Graphical Models 103 (2019) 101028

Contents lists available at ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

Liquid-solid interaction sound synthesis

Haonan Cheng[a], Shiguang Liu[a,b,*]

[a] School of Computer Science and Technology, Division of Intelligence and Computing, Tianjin University, Tianjin 300350, China
[b] Tianjin Key Laboratory of Cognitive Computing and Application, Tianjin 300350, China

ARTICLE INFO

CCS concepts:
Computing methodologies, Computer graphics
Applied computing, Sound music computing

Keywords:
Computer animation
Sound synthesis
Liquid-solid interaction

ABSTRACT

Liquid-solid interaction produces a characteristic sound which is different from the sound of bubbly water flows. This paper explores liquid-solid interaction sound (LSIS) synthesis. The resulting LSIS consists of two components: bubble sound and impact sound, which corresponding to liquid-liquid collision and liquid-solid collision, respectively. To improve the quality of LSIS, we propose a novel sound enrichment method called Feature Transfer Synthesis (FTS), which is designed to compensate for the differences between the real-world recording and the synthesized sound. We also greatly resolve the synchronization problem during blending the two components of LSIS through a key frame algorithm with normal force and grid surface. Moreover, a generalized dipole model for sound radiation is performed to estimate the LSIS pressure at a listener position. We illustrate our approach through a series of experiments and a perceptual user study, demonstrating the utility of our LSIS synthesis pipeline in producing realistic sounds at practical computational times.

1. Introduction

Liquid is common in natural environments, and different forms of liquid would produce different sound effects. Recently, sound simulation [1,4,16,19,21,28,30] attracts more and more interests from researchers in computer graphics community. However, little attention has been paid to sound synthesis of liquid-solid interaction, e.g., water drops falling onto disks with different materials. In this paper, we seek to automatically synthesize the liquid-solid interaction sound (LSIS), which may help distinguish the liquid sounds of different interaction scenes.

In the existing methods of liquid acoustic research, it is known that liquid sound is produced by the bubble vibration caused by liquid-liquid collision. Thus, liquid sound is usually simulated by the harmonic bubble-based method [12,29]. In the process of bubble sound simulation, all of the different forms (spherical or non-spherical) of bubbles [19] and the positions of bubbles [13] in liquid may affect the resulting liquid sound. However, to simulate the LSIS, we need to consider not only the bubble sound generated by bubble vibrations, but also the impact sound caused by liquid-solid collision. To the best of our knowledge, we take the first step to simulate LSIS, instead of just considering liquid sound like previous work. Our approach is positioned to offer practical runtimes and richer, more recognizable LSIS (Fig. 1).

Although there are several researches [2,3,5–7,22] focusing on the solid-solid collision, liquid-solid collision is quite different from the collision between solids. The collision area between liquid and solid is dynamically evolving due to the fluid's characteristics of deformation. One of the main challenges of liquid-solid interaction is that the solid modes are changing continuously (because it is easier for the solid to move air than water). What's more, another challenge for LSIS synthesis is that liquid-solid interaction is a composition of myriad vibration events which incorporates both bubble and solid. Besides, the spectrum of the LSIS depends on the vibration modes of the solid and variation modes of bubble, but the modes are various. That is, liquid-solid interaction may cause myriad and various modes which are visually imperceptible yet audible when liquid impacts on the solid surface. For example, it is difficult to simulate the entire range of collision modes by mesh animation when a water ball falls into a metal box. Brute force analysis of the collision events for each sound clip is expensive, while simply superimposing two types of sounds would produce unrealistic results. The above two major challenges should be overcome to make LSIS practical, which are the focuses of our research.

For the aforementioned challenges, this paper proposed LSIS synthesis, which is a new recording-driven sound synthesis framework accounting for different solid surfaces. LSIS consists of two components: bubble sound and impact sound. The bubble sound and the impact sound are simulated based on the physical liquid-solid model in order to be consistent with the continuous change of input animations. To generate myriad and various modes, a feature transfer synthesis (FTS) is specially designed for sound enrichment. FTS is proposed to compensate for the differences between the real-world recording and sound synthesized. To transfer the acoustic features, we convert the sound signal to an image via short time Fourier transform (STFT) and modify the color transfer algorithm to the spectrograms. To enhance the power of frequency on

[*] Corresponding author.
E-mail address: lsg@tju.edu.cn (S. Liu).

https://doi.org/10.1016/j.gmod.2019.101028
Received 18 July 2018; Received in revised form 22 March 2019; Accepted 21 April 2019
Available online 25 April 2019

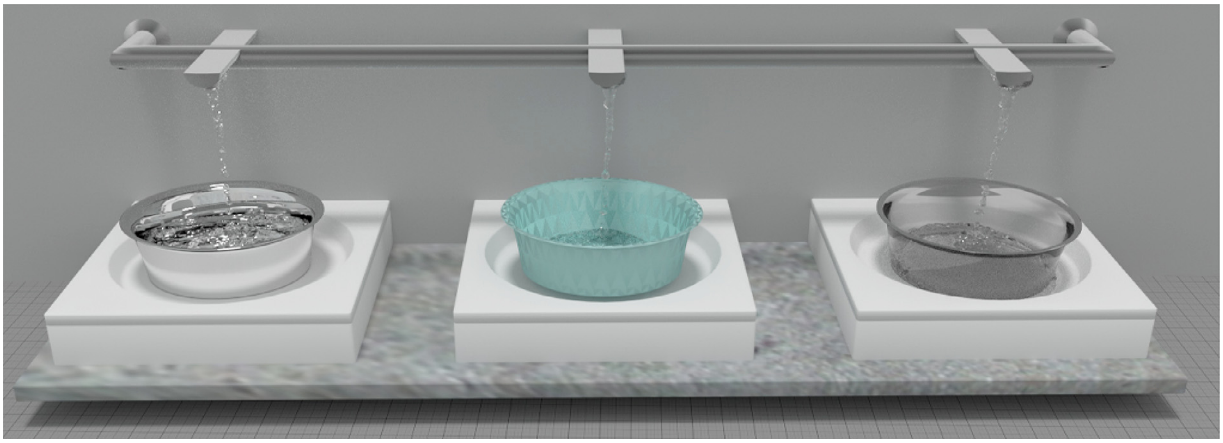


Fig. 1. An example of interaction sound between pouring faucet and basins of different materials. Given an animation of liquid-solid interaction, our system is able to synthesize accompanying foley with different material-aware sound effects.

local pixels, we incorporate the local constraint and position constraint. Later, by using the inverse short time Fourier transform (ISTFT), we can get the enriched sound which contains both recording and synthesized sound features. Thus, we can fully adapt to the range of frequencies and damping of all modes. To estimate a LSIS pressure at a listener position, we perform a generalized dipole model with surface vibration data provided by our animation simulation. Moreover, in order to further resolve the synchronization problem during blending myriad vibration events of the LSIS in an efficient way, we seek to develop a key frame blending algorithm with normal force and grid surface to determine the weight of impact sound. Our method offers the following contributions:

1. A new framework is proposed to synthesize the LSIS for liquid-solid interaction animation. To the best of our knowledge, this is the first attempt to synthesize LSIS accounting for different solid materials.

2. We introduce a novel data-driven binary constrained FTS method, which can greatly enhance the quality of LSIS and distinguish among different solid materials.

3. A key frame blending algorithm with normal force and grid surface is developed to efficiently synchronize the bubble sound and the impact sound with liquid-solid interaction animation.

2. Related work

Fluid simulation has been carried out for several years which can be found widespread success applications [10,15,26,31] in computer graphics and animation. Thus, to achieve more realistic animation, more and more attention has been paid to fluid sound simulation in the computer graphics community. In this paper, we focus on LSIS synthesis which can be classified into two parts, namely the liquid sound synthesis and the impact sound synthesis.

2.1. Liquid sound synthesis

Investigation of liquid sounds produced by bubbles dates back almost a century, however, relatively little work has been done on simulating them. Imura et al. [12] proposed a harmonic bubble based sound synthesis method, unfortunately, this method cannot capture the time-varying spatial structure of three-dimensional sound radiation information. Zheng and James [29] proposed a practical method of automatically synthesizing synchronized liquid sounds from three-dimensional fluid animations. Although this method can synthesize liquid sounds synchronized with the fluid animation, the time consumption is huge because of the complexity of the bubble calculation algorithm. Subsequently, Moss et al. [19] proposed a new physics-based harmonic bubble modelling method. This method takes into account the bubble shapes and proposes complex non-spherical bubbles based on Leighton's bubble theory [14,17,20]. Later, Langlois et al. [13] proposed a complex sound of bubble synthesis technique to deal with the problems of the two aforementioned methods. However, since the bubble shape varies, consideration of the shape with each bubble in one frame would significantly increase the time consumption, and users cannot tell the sound differences generated by the bubble shape. Without affecting the auditory effect, we simplify the sound model for efficiency. What's more, different from the earlier methods of liquid sound synthesis, for LSIS synthesis, we need to consider not only the liquid sound generated by bubble vibrations, but also the impact sound generated by solid vibrations. Thus, our approach is positioned to calculate the vibration of solid which is caused by fluid motion and offer more real sound for collision events between liquid and solid surface.

2.2. Impact sound synthesis

In the last couple of decades, there has been strong interest in digital sound synthesis with modal synthesis techniques for simulating rigid-body sound. The modal model is used to generate impact, rolling, and sliding sounds. We need to consider the vibration of solid surface to generate LSIS, thus our approach is closely related to previous modal synthesis techniques. Van Den Doel et al. [5] used modal models derived from sound samples captured by striking an object at different locations on its surface. For rigid body vibration, the standard linear modal synthesis technique [25] is often used in dynamic deformation model. Subsequently, Ren et al. [24] adopted tetrahedral finite element models and proposed the modal sound synthesis method for different materials. However, this method is only suitable for the rigid-body sounds, and it is not fully applicable to liquid-sound interaction. Moreover, since the fluid is easy to deform, the collision area between liquid and solid surface is dynamically evolving during the collision events which we should solve in our algorithm.

3. Algorithm overview

Our LSIS synthesis pipeline is illustrated in Fig. 2. Firstly, a physically based liquid model [9] is used to simulate the animation of liquid-solid interaction. Then, the fluid field and surface vibration are exported from the liquid-solid interaction model to evaluate the velocity divergence integral. With the given liquid-liquid collision and liquid-solid collision model, we synthesize the bubble sound and impact sound, respectively. Actually, the liquid-solid interaction simulation has limited spatial and temporal resolution to synthesize detailed sound result. To address this problem, we design a new sound enrichment method named

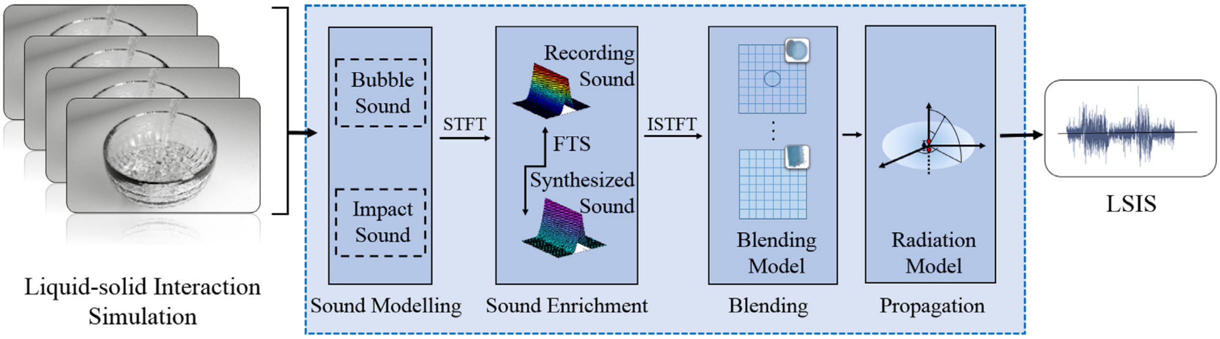


Fig. 2. The LSIS synthesis pipeline. We present an end-to-end solution for simulating LSIS as shown in the blue region. The pipeline can be divided into three major components: sound modelling, FTS, and synchronization. All the components are automatic. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

FTS through feature transfer to enrich the details of the sound. We utilize STFT and ISTFT to complete the conversion of sound signals and spectra. After obtaining the FTS results, we further elegantly synchronize the two parts of the sounds so as to obtain the LSIS. A key frame blending algorithm with normal force and grid surface is proposed for synchronization. Moreover, to estimate the acoustic pressure at listener position, we seek to resolve the sound radiation through a dipole sound model and obtain the final LSIS.

4. Sound modelling

We divide the LSIS into the bubble sound from liquid-liquid collision and impact sound from liquid-solid collision, respectively. Therefore, we will first analyze the physical mechanism of the two collisions and calculate the corresponding acoustic functions.

4.1. Bubble sound

Liquid-liquid collision. For the liquid-liquid collision, although fluid sounds can arise via numerous mechanisms, harmonic bubble based fluid sounds[14] come with almost all kinds of fluid movement. In the case of fluids, sound is primarily generated by bubble formation and resonance, creating pressure waves that travel through both the liquid and air media to the listeners. Thus, we utilize bubble sound to simulate the sound of the liquid-liquid collision. For implementation, we estimate pointlike bubble creation rates and size distributions with ad-hoc stochastic models [8].

Acoustic bubbles. The varying of acoustic bubbles over time in real situations is very complex. In order to ensure the efficiency of the algorithm, the sound modelling of fluid is simplified. However, for important missing sound details, we enrich them by FTS, which will be introduced in Section 5. Therefore, in this paper, the spherical and non-spherical bubbles are modelled as a same sound model. The simplified sound model does not affect the audio results. We show this in the accompanying video. Besides, in a similar experimental environment, the liquid sound synthesized by Langlois et al.'s [13] method needs to take hour as measurement unit, but our simplified model only takes a few seconds.

The Minnaert's formula [18] deduces the resonant frequency of perfectly spherical bubbles, and provides a physics-based approach to generating sound. The sound produced by bubbles can be determined by the resonant frequency. The Minnaert's formula is written as follows:

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

where γ represents the specific heat ratio of the gas, p_0 expresses the gas pressure inside the bubble, ρ is the fluid density, and f_0 denotes the frequency. For air bubbles in water at an atmospheric pressure, Eq. (1) can be simplified as $f_0 r_0 \approx 3\mathrm{m/s}$. We model the bubbles in liquid as damped harmonic resonators and use the Minnaert frequency for resonation. The impulse response equation is

$$S_{Bubble}(t) = A_0 \sin(2\pi f(t)t)e^{-\beta_0 t} \tag{2}$$

where A_0 is determined by the initial excitation of the bubble, and $\beta_0 = \pi f_0 \delta_{tot}$ represents the rate of damping δ_{tot} due to damping. In the standard harmonic oscillator, according to Doel et al. [8] and Moss et al. [19], we use $f(t)$ to substitute f_0, and $f(t) = f_0(1 + \xi\beta_0 t)$, where $\xi \approx 0.1$ by Doel's work. As the bubble survives and grows closer to the surface, it helps to avoid the situation that sound pressure for bubble becomes infinite by adjusting the frequency $f(t)$. The final sound pressure for bubbles in liquid can be expressed as

$$S_{Bubble}(t) = A_0 \sin[2\pi f_0(1 + \xi\beta_0 t)t]e^{-\beta_0 t} \tag{3}$$

4.2. Impact sound

Liquid-solid collision. From the point of view in Langlois et al.'s [13] work, the liquid-solid coupling sounds is an important sound source which have not been explored. Small-scale vibration of solid surface caused by liquid leads to variation in air pressure, which propagates sounds to listeners. However, for real-time applications, linear modal sound synthesis has been widely adopted to synthesize impact sounds. Thus, we utilize the modal model to synthesize impact sound for simulating the sound of liquid-solid collision. The details of parameter transfer between rigid body collision and modal synthesis can be found in [22].

Modal synthesis. To synthesize the impact sound, we calculate the solid material parameters from the real collision sound generated by solids, then apply the material parameters to the collision phenomenon. In order to model the impact sound, we calculate the modal parameters according to multi-level sound spectrum extraction which was inspired

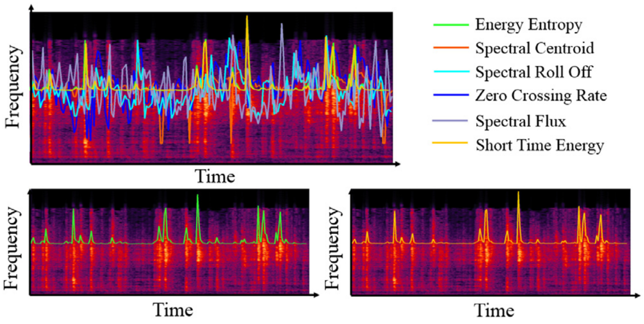


Fig. 3. Spectrogram of original sound signal and six signal features. The original signal is a 32s sound of pouring water onto a metal surface. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

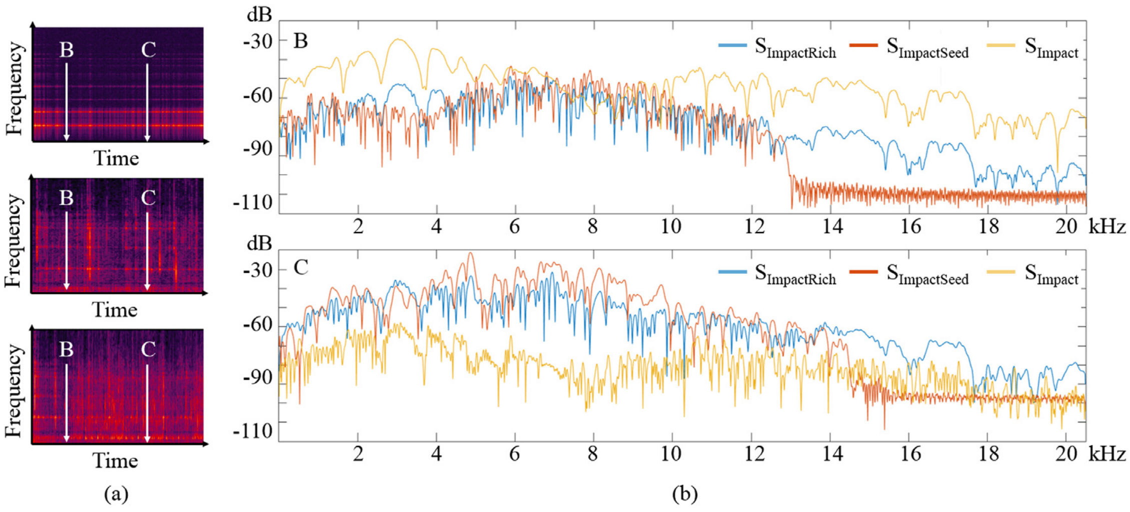


Fig. 4. Spectra comparison between S_{Impact}, $S_{ImpactSeed}$ and $S_{ImpactRich}$. (a) illustrates the spectra for S_{Impact}, $S_{ImpactSeed}$ and $S_{ImpactRich}$. (b) shows frequency analysis plots at arrows (B and C) in (a). The top row in (b) corresponding to the frequency value at time B and the bottom row corresponding to the frequency value at time C. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

by Ren et al.'s method [24].To generate richer modal sound without resort to expensive high resolution simulation, we stochastically inject small-scale collision events such that each collision event corresponds to a mode. The combination of frequency, damping, and amplitude defines the characteristics ϕ of mode i: $\phi_i = (f_i, d_i, a_i)$. The value of ϕ_i depends on the solid material properties, geometry and interaction at runtime. We extract parameters f_i, d_i, and a_i from the real solid collision sound generated by solid with different materials, where f_i indicates the frequency of the mode, d_i represents the damping coefficient, a_i is the excitation amplitude, θ_i represents the initial phase.

Since we will enrich the impact sound by FTS, different from the greedy manner in [24], we only track the top three peak values of the original sound signal. Then, for each extracted mode, we first select a suitable level of the sound spectrum and search for the peak in the sound spectrum, and evaluate the estimated value to obtain the modal parameters. We briefly describe the process of the modal synthesis. The energy spectral density feature of a single mode decaying sinusoidal curve contains peak features. Thus, the sound energy spectral density of the signal is first obtained from the sound waveform in time-domain, and then the peak feature can be found. In order to obtain the modal parameters, we evaluate the decaying sinusoidal model parameters, and the parameter value is $\overline{\phi}_i = (\overline{f}_i, \overline{d}_i, \overline{a}_i)$. Through the iterative adjustment of $\overline{\phi}_i = (\overline{f}_i, \overline{d}_i, \overline{a}_i)$, we can calculate the error of the result of the real energy spectrum parameters and estimated values. If the total error of the two parts is less than a certain threshold, we regard the estimated value $\overline{\phi}_i = (\overline{f}_i, \overline{d}_i, \overline{a}_i)$ as the modal parameter of the current mode.

By analysing the sound energy spectral density, the modal parameters of modes can be obtained. We integrate them to get the final impact sound as follows:

$$S_{Impact}(t) = \sum_i a_i e^{-d_i t} \sin(2\pi f_i t + \theta_i) \quad (4)$$

5. Sound enrichment

We can get plausible LSIS through the above procedures by combining the bubble sound and impact sound. However, when comparing with the recording, we can observe that there exist more sound details than the synthesized sound. What's more, during the liquid-solid interaction, the solid modes are continuously changing. These problems are limited to fluid simulations. Although the detail retention of fluid modelling is getting higher and higher, there are a lot of things that are not resolved in fluid simulations with concern to sound (very small bubbles, bubble popping, etc.) And for liquid-solid interaction animation, there are a lot of insignificant collision events which are visually imperceptible yet audible that lead to detailed recorded sound. So, it is challenging to estimate the parameters from limited liquid-solid interaction model. Thus, in this section, we propose a sound synthesis method for enriching the synthesized sound details. The main idea of the method is regarding the sound spectrogram as an image and exploiting image color transfer techniques to transfer the statistical aspect of a recorded sound to a synthesized one.

In this section, we take the impact sound as an example, and the synthesis process of enriching bubble sound is the same. Given the fluctuations in the original impact sound recordings, there are a lot of unrelated, small fluctuations in them, which can influence the effect of the feature transfer. Thus, we need to extract the sound clip with salient features. We test the six common signal features: spectral roll off, spectral flux, energy entropy, spectral centroid, short time energy and zero crossing rate. The top row of Fig. 3 illustrates the six curves of sound features. In the spectrogram, the color is used to identify intensity which the brighter the color is, the stronger the intensity is. Hence, through the curves, we observe that energy entropy and short time energy can describe the changes of impact sound in a better way as illustrated in

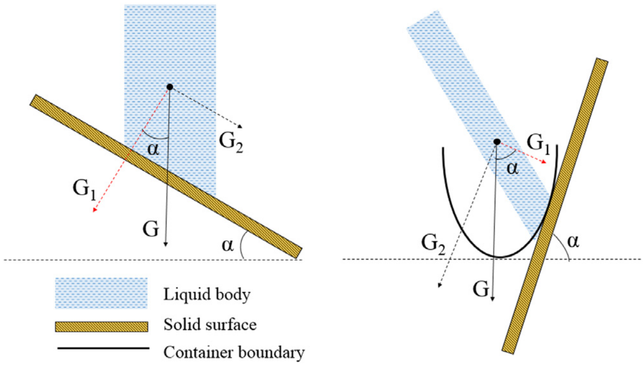


Fig. 5. Normal force calculation for different bottoms of containers with different shapes. The left illustrates the force analysis of liquid hitting on a flat bottom. The right presents the force analysis of liquid hitting on a non-flat bottom. The red arrows in each figure represent the normal force. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

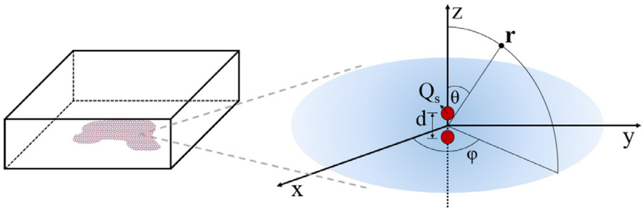


Fig. 6. (Left) A 3D diagram of the interaction area at time *t*. (Right) A dipole sound field geometry and notation.

Table 1
Physical constants used in our simulations.

Parameter	Value	Description
ξ	0.1	rise factor for frequency
ρ	1000 kg/m^3	water density
γ	1.4 J/(kgJ)	specific heat ratio of air
p_0	101.325 kPa	atmospheric pressure
c_0	343 m/s	sound speed in air
ρ_0	1.29 kg/m^3	atmospheric density

the bottom row of Fig. 3. Consequently, we calculate the energy entropy and short time energy for the impact sound recordings. Then we calculate the inflection point of the two curves and extract the impact sound with salient timbral features when the inflection point appears. After the extraction, we can obtain the feature seed (a short recording)[1] from the original input recording of the impact sound.

After we get the feature seed (a 3s recording) of impact sound $S_{ImpactSeed}$ and the synthesized impact sound S_{Impact}. A feature transfer synthesis (FTS) method is proposed to enrich the synthesized impact sound. Firstly, the two input acoustic signals $\{S_{ImpactSeed}, S_{Impact}\}$ are processed as input by applying short term Fourier transform (STFT), respectively. The equation for STFT is shown as follows:

$$STFT_x(\tau,\omega) = \int_t [x(t)W(t-\tau)]e^{-j\omega t}dt \tag{5}$$

where $x(t)$ is the input signal to be analysed, W is the windowing function (Hamming window in our experiments) and τ is the length of time window. Through the STFT, we can get the time-frequency spectrogram which can be viewed as a 2-D image as shown in Fig. 5(a). After we convert the sound signal to an image via STFT, then we can transform the features by image processing techniques. In the following sections, we define the spectra corresponding to feature seed of impact sound $S_{ImpactSeed}$ and synthesized impact sound S_{Impact} represented by the image as $I_S(\mathbf{o})$ and $I_I(\mathbf{o})$ as follows:

$$S_{ImpactSeed}(t) \overset{STFT}{\to} I_S(\mathbf{o}),\ S_{Impact}(t) \overset{STFT}{\to} I_I(\mathbf{o}) \tag{6}$$

$\mathbf{o} = (t, f)$ is a two-dimensional coordinate on the time-frequency (discrete) domain. The spectrogram through the STFT can be viewed as a color map that each pixel value in $I_S(\mathbf{o})$ and $I_I(\mathbf{o})$ indicates the power of the frequency f at the time t. The goal of our work is to compensate for the differences between $S_{ImpactSeed}$ and S_{Impact}. Thus, this means that we would like to transfer the statistical aspects of a recorded sound to a synthesized one between spectra. So we turn the sound enrichment problem into a special color transfer problem.

However, different from general images, in a spectrogram, the intensity of the color denotes the power of the frequency and the position of the color represents the distribution of the frequency. Therefore, in our FTS, we need not only to transfer color values, but also to transfer color distributions. In order to achieve this, we design two constraints in the FTS algorithm. We will describe the two parts in detail in the following two sections.

Table 2
Timings for different scenarios (Fig. 12).

Scenarios	Time consumption (s)		
	FTS	Synchronization	Total
metal basin	0.404(0.360)	0.058(0.044)	0.771
glass bowl	0.410(0.363)	0.057(0.043)	1.027
plastic basin	0.548(0.496)	0.058(0.045)	1.212
wood plate	0.347(0.312)	0.052(0.041)	0.673

Table 3
Recording statistics.

Material	Container	Time (s)	Identifier
Metal		3.0 (30.8)	ME
Wood		3.0 (29.1)	WO
Plastic		3.0 (24.7)	PL
Glass (pouring)		3.0 (33.2)	GLP
Glass (shaking)		3.0 (33.2)	GLS
Porcelain		3.0 (25.1)	PO
Liquid		3.0 (31.2)	LI

5.1. Local constraint

Given the spectra after the STFT, we will first transfer the intensity of the color which correspond to the power of the frequency in sound signals. Reinhard et al. [23] proposed a classical global algorithm for color transfer which matches the average and variance. However, it is not suitable for spectrogram to add the high frequency details missing form synthesized sound. The reason is that the distribution of the frequency-power in the spectrogram is relatively sparse, the overall color transfer to the spectrogram will diminish the strength of the features. Thus, we effectively incorporated the local mean and standard deviation constraint on local pixels in order to enhance the power of the acoustic features.

Firstly, we subtract the mean value from each pixel in the synthesized impact sound as follows:

$$I_I(t_i, f_i)^* = I_I(t_i, f_i) - < I_I(\mathbf{o}) > \tag{7}$$

where $< I_I(\mathbf{o}) >$ represents the mean value of $I_I(\mathbf{o})$ and $I_I(t_i, f_i)$ represents the color value of pixel i in $I_I(\mathbf{o})$. Then, for each pixel in $I_I(\mathbf{o})$, we scale the data points comprising the synthetic image by factors determined by the respective standard deviations:

$$I_{RI}(t_i, f_i) = \frac{\sigma_S}{\sigma_I} I_I(t_i, f_i)^* + < I_S(\mathbf{o}) > \tag{8}$$

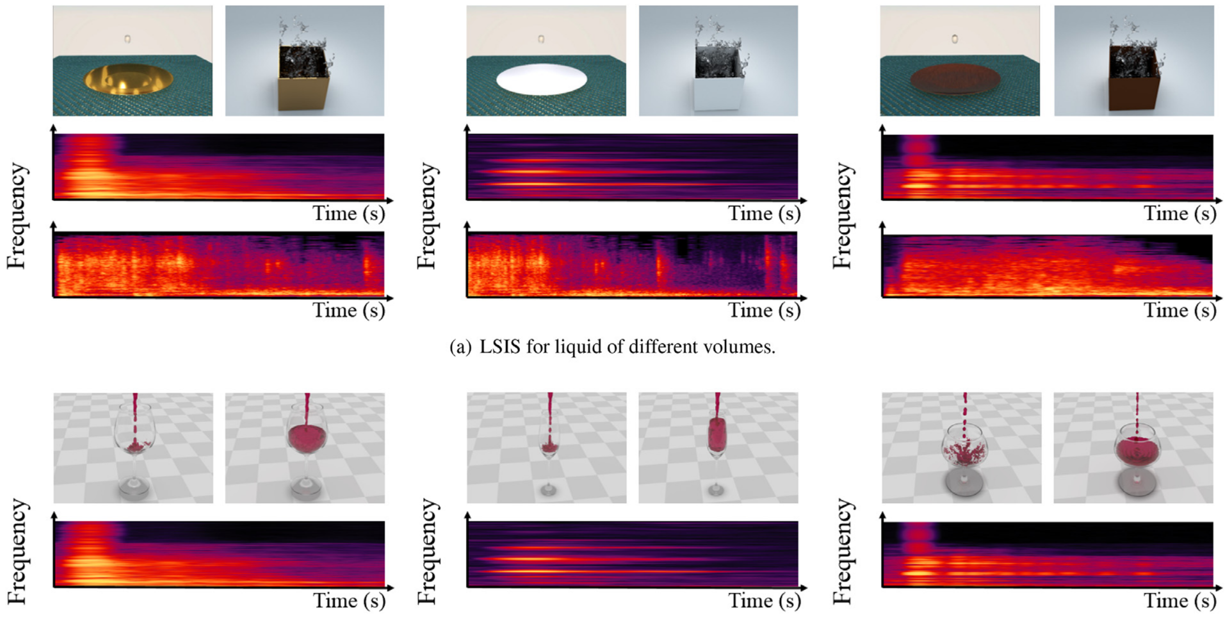
where $I_{RI}(t_i, f_i)$ is the pixel after enrichment in the impact sound spectrogram. $< I_S(\mathbf{o}) >$ represents the mean value of $I_S(\mathbf{o})$, σ_I represents the standard deviation of $I_I(\mathbf{o})$ and the calculation formula is shown as follows:

$$\sigma_I = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(I_I(t_i, f_i) - < I_I(\mathbf{o}) >)^2} \tag{9}$$

where N is the number of pixels in the extracted spectrogram $I_I(\mathbf{o})$ and $I_I(t_i, f_i)$ represents the ith pixel. The standard deviation of feature seed σ_S is calculated in the same way.

In order to avoid the influence of relatively sparse distribution in the spectrogram, we design a local constraint to ensure the intensity of the color transfer. Firstly, we get the maximum value in the $I_S(\mathbf{o})$ and $I_I(\mathbf{o})$, respectively. Then we store the maximum value corresponding

[1] In the experiment, each feature seed is 3s which can provide enough features and ensure a real-time computing efficiency.



(a) LSIS for liquid of different volumes.

(b) LSIS for containers of different sizes.

Fig. 7. Sound results for different scenes. (a) shows the animation frames (top) and spectra (spectra in the middle row and the bottom row correspond to water droplets and water balls, respectively) of sound results for liquid of different volumes. (b) illustrates the animation frames (top) and spectra (bottom) of sound results for different shapes of wine glasses.

pixel and its eight neighbor pixels as a local constraint block. The mean and standard deviation for local blocks are calculated in the same way as above. After we get the local standard deviations of $I_S(\mathbf{o})$ and $I_I(\mathbf{o})$, Eq. (8) becomes the following form after adding local constraints:

$$I_{RI}(t_i, f_i) = \left(w_1 \frac{\sigma_S}{\sigma_I} + w_2 \frac{\sigma_S^l}{\sigma_I^l} \right) \cdot I_I(t_i, f_i)^* + < I_S(\mathbf{o}) > \qquad (10)$$

where σ_I^l and σ_S^l are the local standard deviation of $I_S(\mathbf{o})$ and $I_I(\mathbf{o})$. We can get a preliminary enriched spectrogram $I_{RI}(\mathbf{o})$ after color transfer with local constraint. A final notable detail is that w_1 and w_2 are the adjustable weight values of global and local standard deviation values. In practice, we set different values to test and finally choose $w_1 = w_2 = 0.5$ in our experiment.

5.2. Coordinate constraint

Different from an ordinary image, in a spectrogram, the position of the color represents the distribution of the frequency which should also be considered in the transfer process. This is not involved in traditional color transfer methods. Thus, we need to add coordinate constraint in the FTS to ensure the result spectrogram not only keeps the same color style but also keeps the same color distribution. The spectrogram after transferring should be roughly the same in color distribution and the frequency features of the synthesized impact sound should be retained.

In the spectrogram $I_I(\mathbf{o})$, $\mathbf{o} = (t, f)$, the ordinate f is corresponding to the frequency value. In order to distinguish between the frequency symbol in the sound formula, we choose to use y_f to represent the ordinates in the spectrogram. We use the Euclidean distance from the ordinate of the frequency as constrains and design the following coordinate constraint:

$$I_{RI}(t_i, f_i) = \Theta \cdot \sqrt{\frac{\eta}{(|y_{fi-I} - y_{fmax-S}| + 1)}} \cdot I_t(t_i, f_i)^* + < I_S(\mathbf{o}) > \qquad (11)$$

where $\Theta = w_1 \frac{\sigma_t}{\sigma_s} + w_2 \frac{\sigma_t^l}{\sigma_s^l}$. y_{fi-I} is the ordinate of pixel i in $I_I(\mathbf{o})$ and y_{fmax-S} represents the ordinate of pixel with max color value in $I_S(\mathbf{o})$. In order to avoid the denominator for the 0 case, we add 1 after the distance. The FTS factor η is the variable that controls the intensity. We set $\eta = 10$ in the experiment and test the effect on sound results when η is in different values in Section 7.3.

Through Eq. (11), we can add the high frequency details missing form synthesized sound. The closer to the high frequency information ordinate y_{fmax-S}, the higher the frequency will be. We can thereby get the enriched impact sound whose color distribution is similar to the recording. Because high frequency information is not necessarily concentrated in one area, we each time mark the ordinate y_{fmax-S} used to prevent repeated reading. We repeat the above steps several times (generally 5–6 times in our experiments) to get the final enriched impact sound.

We summarize the FTS in Algorithm 1. A simple example is shown in Fig. 4, and the bottom row (FTS result) of Fig. 4(a) contains both S_{Impact}, $S_{ImpactSeed}$ frequency distribution. For a clearer display of the results of FTS, we choose two points (B and C) for frequency analysis plots as shown in Fig. 4(b). The $S_{ImpactRich}$ result (blue line) matches well with the $S_{ImpactSeed}$ (orange line).

6. Blending and propagation

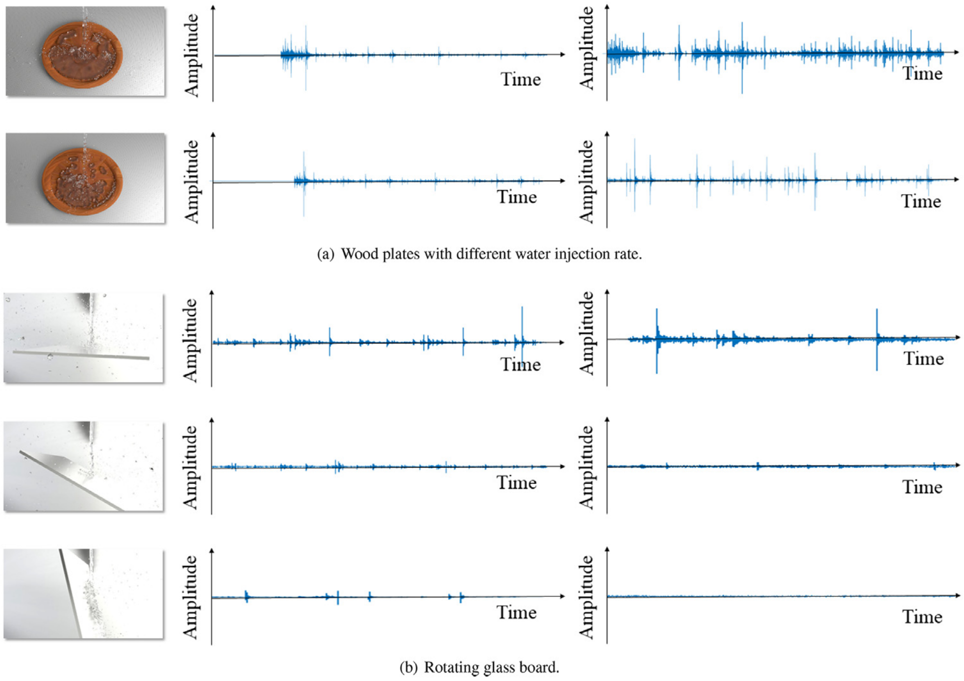
After obtaining the enriched bubble sound and impact sound with FTS, we need to blend the sounds and get the final LSIS. In this step, there are two following problems should be resolved:

(1) The amplitude range of the impact sound should be scaled due to the vibrational modes of a solid (such as continuous water pouring) will be affected by how much water is contained by the solid. Thus, a strategy needs to be designed to synchronize the impact sound.

(2) The sound radiation with near-field scattering should be sought for further improvement of the LSIS.

Synchronization rectification.

To resolve the first problem, we propose a key frame blending algorithm with normal force and grid surface to synchronize the two parts of the sounds. Firstly, we calculate the normal force which excites the solid modes on the collision surface as shown in Fig. 5. As shown in Fig. 5 (left), for the container with a flat bottom, the normal force is easy to calculate. For the container with a non-flat bottom (right part in



(a) Wood plates with different water injection rate.

(b) Rotating glass board.

Fig. 8. Comparison of FTS results with real-word recordings. Given one recording sample (WO and LI for (a), GL and LI for (b)), the features of the recording can be applied to different scenes of the same material (middle), and the results are comparable to the real-world recordings (right).

Algorithm 1 Feature Transfer Synthesis (FTS).

```
Input:
    The synthesized impact sound $S_{Impact}$;
    The feature seed of impact sound $S_{ImpactSeed}$ ;
Output:
    The enriched impact sound $S_{RichImpact}$;
// Transform the sound signals to spectra.
$S_{Impact} \xrightarrow{STFT} I_I(\mathbf{o})$;
$S_{ImpactSeed} \xrightarrow{STFT} I_S(\mathbf{o})$;
// Local constraint parameters calculation (take $S_{Impact}$ as an example).
$[column, line] \leftarrow size(I_I(\mathbf{o}))$
$N \leftarrow column \times line$
initialize $MaxPixel \leftarrow 0$
for $i = 1$ to $N$ do
    Calculate $\sigma_I$ based on Eq. (9);
    Update $MaxPixel$;
    Calculate $\sigma_I^l$ based on Eq. (9);
end for
// Coordinate constraint parameters calculation.
for $i = 1$ to $N$ do
    Update ordinate of the $MaxPixel$;
    Update $I_{RI}(t_i, f_i)$ based on Eq. (11);
end for
// Transform the spectrogram to sound signal.
$I_{RI}(\mathbf{o}) \xrightarrow{ISTFT} S_{RichImpact}$;
```

Fig. 5), we first calculate the angle α between the tangent plane and the horizontal plane of the contact surface. According to the properties of triangles, we can find that the angle of calculating the normal force is equal to the angle between the contact surface and the horizontal surface. Moreover, we assume in our experiments that liquids move at a uniform speed, so the normal force is a component of gravity. Therefore, the normal force $F_{Nor} = G_{liquid} \cdot \cos\alpha$ and when the bottom of the container is horizontal, the normal force is equal to gravity. For continuous inflow scenarios, we choose unit time water injection to calculate liquid gravity.

Then, we consider another key factor, the liquid contained by the solid container. Based on the fact that a large amount of liquid on the collision surface will suppress the solid vibration, we approximate this factor by calculating the volume of liquid per unit collision area. When the liquid surface collides with the solid, there will be a contact area. We calculate the number of meshes N_m that located in the contact area as the scale factor of the interaction sound. For the meshes which located at the boundary of the contact area, the area of these mesh boundary grids are added as the total area $\sum_{i=1}^{N_m} s_{grid\_area_i}$ with the area of a single mesh $s_{grid\_area_i}$. We define the collision area is the area in the direction of motion which is calculated as follows:

$$S_{Coll} = \sum_{i=1}^{N_m} s_{grid\_area_i} \times |\vec{n_s} \cdot \vec{v_u}| \tag{12}$$

where $\vec{n_s}$ is the unit normal vector of a single mesh $s_{grid\_area_i}$ and $\vec{v_u}$ is the unit vector of velocity. Thus, the proportion of sound through changes in normal force F_{Nor} and collision area as follows:

$$W = \frac{F_{Nor}}{V_{grid}/S_{Coll}} = \frac{F_{Nor} \cdot S_{Coll}}{V_{grid}} \tag{13}$$

After we calculate W, then we times the enriched impact sound to blend with enriched bubble sound. The above procedure is calculated as follows:

$$LSIS(\mathbf{r}, t) = W \cdot S_{RichImpact}(\mathbf{r}, t) + S_{RichBubble}(\mathbf{r}, t) \tag{14}$$

where $S_{RichImpact}$ and $S_{RichBubble}$are the impact sound and the bubble sound after enrichment, respectively. After scaling, the bubble sound and the impact sound can be merged to produce the resulting sound $LSIS(t)$ for liquid-solid interaction animation.

Sound radiation.

We approximate the sound field radiating from a dipole as a linear superposition of contributions due to the radiated sound field of point source near the water surface is characterized by dipole acoustic field. Following the theory of linear acoustics, we seek to approximate the acoustic pressure field with the derivation in [27] as follows:

$$P(\mathbf{r}, t) = -\frac{k^2 \rho_0 c_0 D_s(t) \cos\theta}{4\pi r} e^{-jkr}, \mathbf{r} \in \mathbb{R}^3 \tag{15}$$

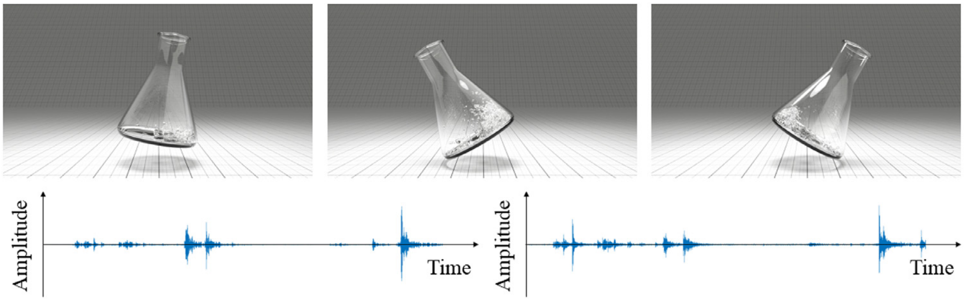


Fig. 9. A conical flask scenario. Sound results with different recordings. We used GLP and GLS as input for sound synthesis and the waveform of corresponding results are illustrated in the bottom row (GLP for the left and GLS for the right) separately.

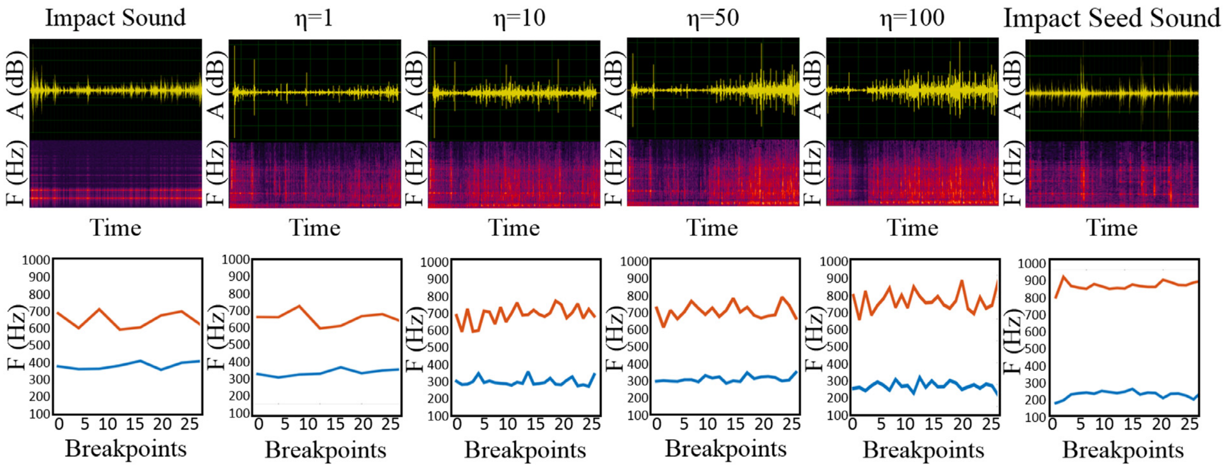


Fig. 10. Spectra of impact sound, impact seed sound and FTS results with different η values. The bottom row is spectral centroid (orange line) and spectral flatness (blue line) varying according to the FTS factor η. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Material recognition rate matrix: recorded sounds.

	Recognized material				
Recording LSIS	Wood (%)	Plastic (%)	Metal (%)	Porcelain (%)	Glass (%)
Wood	**70.1**	29.9	0.0	0.0	0.0
Plastic	22.7	**63.4**	3.9	5.8	4.2
Metal	3.7	1.8	**88.3**	1.9	1.2
Porcelain	0.0	0.0	1.2	**51.2**	47.6
Glass	0.0	0.0	0.0	42.7	**57.3**

where $r = \|\mathbf{r}\|$, $k = \omega/c_0$ is wavenumber. We define the intensity of the sound field as $D_s(t) = Q_s(t)d$, $Q_s(t)$ is the volume flow of a single sound source. The radiation map is shown in Fig. 6. The value of some aforementioned parameters are given in Table 1.

7. Results and discussion

7.1. Implementation

Based on the above algorithms, we synthesized the sound for collision events between liquid and solid surface in different scenes. All experiments were run on a computer configured as follows: Core i5-4460 3.20GHz CPU, NVIDIA GeForce GTX745 GPU, 8GB RAM. All the liquid-solid interaction models were constructed based on an improved FLIP method [9]. We used the particles for bubble sound calculation and grid-particle coupling for impact sound calculation. The resolution of our animation is 1280 by 720, and the duration of the animation is less than 10 s. The sample rate of sound is 44.1 kHz and the window size is 1024 in our experiment. The runtime for different examples is given in Table 2. The number in the bracket represents the self-time of the function and the number outside represents the total-time of the function.

Input recordings In our experiment, there were seven recordings collected in total as shown in Table 3. We recorded sound using a shotgun microphone and applied a denoising process [11] for each recording. Each recording was about 30 s through starting over to get the initial impact many times with water pouring motion. We recorded the sound of pouring water onto different solid surfaces for five recordings (ME, WO, PL, GLP and PO), recorded the sound of pouring water onto a tank of water for recording LI and recorded the sound of shaking the tank of water for recording GLS. In the following experiments, we marked out the recordings we used with different identifiers illustrated in Table 3.

7.2. Liquid-solid interaction examples

LSIS for liquid of different volumes. The top row of Fig. 7 (a) shows the animation frames of water droplets and water balls falling on surfaces of different materials. The animated images from left to right are the collision between metal surface and water, the collision between porcelain surface with the water as well as the collision between wood surface and water. The middle row of Fig. 7(a) shows the spectra of a

Table 5
Material recognition rate matrix: synthesized sounds using our method.

	Recognized material				
Synthesized LSIS	Wood (%)	Plastic (%)	Metal (%)	Porcelain (%)	Glass (%)
Wood	**67.3**	29.7	1.4	1.6	0.0
Plastic	27.7	**65.7**	3.4	2.0	1.2
Metal	4.2	2.3	**86.6**	3.7	3.2
Porcelain	0.0	0.0	2.2	**57.5**	40.3
Glass	0.0	0.0	1.6	47.8	**50.6**

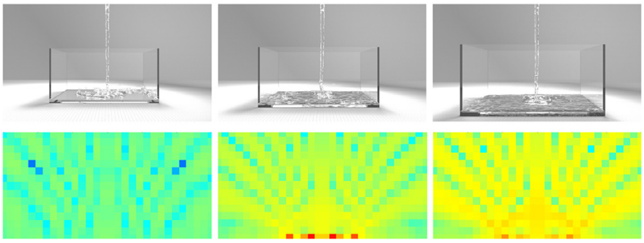


Fig. 11. Sound radiation maps of listener in different positions. From the left to the right, the listener keeps close to the tank.

water droplet falling onto a plate made of different materials. The bottom row of Fig. 7(a) shows the spectra of a water ball falling into a tank made of different materials. Since the volume of a water droplet and the volume of a water ball are different, the content of bubble sound is different. However, for the objects with same material, such as metal plate and metal tank, they should sound like coming from the same material. We can observe the similarity in the spectra although the volumes of liquid are different. Since we utilized same input recordings for the both water droplet and water ball scenes, the results proves that we can synthesize LSIS for liquid of different volumes.

LSIS for containers of different sizes. The top row of Fig. 7(b) shows the animation frames of pouring red wine into different sizes of wine glasses. The bottom row of Fig. 7(b) shows the spectra of sound for different scenes. In these three different scenarios, the water injection rate and the direction of the liquid inflow are the same, so only the shape of the glass leads to the different sound. When the wine glass is thin (the middle part in Fig. 7(b)), according to the Eq. (13), the S_{Coll} is smaller than the other two glasses, thus, the impact sound decays faster than the other two scenes. The sound results prove the reasonableness of our blending algorithm.

7.3. Validation

In this section, we designed four experiments to evaluate the generality and performance of our simulate pipeline. We first compared our sound result with recordings and utilized different recordings for a same scenario. Then we validated the parameter selection for FTS and the effect of sound radiation. The differences of sound results can be heard in the supplementary video.

Comparison with real recordings.
Fig. 8 shows comparisons of the sound results for different scenes. As shown in Fig. 8(a), from a recording of wood plate (WO), the features for wood are estimated and transferred to other virtual wood plates with different volume of water. In Fig. 8(b), the glass board is rotated at different angles which affects the value of normal force. It can be seen that although utilize a same feature seed, we can synthesize LSIS that matches the animation. Moreover, the perception of material is preserved, as can be verified in the accompanying video.

Then, we further tested different recordings in Fig. 9. We can observe that the changes of waveforms on the left and right sides are identical.

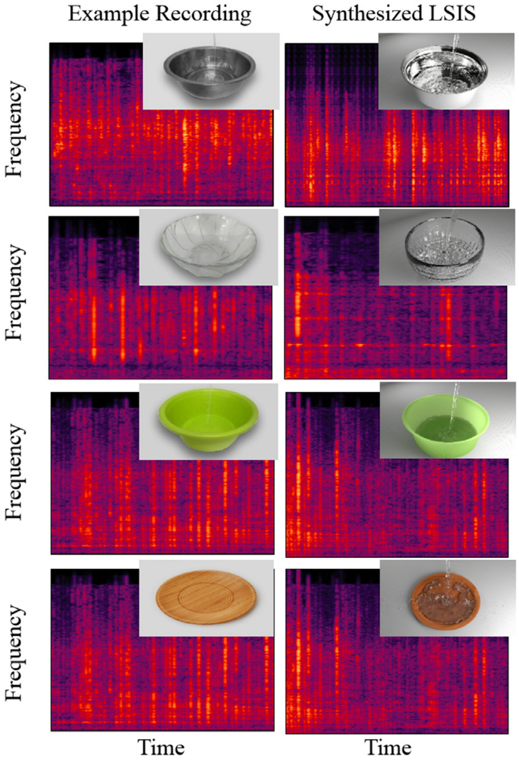


Fig. 12. Spectra comparison between recording and synthesized LSIS for different solid materials.

Therefore, different recordings have an impact on timbre, but they do not affect the synchronization of the results.

Verification of the selection of FTS factor η. In our FTS scheme, we are able to set different FTS factor η for each experiment independently. However, the choice of the FTS factor η used in Eq. (11) forces the resulting spectrum to be as close as possible to the $S_{ImpactSeed}$ and S_{Impact} by keeping both frequency contents. Thus, we get the ideal value through experimental tests. Fig. 10 shows an example of the result LSIS according to varying FTS factor η of spectral centroid and spectral flatness while preserving a constant value for the rest. From the spectra we can see that when $\eta = 10$, the synthesized LSIS can be better combined with the two part of the sound characteristics. According to the experiment, we finally choose $\eta = 10$ for the best performance.

Sound radiation. We show in the video the modification of sound depending on the position of the listener (top row of Fig. 11). The bottom row of Fig. 11 corresponds to glass tank with pouring water while changing its distance with respect to the listener. The time-domain sound radiation of the LSIS causes the sound shift that we can hear in the video.

7.4. Perceptual user study

In order to further evaluate the effectiveness of our approach, we design three experiments. 50 volunteers (30 males and 20 females) are

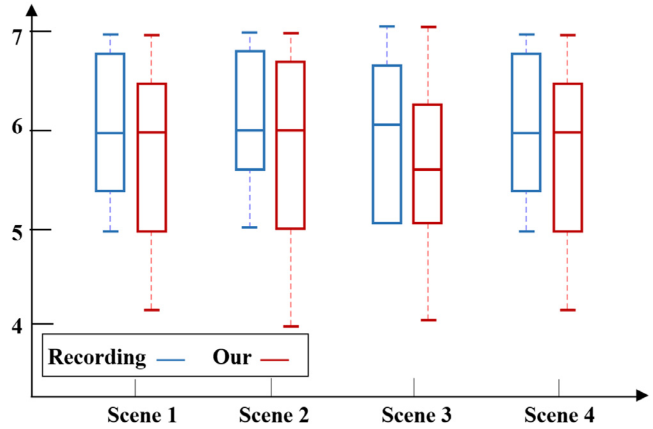


Fig. 13. User survey results of the first experiment. Boxplots show the interquartile range, the minimum and the maximum values of the score for each pair of sound clips.

invited in the survey. Among them, 12 have specialized computer graphics knowledge and all of them have normal hearing. The details of four experiments are illustrated as follows.

Realistic tests. The first experiment is designed to evaluate the similarity of synthesized LSIS and recordings. In the first experiment, we present three pairs of video clips. Each page contains the audio from one of our demo scenarios as shown in Fig. 12 and the recording for same scene. In each scenario, the participants are asked the realistic score of the sound they heard. Participants answer using a Likert scale (1:Not at all, 7:Definitely yes). We can observe that there is no statistically significant difference of the spectra between recorded and simulated sounds in Fig. 12 which also can be verified in the accompanying video. User survey results are analyzed using standard tests and reported in Fig. 13 which also show that the synthesized LSIS and recordings sound similar.

Distinguishability tests. The third experiment is to verify the cumulative recognition rates of the sounding materials in two separate matrices: Table 4 presents the recognition rates of sounds from real-world materials, and Table 5 reflects the recognition rates of sounds from synthesized LSIS. The numbers are normalized with the number of subjects answering the questions. We find that the successful recognition rate of virtual materials using our synthesized sounds compares favorably to the recognition rate of real materials using recorded sounds. We also can find in the tables, for both recorded and synthesized sounds, several subjects have reported difficulty in reliably differentiating between wooden and dull plastic materials and between glass and porcelain.

The third experiment is to verify the match between video clips and audio clips, we design three test scenarios, including water droplets, water ball and faucet experiments as described above. Each scene consists of video clips and LSIS. For the same scene, we mix the video clips produced by different materials and LSIS, volunteers need to match video clips and sound clips. Each material video clip has three levels, from 1 to 3. 1 means the lowest score, i.e., the lowest matching degree between video clips and audio clips. During the test, volunteers do not know the name of the sound clip, they could only hear the sound. Fig. 14 shows the matching results under three test scenarios, respectively. In each figure, the ordinate represents the total score and the red dotted lines mean the average score. Each video clip contains three columns representing the sound fragments produced by three different materials. We can easily find that video clips have the highest matching degree with the same name of the audio clips. For users, different materials of water droplets are the most easy to judge, which means that volunteers can correctly identify the resulting sounds and animations of the water through our methods.

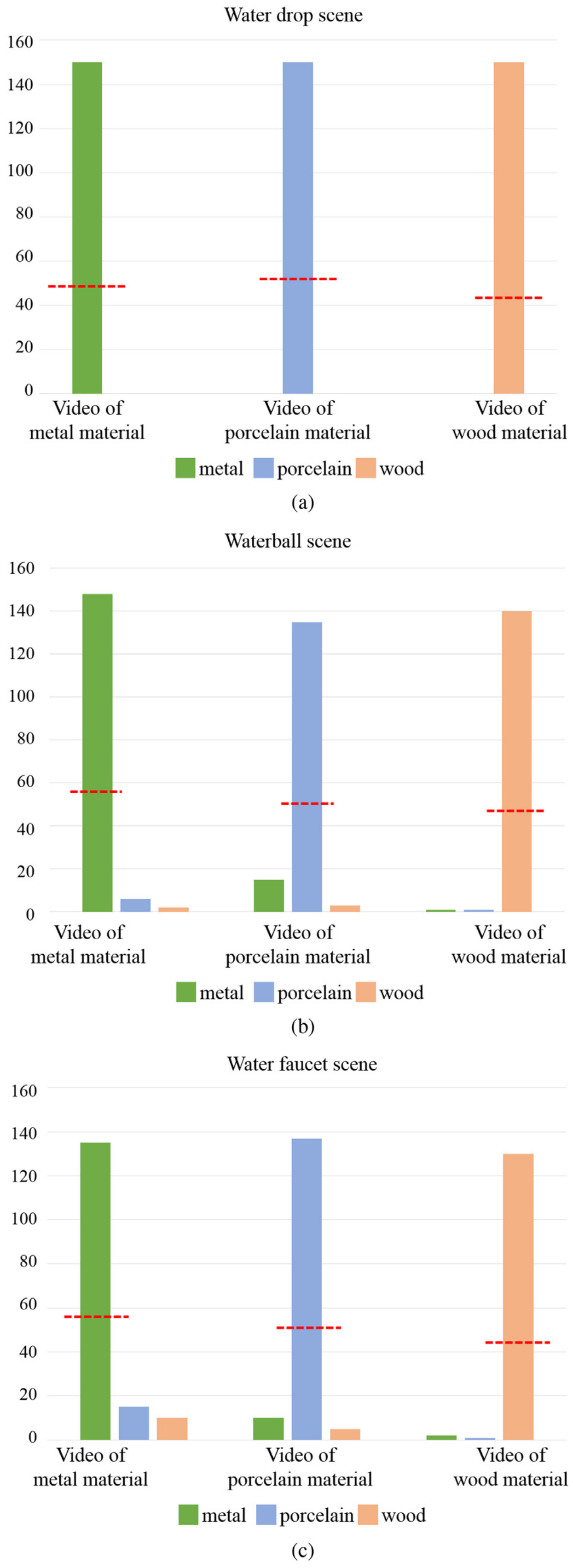


Fig. 14. The matching results of video and audio of different materials in the faucet scene.

8. Conclusion and future work

In this paper, we proposed a novel LSIS synthesis framework accounting for different solid interfaces. To the best of our knowledge, this was the first attempt to synthesize liquid sound considering the influence of liquid-solid interaction. We separated the LSIS into two components: bubble sound and impact sound. Bubble sound and impact sound were based the physical liquid-solid model in order to make the sound more consistent with the input animations. To compensate for the differences between the real-world recording and synthesized sound, we designed a new sound enrichment method called FTS to enrich the synthesized sound. We further estimated the LSIS pressure at a listener position through a generalized dipole model. By synchronizing the two sounds with a proposed grid-volume algorithm, the final high quality LSIS for various scenarios could be synthesized.

Of course, our method still has some room for improvement. For example, the feature extraction of the sounds may be limited by the recordings. If there is no obvious sound features in a liquid-solid interaction scenario, our method may fail. Moreover, sometimes the background recording may be transferred into the synthesized sound, leading to some noise in the sound results. Moreover, the synchronization rectification model we design is relatively simple, which typically only considers the gravity. It is therefore interesting to explore the momentum of the liquid and other factors for further improvement. In the future, we will take into account human perception to strengthen realistic of the LSIS. On the other hand, we will also exploit the ability of GPUs (Graphics Processing Units) to further accelerate our method.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments. This work was supported by the Natural Science Foundation of China under grant nos. 61672375 and 61170118.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.gmod.2019.101028

References

[1] J.N. Chadwick, S.S. An, D.L. James, Harmonic shells:a practical nonlinear sound model for near-rigid thin shells, ACM Trans. Graph. 28 (5) (2009) 1–10.

[2] J.N. Chadwick, C. Zheng, D.L. James, Faster acceleration noise for multibody animations using precomputed soundbanks, in: ACM Eurographics Symposium on Computer Animation, 2012, pp. 265–273.

[3] J.N. Chadwick, C. Zheng, D.L. James, Precomputed acceleration noise for improved rigid-body sound, ACM Trans. Graph. 31 (4) (2012) 103–111.

[4] H. Cheng, S. Liu, Haptic force guided sound synthesis in multisensory virtual reality (VR) simulation for rigid-fluid interaction, in: IEEE VR, 2019, pp. 1–9.

[5] K.D.V. Den, P.G. Kry, D.K. Pai, Foleyautomatic: physically-based sound effects for interactive simulation and animation, in: ACM SIGGRAPG, 2001, pp. 537–544.

[6] K.D.V. Den, D.K. Pai, Synthesis of shape dependent sounds with physical modeling, in: International Conference on Auditory Display, 1996, pp. 1–7.

[7] K.D.V. Den, D.K. Pai, The sounds of physical shapes, Teleoperators Virtual Environ. 7 (4) (1998) 382–395.

[8] K.V.D. Doel, Physically-based models for liquid sounds., ACM Trans. Appl. Percept. 2 (4) (2005) 534–546.

[9] F. Florian, A. Ryoichi, W. Chris, W. Rüdiger, T. Nils, Narrow band FLIP for liquid simulations, Comput. Graph. Forum 35 (2) (2016) 225–232.

[10] Y. Gao, S. Li, L. Yang, H. Qin, A. Hao, An efficient heat-based model for solid-liquid–gas phase transition and dynamic interaction, Graph. Models 94 (6) (2017) 14–24.

[11] Y. Hu, P.C. Loizou, Speech enhancement based on wavelet thresholding the multitaper spectrum, IEEE Trans. Speech Audio Process. 12 (1) (2004) 59–67.

[12] M. Imura, Y. Nakano, Y. Yasumuro, Y. Manabe, K. Chihara, Real-time generation of CG and sound of liquid with bubble, in: ACM SIGGRAPH, 2007, p. 97.

[13] T.R. Langlois, C. Zheng, D.L. James, Toward animating water with complex acoustic bubbles, ACM Trans. Graph. 35 (4) (2016) 95–107.

[14] T.G. Leighton, R.E. Apfel, The Acoustic Bubble, 1994.

[15] S. Liu, Z. Wang, Z. Gong, Q. Peng, Simulation of atmospheric binary mixtures based on two-fluid model, Graph. Models 70 (6) (2008) 117–124.

[16] S. Liu, Z. Yu, Sounding fire for immersive virtual reality, Virtual Real. 19 (3–4) (2015) 291–302.

[17] M.S. Longuet-Higgins, Monopole emission of sound by asymmetric bubble oscillations., J. Fluid Mech. 201 (1989) 525–541.

[18] M. Minnaert, On musical air bubbles and the sound of running water, Philos. Mag. 16 (104) (1933) 235–248.

[19] W. Moss, H. Yeh, J.M. Hong, M.C. Lin, D. Manocha, Sounding liquids: automatic sound synthesis from fluid simulation, ACM Trans. Graph. 29 (3) (2010) 21–34.

[20] L. Ms, Monopole emission of sound by asymmetric bubble oscillations., J. Fluid Mech. 201 (1989) 543–565.

[21] J.F. O'Brien, Synthesizing sounds from physically based motion, in: ACM SIGGRAPH, 2001, pp. 529–536.

[22] J.F. O'Brien, C. Shen, C.M. Gatchalian, Synthesizing sounds from rigid-body simulations, in: ACM Eurographics Symposium on Computer Animation, 2002, pp. 175–181.

[23] E. Reinhard, M. Adhikhmin, B. Gooch, P. Shirley, Color transfer between images, IEEE Comput. Graph. Appl. 21 (5) (2001) 34–41.

[24] Z. Ren, H. Yeh, M.C. Lin, Example-guided physically based modal sound synthesis, ACM Trans. Graph. 32 (1) (2013) 1–16.

[25] A. Shabana, Vibration of discrete and continuous systems, Springer Science & Business Media, 2012.

[26] N. Thürey, R. Keiser, M. Pauly, U. Rüde, Detail-preserving fluid control, Graph. Models 71 (6) (2009) 221–228.

[27] E.G. Williams, Fourier Acoustics, 1999.

[28] Q. Yin, S. Liu, Sounding solid combustibles: non-premixed flame sound synthesis for different solid combustibles, IEEE Trans. Vis. Comput. Graph. 24 (2) (2018) 1179–1189.

[29] C. Zheng, D.L. James, Harmonic fluids, ACM Trans. Graph. 28 (3) (2009) 37–48.

[30] C. Zheng, D.L. James, Toward high-quality modal contact sound, ACM Trans. Graph. 30 (4) (2011) 38–48.

[31] W. Zheng, J.H. Yong, J.C. Paul, Simulation of bubbles, Graph. Models 71 (6) (2009) 229–239.