Physically-based Statistical Simulation of Rain Sound

SHIGUANG LIU, Tianjin University, China
HAONAN CHENG, Tianjin University, China
YIYING TONG, Michigan State University, USA

A typical rainfall scenario contains tens of thousands of dynamic sound sources. A characteristic of the large-scale scene is the strong randomness in raindrop distribution, which makes it notoriously expensive to synthesize such sounds with purely physical methods. Moreover, the raindrops hitting different surfaces (liquid or various solids) can emit distinct sounds, for which prior methods with unified impact sound models are ill-suited.

In this paper, we present a physically-based statistical simulation method to synthesize realistic rain sound, which respects surface materials. We first model the raindrop sound with two mechanisms, namely the initial impact and the subsequent pulsation of entrained bubbles. Then we generate material sound textures (MSTs) based on a specially designed signal decomposition and reconstruction model. This allows us to distinguish liquid surface with bubble sound and different solid surfaces with MSTs. Furthermore, we build a basic rain sound (BR-sound) bank with the proposed raindrop sound clustering method based on a statistical model, and design a sound source activator for simulating spatial propagation in an efficient manner. This novel method drastically decreases the computational cost while producing convincing sound results. Various experiments demonstrate the effectiveness of our sound simulation model.

CCS Concepts: • **Computing methodologies** → *Computer graphics*; • **Applied computing** → *Sound and music computing*.

Additional Key Words and Phrases: sound synthesis, basic rain sound, physical and statistical, rain animation, procedural audio

ACM Reference Format:
Shiguang Liu, Haonan Cheng, and Yiying Tong. 2019. Physically-based Statistical Simulation of Rain Sound. *ACM Trans. Graph.* 38, 4, Article 123 (July 2019), 14 pages. https://doi.org/10.1145/3306346.3323045

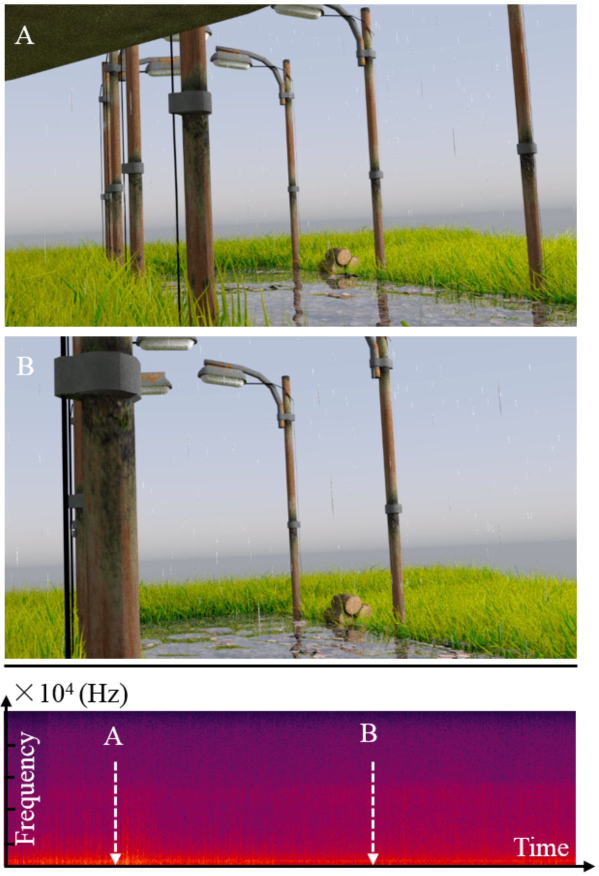


Fig. 1. **Audiovisual simulation of walking along a country road in the rain.** Frames A and B illustrate two positions of the listener. From A to B, as the listener gets closer to the street lights, the metal sound effects become more and more apparent. Our method is able to capture this dynamic change of rain sound as shown in the spectrogram on the bottom row.

1 INTRODUCTION

Rain sound, as a common environmental sound, is widely used in virtual scenes such as animated movies and computer games. It can greatly enhance one's sense of immersion in virtual scenes. There are two obvious characteristics of a rainfall scenario: large scale with a large number of raindrops, and the heterogeneity of interaction surfaces in the scene, which make synthesis of accurate rain sound rather computationally demanding. In this paper, we propose a method that can efficiently resolve these challenges and synthesize realistic surface material-aware rain sound. We show an example produced by our sound synthesis pipeline in Figure 1. (See more in § 6.)

Recent advances in the sound simulation have enabled the synthesis of rain sound and these methods can be classified into two categories, non-physically-based methods and physically-based methods. The non-physically based methods usually employ signal processing to adjust the recorded sound

Authors' addresses: Shiguang Liu, Tianjin University, China, lsg@tju.edu.cn; Haonan Cheng, Tianjin University, China, tjuchn@tju.edu.cn; Yiying Tong, Michigan State University, USA, ytong@msu.edu.

0730-0301/2019/7-ART123 $15.00
https://doi.org/10.1145/3306346.3323045

for authenticity. However, for the methods in this category, as discussed in [Verron and Drettakis 2012; Zita 2003], there are two major limitations: *i*) the indirect correspondence with the parameters of the physical animation model, and *ii*) the need for artificial adjustment. Both limitations can be overcome by physically-based methods with direct matching of physical parameters. Thus, an increasing amount of research effort [Chadwick and James 2011; Langlois et al. 2016; Liu and Yu 2015; Moss et al. 2010; Yin and Liu 2018; Zheng and James 2009] has been devoted to this field. However, due to the vastly different formation mechanisms, the generic physically-based sound synthesis methods mentioned above are ill-suited for the synthesis of rain sound. Targeting rain sound, Zita [2003] designed a physical model by analyzing the relationship between raindrop kinetic energy and velocity. While the rain sound model in [Zita 2003] is able to automatically synchronize with rainfall animation, it is a unified physically-based impact sound model which is not computationally feasible for large virtual environments. Moreover, it does not distinguish surfaces of different materials. In contrast, we propose a physically-based statistical simulation method to synthesize rain sound. In comparison to previous approaches, the advantages of our method include that the proposed rain sound model ensures distinguishable sound for different surface materials through different mechanisms, and our simulation is highly efficient owing to the combination of physical sound synthesis theory and statistical mechanisms.

To achieve more natural rain sound, we augment the rain sound model in [Zita 2003] which only took impact effect of raindrops into consideration. The studies in [Pumphrey et al. 1989] suggest that the impact of a water drop can produce two distinct sounds: one from the initial impact and another from bubble oscillation. The follow-up work [Guo and Williams 1991; Prosperetti et al. 1989] also indicates that the bubble sound should not be ignored since bubbles are responsible for the 14kHz spectral peak of underwater sound by rain. Based on these theoretical observations, we have further differentiated the composition of rain sound and propose a novel sound model for raindrops with two separate mechanisms: first, the initial impact sound, occurs for every impact; second, the bubble oscillation, which, when it occurs, is a stronger acoustic source than the initial impact, but does not occur for every raindrop [Pumphrey et al. 1989] or in every scene. Furthermore, to distinguish different solid surface materials, we enrich the impact sound with features extracted from real rain sound examples based on a modified Variational Mode Decomposition and Spectral Variance (VMD-SV) method to reconstruct the material sound textures (MSTs). Consequently, the new physically-based acoustic equation contains both bubble sound and impact sound for raindrops, and achieves a higher degree of realism.

To improve the throughput of the computational bottleneck for a large number of dynamic sound sources in rainfall scenes, we take advantage of both physically-based sound synthesis theory and statistical mechanisms on the overall rain sound. Fortunately, there have already been several attempts [Guo and Williams 1991; Medwin et al. 1992; Pumphrey and Elmore 1990] to discover the relationships between rain sound level and attributes of rain, such as distribution of raindrops, and rainfall rate. Medwin *et al.* [1992] built the distribution of different size raindrops for rainfall at different scales. However, these methods just summarized some statistical models empirically. Nevertheless, they made it possible to establish the rain sound model through statistics, which can thereby avoid the expensive matching process in purely physically-based methods. Along the same line, we focus on rain sound synthesis through a physically-based statistical technique. The statistical model for generating sound clips is then used to alleviate the high computational complexity of the physical models. We refer to these sound clips as basic rain sounds (BR-sounds) in the rest of the paper. We call the collection of available BR-sounds and MSTs as the BR-sound bank. To further decrease the computation time, based on the similarity among sound sources in rainfall scenarios and the limited human auditory distance, we propose a modified bidirectional sound transport algorithm with a specially designed sound source activator instead of painstakingly calculating all the rain sound sources in a large virtual environment. The sound source activator is devised based on the number of raindrops and the position of the listener with a novel material shader to synchronize the MSTs. Such an activator drastically reduces the simulation cost.

The main contributions of our work can be summarized as follows:

- We propose a hybrid physical and statistical rain sound synthesis system, which can smoothly synchronize with 3D animations. It is thus possible to strike a balance between computation cost and quality using the short sound clips built into our BR-sound bank with clustered physical parameters and a statistical model.
- We propose a novel rain sound model based on two physical mechanisms, which makes it possible to synthesize sound for surfaces of different materials. The solid and liquid surfaces can be distinguished by the impact model and the bubble oscillation model, and the solid surfaces are further differentiated with a newly designed VMD-SV method.
- A visual-to-audio coupling scheme is designed to significantly decrease the simulation cost. The novel sound source activator in the coupling scheme clusters the number of raindrops and the position of the listener as variables for efficient spatial propagation.

2 RELATED WORK

Sound models have long been studied. With the advances in computer hardware, in recent years, ever more attention has been drawn to sound synthesis in virtual environments.

2.1 Rain Sound Anatomy

There were numerous experiments on the various factors influencing on the audio frequencies in rain sound, including the distribution of raindrops, rainfall rate and other attributes in rainfall scenes with different scales. Marshall and Palmer [1948] pointed out that the size of each raindrop is inversely proportional to the number of raindrops in rainfall with different scales through experimental observation. Later, various studies on rain sound were conducted; sound radiation from large raindrops [Jacobus 1991], a raindrop sound production analytical model [Longuet-Higgins 1990], underwater rain noise [Nystuen 1986], rainfall rate [Medwin et al. 1992], temperature [Nystuen 1991] and the effect of salinity [Scofield 1992] were all taken into consideration.

Although previous works only analyzed and investigated the audio frequencies of rain sound, they provided us with a realistic model on the distribution of raindrops. With the studies mentioned above, we had the basic understanding of rain sound frequencies. Furthermore, the characteristics of rain sound are studied experimentally in [Pumphrey et al. 1989], which argued that the bubbles entrained in liquid by the impact of raindrops should not be ignored. Therefore, the method we propose in this paper for generation of BR-sounds with both bubble sound and impact sound is consistent with the observations in these experimental works.

2.2 Sound Simulation

There have been works in the graphics community in both categories of sound simulation, e.g., [Cheng and Liu 2019; Farnell 2010; Misra et al. 2006; Strobl et al. 2006] on non-physically-based methods, and [Cook 1997; Langlois et al. 2016] on physically-based methods. Non-physically-based methods usually rely on recorded audio samples as supplemental input. Since the features of sound textures are extracted from real sound examples, this type of methods greatly depend on the given recordings. With advanced texture generator frameworks [Strobl et al. 2006], the inherent quality of the recorded sounding materials could be captured and the resulting sounds closely resemble the real-world recordings. For instance, Ren et al. [2013] proposed a way to extract perceptually salient features from audio examples, which could then be used to automatically determine material parameters. For rain sound synthesis, Verron and Drettakis [2012] proposed a signal-based method to reproduce environmental sounds that are synthesized from five physically-inspired “sound atoms”. Due to the direct extraction of features from recording signals, the algorithm is highly efficient. However, the geometry of the 3D scene and the listener location within the scene have a clear effect on the rain sound, which cannot be easily captured by granular synthesis [Roads 1988] combined with recordings, due to the lack of information corresponding to spatial factors. Thus, they are better suited for scenes with 1D listener movements.

As a sharp departure from non-physically based methods, Cook [1997] introduced the physically informed stochastic event modeling (PhISEM) algorithm for the synthesis of percussive sound. Later, Cook developed a system for automatic analysis and parametric synthesis of walking sound [Cook 2002]. For raindrop sound, Zita [2003] designed a physical model for raindrop sound by analyzing the relationship between the kinetic energy of a raindrop and its velocity. For the related physically-based liquid sound synthesis, a methodology was developed in [Doel 2005]. Recently, the synthesis of liquid sound has been improved progressively by considering bubble creation rates [Zheng and James 2009], bubble size distributions [Moss et al. 2010], and morphology [Langlois et al. 2016]. These physically-based methods can produce realistic sounds matching visual animations, but as the real physical process is complex, the sound rendering techniques are generally computationally intensive.

Most related to ours are the methods in [Verron and Drettakis 2012; Zita 2003] which can effectively synthesize rain sound. However, the method in [Zita 2003] is based on physical principles, which is a computationally inefficient representation in the sequential form. Moreover, the simulation model only takes impact effect of raindrops into consideration, which cannot distinguish surfaces of different materials. The method in [Verron and Drettakis 2012] is highly efficient, yet the signal-based sound model is less synchronized with the particle animation. Since rain sound has high repeatability and randomness, just using the traditional physically-based method may be redundant with unnecessary repeated calculations, when combining all components to produce rain sound synchronized with the animation. In contrast, our method aims at efficient rain sound synthesis with reasonable synchronization with animations, by leveraging the statistical rain sound analysis and previous physically-based methods developed based on computational fluid dynamics.

2.3 Sound Propagation

Previous work on sound propagation can be classified into two broad categories, namely wave-based methods [Mehra et al. 2013; Raghuvanshi and Snyder 2014; Thompson 2006; Zhang et al. 2018] and geometric acoustic techniques [Cao et al. 2016; Lentz et al. 2007; Schissler et al. 2014]. The wave-based method can accurately simulate all acoustic effects yet are limited to static scenes. Geometric acoustic techniques, which are based on ray theory, provide an efficient solution for dynamic scenes and multiple sources. Among them, Cao et al. [2016] proposed a bidirectional sound transport algorithm which can offer considerable speedup over prior sound propagation algorithms.

Rainfall scenarios are particularly large dynamic scenes with a large number of sound sources, for which geometric acoustic techniques provide an ideal solution. Inspired by the recent work [Cao et al. 2016], we also simulate the sound propagation by bidirectional path tracing. Different from [Cao et al. 2016], taking advantage of the similarity of sound sources in rainfall scenarios and the limited hearing range of human auditory, we design a dynamic sound source activator based on the number of raindrops and the position of the listener which is tailored to rainfall scenarios. This component enables us to further speed up performance of the simulation.

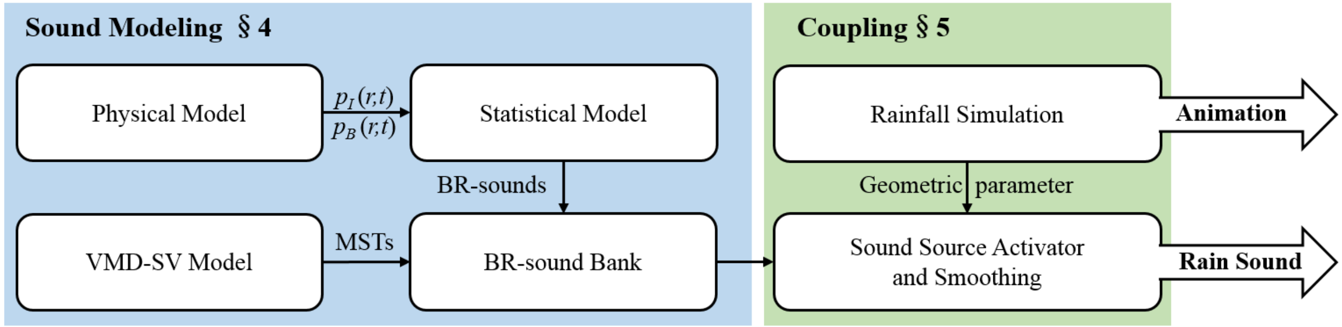


Fig. 2. System overview. Note that there are two main components in the framework, namely sound modelling and sound coupling with animation. The blue block is the subsystem to generate the raindrop sounds and the generation procedure of BR-sounds and MSTs. The green block represents the coupling process of rain sound and rainfall animations.

3 OVERVIEW

Figure 2 illustrates the whole framework of our method, which is composed of two main ingredients, namely rain sound modelling and sound synchronization.

In the sound modelling stage, we propose a new raindrop acoustic model by analyzing two separate mechanisms, initial impact and bubble oscillation. For simplifying the computation, after we build the raindrop acoustic function, we adapt a statistical distribution based on previous studies [Marshall and Palmer 1948; Medwin et al. 1992] to cluster and superimpose raindrop sounds in different parameter ranges. We refer to the synthesized rain sound clips through the superposition and clustering as BR-sounds and the collection of all the BR-sounds as the BR-sound bank. To distinguish surfaces of different materials, we generate MSTs with a designed VMD-SV method and store them in the BR-sound bank.

In the synchronization stage, first, a particle simulation system is used for the animation of rainfall. Then, the geometric and kinematic properties, such as velocity and number of raindrops, listener trajectory, and object location, are exported from the rainfall model. We utilize the above exported data as the input of a sound source activator to model the rain sound as area sources, thereby screening out a large amount of repeated calculation. Moreover, the MSTs are added into the blended sound through a material shader for enrichment. We further elegantly smooth the resulting sound as post-processing to obtain the final rain sound consistent with the rainfall animation.

4 MODELLING RAIN SOUND

We start from the sound of a single raindrop to synthesize the sound of the entire rainfall scene. In this section, we combine physical principles and a statistical model to generate the raindrop sound as the basis for BR-sound. To distinguish surfaces of different materials, we design a VMD-SV method to generate the MSTs. Finally, we build a BR-sound bank with two components: BR-sounds and MSTs.

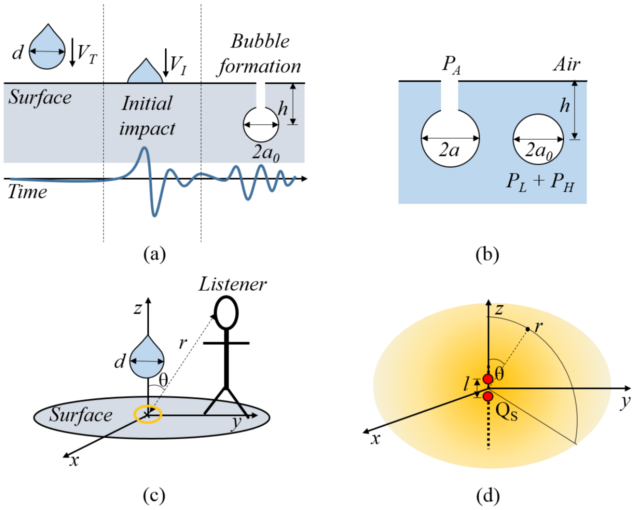


Fig. 3. Models for a raindrop. (a) is the process of a raindrop impacting on the surface and the corresponding acoustic wave patterns. (b) shows the variation of the entrained bubbles. (c) is a listener model and (d) is a close-up view of the orange ellipse in (c).

4.1 Physical Model for Raindrop Sound Generation

What causes raindrop sound? The experiments in [Franz 1959] show that the sound generation process for raindrops impacting a water surface can be divided into two stages, an initial impact and the subsequent bubble formation. Figure 3(a) illustrates the raindrop sound with two sources and their corresponding waveforms. The initial impact sound is a sharp pulse while the bubble sound is a decaying sinusoid. Inspired by previous work [Franz 1959; Guo and Williams 1991; Howe and Hagen 2011], we adjust and integrate the physical models of the two stages to make them suited to 3D rainfall simulators. For impacts on a solid surface, although the splash may also contain bubbles, the bubble sound is inaudible. Thus, the raindrop sound for a solid surface only contains an impact sound. We now introduce the two types of raindrop sound models in detail.

4.1.1 Initial Impact. The four main characteristics of the initial impact sound of a raindrop as shown by previous studies [Franz 1959; Guo and Williams 1991] are:

(1) present for every raindrop impact,
(2) highly transient (short time scale),
(3) with an initial sound pressure amplitude $\rho_0 c_0 V_I$, and
(4) with a radiation efficiency conforming to a dipole pattern.

These observations underpin our proposed impact raindrop sound model. Here, ρ_0 is the density of water, c_0 is the speed of sound in water and V_I is the normal impact velocity of a raindrop. As the impact sound of a raindrop is highly transient, it is difficult to obtain for both experimental observations and numerical simulations. However, with the initial amplitude and radiation efficiency, we can construct a modal acoustic model with a large damping constant to approximate the initial impact sound. Thus, the far-field initial impact raindrop sound pressure is formulated as

$$p_I(r,t) = \frac{k^2 \rho c D_S(t)}{4\pi r} cos\theta e^{-ikr} \tag{1}$$

where ρ is the air density, c is the speed of sound in air, r is the distance from a listener to the sound source, t is time, k is the wavenumber, θ is the polar angle (as shown in Figure 3(c)) and i is the imaginary unit. We define the dipole strength $D_S(t) = 0.05 Q_S(t)$ (0.05 represents the distance in the dipole sound field geometry and the setting is based on experiments.) and illustrate the dipole sound field geometry in Figure 3(d). $Q_S(t)$ is given by

$$Q_S(t) = A_I e^{-\beta_I t} sin(2\pi f_I t) \tag{2}$$

where the amplitude $A_I = \rho_0 c_0 V_I$, β_I is the damping constant and f_I is the frequency (random assignment between 1kHz and 16kHz). Since previous studies have shown that the impact sound frequency does not seem to follow a specific distribution, we sampled it with a uniform distribution. As described above, V_I is the raindrop impact velocity, for which there is no theoretical precise calculation. However, as noted in [Pumphrey and Elmore 1990], there is a connection between the impact velocity V_I and the terminal velocity V_T. Thereby, we approximate the impact velocity V_I as

$$V_I = V_T \sqrt{1 - exp(\frac{-2gz}{V_T^2})} \tag{3}$$

where z is the falling height of a raindrop. The terminal velocity V_T can be calculated from the raindrop diameter $d \in [0.1mm, 5.8mm]$ through a polynomial fit [Dingle and Lee 1972]:

$$V_T = \begin{cases} -17.8951 + 448.9498d + 16.3719d^2 - 45.9516d^3, \\ \qquad\qquad\qquad\qquad\qquad d \leq 1.4mm, \\ 24.1660 + 448.8336d - 75.6265d^2 + 4.2695d^3, \\ \qquad\qquad\qquad\qquad\qquad d > 1.4mm. \end{cases}$$

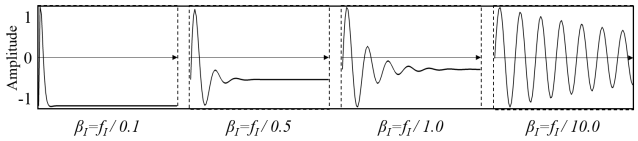


Fig. 4. Different damping constants and the corresponding waveforms. The waveform for $\beta_I = f_I/0.5$ resembles the typical short pulse of impact sound.

Finally, we choose $\beta_I = f_I/0.5$ as a reasonable underdamping (reducing the amplitude to $1/e^2$ in 2 periods) through experiments as illustrated in Figure 4.

4.1.2 Entrained Bubble. The oscillation of entrained bubbles results from the raindrop impact. The associated acoustic bubble sound pressure can be approximated by a decaying sinusoid [Pumphrey and Elmore 1990],

$$p_B(r,t) = \frac{D_B e^{-\beta_B \cdot (t - r/c)}}{r} cos\theta e^{i \cdot (\omega t - kr)} \tag{4}$$

where β_B is a damping constant, and ω is the angular resonance frequency. The angular frequency is evaluated by the Minnaert's formula [Minnaert 1933]:

$$\omega = \frac{1}{a_0} \sqrt{\frac{3\gamma P_0}{\rho_0}}, \tag{5}$$

where a_0 is the bubble radius, γ is the specific heat of the air, and P_0 is the static pressure in the water around the bubble, approximately equal to the atmospheric pressure.

There are three different mechanisms by which a bubble may lose energy, namely viscous losses, thermal losses and acoustic radiation losses. However, the viscous damping rapidly goes to zero for any bubble with a radius $> 0.1mm$. The typical radius for a bubble entrained by a raindrop impact is $0.16mm \sim 0.47mm$, so we only need to consider thermal damping and radiative damping, which can be approximated as

$$\delta_{\text{th}} = \sqrt{\frac{9\omega(\gamma-1)^2}{8G_{\text{th}}}\pi}, \quad \delta_{\text{rad}} = \sqrt{\frac{3\gamma P_0}{\rho_0 c^2}} \tag{6}$$

where G_{th} is a dimensionless constant. Hence, the damping constant can be calculated by $\beta_B = \omega(\delta_{\text{th}} + \delta_{\text{rad}})/2$. We refer the reader to Leighton [1994] for a comprehensive introduction.

The behavior of a bubble after it begins to oscillate was well studied [Langlois et al. 2016; Leighton 1994; Moss et al. 2010; Zheng and James 2009], but it is less clear how it gets the initial dipole strength D_B. The initial energy of the entrained bubbles by raindrop impacts is different from the initial energy of other bubbles in a large body of water. Pumphrey and Elmore [1990] noted that the raindrop size and raindrop impact velocity would affect the sound of entrained bubbles. After entrainment, the bubble is compressed to the radius a_0 by two additional pressures, the hydrostatic pressure $P_H = \rho g h$ and the Laplace pressure $P_L = 2\sigma/a_0$, where σ is the surface tension of water. Figure 3(b) illustrates the bubble radius change after entrainment. Hence, following [Pumphrey

Table 1. Distributions of raindrop diameters. The rainfall is divided into three intensities, namely light rain, heavy rain and very heavy rain. For each scale, we list the probability of the raindrop diameter falling into one of the three intervals based on the data in [Medwin et al. 1992].

raindrop diameter	light rain	heavy rain	very heavy rain
$0.8mm \sim 1.1mm$	84%	32%	24%
$1.1mm \sim 2.2mm$	16%	61%	52%
$> 2.2mm$	0%	7%	24%

and Elmore 1990], we construct the initial dipole strength D_B as

$$D_B = 2hk(\rho g h a_0 + 2\sigma), \tag{7}$$

where h is the vertical distance from the bubble to the water surface, which is approximated as $h = (g/3)^{1/4} d^{3/4} {V_I}^{1/2}$ by equating the gravitational potential energy to the kinetic energy. Finally, we establish the relation between raindrop diameter d and bubble radius. Based on the data in [Pumphrey and Elmore 1990], we can approximate the bubble radius as $a_0 = 15\sqrt{\frac{d}{V_I}}$.

4.2 Statistical Model for BR-sound Generation

To achieve plausible rain sound, we need to estimate the distribution of impacts and the entrainment of bubbles. However, computing sound for each raindrop individually is notoriously expensive. Instead, we present a statistical approach to generate rain sound in an efficient way. Even without knowing frequency and impact velocity from the simulator, we can approximately synthesize rain sound through raindrop radius distribution, since all variables can be directly or indirectly obtained through raindrop radius as shown in the previous section. We cluster the raindrop sounds in small parameter intervals into BR-sound based on statistical distributions as the basic unit for the final rain sound. For a rainfall scene at a particular time, we take into consideration both impact and bubble sound sources as well as the distance from the listener to the sound source. Hence, the BR-sound bank is parameterized with two parameters: number of raindrops and distance from listener.

With different rainfall intensities measured by precipitation rate (light rain or heavy rain), the distribution of raindrop diameter is different. We calculate the percentage of raindrop diameter distribution based on the data in [Medwin et al. 1992] which is listed in Table 1. With the raindrop radius distribution, we can generate BR-sounds by clustering the raindrop sounds in small parameter intervals. However, perhaps surprisingly, bubble sound does not occur for every raindrop impact, but when the bubble oscillation occurs, it is a stronger acoustic source than the initial impact. The observation data in [Pumphrey et al. 1989] shows that raindrops within the diameter range $0.8mm \sim 1.1mm$ can produce bubbles with every impact. Thus, we only calculate the bubble sound for raindrops in this interval.

Considering the number of particles commonly used in rainfall simulation, we subdivide the range of each BR-sound into ten intervals. The starting value and end value of each interval are represented as num_s, num_e (for raindrop number range) and dis_s, dis_e (for distance range). The total raindrop number range is $5000 \sim 10000$ and the length of each interval is 500, i.e., $num_s - num_e = 500$. For each interval, such as $BR[8000, 8500]$, we further divide it into ten intervals according to the distance between the listener and the sound source. The distance range is $0m \sim 10m$ and each interval length is $1m$ ($dis_s - dis_e = 1$). Hence, each BR-sound can be expressed in the form of $BR[8000, 8500, 2, 3]$, of which the former two are the endpoints of the raindrop interval, and the latter two denote the distance interval. We assign these ten intervals as follows:

light rain: [5000,5500], [5500,6000], [6000,6500];
heavy rain: [6500,7000], [7000,7500], [7500,8000], [8000,8500];
very heavy rain: [8500,9000], [9000,9500], [9500,10000].

Moreover, since when a raindrop impacts on a solid surface, the bubble sound is inaudible, we only calculate the impact sound for solid surfaces. Thus, the BR-sound bank can be divided into two parts, one for liquid surfaces BR_{liquid} and the other for solid surfaces BR_{solid}. Specifically, the steps for BR-sound generation can be described as follows:

(1) For each $BR[num_s, num_e, dis_s, dis_e]$, select the median of the interval as the total raindrop number N_M.
(2) Calculate the number of raindrops in three intervals according to N_M and Table 1.
(3) Calculate the raindrop sounds of three intervals separately. The raindrop diameter and its distance are selected randomly in the interval.
(4) Superpose all the impact sound and bubble sound for BR_{liquid} and all the impact sound for BR_{solid}.

In summary, there are 200 BR-sounds (100 for liquid surface and 100 for solid surface) in the BR-sound bank. The total size is 500KB × 200 ≈ 100MB. A 20-second long raindrop sound clip (bubble or impact) takes about 3 seconds to synthesize. Each 5-second long BR-sound generation time takes about 150 seconds to prepare.

4.3 VMD-SV Model for MST Generation

The BR-sound bank can cover the ranges of raindrop variations for most scenes. However, to further distinguish the sound of raindrops falling on surfaces of different materials, we propose a VMD-SV (modified Variational Mode Decomposition and Spectral Variance) method based on signal decomposition and reconstruction to recognize and generate the MSTs (material sound textures). Given a real rainfall recording, the impact sound often mingles with the bubble sound, we therefore seek a suitable way of decomposing the recording to reconstruct the impact sound part.

ALGORITHM 1: MST Generation

```
Input: A recording sound $S_{rec}$
Output: The generated MST $S_{MST}$.
1: Find optimal decomposition number $K$ (Section 4.3.1);
2: $\{u_k\} \leftarrow VMD(S_{rec}, K)$;
3: for each sub-signal $u_k, k = 1, , ..., K$ do
    $f_k$ = FFT($u_k$);
    Calculate the variance $v_k$ of $f_k$;
end
4: Set threshold. $T \leftarrow \max(v_k)/10$;
5: Initialize $S_{MST}$ with a zero array;
6: for $k = 1 : K$ do
    if $(v_k > T)$;
    then
        $S_{MST} \leftarrow v_k/T \times u_k + S_{MST}$;
    end
end
7: Normalize $S_{MST}$ to [-1,1];
```

Yin and Liu [2018] utilized a modified Empirical Mode Decomposition (EMD) algorithm to separate popping sounds (to distinguish combustible material) successfully from fire sound recordings, but the EMD algorithm is not suitable for texture extraction of rain sound, because it is not robust to white noise. However, rain sound contains more frequency information, which is closer to white noise. Therefore, we need a signal decomposition model that is far more resilient to noise. We compared the influence on MST generation with three different decomposition algorithms in § 6, and adopted VMD [Dragomiretskiy and Zosso 2014], a more robust version of EMD, which can recursively decompose a random complex signal into a series of Intrinsic Mode Functions (IMF, amplitude-modulated-frequency-modulated signals, set as u_k in VMD). The main steps of our decomposition process include:

(1) Initialize $\{u_k^1\}$, $\{\omega_k^1\}$, λ^1, K, $n \leftarrow 0$
(2) $n \leftarrow n + 1$
(3) For k from 1 to K
Utilize alternate direction method of multipliers (ADMM) to update u_k^{n+1}, ω_k^{n+1} and λ^{n+1}.
(4) Repeat the above steps until convergence.

Here $\{u_k\}$:=$\{u_1, ..., u_K\}$ and $\{\omega_k\}$:=$\{\omega_1, ..., \omega_K\}$ are shorthand notations for the set of all modes and their center frequencies, respectively. λ is the Lagrangian multiplier and K is the number of modes. From our tests, we observed that the value of K has a direct influence on decomposition accuracy. Hence, we design a procedural way to generate the optimal decomposition number.

4.3.1 Optimal Decomposition Number. Manual determination of K heavily relies on user skills. If we always take a large value for K, it will cause over-decomposition and produce sub-signals that are useless. This will also obstruct the subsequent restructuring process for MSTs. Therefore, we improve the process of setting the K value with a curvature calculation which can automatically obtain the optimal decomposition number K. We start with a small K value, and increase it until the decomposition converges. The procedure is listed below:

(1) Initialize $K = 2$, run the VMD algorithm.
(2) For each sub-signal u_k, after the Hilbert-Huang Transform [Huang et al. 1998], calculate instantaneous frequency and mean value m_{IF}.
(3) Fit quadratic curves with $([1 : K], \mathrm{m}_{IF})$ and the method of least squares, then calculate stationary point p_{sta}.
(4) If $\mathrm{p}_{sta} > K$, update $K \leftarrow K+1$ and repeat Steps (2) and (3). Else, the nearest integer value is selected as K value.

When $K = 2$, the quadratic curve fitting problem will have infinite solutions, therefore, we perform the nearest neighbor interpolation on $([1 : K], m_{IF})$. Determining the optimal K value through the stationary point is based on the observations in our experiments.

4.3.2 SV-based Sound Reconstruction. After we obtain the sub-signals $\{u_k\}$ with the optimal decomposition number K, we are ready to reconstruct the signal for MST generation. In order to determine effective signals in decomposed signals, we choose a threshold T based on the spectral variance v_k. There are several classic features (zero-crossing rate, spectral entropy, etc.) in signal processing for spectral analysis and we choose spectral variance to measure the effectiveness of sub-signals through our experiments. We first analyze the spectrum of sub-signal u_k and calculate the spectral variance v_k. Then, we choose the signals with a spectral variance above the following threshold T:

$$T = \max_k(v_k)/10. \tag{8}$$

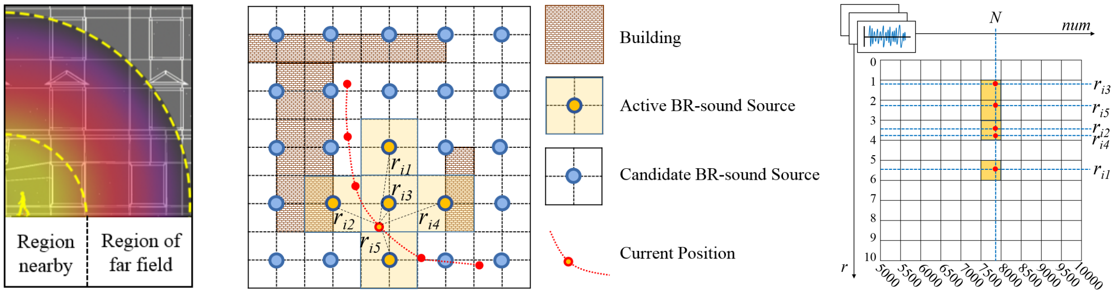
Algorithm 1 summarizes the entire MST generation process.

5 COUPLING WITH RAINFALL ANIMATION

There are two main computational challenges in direct coupling of rainfall simulation with sound synthesis: *i*) a large number of sound sources make the simulation costly at runtime, *ii*) there are no direct physical parameters to match MSTs with the surfaces surrounding the listener. Focusing on handling these challenges, we accelerate the coupling via a novel sound source activator and solve the material issue through a material shader. To further optimize the resulting sound, we design a smoothing process, which can effectively suppress amplitude jumps.

5.1 Sound Source Activation

With the input animation, we can get the time-varying raindrop-numbers and listener (camera) positions. Since human auditory distance is limited, we divide the regions in the rainfall into two types: region nearby and region of far field, as shown in Figure 5(a). The nearby region is a hemisphere with the radius that is set as the maximum value covered by the BR-sound bank. Sound sources in this area are superposed by the following algorithm and the sound sources in the far field



(a) (b) (c)

Fig. 5. The synchronization model in our method. (a) shows the ranges of two types of rain sounds, where the color change represents the attenuation of the sound intensity, yellow indicates the strongest and black indicates the weakest. (b) illustrates the BR-sound source activation model. (c) shows the BR-sound bank used in the synchronization process.

region are superposed directly with a fixed distance value $r_{far} = 10m$. To synthesize the synchronized rain sound for the nearby region, we first need to determine the location and amplitude of the BR-sound source and MSTs for each frame. Since the BR-sound sources in rainfall scenarios are similar, it is unnecessary to repeat calculating a large number of identical sound sources in the process of sound spatialization. Hence, we design the sound source activator and propose a material shader for MSTs to distinguish surfaces of different materials.

5.1.1 BR-sound Activation. In rainfall scenes, since each raindrop is a sound source, the calculation is inevitably costly due to the large number of raindrops. Thus, we build a BR-sound bank with clustered parameters as introduced in § 4. To synthesize sound synchronized with rainfall animation, we need to determine when and where to activate the BR-sound source. To achieve this goal, we introduce a BR-sound source activation process to activate the corresponding sound source over a time span.

First, we rasterize the ground surface of the rainfall animation to a grid. Thus, the impact surface of raindrops is decomposed into a series of BR-sound source blocks as shown in Figure 5(b) and the center of each block is taken as the location of the BR-sound source (the blue or yellow circle in Figure 5(b)). All the BR-sound sources are classified as active BR-sound sources and candidate BR-sound sources. The BR-sound block which contains the listener's current position and its four neighbouring BR-sound blocks are defined as active sound blocks. Correspondingly, the BR-sound sources of all active sound blocks are the active BR-sound sources and the rest are candidate BR-sound sources.

The length ($block_l$) of a BR-sound source block is determined by the BR-sound interval. When a listener is in a BR-sound source block, the maximum distance from the four neighbouring sound sources should be less than the maximum value of the BR-sound interval, written as:

$$\sqrt{(block_l/2)^2 + (3\ block_l/2)^2} \leq 10\text{m}. \tag{9}$$

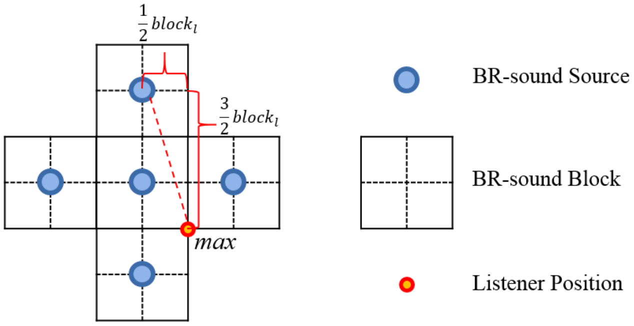


Fig. 6. Determining block length by the maximum distance between the listener and the BR-sound source.

Since the maximum value of the BR-sound interval is 10m, $block_l/2 \approx 3$m. Thus, the length of a BR-sound source block is 6m (see Figure 6).

Next, we discuss how to determine the corresponding BR-sound for each active BR-sound source. For the ith frame, we first calculate the distances between the listener and an active sound sources j $(j = 1, 2, 3, 4, 5)$:

$$r_{ij} = \sqrt{(P(x_i, y_i, z_i) - P_a(x_i^j, y_i^j, z_i^j))^2} \tag{10}$$

where $P(x_i, y_i, z_i)$ (and $P_a(x_i^j.y_i^j, z_i^j)$) represent the positions of the listener (and active sound source j, respectively) at the ith frame. The two numbers, distance r_{ij} and raindrop number N, can uniquely determine a BR-sound as shown in Figure 5(c). Finally, based on the position and BR-sound for each active BR-sound source in each frame, we can generate the final BR-sound through the bidirectional sound transport algorithm [Cao et al. 2016]. As the position of the listener is continuous in time, we found it sufficient to update the active BR-sound source data every ten frames through experimentation.

5.1.2 Material Shader. When the raindrops impact on surfaces of different materials, the rain sound generated can differ.

To distinguish rain sounds for surfaces of distinct materials, we introduce the material shader to enrich the rain sound. Since the MSTs are generated from recordings, they do not contain available physical parameters (like the distance and raindrop number for BR-sounds) for matching the animation. Thus, the activation of MSTs needs to address two main challenges:

(1) A strategy needs to be designed to synchronize the MSTs with the final BR-sound. The core issues are that where MSTs should be added and how long their durations should be.
(2) The amplitude range of the input MSTs should be tuned to a scale balanced with the BR-sound. The main obstacle is the absence of physical parameters for matching.

The solution to the first problem is based on our activation process. When an activated BR-sound source block overlaps with objects in the scene (see Figure 5(b)), the material shader corresponding to the building is activated. Therefore, the duration of a material shader is from the first appearance of an object in an active BR-sound source block to its deactivation. The position of the material sound source is chosen as the center coordinates of the object. Thus, the distance between an object and listener is

$$r_o = \sqrt{(P(x_i, y_i, z_i) - P_b(x_i, y_i, z_i))^2} \tag{11}$$

where $P_b(x_i.y_i, z_i)$ represents the position of the MST sound source at the ith frame.

To resolve the second problem, we scale the amplitude range of an MST as follows:

$$MST' = MST \times \overline{A_{BR}} / \overline{A_{MTS}} \tag{12}$$

where MST' represents the final MST after scaling, MST refers to the original sound texture, $\overline{A_{BR}}$ indicates the average amplitude of the BR-sound in the same activated BR-sound source block and $\overline{A_{MTS}}$ is the average amplitude of the MST. Unlike the smoother change of BR-sound due to the multiple similar sound sources, MST has to be updated frame by frame with the bidirectional sound transport algorithm [Cao et al. 2016].

In summary, the synchronization process involves the following steps:

(1) Rasterize the ground surface of the input animation.
(2) Update the active BR-sound source blocks according to the input animation.
(3) Iterate through the active sound sources and calculate distances between the listener and active sound sources.
(4) Calculate duration and amplitude range of MSTs.
(5) Implement the sound propagation process and generate the final rain sound $rain(t)$ by superposing final BR-sound and final MSTs.

5.2 Smoothing

Since the final rain sound is stitched from the BR-sounds, when the number of raindrops varies greatly in the scene, there can be a sense of splicing. To avoid the amplitude discontinuity, we have smoothed the final rain sound. We first calculate the envelope of the final rain sound with an efficient nonlinear low-pass filter proposed in [Peltola et al. 2007]:

$$e(n) = (1 - b(n))|x(n)| + b(n)e(n-1) \tag{13}$$

where n stands for the current frame number, $x(n)$ is the rectified input, $e(n-1)$ is the previous output of the filter, $b(n)$ is the regulating parameter that combines previous output $e(n-1)$ and current input $x(n)$. The value of $b(n)$ depends on whether the input value goes down or not, in the former case it is $b_{down} = 0.995$, and in the latter case it is $b_{up} = 0.8$. There can be jumps in the envelope curve of synthesized sound (see the yellow curve in Figure 7). Thus, we tried four popular smoothing algorithms, namely Gaussian smoothing, polynomial fit smoothing, exponential smoothing and box smoothing. We found that the polynomial fit smoothing and box smoothing can achieve better performance for our rain sound. As shown in Figure 7(b), we found that the box smoothing method has better effects with large slopes (left dashed box) and the polynomial fit smoothing method

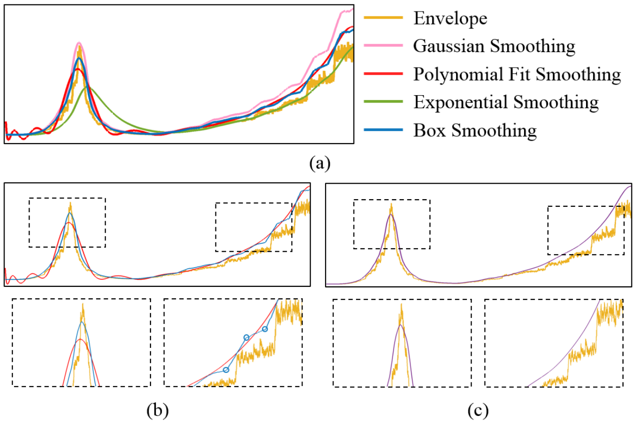


Fig. 7. Comparison among different smoothing methods. We compare the envelope curve and results of four smoothing methods in (a). The two best results are further compared in (b), where the blue circles represent break points. The final smoothing result and the envelope curve are presented in (c).

Table 2. Physical constants and their values in our simulations.

Parameter	Value	Description
g	$9.8\ m/s^2$	gravitational acceleration
G_{th}	$1.6\ \times 10^6 s/m$	thermal damping constant
c	$343\ m/s$	sound speed in air
c_0	$1497\ m/s$	sound speed in water
ρ	$1.29\ kg/m^3$	air density
ρ_0	$1000\ kg/m^3$	water density
γ	1.4	specific heat ratio of air
P_0	$101.325\ kPa$	atmospheric pressure

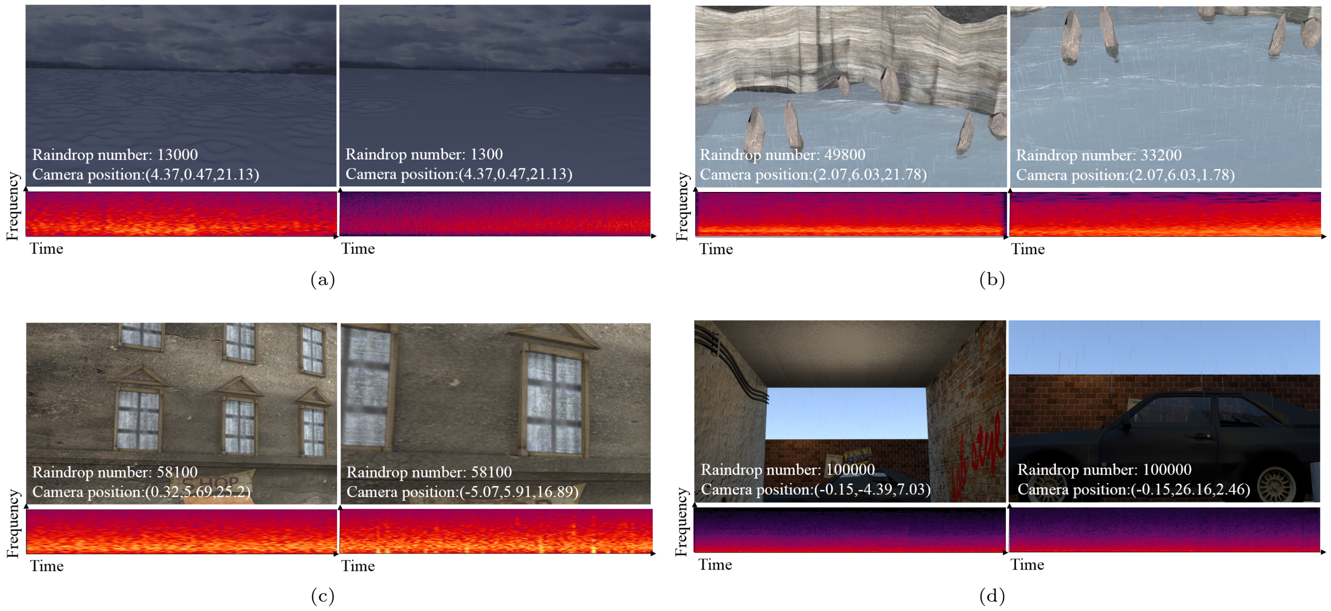

Raindrop number: 13000
Camera position:(4.37,0.47,21.13)
Raindrop number: 1300
Camera position:(4.37,0.47,21.13)
Frequency
Time
Time
(a)
Raindrop number: 49800
Camera position:(2.07,6.03,21.78)
Raindrop number: 33200
Camera position:(2.07,6.03,1.78)
Frequency
Time
Time
(b)
Raindrop number: 58100
Camera position:(0.32,5.69,25.2)
Raindrop number: 58100
Camera position:(-5.07,5.91,16.89)
Frequency
Time
Time
(c)
Raindrop number: 100000
Camera position:(-0.15,-4.39,7.03)
Raindrop number: 100000
Camera position:(-0.15,26.16,2.46)
Frequency
Time
Time
(d)

Fig. 8. Synthesized rain sound that changes with the number of raindrops and the movement of camera. In each group, the top row shows two animation frames and the bottom row demonstrates the corresponding spectrogram of each frame. The parameters of animation are shown at the bottom left corner of each frame.

Table 3. Timing for experiments with different domain sizes.

Example	Domain size (m)	Length (s)	Our total cost without SSA (s)	Our total cost with SSA (s)	Zita's [2003] cost (s)	Speedup
DNR (light rain)	8 × 8 × 8	10	0.261	0.253	11.241	44×
DNR (medium rain)	8 × 8 × 8	10	0.273	0.272	15.178	55×
DNR (heavy rain)	8 × 8 × 8	10	0.235	0.231	20.112	87×
DPL (lake)	20 × 12 × 20	15	0.379	0.362	30.195	83×
DPL (block)	40 × 50 × 20	20	0.842	0.527	50.142	95×
DPL (street)	140 × 19 × 19	25	1.116	0.641	62.884	98×
DSM (country road)	29 × 29 × 26	16	0.615	0.413	37.334	90×

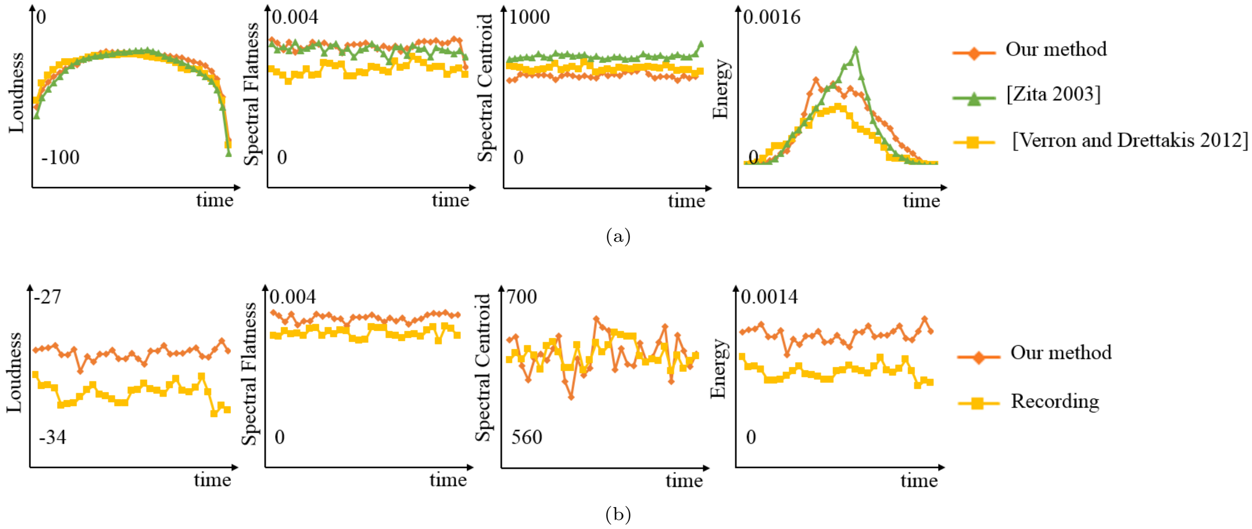
has better effects with small slopes (break points in right dashed box). Thus, we calculate the resulting curves for both methods, and compute their intersections. Then we choose to use one of the curve segments for each interval formed by the intersections based on a simple rule:

(1) if the envelope contains a slope $> \tan(60°)$, use box smoothing;
(2) otherwise, use polynomial fit.

Figure 7(c) presents the final smoothing curve and the original envelope curve, which shows that the piecewise smoothing algorithm can effectively generate ideal results. Then we can get the smoothed rain sound based on the smoothed amplitude. In § 6, we show the waveforms of our synthesized rain sound before and after smoothing, which confirms the effectiveness of our smoothing algorithm.

6 RESULTS AND DISCUSSIONS

We tested our method on various rainfall scenes at different scales. All the experiments were conducted with the same hardware: Intel Core i5-4460 3.20 GHz CPU, Nvidia GeForce GTX 745 GPU, 8 GB RAM. In our results, we synthesized all the BR-sounds at a sampling rate of 44,100 Hz and the animation frame rate is 30fps. Our implementation made use of Matlab to load and play sounds. All the rain scene models were constructed by POPs system in Houdini FX 15. The values of the parameters are given in Table 2. We refer the reader to the accompanying video for all our animation and audio results.



(a)

(b)

Fig. 9. Comparison with state-of-the-art methods and recorded sound. (a) is the spectra comparison with [Verron and Drettakis 2012; Zita 2003] and line charts of content-based descriptors. (b) is the spectra comparison with recorded sound and line charts of content-based descriptors. The audio descriptors are loudness, spectral flatness, spectral centroid and energy, respectively.

6.1 Rainfall scenarios

First, we synthesized the rain sound for different rainfall scenarios. In different rainfall animations, we change the number of raindrops, position of listener and surface material to validate the effectiveness of our method. All the sounds were synthesized with and without the sound source activator (SSA for short in Table 3). We compared the sound generated by our approach to the sound synthesized by a purely physical model [Zita 2003] [1], and the results are shown in Table 3 which demonstrate that our method is about 40∼100× faster.

Different numbers of raindrops. Figure 8(a) presents two frames with different numbers of raindrops. From the spectra on the right side, we can see that the sound results are different when the number of raindrops changes. In this scenario, due to the small domain size, there are only 9 sound sources in total, so the effect of the sound source activator is less significant. From the three rows of "DNR" in Table 3 (1k-5k raindrops for light rain, 5k-10k raindrops for medium rain and 10k-15k raindrops for heavy rain), we can observe that the rain intensity (number of raindrops) has no effects on the efficiency of our algorithm, but obviously affects Zita's method [2003]. This is one advantage of our BR-sound bank: the computational cost is independent of the precipitation intensity, as long as the domain size remains unchanged, since we only need to select different BR-sounds with the number of raindrops simply used as one of the parameters.

Different locations of the listener. Figures 8(b)- 8(d) show frames of a video with the listener (camera) in different positions. In each group, the right column is the corresponding spectrogram. From the frequency distribution of the spectrogram, we can observe the difference of sound when the camera is in different positions. For a liquid surface (Figure 8(b)), when the listener is close to the surface of a lake, the low frequency sound caused by bubble vibration is more noticeable. However, for a solid surface with the same material (Figure 8(c)), the change in the listener's position has little effect on the resulting rain sound. When the listener walks from indoors to outdoors (Figure 8(d)), the change of rain sound is obvious due to the occlusion effects of the buildings in the experiment. In these cases, our sound synthesis is around 90× faster than [Zita 2003] owing to that the sound source activator effectively reduces the simulation cost.

Different surface materials. A rainfall scenario with a cloth canopy and metal street lamps is explored in our experiment (see Figure 1). As shown in Figure 1 (see point A on the bottom row), when the listener is below the canopy, although there is only little information about the canopy (top left corner), we can clearly perceive how our position affects the change of rain sound. Moreover, we compared the sound results with and without the MSTs, and experiments demonstrate that the exploration of material sound can greatly enhance users' perceptions of scenes.

6.2 Comparison and Validation

Comparison with the state-of-the-arts: Although Zita [2003] provided a fine solution for rain sound synthesis, it only takes into account the impact sound generated by raindrops and the surface. However, only considering the impact sound would make the synthesized sound resemble white noise. The method in [Verron and Drettakis 2012] is a granular synthesis

[1] Since our method is sequential, we only compare with the sequential method in [Zita 2003].

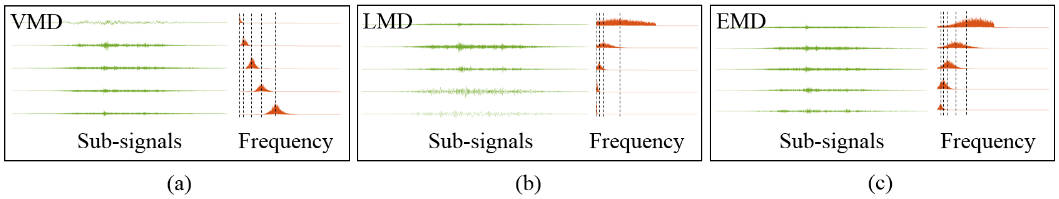


Fig. 10. Result comparison among different decomposition algorithms. From (a) to (c), the corresponding algorithms are VMD, LMD, and EMD, respectively. We mark the frequency center with the dotted lines.

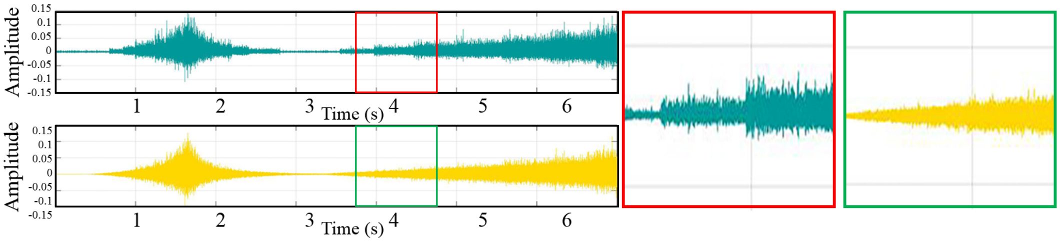


Fig. 11. Comparison of sound results before and after smoothing. The green waveform is the result without smoothing and the yellow waveform is the result after smoothing. The red box on the right shows the magnified view for comparing the details.

approach with the composition of four rain sound atoms, and thus its results may contain some noticeable traces of stitching.

To validate the sound effects, we compared the synthesized sound generated by [Verron and Drettakis 2012; Zita 2003] with our work. In order to visualize the differences among the sounds, we calculate the content-based descriptors for comparison of various aspects of the sound. Audio descriptors are calculated through the method proposed in [Schwarz and Schnell 2010]. In the spectral flatness line chart, we find that since the method of [Verron and Drettakis 2012] is particle-based, the spectral flatness value is lower which varies more widely than physically-based methods. In the energy line chart, we observe that the energy value of Zita's result is higher and the change of energy [2] is steeper, so the resulting sounds are closer to white noise.

Comparison with natural rain sound: To verify our experimental results, we also compared the synthesized rain sounds with the natural rain sounds from a website [3]. Since the waveforms of the white noise and the rain sound are similar, it is difficult to compare the differences from the spectra as shown in Figure 9(b). In order to compare the difference between the timbre of the sounds, we again use the audio descriptors calculated through the method proposed in [Schwarz and Schnell 2010]. It can be observed from the line charts that the descriptors share similar distributions. Rain sound recording is often mixed with some other noises, such as wind and thunder, so our synthesized result has a higher loudness and energy value.

Validation on the effect of decomposition: To find the most suitable decomposition algorithm for rain sound, we decompose the same rain sound recording with three decomposition algorithms (EMD, LMD, VMD). The results are illustrated in Figure 10, where we show the top five sub-signals and the corresponding frequency distributions. It is clear that the frequency distributions of sub-signals based on LMD and EMD are overlapping. Thus, VMD is more suitable for rain sound decomposition.

Validation on the effect of smoothing: Figure 11 shows the sound waveforms before and after smoothing, where the green waveform is the result before smoothing and the yellow waveform is the smoothed result of the green waveform. For clarity, we magnify the sound in the corresponding color box. It can be seen that the acoustic wave discontinuity has been effectively suppressed.

6.3 User study

To further assess the effectiveness of our approach, we designed three perceptual user studies to evaluate the quality of the synthesized results. We invited 50 participants (24 females and 26 males) with ages ranging from 18 to 50. Among them, 10 have specialized computer graphics knowledge and all of them have normal hearing.

The first experiment is to evaluate the similarity of synthesized sound and natural rain sound. In this experiment, the participant is shown a series of audio clips. The collections consist of three synthesized samples from our method, three audio clips from Zita's method and four real audio clips that are recordings of natural rain. In the experiment, one clip is shown per page and the participant is asked to rate 0 or 1 for each clip, where 1 is labeled "natural rain sound" and

[2] We calculated noise part energy which estimates the power of the noise (non-harmonic) part of the signal. More details can be found in [Schwarz and Schnell 2010].

[3] www.audioblocks.com

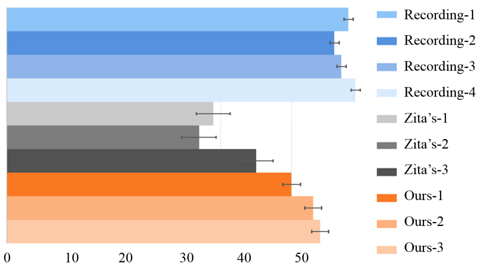


Fig. 12. User survey results of the first experiment. The error bars are at 95% confidential level.

	Animation 1	Animation 2	Animation 3
Audio 1	8.43	3.26	2.45
Audio 2	1.26	9.21	1.63
Audio 3	2.33	2.62	8.95

Fig. 13. User survey results of the third experiment. The numbers in each box represent score and the highest score for each column is marked in orange.

0 "synthesized rain sound". In addition, participants will not be told the number of natural rain sounds and synthesized rain sounds so that the participant can give all 0 or all 1 scores. Figure 12 presents the sum of the scores of the ten audio clips. It can be seen that over 80% of participants think the synthesized rain sounds by our method are natural. In fact, all of our scores are higher than Zita's scores, which indicates that the rain sound synthesized by our method is more realistic than that synthesized by Zita's method.

The second experiment is to evaluate the quality of sound results which are synthesized by Zita's method and our method, respectively. In the experiment, we present five pairs of audio clips. Each page contains the audio from one of our demo scenarios and the audio synthesized by Zita's method with the same parameter interval. For every pair of audio clips, a paired T-test is conducted to evaluate the difference between the synthesized rain sound by Zita's method and the synthesized rain sound by our method. In each scenario, the participant is asked three questions: "Are these two audio/video clips the same or different?", "Which audio/video clip do you prefer?" and "How strongly do you feel about this preference?" The score for each clip is on a scale from 1 to 10, where 1 is labeled "Do not prefer" and 10 "Very much prefer".

A paired T-test is performed on these scores to check if μ_a is significantly greater than μ_b using the following hypotheses $H_0 : \mu_a \leq \mu_b$, $H_1 : \mu_a > \mu_b$, where μ_a represents the score of the synthesized sound by our method, while μ_b represents the score of the synthesized sound by Zita's method. Hypothesis H_0 means that Zita's sound results have a higher score. Table 4 shows all 5 paired T-test results. Note that all P values are less than 0.0005, and all T values are higher than 1.6839, indicating that H_0 is rejected with statistical significance while H_1 is accepted. This concludes that our result achieves a significantly better sound effect than Zita's result.

The third experiment is used to verify the synchronization of sound and animation. To verify the effectiveness of our mapping function, there are three scenes in the experiment. In different scenarios, the number of raindrops and the change of listener position are different. We use the three synthesized rain sounds to dub three animations separately and the volunteers are asked to rate the video with different sounds on a scale from 1 to 10, where 1 is labeled as "Not synchronized" and 10 "Very synchronous". Figure 13 shows the average score of the experiment and we mark the highest score of each column in orange. It can be observed that the animation clip and the audio clip whose names are the same have the highest score, which means the subject can correctly select the matching sounds and animations.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a novel physically-based statistical method for generating rain sound, which allowed for a more realistic simulation that varies with the location of a listener. The two mechanisms in the raindrop model exhibited more detailed visual correspondences. Moreover, we enriched the impact sound on different materials with MSTs, which were generated by a novel VMD-SV method. With the statistic model, we generated a bank of clean sound clips (BR-sounds) without the interference of environmental sounds, which can be hard to avoid in recordings. We proposed a novel visual-to-audio mapping scheme which drastically reduced simulation cost. Our user study suggested that the perceived realism of rain sounds synthesized by our approach is comparable to recorded sound.

Although our rain sound model enables synthesizing natural rain sound, there are still some limitations. The example guided MST generation may limit the category of sound. Building sound models for material sounds is an interesting direction to explore. Also, improving the simple assumptions about spherical raindrops as mentioned in the modeling section could also be considered for future work. Moreover, it remains another future work to investigate how to accelerate the sound synthesis for large scale dynamic environments. Perhaps exploring parallel computing techniques such as GPU (Graphics Processing Units) computing may further speed up the simulation. Finally, finding overall contributions from other environmental variables, such as the strength of wind, on the sound of rain remains an open problem.

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. Prof. Tong was supported in part by NSF under grant no. DMS-1721024. We would like to thank Emily Ribando-Gros for proofreading.

Table 4. User survey results of the second experiment: audio synthesized by our method vs. audio synthesized by Zita's method.

Scenarios	Same	Diff	Prefer Zita's	Prefer Ours	T value	P value
soft rain	10% (5)	90% (45)	30% (15)	70% (35)	14.382	< 0.0005
middle rain	14% (7)	86% (43)	34% (17)	66% (33)	4.756	< 0.0005
heavy rain	16% (8)	84% (42)	24% (12)	76% (38)	6.789	< 0.0005
lake	2% (1)	98% (49)	14% (7)	86% (43)	4.213	< 0.0005
block	2% (1)	98% (49)	18% (9)	82% (41)	5.786	< 0.0005

REFERENCES

C. Cao, Z. Ren, C. Schissler, D. Manocha, and K. Zhou. 2016. Interactive sound propagation with bidirectional path tracing. *ACM Transactions on Graphics* 35, 6 (2016), 180:1–180:11.

J.N. Chadwick and D.L. James. 2011. Animating fire with sound. *ACM Transactions on Graphics* 30, 4 (2011), 84:1–84:8.

H. Cheng and S. Liu. 2019. Haptic force guided sound synthesis in multisensory virtual reality (VR) simulation for rigid-fluid interaction. In *Proceedings of IEEE VR*. 1–9.

P.R. Cook. 1997. Physically informed sonic modeling (PHISM): Synthesis of percussive sounds. *Computer Music Journal* 21, 3 (1997), 38–49.

P.R. Cook. 2002. Modeling bill's gait: Analysis and parametric synthesis of walking sounds. In *Proceedings of Audio Engineering Society Conference*. 73–78.

N. Dingle and Y. Lee. 1972. Terminal fallspeeds of raindrops. *Journal of Applied Meteorology* 11, 5 (1972), 877–879.

K.V.D. Doel. 2005. Physically-based models for liquid sounds. *ACM Transactions on Applied Perception* 2, 4 (2005), 534–546.

K. Dragomiretskiy and D. Zosso. 2014. Variational mode decomposition. *IEEE Transactions on Signal Processing* 62, 3 (2014), 531–544.

A. Farnell. 2010. *Designing Sound*. The MIT Press.

G.J. Franz. 1959. Splashes as sources of sound in liquids. *Journal of the Acoustical Society of America* 31, 8 (1959), 1080–1096.

Y.P. Guo and J.F. Williams. 1991. A theoretical study on drop impact sound and rain noise. *Journal of Fluid Mechanics* 227 (1991), 345–355.

M. S. Howe and N. A. A. Hagen. 2011. On the impact noise of a drop falling on water. *Journal of Sound and Vibration* 330, 4 (2011), 625–635.

N.E. Huang, Z. Shen, S.R. Long, M.C. Wu, H.H. Shi, Q. Zheng, N.C. Yen, C.T. Chi, and H.H. Liu. 1998. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. 454, 1971 (1998), 903–995.

P.W. Jacobus. 1991. *Underwater sound radiation from large raindrops*. Technical Report. Naval Postgraduate School Monterey CA.

T.R. Langlois, C. Zheng, and D.L. James. 2016. Toward animating water with complex acoustic bubbles. *ACM Transactions on Graphics* 35, 4 (2016), 95:1–95:13.

T. Leighton. 1994. *The acoustic bubble*. Academic press.

T. Lentz, D. Schröder, M. Vorländer, and I. Assenmacher. 2007. Virtual reality system with integrated sound field simulation and reproduction. *EURASIP Journal on Advances in Signal Processing* 2007, 1 (2007), 1–19.

S. Liu and Z. Yu. 2015. Sounding fire for immersive virtual reality. *Virtual Reality* 19, 3–4 (2015), 291–302.

M.S. Longuet-Higgins. 1990. An analytic model of sound production by raindrops. *Fluid Mechanics* 214 (1990), 395–410.

J.S. Marshall and W.M.K. Palmer. 1948. The distribution of raindrops with size. *Journal of Meteorology* 5 (1948), 165–166.

H. Medwin, J.A. Nystuen, P.W. Jacobus, L.H. Ostwald, and D.E. Snyder. 1992. The anatomy of underwater rain noise. *Journal of the Acoustical Society of America* 92, 3 (1992), 1613–1623.

R. Mehra, N. Raghuvanshi, L. Antani, A. Chandak, S. Curtis, and D. Manocha. 2013. Wave-based sound propagation in large open scenes using an equivalent source formulation. *ACM Transactions on Graphics* 32, 2 (2013), 19:1–19:13.

M. Minnaert. 1933. On musical air-bubbles and the sounds of running water. *Philosophical Magazine* 16, 104 (1933), 235–248.

A. Misra, P. R. Cook, and G. Wang. 2006. TAPESTREA: sound scene modeling by example. In *Proceedings of SIGGRAPH Sketches*. 177.

W. Moss, H. Yeh, J.M. Hong, M.C. Lin, and D. Manocha. 2010. Sounding liquids: Automatic sound synthesis from fluid simulation. *ACM Transactions on Graphics* 29, 3 (2010), 21:1–21:13.

J.A. Nystuen. 1986. Rainfall measurements using underwater ambient noise. *Journal of the Acoustical Society of America* 79, 4 (1986), 972–982.

J.A. Nystuen. 1991. Ambient sound in the ocean produced by heavy precipitation and the subsequent predictability of rainfall rate. *Journal of the Acoustical Society of America* 90, 4 (1991), 2301.

L. Peltola, C. Erkut, P.R. Cook, and V. Valimaki. 2007. Synthesis of hand clapping sounds. *IEEE Transactions on Audio, Speech, and Language Processing* 15, 3 (2007), 1021–1029.

A. Prosperetti, L.A. Crum, and H.C. Pumphrey. 1989. The underwater noise of rain. *Journal of Geophysical Research Oceans* 94, C3 (1989), 3255–3259.

H.C. Pumphrey, L.A. Crum, and L. Bjørnø. 1989. Underwater sound produced by individual drop impacts and rainfall. *Journal of the Acoustical Society of America* 85, 4 (1989), 1518–1526.

H.C. Pumphrey and P.A. Elmore. 1990. The entrainment of bubbles by drop impacts. *Journal of Fluid Mechanics* 220 (1990), 539–567.

N. Raghuvanshi and J. Snyder. 2014. Parametric wave field coding for precomputed sound propagation. *ACM Transactions on Graphics* 33, 4 (2014), 38:1–38:11.

Z. Ren, H. Yeh, and M.C. Lin. 2013. Example-guided physically based modal sound synthesis. *ACM Transactions on Graphics* 32, 1 (2013), 1:1–1:16.

C. Roads. 1988. Introduction to granular synthesis. *Computer Music Journal* 12, 2 (1988), 11–13.

C. Schissler, R. Mehra, and D. Manocha. 2014. High-order diffraction and diffuse reflections for interactive sound propagation in large environments. *ACM Transactions on Graphics* 33, 4 (2014), 39:1–39:12.

D. Schwarz and N. Schnell. 2010. Descriptor-based sound texture sampling. In *Proceedings of International Conference on Sound and Music Computing*. 510–515.

C.D. Scofield. 1992. *Oscillating microbubbles created by water drops falling on fresh and salt water: amplitude, damping and the effects of temperature and salinity*. Ph.D. Dissertation. Naval Postgraduate School.

G. Strobl, G. Eckel, and D. Rocchesso. 2006. Sound texture modeling: A survey. In *Proceedings of International Conference on Sound and Music Computing*. 1–7.

L.L. Thompson. 2006. A review of finite-element methods for time-harmonic acoustics. *Journal of the Acoustical Society of America* 20, 3 (2006), 1315–1330.

C. Verron and G. Drettakis. 2012. Procedural audio modeling for particle-based environmental effects. In *Proceedings of Audio Engineering Society Convention 133*.

Q. Yin and S. Liu. 2018. Sounding solid combustibles: non-premixed flame sound synthesis for different solid combustibles. *IEEE Transactions on Visualization and Computer Graphics* 24, 2 (2018), 1179–1189.

Z. Zhang, N. Raghuvanshi, J. Snyder, and S. Marschner. 2018. Ambient sound propagation. *ACM Transactions on Graphics* 37, 6 (2018), 184:1–184:10.

C. Zheng and D.L. James. 2009. Harmonic fluids. *ACM Transactions on Graphics* 28, 3 (2009), 37:1–37:12.

A. Zita. 2003. *Computational real-time sound synthesis of rain*. Master's thesis. Linköping University.