Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2012)
P. Kry and J. Lee (Editors)

Faster Acceleration Noise for Multibody Animations using Precomputed Soundbanks

Jeffrey N. Chadwick, Changxi Zheng, and Doug L. James

Cornell University

Abstract

We introduce an efficient method for synthesizing rigid-body acceleration noise for complex multibody scenes. Existing acceleration noise synthesis methods for animation require object-specific precomputation, which is prohibitively expensive for scenes involving rigid-body fracture or other sources of small, procedurally generated debris. We avoid precomputation by introducing a proxy-based method for acceleration noise synthesis in which precomputed acceleration noise data is only generated for a small set of ellipsoidal proxies and stored in a ***proxy soundbank****. Our proxy model is shown to be effective at approximating acceleration noise from scenes with lots of small debris (e.g., pieces produced by rigid-body fracture). This approach is not suitable for synthesizing acceleration noise from larger objects with complicated non-convex geometry; however, it has been shown in previous work that acceleration noise from objects such as these tends to be largely masked by modal vibration sound. We manage the cost of our proxy soundbank with a new wavelet-based compression scheme for acceleration noise and use our model to significantly improve sound synthesis results for several multibody animations.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation; H.5.5 [Information Systems]: Information Interfaces and Presentation—Sound and Music Computing

1. Introduction

Simulations of multibody dynamics for complex scenes – tumbling piles of rocks, shattering panes of glass, etc. – can be used to produce compelling animations with rich visual behavior. These simulations should also provide an equally rich source of sound. Unfortunately, most approaches for synthesizing rigid-body sound produce unconvincing results for these scenarios due to their inability to resolve certain physical phenomena. Rigid-body sound is most often computed with the linear modal sound algorithm [vd-DKP01, OSG02, BDT*08] in conjunction with some method for evaluating modal *acoustic transfer* functions (e.g., [JBP06, CAJ09]). This approach ignores aspects of an object's physical behavior which are potentially important for sound synthesis. Linear modal sound synthesis considers how an object vibrates, and how these vibrations produce sound in the surrounding medium. However, certain objects experience few, if any, physical vibrations at frequencies within the human hearing range (roughly 20-20000 Hz). For these objects, an important source of sound is *acceleration noise*; that is, acoustic perturbations resulting from an object experiencing rapid rigid-body acceleration. Acceleration noise typically takes the form of short, transient "clicks" immediately following collisions between pairs of objects. Sound synthesis pipelines based on only the modal sound algorithm tend to produce results in which impacts sound blurred or indistinct due to this method's failure to correctly resolve transient collision sounds. This omission is particularly noticeable when synthesizing sound from small objects (e.g., rigid debris generated from fracture simulations or some other procedural method).

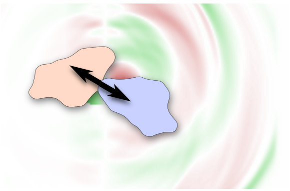

Figure 1: *When two objects collide, the resulting rigid-body accelerations produce pressure fluctuations which are interpreted as sound.*

Acceleration noise has been extensively studied outside of the computer animation community. Numerous theoretical and experimental studies have analyzed acceleration noise produced by simple objects (spheres, cylinders, plates, etc.) [RWJ79a, RWJ79b, ENNS81, YZ92, WGM94, CL01]. Others have studied acceleration noise numerically [SLC99, LCM01, RO07, MKP11]; however, the models employed here are impractical for animation sound synthesis. O'Brien et al. [OCE01] synthesize sound directly from surface motion computed using a physics-based solver but do not consider wave propagation effects which are necessary for convincing acceleration sound synthesis. Recently, Chadwick et al. [CZJ12] proposed a method for synthesizing acceleration noise from physically based rigid-body animations. In this work, the authors use Hertz contact theory [Her82, Joh85] to generate physics-based continuous rigid-body accelerations from each contact event in a simulation. Next, they introduce an object-centric representation for acceleration noise – Precomputed Acceleration Noise (PAN) – which enables efficient synthesis of acceleration noise due to arbitrary rigid-body accelerations.

The methods of [CZJ12] require extensive precomputation on a per-object basis. This approach guarantees physical accuracy and is not a severe bottleneck for scenes involving a few predetermined objects. However, it presents a challenge when attempting to synthesize sound from simulations with many unique objects. Moreover, the examples presented in [CZJ12] suggest that acceleration noise makes a large contribution relative to modal sound primarily in scenes with large ensembles of small objects. We also observe that the acceleration noise produced by such objects tends to be fairly simple when compared to sound produced by larger non-convex objects (bowls, mugs, etc.). As a result, it is important to develop simpler models for approximating sound from complex multibody scenes with many small objects (e.g., debris from fracture simulations). We address this with two main contributions:

1. *Proxy object soundbank*: We build PAN representations for a set of proxy ellipsoids and compute sound from arbitrary objects by fitting them to appropriate proxies based on their physical properties. By introducing a scaling relationship for the precomputed representation introduced in [CZJ12], we are able to limit the space of proxy ellipsoids to a few dozen objects.
2. *Memory-efficient PAN representation*: A common issue arising in both modal and acceleration sound synthesis is the large amount of storage required for precomputed sound data. To address this, we introduce a wavelet-based representation for precomputed acceleration noise. This enables acceleration sound synthesis which is more efficient in both time and space. Our PAN representation also requires a less costly preprocess than the representation used in [CZJ12]. We further reduce the size of our proxy soundbank by fully exploiting symmetries arising in our precomputed data set, allowing us to limit the total size of this data set to approximately 5-26MB, depending on the amount of compression used.

Other Related Work: In [ZJ10], the authors introduced a method for synthesizing modal sound from rigid-body fracture simulations. An important component in this method is an ellipsoidal proxy soundbank storing precomputed modal sound data. Chadwick et al. [CZJ12] showed that the addition of acceleration noise significantly improves the fracture sound results of [ZJ10] for certain scenes, but their results are limited to scenes with a small number of objects. In this work, we further improve upon these results by applying our proxy model to fracture simulations involving hundreds of pieces. Our work is similar in spirit to [ZJ10], but in this work we address the challenges of developing a proxy model for a fundamentally different sound phenomenon.

2. Background

The sound produced during rigid-body collisions is the result of two sources: "ringing noise" [RWJ79b] and "acceleration noise" [RWJ79a]. Ringing noise is produced when an object vibrates, whereas acceleration noise is produced when an object as a whole experiences rapid rigid-body acceleration. Existing rigid-body sound pipelines typically synthesize ringing noise with the linear modal sound model [vdDKP01, OSG02, BDT*08], which fails to resolve the latter phenomenon.

2.1. Acceleration Noise Physics

Following the notation of [CZJ12], we consider a rigid-body object O, its boundary $\partial\Omega$ and its exterior domain Ω. Acoustic pressure in Ω is governed by the wave equation

$$\frac{1}{c^2}\frac{\partial^2 p(\mathbf{x},t)}{\partial t^2} = \nabla^2 p(\mathbf{x},t), \quad \mathbf{x} \in \Omega, \tag{1}$$

subject to boundary conditions

$$\nabla p(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x}) = -\rho a_n(\mathbf{x},t), \tag{2}$$

where c and ρ refer to the speed of sound and density of air (343.2m/s and 1.2041kg/m^3 for air at standard temperature and pressure). $\mathbf{n}(\mathbf{x})$ and $a_n(\mathbf{x},t)$ denote the surface normal and time-varying normal acceleration at position $\mathbf{x}$ on the surface of O. It is clear from (2) that arbitrary surface accelerations – including those due to rigid-body motion – contribute to the sound produced by O.

Previous approaches for computing rigid-body sound have assumed that O's visual motion is governed by rigid-body dynamics, and that this motion is independent of any deformations experienced by O. Surface deformations in O are modeled by simulating dynamics in a small basis of linear modes. Under these assumptions, O's surface acceleration can be written as a sum of rigid-body acceleration and acceleration due to modal vibration: $a_n(\mathbf{x},t) = a_n^R(\mathbf{x},t) + a_n^M(\mathbf{x},t)$.

By the linearity of (1) and (2), the acoustic pressure field produced by O's motion can be similarly written as $p(\mathbf{x},t) = p^R(\mathbf{x},t) + p^M(\mathbf{x},t)$. Since most rigid-body sound pipelines only synthesize sound due to modal vibrations, the term $p^R(\mathbf{x},t)$ – acoustic pressure due to rigid-body acceleration – has traditionally been neglected.

2.2. Acceleration Noise Synthesis

In this section, we briefly review the acceleration noise synthesis methods introduced in [CZJ12].

2.2.1. Continuous Contact Force Estimation

A common technique for resolving collisions in rigid-body simulations is to apply repulsive *impulses* to colliding objects (e.g., [GBF03]). For the purpose of physics-based animation of rigid-body dynamics, these impulses produce instantaneous changes in the linear and angular momentum of colliding objects. While this method is sufficient for visual dynamics simulation, instantaneously updating linear and angular velocities does not provide a satisfactory continuous acceleration boundary condition of the form (2). In [CZJ12], the authors appeal to Hertz contact theory [Her82] to produce physically plausible continuous acceleration profiles from rigid-body impacts. The time-dependence of acceleration for each impact is given by a half-sine pulse [Joh85]:

$$S(t;t_0,\tau) = \begin{cases} \sin\left(\frac{\pi(t-t_0)}{\tau}\right) & \text{if } t_0 \le t \le t_0+\tau \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where t_0 is the time of impact between two objects. The collision time scale τ is estimated from the local contact geometry and material parameters of the colliding objects. See §3 in [CZJ12] for further details.

2.2.2. Precomputed Acceleration Noise

Let $\mathbf{a}(t) = [a_1(t)\ a_2(t)\ a_3(t)]^T$, $\boldsymbol{\alpha}(t) = [\alpha_1(t)\ \alpha_2(t)\ \alpha_3(t)]^T$ and $\mathbf{x}_0$ refer to the translational acceleration, angular acceleration and center of mass position of a rigid body O at time t in O's coordinate frame. If we assemble the translational and rotational accelerations in to a single vector $\mathbf{z}(t) = [a_1(t)\ a_2(t)\ a_3(t)\ \alpha_1(t)\ \alpha_2(t)\ \alpha_3(t)]$, then rigid surface acceleration a_n^R at point $\mathbf{x}$ on O's surface can be written

$$a_n^R(\mathbf{x},t) = \sum_{i=1}^{6} z_i(t) g_i(\mathbf{x}), \tag{4}$$

where g_i $(i = 1,\ldots,6)$ are functions depending only on O's geometry (see §4.1 in [CZJ12]). We can solve (1) independently for each of 6 surface accelerations of the form $a_n(\mathbf{x},t) = z_i(t)g_i(\mathbf{x})$ and reconstruct the total pressure by summing these solutions. Let $p_i(\mathbf{x},t)$ refer to the solution of (1) with $a_n(\mathbf{x},t) = z_i(t)g_i(\mathbf{x})$. In [CZJ12], the authors approximate $z_i(t)$ with interpolating basis functions $\psi(t;h)$:

$$z_i(t) \approx \sum_{k=0}^{\infty} z_i(kh)\psi(t-kh;h). \tag{5}$$

$\psi(t;h)$ is a Mitchell-Netravali cubic filter [MN88]

$$\psi(t;h) = \frac{1}{18}\begin{cases} -15y^3+18y^2+9y+2 & |t| \le h \\ 5(1+y)^3-3(1+y)^2 & h \le |t| \le 2h, \\ 0 & \text{otherwise}, \end{cases} \tag{6}$$

where $y \equiv 1-|t|$. The wave equation is solved with surface acceleration $a_n = \psi(t;h)g_i(\mathbf{x})$ for some $h > 0$, yielding solution $p_i^{(h)}(\mathbf{x},t)$. It then follows from (5) that the total pressure for the i^{th} rigid acceleration term can be approximated by

$$p_i(\mathbf{x},t) \approx \sum_{k=0}^{\infty} z_i(kh)p_i^{(h)}(\mathbf{x},t-kh) \tag{7}$$

That is, given the functions $p_i^{(h)}$, we can use (7) to approximate acceleration noise due to arbitrary rigid-body accelerations, up to a temporal resolution defined by h. The parameter h is chosen heuristically based on O's geometry and material parameters to ensure that (5) adequately interpolates rigid-body accelerations encountered by O during simulations (see §4.3 of [CZJ12]).

The pressure field $p_i^{(h)}$ can be found by, for example, timestepping (1) on a finite difference grid or finite element mesh (e.g., [MKP11]); however, this approach would prove too costly for animation sound synthesis. Instead, [CZJ12] introduced a representation for $p_i^{(h)}$ – Precomputed Acceleration Noise (PAN) – which allows for efficient evaluation of $p_i^{(h)}$ at arbitrary positions and times. For a listening position $\mathbf{x}$, let (R,θ,ϕ) be the spherical coordinates of $\mathbf{x}$ relative to object O's center of mass (θ and ϕ are the azimuthal and polar angles, respectively). For a fixed angular direction (θ,ϕ), $p_i^{(h)}$ is approximated by the following series (dropping subscripts and superscripts from $p_i^{(h)}$ for brevity):

$$p(R,\theta,\phi,t) \approx \sum_{k=1}^{N} \frac{1}{R^k} q_k\left(\theta,\phi,t-\frac{R}{c}\right). \tag{8}$$

N is the number of series terms, c is the speed of sound, and the functions q_k are chosen so that (8) accurately approximates $p(\mathbf{x},t)$ in the direction (θ,ϕ). In §4.1 we present a method for choosing the functions q_k that achieves significant compression over the representation used in [CZJ12].

3. Acceleration Noise Proxy Geometry

The sound synthesis methods discussed in §2.2 have been shown to significantly improve results for a variety of objects when used in conjunction with linear modal sound (coins, dice, keys, etc.). Unfortunately, computing the PAN fields discussed in §2.2.2 requires lengthy precomputation for each unique object in a scene. Specifically, the wave equation must be solved for short acceleration pulses in each rigid-body degree of freedom. While this approach is sufficient for scenes involving a reasonably small number of predetermined objects, it does not scale well to scenes with hundreds or thousands of unique objects. In [CZJ12] it is shown

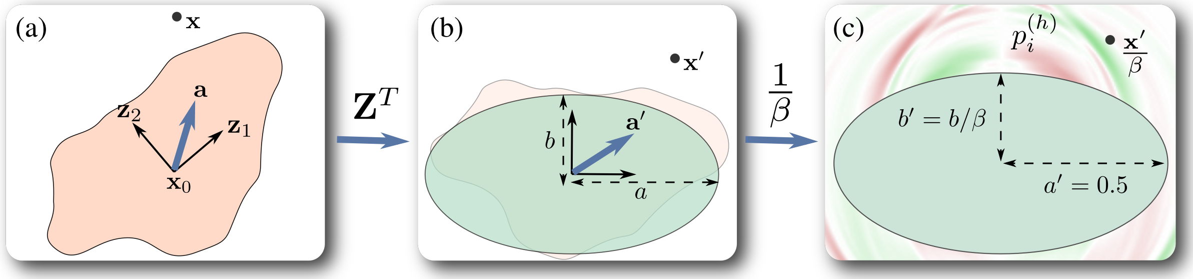

Figure 2: ***Proxy Acceleration Sound Synthesis:*** *We synthesize sound due to motion of object O at listening position $\mathbf{x}$.* (a) *An object O undergoes translational acceleration $\mathbf{a}$. O's center of mass $\mathbf{x}_0$ and principle axes of inertia $\mathbf{z}_1, \mathbf{z}_2$ are shown;* (b) *We fit an ellipsoidal proxy to O according to its principle moments of inertia, and transform $\mathbf{x} \to \mathbf{x}', \mathbf{a} \to \mathbf{a}'$ in to the axis-aligned proxy ellipsoid space;* (c) *We scale the proxy ellipsoid to match a reference ellipsoid with unit x-axis length and synthesize sound using this ellipsoid's PAN functions $p_i^{(h)}$ and the scaling relationships* (10) *(in this figure, we assume $0 < \beta < 1$).*

that acceleration noise can make a significant contribution to scenes with large quantities of small debris (e.g., fracture simulations). As such, it is important to develop scalable methods for synthesizing acceleration noise from multibody scenes. In this section, we introduce an ellipsoidal proxy model for acceleration noise. By storing PAN fields for a small set of proxy objects, we are able to efficiently synthesize plausible acceleration noise for scenes with thousands of unique objects.

3.1. Scaling Relationships

In this section we establish scaling relationships between the acceleration noise produced by objects O and O_β, where O_β is identical to O but has been uniformly scaled by $\beta > 0$. Following the notation of §2.2.2, the boundary conditions used to solve the wave equation for pressure fields $p_i^{(h)}$ ($i = 1, \ldots, 6$) due to object O are

$$\nabla p_i^{(h)} \cdot \mathbf{n}(\mathbf{x}) = \psi(t;h) \left\{ \begin{array}{cc} -\rho\, \mathbf{e}_i \cdot \mathbf{n}(\mathbf{x}) & i = 1,2,3 \\ -\rho\,(\mathbf{e}_{i-3} \times (\mathbf{x} - \mathbf{x}_0)) \cdot \mathbf{n}(\mathbf{x}) & i = 4,5,6 \end{array} \right. \tag{9}$$

where $\mathbf{e}_i \in \mathbb{R}^3$ is the vector with components $e_{ij} = \delta_{ij}$. Now, suppose that the exterior domains of O and O_β are Ω and Ω_β, respectively. Let P refer to pressure due to the scaled object O_β. The following scaling relationships hold:

$$P_i^{(\beta h)}(\mathbf{x}, t) = \left\{ \begin{array}{cc} \beta p_i^{(h)}\!\left(\frac{\mathbf{x}}{\beta}, \frac{t}{\beta}\right) & i = 1,2,3 \\ \beta^2 p_i^{(h)}\!\left(\frac{\mathbf{x}}{\beta}, \frac{t}{\beta}\right) & i = 4,5,6 \end{array} \right. \quad \mathbf{x} \in \Omega_\beta. \tag{10}$$

See Appendix A for a proof of this result.

3.2. Proxy Soundbank

To avoid building PAN representations for each unique object in a scene, we instead map objects to ellipsoidal proxies according to their physical properties and build PAN representations for only these proxy objects. By exploiting the scaling relationships presented in §3.1, we can reduce the three dimensional set of all ellipsoids to a much smaller two dimensional set. Every ellipsoid in $\mathbb{R}^3$ is equivalent – up to scaling and rigid transformation – to an ellipsoid defined by $\frac{x^2}{A^2} + \frac{y^2}{B^2} + \frac{z^2}{C^2} = 1$ where $A = 0.5$; that is, ellipsoids with unit length in the x-axis. We also only need to consider ellipsoids for which $C \leq B \leq 0.5$. See §5 for soundbank precomputation details.

3.3. Proxy Sound Synthesis

Consider a rigid-body object O and let $\mathbf{x}_0$, $\mathbf{M}$ and V denote its center of mass, moment of inertia matrix, and volume. The ellipsoidal proxy used to represent O will be chosen so that its principle moments of inertia match those of O. A similar procedure was used to select proxy geometry for modal sound synthesis in [ZJ10]. $\mathbf{M}$ is real and symmetric and can be diagonalized to yield an orthonormal basis $\mathbf{Z}$ and diagonal matrix $\mathbf{D}$ such that $\mathbf{M} = \mathbf{Z}\mathbf{D}\mathbf{Z}^T$. The columns of $\mathbf{Z}$ and diagonal entries of $\mathbf{D}$ are the principle axes and principle moments of inertia for O [GPS02]. For an axis-aligned ellipsoid with mass m, it is straightforward to derive $\mathbf{M}$ from the definition of the moment of inertia [GPS02]:

$$\mathbf{M}_{ellipsoid} = \frac{m}{5} \begin{pmatrix} B^2 + C^2 & 0 & 0 \\ 0 & A^2 + C^2 & 0 \\ 0 & 0 & A^2 + B^2 \end{pmatrix}. \tag{11}$$

Assuming that $D_{11} \leq D_{22} \leq D_{33}$, we set $\mathbf{M}_{ellipsoid} \equiv \mathbf{D}$ and solve the resulting system of equations to obtain ellipse parameters $A \geq B \geq C$. Next, we uniformly rescale (A, B, C) so that the volume of the resulting ellipsoid matches V. Assuming that this ellipsoid has the same density as O, identifying its volume with O's ensures that it also has the same mass as O. The principle moments of inertia for this ellipsoid also match O's up to a scaling factor. This implies that O and its proxy ellipsoid exhibit similar rigid accelerations when subjected to the same external force and ensures that the magnitude of acceleration noise produced by this ellipsoid is consistent with the sound produced by O. Finally, we identify this ellipsoid with a proxy ellipsoid $(A' = 0.5, B', C')$ with

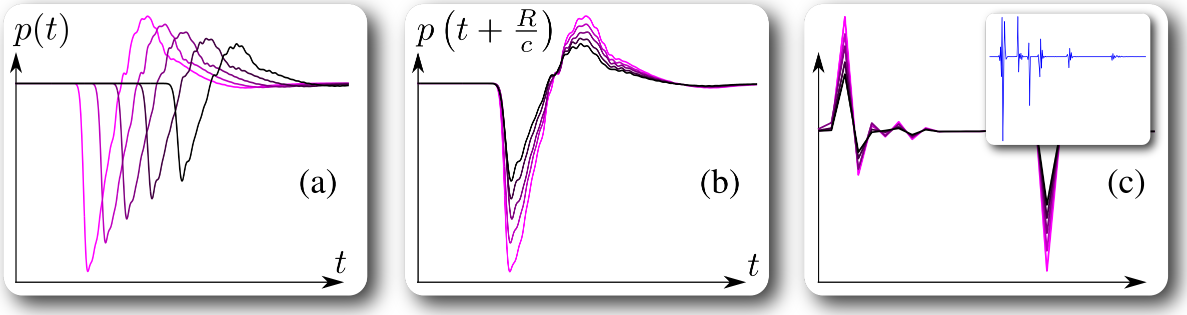

Figure 3: ***Precomputed Acceleration Noise Compression:*** (a) *Acceleration noise signals evaluated with a wave equation solver at several radii R_i ($i = 1,\ldots,5$) in a fixed listening direction;* (b) *Acceleration noise signals time-shifted according to* (13)*;* (c) *A subset of the coefficients from wavelet decompositions of the time-shifted functions from* (b)*. The inset shows a larger set of wavelet coefficients for one of these functions. We compress PAN functions by storing only sufficiently large wavelet coefficients.*

unit length in the x-axis. We choose a scaling factor $\beta = 2A$ so that $(A,B,C) = (\beta A',\beta B',\beta C')$.

To synthesize sound from O, collision forces are estimated using O's original geometry and the methods discussed in §2.2.1. To approximate O's sound contribution at listening position $\mathbf{x}$, we begin by transforming $\mathbf{x}$ in to the coordinate frame of O's proxy ellipsoid: $\mathbf{x}' = \mathbf{Z}^T(\mathbf{x}-\mathbf{x}_0)$. Likewise, we rotate the translational and rotational accelerations $\mathbf{a}(t)$ and $\boldsymbol{\alpha}(t)$ applied to O to find the accelerations $\mathbf{a}'$ and $\boldsymbol{\alpha}'$ acting on the proxy ellipsoid: $\mathbf{a}'(t) = \mathbf{Z}^T\mathbf{a}(t)$, $\boldsymbol{\alpha}'(t) = \mathbf{Z}^T\boldsymbol{\alpha}(t)$. Suppose that O's proxy is parametrized by $(A,B,C) = (\beta A',\beta B',\beta C')$, where (A',B',C') has unit length ($A' = 0.5$). Assuming that we have precomputed $p_i^{(h)}$ ($i = 1,\ldots,6$, $h > 0$) for O's unscaled proxy ellipsoid, we can evaluate $P_i^{(\beta h)}$ for the desired ellipsoid (A,B,C) using the scaling relationships (10). Finally, we use the PAN functions $P_i^{(\beta h)}$ and (7) to recover the total acceleration noise $P(\mathbf{x}',t)$ due to $\mathbf{a}'(t)$ and $\boldsymbol{\alpha}'(t)$ acting on the proxy ellipsoid (A,B,C). Figure 2 summarizes the process of fitting an ellipsoid to O and synthesizing sound from this proxy.

4. Proxy Soundbank Representation

The PAN representation introduced in [CZJ12] stores time signals q_k ($k = 1,\ldots,N$) at a discrete set of angular directions surrounding an object. The signals in each direction are explicitly discretized and stored at some sampling frequency f. Storing these fields at a reasonably high angular resolution can require on the order of 10-100 MB of storage per object. While this may not be particularly expensive in scenes with only a few unique objects, storing precomputed data for a large set of proxy objects could become prohibitively expensive if we use these methods directly. In this section, we discuss techniques for building a memory-efficient representation for our proxy soundbank.

4.1. Precomputed Acceleration Noise Compression

As we originally discussed in §2.2.2, precomputed acceleration noise functions $p_i^{(h)}$ are represented by discretizing the angular space (θ,ϕ) about object O's center of mass $\mathbf{x}_0$ and associating with each angular direction the series representation (8) for $p_i^{(h)}$. For the remainder of this section, we will drop subscripts and superscripts and refer to the function to be approximated simply as $p(\mathbf{x},t)$. To find the values of the functions q_k in each direction (θ,ϕ) the true values of $p(R,\theta,\phi,t)$ are computed at a discrete set of radii $R_1,\ldots,R_M$ by solving (1,9) on O's exterior domain Ω. We can write $p(R_i,\theta,\phi,t)$ as a time series

$$\left\{p_i^0, p_i^1, p_i^2, \ldots\right\} \quad \text{where} \quad p_i^\ell = p(R_i,\theta,\phi,\ell\Delta t). \tag{12}$$

Δt is the simulation time step used to solve (1). In [CZJ12], the functions $q_k(\theta,\phi,t)$ from (8) are also discretized at a sampling rate of $f = 1/\Delta t$. A least-squares system is built by enforcing the condition that (8) holds at each radius R_i and each time sample of (12). The system is solved for the full set of samples for the functions q_k in such a way that the functions are temporally smooth. This approach was successfully applied to a number of example objects; however, it tends to result in very large least-squares systems and memory-intensive representations for the resulting PAN fields.

We introduce a new fitting approach which simultaneously allows for compression of the PAN functions and a less expensive fitting process. Observe that if (8) holds with equality, then the time-shift $t - R/c$ in the right hand side can be moved to the left while preserving equality:

$$p\left(R,\theta,\phi,t+\frac{R}{c}\right) = \sum_{k=1}^{N} \frac{1}{R^k} q_k(\theta,\phi,t). \tag{13}$$

Figure 3 (a) illustrates the function $p(R,\theta,\phi,t)$ evaluated at five radii in a fixed listening direction and figure 3 (b) illustrates the time-shifted signals $p(R,\theta,\phi,t+R/c)$.

Rather than discretizing the PAN functions q_k and explicitly computing their samples, we instead choose to represent the functions q_k in a wavelet basis. Shifting $p(R,\theta,\phi,t)$ in time temporally aligns these signals, allowing us to represent them using the same wavelet basis. Figure 3 (c) illustrates some of the wavelet coefficients for the functions $p(R_i,\theta,\phi,t+R_i/c)$. We represent these signals using a Daubechies wavelet family [Dau88]. Let $\widehat{\mathbf{p}}_i$ be the vector of coefficients in the wavelet basis for the function $p(R_i,\theta,\phi,t+R_i/c)$ and let $\widehat{p}_i^j$ be the j^{th} coefficient. Similarly, let $\widehat{q}_k^j$ be the j^{th} wavelet coefficient for the function $q_k(\theta,\phi,t)$ (θ and ϕ fixed). It follows from (13) and the linearity of the wavelet transform that

$$\widehat{p}_i^j = \sum_{k=1}^{N} \frac{1}{R^k}\widehat{q}_k^j. \tag{14}$$

(14) encodes a $M \times N$ least-squares system for each wavelet coefficient, where M is the number of sampling radii, and N is the number of series terms (5 and 2, respectively, in our examples). We solve (14) repeatedly to recover the full set of wavelet coefficients for the PAN functions q_k. These small, $M \times N$ least-squares systems are significantly easier to solve than the systems appearing in [CZJ12], which tended to have matrix dimensions numbering in the thousands. As in [CZJ12], we discretize the angular space surrounding object O, and repeat this procedure for each direction. This provides us with a representation of $p_i^{(h)}$ which can be efficiently evaluated at each point in space and time.

We compress the PAN fields q_k by only storing certain wavelet coefficients. From Figure 3 it is clear that many wavelet coefficients in $\widehat{\mathbf{p}}_i$ are close to zero. We define a tolerance $\varepsilon > 0$ and choose to store coefficient j if and only if $\exists i : |\widehat{p}_i^j| \geq \varepsilon \|\widehat{\mathbf{p}}_i\|_\infty$.

In practice, we use (7) to write the total pressure as

$$p(\mathbf{x},t) = \sum_{k=0}^{\infty}\sum_{i=1}^{6} z_i(kh)\, p_i^{(h)}(\mathbf{x},t-kh) \tag{15}$$

The inner sum can be evaluated efficiently by summing wavelet coefficients for each directional PAN field and performing a single wavelet reconstruction.

4.2. High-frequency Suppression

Suppose that the PAN fields discussed in §4.1 are stored at some sampling frequency f (so that $\Delta t = 1/f$ in (12)). To evaluate $p_i^{(h)}$, we must reconstruct the time signals q_k – sampled at frequency f – from their wavelet coefficients. Note, however, that when invoking the scaling relationships (10) the effective sampling frequency for $P_i^{(\beta h)}$ is $f' = f/\beta$. Assuming that f is relatively high (96 kHz for our proxy ellipsoids) and β is small, the effective frequency f' may significantly exceed frequencies necessary for high-quality audio synthesis (44-96 kHz). Fortunately, our wavelet PAN representation provides us with a convenient way to reconstruct the signals q_k at approximately the desired output sampling frequency, while suppressing content above this frequency.

As before, let $\widehat{\mathbf{q}}_k$ be the vector of wavelet coefficients for the time signal q_k, which is assumed to be sampled at frequency f. Coefficients are stored in $\widehat{\mathbf{q}}_k$ as follows: $\widehat{\mathbf{q}}_k(0)$ stores the wavelet smoothing coefficient, and $\widehat{\mathbf{q}}_k(2^\ell,\ldots,2^{\ell+1}-1)$ stores the detail coefficients for level $\ell \geq 0$ of the wavelet basis. Intuitively, coefficients at higher indices in $\widehat{\mathbf{q}}_k$ represent higher-frequency content than coefficients at lower indices. Assuming that $\widehat{\mathbf{q}}_k$ has length T (assumed to be a power of 2), this storage scheme has the property that the vector $\widehat{\mathbf{q}}_k' = \widehat{\mathbf{q}}_k(0,\ldots,T/2-1)/\sqrt{2}$ stores the wavelet coefficients for a signal q_k', which is sampled at frequency $f/2$, and is similar to q_k but lacks high-frequency details from the original signal. This relationship allows us to reconstruct the PAN time signal q_k with sampling frequency within a factor of 2 of the desired audio output frequency. Algorithm 1 summarizes the process of reconstructing scaled PAN functions q_k. Note, however, that in practice we do not independently reconstruct the functions q_k (see the last paragraph of §4.1). This process guarantees that we do not introduce aliasing artifacts by synthesizing details at frequencies significantly above the desired audio sampling frequency.

4.3. Ellipsoid Proxy Symmetries

We can further reduce PAN storage for ellipsoids by noting that ellipsoids centered at the origin are symmetrical about each axis. Moreover, symmetries in the boundary conditions (9) allow us to conclude that the following relationships hold for $x,y,z \geq 0$:

$$\begin{aligned} p_1(\mathbf{x}_0) &= p_1(\mathbf{x}_1) = p_1(\mathbf{x}_4) = p_1(\mathbf{x}_5) && (16)\\ &= -p_1(\mathbf{x}_2) = -p_1(\mathbf{x}_3) = -p_1(\mathbf{x}_6) = -p_1(\mathbf{x}_7) \\ p_2(\mathbf{x}_0) &= p_2(\mathbf{x}_1) = p_2(\mathbf{x}_2) = p_2(\mathbf{x}_3) && (17)\\ &= -p_2(\mathbf{x}_4) = -p_2(\mathbf{x}_5) = -p_2(\mathbf{x}_6) = -p_2(\mathbf{x}_7) \\ p_3(\mathbf{x}_0) &= p_3(\mathbf{x}_2) = p_3(\mathbf{x}_4) = p_3(\mathbf{x}_6) && (18)\\ &= -p_3(\mathbf{x}_1) = -p_3(\mathbf{x}_3) = -p_3(\mathbf{x}_5) = -p_3(\mathbf{x}_7) \\ p_4(\mathbf{x}_0) &= p_4(\mathbf{x}_2) = p_4(\mathbf{x}_5) = p_4(\mathbf{x}_7) && (19)\\ &= -p_4(\mathbf{x}_1) = -p_4(\mathbf{x}_3) = -p_4(\mathbf{x}_4) = -p_4(\mathbf{x}_6) \\ p_5(\mathbf{x}_0) &= p_5(\mathbf{x}_3) = p_5(\mathbf{x}_4) = p_5(\mathbf{x}_7) && (20)\\ &= -p_5(\mathbf{x}_1) = -p_5(\mathbf{x}_2) = -p_5(\mathbf{x}_5) = -p_5(\mathbf{x}_6) \\ p_6(\mathbf{x}_0) &= p_6(\mathbf{x}_1) = p_6(\mathbf{x}_6) = p_6(\mathbf{x}_7) && (21)\\ &= -p_6(\mathbf{x}_2) = -p_6(\mathbf{x}_3) = -p_6(\mathbf{x}_4) = -p_6(\mathbf{x}_5) \end{aligned}$$

where $\mathbf{x}_0 = (x,y,z)$, $\mathbf{x}_1 = (x,y,-z)$, $\mathbf{x}_2 = (-x,y,z)$, $\mathbf{x}_3 = (-x,y,-z)$, $\mathbf{x}_4 = (x,-y,z)$, $\mathbf{x}_5 = (x,-y,-z)$, $\mathbf{x}_6 = (-x,-y-z)$, and $\mathbf{x}_7 = (-x,-y,-z)$. Our precomputed soundbank only stores PAN fields for directions in the positive $(x \geq 0, y \geq 0, z \geq 0)$ octant. We use (16-21) to evaluate these fields in all other octants.

5. Results

Implementation Details: We synthesize sound using a precomputed soundbank with 66 ellipsoids. The ellipsoids have

Algorithm 1: Reconstructs scaled precomputed acceleration noise functions at a sampling rate f' within a factor of 2 of the desired output sampling rate f_{out}. The `waverec` function reconstructs a time signal from T wavelet coefficients.

```
input  : PAN wavelet coefficients $\widehat{\mathbf{q}}_k$, scaling factor $\beta$, PAN
         sampling rate $f$, output sampling rate $f_{out}$
output: Time signal $q'_k(t)$ and its sampling rate $f'$
begin
    $T \leftarrow \texttt{length}(\widehat{\mathbf{q}}_k)$
    while $f/\beta > 2f_{out}$ do
        $f \leftarrow f/2$
        $T \leftarrow T/2$
        $\widehat{\mathbf{q}}_k \leftarrow \widehat{\mathbf{q}}_k(0:T-1)/\sqrt{2}$
    $q'_k(t) \leftarrow \texttt{waverec}(\widehat{\mathbf{q}}_k, T)$
    return $[q'_k(t), f/\beta]$
```

parameters $0.5\text{m} = A >= B >= C$ with B and C varying between 0.025m and 0.5m in increments of 0.0475m. We chose this increment to be sufficiently small to guarantee smooth variance of PAN fields across the proxy soundbank (see Figure 4). Since this set is well-sampled, we fit objects to the nearest ellipsoids in the soundbank rather than interpolating between ellipsoids, as the latter approach would require longer synthesis times. We evaluate the pressure time series (12) for each ellipsoid on a 500^3 finite difference grid with a time step of $\Delta t = 1/96000\text{s}$ and use perfectly matched layers [LT97] to avoid reflections from the domain boundary. These high-resolution simulations were carried out over the course of several days on a set of eight 32-core Intel X7560 machines. Figure 6 illustrates the ellipsoids in our proxy set.

For these reference ellipsoids, we choose a time scale $h = 10^{-4}\text{s}$. This time scale was chosen based on results from [CZJ12]. Specifically, the ball bearing example (a steel sphere of radius 0.0075m) in this paper is assigned a time scale of $h_{ball} = 7.3 \times 10^{-6}\text{s}$. Our soundbank time scale of $h = 10^{-4}$ was chosen conservatively to guarantee that an equivalently scaled sphere from our proxy set will have a PAN time scale of approximately $h_{ball}/5$. We find that this time scale is sufficiently small to interpolate contact force profiles of the form (3) encountered in our simulations.

We compute the wavelet transforms discussed in §4.1 with a Daubechies wavelet family with 5 vanishing moments using the *GNU Scientific Library* implementation of the wavelet transform (http://www.gnu.org/software/gsl/). We find that this basis achieves a suitable compromise between performance and compression.

The positive octant $(x \geq 0, y \geq 0, z \geq 0)$ associated with each proxy is discretized by uniformly triangulating the unit sphere in this octant with 64 triangles and 45 vertices. Each vertex represents a direction in which PAN data is stored, and we use linear interpolation to synthesize sound in arbitrary directions. Our experiments show that storing proxy data at this resolution does not introduce significant errors relative to solutions computed with a finite difference solver.

Sound from multibody examples like the ones simulated for this paper tend to exhibit high dynamic range. As a result, normalizing pressure time series to have unit infinity norm tends to produce sounds in which certain parts are abnormally quiet. We address this by post-processing our results with dynamic range compression using *Adobe Soundbooth*. We also present some results post-processed with artificial environmental reverb.

Precomputed Acceleration Noise Compression: We find that with a PAN compression parameter of $\varepsilon = 0.01$ produces compressed PAN fields that exhibit small errors relative to the explicit precomputed solutions (12) (on the order of 1-5%). While increasing this parameter does increase numerical error, noticeable differences in sounds synthesized using our proxy soundbank only become apparent at higher values of ε. See the accompanying result video for comparisons of sounds synthesized from soundbanks with varying ε. The following table details proxy data storage sizes for numerous values of ε (for reference, uncompressed PAN fields stored at the same resolution require 293MB):

ε	0.01	0.02	0.04	0.08	0.16	0.32	0.64
Size (MB)	26	20	16	12	7.3	4.5	2.6

The compressed PAN representation introduced in §4.1 is of general use, even for examples not computed using proxies. We apply the wavelet fitting procedure to several example objects from [CZJ12] and compare our results to the methods originally proposed in this paper. See Table 1 for a comparison of memory usage/synthesis times and the supplemental video for a comparison of acceleration noise results computed with these two approaches.

Model	PAN size (MB)		Synthesis time (s)	
	[CZJ12]	Current result	[CZJ12]	Current result
Plate	108	12	3.72	2.02
Mug	1131	76	10.33	3.42
Dice	35	6.9	0.40	0.19
Rounded Dice	35	6.1		
Coin	35	9.3	16.00	7.70

Table 1: ***Precomputed Acceleration Noise Compression:*** *We compare memory use and acceleration sound synthesis times to those of [CZJ12] for a selection of models and example scenes from that paper's results. Results are reported for PAN fields with 3200 discrete angular directions to coincide with the original PAN results. For all examples, we choose the wavelet compression tolerance to be $\varepsilon = 0.04$. This was determined experimentally as roughly the largest ε we could use before producing noticeably different results.*

Proxy Validation: In [CZJ12], the authors compute acceleration noise for two fracture simulations by explicitly building PAN representations for every piece produced in the simulations. Using the same simulation data, we compare these

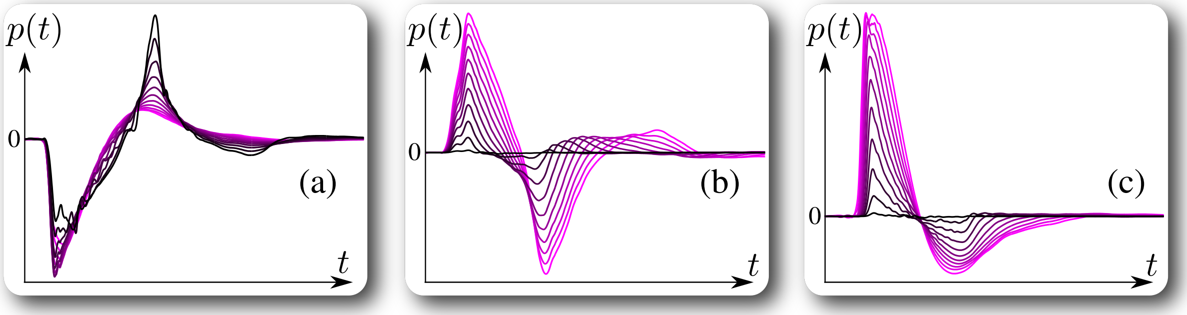

Figure 4: ***Varying precomputed acceleration noise over the proxy soundbank:*** *The precomputed acceleration noise function $p_3^{(h)}$ (translation in the z-axis) evaluated with several proxy ellipsoids.* (a) *Varying ellipsoid parameter C between 0.025m and 0.5m while $A = B$ are held fixed at 0.5m.* (b) *Varying parameter B between 0.025m and 0.5m while $A = 0.5m$ and $C = 0.025m$.* (c) *Varying parameters B and C simultaneously ($B = C$) between 0.025m and 0.5m with $A = 0.5m$.*

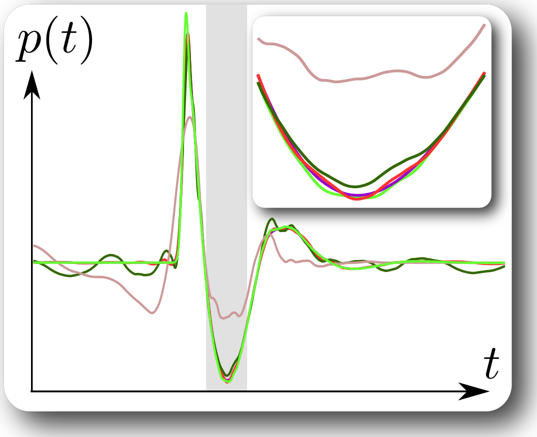

Figure 5: ***Varying wavelet compression:*** *We visualize $p_5^{(h)}$ at a fixed position with varying levels of wavelet compression. The object considered here is an ellipsoid with $a = 0.5m$, $b = 0.405m$ and $c = 2625m$. The inset shows a close-up of the highlighted region. Signals compressed with $\varepsilon = 0.01$ and $\varepsilon = 0.04$ (purple and red, respectively) exhibit good agreement with the finite difference solution (light green) with small errors arising from angular discretization. Fields compressed with $\varepsilon = 0.16$ and $\varepsilon = 0.64$ (dark green and pink, respectively) exhibit more significant errors.*

results with sounds computed using our proxy soundbank. The original approach requires many hours of precomputation to build PAN fields for each object in these scenes. This approach also requires hundreds to thousands of MB of storage for PAN fields. Our method avoids this cost by synthesizing all acceleration noise with ellipsoidal proxies. We also present comparisons with sounds synthesized using a simpler proxy model in which each object O with volume V is approximated by a spherical proxy with volume V. We find that this method results in significant degradation of quality compared to our results. In particular, the contribution of acceleration noise tends to be severely underestimated by this method. This suggests that our approach is indeed capturing acceleration noise phenomena that is difficult to resolve with simpler techniques. See the supplemental video for these comparisons.

EXAMPLE (Rock Pile): To test the scalability of our method, we model a scene with 1000 unique, procedurally generated rocks and synthesize acceleration noise from the resulting simulation. We compute two examples: one with small rocks (≈ 1-5cm in diameter) and one with larger rocks (≈ 2-20cm in diameter). While the example with smaller rocks produces some modal sound, acceleration noise dominates this result. The example with larger rocks produces significantly louder modal sound, but the addition of acceleration noise still complements this example by introducing details not present in the modal result.

EXAMPLE (Glass Fracture): In this simulation, a glass pane falls to the ground and shatters in to 315 small pieces. Without acceleration noise the debris produces very little sound.

EXAMPLE (Breaking Plates): We simulate 10 plates falling to the ground and breaking. Many of the objects generated in this simulation produce no modal sound and the addition of acceleration noise produces a substantially richer and more detailed result.

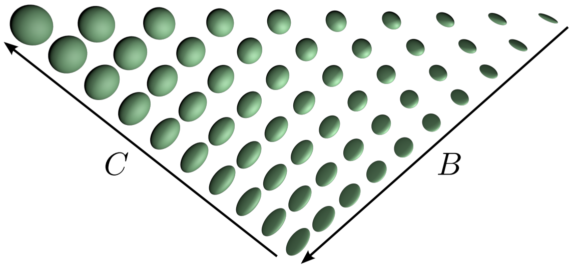

Figure 6: ***Ellipsoid Proxy Soundbank:*** *All ellipsoid objects for which PAN fields are precomputed. Our results are computed by fitting objects to scaled ellipsoids from this set.*

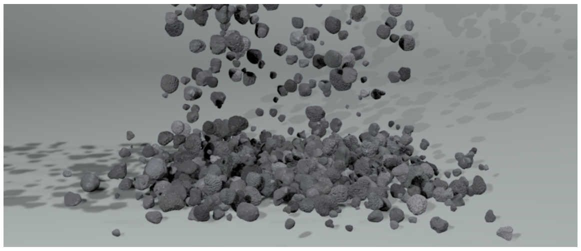

Figure 7: ***Rock pile:*** *Synthesizing acceleration noise for this falling pile of 1000 procedurally generated rocks would require extensive precomputation to exactly resolve each object's contribution. Instead, we approximate each object with a proxy ellipsoid and synthesize acceleration noise with data from our precomputed soundbank.*

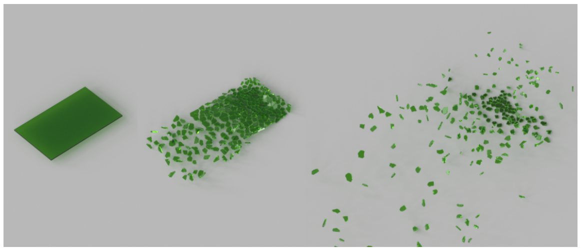

Figure 8: ***Glass fracture:*** *This fracture simulation generates over 300 small objects with no audible vibration modes. Our method allows us to recover sound from this example by efficiently synthesizing acceleration noise for each piece using our proxy soundbank.*

Example	Duration (s)	Δt (ms)	# impulses	Synthesis time (s)
Single plate fracture	5	0.025	1293	1
Multiple plate fracture	4	0.25	5921	40
Glass fracture (71 pieces)	2	0.025	2507	2
Glass fracture (316 pieces)	3	0.025	18787	12
Rock pile (large)	6	0.25	210741	192
Rock pile (small)	6	0.25	96579	97

Table 2: ***Sound Synthesis Statistics:*** *Acceleration sound synthesis times for our examples. The duration and Δt columns report the length and time step duration for the rigid-body simulation. # impulses refers to number of impulses used for sound synthesis.*

6. Conclusion

We presented an efficient method for synthesizing rigid-body acceleration noise from complex multibody scenes with hundreds to thousands of objects. We avoid precomputing acceleration noise data for each object in a scene by introducing an ellipsoid proxy model for acceleration sound. We build a soundbank of precomputed acceleration noise data for a set of ellipsoid proxies and limit the size of this data set by making use of a new wavelet compression scheme for precomputed acceleration noise data. As a result, the proxy soundbank only requires between 5 and 26MB of memory, depending on the amount of compression applied. This method introduces significant detail when applied to rigid-body fracture simulations and other simulations with large quantities of procedurally generated debris.

Limitations and Future Work: Our method computes sound independently from each object in a scene and adds these sounds together to recover the complete result. Ignoring acoustic interactions between objects may fail to capture interesting sound phenomena, particularly in scenes involving many bodies stacked on top of each other (e.g., Figure 7). Existing brute force methods for resolving this phenomena are far too costly for animation sound synthesis. Developing efficient methods for resolving acoustic interactions between objects for both modal and acceleration sound is a challenging problem and an interesting area for future work.

Our results currently include only modal and acceleration sound from the objects in each scene. We do not currently synthesize sound from the ground plane. Zheng and James [ZJ10] synthesized modal sound for fracture examples, and included modal sound from the ground plane by synthesizing sound from a concrete slab. Including ground plane noise would likely enhance the realism of our results somewhat.

Our experiments show that our proxy-based synthesis pipeline is particularly effective for scenes involving small debris-like objects, producing results similar to those generated with object-specific precomputation. This is advantageous, as it is precisely objects like this for which acceleration noise is the dominant sound source. While our method does not accurately predict acceleration noise for large, non-convex objects, the contribution of acceleration noise for these objects is typically less significant relative to that of modal sound. Nevertheless, enriching our proxy database with additional object categories to better approximate acceleration noise from larger, non-convex objects is an interesting area for future work.

Acknowledgments: We would like to thank the anonymous reviewers for their helpful feedback. This work was supported in part by the National Science Foundation (HCC-0905506), an NSERC Postgraduate Scholarship, fellowships from the Alfred P. Sloan Foundation and the John Simon Guggenheim Memorial Foundation, and donations from Side Effects Software, Pixar, Autodesk, and Vision Research. This research was conducted in conjunction with the Intel Science and Technology Center – Visual Computing. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or others.

References

[BDT*08] Bonneel N., Drettakis G., Tsingos N., Viaud-Delmon I., James D.: Fast Modal Sounds with Scalable Frequency-Domain Synthesis. *ACM Transactions on Graphics 27*, 3 (Aug. 2008), 24:1–24:9. 1, 2

[CAJ09] Chadwick J. N., An S. S., James D. L.: Harmonic

Shells: A Practical Nonlinear Sound Model for Near-Rigid Thin Shells. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2009) 28*, 3 (Dec. 2009). 1

[CL01] Chaigne A., Lambourg C.: Time-domain simulation of damped impacted plates. i. theory and experiments. *Journal of the Acoustical Society of America 109*, 4 (2001), 1422–1432. 2

[CZJ12] Chadwick J. N., Zheng C., James D. L.: Precomputed acceleration noise for improved rigid-body sound. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012) 31*, 4 (Aug. 2012). 2, 3, 5, 6, 7

[Dau88] Daubechies I.: Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics 41*, 7 (1988). 6

[ENNS81] Endo M., Nishi S., Nakagawa M., Sakata M.: Sound radiation from a circular cylinder subjected to elastic collision by a sphere. *Journal of Sound and Vibration 75*, 2 (1981), 285–302. 2

[GBF03] Guendelman E., Bridson R., Fedkiw R.: Nonconvex rigid bodies with stacking. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003) 22*, 3 (Aug. 2003). 3

[GPS02] Goldstein H., Poole C., Safko J.: *Classical Mechanics*, third ed. Addison Wesley, San Francisco, California, 2002. 4

[Her82] Hertz H.: Über die Berührung fester elastiche Körper and über die harte (On the contact of elastic solids). *J. reine und angewandte Mathematk 92* (1882), 156–171. 2, 3

[JBP06] James D. L., Barbič J., Pai D. K.: Precomputed Acoustic Transfer: Output-sensitive, accurate sound generation for geometrically complex vibration sources. *ACM Transactions on Graphics 25*, 3 (July 2006), 987–995. 1

[Joh85] Johnson K. L.: *Contact Mechanics*. Cambridge University Press, 1985. 2, 3

[LCM01] Lambourg C., Chaigne A., Matignon D.: Time-domain simulation of damped impacted plates. ii. numerical model and results. *Journal of the Acoustical Society of America 109*, 4 (2001), 1433–1447. 2

[LT97] Liu Q.-H., Tao K.: The perfectly matched layer for acoustic waves in absorptive media. *Journal of the Acoustical Society of America 102*, 4 (1997), 2072–2082. 7

[MKP11] Mehraby K., Khademhosseini H., Poursina M.: Impact Noise Radiated by Collision of Two Spheres: Comparison Between Numerical Simulations, Experiments and Analytical Results. *Journal of Mechanical Science and Technology 25*, 7 (2011), 1675–1685. 2, 3

[MN88] Mitchell D. P., Netravali A. N.: Reconstruction filters in computer-graphics. In *Proceedings of SIGGRAPH 1988* (1988), pp. 221–228. 3

[OCE01] O'Brien J. F., Cook P. R., Essl G.: Synthesizing sounds from physically based motion. In *Proceedings of ACM SIGGRAPH 2001* (Aug. 2001), Computer Graphics Proceedings, Annual Conference Series, pp. 529–536. 2

[OSG02] O'Brien J. F., Shen C., Gatchalian C. M.: Synthesizing sounds from rigid-body simulations. In *ACM SIGGRAPH Symposium on Computer Animation* (July 2002), pp. 175–181. 1, 2

[RO07] Ross A., Ostiguy G.: Propagation of the initial transient noise from an impacted plate. *Journal of Sound and Vibration 301*, 1 (2007), 28–42. 2

[RWJ79a] Richards E. J., Wescott M. E., Jayapalan R. K.: On the prediction of impact noise, i: Acceleration noise. *Journal of Sound and Vibration 62*, 4 (1979), 547–575. 2

[RWJ79b] Richards E. J., Wescott M. E., Jayapalan R. K.: On the prediction of impact noise, ii: Ringing noise. *Journal of Sound and Vibration 65*, 3 (1979), 419–451. 2

[SLC99] Schedin S., Lambourge C., Chaigne A.: Transient sound fields from impacted plates: Comparison between numerical simulations and experiments. *Journal of Sound and Vibration 221*, 3 (1999), 471–490. 2

[vdDKP01] van den Doel K., Kry P. G., Pai D. K.: FoleyAutomatic: Physically Based Sound Effects for Interactive Simulation and Animation. In *Proceedings of ACM SIGGRAPH 2001* (Aug. 2001), Computer Graphics Proceedings, Annual Conference Series, pp. 537–544. 1, 2

[WGM94] Wåhlin A. O., Gren P. O., Molin N.-E.: On structure borne sound: Experiments showing the initial transient acoustic wave field generated by an impacted plate. *Journal of the Acoustical Society of America 96*, 5 (1994), 2791–2797. 2

[YZ92] Yufang W., Zhongfang T.: Sound Radiated from the Impact of Two Cylinders. *Journal of Sound and Vibration 159*, 2 (1992), 295–303. 2

[ZJ10] Zheng C., James D. L.: Rigid-body fracture sound with precomputed soundbanks. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2010) 29*, 3 (July 2010). 2, 4, 9

Appendix A: Scaling Relationship Proof

Following the notation of §3.1, consider object O with PAN fields $p_i^{(h)}$ and object O_β – a uniformly scaled version of object O. We first show that (1) holds for $P_i^{(\beta h)}$ on the domain Ω_β. Consider $\mathbf{x} \in \Omega_\beta, t \geq 0$ and let $\mathbf{y} = \mathbf{x}/\beta, \tau = t/\beta$. It is clear that if $\mathbf{x} \in \Omega_\beta$ then $\mathbf{y} \in \Omega$. For $i = 1,2,3$ we have $\frac{1}{c^2}\frac{\partial^2 P_i^{(\beta h)}(\mathbf{x},t)}{\partial t^2} = \frac{1}{\beta c^2}\frac{\partial^2 p_i^{(h)}(\mathbf{y},\tau)}{\partial \tau^2}$ by definition (10) and the chain rule. We also have $\nabla^2 P_i^{(\beta h)}(\mathbf{x},t) = \frac{1}{\beta}\nabla^2_{\mathbf{y}} p_i^{(h)}(\mathbf{y},t)$. Since $p_i^{(h)}$ satisfies (1) on Ω, it follows from these equations that $\frac{1}{c^2}\frac{\partial^2 P_i^{(\beta h)}(\mathbf{x},t)}{\partial t^2} = \nabla^2 P_i^{(\beta h)}(\mathbf{x},t)$ on Ω_β, as required. The proof for $i = 4,5,6$ is analogous.

Next, we show that the boundary conditions (9) hold for $P_i^{(\beta h)}$. Let $\mathbf{x} \in \partial\Omega_\beta$ and define $\mathbf{y}$ and τ as before. Clearly $\mathbf{y} \in \partial\Omega$ and $\mathbf{n}_\beta(\mathbf{x}) = \mathbf{n}(\mathbf{y})$ where $\mathbf{n}$ and $\mathbf{n}_\beta$ are the normal fields on O and O_β, respectively. We will prove that (9) holds for $i = 4,5,6$ since the proof for $i = 1,2,3$ is similar. Following from (9) and (10), for $i = 4,5,6$, we have $\nabla P_i^{(\beta h)}(\mathbf{x},t) \cdot \mathbf{n}(\mathbf{x}) = \beta \nabla_{\mathbf{y}} p_i^{(h)}(\mathbf{y},\tau) \cdot \mathbf{n}(\mathbf{y})$ by the definition of $P_i^{(\beta h)}$ and the chain rule. We also have $\psi(t;\beta h) g_i(\mathbf{x}) = \psi(\beta\tau;\beta h) g_i(\mathbf{x}) = \beta\psi(\tau;h) g_i(\mathbf{y})$, which is true since $\psi(\beta\tau;\beta h) = \psi(\tau;h)$ and $g_i(\mathbf{x}) = -\rho\,(\mathbf{e}_{i-3} \times (\mathbf{x} - \mathbf{x}_0)) \cdot \mathbf{n}(\mathbf{x}) = -\beta\rho\,(\mathbf{e}_{i-3} \times (\mathbf{y} - \mathbf{y}_0)) \cdot \mathbf{n}(\mathbf{y})$. Therefore,

$$\begin{aligned} \nabla P_i^{(\beta h)}(\mathbf{x},t) \cdot \mathbf{n}(\mathbf{x}) &= \beta \nabla_y p_i^{(h)}(\mathbf{y},\tau) \\ &= \beta \Psi(\tau;h) g_i(\mathbf{y}) \text{ since } p_i^{(h)} \text{ satisfies (9)} \\ &= \Psi(t;\beta h) g_i(\mathbf{x}), \end{aligned}$$

as required. □