Example-Guided Physically Based Modal Sound Synthesis

ZHIMIN REN and HENGCHIN YEH and MING C. LIN
University of North Carolina at Chapel Hill
http://gamma.cs.unc.edu/AUDIO_MATERIAL

Linear modal synthesis methods have often been used to generate sounds for rigid bodies. One of the key challenges in widely adopting such techniques is the lack of automatic determination of satisfactory material parameters that recreate realistic audio quality of sounding materials. We introduce a novel method using pre-recorded audio clips to estimate material parameters that capture the inherent quality of recorded sounding materials. Our method extracts perceptually salient features from audio examples. Based on psychoacoustic principles, we design a parameter estimation algorithm using an optimization framework and these salient features to guide the search of the best material parameters for modal synthesis. We also present a method that compensates for the differences between the real-world recording and sound synthesized using solely linear modal synthesis models to create the final synthesized audio. The resulting audio generated from this sound synthesis pipeline well preserves the same sense of material as a recorded audio example. Moreover, both the estimated material parameters and the residual compensation naturally transfer to virtual objects of different sizes and shapes, while the synthesized sounds vary accordingly. A perceptual study shows the results of this system compares well with real-world recordings in terms of material perception.

Categories and Subject Descriptors: H.5.5 [**Information Systems**]: Sound and Music Computing—*Signal Synthesis*; I.3.6 [**Computer Graphics**]: Methods and Techniques—*Interaction Techniques*; G.1.6 [**Mathematics of Computing**]: Optimization

General Terms:

Additional Key Words and Phrases: Sound Synthesis, Parameter Estimation, Material Properties

ACM Reference Format:

Ren, Z., Yeh, H., and Lin, M. C. 2012. Example-Guided Physically Based Modal Sound Synthesis. ACM Trans. Graph. VV, N, Article XXX (Month YYYY), 16 pages.

The support of National Science Foundation, U.S. Army Research Office, Intel Corporation, and Carolina Development Foundation is gratefully acknowledged. Authors' addresses: Z. Ren, H. Yeh and M. C. Lin; email: {zren, hyeh, lin}@cs.unc.edu

DOI 10.1145/XXXXXXX.YYYYYYY
http://doi.acm.org/10.1145/XXXXXXX.YYYYYYY

DOI = 10.1145/XXXXXXX.YYYYYYY
http://doi.acm.org/10.1145/XXXXXXX.YYYYYYY

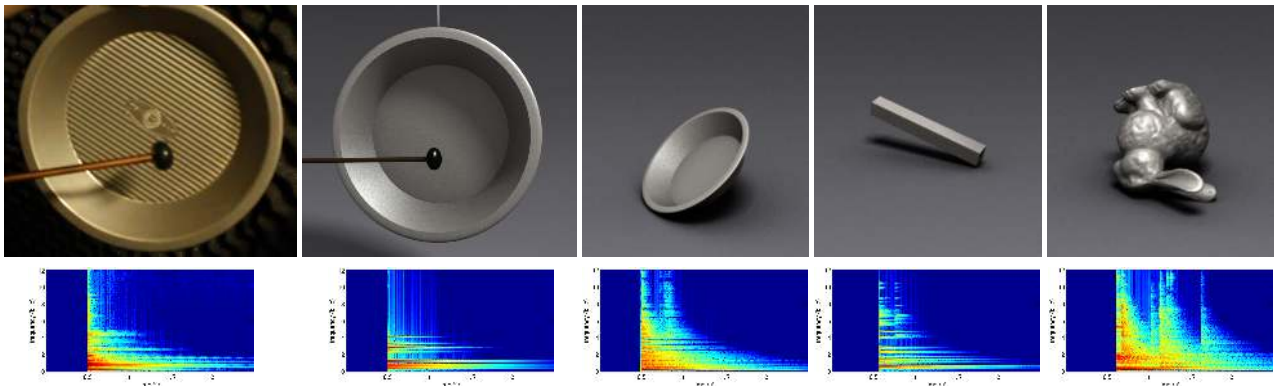
1. INTRODUCTION

Sound plays a prominent role in a virtual environment. Recent progress has been made on sound synthesis models that automatically produce sounds for various types of objects and phenomena. However, it remains a demanding task to add high-quality sounds to a visual simulation that attempts to depict its real-world counterpart. Firstly, there is the difficulty for digitally synthesized sounds to emulate real sounds as closely as possible. Lack of true-to-life sound effects would cause a visual representation to lose its believability. Secondly, sound should be closely synchronized with the graphical rendering in order to contribute to creation of a compelling virtual world. Noticeable disparity between the dynamic audio and visual components could lead to a poor virtual experience for users.

The traditional sound effect production for video games, animation, and movies is a laborious practice. Talented Foley artists are normally employed to record a large number of sound samples in advance and manually edit and synchronize the recorded sounds to a visual scene. This approach generally achieves satisfactory results. However, it is labor-intensive and cannot be applied to all interactive applications. It is still challenging, if not infeasible, to produce sound effects that precisely capture complex interactions that cannot be predicted in advance.

On the other hand, *modal synthesis* methods are often used for simulating sounds in real-time applications. This approach generally does not depend on any pre-recorded audio samples to produce sounds triggered by all types of interactions, so it does not require manually synchronizing the audio and visual events. The produced sounds are capable of reflecting the rich variations of interactions and also the geometry of the sounding objects. Although this approach is not as demanding during run time, setting up good initial parameters for the virtual sounding materials in *modal analysis* is a time-consuming and non-intuitive process. When faced with a complicated scene consisting of many different sounding materials, the parameter selection procedure can quickly become prohibitively expensive and tedious.

Although tables of material parameters for stiffness and mass density are widely available, directly looking up these parameters in physics handbooks does not offer as intuitive, direct control as using a recorded audio example. In fact, sound designers often record their own audio to obtain the desired sound effects. This paper presents a new data-driven sound synthesis technique that preserves the realism and quality of audio recordings, while exploiting all the advantages of physically based modal synthesis. We introduce a computational framework that takes just one example audio recording and estimates the intrinsic *material parameters* (such as stiff-



(a) (b) (c) (d) (e)

Fig. 1: From the recording of a real-world object (a), our framework is able to find the material parameters and generates similar sound for a replicate object (b). The same set of parameters can be transfered to various virtual objects to produce sounds with the same material quality ((c), (d), (e)).

ness, damping coefficients, and mass density) that can be directly used in modal analysis.

As a result, for objects with different geometries and run-time interactions, different sets of modes are generated or excited differently, and different sounds are produced. However, if the material properties are the same, they should all sound like coming from the same material. For example, a plastic plate being hit, a plastic ball being dropped, and a plastic box sliding on the floor generate different sounds, but they all sound like 'plastic', as they have the same material properties. Therefore, if we can deduce the material properties from a recorded sound and *transfer* them to different objects with rich interactions, the *intrinsic quality* of the original sounding material is preserved. Our method can also compensate the differences between the example audio and the modal-synthesized sound. Both the material parameters and the residual compensation are capable of being transfered to virtual objects of varying sizes and shapes and capture all forms of interactions. Fig. 1 shows an example of our framework. From one recorded impact sound (Fig. 1a), we estimated material parameters, which can be directly applied to various geometries (Fig. 1c, 1d, 1e) to generate audio effects that automatically reflect the shape variation while still preserve the same sense of material. Fig. 2 depicts the pipeline of our approach, and its various stages are explained below.

Feature extraction: Given a recorded impact audio clip, from which we first extract some high-level *features*, namely, a set of damped sinusoids with constant frequencies, dampings, and initial amplitudes (Sec. 4). These features are then used to facilitate estimation of the material parameters (Sec. 5), and guide the residual compensation process (Sec. 6).

Parameter estimation: Due to the constraints of the sound synthesis model, we assume a limited input from just one recording and it is challenging to estimate the material parameters from one audio sample. To do so, a virtual object of the same size and shape as the real-world object used in recording the example audio is created. Each time an estimated set of parameters are applied to the virtual object for a given impact, the generated sound, as well as the feature information of the resonance modes, are compared with the real world example sound and extracted features respectively using a difference metric. This metric is designed based on *psychoacoustic* principles, and aimed at measuring both the audio material resemblance of two objects and the perceptual similarity between two sound clips. The optimal set of material parameters is thereby determined by minimizing this perceptually inspired metric function (see Sec. 5). These parameters are readily transferable to other virtual objects of various geometries undergoing rich interactions, and the synthesized sounds preserve the intrinsic quality of the original sounding material.

Residual compensation: Finally, our approach also accounts for the residual, i.e. the approximated differences between the real-world audio recording and the modal synthesis sound with the estimated parameters. First, the residual is computed using the extracted features, the example recording, and the synthesized audio. Then at run-time, the residual is transfered to various virtual objects. The transfer of residual is guided by the transfer of modes, and naturally reflects the geometry and run-time interaction variation (see Sec. 6).

Our key contributions are summarized below:

—A feature-guided parameter estimation framework to determine the optimal material parameters that can be used in existing modal sound synthesis applications.

—An effective residual compensation method that accounts for the difference between the real-world recording and the modal-synthesized sound.

—A general framework for synthesizing rigid-body sounds that closely resemble recorded example materials.

—Automatic transfer of material parameters and residual compensation to different geometries and runtime dynamics, producing realistic sounds that vary accordingly.

2. RELATED WORK

In the last couple of decades, there has been strong interest in digital sound synthesis in both computer music and computer graphics communities due to the needs for auditory display in virtual environment applications. The traditional practice of Foley sounds is still widely adopted by sound designers for applications like video games and movies. Real sound effects are recorded and edited to match a visual display. More recently, *granular synthesis* became

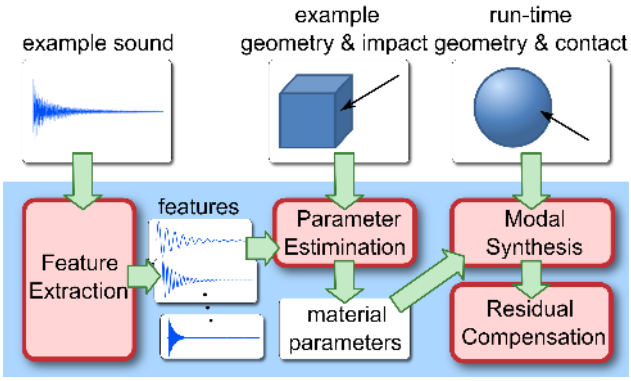


Fig. 2: Overview of the example-guided sound synthesis framework (shown in the blue block): Given an example audio clip as input, features are extracted. They are then used to search for the optimal material parameters based on a perceptually inspired metric. A residual between the recorded audio and the modal synthesis sound is calculated. At run-time, the excitation is observed for the modes. Corresponding rigid-body sounds that have a similar audio quality as the original sounding materials can be automatically synthesized. A modified residual is added to generate a more realistic final sound.

a popular technique to create sounds with computers or other digital synthesizers. Short grains of sounds are manipulated to form a sequence of audio signals that sound like a particular object or event. Roads [2004] gave an excellent review on the theories and implementation of generating sounds with this approach. Picard et al. [2009] proposed techniques to mix sound grains according to events in a physics engine.

Physically Based Sound Synthesis: Another approach for simulating sound sources is using physically based simulation to synthesize realistic sounds that automatically synchronize with the visual rendering. Generating sounds of interesting natural phenomena like fluid dynamics and aerodynamics have been proposed [Dobashi et al. 2003; 2004; Zheng and James 2009; Moss et al. 2010; Chadwick and James 2011]. The ubiquitous rigid-body sounds play a vital role in all types of virtual environments, and these sounds are what we focus on in this paper. O'Brien et al. [2001] proposed simulating rigid bodies with deformable body models that approximates solid objects' small-scale vibration leading to variation in air pressure, which propagates sounds to human ears. Their approach accurately captures surface vibration and wave propagation once sounds are emitted from objects. However, it is far from being efficient enough to handle interactive applications. Adrien [1991] introduced *modal synthesis* to digital sound generation. For real-time applications, *linear modal sound synthesis* has been widely adopted to synthesize rigid-body sounds [van den Doel and Pai 1998; O'Brien et al. 2002; Raghuvanshi and Lin 2006; James et al. 2006; Zheng and James 2010]. This method acquires a modal model (i.e. a bank of damped sinusoidal waves) using *modal analysis* and generates sounds at runtime based on excitation to this modal model. Moreover, sounds of complex interaction can be achieved with modal synthesis. Van den Doel et al. [2001] presented parametric models to approximate contact forces as excitation to modal models to generate impact, sliding, and rolling sounds. Ren et al. [2010] proposed including normal map information to simulate sliding sounds that reflect contact surface details. More recently, Zheng and James [2011] created highly realistic contact sounds with linear modal synthesis by enabling non-rigid sound phenomena and modeling vibrational contact damping. Moreover, the standard modal synthesis can be accelerated with techniques proposed by [Raghuvanshi and Lin 2006; Bonneel et al. 2008], which make synthesizing a large number of sounding objects feasible at interactive rates.

The use of linear modal synthesis is not limited to creating simple rigid-body sounds. Chadwick et al. [2009] used modal analysis to compute linear mode basis, and added nonlinear coupling of those modes to efficiently approximate the rich thin-shell sounds. Zheng and James [2010] extended linear modal synthesis to handle complex fracture phenomena by precomputing modal models for ellipsoidal sound proxies.

However, few previous sound synthesis work addressed the issue of how to determine material parameters used in modal analysis to more easily recreate realistic sounds.

Parameter Acquisition: Spring-mass [Raghuvanshi and Lin 2006] and finite element [O'Brien et al. 2002] representations have been used to calculate the modal model of arbitrary shapes. Challenges lie in how to choose the material parameters used in these representations. Pai et al. [2001] and Corbett et al. [2007] directly acquires a modal model by estimating modal parameters (i.e. amplitudes, frequencies, and dampings) from measured impact sound data. A robotic device is used to apply impulses on a real object at a large number of sample points, and the resulting impact sounds are analyzed for modal parameter estimation. This method is capable of constructing a virtual sounding object that faithfully recreates the audible resonance of its measured real-world counterpart. However, each new virtual geometry would require a new measuring process performed on a real object that has exactly the same shape, and it can become prohibitively expensive with an increasing number of objects in a scene. This approach generally extracts hundreds of parameters for one object from many audio clips, while the goal of our technique instead is to estimate the few parameters that best represent one *material* of a sounding object from only *one* audio clip.

To the best of our knowledge, the only other research work that attempts to estimate sound parameters from one recorded clip is by Lloyd et al. [2011]. Pre-recorded real-world impact sounds are utilized to find peak and long-standing resonance frequencies, and the amplitude envelopes are then tracked for those frequencies. They proposed using the tracked time-varying envelope as the amplitude for the modal model, instead of the standard damped sinusoidal waves in conventional modal synthesis. Richer and more realistic audio is produced this way. Their data-driven approach estimates the modal parameters instead of material parameters. Similar to the method proposed by Pai et al. [2001], these are per-mode parameters and not transferable to another object with corresponding variation. At runtime, they randomize the gains of all tracked modes to generate an illusion of variation when hitting different locations on the object. Therefore, the produced sounds do not necessarily vary correctly or consistently with hit points. Their adopted resonance modes plus residual resynthesis model is very similar to that of SoundSeed Impact [Audiokinetic 2011], which is a sound synthesis tool widely used in the game industry. Both of these works extract and track resonance modes and modify them with signal processing techniques during synthesis. None of them attempts to fit the extracted per-mode data to a modal sound synthesis model, i.e. estimating the higher-level *material parameters*.

In computer music and acoustic communities, researchers proposed methods to calibrate physically based virtual musical instruments.

For example, Välimäki et al. [1996; 1997] proposed a physical model for simulating plucked string instruments. They presented a parameter calibration framework that detects pitches and damping rates from recorded instrument sounds with signal processing techniques. However, their framework only fits parameters for strings and resonance bodies in guitars, and it cannot be easily extended to extract parameters of a general rigid-body sound synthesis model. Trebian and Oliveira [2009] presented a sound synthesis method with linear digital filters. They estimated the parameters for recursive filters based on pre-recorded audio and re-synthesized sounds in real time with digital audio processing techniques. This approach is not designed to capture rich physical phenomena that are automatically coupled with varying object interactions. The relationship between the perception of sounding objects and their sizes, shapes, and material properties have been investigated with experiments, among which Lakatos et al. [1997] and Fontana [2003] presented results and studied human's capability to tell materials, sizes, and shapes of objects based on their sounds.

Modal Plus Residual Models: The sound synthesis model with a deterministic signal plus a stochastic residual was introduced to spectral synthesis by Serra and Smith [1990]. This approach analyzes an input audio and divides it into a deterministic part, which are time-variant sinusoids, and a stochastic part, which is obtained by spectral subtraction of the deterministic sinusoids from the original audio. In the resynthesis process, both parts can be modified to create various sound effects as suggested by Cook [1996; 1997; 2002] and Lloyd et al. [2011]. Methods for tracking the amplitudes of the sinusoids in audio dates back to Quateri and McAulay [1985], while more recent work [Serra and Smith III 1990; Serra 1997; Lloyd et al. 2011] also proposes effective methods for this purpose. All of these works directly construct the modal sounds with the extracted features, while our modal component is synthesized with the estimated material parameters. Therefore, although we adopt the same concept of modal plus residual synthesis for our framework, we face different constraints due to the new objective in material parameter estimation, and render these existing works not applicable to the problem addressed in this paper. Later, we will describe our feature extraction (Sec. 4) and residual compensation (Sec. 6) methods that are suitable for material parameter estimation.

3. BACKGROUND

Modal Sound Synthesis: The standard linear modal synthesis technique [Shabana 1997] is frequently used for modeling of dynamic deformation and physically based sound synthesis. We adopt tetrahedral finite element models to represent any given geometry [O'Brien et al. 2002]. The displacements, $\mathbf{x} \in \mathbb{R}^{3N}$, in such a system can be calculated with the following linear deformation equation:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}, \tag{1}$$

where $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ respectively represent the mass, damping and stiffness matrices. For small levels of damping, it is reasonable to approximate the damping matrix with *Rayleigh damping*, i.e. representing damping matrix as a linear combination of mass matrix and stiffness matrix: $\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}$. This is a well-established practice and has been adopted by many modal synthesis related works in both graphics and acoustics communities. After solving the generalized eigenvalue problem

$$\mathbf{K}\mathbf{U} = \mathbf{\Lambda}\mathbf{M}\mathbf{U}, \tag{2}$$

the system can be decoupled into the following form:

$$\ddot{\mathbf{q}} + (\alpha\mathbf{I} + \beta\mathbf{\Lambda})\dot{\mathbf{q}} + \mathbf{\Lambda}\mathbf{q} = \mathbf{U}^T\mathbf{f}, \tag{3}$$

where $\mathbf{\Lambda}$ is a diagonal matrix, containing the eigenvalues of Eqn. 2; $\mathbf{U}$ is the eigenvector matrix, and transforms $\mathbf{x}$ to the decoupled deformation bases $\mathbf{q}$ with $\mathbf{x} = \mathbf{U}\mathbf{q}$.

The solution to this decoupled system, Eqn. 3, are a bank of *modes*, i.e. damped sinusoidal waves. The i'th mode looks like:

$$q_i = a_i e^{-d_i t}\sin(2\pi f_i t + \theta_i), \tag{4}$$

where f_i is the frequency of the mode, d_i is the damping coefficient, a_i is the excited amplitude, and θ_i is the initial phase.

The frequency, damping, and amplitude together define the *feature* ϕ of mode i:

$$\phi_i = (f_i, d_i, a_i) \tag{5}$$

and will be used throughout the rest of the paper. We ignore θ_i in Eqn. 4 because it can be safely assumed as zero in our estimation process, where the object is initially at rest and struck at $t = 0$. f and ω are used interchangeably to represent frequency, where $\omega = 2\pi f$.

Material properties: The values in Eqn. 4 depend on the material properties, the geometry, and the run-time interactions: a_i and θ_i depend on the run-time excitation of the object, while f_i and d_i depend on the geometry and the material properties as shown below. Solving Eqn. 3, we get

$$d_i = \frac{1}{2}(\alpha + \beta\lambda_i), \tag{6}$$

$$f_i = \frac{1}{2\pi}\sqrt{\lambda_i - \left(\frac{\alpha + \beta\lambda_i}{2}\right)^2}. \tag{7}$$

We assume the Rayleigh damping coefficients, α and β, can be transfered to another object with no drastic shape or size change. Empirical experiments were carried out to support this assumption. Please refer to [Ren et al. 2012] for more detail. The eigenvalues λ_i's are calculated from $\mathbf{M}$ and $\mathbf{K}$ and determined by the geometry and tetrahedralization as well as the material properties: in our tetrahedral finite element model, $\mathbf{M}$ and $\mathbf{K}$ depend on mass density ρ, Young's modulus E, and Poisson's ratio ν, if we assume the material is *isotropic* and *homogeneous*.

Constraint for modes: We observe modes in the adopted linear modal synthesis model have to obey some constraint due to its formulation. Because of the Rayleigh damping model we adopted, all estimated modes lie on a circle in the (ω, d)-space, characterized by α and β. This can be shown as follows. Rearranging Eqn. 6 and Eqn. 7 as

$${\omega_i}^2 + \left(d_i - \frac{1}{\beta}\right)^2 = \left(\frac{1}{\beta}\sqrt{1 - \alpha\beta}\right)^2 \tag{8}$$

we see that it takes the form of ${\omega_i}^2 + (d_i - y_c)^2 = R^2$. This describes a circle of radius R centered at $(0, y_c)$ in the (ω, d)-space, where R and y_c depend on α and β. This constraint for modes restricts the model from capturing some sound effects and renders it impossible to make modal synthesis sounds with Rayleigh damping exactly the same as an arbitrary real-world recording. However, if a circle that best represents the recording audio is found, it is possible to preserve the same sense of material as the recording. It is shown in Section 4 and 5.3, how a proposed pipeline achieves this.

4. FEATURE EXTRACTION

An example impact sound can be represented by high-level features collectively.

We first analyze and decompose a given example audio clip into a set of features, which will later be used in the subsequent phases of our pipeline, namely the parameter estimation and residual compensation parts. Next we present the detail of our feature extraction algorithm.

Multi-level power spectrogram representation: As shown in Eqn. 5, the feature of a mode is defined as its frequency, damping, and amplitude. In order to analyze the example audio and extract these feature values, we use a time-varying frequency representation called *power spectrogram*. A power spectrogram $\mathbf{P}$ for a a time domain signal $\mathbf{s}[n]$, is obtained by first breaking it up into overlapping frames, and then performing windowing and Fourier transform on each frame:

$$\mathbf{P}[m,\omega] = \left| \sum_n \mathbf{s}[n]\mathbf{w}[n-m]e^{-j\omega n} \right|^2, \qquad (9)$$

where $\mathbf{w}$ is the window applied to the original time domain signal [Oppenheim et al. 1989]. The power spectrogram records the signal's power spectral density within a *frequency bin* centered around $\omega = 2\pi f$ and a *time frame* defined by m.

When computing the power spectrogram for a given sound clip, one can choose the resolutions of the time or frequency axes by adjusting the length of the window $\mathbf{w}$. Choosing the resolution in one dimension, however, automatically determines the resolution in the other dimension. A high frequency resolution results in a low temporal resolution, and vice versa.

To fully accommodate the range of frequency and damping for all the modes of an example audio, we compute multiple levels of power spectrograms, with each level doubling the frequency resolution of the previous one and halving the temporal resolution. Therefore, for each mode to be extracted, a suitable level of power spectrogram can be chosen first, depending on the time and frequency characteristics of the mode.

Global-to-local scheme: After computing a set of multi-level power spectrograms for a recorded example audio, we *globally* search through all levels for peaks (local maxima) along the frequency axis. These peaks indicate the frequencies where potential modes are located, some of which may appear in multiple levels. At this step the knowledge of frequency is limited by the frequency resolution of the level of power spectrogram. For example, in the level where the window size is 512 points, the frequency resolution is as coarse as 86 Hz. A more accurate estimate of the frequency as well as the damping value is obtained by performing a *local shape fitting* around the peak.

The power spectrogram of a damped sinusoid has a 'hill' shape, similar to the blue surface shown in Fig. 3b. The actual shape contains information of the damped sinusoid: the position and height of the peak are respectively determined by the frequency and amplitude, while the slope along the time axis and the width along the frequency axis are determined by the damping value. For a potential mode, a damped sinusoid with the initial guess of (f, d, a) is synthesized and added to the sound clip consisting of all the modes collected so far. The power spectrogram of the resulting sound clip is computed (shown as the red hill shape in Fig. 3b), and compared locally with that of the recorded audio (the blue hill shape in Fig. 3b)). An optimizer then searches in the continuous (f, d, a)-space to minimize the difference and acquire a refined estimate of the frequency, damping, and amplitude of the mode at question. Fig. 3 illustrates this process.

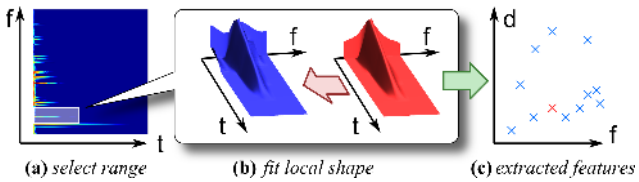


Fig. 3: Feature extraction from a power spectrogram. (a) A peak is detected in a power spectrogram at the location of a potential mode. f=frequency, t=time. (b) A local shape fitting of the power spectrogram is performed to estimate the frequency, damping and amplitude of the potential mode. (c) If the fitting error is below a certain threshold, we collect it in the set of extracted features, shown as the red cross in the feature space. (Only the frequency f and damping d are shown here.)

The local shape fittings for all potential modes are performed in a greedy manner. Among all peaks in all levels, the algorithm starts with the one having the highest average power spectral density. If the shape fitting error computed is above a predefined threshold, we conclude that this level of power spectrogram is not sufficient in capturing the feature characteristics and thereby discard the result; otherwise the feature of the mode is collected. In other words, the most suitable time-frequency resolution (level) for a mode with a particular frequency is not predetermined, but dynamically searched for. Similar approaches have been proposed to analyze the sinusoids in an audio clip in a multi-resolution manner (e.g. Levine et al. [1998], where the time-frequency regions' power spectrogram resolution is predetermined).

We have tested the accuracy of our feature extraction with 100 synthetic sinusoids with frequencies and damping values randomly drawn from $[0, 22050.0]$(Hz) and $[0.1, 1000](s^{-1})$ respectively. The average relative error is 0.040% for frequencies and 0.53% for damping values, which are sufficient for our framework.

Comparison with existing methods: The SMS method [Serra and Smith III 1990] is also capable of estimating information of modes. From a power spectrogram, it tracks the amplitude envelope of each peak over time, and a similar method is adopted by Lloyd et al. [2011]. Unlike our algorithm, which fits the entire local hill shape, they only track a single peak value per time frame. In the case where the mode's damping is high or the signal's background is noisy, this method yields high error.

Another feature extraction technique was proposed by Pai et al. [2001] and Corbett et al. [2007]. The method is known for its ability to separate modes within one frequency bin. In our framework, however, the features are only used to guide the subsequent parameter estimation process, which is not affected much by replacing two nearly duplicate features with one. Our method also offers some advantages and achieves higher accuracy in some cases compared with theirs. First, our proposed greedy approach is able to reduce the interference caused by high energy neighboring modes. Secondly, these earlier methods use a fixed frequency-time resolution that is not necessarily the most suitable for extracting all modes, while our method selects the appropriate resolution dynamically.

The detailed comparisons and data can be found in Appendix A.

5. PARAMETER ESTIMATION

Using the extracted features (Sec. 4) and psychoacoustic principles (as described in this section), we introduce a parameter estimation algorithm based on an optimization framework for sound synthesis.

5.1 An Optimization Framework

We now describe the optimization work flow for estimating material parameters for sound synthesis. In the rest of the paper, all data related to the example audio recordings are called *reference* data; all data related to the virtual object (which are used to estimate the material parameters) are called *estimated* data, and are denoted with a tilde, e.g. $\tilde{f}$.

Reference sound and features: The *reference sound* is the example recorded audio, which can be expressed as a time domain signal $\mathbf{s}[n]$. The *reference features* $\mathbf{\Phi} = \{\phi_i\} = \{(f_i, d_i, a_i)\}$ are the features extracted from the reference sound, as described in Sec. 4.

Estimated sound and features: In order to compute the *estimated sound* $\tilde{\mathbf{s}}[n]$ and *estimated features* $\tilde{\mathbf{\Phi}} = \{\tilde{\phi}_j\} = \{(\tilde{f}_j, \tilde{d}_j, \tilde{a}_j)\}$, we first create a virtual object that is roughly the same size and geometry as the real-world object whose impact sound was recorded. We then tetrahedralize it and calculate its mass matrix $\mathbf{M}$ and stiffness matrix $\mathbf{K}$. As mentioned in Sec. 3, we assume the material is isotropic and homogeneous. Therefore, the initial $\mathbf{M}$ and $\mathbf{K}$ can be found using the finite element method, by assuming some initial values for the Young's modulus, mass density, and Poisson's ratio, E_0, ρ_0, and ν_0. The assumed eigenvalues λ_i^0's can thereby be computed. For computational efficiency, we make a further simplification that the Poisson's ratio is held as constant. Then the eigenvalue λ_i for general E and ρ is just a multiple of λ_i^0:

$$\lambda_i = \frac{\gamma}{\gamma_0}\lambda_i^0 \tag{10}$$

where $\gamma = E/\rho$ is the ratio of Young's modulus to density, and $\gamma_0 = E_0/\rho_0$ is the ratio using the assumed values.

We then apply a unit impulse on the virtual object at a point corresponding to the actual impact point in the example recording, which gives an excitation pattern of the eigenvalues as Eqn. 4. We denote the excitation amplitude of mode j as a_j^0. The superscript 0 notes that it is the response of a unit impulse; if the impulse is not unit, then the excitation amplitude is just scaled by a factor σ,

$$a_j = \sigma a_j^0 \tag{11}$$

Combining Eqn. 6, Eqn. 7, Eqn.10, and Eqn.11, we obtain a mapping from an assumed eigenvalue and its excitation (λ_j^0, a_j^0) to an estimated mode with frequency $\tilde{f}_j$, damping $\tilde{d}_j$, and amplitude $\tilde{a}_j$:

$$(\lambda_j^0, a_j^0) \xrightarrow{\{\alpha,\beta,\gamma,\sigma\}} (\tilde{f}_j, \tilde{d}_j, \tilde{a}_j). \tag{12}$$

The estimated sound $\tilde{\mathbf{s}}[n]$, is thereby generated by mixing all the estimated modes,

$$\tilde{\mathbf{s}}[n] = \sum_j \left(\tilde{a}_j e^{-\tilde{d}_j (n/F_s)} \sin(2\pi \tilde{f}_j (n/F_s))\right) \tag{13}$$

where F_s is the sampling rate.

Difference metric: The estimated sound $\tilde{\mathbf{s}}[n]$ and features $\tilde{\mathbf{\Phi}}$ can then be compared against the reference sound $\mathbf{s}[n]$ and features $\mathbf{\Phi}$, and a difference metric can be computed. If such difference metric function is denoted by Π, the problem of parameter estimation becomes finding

$$\{\alpha, \beta, \gamma, \sigma\} = \underset{\{\alpha,\beta,\gamma,\sigma\}}{\arg\min}\, \Pi. \tag{14}$$

An optimization process is used to find such parameter set. The most challenging part of our work is to find a suitable metric function that can truly reflect what we view as the difference. Next we discuss the details about the metric design in Sec. 5.2 and the optimization process in Sec. 5.3.

5.2 Metric

Given an impact sound of a real-world object, the goal is to find a set of material parameters such that when they are applied to a virtual object of the same size and shape, the synthesized sounds have the similar auditory perception as the original recorded sounding object. By further varying the size, geometry, and the impact points of the virtual object, the intrinsic 'audio signature' of each material for the synthesized sound clips should closely resemble that of the original recording. These are the two criteria guiding the estimation of material parameters based on an example audio clip:

(1) the perceptual similarity of two sound clips;

(2) the audio material resemblance of two generic objects.

The perceptual similarity of sound clips can be evaluated by an 'image domain metric' quantified using the power spectrogram; while the audio material resemblance is best measured by a 'feature domain metric' – both will be defined below,

Image domain metric: Given a reference sound $\mathbf{s}[n]$ and an estimated sound $\tilde{\mathbf{s}}[n]$, their power spectrograms are computed using Eqn. 9 and denoted as two 2D images: $\mathbf{I} = \mathbf{P}[m, \omega]$, $\tilde{\mathbf{I}} = \tilde{\mathbf{P}}[m, \omega]$. An image domain metric can then be expressed as

$$\Pi_{\textit{image}}(\mathbf{I}, \tilde{\mathbf{I}}). \tag{15}$$

Our goal is to find an estimated image $\tilde{\mathbf{I}}$ that minimizes a given image domain metric. This process is equivalent to image registration in computer vision and medical imaging.

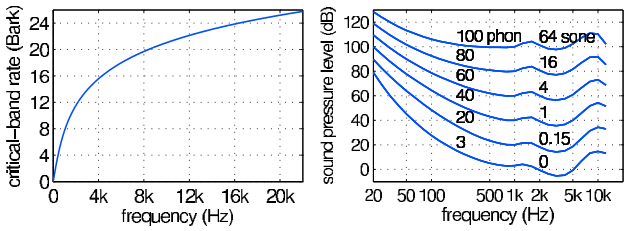
Feature domain metric: A feature $\phi_i = (f_i, d_i, a_i)$ is essentially a three dimensional point. As established in Sec. 3, the set of features of a sounding object is closely related to the material properties of that object. Therefore a metric defined in the feature space is useful in measuring the audio material resemblance of two objects. In other words, a good estimate of material parameters should map the eigenvalues of the virtual object to similar modes as that of the real object. A feature domain metric can be written as

$$\Pi_{\textit{feature}}(\mathbf{\Phi}, \tilde{\mathbf{\Phi}}) \tag{16}$$

and the process of finding the minimum can be viewed as a point set matching problem in computer vision.

Hybrid metric: Both the auditory perceptual similarity and audio material resemblance would need to be considered for a generalized framework, in order to extract and transfer material parameters for modal sound synthesis using a recorded example to guide the automatic selection of material parameters. Therefore, we propose a novel 'hybrid' metric that takes into account of both:

$$\Pi_{\textit{hybrid}}(\mathbf{I}, \mathbf{\Phi}, \tilde{\mathbf{I}}, \tilde{\mathbf{\Phi}}). \tag{17}$$



(a) (b)

Fig. 4: Psychoacoustics related values: (a) the relationship between critical-band rate (in Bark) and frequency (in Hz); (b) the relationship between loudness level L_N (in phon), loudness L (in sone), and sound pressure level L_p (in dB). Each curve is an *equal-loudness contour*, where a constant loudness is perceived for pure steady tones with various frequencies.

Next, we provide details on how we design and compute these metrics.

5.2.1 *Image Domain Metric.* Given two power spectrogram images $\mathbf{I}$ and $\tilde{\mathbf{I}}$, a naive metric can be defined as their squared difference: $\Pi_{image}(\mathbf{I}, \tilde{\mathbf{I}}) = \sum_{m,\omega} \left(\mathbf{P}[m,\omega] - \tilde{\mathbf{P}}[m,\omega]\right)^2$. There are, however, several problems with this metric. The frequency resolution is uniform across the spectrum, and the intensity is uniformly weighted. As humans, however, we distinguish lower frequencies better than the higher frequencies, and mid-frequency signals appear louder than extremely low or high frequencies [Zwicker and Fastl 1999]. Therefore, directly taking squared difference of power spectrograms overemphasizes the frequency differences in the high-frequency components and the intensity differences near both ends of the audible frequency range. It is necessary to apply both *frequency* and *intensity* transformations before computing the image domain metric. We design these transformations based on psychoacoustic principles [Zwicker and Fastl 1999].

Frequency transformation: Studies in psychoacoustics suggested that humans have a limited capacity to discriminate between nearby frequencies, i.e. a frequency f_1 is not distinguishable from f_2 if f_2 is within $f_1 \pm \Delta f$. The indistinguishable range Δf is itself a function of frequency, for example, the higher the frequency, the larger the indistinguishable range. To factor out this variation in Δf a different frequency representation, called *critical-band rate* z, has been introduced in psychoacoustics. The unit for z is *Bark*, and it has the advantage that while Δf is a function of f (measured in Hz), it is constant when measured in Barks. Therefore, by transforming the frequency dimension of a power spectrogram from f to z, we obtain an image that is weighted according to human's perceptual frequency differences. Fig. 4a shows the relationship between critical-band rate z and frequency f, $z = Z(f)$.

Intensity transformation: Sound can be described as the variation of pressure, $p(t)$, and human auditory system has a high dynamical range, from 10^{-5} Pa (threshold of hearing) to 10^2 Pa (threshold of pain). In order to cope with such a broad range, the *sound pressure level* is normally used. For a sound with pressure p, its sound pressure level L_p in decibel (abbreviated to dB-SPL) is defined as

$$L_p = 20 \log(p/p_0), \tag{18}$$

where p_0 is a standard reference pressure. While L_p is just a physical value, *loudness* L is a perceptual value, which measures human sensation of sound intensity. In between, *loudness level* L_N relates the physical value to human sensation. Loudness level of a sound is defined as the sound pressure level of a 1-kHz tone that is perceived as loud as the sound. Its unit is *phon*, and is calibrated such that a sound with loudness level of 40 phon is as loud as a 1-kHz tone at 40 dB-SPL. Finally, loudness L is computed from loudness level. Its unit is *sone*, and is defined such that a sound of 40 phon is 1 sone; a sound twice as loud is 2 sone, and so on.

Fig. 4b shows the relationship between sound pressure level L_p, loudness level L_N and loudness L according to the international standard [ISO 2003]. The curves are *equal-loudness contours*, which are defined such that for different frequency f and sound pressure level L_p, the perceived loudness level L_N and loudness L is constant along each equal-loudness contour. Therefore the loudness of a signal with a specific frequency f and sound pressure level L_p can be calculated by finding the equal-loudness contour passing (f, L_p).

There are other psychoacoustic factors that can affect the human sensation of sound intensity. For example, van den Doel et al. [van den Doel and Pai 2002; van den Doel et al. 2004] considered the 'masking' effect, which describes the change of audible threshold in the presence of multiple stimuli, or modes in this case. However, they did not handle the loudness transform above the audible threshold, which is critical in our perceptual metric. Similar to the work by van den Doel and Pai [1998], we have ignored the masking effect.

Psychoacoustic metric: After transforming the frequency f (or equivalently, ω) to the critical-band rate z and mapping the intensity to loudness, we obtain a transformed image $\mathbf{T}(\mathbf{I}) = \mathbf{T}(\mathbf{I})[m, z]$. Different representations of a sound signal is shown in Fig. 5. Then we can define a psychoacoustic image domain metric as

$$\Pi_{psycho}(\mathbf{I}, \tilde{\mathbf{I}}) = \sum_{m,z} \left(\mathbf{T}(\mathbf{I})[m,z] - \mathbf{T}(\tilde{\mathbf{I}})[m,z]\right)^2 \tag{19}$$

Similar transformations and distance measures have also been used to estimate the perceived resemblance between music pieces [Morchen et al. 2006; Pampalk et al. 2002].

5.2.2 *Feature Domain Metric.* As shown in Eqn. 8, in the (ω, d)-space, modes under the assumption of Rayleigh damping lie on a circle determined by damping parameters α and β, while features extracted from example recordings can be anywhere. Therefore, it is challenging to find a good match between the reference features Φ and estimated features $\tilde{\Phi}$. Fig. 6a shows a typical matching in the (f, d)-space. Next we present a feature domain metric that evaluates such a match.

In order to compute the feature domain metric, we first transform the frequency and damping of feature points to another different 2D space. Namely, from (f_i, d_i) to (x_i, y_i), where $x_i = \mathrm{X}(f_i)$ and $y_i = \mathrm{Y}(d_i)$ encode the frequency and damping information respectively. With suitable transformations, the Euclidean distance defined in the transformed space can be more useful and meaningful for representing the perceptual difference. The distance between two feature points is thus written as

$$D(\phi_i, \tilde{\phi}_j) \equiv \left\| \left(\mathrm{X}(f_i), \mathrm{Y}(d_i)\right) - \left(\mathrm{X}(\tilde{f}_j), \mathrm{Y}(\tilde{d}_j)\right) \right\|. \tag{20}$$

Frequency and damping are key factors in determining material agreement, while amplitude indicates relative importance of modes. That is why we measure the distance between two feature

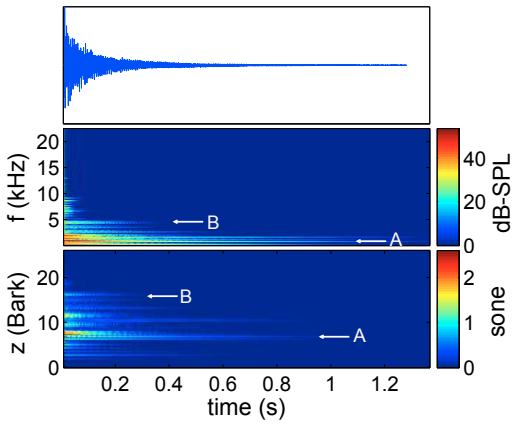


Fig. 5: Different representation of a sound clip. Top: time domain signal $\mathbf{s}[\mathbf{n}]$. Middle: original image, power spectrogram $P[m, \omega]$ with intensity measured in dB. Bottom: image transformed based on psychoacoustic principles. The frequency f is transformed to *critical-band rate* z, and the intensity is transformed to *loudness*. Two pairs of corresponding modes are marked as A and B. It can be seen that the frequency resolution decreases toward the high frequencies, while the signal intensities in both the higher- and lower-end of the spectrum are de-emphasized.

points in the 2D (f, d)-space and use amplitude to weigh that distance.

For frequency, as described in Sec. 5.2.1 we know that the frequency resolution of human is constant when expressed as critical-band rate and measured in Barks: $\Delta f(f) \propto \Delta z$. Therefore it is a suitable frequency transformation

$$X(f) = c_z Z(f) \tag{21}$$

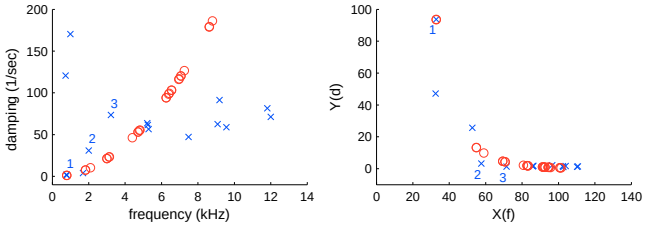
where c_z is some constant coefficient.

For damping, although human can roughly sense that one mode damps faster than another, directly taking the difference in damping value d is not feasible. This is due to the fact that humans cannot distinguish between extremely short bursts [Zwicker and Fastl 1999]. For a damped sinusoid, the inverse of the damping value, $1/d_i$, is proportional to its duration, and equals to how long before the signal decays to e^{-1} of its initial amplitude. While distance measured in damping values overemphasizes the difference between signals with high d values (corresponding to short bursts), distance measured in durations does not. Therefore

$$Y(d) = c_d \frac{1}{d} \tag{22}$$

(where c_d is some constant coefficient) is a good choice of damping transformation. The reference and estimated features of data in Fig. 6a are shown in the transformed space in Fig. 6b.

Having defined the transformed space, we then look for matching the reference and estimated feature points in this space. Our matching problem belongs to the category where there is no known correspondence, i.e. no prior knowledge about which point in one set should be matched to which point in another. Furthermore, because there may be several estimated feature points in the neighborhood of a reference point or vice versa, the matching is not necessarily a one-to-one relationship. There is also no guarantee that an exact matching exist, because (1) the recorded material may not obey the Rayleigh damping model, (2) the discretization of the virtual object and the assumed hit point may not give the exact eigenvalues and excitation pattern of the real object. Therefore we are merely looking for a partial, approximate matching.



(a) (b)

Fig. 6: Point set matching problem in the feature domain: (a) in the original frequency and damping, (f, d)-space. (b) in the transformed, (x, y)-space, where $x = X(f)$ and $y = Y(d)$. The blue crosses and red circles are the reference and estimated feature points respectively. The three features having the largest energies are labeled 1, 2, and 3.

The simplest point-based matching algorithm that solves problems in this category (i.e. partial, approximate matching without known correspondence) is Iterative Closest Points. It does not work well, however, when there is a significant number of feature points that cannot be matched [Besl and McKay 1992], which is possibly the case in our problem. Therefore, we define a metric, *Match Ratio Product*, that meets our need and is discussed next.

For a reference feature point set $\mathbf{\Phi}$, we define a *match ratio* that measures how well they are matched by an estimated feature point set $\tilde{\mathbf{\Phi}}$. This *set-to-set* match ratio, defined as

$$R(\mathbf{\Phi}, \tilde{\mathbf{\Phi}}) = \frac{\sum_i w_i R(\phi_i, \tilde{\mathbf{\Phi}})}{\sum_i w_i}, \tag{23}$$

is a weighted average of the *point-to-set* match ratios, which are in turn defined as

$$R(\phi_i, \tilde{\mathbf{\Phi}}) = \frac{\sum_j \tilde{u}_{ij} k(\phi_i, \tilde{\phi}_j)}{\sum_j \tilde{u}_{ij}}, \tag{24}$$

a weighted average of the *point-to-point* match scores $k(\phi_i, \tilde{\phi}_j)$. The point-to-point match score $k(\phi_i, \tilde{\phi}_j)$, which is directly related to the distance of feature points (Eqn. 20), should be designed to give values in the continuous range $[0, 1]$, with 1 meaning that the two points coincide, and 0 meaning that they are too far apart. Similarly $R(\phi_i, \tilde{\mathbf{\Phi}}) = 1$ when ϕ_i coincides with an estimated feature point, and $R(\mathbf{\Phi}, \tilde{\mathbf{\Phi}}) = 1$ when all reference feature points are perfectly matched. The weight w_i and $\tilde{u}_{ij}$ in Eqn. 23 and Eqn. 24 are used to adjust the influence of each mode. The match ratio for the estimated feature points, $\tilde{R}$, is defined analogously

$$\tilde{R}(\mathbf{\Phi}, \tilde{\mathbf{\Phi}}) = \frac{\sum_j \tilde{w}_j R(\tilde{\phi}_j, \mathbf{\Phi})}{\sum_i \tilde{w}_j} \tag{25}$$

The match ratios for the reference and the estimated feature point sets are then combined to form the *Match Ratio Product* (MRP), which measures how well the reference and estimated feature point sets match with each other,

$$\Pi_{MRP}(\mathbf{\Phi}, \tilde{\mathbf{\Phi}}) = -R\tilde{R}. \tag{26}$$

The negative sign is to comply with the minimization framework. Multiplying the two ratios penalizes the extreme case where either one of them is close to zero (indicating poor matching).

The normalization processes in Eqn. 23 and Eqn. 25 are necessary. Notice that the denominator in Eqn. 25 is related to the number of estimated feature points inside the audible range, $\tilde{N}_{\text{audible}}$ (in fact $\sum_j \tilde{w}_j = \tilde{N}_{\text{audible}}$ if all $\tilde{w}_j = 1$). Depending on the set of parameters, $\tilde{N}_{\text{audible}}$ can vary from a few to thousands. Factoring out $\tilde{N}_{\text{audible}}$ prevents the optimizer from blindly introducing more modes into the audible range, which may increase the absolute number of matched feature points, but may not necessarily increase the match ratios. Such averaging techniques have also been employed to improve the robustness and discrimination power of point-based object matching methods [Dubuisson and Jain 1994; Gope and Kehtarnavaz 2007].

In practice, the weights w's and u's, can be assigned according to the relative energy or perceptual importance of the modes. The point-to-point match score $k(\phi_i, \tilde{\phi}_j)$, can also be tailored to meet different needs. The constants and function forms used in this section are listed in Appendix B.

5.2.3 *Hybrid Metric.* Finally, we combine the strengths from both image and feature domain metrics by defining the following hybrid metric:

$$\Pi_{hybrid} = \frac{\Pi_{psycho}}{|\Pi_{MRP}|}. \tag{27}$$

This metric essentially weights the perceptual similarity with how well the features match, and by making the match ratio product as the denominator, we ensure that a bad match (low MRP) will boost the metric value and is therefore highly undesirable.

5.3 Optimizer

We use the Nelder-Mead method [Lagarias et al. 1999] to minimize Eqn. 14, which may converge into one of the many local minima. We address this issue by starting the optimizer from many starting points, generated based on the following observations.

First, as elaborated by Eqn. 8 in Sec. 3, the estimated modes are constrained by a circle in the (ω, d)-space. Secondly, although there are many reference modes, they are not evenly excited by a given impact– we observe that usually the energy is mostly concentrated in a few dominant ones. Therefore, a good estimate of α and β must define a circle that passes through the neighborhood of these dominant reference feature points. We also observe that in order to yield a low metric value, there must be at least one dominant estimated mode at the frequency of the *most* dominant reference mode.

We thereby generate our starting points by first drawing two dominant reference feature points from a total of $N_{dominant}$ of them, and find the circle passing through these two points. This circle is potentially a 'good' circle, from which we can deduce a starting estimate of α and β using Eqn. 8. We then collect a set of eigenvalues and amplitudes (defined in Sec. 5.1) $\{(\lambda_j^0, a_j^0)\}$, such that there does not exist any (λ_k^0, a_k^0) that simultaneously satisfies $\lambda_k^0 < \lambda_j^0$ and $a_k^0 > a_j^0$. It can be verified that the estimated modes mapped from this set always includes the one with the highest energy, for any mapping parameters $\{\alpha, \beta, \gamma, \sigma\}$ used in Eqn. 12. Each (λ_j^0, a_j^0) in this set is then mapped and aligned to the frequency of the most dominant reference feature point, and its amplitude is adjusted to be identical as the latter. This step gives a starting estimate of γ and σ. Each set of $\{\alpha, \beta, \gamma, \sigma\}$ computed in this manner is a starting point, and may lead to a different local minimum. We choose the set which results in the lowest metric value to be our estimated parameters. Although there is no guarantee that a global minimum will be met, we find that the results produced with this strategy are satisfactory in our experiments, as discussed in Sec. 7.

6. RESIDUAL COMPENSATION

With the optimization proposed in Sec. 5, a set of parameters that describe the material of a given sounding object can be estimated, and the produced sound bears a close resemblance of the material used in the given example audio. However, linear modal synthesis alone is not capable of synthesizing sounds that are as rich and realistic as many real-world recordings. Firstly, during the short period of contact, not all energy is transformed into stable vibration that can be represented with a small number of damped sinusoids, or modes. The stochastic and transient nature of the non-modal components makes sounds in nature rich and varying. Secondly, as discussed in Sec. 3, not all features can be captured due to the constraints for modes in the synthesis model. In this section we present a method to account for the *residual*, which approximates the difference between the real-world recordings and the modal synthesis sounds. In addition, we propose a technique for transferring the residual with geometry and interaction variation. With the residual computation and transfer algorithms introduced below, more realistic sounds that automatically vary with geometries and hitting points can be generated with a small computation overhead.

6.1 Residual Computation

In this section we discuss how to compute the residual from the recorded sound and the synthesized modal sound generated with the estimated parameters.

Previous works have also looked into capturing the difference between a source audio and its modal component [Serra and Smith III 1990; Serra 1997; Lloyd et al. 2011]. In these works, the modal part is directly tracked from the original audio, so the residual can be calculated by a straightforward subtraction of the power spectrograms. The synthesized modal sound in our framework, however, is generated solely from the estimated material parameters. Although it preserves the intrinsic quality of the recorded material, in general the modes in our synthesized sounds are not perfectly aligned with the recorded audio. An example is shown in Fig. 7a and Fig. 7c. It is due to the constraints in our sound synthesis model and discrepancy between the discretized virtual geometries and the real-world sounding objects. As a result, direct subtraction does not work in this case to generate a reasonable residual. Instead, we first compute an intermediate data, called the *represented sound*. It corresponds to the part in the recorded sound that is captured, or represented, by our synthesized sound. This represented sound (Fig. 7d) can be directly subtracted from the recorded sound to compute the residual (Fig. 7e).

The computation of the represented sound is based on the following observations. Consider a feature (described by ϕ_i) extracted from the recorded audio. If it is perfectly captured by the estimated modes, then it should not be included in the residual and should be completely subtracted from the recorded sound. If it is not captured

at all, it should not be subtracted from the recorded sound, and if it is approximated by an estimated mode, it should be partially subtracted. Since features closely represent the original audio, they can be directly subtracted from the recorded sound.

The point-to-set match ratio $R(\phi_i, \tilde{\mathbf{\Phi}})$ proposed in Sec. 5.2 essentially measures how well a reference feature ϕ_i is represented (matched) by all the estimated modes. This match ratio can be conveniently used to determine how much of the corresponding feature should be subtracted from the recording.

The represented sound is therefore obtained by adding up all the reference features that are respectively weighted by the match ratio of the estimated modes. And the power spectrogram of the residual is obtained by subtracting the power spectrogram of the represented sound from that of the recorded sound. Fig. 7 illustrates the residual computation process.

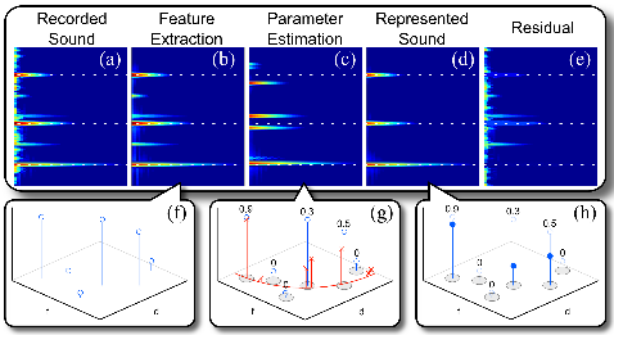


Fig. 7: Residual computation. From a recorded sound (a), the reference features are extracted (b), with frequencies, dampings, and energies depicted as the blue circles in (f). After parameter estimation, the synthesized sound is generated (c), with the estimated features shown as the red crosses in (g), which all lie on a curve in the (f, d)-plane. Each reference feature may be approximated by one or more estimated features, and its match ratio number is shown. The represented sound is the summation of the reference features weighted by their match ratios, shown as the solid blue circles in (h). Finally, the difference between the recorded sound's power spectrogram (a) and the represented sound's (d) are computed to obtain the residual (e).

6.2 Residual Transfer

Residual of one particular instance (i.e. one geometry and one hit point) can be obtained through the above described residual computation method. However, when synthesizing sounds for a different geometry undergoing different interaction with other rigid bodies, the residual audio needs to vary accordingly. Lloyd et al. [2011] proposed applying a random dip filter on the residual to provide variation. While this offers an attractive solution for quickly generating modified residual sound, it does not transfer accordingly with the geometry change or the dynamics of the sounding object.

6.2.1 *Algorithm.* As discussed in previous sections, *modes* transfer naturally with geometries in the modal analysis process, and they respond to excitations at runtime in a physical manner. In other words, the modal component of the synthesized sounds already provides transferability of sounds due to varying geometries and dynamics. Hence, we compute the transferred residual under the guidance of modes as follows.

Given a source geometry and impact point, we know how to transform its modal sound to a target geometry and impact points. Equivalently, we can describe such transformation as acting on the power spectrograms, transforming the modal power spectrogram of the source, $\mathbf{P}^s_{modal}$, to that of the target, $\mathbf{P}^t_{modal}$:

$$\mathbf{P}^s_{modal} \xrightarrow{H} \mathbf{P}^t_{modal} \tag{28}$$

where H is the transform function. We apply the same transform function H to the *residual* power spectrograms

$$\mathbf{P}^s_{residual} \xrightarrow{H} \mathbf{P}^t_{residual} \tag{29}$$

where the source residual power spectrogram is computed as described in Sec. 6.1.

More specifically, H can be decomposed into per-mode transform functions, $H_{i,j}$, which transforms the power spectrogram of a source mode $\phi^s_i = (f^s_i, d^s_i, a^s_i)$ to a target mode $\phi^t_j = (f^t_j, d^t_j, a^t_j)$. $H_{i,j}$ can further be described as a series of operations on the source power spectrogram $\mathbf{P}^s_{modal}$: (1) the center frequency is shifted from f^s_i to f^t_j; (2) the time dimension is stretched according to the ratio between d^s_i and d^t_j; (3) the height (intensity) is scaled pixel-by-pixel to match $\mathbf{P}^t_{modal}$. The per-mode transform is performed in the neighborhood of f^s_i, namely between $\frac{1}{2}(f^s_{i-1} + f^s_i)$ and $\frac{1}{2}(f^s_i + f^s_{i+1})$, to that of f^t_j, namely between $\frac{1}{2}(f^t_{j-1} + f^t_j)$ and $\frac{1}{2}(f^t_j + f^t_{j+1})$.

The per-mode transform is performed for all pairs of source and target modes, and the local residual power spectrograms are 'stitched' together to form the complete $\mathbf{P}^t_{residual}$. Finally, the time-domain signal of the residual is reconstructed from $\mathbf{P}^t_{residual}$, using an iterative inverse STFT algorithm by Griffin and Lim [2003]. Algorithm 1 shows the complete feature-guided residual transfer algorithm. With this scheme, the transform of the residual power spectrogram is completely guided by the appropriate transform of modes. The resulting residual changes consistently with the modal sound. Since the modes transform with the geometry and dynamics in a physical manner, the transferred residual also faithfully reflects this variation.

Algorithm 1: Residual Transformation at Runtime

Input: *source* modes $\mathbf{\Phi}^s = \{\phi^s_i\}$, *target* modes $\mathbf{\Phi}^t = \{\phi^t_j\}$, and source residual audio $\mathbf{s}^s_{residual}[n]$
Output: target residual audio $\mathbf{s}^t_{residual}[n]$
$\mathbf{\Psi} \leftarrow$ DetermineModePairs($\mathbf{\Phi}^s$, $\mathbf{\Phi}^t$)
foreach *mode pair* $(\phi^s_k, \phi^t_k) \in \mathbf{\Psi}$ **do**
  $\mathbf{P}^{s\prime} \leftarrow$ ShiftSpectrogram($\mathbf{P}^s$, Δfrequency)
  $\mathbf{P}^{s\prime\prime} \leftarrow$ StretchSpectrogram($\mathbf{P}^{s\prime}$, damping_ratio)
  $\mathbf{A} \leftarrow$ FindPixelScale($\mathbf{P}^t$, $\mathbf{P}^{s\prime\prime}$)
  $\mathbf{P}^s_{residual}{}' \leftarrow$ ShiftSpectrogram($\mathbf{P}^s_{residual}$, Δfrequency)
  $\mathbf{P}^s_{residual}{}'' \leftarrow$ StretchSpectrogram($\mathbf{P}^s_{residual}{}'$, damping_ratio)
  $\mathbf{P}^t_{residual}{}'' \leftarrow$ MultiplyPixelScale($\mathbf{P}^s_{residual}{}''$, $\mathbf{A}$)
  $(\omega_{start}, \omega_{end}) \leftarrow$ FindFrequencyRange(ϕ^t_{k-1}, ϕ^t_k)
  $\mathbf{P}^t_{residual}$ [m, $\omega_{start}, \ldots, \omega_{end}$] $\leftarrow \mathbf{P}^t_{residual}{}''$ [m, $\omega_{start}, \ldots, \omega_{end}$]
end
$\mathbf{s}^t_{residual}[n] \leftarrow$ IterativeInverseSTFT($\mathbf{P}^t_{residual}$)

Note that a 'one-to-one mapping' between the source and target modes is required. If the target geometry is a scaled version of the source geometry, then there is a natural correspondence between

the modes. If the target geometry, however, is of different shape from the source one, such natural correspondence does not exist. In this case, we pick the top $N_{dominant}$ modes with largest energies from both sides, and pair them from low frequency to high frequency.

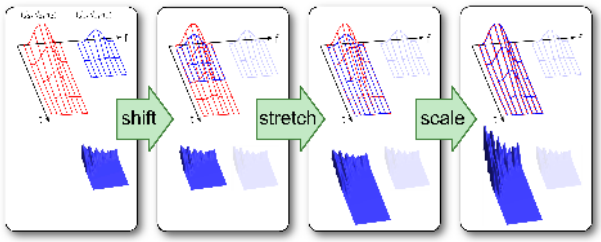


Fig. 8: Single mode residual transform: The power spectrogram of a source mode (f_1, d_1, a_1) (the blue wireframe), is transformed to a target mode (f_2, d_2, a_2) (the red wireframe), through frequency-shifting, time-stretching, and height-scaling. The residual power spectrogram (the blue surface at the bottom) is transformed in the exact same way.

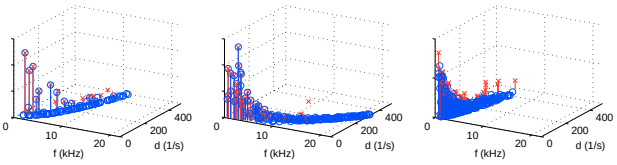
6.2.2 *Implementation and Performance.* The most computation costly part of residual transfer is the iterative inverse STFT process. We are able to obtain acceptable time-domain reconstruction from the power spectrogram when we limit the iteration of inverse STFT to 10. Hardware acceleration is used in our implementation to ensure fast STFT computation. More specifically, CUFFT, a CUDA implementation of Fast Fourier Transform, is adopted for parallelized inverse STFT operations. Also note that residual transfer computation only happens when there is a contact event, the obtained time-domain residual signal can be used until the next event. On an NVIDIA GTX 480 graphics card, if the contact events arrive at intervals around $1/30$s, the residual transfer in the current implementation can be successfully evaluated in time.

7. RESULTS AND ANALYSIS

Parameter estimation: Before working on real-world recordings, we design an experiment to evaluate the effectiveness of our parameter estimation with synthetic sound clips. A virtual object with known material parameters $\{\alpha, \beta, \gamma, \sigma\}$ and geometry is struck, and a sound clip is synthesized by mixing the excited modes. The sound clip is entered to the parameter estimation pipeline to test if the same parameters are recovered. Three sets of parameters are tested and the results are shown in Fig.9.

This experiment demonstrates that if the material follows the Rayleigh damping model, the proposed framework is capable of estimating the material parameters with high accuracy. Below we will see that real materials do not follow the Rayleigh damping model exactly, but the presented framework is still capable of finding the closest Rayleigh damping material that approximates the given material.

We estimate the material parameters from various real-world audio recordings: a wood plate, a plastic plate, a metal plate, a porcelain plate, and a glass bowl. For each recording, the parameters are estimated using a virtual object that is of the same size and shape as the one used to record the audio clips. When the virtual object is hit at the same location as the real-world object, it produces a



	truth	estimated	relative error
α	9.2003e+1	9.1995e+1	9.31e-5
β	1.8297e-7	1.8299e-7	9.30e-5
γ	3.6791e+0	3.6791e+0	3.91e-6
σ	2.1873e-3	2.1872e-3	5.61e-5

	truth	estimated	relative error
α	3.9074e+0	3.9069e+0	1.27e-4
β	3.3935e-8	3.3935e-8	1.62e-6
γ	3.4186e+0	3.4186e+0	1.17e-6
σ	9.0013e-6	9.0009e-6	4.67e-5

	truth	estimated	relative error
α	3.1425e+1	3.1428e+1	9.93e-5
β	7.0658e-7	7.0663e-7	7.61e-5
γ	7.3953e+0	7.3953e+0	3.00e-6
σ	3.5842e-9	3.5847e-9	1.46e-4

Fig. 9: Results of estimating material parameters using synthetic sound clips. The intermediate results of the feature extraction step are visualized in the plots. Each blue circle represents a synthesized feature, whose coordinates (x, y, z) denote the frequency, damping, and energy of the mode. The red crosses represent the extracted features. The tables show the truth value, estimated value, and relative error for each of the parameters.

Table I. : Estimated parameters

	Parameters			
Material	α	β	γ	σ
Wood	2.1364e+0	3.0828e-6	6.6625e+5	3.3276e-6
Plastic	5.2627e+1	8.7753e-7	8.9008e+4	2.2050e-6
Metal	6.3035e+0	2.1160e-8	4.5935e+5	9.2624e-6
Glass	1.8301e+1	1.4342e-7	2.0282e+5	1.1336e-6
Porcelain	3.7388e-2	8.4142e-8	3.7068e+5	4.3800e-7

Refer to Sec. 3 and Sec. 5 for the definition and estimation of these parameters.

sound similar to the recorded audio, as shown in Fig. 10 and the supplementary video.

Fig. 11 compares the refenece features of the real-world objects and the estimated features of the virtual objects as a result of the parameter estimation. The parameter estimated for these materials are shown in Table. I.

Transfered parameters and residual: The parameters estimated can be transfered to virtual objects with different sizes and shapes. Using these material parameters, a different set of resonance modes can be computed for each of these different objects. The sound synthesized with these modes preserves the intrinsic material quality of the example recording, while naturally reflect the variation in virtual object's size, shape, and interactions in the virtual environment.

Moreover, taking the difference between the recording of the example real object and the synthesized sound from its virtual counterpart, the residual is computed. This residual can also be transfered to other virtual objects, using methods described in Sec. 6.

Fig. 12 gives an example of this transferring process. From an example recording of a porcelain plate (a), the parameters for the porcelain material are estimated, and the residual computed (b). The parameters and residual are then transfered to a smaller porcelain plate (c) and a porcelain bunny (d).

Comparison with real recordings: Fig. 13 shows a comparison of the transferred results with the real recordings. From a recording of glass bowl, the parameters for glass are estimated (column (a)) and transfered to other virtual glass bowls of different sizes. The synthesized sounds ((b) (c) (d), bottom row) are compared with the real-world audio for these different-sized glass bowls ((b) (c) (d),

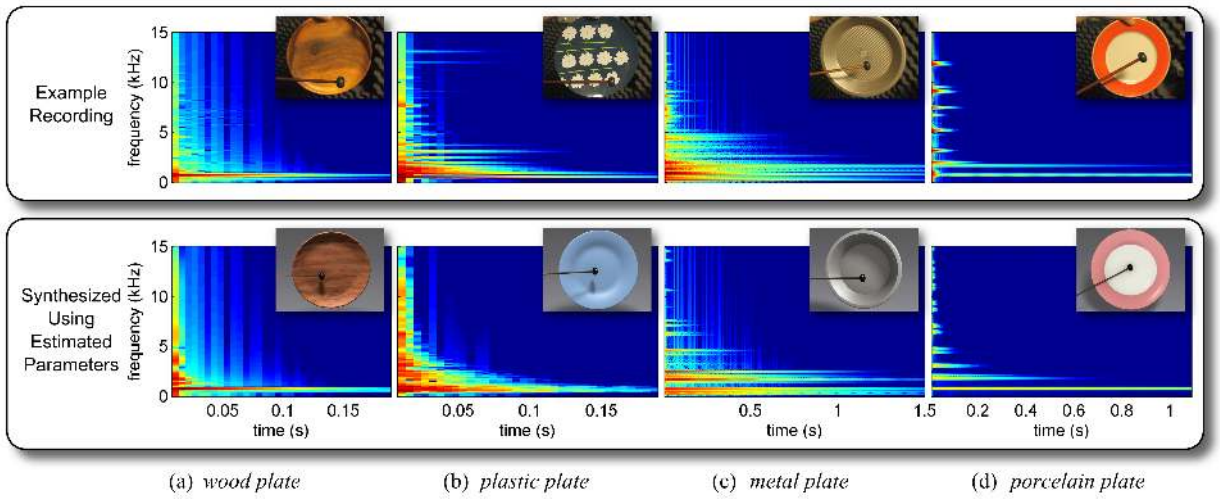
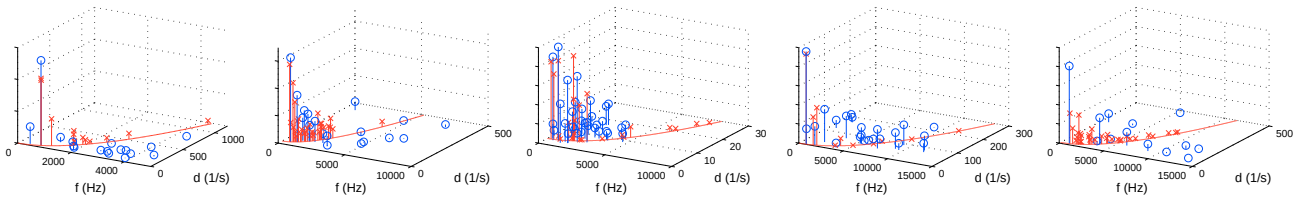


Fig. 10: Parameter estimation for different materials. For each material, the material parameters are estimated using an example recorded audio (top row). Applying the estimated parameters to a virtual object with the same geometry as the real object used in recording the audio will produce a similar sound (bottom row).



(a) wood plate (b) plastic plate (c) metal plate (d) porcelain plate (e) glass bowl

Fig. 11: Feature comparison of real and virtual objects. The blue circles represent the reference features extracted from the recordings of the real objects. The red crosses are the features of the virtual objects using the estimated parameters. Because of the Rayleigh damping model, all the features of a virtual object lie on the depicted red curve on the (f, d)-plane.

top row). It can be seen that although the transfered sounds are not identical to the recorded ones, the overall trend in variation is similar. Moreover, the perception of material is preserved, as can be verified in the accompanying video. More examples of transferring the material parameters as well as the residuals are demonstrated in the accompanying video.

Example: a complicated scenario We applied the estimated parameters for various virtual objects in a scenario where complex interactions take place, as shown in Fig. 14 and the accompanying video.

Performance: Table II shows the timing for our system running on a single core of a 2.80 GHz Intel Xeon X5560 machine. It should be noted that the parameter estimation is an offline process: it needs to be run only once per material, and the result can be stored in a database for future reuse.

For each material in column one, multiple starting points are generated first as described in Sec. 5.3, and the numbers of starting points are shown in column two. From each of these starting points, the optimization process runs for an average number of iterations (column three) until convergence. The average time taken for the process to converge is shown in column four. The convergence is defined as when both the step size and the difference in metric value are lower than their respective tolerance values, Δ_x and Δ_{metric}. The numbers reported in Table II are measured with Δ_x = 1e-4 and Δ_{metric} = 1e-8.

Table II. : Offline Computation for Material Parameter Estimation

Material	#starting points	average #iteration	average time (s)
Wood	60	1011	46.5
Plastic	210	904	49.4
Metal	50	1679	393.5
Porcelain	80	1451	131.3
Glass	190	1156	68.9

8. PERCEPTUAL STUDY

To assess the effectiveness of our parameter estimation algorithm, we designed an experiment to evaluate the auditory perception of the synthesized sounds of five different materials. Each subject is presented with a series of 24 audio clips with no visual image or graphical animation. Among them, 8 are audio recordings of sound

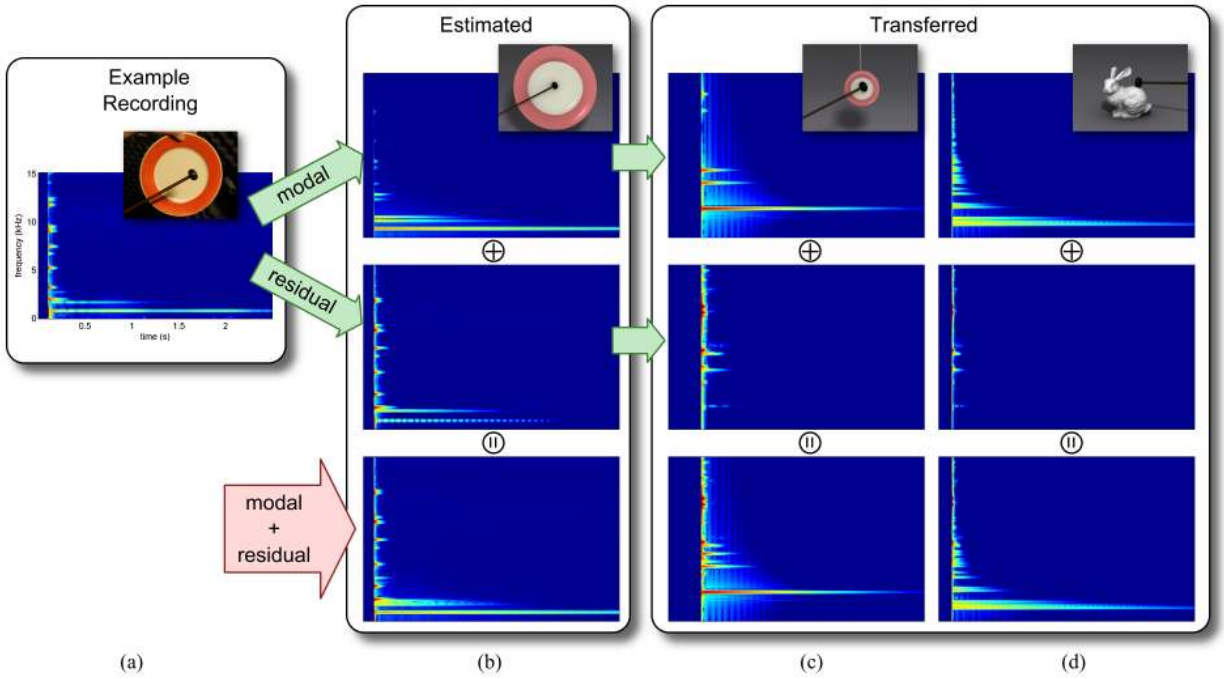


Fig. 12: Transfered material parameters and residual: from a real-world recording (a), the material parameters are estimated and the residual computed (b). The parameters and residual can then be applied to various objects made of the same material, including (c) a smaller object with similar shape; (d) an object with different geometry. The transfered modes and residuals are combined to form the final results (bottom row).

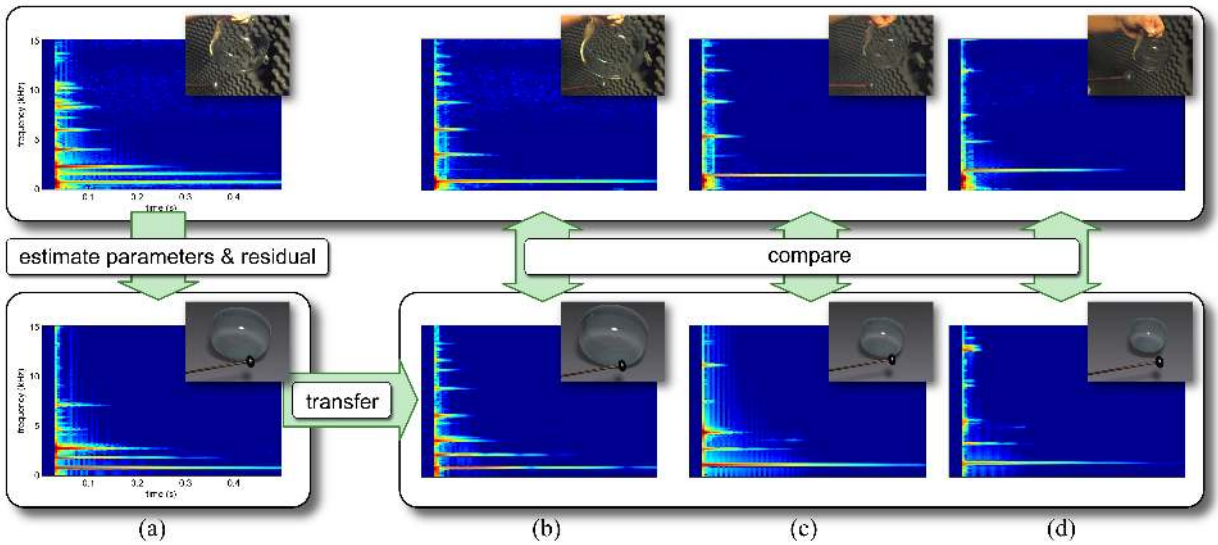


Fig. 13: Comparison of transfered results with real-word recordings: from one recording (column (a), top), the optimal parameters and residual are estimated, and a similar sound is reproduced (column (a), bottom). The parameters and residual can then be applied to different objects of the same material ((b), (c), (d), bottom), and the results are comparable to the real-world recordings ((b), (c), (d), top).

generated from hitting a real-world object, and 16 are synthesized using the techniques described in this paper. For each audio clip, the subject is asked to identify among a set of 5 choices (wood, plastic, metal, porcelain, and glass), from which the sound came. A total of 53 subjects (35 women and 18 men), from age of 22 to 71, participated in this study. The 8 real objects are: a wood plate, a plastic plate, a metal plate, a porcelain plate, and four glass bowls with different sizes. The 16 virtual objects are: three different shapes (a plate, a stick, and a bunny) for each of these four materials: wood,

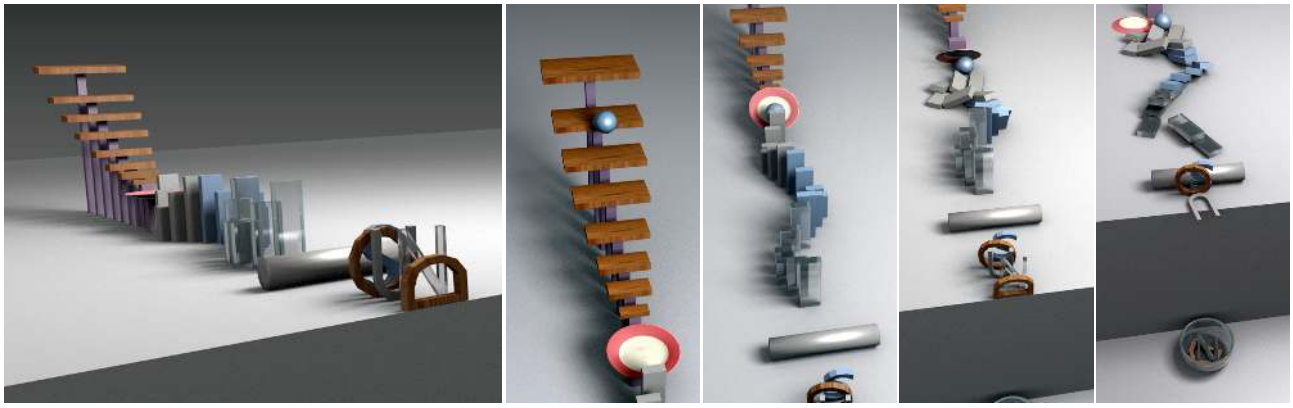


Fig. 14: The estimated parameters are applied to virtual objects of various sizes and shapes, generating sounds corresponding to all kinds of interactions such as colliding, rolling, and sliding.

plastic, metal, and porcelain, plus four glass bowls with different sizes.

We show the cumulative recognition rates of the sounding materials in two separate matrices: Table III presents the recognition rates of sounds from real-world materials, and Table IV reflects the recognition rates of sounds from synthesized virtual materials. The numbers are normalized with the number of subjects answering the questions. For example, Row 3 of Table III means that for a given *real-world* sound recorded from hitting a metal object, none of the subjects thought it came from wood or plastic, 66.1% of them thought it came from metal, 9.7% of them thought it came from porcelain and 24.2% of them thought it came from glass. Correspondingly, Row 3 of Table IV shows that for a sound *synthesized* with our estimated parameters for metal, the percentage of subjects thinking that it came from wood, plastic, metal, porcelain or glass respectively.

We found that the successful recognition rate of virtual materials using our synthesized sounds compares favorably to the recognition rate of real materials using recorded sounds. The difference of the recognition rates (recorded minus synthesized) is close to zero for most of the materials, with 95% confidence intervals shown in Table V. A confidence interval covering zero means that the difference in recognition rate is not *statistically significant*. If both endpoints of a confidence interval are positive, the recognition rate of the real material is significantly higher than that of the virtual material; if both endpoints are negative, the recognition rate of the real material is significantly lower.

In general, for both recorded and synthesized sounds, several subjects have reported difficulty in reliably differentiating between wooden and dull plastic materials and between glass and porcelain. On the other hand, some of the subjects suggested that we remove redundant audio clips, which are in fact *distinct* sound clips of recordings generated from hitting real materials and their synthesized counterparts.

Table III. : Material Recognition Rate Matrix: Recorded Sounds

	Recognized Material				
Recorded Material	Wood (%)	Plastic (%)	Metal (%)	Porcelain (%)	Glass (%)
Wood	50.7	47.9	0.0	0.0	1.4
Plastic	37.5	37.5	6.3	0.0	18.8
Metal	0.0	0.0	66.1	9.7	24.2
Porcelain	0.0	0.0	1.2	15.1	83.7
Glass	1.7	1.7	1.7	21.6	73.3

Table IV. : Material Recognition Rate Matrix: Synthesized Sounds Using Our Method

	Recognized Material				
Synthesized Material	Wood (%)	Plastic (%)	Metal (%)	Porcelain (%)	Glass (%)
Wood	52.8	43.5	0.0	0.0	3.7
Plastic	43.0	52.7	0.0	2.2	2.2
Metal	1.8	1.8	69.6	15.2	11.7
Porcelain	0.0	1.1	7.4	29.8	61.7
Glass	3.3	3.3	3.8	40.4	49.2

Table V. : 95% Confidence Interval of Difference in Recognition Rates

Wood(%)	Plastic(%)	Metal(%)	Porcelain(%)	Glass (%)
(-17.1; 12.9)	(-44.7; 14.3)	(-18.2; 11.3)	(-27.7; -1.6)	(12.6; 35.6)

9. CONCLUSION AND FUTURE WORK

We have presented a novel data-driven, physically based sound synthesis algorithm using an example audio clip from real-world recordings. By exploiting psychoacoustic principles and feature identification using linear modal analysis, we are able to estimate the appropriate material parameters that capture the intrinsic audio properties of the original materials and transfer them to virtual objects of different sizes, shape, geometry and pair-wise interaction. We also propose an effective residual computation technique to compensate for linear approximation of modal synthesis.

Although our experiments show successful results in estimating the material parameters and computing the residuals, it has some limitations. Our model assumes linear deformation and Rayleigh damping. While offering computational efficiency, these models cannot always capture all sound phenomena that real world ma-

terials demonstrate. Therefore, it is practically impossible for the modal synthesis sounds generated with our estimated material parameters to sound exactly the same as the real-world recording. Our feature extraction and parameter estimation depend on the assumption that the modes do not couple with one another. Although it holds for the objects in our experiments, it may fail when recording from objects of other shapes, e.g. thin shells where nonliear models would be more appropriate [Chadwick et al. 2009].

We also assume that the recorded material is homogeneous and isotropic. For example, wood is highly anisotropic when measured along or across the direction of growth. The anisotropy greatly affects the sound quality and is an important factor in making high-precision musical instruments.

Because the sound of an object depends both on its geometry and material parameters, the geometry of the virtual object must be as close to the real-world object as possible to reduce the error in parameter estimation. Moreover, the mesh discretization must also be adequately fine. For example, although a cube can be represented by as few as eight vertices, a discretization so coarse not only clips the number of vibration modes but also makes the virtual object artificially stiffer than its real-world counterpart. The estimated γ, which encodes the stiffness, is thus unreliable. These requirements regarding the geometry of the virtual object may affect the accuracy of the results using this method.

Although our system is able to work with an inexpensive and simple setup, care must be taken in the recording condition to reduce error. For example, the damping behavior of a real-world object is influenced by the way it is supported during recording, as energy can be transmitted to the supporting device. In practice, one can try to minimize the effect of contacts and approximate the system as free vibration, or one can rigidly fix some points of the object to a relatively immobile structure and model the fixed points as part of the boundary conditions in the modal analysis process. It is also important to consider the effect of room acoustics. For example, a strong reverberation will alter the observed amplitude-time relationship of a signal and interfere with the damping estimation.

Despite these limitations, our proposed framework is general, allowing future research to further improve and use different individual components. For example, the difference metric now considers the psychoacoustic factors and material resemblance through power spectrogram comparison and feature matching. It is possible that more factors can be taken into account, or a more suitable representation, as well as a different similarity measurement of sounds can be found.

The optimization process approximates the global optimum by searching through all 'good' starting points. With a deeper investigation of the parameter space and more experiments, the performance may be possibly improved by designing a more efficient scheme to navigate the parameter space, such as starting-point clustering, early pruning, or a different optimization procedure can be adopted.

Our residual computation compensates the difference between the real recording and the synthesized sound, and we proposed a method to transfer it to different objects. However, it is not the only way – much due to the fact that the origin and nature of residual is unknown. Meanwhile, it still remains a challenge to acquire recordings of only the stuck object and completely remove input from the striker. Our computed residual is inevitably polluted by the striker to some extent. Therefore, future solutions for separating sounds from the two interacting objects should facilitate a more accurate computation for residuals from the struck object.

When transferring residual computed from impacts to continuous contacts (e.g. sliding and rolling), there are certain issues to be considered. Several previous work have approximated continuous contacts with a series of impacts and have generated plausible *modal* sounds. Under this approximation, our proposed feature-guided residual transfer technique can be readily adopted. However, the effectiveness of this direct mapping needs further evaluation. Moreover, future study on continuous contact sound may lead to an improved modal synthesis model different than the impact-based approximation, under which our residual transfer may not be applicable. It is then also necessary to reconsider how to compensate the difference between a real continuous contact sound and the modal synthesis sound.

In this paper, we focus on designing a system that can quickly estimate the optimal material parameters and compute the residual merely based on a *single* recording. However, when a small number of recordings of the same material are given as input, machine learning techniques can be used to determine the set of parameters with maximum likelihood, and it could be an area worth exploring. Finally, we would like to extend this framework to other non-rigid objects and fluids, and possibly nonlinear modal synthesis models as well.

In summary, data-driven approaches have proven useful in areas in computer graphics, including rendering, lighting, character animation, and dynamics simulation. With promising results that are transferable to virtual objects of different geometry, sizes, and interactions, this work is the first rigorous treatment of the problem on automatically determining the material parameters for physically based sound synthesis using a single sound recording, and it offers a new direction for combining example-guided and modal-based approaches.

REFERENCES

ADRIEN, J.-M. 1991. Representations of musical signals. MIT Press, Cambridge, MA, USA, Chapter The missing link: modal synthesis, 269–298.

AUDIOKINETIC. 2011. Wwise SoundSeed Impact. `http://www.audiokinetic.com/en/products/wwise-add-ons/soundseed/introduction`.

BESL, P. J. AND MCKAY, N. D. 1992. A method for registration of 3-D shapes. *IEEE Transactions on pattern analysis and machine intelligence*, 239–256.

BONNEEL, N., DRETTAKIS, G., TSINGOS, N., VIAUD-DELMON, I., AND JAMES, D. 2008. Fast modal sounds with scalable frequency-domain synthesis. *ACM Transactions on Graphics (TOG) 27,* 3, 24.

CHADWICK, J. N., AN, S. S., AND JAMES, D. L. 2009. Harmonic shells: a practical nonlinear sound model for near-rigid thin shells. In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers*. ACM, New York, NY, USA, 1–10.

CHADWICK, J. N. AND JAMES, D. L. 2011. Animating fire with sound. In *ACM Transactions on Graphics (TOG)*. Vol. 30. ACM, 84.

COOK, P. R. 1996. Physically informed sonic modeling (PhISM): percussive synthesis. In *Proceedings of the 1996 International Computer Music Conference*. The International Computer Music Association, 228–231.

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

COOK, P. R. 2002. *Real Sound Synthesis for Interactive Applications*. A. K. Peters, Ltd., Natick, MA, USA.

CORBETT, R., VAN DEN DOEL, K., LLOYD, J. E., AND HEIDRICH, W. 2007. Timbrefields: 3d interactive sound models for real-time audio. *Presence: Teleoperators and Virtual Environments 16,* 6, 643–654.

DOBASHI, Y., YAMAMOTO, T., AND NISHITA, T. 2003. Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics. *ACM Trans. Graph. 22,* 3, 732–740.

DOBASHI, Y., YAMAMOTO, T., AND NISHITA, T. 2004. Synthesizing sound from turbulent field using sound textures for interactive fluid simulation. In *Computer Graphics Forum*. Vol. 23. Wiley Online Library, 539–545.

DUBUISSON, M. P. AND JAIN, A. K. 1994. A modified hausdorff distance for object matching. In *Proceedings of 12th International Conference on Pattern Recognition*. Vol. 1. IEEE Comput. Soc. Press, 566–568.

FONTANA, F. 2003. *The sounding object*. Mondo Estremo.

GOPE, C. AND KEHTARNAVAZ, N. 2007. Affine invariant comparison of point-sets using convex hulls and hausdorff distances. *Pattern recognition 40,* 1, 309–320.

GRIFFIN, D. AND LIM, J. 2003. Signal estimation from modified short-time Fourier transform. *Acoustics, Speech and Signal Processing, IEEE Transactions on 32,* 2, 236–243.

ISO. 2003. *ISO 226: 2003: AcousticsNormal equalloudness-level contours*. International Organization for Standardization.

JAMES, D., BARBIČ, J., AND PAI, D. 2006. Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. In *ACM SIGGRAPH 2006 Papers*. ACM, 995.

LAGARIAS, J. C., REEDS, J. A., WRIGHT, M. H., AND WRIGHT, P. E. 1999. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on Optimization 9,* 1, 112–147.

LAKATOS, S., MCADAMS, S., AND CAUSSÉ, R. 1997. The representation of auditory source characteristics: Simple geometric form. *Attention, Perception, & Psychophysics 59,* 8, 1180–1190.

LEVINE, S. N., VERMA, T. S., AND SMITH, J. O. 1998. Multiresolution sinusoidal modeling for wideband audio with modifications. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*. Vol. 6. IEEE, 3585–3588 vol. 6.

LLOYD, D. B., RAGHUVANSHI, N., AND GOVINDARAJU, N. K. 2011. Sound Synthesis for Impact Sounds in Video Games. In *Proceedings of Symposium on Interactive 3D Graphics and Games*.

MORCHEN, F., ULTSCH, A., THIES, M., AND LOHKEN, I. 2006. Modeling timbre distance with temporal statistics from polyphonic music. *Audio, Speech, and Language Processing, IEEE Transactions on 14,* 1 (Jan.), 81– 90.

MOSS, W., YEH, H., HONG, J., LIN, M., AND MANOCHA, D. 2010. Sounding Liquids: Automatic Sound Synthesis from Fluid Simulation. *ACM Transactions on Graphics (TOG)*.

O'BRIEN, J. F., COOK, P. R., AND ESSL, G. 2001. Synthesizing sounds from physically based motion. In *Proceedings of ACM SIGGRAPH 2001*. ACM Press, 529–536.

O'BRIEN, J. F., SHEN, C., AND GATCHALIAN, C. M. 2002. Synthesizing sounds from rigid-body simulations. In *The ACM SIGGRAPH 2002 Symposium on Computer Animation*. ACM Press, 175–181.

OPPENHEIM, A. V., SCHAFER, R. W., AND BUCK, J. R. 1989. *Discrete-time signal processing*. Vol. 1999. Prentice hall Englewood Cliffs, NJ:.

PAI, D. K., DOEL, K. V. D., JAMES, D. L., LANG, J., LLOYD, J. E., RICHMOND, J. L., AND YAU, S. H. 2001. Scanning physical interaction behavior of 3d objects. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '01. ACM, New York, NY, USA, 87–96.

PAMPALK, E., RAUBER, A., AND MERKL, D. 2002. Content-based organization and visualization of music archives. In *Proceedings of the tenth ACM international conference on Multimedia*. ACM, 570–579.

PICARD, C., TSINGOS, N., AND FAURE, F. 2009. Retargetting example sounds to interactive physics-driven animations. In *AES 35th International Conference-Audio for Games, London, UK*.

QUATIERI, T. AND MCAULAY, R. 1985. Speech transformations based on a sinusoidal representation. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '85.* Vol. 10. 489 – 492.

RAGHUVANSHI, N. AND LIN, M. 2006. Symphony: Real-time physically-based sound synthesis. In *Proceedings of Symposium on Interactive 3D Graphics and Games*.

REN, Z., YEH, H., AND LIN, M. 2010. Synthesizing contact sounds between textured models. In *Virtual Reality Conference (VR), 2010 IEEE*. 139 –146.

REN, Z., YEH, H., AND LIN, M. C. 2012. Geometry-Invariant Material Perception: Analysis and Evaluation of Rayleigh Damping Model. *UNC Technical Report*.

ROADS, C. 2004. *Microsound*. The MIT Press.

SERRA, X. 1997. Musical sound modeling with sinusoids plus noise. *Musical signal processing*, 497–510.

SERRA, X. AND SMITH III, J. 1990. Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal 14,* 4, 12–24.

SHABANA, A. 1997. *Vibration of discrete and continuous systems*. Springer Verlag.

TREBIEN, F. AND OLIVEIRA, M. 2009. Realistic real-time sound resynthesis and processing forinteractive virtual worlds. *The Visual Computer 25*, 469–477.

VÄLIMÄKI, V., HUOPANIEMI, J., KARJALAINEN, M., AND JÁNOSY, Z. 1996. Physical modeling of plucked string instruments with application to real-time sound synthesis. *Journal of the Audio Engineering Society 44,* 5, 331–353.

VÄLIMÄKI, V. AND TOLONEN, T. 1997. Development and calibration of a guitar synthesizer. *PREPRINTS-AUDIO ENGINEERING SOCIETY*.

VAN DEN DOEL, K., KNOTT, D., AND PAI, D. K. 2004. Interactive simulation of complex audiovisual scenes. *Presence: Teleoper. Virtual Environ. 13*, 99–111.

VAN DEN DOEL, K., KRY, P., AND PAI, D. 2001. FoleyAutomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM New York, NY, USA, 537–544.

VAN DEN DOEL, K. AND PAI, D. K. 1998. The sounds of physical shapes. *Presence: Teleoper. Virtual Environ. 7*, 382–395.

VAN DEN DOEL, K. AND PAI, D. K. 2002. Measurements of perceptual quality of contact sound models. In *In Proceedings of the International Conference on Auditory Display (ICAD 2002*. 345–349.

ZHENG, C. AND JAMES, D. L. 2009. Harmonic fluids. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*. ACM, New York, NY, USA, 1–12.

ZHENG, C. AND JAMES, D. L. 2010. Rigid-body fracture sound with precomputed soundbanks. *ACM Trans. Graph. 29*, 69:1–69:13.

ZHENG, C. AND JAMES, D. L. 2011. Toward high-quality modal contact sound. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011) 30,* 4 (Aug.).

ZWICKER, E. AND FASTL, H. 1999. *Psychoacoustics: Facts and models*, 2nd updated edition ed. Vol. 254. Springer New York.