spherical harmonic decomposition it automatically switches to a single mode approximation based on the total volume of the bubble. Since this only happens with large, low-frequency bubbles, we have not noticed any significant issues resulting from this approximation or the transition between the two generation methods.

5 Implementation and Results

Our shallow water implementation is based on the work of Thürey et al. [2007a; 2007b] and Hess [2007] while our hybrid SPH-grid-based simulator is based on the work of Hong et al. [2008]. The rendering for the shallow water simulation is performed in real time using OpenGL and custom vertex and fragment shaders while the rendering for the hybrid simulator is done off-line using a forward ray tracer. In both cases, once the amplitude and frequency of the bubble sound was calculated, the final audio was rendered using

The Synthesis ToolKit [Cook and Scavone].

5.1 Benchmarks

We have tested our integrated sound synthesis system on the following scenarios.

5.1.1 Hybrid Grid-SPH Simulator

Pouring Water: In this scenario, water is poured from a spigot above the surface as shown in Fig. 1. The initial impact creates a large bubble as well as many small bubbles. The large bubble disperses into smaller bubbles as it is bombarded with water from above. The generated sound takes into account the larger bubbles as well as all the smaller ones, generating the broad spectrum of sound heard in the supplementary video. Up to 1,530 bubbles were processed in one simulation frame to generate the sounds.

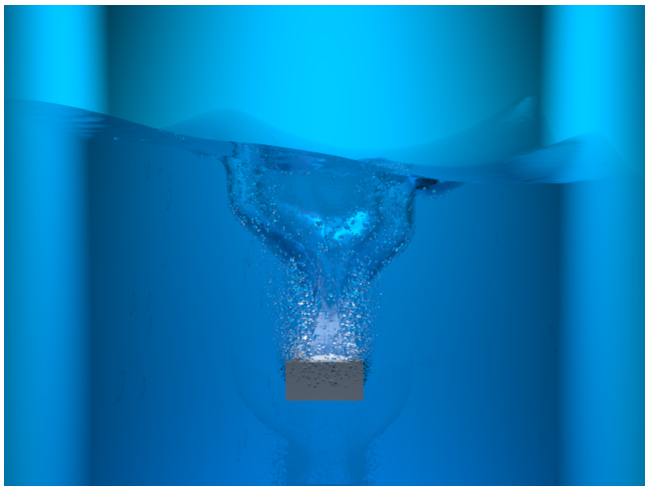

Figure 4: *Sound is generated as five objects fall into a tank of water one after another.*

Five Objects: In this benchmark, shown in Fig. 4, five objects are dropped into a tank of water in rapid succession, creating many small bubbles and one large bubble as each one plunges beneath the water surface. The video shows the animation and the sound resulting from the initial impacts as well as the subsequent bubbles and sound generated by the sloshing of the water around the tank. We used ten spherical harmonic modes and processed up to 15,000 bubbles.

Dam Break: In this benchmark, shown in Fig. 5, we simulate the "dam break" scenario that has been used before in fluid simulation, however, we generate the associated audio automatically. We processed up to 15,000 bubbles using five spherical harmonic modes. This benchmark also demonstrates the creation of a tube-shaped bubble when the wave breaks as the water sloshes back to the left, highlighting the importance of our spherical harmonic decomposition to handle scenarios such as this, where highly non-spherical bubbles are generated (please see the supplementary video).

5.1.2 Shallow Water Simulator

Brook: Here we simulate the sound of water as it flows in a small brook. We demonstrate the interactive nature of our method by increasing the flow of water half way through the demo, resulting in higher velocities and curvatures of the water surface and therefore, louder and more turbulent sound.

Duck: As shown in Fig. 7, as a user interactively moves a duck around a bathtub, our algorithm automatically generates the associated audio. The waves created by the duck produces regions of high curvature and velocity, creating resonating bubbles.

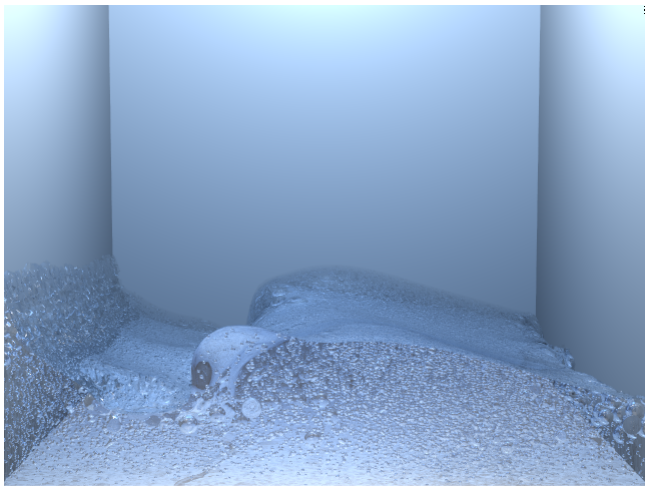

Figure 5: *A "dam-break" scenario, a wall of water is released,*

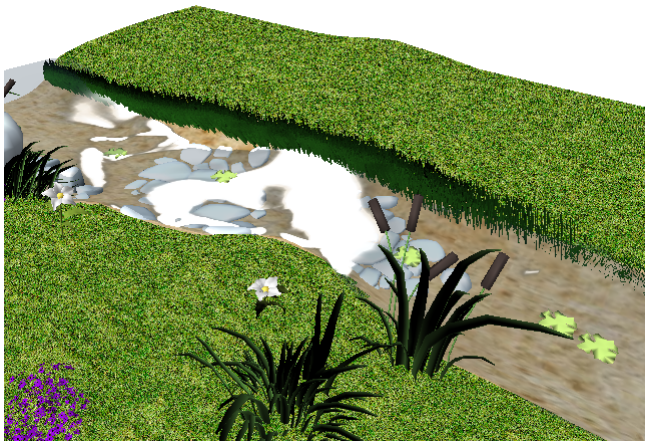

Figure 6: *Real-time sounds are automatically generated from an interactive simulation of a creek flowing through a meadow.*

5.2 Preliminary User Study

We conducted a preliminary user study and received encouraging feedback. 30 participants were asked to rank the perceived realism of the video clips seen in the supplementary video. For each video, it was played with the synthesized audio and without. The results, shown in Table 1, confirm that the synthesized audio consistently increased the perceived realism of the audio-visual experience. We also asked participants to rate the realism of various audio clips, including both our synthesized audio and clips recorded in nature. The results for the recorded clips (8.3 ± 1.7) were higher than our synthesized audio (5.6 ± 2.5); however, our clips were within one standard deviation from recorded sounds that have additional suggestive aural cues from nature.

5.3 Limitations and Future Work

Although our method generates realistic sounds for multiple benchmarks, there are some limitations of our technique. Since we are generating sound from bubbles, the quality of the synthesized sounds depends on the accuracy and correctness of bubble formation from the fluid simulator. Continued research on fluid simulations involving bubbles would improve the quality and accuracy of

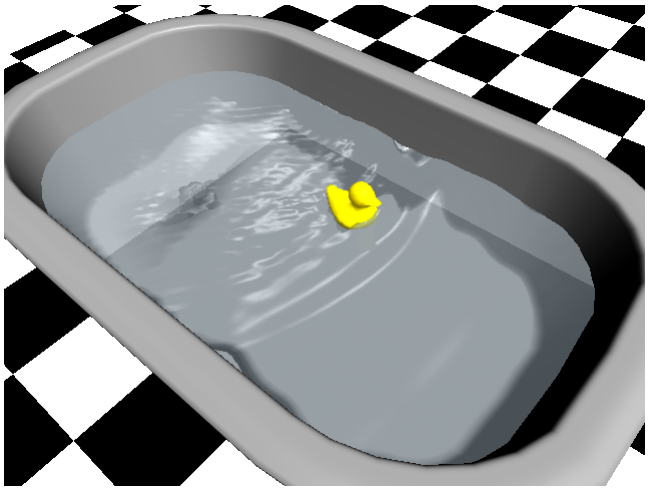

Figure 7: *Sounds are automatically generated as a (invisible) user moves a duck in a bathtub.*

	Without Audio	With Audio
Five Objects	7.4 ± 1.7	8.0 ± 1.8
Dam Break	6.8 ± 2.0	7.2 ± 1.8
Pouring	6.7 ± 2.0	7.5 ± 1.8
Creek	5.3 ± 1.9	5.4 ± 2.3
Duck	5.7 ± 2.0	6.8 ± 2.0

Table 1: *Results of Preliminary User Study*

the sound generated using our approach.

For non-star-shaped bubbles, because they cannot be decomposed into spherical harmonics, we are forced to revert to the simple spherical bubble approximation. There has been some recent work on simulating general bubble oscillations using a boundary element method [Pozrikidis 2004]. We could provide more accuracy for complex bubble shapes using a similar technique, but not without substantially higher costs.

6 Conclusions

We present the first automatic, physically-based synthesis method that generates liquid sounds directly from the fluid simulator based on bubble resonance. Our approach is general and applicable to different types of fluid simulation methods commonly used in computer graphics. It can run at interactive rates and its sound quality depends on the physical correctness of the fluid simulators. Our initial user studies suggest that the perceived realism of liquid sounds generated using our approach is comparable to recorded sounds in similar settings.

References

BONNEEL, N., DRETTAKIS, G., TSINGOS, N., VIAUD-DELMON, I., AND JAMES, D. 2008. Fast modal sounds with scalable frequency-domain synthesis. *ACM Trans. Graph. 27*, 3, 1–9.

BRIDSON, R., AND MÜLLER-FISCHER, M. 2007. Fluid simulation: SIGGRAPH 2007 course notes. In *ACM SIGGRAPH 2007 courses*, ACM, San Diego, California, 1–81.

COOK, P. R., AND SCAVONE, G. P. The synthesis ToolKit in c++ 4.3.1.

DEANE, G. B., AND STOKES, M. D. 2002. Scale dependence of bubble creation mechanisms in breaking waves. *Nature 418*, 6900, 839–844.

DING, J., TSAUR, F. W., LIPS, A., AND AKAY, A. 2007. Acoustical observation of bubble oscillations induced by bubble popping. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics 75*, 4 Pt 1 (Apr.).

DOBASHI, Y., YAMAMOTO, T., AND NISHITA, T. 2003. Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics. *ACM Trans. Graph. 22*, 3, 732–740.

DOBASHI, Y., YAMAMOTO, T., AND NISHITA, T. 2004. Synthesizing sound from turbulent field using sound textures for interactive fluid simulation. *Computer Graphics Forum 23*, 3, 539–545.

FLORENS, J., AND CADOZ, C. 1991. *The physical model: modeling and simulating the instrumental universe*. MIT Press, 227–268.

FOSTER, N., AND FEDKIW, R. 2001. Practical animation of liquids. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 23–30.

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

HESS, P. 2007. *Extended Boundary Conditions for Shallow Water Simulations*. PhD thesis, ETH Zurich.

HONG, J., LEE, H., YOON, J., AND KIM, C. 2008. Bubbles alive. In *ACM SIGGRAPH 2008 papers*, ACM, Los Angeles, California, 1–4.

IMURA, M., NAKANO, Y., YASUMURO, Y., MANABE, Y., AND CHIHARA, K. 2007. Real-time generation of CG and sound of liquid with bubble. In *ACM SIGGRAPH 2007 posters*, ACM, San Diego, California, 97.

JAMES, D. L., BARBIČ, J., AND PAI, D. K. 2006. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Trans. Graph. 25*, 3, 987–995.

LEIGHTON, T. G. 1994. *The acoustic bubble*.

LONGUET-HIGGINS, M. S. 1989. Monopole emission of sound by asymmetric bubble oscillations. part 1. normal modes. *Journal of Fluid Mechanics 201*, 525–541.

LONGUET-HIGGINS, M. S. 1989. Monopole emission of sound by asymmetric bubble oscillations. part 2. an initial-value problem. *Journal of Fluid Mechanics 201*, 543–565.

LONGUET-HIGGINS, M. S. 1991. Resonance in nonlinear bubble oscillations. *Journal of Fluid Mechanics 224*, 531–549.

LONGUET-HIGGINS, M. S. 1992. Nonlinear damping of bubble oscillations by resonant interaction. *The Journal of the Acoustical Society of America 91*, 3 (Mar.), 1414–1422.

LORENSEN, W. E., AND CLINE, H. E. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Comput. Graph. 21*, 4, 163–169.

MIHALEF, V., METAXAS, D., AND SUSSMAN, M. 2009. Simulation of two-phase flow with sub-scale droplet and bubble effects. *Proceedings of Eurographics 2009 28*.

MINNAERT, M. 1933. On musical air bubbles and the sound of running water. *Philosophical Magazine 16*, 235–248.

NARAIN, R., KWATRA, V., LEE, H., KIM, T., CARLSON, M., AND LIN, M. 2007. Feature-Guided dynamic texture synthesis on continuous flows. In *Feature-Guided Dynamic Texture Synthesis on Continuous Flows*.

O'BRIEN, J. F., COOK, P. R., AND ESSL, G. 2001. Synthesizing sounds from physically based motion. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 529–536.

O'BRIEN, J. F., SHEN, C., AND GATCHALIAN, C. M. 2002. Synthesizing sounds from rigid-body simulations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, San Antonio, Texas, 175–181.

PLESSET, M., AND PROSPERETTI, A. 1977. Bubble dynamics and cavitation. 145–185.

POZRIKIDIS, C. 2004. Three-dimensional oscillations of rising bubbles. *Engineering Analysis with Boundary Elements 28*, 4 (Apr.), 315–323.

PROSPERETTI, A., AND OGUZ, H. 1993. The impact of drops on liquid surfaces and the underwater noise of rain. 577–602.

PUMPHREY, H. C., AND ELMORE, P. A. 1990. The entrainment of bubbles by drop impacts. *Journal of Fluid Mechanics 220*, 539–567.

RAGHUVANSHI, N., AND LIN, M. C. 2006. Interactive sound synthesis for large scale environments. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, ACM, Redwood City, California, 101–108.

RAYLEIGH, L. 1917. On pressure developed in a liquid during the collapse of a spherical cavity. *Philosophical Magazine*.

SCAVONE, G. P., AND COOK, P. R. 1998. Real-time computer modeling of woodwind instruments. Acoustical Society of America, Woodbury, NY.

STAM, J. 1999. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 121–128.

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 *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, San Diego, California, 191–198.

THÜREY, N., MÜLLER-FISCHER, M., SCHIRM, S., AND GROSS, M. 2007. Real-time breaking waves for shallow water simulations. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, IEEE Computer Society, 39–46.

VAN DEN DOEL, K., KRY, P. G., AND PAI, D. K. 2001. FoleyAutomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 537–544.

VAN DEN DOEL, K. 2005. Physically based models for liquid sounds. *ACM Trans. Appl. Percept. 2*, 4, 534–546.