Sounding Liquids:
Automatic Sound Synthesis from Fluid Simulation

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

We present a novel approach for synthesizing liquid sounds directly from visual simulation of fluid dynamics. Our approach takes advantage of the fact that the sound generated by liquid is mainly due to the vibration of resonating bubbles in the medium and performs automatic sound synthesis by coupling physically-based equations for bubble resonance with multiple fluid simulators. We effectively demonstrate our system on several benchmarks using a real-time shallow-water fluid simulator as well as a hybrid grid-SPH simulator.

Categories and Subject Descriptors: H.5.5 [**Information Interfaces and presentation (e.g., HCI)**]: Sound and Music Computing—*Modeling*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Animation*

General Terms:

Additional Key Words and Phrases: sound simulation, liquids

ACM Reference Format:

Moss, W., Yeh, H., Hong, J. M., Lin, M. C., Manocha, D. 2009. Sounding Liquids: Automatic Sound Synthesis from Fluid Simulation

This work was supported in part by Army Research Office, Intel, National Science Foundation, and RDECOM.
Authors addresses:
W. Moss, H. Yeh, M. Lin and D. Manocha; email: {wmoss, hyeh, lin, dm}@cs.unc.edu
J. Hong; email: jmhong@dongguk.edu

DOI 10.1145/1559755.1559763
http://doi.acm.org/10.1145/1559755.1559763

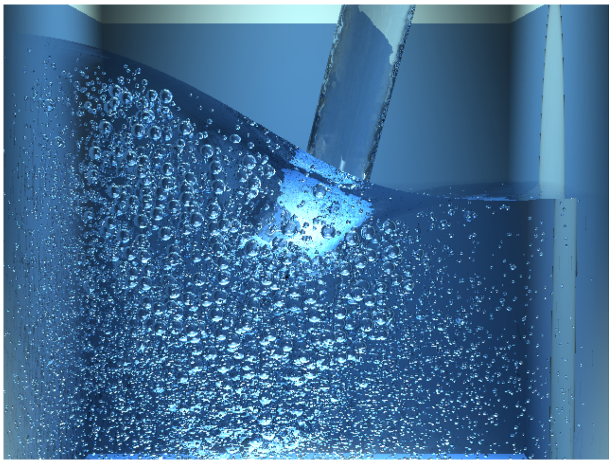


Fig. 1. Liquid sounds are generated automatically from a visual simulation of pouring water.

1. INTRODUCTION

Auditory display provides a natural, intuitive human-computer interface for many desktop applications including video games, training systems, computer aided design, scientific visualization and assistive technology 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 the sound generated by 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 for non-spherical bubbles [Leighton 1994]. 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 to balance between computational cost and quality.

We also decouple sound rendering rates (44,000 Hz) from graphical updates (30-60 Hz) by distributing the bubble processing 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 (Fig. 1) and liquid-object interaction (Fig. 7).

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; Trebien and Oliveira 2009; Picard et al. 2009], the sound resulting from objects moving rapidly through air [Dobashi et al. 2003; 2004] and the sound of woodwinds and other instruments [Florens and Cadoz 1991; Scavone and Cook 1998].

Liquid Sounds: 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 [1989b; 1989a] and non-linear oscillations [1991]. T. G. Leighton's [1994] excellent text covers the broad field of bubble acoustics and provides many of the foundational theories for our work.

Our work is inspired by van den Doel [2005], who introduced the first method in computer graphics 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, van den Doel provides a simple technique for generating fluid sounds through the adustment of various parameters. Our work generalizes this approach by enabling visual simulation of fluid dynamics to determine these parameters automatically, making it possible to synthesize liquid sounds directly from fluid simulation. We also introduce efficient methods for handling non-spherical bubbles, which occur frequently in nature. Other previous liquid sound synthesis methods provide limited physical basis for the generated sounds [Imura et al. 2007].

Harmonic Fluids: Concurrent with our work, [Zheng and James 2009] coupled a fluid simulator with sound synthesis. We highlight the similarities and differences between the two works here. Both [Zheng and James 2009] and our work share one notable contribution: the integration of fluid simulation with bubble-based sound synthesis to automatically generate liquid sounds. Beyond this, however, the focuses of these two papers are different. Zheng and James consider a specific fluid simulator that uses a single-bubble model, relying on a synthesis method identical to [van den Doel 2005]. They do not address real-time synthesis, issues surrounding multiple fluid simulators or coping with non-spherical bubbles encountered in a variety of existing fluid simulators, instead focusing on the propagation of sound – both from the bubble to the water surface and the water surface to the listener.

On the other hand, our work considers different types of fluid simulators and deals with the challenge of real-time sound synthesis for all of them. By handling only bubbles meshes (and not individually identified bubbles) at visual rendering rates, we ensure that our solution is as generic as possible. Our system can automatically handle all types of bubbles (spherical and non-spherical) and the interactions between those bubbles that occur naturally. In addition, by coupling our synthesis technique to a real-time fluid simulator, we also demonstrate the possibility for interactive sound synthesis and synthesis without explicitly simulated bubble formation. Like many earlier papers on sound synthesis, we do not address sound propagation in our work, leaving that to other works (such as [Zheng and James 2009]). In general, numerical sound radiation is compute-intensive and often requires many hours of compute time on super-computing platforms, as reported in [Zheng and James 2009].

Finally, we also conduct a user study to assess the realism of synthesized sounds using our approach. To that end, although these two works share a common theme, they actually address two distinct and complementary aspects of sound rendering for fluids.

3. LIQUID SOUND PRINCIPLES

Sound is produced by the surface vibrations of an object under force(s). These vibrations travel through the surrounding medium to the human ear and the changes in pressure are perceived as sound. In the case of fluids, sound is primarily generated by bubble formation and resonance, creating pressure waves that travel though both the liquid and air media to the ear. Although an impact between a solid and a liquid will generate some sound directly, the amplitude is far lower than the sound generated by the created bubbles. We refer the reader to Leighton's [1994] excellent text on bubble acoustics for more detail, and present an overview of the key concepts below.

3.1 Spherical Bubbles

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 external sound sources rarely exist in fluids and the interactions between resonating bubbles create a minimal effect (while greatly increasing the computational cost), we assume that a 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, since 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 of the surrounding water, 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 specific heat of the gas (≈ 1.4 for air), p_0 is the gas pressure inside the bubble at equilibrium (i.e. when balanced with the pressure of the surrounding water) and ρ the density of the surrounding fluid. For air bubbles in water, 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 can restrict our model to the corresponding bubbles 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 peritinant equations here. 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

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

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. For single-mode bubbles in low concentration, we replace f_0 in the standard harmonic oscillator equation with $f(t)$, where $f(t) = f_0(1 + \xi\beta_0 t)$, which helps mitigate the approximation of the bubble being in an infinite volume of water by adjusting the frequency as it rises 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.

To find the initial amplitude, A_0, in Eqn. 3, [Longuet-Higgins 1992] considers a bubble with mean radius r_0 that oscillates with a displacement ϵr_0, the pressure p at distance l is given by

$$p(t) = -\frac{4\pi^2 \epsilon r_0^3 f_0^2}{l} sin(2\pi f_0 t). \tag{4}$$

Simplifying by plugging in f_0 from 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 1% to 10% of the mean bubble radius r_0. Therefore, we can set

$$A_0 = \epsilon r_0 \tag{5}$$

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

$$g(\epsilon) \propto \epsilon^{-\mu}, \tag{6}$$

where g is the probability density function of ϵ. By carefully choosing the *scaling exponent* μ, we can ensure that most of the values of ϵ are within the desired range, i.e. below 10%. This gives us a final equation for the pressure wave created by an oscillating spherical bubble (i.e. what travels through the water, then air, to our ear) of

$$p(t) = \epsilon r_0 sin(2\pi f(t)t)e^{-\beta_0 t} \qquad \epsilon \in [0.01, 0.1] \tag{7}$$

3.2 Generalization to Non-Spherical Bubbles

The approximations given above assume that the shape of the bubble is spherical. Given that an isolated bubble converges to a spherical shape, the previous method is a simple and reasonable approximation. That said, we expect non-spherical bubbles to arise frequently in more complex and turbulent scenarios. For example, studies of bubble entrapment by ocean waves have shown that breaking waves create long, tube-like bubbles. We illustrate the necessity of handling these types of bubbles in our "dam break" scenario (see Sec. 5). Longuet-Higgins also 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 1989b]. 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 heard 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.

$$r(\theta, \phi) = r_0 + \sum c_n^m Y_n^m(\theta, \phi). \tag{8}$$

Section 3.6 of [Leighton 1994] presents a full derivation for this equation. 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

$$f_n^2 \approx \frac{1}{4\pi^2}(n-1)(n+1)(n+2)\frac{\sigma}{\rho r_0^3} \tag{9}$$

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 $e^{-\beta_n t}$, where

$$\beta_n = (n+2)(2n+1)\frac{\nu}{\rho r_0^2} \tag{10}$$

and 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.

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

$$r(\theta, \varphi; t) \sim r_0 + \sum_n c_n^0(t) Y_n^0(\theta, \varphi)\cos(2\pi f_n t + \vartheta), \tag{11}$$

and as with a spherical bubble, each n^{th} harmonic mode radiates a pressure wave 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 1989b; 1989a], which dies

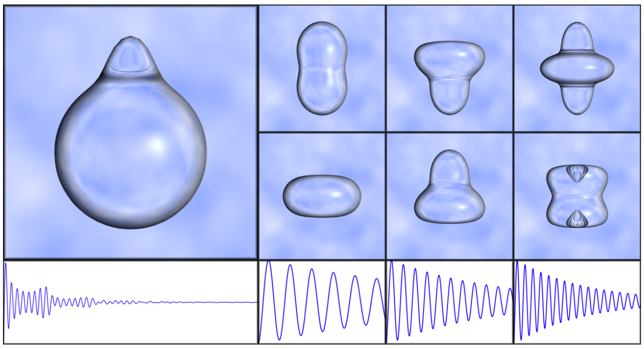


Fig. 2. Here we show a simple bubble decomposed into spherical harmonics. The upper left shows the original bubble. The two rows on the upper right show the two octaves of the harmonic deviations from the sphere. Along the bottom is the sound generated by the bubble and the components for each harmonic.

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 as fast as the shape oscillation. Leighton proposes the following equation for p_n

$$p_n(t) = -\frac{1}{l}\left(\frac{(n-1)(n+2)(4n-1)}{2n+1}\frac{\sigma c_n^2}{r_0^2}\right) \left(\frac{\omega_n^2}{\sqrt{(4\omega_n^2-\omega_b^2)^2+(4\beta_n\omega_n)^2}}\right) e^{-\beta_n t} cos(2\omega_n t) \quad (12)$$

where c_n is the shorthand for c_n^0, the coefficient of the n^{th} zonal spherical harmonic from Eqn. (11), $\omega_n = 2\pi f_n$, $\omega_b = 2\pi f_b = 2\pi(f_0^2 - \beta_0^2)^{\frac{1}{2}}$ is the angular frequency of the radial (0^{th}) mode (shifted due to damping), and β_n is the damping factor whose value is determined by Eqn. (10). Using Eqns. (10) and (12) we can determine the time evolution of each of the n spherical harmonic modes.

In order to determine the number of spherical harmonics to be used, several factors need to be considered. First notice that mode n oscillates at a frequency of $2f_n$, creating a range of n whose resulting pressure waves are audible. We define N_{aud} to be the number of these audible n's. N_{aud} can be derived using Eqn. (9), the radius r_0 of a bubble and the human audible range (20 to 20,000 Hz).

The second term in Eqn. (12) depends on $1/(4\omega_n^2 - \omega_b^2)$, which means that as $2\omega_n$ approaches ω_b (thus $2f_n$ 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. 3. Therefore we select the most important modes in the spherical harmonic decomposition (described in section 4.2.4), by choosing values of n with frequencies close to $\frac{1}{2}f_b$ and truncating the rest of the modes (corresponding to the left and the right tails in Fig. 3). We compute the initial energy for each mode, E_n (proportional to $|p_n|^2$), and collect the modes starting from the largest E_n, until (1) E_n is less than a given percentage, p, of the largest mode, E_{max}; or (2) the sum of energy of the modes not yet selected is less than a percentage, p, of the total energy of all audible modes, E_{total}. The number of modes selected by (1) is denoted as $N_{ind}(p)$, and that by (2) as $N_{tot}(p)$. Some typical values for different r_0's are shown in Table I. One may choose either one of the two criteria or a combination of both. As indicated in Table I, 8 modes seems sufficient for various sizes of bubble radii using the criterion (1), where the E_n falls below 1% of E_{max}. Therefore, a fixed number of modes, say 8 to 10, can be used in practice.

Table I. Number of modes selected by the two criteria for various typical r_0's.

r_0 (m)	N_{aud}	$N_{ind}(1\%)$ ($t=0$)	$N_{tot}(10\%)$ ($t=0$)	$N_{ind}(1\%)$ ($t=10^{-3}$s)	$N_{tot}(10\%)$ ($t=10^{-3}$s)
0.5	1881	4	1109	4	87
0.05	90	8	106	8	12
0.005	20	4	1	4	1

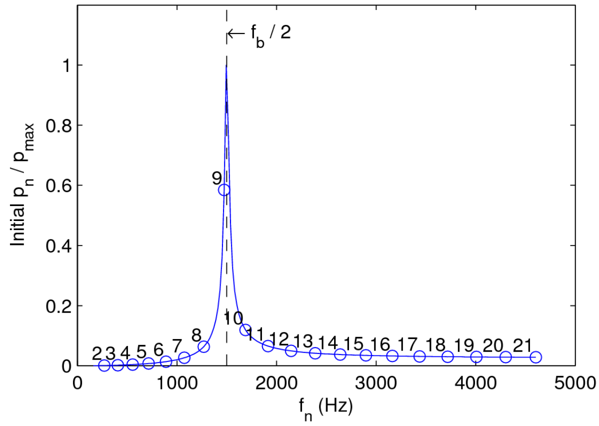


Fig. 3. A plot of the initial amplitude vs. frequency. From the plot it is clear that as f_n (the frequency of the bubble) approaches $\frac{1}{2}f_b$ (the damping shifted frequency) the initial amplitude increases dramatically. We, therefore, use harmonics where $f_n \approx \frac{1}{2}f_b$ because they have the largest influence on the initial amplitude.

Furthermore, recall that in Eqn. (12) the pressure decays exponentially with a rate β_n, where Eqn. (10) tells us that β_n increases with n and decreases with r_0. If we choose to ignore the initial "burst" and only look at the pressure wave a short time (e.g. 0.001 s) after the creation of the bubble, then we can drop out even more modes at the beginning. This step is optional and the effect is shown in the rightmost two columns of Table I.

Eqns. (7) and (12) provide the mechanism for computing the sound generated by either single or multi-mode bubbles, respectively. The pressure waves created by the oscillating bubble travel through the surrounding water, into the air and to the listener. Since we do not consider propagation in this work, we assume a fixed distance between the listener and each bubble using Eqns. (7) and (12) to model the pressure at the listener's ear.

3.3 Statistical Generation

In the case where the fluid simulator does not handle bubble generation, we present a statistical approach for generating sound. For a scene at a particular time instant, we consider how many bubbles are created and what they sound like. The former is determined by a bubble generation criteria and the latter is determined by a radius distribution model. As a result, even without knowing the exact motion and interaction of each bubble from the fluid simulator, a statistical approach based on our bubble generation criteria and radius distribution model provide sufficient information for approximating the sound produced in a given scene.

3.3.1 *Bubble Generation Criteria.* Our goal is to examine only the physical and geometrical properties of the simulated fluid, such as fluid velocity and the shape of the fluid surface, and be able to determine when and where a bubble should be generated. Recent works in visual simulation use curvature alone [Narain et al. 2007], or curvature combined with Weber number [Mihalef et al. 2009] as the bubble generation criteria.

In our work, we follow the approach presented by Mihalef et al. [2009]. The Weber number is defined as

$$We = \frac{\rho \Delta U^2 L}{(\sigma)} \tag{13}$$

where ρ is the density of the fluid, ΔU is the relative gas-liquid velocity, L is the characteristic length of the local liquid geometry and σ is the surface tension coefficient [Sirignano 2000]. This dimensionless number We can be viewed as the ratio of the kinetic energy (proportional to $\rho \Delta U^2$) to the surface tension energy (proportional to σ / L). Depending on the local shape, when this ratio is beyond a critical value, the gas has sufficient kinetic energy to "break into" the liquid surface and form a bubble; while at lower Weber numbers, the surface tension energy is able to separate the water and air.

Besides the Weber number, we also need to consider the limitation of a fluid simulator. In computer graphics, fluid dynamics is usually solved on a large-scale grid, with small-scale details such as bubbles and droplets added in at regions where the large-scale simulation behaves poorly, namely regions of high curvature. This is because a bubble is formed when the water surface curls back and closes up, at which site the local curvature is high.

Combining the effects of the Weber number and the local geometry, we evaluate the following parameter on the fluid surface

$$\Gamma = u^2 \kappa, \tag{14}$$

where u is the liquid velocity and κ is the local curvature of the surface. The term u^2 encodes the Weber number, because in Eqn. 13 ρ, σ and L (which is taken to be the simulation grid length dx) are constants, and $\Delta U^2 = u^2$ since the air is assumed to be static. Bubbles are generated at regions where Γ is greater than a threshold Γ_0. The criteria also matches what we observe in nature–a rapid river (larger u) is more likely to trap bubbles than a slow one. In the ocean, bubbles are more likely to form near a wave (larger κ) than on a flat surface–our bubble generation mechanism captures both of these characteristics.

3.3.2 *Bubble Distribution Model.* Once we have determined a location for a new bubble using the generation criteria, we select its radius at random according to a radius distribution model. Works on bubble entrapment by rain [Pumphrey and Elmore 1990] and ocean waves [Deane and Stokes 2002] suggest that bubbles are created in a power law ($r^{-\alpha}$) distribution, where α determines the ratio of small to large bubbles. In nature, the α takes value from 1.5 to 3.3 for breaking ocean waves [Deane and Stokes 2002] and ≈ 2.9 for rain [Pumphrey and Elmore 1990], thus in simulation it can be set according to the scenario. The radius affects both the oscillation frequency (Eqn. 1) and the initial excitation (Eqn. 5) of the bubble. Plugging in the initial excitation factor ϵ selected by Eqn. 6, the sound for the bubble can be fully determined by Eqn. 7. Combining the genration criteria and the radius distribution model, our approach approximate the number of sound sources and the characteristics of their sounds plausibly in a physically-based manner for a dynamic scene.

4. INTEGRATION WITH FLUID DYNAMICS

There are many challenging computational issues in the direct coupling of fluid simulation with sound synthesis. As mentioned earlier, the three commonly used categories of 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. [2007; 2007] and Hess [2007]. The other is a hybrid grid-SPH approach, taken heavily from the work of Hong et al. [2008]. We present a brief overview of the fluid simulator methods below and describe how we augment the existing fluid simulation methods to generate audio. We refer the reader to [Thürey et al. 2007; Hess 2007; Hong et al. 2008] for full details on the fluid dynamics simulations.

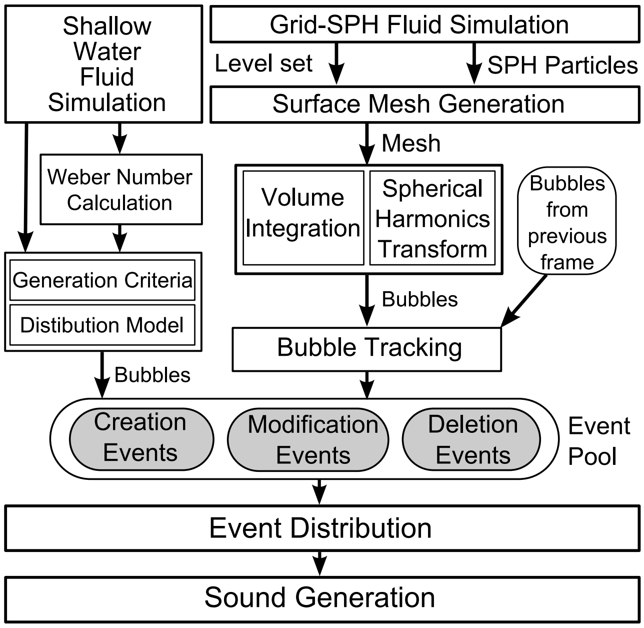


Fig. 4. An overview of our liquid sound synthesis system

4.1 Shallow Water Method

4.1.1 *Dynamics Equations.* The shallow water equations approximate the full Navier-Stokes equations by reducing the dimensionality from 3D to 2D, with the water surface represented as a height field. This approximation works well for situations where the velocity of the fluid does not vary along the vertical axis and the liquid has low viscosity. The height field approximation restricts us to a single value for the fluid along the vertical axis, making it unable to model breaking waves or other similar phenomena.

The evolution of the height field, $H(x,t)$, in time is governed by the following equations:

$$\frac{\partial H}{\partial t} = -v \cdot \nabla H - H\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}\right)$$

$$\frac{\partial v_x}{\partial t} = -v \cdot \nabla v_x - g\frac{\partial H}{\partial x}$$

$$\frac{\partial v_y}{\partial t} = -v \cdot \nabla v_y - g\frac{\partial H}{\partial y}$$

where we assume the gravitation force, $g = (0,0,g)^T$ is along the z-axis and v is the horizontal velocity of the fluid. We use a staggered grid of size $N_x \times N_y$ with equal grid spacing Δx and use a semi-Lagrangian advection step to solve the equations.

4.1.2 *Rigid Bodies.* Due to the 2D nature of the shallow water equations, rigid bodies must be explicitly modeled and coupled to the fluid simulation. This is complicated by the fact that our rigid bodies are 3D, whereas, our fluid simulation is 2D. We therefore cannot apply the method for fluid-rigid body coupling presented in previous works [Carlson et al. 2004; Batty et al. 2007; Robinson-Mosher et al. 2008], as our cells encompass an entire column of water and it is unlikely a rigid body will be large enough to fill a full vertical column. To that end, we explicitly model the interactions between the fluid simulation and the rigid body simulation using two one-way coupling steps.

The rigid body is coupled to the fluid in two ways, a buoyancy force and drag and lift forces resulting from the fluid velocity. The buoyancy force is calculated by projecting the area of each triangle up to the water surface, counting downward facing triangles positive and upward facing ones negative. The resulting force is calculated as,

$$f_{bouy} = -g\rho \sum_{i=1}^{n} -sign(n_i \cdot e_z)V_i,$$

where ρ is the density of the fluid, n_i and V_i are the normal and projected volume of triangle i and e_z points in the upward direction. The drag and lift forces are also calculated per face and point opposite and tangential to the relative velocity of the face and the fluid, respectively. Exact equations can be found in [Hess 2007].

The fluid is coupled to the object in two ways as well, through the surface height and the fluid velocity. The height is adjusted based on the amount of water displaced by the body on a given time step. This is again calculated per face, but this time the face is projected in the direction of the relative velocity. This can create both positive and negative values for the volume displaced, which is desirable for generating both the wave in front of a moving body and the wake behind. The fluid velocity of the cells surrounding a rigid body are adjusted as the water is dragged along with the body. The adjustment is calculated using the percentage of the column of water filled by the rigid body, the relative velocities and a scaling constant. More details can again be found in [Hess 2007].

4.2 Grid-SPH Hybrid Method

4.2.1 *Dynamics Equations.* We use an octree grid to solve the invicid incompressible Navier-Stokes equations [Losasso et al. 2004], which are

$$u_f + (u \cdot \nabla)u + \nabla p/\rho = f$$
$$\nabla \cdot u = 0$$

where u is the fluid velocity, p is the pressure, ρ is the density and f is the external forcing term. Although this provides a highly detailed simulation of the water, it would be too computationally expensive to refine the grid down to the level required to simulate the smallest bubbles. To resolve this, we couple the grid-based solver with bubble particles, modeled using SPH particles [Müller et al. 2003; Müller et al. 2005; Adams et al. 2007]. The motion of the particles is determined by the sum of the forces acting on that particle. The density of particles at a point, i, defined as $\rho_i = \sum m_j W(x_{ij}, r_j)$ where $W(x, r)$ is the radial symmetric basis function with support r defined in [Müller et al. 2003] and m_j and r_j are the mass and radius of particle j. We model the interactions of the bubbles with the fluid simulator and each other through a series of forces acting on the bubble particles:

(1) A repulsive force between particles to model the pressure between air particles, that drops to zero outside the support $W(x, r)$
(2) Drag and lift forces defined in terms of the velocity at the grid cells and the radius and volume of the particles, respectively
(3) A heuristic vorticity confinement term based on the vorticity confinement term from [Fedkiw et al. 2001]
(4) A cohesive force between bubble particles to model the high contrast between the densities of the surrounding water and the air particles
(5) A buoyancy force proportional to the volume of the particle

To model the effects of the bubbles on the water, we add the reactionary forces from the drag and lift forces mentioned above as external forcing terms into the incompressible Navier-Stokes equations given above.

4.2.2 *Bubble Extraction.* 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 4.2.4. 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 (as discussed in Sec. 3.2). Once we have the desired number of spherical harmonics, we determine the resonant frequencies using Eqn. (9).

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 we simply use the radius and Eqn. (7) to generate the sound. 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 single bubbles 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 for a level set bubble to find the spherical harmonics and generate audio.

4.2.3 *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 speed of a bubble. We then define `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 `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 approach because of its generality and its ability to uniformly handle both level set and SPH bubbles, as well as other types of fluid simulators.

4.2.4 *Spherical Harmonic Decomposition.* In order to decompose a mesh, M, into a set of the spherical harmonics that approximate it, we assume 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. (8).

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. (8). 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.

4.3 Decoupling Sound Update from Graphical Rendering

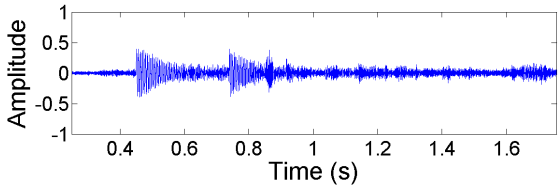
Since computing the fluid dynamics at 44,000 Hz, the standard frequency for good quality audio, would add an enormous computation burden, we need to reconcile the difference between the fluid simulator time step, T_{sim} (30-60 Hz), and the audio generation time step, T_{audio} (44,000 Hz). We can use Eqns. (1) and (9) to calculate the resonant frequency at each T_{sim} and then use Eqns. (7) and (12) to generate the impulse response for all the T_{audio}'s until the subsequent T_{sim}. Naively computing the impulse response 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 ~733 T_{audio} steps between the current and last T_{sim}.

5. IMPLEMENTATION AND RESULTS

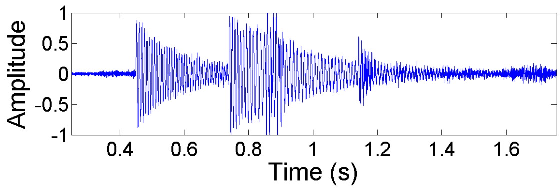
The rendering for the shallow water simulation is performed in real time using OpenGL with custom vertex and fragment shaders while the rendering for the hybrid simulator is done off-line using a forward ray tracer. In both cases, once the amplitude and frequency of the bubble sound is calculated, the final audio is rendered using The Synthesis ToolKit [Cook and Scavone].

5.1 Benchmarks

We have tested our integrated sound synthesis system on the following scenarios (as shown in the supplementary videos).



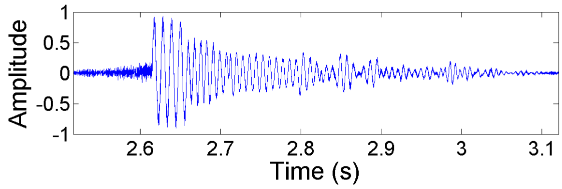
(a) Spherical Harmonic Decomposition



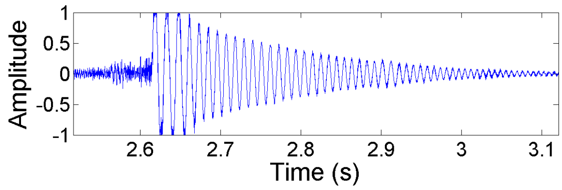
(b) Minimum Enclosing Sphere

Fig. 5. Wave plots showing the frequency response of the pouring benchmark. We have highlighted the moments surrounding the initial impact of the water and show our method (top) and a single-mode method (bottom) where the frequency for each bubble is calculated using the volume of the minimum enclosing sphere.

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 smaller 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. An average of 11,634 bubbles were processed per simulation frame to generate the sounds. Fig. 5 shows plots of the sound generated using our method and a single-mode version using the volume of the minimum enclosing sphere to calculate the volume.
Five Objects: In this benchmark, shown in Fig. 7, five objects are dropped into a tank of water in rapid succession, creating many small bubbles and a 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 in a single frame. Fig. 6 shows the wave plots for our method and the minimum enclosing sphere method. As one can observe, using the spherical harmonic decomposition creates a fuller sound, whereas the minimum enclosing sphere method creates one frequency that decays over time.



(a) Spherical Harmonic Decomposition



(b) Minimum Enclosing Sphere

Fig. 6. Wave plots showing the frequency response of the five objects benchmark. We have highlighted the impact of the final, largest object. The top plot shows our method and the bottom, a single-mode method where the frequency for each bubble is calculated using the volume of the minimum enclosing sphere.

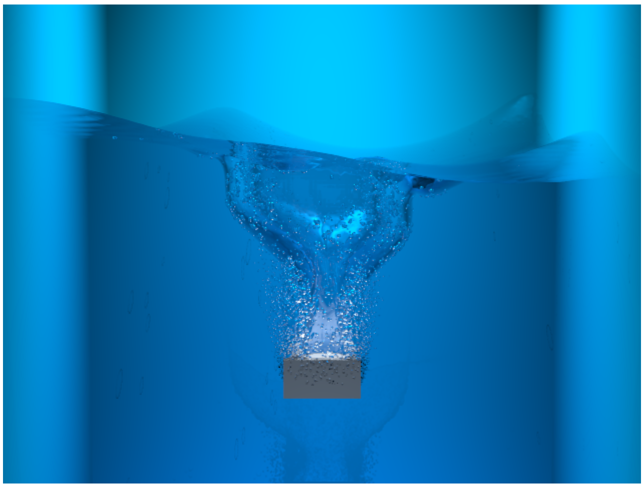
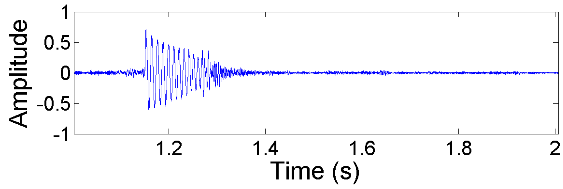


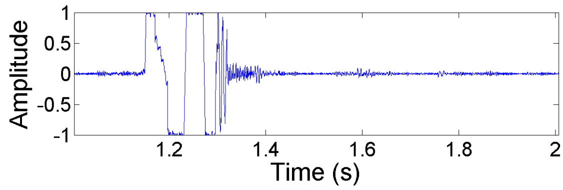
Fig. 7. Sound is generated as five objects fall into a tank of water one after another.

Dam Break: In this benchmark, shown in Fig. 9, we simulate the ”dam break” scenario that has been used before in fluid simulation, however, we generate the associated audio automatically. We processed an average of 13,589 bubbles per frame using five spherical harmonic modes. This benchmark also demonstrates the creation of a tube-shaped bubble as the right-to-left wave breaks, something that studies in engineering [Longuet-Higgins 1990] have shown to be the expected result of breaking waves. The creation of highly non-spherical, tube-like bubbles highlight the need for the spherical harmonic decomposition to handle bubbles of arbitrary shapes. This is illustrated in the supplementary video and Fig. 8, where the minimum enclosing sphere method creates a highly distorted wave plot when the tube-shaped bubble is created.

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.



(a) Spherical Harmonic Decomposition



(b) Minimum Enclosing Sphere

Fig. 8. Wave plots showing the frequency response for the dam break scenario. We highlight the moment when the second wave crashes (from right to left) forming a tube-shaped bubble. The top plot shows our method and the bottom, a single-mode method where the frequency for each bubble is calculated using volume of the minimum enclosing sphere.

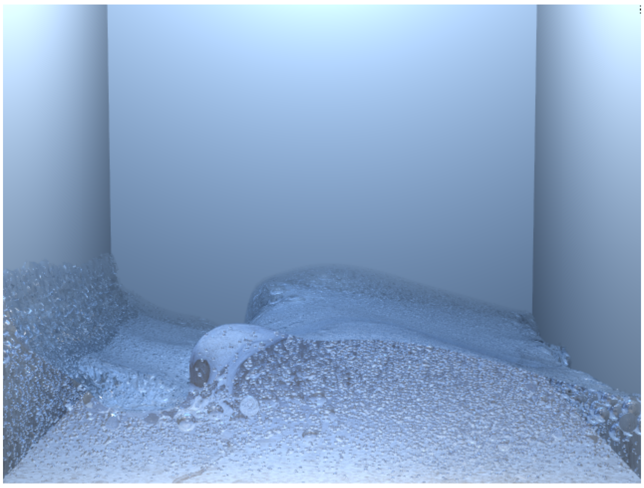


Fig. 9. A “dam-break” scenario, a wall of water is released, creating turbulent waves and sound as the water reflects off the far wall.

Duck: As shown in Fig. 11, 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.

5.2 Timings

Tables II and III show the timings for our system running on a single core of a 2.66GHz Intel Xeon X5355. Table II shows the number of seconds per frame for our sound synthesis method integrated with grid-SPH hybrid method. Column two displays the compute time of the fluid simulator [Hong et al. 2008]. Columns three, four and five break down the specifics of the synthesis process, and column six provides the total synthesis time. Column three represents the time spent extracting the bubble surface meshes from the level set and SPH particles (described in section 4.2.2). Column four is

Fig. 10. Real-time sounds are automatically generated from an interactive simulation of a creek flowing through a meadow.

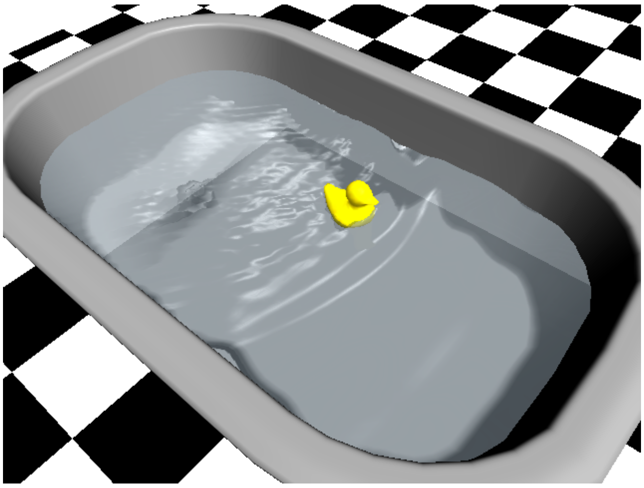


Fig. 11. Sounds are automatically generated as a (invisible) user moves a duck in a bathtub.

the time spent performing the spherical harmonic decomposition and spherical volume calculation (section 3.2) and column five is the time spent tracking the bubbles (section 4.2.3) and generating the audio (section 3).

Table III show the timings the shallow water simulator. Column one (Simulation) includes the time for both the shallow water simulation and the sound synthesis and column two (Display) is the time required to graphically render the water surface and scene to the screen. From the table we can see that both simulations run at around 55 frames per second, leaving compute time for other functions while remaining real-time.

Table III. Shallow Water Benchmark Timings (msec per frame)

	Simulation	Display
Creek Flowing	4.74 msec	12.80 msec
Duck in the Tub	7.59 msec	10.93 msec

5.3 Comparison with Harmonic Fluids

A quick comparison of the timings for our method vs. Harmonic Fluids [2009] shows that our shallow water sound synthesis technique runs in real time (including sound synthesis, fluid simulation, and graphical rendering). This makes our approach highly suitable for real-time applications, like virtual environments or computer games. It is also important to note that our benchmarks highlight more turbulent scenarios than those shown in [Zheng and James 2009], thus generating more bubbles *per simulation frame*. Our method also runs in a few seconds on a typical single-core PC, instead of many hours on a many-core platform (such as [Zheng and James 2009] for computing sound radiation). The most time-consuming step in our current implementation is surface extraction using a standard Marching Cubes algorithm [Lorensen and Cline 1987]. A more efficient variation of the Marching Cubes algorithm could offer additional performance improvements.

6. USER STUDY

To assess the effectiveness of our approach, we designed a set of experiments to solicit user feedback on our method. Specifically, we were looking to explore (a) the perceived realism of our method relative to real audio, video without audio and video with less than perfectly synched audio and (b) whether subjects could determine a difference and had a preference between our method and a simple approximation based on a single-mode bubble. The study consists of four parts, each containing a series of audio or video clips. Please refer to the supplementary video and the project website:

`http://gamma.cs.unc.edu/SoundingLiquids/`

for the set of comparison video clips used in our experiments. The next section details the procedure for each section of our user study.

6.1 Procedure

In sections I and II, each subject is presented a series of audio or video clips. In both cases, one clip is shown per page and the subject is asked to rate the clip on a scale from 1 to 10, where 1 was labeled “Not Realistic” and 10 “Very Realistic.” In sections III and IV, the subject is shown two audio or video clips side by side. In both cases, the subject is asked “Are these two audio/video clips the same or different?” If they respond “different”, we then ask “Which audio/video clip do you prefer?” and “How strongly do you feel about this preference?” The following sections detail the specific video and audio clips shown. In all the sections, the order of the clips is randomized and in sections III and IV, which clip appears on the left or the right is also random. The subject is also always given the option to skip either an individual question or an entire section and can, of course, quit at any time.

Section I: In this section the subject is shown a series of audio clips. The clips consist of five audio clips from our method and four real audio recordings of natural phenomena.

Section II: In this section, the subject is shown a series of video clips. These videos consist of the five benchmarks we produced, each shown with and without the audio we generated.

Table II. Hybrid Grid-SPH Benchmark Timings (seconds per frame)

	Average Bubbles per Frame	Fluid Simulation	Sound Synthesis			
			Surface Generation	Bubble Integration	Tracking & Rendering	Total
Pouring	11,634	1,259 s	10.20 s	1.77 s	0.18 s	12.15 s
Five Objects	1,709	1,119 s	2.37 s	0.21 s	0.94 s	3.52 s
Dam Break	13,987	3,460 s	39.92 s	1.45 s	1.13 s	42.50 s

Section III: Here the subject is presented with six pairs of audio clips. Each page contains the audio from one of our demo scenarios generated using the hybrid grid-SPH simulator paired with either the identical audio clip (to establish a baseline) or the same demo scenario using audio generated with the simplified, Minimal Enclosing Sphere method (denoted as MES in the table).
Section IV: This section is very similar to the previous experimental setup, however, we show the subjects the video associated with the audio they just heard. There are nine pairs of videos. Each page again contains the video and audio from one of our demo scenarios generated using the hybrid grid-SPH simulator paired with either the identical video clip (again, to establish a baseline), the video clip using the MES method or a video clip where we acted as the foley artist, mixing and syncing pre-existing audio clips to our video clip. By adding the video clip with pre-existing audio clips, we intended to evaluate the experience of using manually synched pre-recorded audio clips compared to the audio-visual experience of using our method.

6.2 Results

Table IV. Section I Results: Audio Only

	Mean	Std.	Mean Diff.	Std.
Beach	7.45	2.14	1.67	1.92
Raining	8.69	1.57	2.9	1.53
River	8.17	1.79	2.37	1.57
Splash	7.04	2.44	1.25	2
Pouring	4.74	2.33	-1.05	1.73
Five Objects	4.73	2.26	-1.07	1.52
Dam Break	4.92	2.17	-0.87	1.56
Brook	5.23	2.25	-0.56	1.88
Duck	6.69	2.18	0.89	1.75

The means and standard deviations for section I. Column one is the mean score given by the subject, whereas, column three is the mean of the difference a given question's score was from the mean score for this subject. We calculated this quantity in attempt to mitigate the problem of some subjects scoring all clips high and some subjects scoring all clips low. The top group represents the real sounds and the bottom group represents the sounds generated using our method. All 97 subjects participated in this section.

Tables IV, V, VI and VII show the results from Sections I - IV of our user study. In many of the subsequent sections we refer to the difference of means test. The test looks at the means and standard errors of two groups of subjects, and determines whether or not we can reject the null hypothesis that the difference we observe between the two means is the result of chance or is statistically significant. The formula for the difference of means can be found in most introductory statistics texts, but we present it below for reference:

$$t = \frac{\Delta M_{observed} - \Delta M_{expected}}{\sqrt{SE_1^2 + SE_2^2}}$$

Table V. Section II Results: Video vs. Visual Only

	Mean	Std.	Mean Diff.	Std.
Pouring	5.95	2.16	0.3	1.66
Pouring (No audio)	4.91	2.22	-0.65	1.7
Five Objects	6.65	2.18	1	1.57
Five Objects (No audio)	6.02	2.48	0.41	1.86
Dam Break	5.87	2.3	0.22	1.72
Dam Break (No audio)	5.36	2.48	-0.23	1.85
Brook	4.52	2.49	-1.13	1.84
Brook (No audio)	3.83	2.29	-1.78	1.61
Duck	6.3	2.45	0.65	2.23
Duck (No audio)	4.92	2.33	-0.7	2.01

The means and standard deviations for section II. Column one is the mean score given by the subjects, whereas column three is the mean of the difference a given question's score was from the mean score for this subject. A total of 87 out of 97 subjects chose to participated in this section.

where $\Delta M_{observed}$ is the difference of the observed means, $\Delta M_{expected}$ is the expected difference of the means (for the null hypothesis, this is always 0) and SE_1 and SE_2 are the standard errors for the two observed means (where $SE = \sigma/\sqrt{N}$). t is the t-value of that difference of means test. We choose a value of three on that t-distribution as our cutoff to determine statical significance.

6.2.1 *Demographics.* A total of 97 subjects participated in our study. They were allowed to quit during any section, at any time, so not all 97 completed all sections. 72% of our subjects where male and 28% were female. Their ages ranged from 17 to 65, with a mean of 25. About 82% of subjects owned an iPod or other portable music device and they listened to an average of 13 hours of music per week.

6.2.2 *Mean Subject Difference.* Tables IV and V show the two sections where the subject was asked to rate each video or audio clip individually. For those two sections, along with calculating a regular mean and standard deviation, we also computed a measure that we call the "mean subject difference." Some subjects tended to rate everything low, while some tended to rate everything high. Such individual bias could unnecessarily increase the standard deviation–especially since these ratings are most valuable when compared to other questions in each section. To calculate the mean subject difference, we first take the mean across all questions in a section for each subject, then instead of examining the absolute score for any given question we examine the difference from the mean. So, the mean values will be centered around 0, with the ones subjects preferred as positive.

6.2.3 *Section I and II.* Tables IV and V present a few interesting results. As we noted above, the subjects were allowed to skip any question or any section of the study. While 97 people participated in section I, only 87 participated in section II. In Table IV, the difference of means test clearly shows that the difference between the mean of the real sounds and the computer synthesized sounds is statistically significant. This difference is not surprising given the extra auditory clues that recorded sounds have that syn-

Table VI. Section III Results: Audio Only for Ours vs. Single-Mode

	Same	Diff	Prefer Ours	Prefer MES	Mean Strength Ours	Mean Strength MES
Pouring	21.8% (17)	78.2% (61)	68.9% (42)	31.1% (19)	6.36	5.42
Five Objects	27.6% (21)	72.4% (55)	54.7% (29)	45.3% (24)	5.86	5.17
Dam Break	2.6% (2)	97.4% (76)	77.3% (58)	22.7% (17)	7.29	5.82

Columns one and two show the percentage (and absolute number) of people who found our videos to be the same or different than the minimal enclosing sphere method. Columns three and four show, of the people who said they were different, the percentage that preferred ours or the MES method and finally columns five and six show the mean of the stated strength of the preference for those who preferred our method and the MES method. A total of 78 subjects participated in this section.

Table VII. Section IV Results: Video for Ours vs. Single-Mode(top) & Ours vs. Recorded(bottom)

	Same	Diff	Prefer Ours	Prefer Other	Mean Strength Ours	Mean Strength Other
Pouring	16.7% (12)	83.3% (60)	73.3% (44)	26.7% (16)	6.75	5.75
Five Objects	43.2% (32)	56.8% (42)	48.7% (19)	51.3% (20)	6.42	6.2
Dam Break	5.3% (4)	94.7% (71)	83.3% (55)	16.7% (11)	7.35	6.64
Pouring	1.4% (1)	98.6% (72)	65.7% (46)	34.3% (24)	7.13	6.79
Five Objects	1.3% (1)	98.7% (74)	94.4% (67)	5.6% (4)	8.75	5.33
Dam Break	2.8% (2)	97.2% (69)	60.6% (40)	39.4% (26)	7.65	7.19

The top group shows our method versus the minimal enclosing sphere method and the bottom group shows our method versus the prerecorded and synched sounds. Columns one and two show the percentage (and absolute number) of people who found the two videos to be the same or different. Columns three and four show, of the people who said they were different, the percentage that preferred ours or the other method (either MES or prerecorded) and finally columns five and six show the mean of the stated strength of the preference for those who preferred our method and the other method. A total of 75 subjects participated in this section.

thesized sounds lack. That said, the mean for the duck being moved interactively in the bathtub and the real splashing sound are not statistically different. In the best case, our method is able to produce sounds with comparable perceived realism to recorded sounds. In addition, in three recorded sounds (beach, raining and river), there are multiple sound cues from nature, such as wind, birds and acoustic effects of the space where the recordings were taken. We conjecture that the subjects tend to rate them higher because of the multiple aural cues that strengthen the overall experience. Therefore, although the perceived realism of our synthesized sounds is scored lower than the perceived realism of the recorded sounds, the fact that our synthesized sounds are no more than one standard deviation away from the recorded sounds without the presence of multiple aural cues is notable.

In Table V, two benchmarks have a statistically significant difference between the means of the video with and without audio: the duck in the bathtub and the pouring water demos. It shows that for these two cases, we can conclusively state that the sound effects generated using our method enhances the perceived realism for the subjects. Although the the results of other cases are statistically inconclusive, they show a difference in the means that suggests the perceived realism is enhanced by using audio generated using our methods.

When comparing the perceived realism of audio only, visual only, and visual with audio from Tables IV and V, we see that for demos with less realistic graphics, like the flowing creek and the duck in the tub, the combined visual-audio experience does not surpass the perceived realism of the audio alone. For benchmarks with more realistic rendering, this is not the case, suggesting that the subject's perception of realism is heavily influenced by the visual cues, as well as the audio.

6.2.4 *Our method vs. Single-Mode Approximation.* Based on the results from Tables VI and VII, subjects clearly preferred our method to the method using the minimal enclosing sphere approximation. We believe these studies suggest that when presented with a clear choice, the subjects prefer our method. In addition, the degree of preference, as indicated by the "mean strength" for our method is more pronounced. We also see that the percentage of people who were able to discern the difference between the sounds generated by our method vs. MES approximation is highest in the Dam-Break benchmark, where the bubbles were most non-spherical. Interestingly, Table VII shows their ability to discern the difference becomes less acute when graphical animation is introduced.

6.2.5 *Roles of Audio Realism and AV Synchronization.* We did not include the results for the comparisons of the same clips in Tables VI and VII, however, in each case close to 90% were able to detect the same video or audio clips. Earlier studies [van den Doel and Pai 2002; van den Doel 2005] suggested that the subjects were not necessarily able to detect the difference between single vs. multi-mode sounds or discern the same sounds when played again. Our simple test was designed to provide a calibration of our subject's ability to discern similar sounds in these sets of tests.

We can also see in Table VII that subjects reliably preferred our method to those videos using manually synchronized, recorded sounds of varying quality. This study shows that simply adding sound effects to silent 3D animation of fluids does *not* automatically improve the perceived realism – the audio needs to be both realistic and seamlessly synchronized in order to improve the overall audio-visual experience.

6.2.6 *Analysis.* From this study, we see several interesting results. First, although we feel this work presents a significant step in computer synthesized sounds for liquids, the subjects still prefer real, recorded audio clips when no additional sound cues were generated, as shown in Table IV. Second, Table V shows that our method appears to consistently improve the perceived visual-audio experience – most significantly in the case of interactive demos such as the rubber duck moving in a bath tub. Third, in side-by-side tests (Tables VI and VII top) for the audio only and audio-visual experiences, the subjects consistently prefer the sounds generated by our method over the sounds of single-sphere approximation. Finally, when audio is added to graphical animations (Table VII bottom), the audio must be both realistic and synchronized seamlessly with the visual cues to improve the perceived realism of the overall experience.

7. CONCLUSION, LIMITATIONS, AND FUTURE WORK

We present an automatic, physically-based synthesis method based on bubble resonance that generates liquid sounds directly from a fluid simulator. 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 user study suggests that the perceived realism of liquid sounds generated using our approach is comparable to recorded sounds in similar settings.

Although our method generates adequately 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. We also used a simplified model for the bubble excitation. Although no analytic solution exists, a more complex approximation could potentially help. Continued research on fluid simulations involving bubbles and bubble excitation would improve the quality and accuracy of the sound generated using our approach, specifically we expect that as fluid simulators are better able to generate the varied distribution of bubbles occuring in nature, the high frequency noise present in some of our demonstrations would be reduced.

For non-star-shaped bubbles, because they cannot be decomposed into spherical harmonics, we are forced to revert to the simple volume-based approximation. Since bubbles tend to be spherical (and rapidly become spherical without external forces), this happens rarely. It can, however, be see in the pouring water demo, when a ring-shaped bubble forms soon after the initial impact. There has been some recent work on simulating general bubble oscillations using a boundary element method [Pozrikidis 2004] and we could provide more accuracy for complex bubble shapes using a similar technique, but not without substantially higher computational costs.

REFERENCES

ADAMS, B., PAULY, M., KEISER, R., AND GUIBAS, L. J. 2007. Adaptively sampled particle fluids. In *ACM SIGGRAPH 2007 papers*. ACM, San Diego, California, 48.

BATTY, C., BERTAILS, F., AND BRIDSON, R. 2007. A fast variational framework for accurate solid-fluid coupling. *ACM Trans. Graph. 26,* 3, 100.

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.

CARLSON, M., MUCHA, P. J., AND TURK, G. 2004. Rigid fluid: animating the interplay between rigid bodies and fluid. In *ACM SIGGRAPH 2004 Papers*. ACM, Los Angeles, California, 377–384.

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.

FEDKIW, R., STAM, J., AND JENSEN, H. W. 2001. Visual simulation of smoke. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 15–22.

FLORENS, J. AND CADOZ, C. 1991. The physical model: modeling and simulating the instrumental universe. In *Representations of musical signals*. 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. Ph.D. 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. 1989a. Monopole emission of sound by asymmetric bubble oscillations. part 1. normal modes. *Journal of Fluid Mechanics 201*, 525–541.

LONGUET-HIGGINS, M. S. 1989b. 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. 1990. Bubble noise spectra. *The Journal of the Acoustical Society of America 87,* 2 (Feb.), 652–661.

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.

LOSASSO, F., GIBOU, F., AND FEDKIW, R. 2004. Simulating water and smoke with an octree data structure. In *ACM SIGGRAPH 2004 Papers*. ACM, Los Angeles, California, 457–462.

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.

MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, San Diego, California, 154–159.

MÜLLER, M., SOLENTHALER, B., KEISER, R., AND GROSS, M. 2005. Particle-based fluid-fluid interaction. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, Los Angeles, California, 237–244.

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.

PICARD, C., TSINGOS, N., AND FAURE, F. 2009. Retargetting example sounds to interactive Physics-Driven animations. *AES 35th International Conference, Audio in Games*.

PLESSET, M. AND PROSPERETTI, A. 1977. Bubble dynamics and cavitation. *9*, 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. *25*, 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*.

ROBINSON-MOSHER, A., SHINAR, T., GRETARSSON, J., SU, J., AND FEDKIW, R. 2008. Two-way coupling of fluids to rigid and deformable solids and shells. *ACM Trans. Graph. 27,* 3, 1–9.

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

SIRIGNANO, W. A. 2000. Fluid dynamics and transport of droplets and sprays. *Journal of Fluids Engineering 122,* 1 (Mar.), 189–190.

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., 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.

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.

TREBIEN, F. AND OLIVEIRA, M. 2009. Realistic real-time sound resynthesis and processing for interactive virtual worlds. *The Visual Computer 25,* 5 (May), 469–477.

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

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. AND PAI, D. K. 2002. Measurements of perceptual quality of contact sound models. In *Proc. of the International Conference on Auditory Display (ICAD 2002)*. Kyoto, Japan, 345—349.

ZHENG, C. AND JAMES, D. L. 2009. Harmonic fluids. In *ACM SIGGRAPH 2009 papers*. ACM, New Orleans, Louisiana, 1–12.