Acoustic texture rendering for extended sources in complex scenes

ZECHEN ZHANG, Cornell University
NIKUNJ RAGHUVANSHI, Microsoft Research
JOHN SNYDER, Microsoft Research
STEVE MARSCHNER, Cornell University

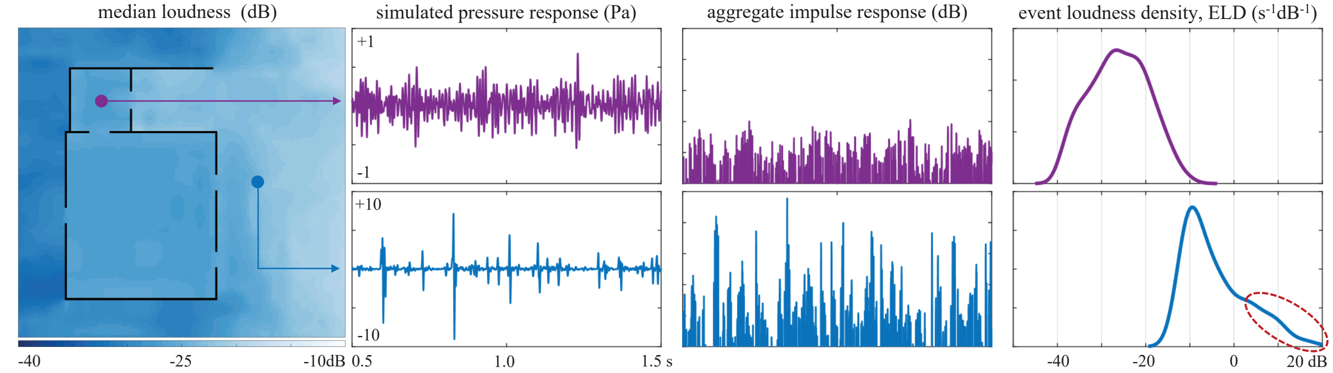

Fig. 1. Capturing acoustic texture variation in RiverHouse. The source in this simple scene represents a river located to the right of the house. We illustrate what our method does at two particular points: inside a small room (top row) and outside near the river (bottom row). Running a numerical simulation of sound transport which stochastically emits pulses over the source into this scene, we obtain pressure responses for each listener position (second column). After sparsity-regularized deconvolution, these are converted to an aggregate impulse response representing arrival events at the listener (third column). We then extract an event loudness density, encoding temporal density of arrivals at the listener position as a function of loudness (fourth column). Source events are multiplied inside the house due to reverberation yielding many similarly quiet arrivals. Whereas outside and near the source, fewer arrivals are recorded overall, with some infrequent but loud events that stand out from the rest (dashed red area).

Extended stochastic sources, like falling rain or a flowing waterway, provide an immersive ambience in virtual environments. In complex scenes, the rendered sound should vary naturally with listener position, differing not only in overall loudness but also in texture, to capture the indistinct murmur of a faraway brook versus the bright babbling of one up close. Modeling an ambient sound as a collection of random events such as individual raindrop impacts or water bubble oscillations, this variation can be seen as a change in the statistical distribution of events heard by the listener: the arrival rate of nearby, louder events relative to more distant or occluded, quieter ones. Reverberation and edge diffraction from scene geometry multiply and mix events more extensively compared to an empty scene and introduce salient spatial variation in texture. We formalize the notion of acoustic texture by introducing the *event loudness density* (ELD), which relates the rapidity of received events to their loudness. To model spatial variation in texture, the ELD is made a function of listener location in the scene. We show that this ELD field can be extracted from a single wave simulation for each extended source and rendered flexibly using a granular synthesis pipeline, with grains derived procedurally or from recordings. Our system yields believable, real-time changes in acoustic texture as the listener moves, driven by sound propagation in the scene.

CCS Concepts: • **Applied computing** → **Sound and music computing**; • **Computing methodologies** → **Virtual reality**;

Additional Key Words and Phrases: Acoustic texture, diffraction, extended source, event loudness density, granular synthesis, perceptual coding, sound propagation, spatial audio, wave simulation

ACM Reference Format:
Zechen Zhang, Nikunj Raghuvanshi, John Snyder, and Steve Marschner. 2019. Acoustic texture rendering for extended sources in complex scenes. *ACM Trans. Graph.* 38, 6, Article 222 (November 2019), 9 pages. https://doi.org/10.1145/3355089.3356566

1 INTRODUCTION

Many types of natural sound sources can be modeled as the superposition of spectrally-similar atomic *source events* overlapping in time and distributed over the source's spatial extent, such as individual rain drops, oscillating bubbles in a stream, or bird tweets in a flock. These events occupy similar frequency bands within human auditory perception so we perceive them as an aggregate. Yet not all detail is lost and we are able to hear certain statistical properties which we call the *acoustic texture*. For instance, a faraway stream sounds noise-like but becomes a more distinct babbling with individually recognizable drips and gurgles as one gets closer. The

Authors' addresses: Zechen Zhang, zz335@cornell.edu, Cornell University; Nikunj Raghuvanshi, nikunjr@microsoft.com, Microsoft Research; John Snyder, johnsny@microsoft.com, Microsoft Research; Steve Marschner, srm@cs.cornell.edu, Cornell University.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2019 Association for Computing Machinery.

0730-0301/2019/11-ART222 $15.00
https://doi.org/10.1145/3355089.3356566

texture varies not only with distance but also from sound propagation within the scene: rain sounds different when heard indoors through a door compared to outside, not just because it gets fainter, but also because events get mixed at the door and are multiplied via reverberation within the room.

We seek to efficiently capture and render this acoustic texture variation to improve the realism of ambiences in games and virtual reality. The problem is challenging. Brute force rendering where each individual source event's emitted signal is convolved with its acoustic impulse response captures all audible detail but at tremendous CPU cost. Zhang et al. [2018] models variation in overall loudness and directionality from a single wave simulation, producing a constant far-field texture lacking the spatial variation we wish to model. Our approach also performs a wave solution to efficiently propagate a massive superposition of sound radiated from the source events. Instead of emitting a temporally and spatially dense signal representing idealized noise, we stay closer to reality by stochastically emitting band-limited pulses over time and source extent. At each potential listener location, we deconvolve away the injected pulse to obtain an *aggregate impulse response* capturing times and loudnesses of *arrival events* after propagation through the scene. We employ a sparsity-regularized deconvolution in the time domain that contends with numerical dispersion errors to produce sharp estimates of event arrival time.

Our key contribution is to formulate and compactly encode acoustic texture with a novel *event loudness density* (ELD) computable from this propagated aggregate response. Assuming a stationary stochastic process, we observe that it is the distribution of event loudnesses in terms of their arrival rate at the listener that determine the perceived acoustic texture. We thus define the ELD as the temporal frequency, or density, of events received at the listener as a function of their loudness.

The ELD distills effects of distance and environmental interaction on sound as it propagates from a given extended source volume to the listener location. With the listener near a large source, a broader, flatter ELD is obtained representing a variety of event loudnesses with some nearby loud events heard over a background of quieter ones. Further away or in a more occluded part of the scene, the ELD becomes quieter overall and more peaked. This reflects extensive mixing of similarly quiet sounds, resulting in a faint and noise-like texture which gives a distant impression. Temporal density is increased by reverberation, resulting in an ELD with larger integral (total event frequency) indoors compared to the same sound source in free field with no scene geometry. Figure 2 provides more detail on the relationship between acoustic texture and ELD.

The ELD varies smoothly over space, yielding a compact representation after compression in our test scenes with sizes about 1MB per source, including an orthogonal directional representation borrowed from [Zhang et al. 2018]. Run-time rendering decompresses and interpolates the ELD at the dynamic listener location, and then performs granular synthesis to superimpose many fragments of sound, or *grains*, with statistics governed by the decoded ELD. The grains may be generated procedurally or extracted from recordings. Our results and accompanying video show that extracting a field of ELDs driven by sound propagation in a scene yields a natural-sounding perceived texture. Overall, we propose the first practical system to capture real-time spatial variations in acoustic texture for extended sources in complex scenes.

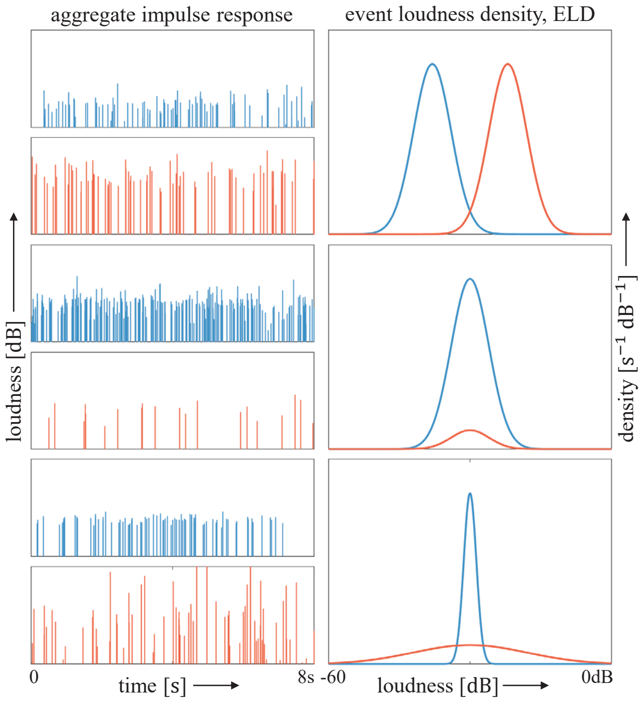

Fig. 2. Three examples relating the aggregate impulse response of arrival events, left, and the corresponding event loudness density (ELD), right. Refer to the accompanying video to hear the corresponding audio renderings. Shifting the distribution horizontally, along the loudness axis, corresponds to making all arrivals louder or quieter (top). Scaling up the ELD without changing its shape corresponds to adding more events drawn from the same loudness distribution (middle). Making the ELD broader, with a fixed area below the curve, changes the arrivals from a train of similar loudnesses to a train with large variations in loudness (bottom).

2 PRIOR WORK

Geometric acoustics. GA methods are surveyed in [Savioja and Svensson 2015], amounting to a small-wavelength approximation to the wave equation able to render spatial audio effects at interactive rates and support dynamic geometry. Neglecting diffraction, bi-directional path tracing has shown promise [Cao et al. 2016]. To account for the prominent role of diffraction in audible-wavelength sound propagation, stochastic scattering can be performed at or near geometric edges where sound paths bend. Techniques are being investigated [Schissler et al. 2014] but the general problem remains open [Savioja and Svensson 2015]. Typical extended sources we consider comprise thousands of source events; it becomes very costly to obtain converged estimates that find all paths connecting each source event to the listener.

Wave simulation. Solving the wave equation directly captures diffraction [Hamilton and Bilbao 2017] and explores all these paths systematically but implicitly, but at a computational cost that prohibits interactive movement of sources and listener. Previous systems [Raghuvanshi and Snyder 2014; Raghuvanshi et al. 2010] precompute the simulation and extract compact perceptual information, enabling real-time rendering of propagation effects.

These wave-based techniques consider a single point source; our interest lies in large ambient sources comprising thousands of independent source events. Although we use a band-limited pulse emitted from a point source like Raghuvanshi and Snyder [2014], we introduce them over many points on the source within the same simulation, propagating a massive superposition of their resulting radiated and scattered wavefronts within the scene. Our parameterization of the aggregate acoustic response also differs. Raghuvanshi and Snyder [2014] extract loudness for the direct sound, early reflections, and late reverberation transient phases of the impulse response for rendering perceptually important acoustic effects on sound emitted from a single coherent point source. Our work focuses on capturing the variable statistical texture of ambient sounds and their modification by sound propagation.

To model liquid sounds, there has also been work on near-field wave-based modeling of generation and propagation from sources such as water bubbles in a container, although for offline rendering [Langlois et al. 2016]. We target large-scale propagation effects from big stochastic sources within complex scenes, not limited to liquids.

Incoherent ambient sounds. Zhang et al. [2018] distributes an ideally incoherent signal over the spatio-temporal extent of the ambient source in a precomputed wave simulation. It then extracts and reconstructs two salient parameters as a function of 3D listener position: aggregate loudness and directionality. These parameters modulate a fixed input sound clip representing ocean waves, rain, or other ambient sources in the far field. Its assumption of ideal, steady, temporal incoherence is satisfied only when the listener is extremely far from the source, and thus neglects the changing texture of what is heard as the listener gets closer. Our representation includes directional effects using the same representation (low-order spherical harmonics), but also captures the salient acoustic texture variation.

Perceptual statistics of sound texture. Research in sound perception [McDermott et al. 2009; McDermott and Simoncelli 2011] has developed statistical descriptions that can be extracted from recorded sounds to summarize the perceived sound texture. These methods focus on stochastic temporal fluctuations in acoustic energy across different audio frequency bands, and they can be used to transform Gaussian random noise into sounds matching the original. We are interested in a related but different problem: characterizing and rendering the spatially-varying *modification* on sound texture due to propagation within a complex scene. Our representation is independent of the particular sounds emitted by the source. This modification is what we term acoustic texture, captured in the ELD. Once it has been extracted for a source volume and scene, we let the user specify any grain sounds, synthesizing the resulting radiated sound and applying the acoustic texture modification in real-time.

Granular sound synthesis. Granular synthesis is a broad term encompassing many techniques in audio processing and synthesis that break a sound into short units called *grains* and then reassemble them by concatenation or blending to modify (e.g., pitch shift) the original sound or to synthesize new sounds [Gabor 1946; Roads 2004; Verron et al. 2009]. Specialized for the sound of rain, [Miklavcic et al. 2004; Zita 2003] physically model the impact sounds of drops as grains, clustering them near the listener in free space. Liu et al. [2019] further propose the *material sound texture* representing rain drop sounds over various materials. Our acoustic texture instead represents propagation effects of an extended homogeneous source.

Our run-time employs granular synthesis, where grains are individual source event sounds which are generated and mixed stochastically to render the output. Our focus is to inform this process with the virtual environment.

3 PRECOMPUTED SOUND TRANSPORT

The first phase of our system is a precomputation that depends only on the scene and the source volume, in which we simulate sound wave propagation. The results of this simulation will be processed to derive the event loudness density (ELD) at listener positions throughout the scene for later use in run-time rendering.

Our precomputation uses the finite difference time domain (FDTD) method [Taflove and Hagness 2005] to propagate sounds, which numerically solves the wave equation

$$\frac{1}{c^2}\frac{\partial^2 p}{\partial t^2}(x,t) - \nabla^2 p(x,t) = s(x,t) \tag{1}$$

on a discrete grid in space-time, where x is 3D spatial location, t is time, $c = 340$m/s is the speed of sound, and p is acoustic pressure. The source term $s(x,\ t)$ models the ambient source as an aggregation of events and will be explained later. The simulation duration is denoted $T = T_S + T_D$, where T_S is the length of time in which the source continues to emit event pulses and T_D is the time required for a wave to propagate across the diagonal of the simulation domain. The fixed value $T_S = 2$s builds reliable and spatially-smooth statistics for the ELD in all our experiments. Perfectly-matched layers [Berenger 1994] are used to avoid reflection at the domain boundaries.

Care must be taken to keep the simulation stable and avoid numerical dispersion and dissipation errors; see [Zhang et al. 2018] for a discussion. Our experiments use a voxel size of $\Delta x = 0.25$m and a time step of $\Delta t = 0.25$ms, which implies a Nyquist frequency of 2000Hz.

Source event signal. The source event pulse introduced into the wave solver should follow two rules of thumb to minimize numerical error. First, its power spectral density should be bandlimited in the frequency domain, vanishing near DC (0Hz), to avoid residual particle velocity, and before the simulation Nyquist, to reduce dispersion errors that cause ringing in FDTD. Second, it should be compact in the time domain, minimizing overlap among neighboring event signals so that arriving events can be distinguished.

The Gaussian derivative is a compact, zero-mean signal. We tune its single parameter to avoid extensive energy approaching the simulation Nyquist via

$$s_0(t) = -\frac{t\sqrt{e}}{\sigma}e^{-\frac{t^2}{2\sigma^2}} \tag{2}$$

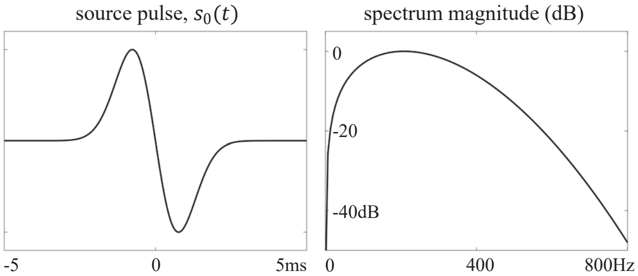

Fig. 3. Band-limited source event pulse used by the solver, $s_0(t)$. Left: Source pulse in time-domain. Right: energy spectral density.

where $\sigma = 3\,\Delta t$ and the maximum amplitude of $s_0(t)$ is 1. We also define the width of the pulse t_s such that $|s_0(t_s/2)| = 0.01$, yielding $t_s = 5.4$ms. Figure 3 shows the signal and its power spectral density.

Source event placement. We then introduce this source pulse identically at random onset times over the source duration T_S and random 3D locations over the source. Since the arrival event density at the listener integrates over the source, we fix the temporal event density across the entire volume of the source as $d_0 = 0.1/t_s$. A simple tradeoff governs our choice of this density. Higher density allows a shorter, less expensive simulation but makes it more likely arrivals will be incorrectly merged. Lower density avoids mergers but requires a longer and more expensive simulation to collect sufficient statistics. While $T_S = 2$s works well in our tests, for larger sources it can be increased to ensure ELD convergence. This is readily ascertained by analyzing the smoothness of spatial variation in ELD statistics, as shown in Figure 6.

4 ELD EXTRACTION

Wave simulation produces a time-varying pressure response $p(t;x)$ at each listener position. This data is processed as the simulation runs to accumulate a measurement of the ELD at each x. Doing this robustly and efficiently is challenging and must contend with numerical error. Dispersion errors in a bandlimited FDTD simulation cause "ringing", creating many lagging and attenuated copies of a single arrival event. Standard frequency-domain deconvolution exacerbates ringing and impractically requires storing the entire response in memory. We seek a streaming method which manages event aliasing and avoids assembling the entire response before extraction.

4.1 Deconvolution

Deconvolving the pressure response $p(t;x)$ with respect to the source pulse $s_0(t)$ recovers the aggregate impulse response $h(t;x)$ by inverting

$$p(t;\,x) \;=\; h(t;\,x) \,*\, s_0(t). \tag{3}$$

Inspired by compressive sensing [Candes et al. 2006; Donoho et al. 2006], we propose a sparsity-regularized time-domain deconvolution which reduces sensitivity to ringing. We interpret (3) as a sparse superposition of time-shifted copies of $s_0(t)$, converting deconvolution into an L_2 minimization problem with L_1 regularization (LASSO [Tibshirani 1996]):

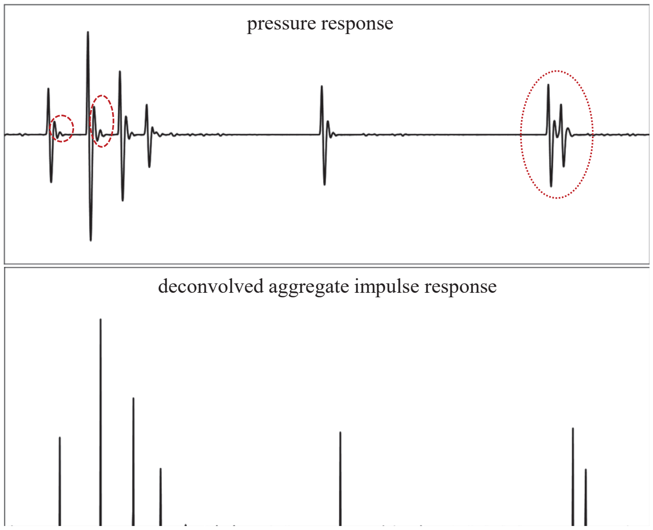

Fig. 4. L_1-regularized, least-squares deconvolution. The top image shows the received pressure signal; the bottom shows the deconvolved result. Our method is designed to cope with the ringing resulting from numerical dispersion in order to distinguish arrival events. Parts of the pressure response marked in dashed red at left show ringing "copies" of the original Gaussian derivative pulse injected into the simulation. Our method does not confuse these for additional arrivals. Two pulses overlap in the area marked in dotted red at right; our method correctly resolves two separate arrival events.

$$\arg\min_{h}\; \frac{1}{2}\big\|A\,h - p\big\|_2^2 \;+\; \lambda\,\|h\|_1 \tag{4}$$

where each column of the square matrix A is a time-shifted version of the elementary pulse $s_0(t)$ and λ is a regularization parameter.

Applying this idea directly on the entire space-time pressure response is expensive. We instead extract $h(t;x)$ in non-overlapping segments of length $T_0 = 10\Delta t$ and accumulate ELD statistics from each. Since the input pulse has a finite duration t_s, one must consider an input window with $t_s/2$ extra duration on either side to ensure that any output peak is fully captured within the analyzed segment. Thus, $p(t)$ is analyzed using overlapping input segments of duration $t_s + T_0$ that increment by T_0. In each segment, the matrix A is $n \times n$ where $n = (t_s + T_0)/\Delta t$; the first column contains the shifted pulse $s_0(t + t_s/2)$ and the last column contains $s_0(t - T_0 - t_s/2)$.

We then apply the alternating direction method of multipliers (ADMM) to solve (4) in each segment [Boyd et al. 2011]. We regularize using $\lambda = 0.1\,||A^T b||_\infty$; a standard choice that balances sparsity and convergence rate. The result is an estimate of $h(t;\,x)$ of length n. We discard the overlapping portions of the time segment from this output, using only the middle portion (of width T_0) for ELD accumulation.

Figure 4 shows the output of this method on input representing our wave simulator's pressure response when emitting a few pulses into the free field (i.e. without scene geometry). While the input is degraded by numerical dispersion, deconvolution still recovers narrow spikes.

4.2 Accumulation and encoding

At each simulation voxel, deconvolution over each time segment yields a corresponding segment of the aggregate impulse response from which it is straightforward to accumulate the ELD. We apply peak detection (using a simple relative min/max detector on three values adjacent in time), extract the peak's amplitude A, compute its loudness via $L = 10 \log_{10} A^2$, and accumulate peak loudnesses into a running histogram. Because max loudness is unknown when encoding begins, we accumulate into a histogram of conservatively large span: $[-60, 60]$dB with bin width of 3dB. The overall computation allows streaming, requiring only the time-varying pressure response over one time segment and the accumulated histogram (40 bins) as its stored state. We note that the just-noticeable difference for human perception of loudness is 1dB under ideal conditions; 3dB provides a good balance between quality and memory use.

Once simulation completes, the histogram is converted to the encoded ELD as follows. We first extract the maximum loudness received, quantized to 3dB, as a separate channel denoted L_m. We then store temporal densities at the next 12 loudness bins of width 3dB descending from L_m. This yields an encoding range of $3 \times 12 = 36$ for a minimum loudness of 36dB below the maximum which we have found sufficient in our experiments. The range can be expanded to include more of the softer end of the distribution at the expense of additional storage.

Event density in each bin is divided by the total temporal density over the whole simulated source, which we denoted d_0 in Sec. 3. Encoding relative rather than absolute density factors out dependence on the arbitrary density of the simulation source and allows run-time substitution of a source of different density via simple scaling. The relative densities are quantized in the range $[0, 20]$ with a quantum of $1/3$.

We spatially down-sample listener positions from what is simulated by a factor of 4 in each dimension. Overall the process produces 13 parameter fields containing, for each listener location, L_m and relative event densities in 12 ELD loudness bins offset from it. As with previous work [Raghuvanshi and Snyder 2014; Zhang et al. 2018], the parameter fields are spatially smooth and can be compressed. We compress the raw data by applying lossless LZW coding to the running difference along x scanlines.

4.3 Spatialization

In addition to the above ELD information, we also extract and encode an overall directional distribution of energy at each listener position x as the simulation runs. Our method follows [Zhang et al. 2018] but we summarize here. As in [Raghuvanshi and Snyder 2018] (Eq. 11, here corrected), we compute the directional acoustic power flux using the formula

$$f(x,t) = -p(x,t)\, v(x,t) = p(x,t) \int_{-\infty}^{t} \nabla p(x,\tau)\, d\tau. \tag{5}$$

We then aggregate directional energy in terms of spherical harmonics (SH). At each time step t, the acoustic power flux direction $\hat{f}(x,t) = f(x,t) / \|f(x,t)\|$ is projected onto a low-order (4 in our experiments) SH basis and the SH coefficient vector accumulated via

$$E_{l,m} = \frac{1}{T} \int_0^T p^2(t)\, Y_{l,m}(\hat{f}(t))\, dt \tag{6}$$

where Y_{lm} are the (real-valued) SH basis functions. At run-time, the mono sound signal is synthesized as discussed in the next section, and then spatialized per this spherical energy distribution (while ignoring inter-aural phase) to produce an output binaural signal.

5 RUN-TIME RENDERING

Our run-time rendering uses granular synthesis [Roads 2004], which controls meso-scale sound texture by mixing micro-scale acoustical grains, so as to synthesize the sound of an extended source. We assume a collection of brief grain sounds such as rain drops or bird tweets; how we generate them is discussed later. Our goal is to simulate the sound due to a source that emits these grains randomly, and our approach is to produce an acoustic texture by mixing many random grains drawn according to the ELD at the listener's position.

5.1 Grain blending

We first decompress and spatially interpolate at the current listener location x, to obtain the ELD $E(L; x)$ where L is loudness. We henceforth drop the listener position, x. Recall that $E(L)$ relates the relative temporal event density to loudness. We employ a straightforward method that is cache-friendly and obtains real-time performance. It scales and adds grain signals to the output audio buffer such that their loudness and temporal density statistics respect $E(L)$.

Because grains can be longer than the audio buffer size (in our case 1024 audio samples), our method queues active grains. For each, it stores the grain onset time, the grain index in its collection, and the grain amplitude, so that its signal can be synthesizing coherently across multiple output audio buffers. Two simple algorithms drive synthesis, detailed in Figure 5. We define overall grain density (i.e., grains generated per second) as $D = d_0' \int E(L)\, dL$. The integral yields the total number of events per second at the listener relative to the source's event density. This is multiplied with d_0', the absolute event density of the source, which is specified by the sound designer based on physical or artistic concerns and can be modified in real-time if desired. This yields D, the expected number of grains we will superpose per second. Grain density D changes as the listener moves, and is updated at the start of every audio buffer.

To stochastically draw the grain amplitude requires a simple procedure we perform on the fly. The computation is minimal because $E(L)$ is represented with just 12 loudness bins. We first compute $E(L)/D$ to form a probability density function (PDF) providing the probability for each loudness bin. We then integrate to form the cumulative distribution function (CDF). Generating a uniform random number q in $[0, 1]$, we compute the loudness corresponding to q by inverting the CDF. Finally, we relate grain amplitude to loudness via $A = 10^{L/20}$.

Per-buffer loudness adjustment. This method works well for impulsive but not long-lasting grains, such as the human utterances in BabblingCrowd. Long grains can potentially span large changes in loudness as the listener moves through the scene, e.g. if the listener walks away from the source into an occluded room. To provide better temporal continuity, we make a simple adjustment that scales

```
R: audio sample rate [44100Hz]
rand(): uniform random number generator in [0,1]

Runtime synthesis

for each output audio buffer
Call Add new grains
for each active grain
    synthesize signal remainder and add to buffer, scaled by $A$
    if grain now complete, dequeue

Add new grains

for each time sample $t$ in buffer
    if rand() $< D/R$,    // decide to generate grain
        generate random index $i$ into grain collection
        draw grain amplitude $A$ from ELD $E$
        enqueue grain at $(t, i, A)$
```

Fig. 5. Granular rendering of the event loudness density (ELD).

loudness in each current audio buffer to vary with the grain's current ELD max. Specifically, suppose a grain was initially generated with loudness L when its ELD max was L_m^{orig}. Later its ELD max becomes L_m^{curr}. We then render this grain in the current audio buffer with loudness $L - L_m^{\text{orig}} + L_m^{\text{curr}}$. No cross-fading was found necessary, although it might be needed for very fast listener motion.

Atmospheric attenuation. Without atmospheric attenuation, high frequencies can sound unnaturally harsh when standing far from the sound source. Our solver lacks such modeling, but we find a simple approximation sufficient in practice. We employ standard analytical formulae relating propagation distance to atmospheric attenuation [ISO 9613-1 1993]. We conservatively estimate distance to the source as $r \approx 10^{(L_0 - L)/20}$, where L_0 is the empirically determined loudness of the source at 1m distance, and L is the ELD's average loudness across all arriving events at the listener location. Our formula assumes attenuation from a point source; since the loudness falls off more gradually for an extended source, we under-estimate the distance. We implement the computed frequency-dependent attenuation using a 4-band equalization filterbank with center frequencies of {125,600,2400,9600}Hz.

5.2 Grain collection creation

Our rendering method requires a collection of grains as input; they should be spectrally and semantically similar and short. Most of our grains persist for just a few tens of milliseconds, but some (e.g. human utterances) are longer, up to a few seconds. Each time a grain is added in the algorithm from Fig. 5, we randomly select its index i from the collection. Our examples use several different approaches to generate these grain collections.

For water sounds (both drop impacts and flows) we applied a procedural model based on bubble oscillations [Doel 2005]. Although this model is fast enough to perform procedural synthesis in real time, we simplify the implementation by pre-generating a random collection of 1000 grains, thereafter treating them as recordings.

For the crowd and bird-flock grain collections, we manually segmented recorded clips into individual events. We obtained 45 tweet grains from a single recorded clip of starling calls, and 2000 grains (English utterances) from a database of recordings [Shtooka 2010]. For the lapping waves, we segmented a recording applying a Kaiser-Bessel sliding window [Bosi and Goldberg 2012], with 2 second width, 1 second sliding step, and $\beta = 10$, yielding a collection of 19 grains.

6 RESULTS

We precompute a single FDTD simulation and encode the ELD and SH directional parameter fields for each scene per sound source, which takes 1–16 hours on a desktop computer with Intel i7-8700K CPU @ 3.70GHz with 6 cores and 32G RAM. In all our test scenes, the compressed data is about 1MB per sound source; further information is reported in Table 1. Our run-time implementation is integrated in Unreal Engine 4.

Results for ambient soundscapes in four scenes are included in the supplementary video. RiverHouse includes a linear stream of water with nearby building geometry. RainOverPool models the sound of light rain hitting a rectangular water pool in a house and garden scene. StylizedKingdom demonstrates outdoor acoustic texture effects from a flock of starlings in a tree canopy and water lapping around a lake. BabblingCrowd includes a single extended source in a small room representing a babbling crowd.

In most scenes, run-time grain density, d_0', varies from 18 to 90 s^{-1}. The exception is procedural water sounds, where we use a higher density (around 1800 s^{-1} for rainfall and 9000 s^{-1} for stream flow). Recall that arrival density at the listener is further modified, compared to the source density, by the ELD. As the listener moves, we plot the ELD corresponding to that location, annotated with green bars for the 50th (median) and 95th percentile loudnesses across all arriving events, and the average loudness. The video also demonstrates the enhanced realism provided by rendering the spatially-varying acoustic texture using the ELD, compared to a fixed texture (labeled "loudness variation only" in the video) as in [Zhang et al. 2018].

Our approach first produces a mono signal blending across all grains, and then spatializes the blended result using an aggregate directional representation collected over the entire source. Ideally, each grain would be spatialized separately since each exhibits a separate arrival direction. Our simple model compactly encodes the main effects, becoming less convincing with a collection of directional and highly recognizable/distinguishable grains, as with a listener standing in the midst of a babbling crowd.

RiverHouse. Near the stream source, crisp gurgling sounds are audible. The acoustic texture stays nearly constant as the listener walks along its bank. As the listener moves away from the stream, these distinct gurglings gradually disappear and the texture smoothly changes to become more noise-like. Inside the house, reverberation causes a Gaussian ELD shape, essentially capturing the room's decay time characteristic. Behind the house outdoors, edge diffraction rather than reverberation serves to mix events arriving around the sides of the building from across the source, resulting in a unimodal

Table 1. Precomputation data.

scene (source)	scene (m)	scene voxels	source box (m)	source voxels	bake time (h)	bake RAM (GB)	run-time RAM (MB, encoded)
RiverHouse	45 × 40 × 5.0	0.90×10^6	7.0 × 40 × 0.50	8.0×10^3	1.0	1.6	0.40
RainOverPool	35 × 55 × 3.0	0.70×10^6	8.0 × 16 × 0.50	4.0×10^3	10	1.2	0.80
StylizedKingdom (starlings)	53 × 32 × 8.0	0.80×10^6	10 × 4.0 × 1.0	2.5×10^3	3.3	2.2	0.50
StylizedKingdom (lake)	60 × 60 × 8.0	2.0×10^6	35 × 45 × 1.0	17×10^3	16	3.7	1.2
BabblingCrowd	40 × 30 × 14	1.4×10^6	18 × 10 × 1.0	10×10^3	4.2	2.7	0.50

but non-Gaussian ELD without loud outliers and conveying a distant source. Outdoors but between two walls, the result sounds like the expected fusion of indoors and outdoors.

RainOverPool. Individual rain drops are audible close to the pool, becoming gradually indistinct farther away, and suddenly more noise-like and denser as the listener enters the house.

StylizedKingdom. The flock of starlings tweeting spatializes well overhead. As the listener moves away from it, the individual calls merge into a more uniform, high pitched texture. The scene also demonstrates our system's ability to render multiple sources.

BabblingCrowd. A babbling crowd gathers in a small room in this scene. The listener begins in a large indoor courtyard outside the room's open door. Acoustic texture varies convincingly as the listener moves from near to far or side to side across the opening. Inside the room in the midst of the crowd, we obtain a plausible result even though grain utterances fail to merge into recognizable conversation.

7 CONCLUSION

Our system models ambient sounds by randomly distributing identical pulses over the source's space-time extent and propagating these through a synthetic scene via a single wave simulation. To capture the nuanced spatial variation in the perceived sound texture, we focus on how propagation changes not only overall loudness but also the relative arrival density of events at different loudnesses. We formulate the *event loudness density*, statistically relating the temporal density of event arrivals as a function of their loudness, and show how the ELD can be extracted from simulation via careful deconvolution, and used to govern run-time synthesis with free listener movement. The encoded parameters are spatially smooth and compress well, letting our system obtain convincing variation of acoustic texture in complex scenes while maintaining a small run-time memory budget.

Unlike geometric (ray tracing) techniques, our wave propagation approach accounts for diffraction so that sound is heard around obstacles and through portals even when the source isn't directly visible (e.g., behind the RiverHouse). However, diffraction produces low-pass filtering effects on each event, which our deconvolution method does not currently recover. Rather than analyzing arrival events separately, this cue might be inexpensively restored by applying an equalization filter driven by aggregate frequency content over all received pulses in the response.

Reverberation affects ambient sources, but quite differently from how it affects point sources. Since ambient sounds are sustained in time and comprise many similar and overlapping events, we expect that reverberation primarily serves to increase the temporal event density, captured in the ELD. This effect is audible in the demos, e.g. as the listener walks from outdoors into the RiverHouse. But reverberation also exhibits a decaying "tail" (RT60) effect that can't be realized just by changing overall density and so is ignored by our approach. The effect is more saliently missed as event arrivals get very sparse, so that the gaps between them are long enough to hear the reverb falloff from a single event.

Our simple spatialization assumes that phases of sound arriving from different directions are decorrelated due to mixing of many micro-events at the listener, making interaural phase difference less significant than the head shadowing (interaural level difference) that we do model. As with reverberation, this shortcut becomes less convincing as events get temporally sparse at the listener, so that the interaural phase cue of a single event (arriving in a single direction) becomes more audible. When event arrivals become sufficiently sparse, our method might be extended by rendering the loudest individual events as separate coherent point sources.

Our system currently applies spatialization on the aggregate directionality of energy over all event arrivals. In fact, acoustic texture varies with direction. Standing between an extended source and the door to a room which functions as a sound reservoir, the ELD is broader (more near-field sounding) in the ear facing the source and more peaked (far-field sounding) in the ear facing the door through which room reverberation gets mixed.

Finally, a complete solution would offer some way to capture near-field effects where our model breaks down and the arrival time and energy of successive events become audibly correlated. As shown in our crowd babble scene, while a reasonable rendering is obtained as long as utterances remain unintelligible, the full cocktail party effect, coherently preserving both the directional and semantic content of sounds made by each nearby speaker, remains for future work.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive feedback. The recording of Sturnus vulgaris (European starling) is by Wilbur

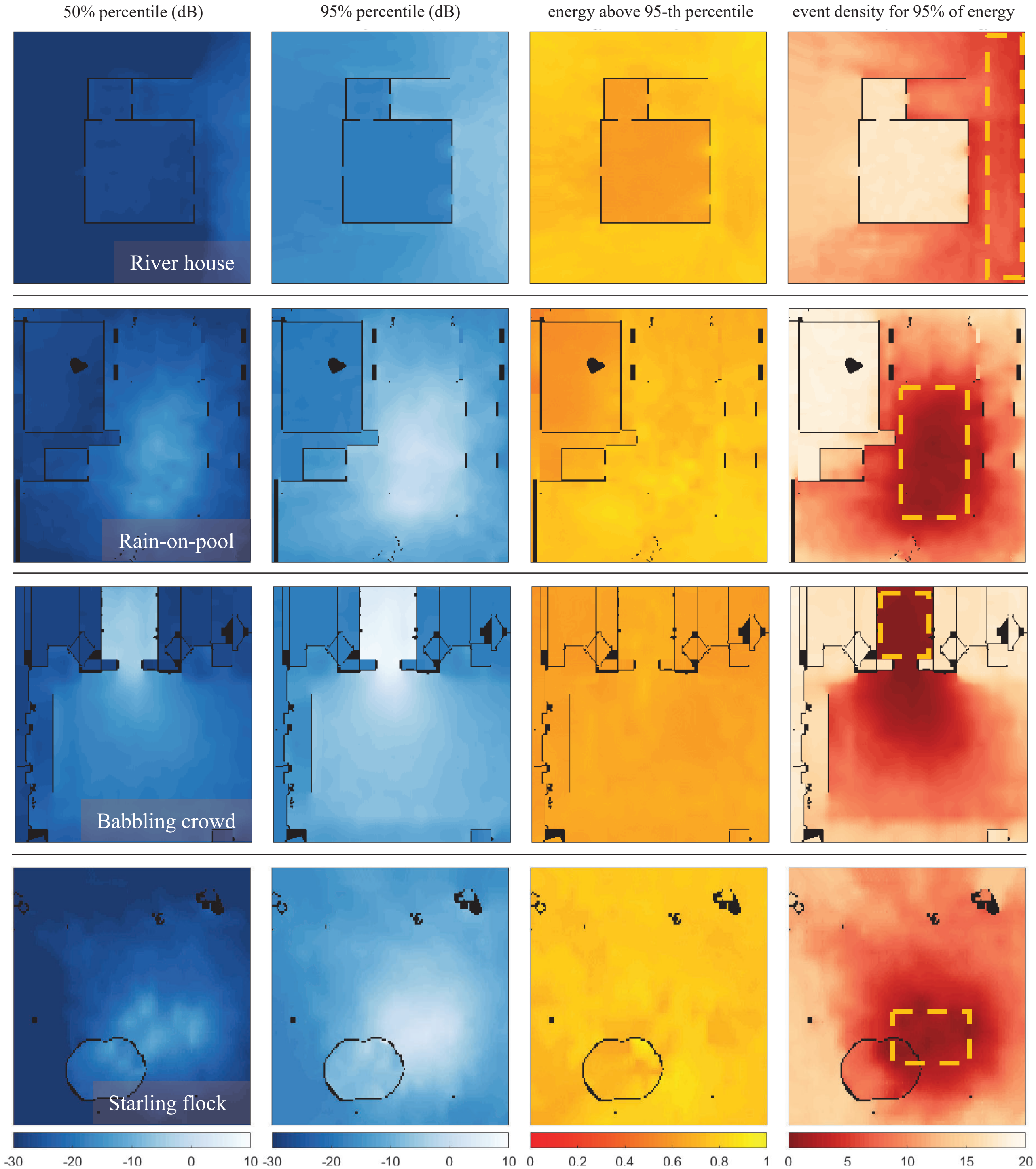

Fig. 6. Spatial maps of ELD characteristics. We show results for three scenes each with a single ambient source, indicated by the dashed yellow area in the fourth column. The leftmost two columns show median (50th percentile) and 95th percentile loudness over all arriving events, respectively. These indicate the smoothness of our parameterization. The third column plots the fraction of energy remaining above the 95th percentile: brighter areas indicate more loud arrivals that stand out over the rest. Note how this property diminishes inside the river house, conveying the increased mixing of grains that happens indoors due to reverberation. The fourth column maps summed arrival density over the biggest loudnesses containing 95% of total energy. Note how it increases in enclosed spaces from reverberation but decreases in the vicinity of the source.

Hershberger, courtesy of the Macaulay Library at the Cornell Lab of Ornithology (ML107248). We are grateful to Mandy Xia for her help with the supplementary video recording.

REFERENCES

Jean-Pierre Berenger. 1994. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics* 114, 2 (1994), 185–200.

Marina Bosi and Richard E Goldberg. 2012. *Introduction to digital audio coding and standards*. Vol. 721. Springer Science & Business Media.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3, 1 (2011), 1–122.

Emmanuel J Candes, Justin K Romberg, and Terence Tao. 2006. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 59, 8 (2006), 1207–1223.

Chunxiao Cao, Zhong Ren, Carl Schissler, Dinesh Manocha, and Kun Zhou. 2016. Interactive sound propagation with bidirectional path tracing. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 180.

Kees van den Doel. 2005. Physically based models for liquid sounds. *ACM Transactions on Applied Perception (TAP)* 2, 4 (2005), 534–546.

David L Donoho et al. 2006. Compressed sensing. *IEEE Transactions on information theory* 52, 4 (2006), 1289–1306.

Dennis Gabor. 1946. Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering* 93, 26 (1946), 429–441.

Brian Hamilton and Stefan Bilbao. 2017. FDTD Methods for 3-D Room Acoustics Simulation With High-Order Accuracy in Space and Time. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 25, 11 (Nov. 2017), 2112–2124. https://doi.org/10.1109/TASLP.2017.2744799

ISO 9613-1 1993. *Acoustics – Attenuation of sound during propagation outdoors – Part 1: Calculation of the absorption of sound by the atmosphere.* Standard. International Organization for Standardization, Geneva, CH.

Timothy R. Langlois, Changxi Zheng, and Doug L. James. 2016. Toward Animating Water with Complex Acoustic Bubbles. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2016)* 35, 4 (July 2016). https://doi.org/10.1145/2897824.2925904

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

J. H. McDermott, A. J. Oxenham, and E. P. Simoncelli. 2009. Sound texture synthesis via filter statistics. In *2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. 297–300. https://doi.org/10.1109/ASPAA.2009.5346467

Josh H. McDermott and Eero P. Simoncelli. 2011. Sound Texture Perception via Statistics of the Auditory Periphery: Evidence from Sound Synthesis. *Neuron* 71, 5 (2011), 926 – 940. https://doi.org/10.1016/j.neuron.2011.06.032

Stanley J Miklavcic, Andreas Zita, and Per Arvidsson. 2004. *Computational real-time sound synthesis of rain.* Department of Science and Technology (ITN), Campus Norrköping, Linköping.

Nikunj Raghuvanshi and John Snyder. 2014. Parametric Wave Field Coding for Precomputed Sound Propagation. *ACM Trans. Graph.* 33, 4, Article 38 (July 2014), 11 pages. https://doi.org/10.1145/2601097.2601184

Nikunj Raghuvanshi and John Snyder. 2018. Parametric Directional Coding for Precomputed Sound Propagation. *ACM Trans. Graph.* 37, 4, Article 108 (July 2018), 14 pages. https://doi.org/10.1145/3197517.3201339

Nikunj Raghuvanshi, John Snyder, Ravish Mehra, Ming C. Lin, and Naga K. Govindaraju. 2010. Precomputed Wave Simulation for Real-Time Sound Propagation of Dynamic Sources in Complex Scenes. *ACM Transactions on Graphics (proceedings of SIGGRAPH 2010)* 29, 3 (July 2010).

Curtis Roads. 2004. *Microsound.* MIT press.

Lauri Savioja and U Peter Svensson. 2015. Overview of geometrical room acoustic modeling techniques. *The Journal of the Acoustical Society of America* 138, 2 (2015), 708–730.

Carl Schissler, Ravish Mehra, and Dinesh Manocha. 2014. High-order diffraction and diffuse reflections for interactive sound propagation in large environments. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 39.

Shtooka. 2010. Project Shtooka. http://shtooka.net/ Accessed: 2019-05-18.

Allen Taflove and Susan C Hagness. 2005. *Computational electrodynamics: the finite-difference time-domain method.* Artech house.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), 267–288.

Charles Verron, Mitsuko Aramaki, Richard Kronland-Martinet, and Grégory Pallone. 2009. A 3-D immersive synthesizer for environmental sounds. *IEEE Transactions on Audio, Speech, and Language Processing* 18, 6 (2009), 1550–1561.

Zechen Zhang, Nikunj Raghuvanshi, John Snyder, and Steve Marschner. 2018. Ambient Sound Propagation. *ACM Trans. Graph.* 6 (11 2018). https://doi.org/10.1145/3272127.3275100

Andreas Zita. 2003. *Computational Real-Time Sound Synthesis of Rain.* Linköping University, Department of Science and Technology. 40 pages.