AUDIO-MATERIAL MODELING AND RECONSTRUCTION FOR MULTIMODAL INTERACTION

Charles Auston Baker Sterling

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2019

Approved by:

Ming C. Lin

Dinesh Manocha

Gary Bishop

Jack Snoeyink

Vladimir Jojic

ABSTRACT

Charles Auston Baker Sterling: Audio-Material Modeling and Reconstruction for Multimodal Interaction
(Under the direction of Ming C. Lin)

Interactive virtual environments enable the creation of training simulations, games, and social applications. These virtual environments can create a sense of presence in the environment: a sensation that its user is truly in another location. To maintain presence, interactions with virtual objects should engage multiple senses. Furthermore, multisensory input should be consistent, *e.g.*, a virtual bowl that visually appears plastic should also sound like plastic when dropped on the floor.

In this dissertation, I propose methods to improve the perceptual realism of virtual object impact sounds and ensure consistency between those sounds and the input from other senses. Recreating the impact sound of a real-world object requires an accurate estimate of that object's material parameters. The material parameters that affect impact sound—collectively forming the audio-material—include the material damping parameters for a damping model. I propose and evaluate damping models and use them to estimate material damping parameters for real-world objects. I also consider how interaction with virtual objects can be made more consistent between the senses of sight, hearing, and touch.

First, I present a method for modeling the damping behavior of impact sounds, using generalized proportional damping to both estimate more expressive material damping parameters from recorded impact sounds and perform impact sound synthesis. Next, I present a method for estimating material damping parameters in the presence of confounding factors and with no knowledge of the object's shape. To accomplish this, a probabilistic damping model captures various external effects to produce robust damping parameter estimates. Next, I present a method for consistent multimodal interaction with textured surfaces. Texture maps serve as a single unified representation of mesoscopic detail for the purposes of visual rendering, sound synthesis, and rigid-body simulation. Finally, I present a method for geometry and material classification using multimodal audio-visual input. Using this method, a real-world scene can be scanned and virtually reconstructed while accurately modeling both the visual appearances and audio-material parameters of each object.

ACKNOWLEDGMENTS

I would like to thank the members of the GAMMA research group. I've loved hearing about all the various research projects going on in the group, and sharing ideas about how to improve. I'd like to give a particular mention to the other people working on sound-related projects, as we were able to discuss the more technical details: Nic Morales, Atul Rungta, Carl Schissler, Justin Wilson, Zhenyu Tang, and Sam Lowe. Thank you to everyone who participated in my user studies, which I know were often difficult and tedious. The computer science department was a great place to work: the students across all the various research groups were friendly and supportive, while the staff were always ready to help with the issues that inevitably popped up.

Thank you to all of my direct collaborators, including Justin Wilson, Nicholas Rewkowski, and Sam Lowe, whose contributions show out throughout this document. And thank you to my advisor Ming Lin, who was both a reliable collaborator and thoughtful mentor. You kept me on the right path even when I started to stray, and your knowledge of how to construct a solid paper submission is unmatched.

Finally, thank you to the people who kept me sane. My parents and sister always provided a place of stability to return to over breaks, and I love them all. The UNC marathon team kept me active and provided another avenue for growth and achievement in parallel with my research. Through the team I also came to know some great people, including but not limited to Daniela DeCristo, Max Patel, Sean McCaffery, Yuuki Butler, and Michael Gong. And finally, thank you to everyone who joined in with my little board game group; I'm glad I've had the chance to try so many games with you all.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Introduction

Virtual environments have found applications for different user interaction scenarios. Interactive training simulations let users practice high-risk tasks, such as performing surgery or piloting an airplane, with low risk. Immersive story-driven video games let users interact with another environment or involve themselves in an engaging narrative. Emerging social applications let multiple users from around the world unite in one virtual location and feel as though they are in the same space.

In all three scenarios, users should be able to forget their presence in the real world and temporarily experience a *sense of presence* in the virtual environment (Lombard and Jones, 2015; Lee, 2006). If users are reminded that the virtual environment is fake, they experience a *break in presence*, which reduces the emotional weight of the virtual environment and makes it less effective at its intended goal. Thus, avoiding these breaks in presence can improve the quality of users' experiences. Training simulations feel more lifelike, video games convey more powerful emotions, and social interactions with other users flow more naturally.

Virtual environments most commonly recreate input to the senses of sight and hearing. The visual appearances and audio of the real world are relatively easily replaceable with those of a virtual world. A virtual

Figure 1.1: A virtual environment with interactive objects of different shapes and materials. The objects in these scenes should produce realistic impact sounds consistent with their visual appearances.

reality (VR) headset such as the Oculus Rift or the HTC Vive replaces the visual input, while headphones (sometimes built into the VR headset) replace the audio input. Examples of VR-enabled environments are shown in Figure 1.1. Humans also rely heavily on the sense of touch, but virtual environments are limited by current hardware, which cannot effectively recreate complete input for the sense of touch.

Undesirable breaks in presence have many causes; a common cause is violation of a user's expectations about their interactions. Each sense creates expectations for the other senses; *sensory conflict* occurs when senses provide conflicting expectations that violate one another. For example, if a table looks like wood when visually inspected, but sounds like ringing metal when struck, the user's expectations have been violated and a break in presence is likely. For another example, a rough surface will visually have a diffuse scattering of light instead of a sharp reflection, and if the user feels that surface with a stylus, the roughness they feel should match the roughness they see.

Maintaining users' expectations about their interactions does not require perfectly realistic virtual environments. Some studies have found that visual rendering quality has little effect on presence (Zimmons and Panter, 2003) (though this is still an open area of research (Slater et al., 2009)), and other studies have found that virtual environments with deliberately low-fidelity visuals still evoke strong desired emotional responses (Slater et al., 2006b,a). As long as the user can establish consistent expectations about the environment, sensory conflict can be avoided, regardless of the objective real-world accuracy of the recreations. Consistent sensory expectations cause the sensation of *perceptual realism*, protecting a user's sense of presence from sensory conflict.

A common source of sensory conflict is interaction with objects. Objects, such as furniture, tableware, and musical instruments, are common in real and virtual environments, and interaction with objects involves multiple senses. As objects are moved or struck, we expect them to produce *impact sounds*. To create realistic impact sounds, my work uses *modal sound synthesis*, a physically-based method which models vibrations in struck objects (O'Brien et al., 2002). When using physically-based methods for sound synthesis, perceptual realism depends on an object's *material parameters*. The material parameters affecting impact sounds can be collectively referred to as the *audio-material*, in contrast with parameters affecting the visual appearance or haptic texture of the surface. Since users have expectations about how virtual objects should sound from their other senses (Fujisaki et al., 2014), it is important to use accurate material parameters.

The audio-material of a struck object affects the impact sound's rate of decay, *e.g.*, a plastic object has a short-lasting sound while a metal object has a long-lasting sound. Different materials cause different

Figure 1.2: Images of objects and their virtual reconstructions. The top row shows pictures of the real objects, while the bottom row shows manually-constructed meshes and textures modeling the objects. We seek to create realistic multimodal interactions with these objects.

amounts of *damping*, producing different decay rates. *Damping models* are a common approximation that simplify computations by modeling the damping as a function of an object's mass and stiffness. When performing modal sound synthesis, selecting realistic damping rates is important for recreating the sound of the appropriate material.

In order to interact with virtual objects, they must first be created by defining properties such as its shape and material parameters. A common way of creating virtual environments and objects is modeling existing real-world objects. Figure 1.2 shows examples of real-world objects that have been modeled by hand, though this process can be automated to a limited degree. The shape of an object can be acquired through a 3D scan, and the object's material parameters can be acquired through vision or its impact sounds. The acquired properties can be used to *virtualize* the original object, reconstructing a digital version of the object for interaction in virtual environments. However, few methods have attempted to combine these two input modalities (vision and impact sounds) in a single coupled process. An object reconstruction method that ensures consistency between input modalities could better recreate virtual objects with minimal sensory conflict.

In this dissertation, I propose methods for multimodal interaction and object reconstruction. These methods are evaluated through comparisons to ground truth and perceptual experiments. Comparisons to ground truth analyze the difference between my results, prior results, and a referenced ground truth. Perceptual experiments consist of user studies to analyze perception of objects and sounds, and are frequently used in my work due to the emphasis on ensuring perceptual realism. In some user studies, subjects report their opinions on the realism, effectiveness, or similarity of real or synthetic virtual objects. In other user studies, subjects must complete tasks using either my methods or those from previous work. User studies directly

evaluate the methods with respect to user expectations, providing insight into the methods' performance in an immersive setting.

1.1.1 Thesis Statement

"Interaction with objects in virtual environments can be made more perceptually realistic by using expressive object material models that account for real-world phenomena and by reducing multimodal sensory conflict."

In this dissertation, I describe research that improves interaction with virtual objects by improving material modeling and object virtualization. The contributions proposed by myself and my collaborators include:

Damping Modeling for Modal Sound Synthesis: We propose a novel method for deriving new material damping models which are able to express a wider range of damping behaviors than the traditional Rayleigh damping model. We extend modal sound synthesis to support these new models, We also propose a method for material parameter estimation that uses a single impact sound to accurately estimate damping parameters for *any* damping model. Perceptual evaluation demonstrates that no single existing damping model best represents the damping behavior of every material, and thus multiple damping models should be considered.

Robust Material Parameter Estimation: We propose a novel method for estimating material damping parameters from recorded impact sounds. We model the observed damping values in recorded sounds with a probabilistic model, which expressly models multiple external factors affecting estimation of material damping. This method requires no information about the shape of the object or the locations of the impacts, and reduces the effect of external factors to produce more accurate estimates of the material damping parameters in suboptimal recording environments. Perceptual evaluation shows that sounds synthesized using our estimated material parameters are comparable in realism to those of previous work and human-tuned sounds. Given that our method places fewer requirements on the inputs, our method significantly reduces manual effort needed to obtain high-quality results.

Multimodal Surface Interaction: We propose a method for using a single texture map as a unified representation of detail for visual rendering, audio rendering, tactile rendering, and physical simulation. Our method runs in real time and allows for multimodal interaction with textured surfaces, while the unified representation ensures consistency between senses. In task-based user evaluation, our method improves

results over alternative, conflicting representations of detail. In a comparison study, we identify situations where each of our two texture representations are most effective.

Multimodal Object Classification We propose a method for estimating both an object's material and geometry leveraging both audio and visual input. The method takes as input an impact sound and (optionally) a voxelized estimate of the object's shape. We perform quantitative evaluation on datasets in which the output is a geometry class (such as "chair" or "dresser"), and on datasets in which the output is a specific geometric model (a retrieval task). Our method results in state-of-the-art accuracy for these sets of inputs, while proving competitive against methods using different sets of inputs.

1.2 Main Contributions

In this section, I discuss my primary areas of research.

1.2.1 Interactive Modal Sound Synthesis Using Generalized Proportional Damping

In order to create higher quality modal sound, this research aims to improve modeling of material damping. We (myself and Ming C. Lin) more accurately capture real-word damping behavior by considering more expressive damping models. We apply these expressive models to modal sound synthesis to better recreate the sounds of real-world materials.

Material damping is a complex phenomenon, and is difficult to accurately model. For example, the presence of damping may give rise to *complex* modes of vibration, which are more difficult to model than *normal* modes (Caughey and O'Kelly, 1965). In practice, approximations are used to produce computationally simpler models using only normal modes. The most common approximation is to assume all damping is viscous and that the decay rate of a material is a linear combination of its density and stiffness. This model is referred to as Rayleigh (or linearly proportional) damping, and produces only normal modes. It is the de-facto damping model for modal sound synthesis, but has always been understood to be an approximation for convenience.

Other damping models are common in material and structural analysis, but have not been thoroughly examined for interactive sound synthesis. Caughey damping is a polynomial extension of the linear Rayleigh damping model (Caughey, 1960; Caughey and O'Kelly, 1965). Generalized proportional damping (GPD) is the most general damping model to date that limits vibrations to normal modes (Adhikari, 2006). These

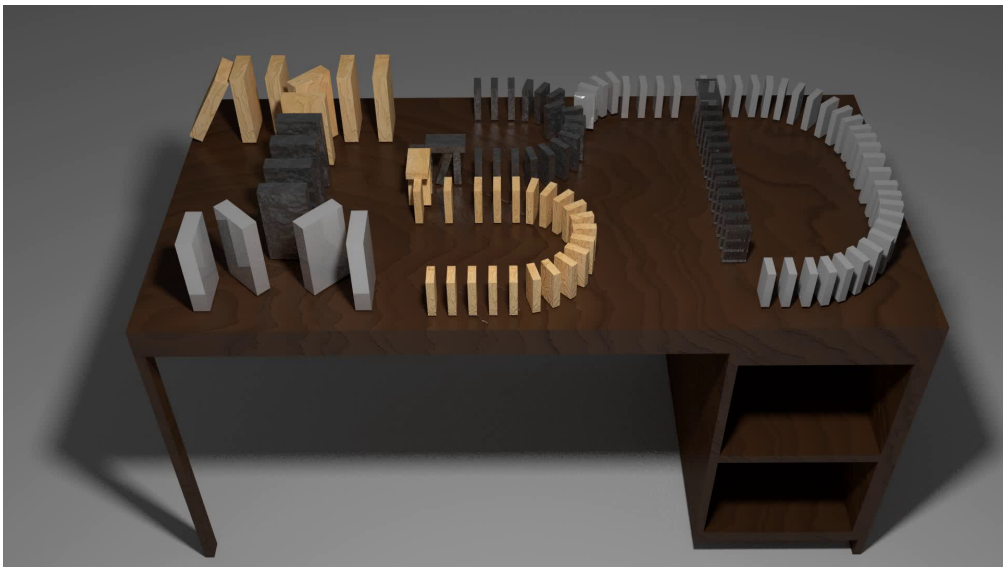


Figure 1.3: A scenario with dominoes made of different materials. Each material uses a set of material parameters estimated from recorded impact sounds, including parameters for a damping model.

alternative damping models may be able to improve sound quality by providing a better fit to observed real-world damping.

We propose a method that employs Generalized Proportional Damping to create alternative damping models for sound synthesis and we propose specific damping models within the larger space of GPD functions. These damping models are more expressive, enabling them to model damping behavior that would be coarsely approximated by the Rayleigh damping model. We also propose a method for estimating the damping parameters of a real-world object using a recorded impact sound as input. This parameter estimation method works for any arbitrary damping model, producing estimates of the parameters specific to that model. We also conduct a user study to evaluate the perceptual differences between multiple damping models. Figure 1.3 shows one scenario with objects creating sound based on materials estimated from recorded impact sounds. More results can be found in Chapter 3 and online: `http://gamma.cs.unc.edu/gpdsynth/`.

1.2.2 Audio-Material Reconstruction for Virtualized Reality using a Probabilistic Damping Mode

Recorded impact sounds can be used to estimate damping parameters, but resulting parameters may be inaccurate if the sounds are recorded in noisy and uncontrolled recording environments. This research explores a novel probabilistic damping model for estimating material damping parameters while reducing the

Figure 1.4: A small porcelain plate (left) and a small travertine tile (right) being struck to produce impact sounds. Both objects are supported by a gripping hand. Methods for material damping parameter estimation should be robust to these external damping factors.

confounding effect of the recording environment. My collaborators on this research are Nicholas Rewkowski, Roberta L. Klatzky, and Ming C. Lin.

While recent methods have been able to estimate material damping parameters (Ren et al., 2013b), they assume all observed decay in amplitude is due to material damping and not any other source. However, multiple external factors produce effects similar to material damping, causing error in material damping estimates.

One external factor is *support damping*. In the real world, an object struck for recording must be supported in some way, *e.g.*, held by hand or left to rest on another surface. The interface between the object and its *support* introduces additional damping, as energy is transferred from the vibrating object to the support. Figure 1.4 shows multiple objects supported by a hand while being struck, altering the produced sound. Figure 4.4 provides a more in-depth example: depending on how the bowl is held, it produces dramatically different sound.

Other external factors include complex modes of vibration, room acoustics, and error in the feature extraction step. Complex modes of vibration are not captured by standard damping models, which only model normal modes of vibration. Room acoustics—reflections off walls—extend the length of sounds when recording is performed in an enclosed room. Feature extraction steps are common in most damping parameter estimation methods, but even with clean input these steps often introduce their own error.

In realistic, uncontrolled environments with significant effect from external factors, the parameters estimated by current methods are not truly material parameters. Instead, they are parameters modeling *both* the material and the environment used for the recording and thus do not generalize to arbitrary environments.

We propose a practical and efficient method to estimate material damping parameters from recorded impact sounds, while accounting for the external factors present in the recording environment. We explicitly model the external factors using a probabilistic damping model. For a given frequency of vibration and parameters describing the recording environment, this model provides a probability distribution of possible observable damping values. Using multiple impact sounds as input, the probabilistic damping model can be optimized to fit the sounds, providing an estimate of the material damping parameters separately from the external factors. The optimized damping parameters can be those of any real-valued damping model, and we propose one additional hybrid model combining Rayleigh and power law damping.

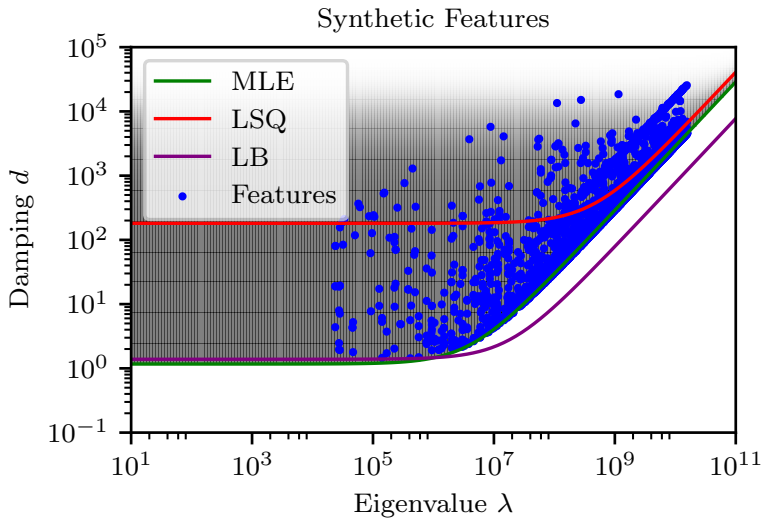


Figure 1.5: Parameter estimation on sound features. Each feature is one extracted mode of vibration, consisting of an eigenvalue λ_i (approximately the square of its frequency) and its corresponding damping coefficient d_i. Estimated Rayleigh damping curves are plotted, with the variation from the curve caused by external factors. Our method is labeled MLE, and provides the ideal fit to the lower bound of the sound features

Our method is more applicable to real-world recordings taken in less controlled environments. The method is fast, requires no prior knowledge about the recorded object, and can use multiple recordings to improve accuracy. Figure 1.5 shows a visual representation of the material damping models as estimated from the real-world sound features shown as points. In the absence of external damping factors, all of these

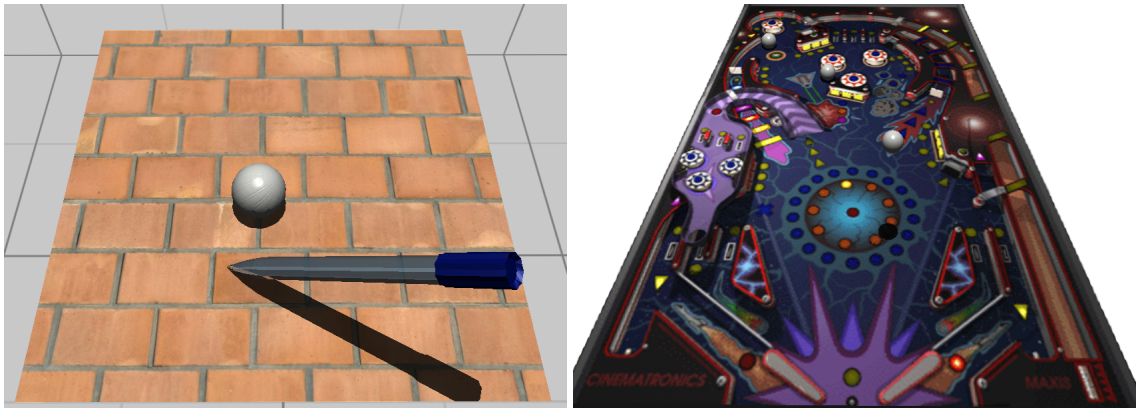


Figure 1.6: A selection of applications based on our method for multimodal interaction: a virtual environment with a normal mapped surface (left) and a pinball game created through a normal mapped surface (right). In both environments, the texture map informs all interaction modalities.

points would fall along one line representing the damping from the material alone. However, external factors cause the points to vary, throwing off a more traditional least squares (LSQ) approach while our method (MLE) provides a better fit. In perceptual evaluation, subjects found that sounds synthesized using parameters estimated with our method were comparable in quality to those of previous work and human hand-tuned parameters. Therefore, our method requires significantly less manual effort to produce high quality results. More results are available in Chapter 4 and online: `http://gamma.cs.unc.edu/ProbDampModel/`.

1.2.3 Integrated Multimodal Interaction Using Texture Representations

There have been a few efforts to unify interaction in virtual environments across senses (see Section 2.2). However, they do not clearly consider sensory conflict, nor have any brought together all of visual rendering, haptic rendering, sound rendering, and physical simulation. Sensory conflict is particularly important when considering textured objects, which are often modeled through approximations. In this line of research, we (myself and Ming C. Lin) use texture representations of detail—normal and relief maps—as a unified source of information for all interaction modalities.

Interaction with textured surfaces via haptic rendering, sound rendering, and rigid-body simulation have each been independently explored (Otaduy et al., 2004; Ren et al., 2010), but have not been integrated together consistently. For example, a previous method for sound rendering of contacts with textured surfaces (Ren et al., 2010) displays a pen sliding smoothly across highly bumpy surfaces. While the generated sound from this interaction is dynamic and realistic, the smooth *visual* movement of the pen does not match the texture

implied by the sound. In order to minimize sensory conflict, it is critical to present a unified and seamlessly integrated multimodal display to users, ensuring rendering is consistent across the senses of sight, hearing, and touch.

In the real world, objects behave differently when bouncing, sliding, or rolling on bumpy or rough surfaces than they do on flat surfaces. In a virtual environment, a bumpy or rough surface can be represented by its visual texture equivalent mapped to a flat surface. While the surface would appear visually complex, the underlying flat surface would cause simple physical behavior, causing sensory conflict and breaking the sense of presence. In order to model such physical behavior, a physics simulator would require a fine triangle mesh with sufficient surface detail, but in most cases a sufficiently fine mesh is unavailable or would require prohibitive amounts of memory. Since texture maps contain information about the fine detail of a mapped surface, it is possible to use that information to recreate the physical behavior of the fine triangle mesh.

To accomplish this, we propose a new method for simulation of physical behaviors for rigid objects textured with normal maps. We also propose methods for seamlessly integrated multimodal interaction using normal and relief maps. By using a single representation of surface detail across all interaction modalities, we reduce sensory conflict for users. See Figure 1.6 for examples of interaction with textured surfaces. With our method, a virtual pen is controlled through a haptic device, allowing a user to interact with the environment while feeling forces in response. A simulated ball rolls on the surface, its motions affected by both the surface texture and the pen. Contacts between the pen, ball, and surface create physically-based sound, bringing together sight, hearing, touch, and physical simulation.

We evaluate our methods through texture identification and representation comparison user studies. In the texture identification study, subjects were asked to identify the surface displayed to them, but in some trials certain interaction modalities were removed. When all modalities were present using our method, performance on the task was at its highest, demonstrating that the senses were not in conflict with one another. In the representation comparison study, subjects answered questionnaires as they interacted with either normal-mapped or relief-mapped surfaces. When all modalities were present, subjects found the relief-mapped surfaces to be more realistic. More results are available in Chapter 5 and online: `http://gamma.cs.unc.edu/MultiDispTexture/`.

1.2.4 Impact Sound Neural Network for Audio-Visual Object Classification

A real-world object reconstructed in virtual reality should minimize sensory conflict by ensuring consistency between the object's shape, surface appearance, and audio-material. Object shape and surface appearance have historically been estimated through visual cues. Similarly, the audio-material can been estimated through audio cues, as I demonstrate in Chapters 3 and 4. However, if an object's shape and audio-material are estimated separately through independent methods, sensory conflict may appear. Visual methods for shape reconstruction cannot determine internal object structure (*e.g.*, whether an object is solid or hollow) while audio methods for material estimation are underconstrained (multiple shape/material combinations may produce the same impact sound).

These visual and audio cues can complement one another. Impact sounds provide information about internal object structure that visual methods cannot see. Visual estimates of an object's shape provide constraints to audio-based material parameter estimation. Therefore, estimation of either shape or material could benefit from using both visual and audio modalities of input.

In this research area, my collaborators—Justin Wilson, Sam Lowe, and Ming C. Lin—and I explore the combination of these modalities. We propose a method for estimating both an object's material and shape geometry using combined audio-visual inputs. As a visual input, we use a coarse voxelized shape representation which can be acquired from a rough 3D visual reconstruction or a synthetic dataset such as ModelNet (Wu et al., 2015). As an audio input, we use a single impact sound from the object in question, which can be acquired from a recording of a real-world sound or from modal sound synthesis on a virtual object.

Our method uses a novel neural network architecture, called the Impact Sound Neural Network (ISNN), to process and fuse these two inputs. We present an audio-only network (ISNN-A) for material and geometry classification which uses convolutional layers to process an input sound encoded as a spectrogram. We also present a multimodal network (ISNN-AV) which fuses ISNN-A and VoxNet (Maturana and Scherer, 2015) to jointly produce estimates of material and geometry.

We perform quantitative evaluation on multiple datasets. The synthetic ModelNet10 and ModelNet40 datasets (Wu et al., 2015) produce classifications to object classes such as "table" or "dresser". We synthesize sounds for each ModelNet object, and our ISNN networks obtain higher classification accuracy than baselines for both audio-only and audio-visual inputs. We present a new dataset, RSAudio, consisting of both recorded

and synthesized sounds, where each sound classifies to a specific shape geometry. On this dataset and other audio-only impact sound datasets (Arnab et al., 2015; Zhang et al., 2017b), our ISNN-A network also outperforms baselines. Finally, we present a utility for scene reconstruction in which impact sounds can be recorded to classify and segment objects. More results are available in Chapter 6 and online: `http://gamma.cs.unc.edu/ISNN/`.

CHAPTER 2: BACKGROUND

In this chapter, I review work related to the main aspects of this research. I also provide a mathematical background of modal sound synthesis.

2.1 Sound Synthesis

Sound synthesis techniques recreate natural sounds for virtual environments. Sounds are dynamic and can be created by a variety of sound sources. Different types of sound sources produce different types of sounds, so different models are needed. Examples of sound sources that have been modeled are liquids (Langlois et al., 2016; Moss et al., 2010), paper (Schreck et al., 2016), and fire (Chadwick and James, 2011).

In this dissertation, the focus is on sounds created by rigid objects. Strings and drums can be simulated through physical models such as the Karplus-Strong algorithm (Karplus and Strong, 1983) and digital waveguide synthesis (Smith, 1992). Simple objects with known analytical vibration patterns can be simulated through additive synthesis, where individual sine waves are added together to create more complex sounds (van den Doel and Pai, 1996). Arbitrary rigid objects use the same additive synthesis method, but to determine their frequencies of vibration, or *modes of vibration*, discretized models of the objects need to be analyzed first (Morrison and Adrien, 1993; O'Brien et al., 2002). This is referred to as *modal sound synthesis*, consisting of a precomputation step called "modal analysis" and a runtime synthesis step called "modal synthesis". We now review the details of this method and explain the need for accurate damping parameters.

2.1.1 Modal Analysis

When a rigid object is struck, it vibrates in response, though these vibrations may be imperceptible to the eye. As the surface of the object vibrates and deforms, the surrounding air is rapidly compressed and expanded, creating pressure waves which propagate through the environment. Our ears perceive the variation in air pressure as sound. The standard range of human hearing covers sound waves between 20 Hz and 20

kHz. In modal analysis, the shape and material parameters of the object are analyzed to decompose the vibrations into a set of *modes of vibration*. Each mode of vibration describes one independent component of the overall vibration as the object oscillates sinusoidally over time. Each object has a different set of modes depending on the object's shape and material. Vibrations from an impact can roughly be represented as a linear combination of normal modes with different amplitudes, frequencies, and phases.

Modal analysis is often performed numerically, where the object is represented using a discretized model such as a FEM mesh or spring-mass thin-shell system. Regardless of the choice of discretization, we can consider the dynamics of the system as it vibrates using a system of equations:

$$\mathbf{M}\ddot{\mathbf{r}} + \mathbf{C}\dot{\mathbf{r}} + \mathbf{K}\mathbf{r} = \mathbf{f} \tag{2.1}$$

Here, $\mathbf{r}$ is a vector of vertex displacements, where a vector of all zeros represents the object at rest. Since we usually work with three-dimensional objects, an object with n discrete elements would have a $\mathbf{r} \in \mathbb{R}^{3n}$. $\mathbf{f}$ is the vector of forces applied to each element, inducing vibrations. $\mathbf{M}$ is the mass matrix, which describes the distribution of mass throughout the object. $\mathbf{C}$ is the viscous damping matrix, which describes how the velocity of the elements $\dot{\mathbf{r}}$ decays over time. $\mathbf{K}$ is the stiffness matrix, in which the connectivity of the elements is defined. Given these matrices, we can properly simulate the vibration of the object in response to an impulse. $\mathbf{M}$ and $\mathbf{K}$ can be constructed through knowledge of the shape of the object and its material parameters, notably its density, Poisson's ratio, and Young's modulus. The damping matrix $\mathbf{C}$, is not as simple to construct.

Modal analysis examines the eigenvalues and eigenvectors of the system in free vibration, that is, with $f = 0$ after some initial impulse has been applied. Temporarily ignoring damping, we can set up a generalized eigenvalue problem of the form:

$$\mathbf{K}\mathbf{v} = \lambda\mathbf{M}\mathbf{v} \tag{2.2}$$

Finding this eigendecomposition and combining the eigenvectors into a matrix $\mathbf{\Phi}$ allows the matrices $\mathbf{M}$ and $\mathbf{K}$ to be diagonalized. Specifically, the eigenvectors are mass-normalized such that:

$$\mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi} = \mathbf{I} \quad \text{and} \quad \mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi} = \mathbf{\Omega^2} \tag{2.3}$$

The matrix $\mathbf{\Phi}$ can be intuitively described as a matrix that transforms between object space and mode space: each column of $\mathbf{\Phi}$ contains the shape of a normal mode, while $\mathbf{\Phi}^T\mathbf{f}$ converts forces on elements to normal mode amplitudes. The natural undamped frequencies of the system are contained in the diagonal matrix $\mathbf{\Omega}$, while their squares in $\mathbf{\Omega}^2$ are the eigenvalues of the system. We can continue the decoupling by considering the system in mode space $\mathbf{z} = \mathbf{\Phi}^T\mathbf{r}$:

$$\mathbf{\Phi^T M \Phi \ddot{z} + \Phi^T C \Phi \dot{z} + \Phi^T K \Phi z = \Phi^T f} \tag{2.4}$$

$$\mathbf{\ddot{z} + \Phi^T C \Phi \dot{z} + \Omega^2 z = \Phi^T f} \tag{2.5}$$

Equation (2.5) now runs into problems with the damping matrix $\mathbf{C}$. While $\mathbf{\Phi}$ diagonalizes $\mathbf{M}$ and $\mathbf{K}$, if it does not diagonalize $\mathbf{C}$ then the system does not properly decouple and the resulting modes are not linearly independent. The linearly *dependent* modes are called complex modes, and accurately modeling them is much more difficult compared to the linearly *independent* normal modes (Imregun and Ewins, 1995). We must now consider methods for modeling damping behavior and constructing appropriate $\mathbf{C}$ matrices.

2.1.2 Damping Modeling

Damping has long been a concern in analysis of vibrations of buildings and other structures (Nashif et al., 1985; Adhikari and Woodhouse, 2001). There are a number of ways to model material-based damping to varying degrees of accuracy (Woodhouse, 1998; Slater et al., 1993), and standard tests have been designed to consistently measure damping in materials (E756, 2017). Complex models are often required to produce accurate fits to observed damping behavior (Adhikari, 2001).

To construct appropriate $\mathbf{C}$ matrices, for sound synthesis purposes we restrict ourselves to *classical* damping with only normal modes, which means all of our damping matrices must be diagonalizable by $\mathbf{\Phi}$. Various damping models have been developed that guarantee only normal modes (Caughey, 1960). These damping models typically have real-valued parameters that vary between materials. In this dissertation, α_j is used to represent these *damping parameters*. However, be aware that each damping model has a different definition of α_j.

The most popular model is Rayleigh damping (Rayleigh, 1896), in which the damping is a linear combination of mass and stiffness:

$$\mathbf{C} = \alpha_1\mathbf{M} + \alpha_2\mathbf{K} \tag{2.6}$$

α_1 and α_2 are the real-valued parameters in this Rayleigh damping model. Rayleigh damping has been, to the best of our knowledge, the only damping model used for sound synthesis in computer graphics.

Caughey and O'Kelly proposed a more general model, now known as *Caughey damping* or a *Caughey series* (Caughey and O'Kelly, 1965), which they proved to be a necessary and sufficient condition for normal modes:

$$\mathbf{C} = \mathbf{M}\sum_{j=0}^{n-1}\alpha_j(\mathbf{M}^{-1}\mathbf{K})^j \tag{2.7}$$

All α_j are real-valued parameters for Caughey damping models. In practice, the series could truncated after a few terms.

For a given damping model, the real-valued parameters α_j are the *damping parameters* which define the damping of each mode. By varying these values, the same object can be made to sound like a wide range of materials. Damping parameters have been shown to be perceptually geometry-invariant for a wide range of geometries under the Rayleigh damping model (Ren et al., 2013a); it is reasonable to assume this holds for other damping models as well. Thus, if damping parameters can be estimated for a metal bowl, synthesizing sound for a solid cube with those parameters will produce a metallic sound. However, the geometry-invariance assumption has only been thoroughly tested on thick, very rigid objects (Ren et al., 2013a), and the assumption may fail for thin-shelled objects (Chadwick et al., 2009), less rigid objects, objects with loosely-coupled points of self-collision, or objects demonstrating nonlinear vibrational behavior.

2.1.3 Modal Synthesis

With these damping models, we have a damping matrix guaranteed to be diagonalizable by $\mathbf{\Phi}$. With the system diagonalized, the free-vibration form is now decoupled into independent second order differential equations:

$$\ddot{z}_i + c_i\dot{z}_i + \omega_{in}^2 z_i = 0 \tag{2.8}$$

c_i is an entry in the diagonalized damping matrix corresponding to the i'th mode of vibration, and is discussed in more detail in Section 2.1.4. These equations each have known analytical solutions as damped sinusoids:

$$z_i(t) = a_i e^{-d_i t}\cos(\omega_{id}t) \tag{2.9}$$

a_i is the amplitude of the sinusoid, while the damping coefficient $d_i = c_i/2$ defines the rate at which the amplitude decreases. ω_{in} in Equation (2.8) is the natural undamped frequency of oscillation, but in the presence of damping we use the *damped* frequency ω_{id}:

$$\omega_{id} = \sqrt{\omega_{in}^2 - d_i^2} \tag{2.10}$$

2.1.4 Obtaining Damping Coefficients

In practice, we do not actually want to perform the matrix operations in the damping models. Through heavy use of Equation (2.3), we can find analytical solutions for how $\mathbf{C}$ is diagonalized and compute c_i in terms of the corresponding eigenvalue ω_{in}^2. The solution for Rayleigh damping is common in modal sound synthesis work:

$$\begin{aligned} \mathbf{\Phi}^T\mathbf{C}\mathbf{\Phi} &= \mathbf{\Phi}^T\alpha_1\mathbf{M}\mathbf{\Phi} + \mathbf{\Phi}^T\alpha_2\mathbf{K}\mathbf{\Phi} \\ &= \alpha_1\mathbf{I} + \alpha_2\mathbf{\Omega^2} \\ c_i &= \alpha_1 + \alpha_2\omega_{in}^2 \end{aligned} \tag{2.11}$$

Caughey damping is slightly more involved, but leads to a fairly intuitive solution:

$$\begin{aligned} \mathbf{\Phi}^T\mathbf{C}\mathbf{\Phi} &= \mathbf{\Phi}^T\mathbf{M}\sum_{j=0}^{n-1}\alpha_j(\mathbf{M}^{-1}\mathbf{K})^j\mathbf{\Phi} \\ &= \mathbf{\Phi}^{-1}\sum_{j=0}^{n-1}\alpha_j(\mathbf{\Phi\Omega}^2\mathbf{\Phi}^{-1})^j\mathbf{\Phi} \\ &= \sum_{j=0}^{n-1}\alpha_j\mathbf{\Omega}^{2^j} \\ c_i &= \sum_{j=0}^{n-1}\alpha_j{\omega_{in}^2}^j \end{aligned} \tag{2.12}$$

Using these solutions, the damping rates for each mode of vibration can be determined.

2.1.5 Real-Time Synthesis

For real-time synthesis, a preprocessing step is first performed for a given object and material. In this step, the eigendecomposition is performed and the resulting $\mathbf{\Phi^T}$ and each mode's d_i and ω_{id} are saved.

At runtime, an applied force $\mathbf{f}$ is transformed to mode space by $\mathbf{\Phi^T}$, and the resulting vector contains the amplitudes with which to excite each mode. The resulting damped sinusoids can be combined and sampled at 44.1 kHz to produce the sound itself. Tools for performing additive synthesis and modal sound synthesis are plentiful; examples include the Synthesis ToolKit (Cook and Scavone, 1999) and the Faust programming language (Michon et al., 2017; Michon and Smith, 2011).

For interactive applications, as a user performs actions to create sounds, sound synthesis algorithms must run fast enough to generate sound in real time. The computation requirements at runtime are proportional to the complexity of the analyzed input shape, making some objects' sounds too slow for real-time applications without optimizations. Vibration modes can be culled based on psychoacoustic principles, for example, humans cannot tell the difference between two frequencies very close to one another, so those modes can be combined into one (Raghuvanshi and Lin, 2006). If an object has any geometric symmetries, these can be exploited to reduce memory usage and caching requirements (Langlois et al., 2014). Synthesis can be done in frequency space to further improve performance (Bonneel et al., 2008). When performing real-time synthesis, vectorization (van Walstijn and Mehes, 2017) and parallelism on CPUs (Bilbao et al., 2013) and GPGPUs (Webb, 2014) are effective, as each mode of vibration can be synthesized independently.

2.1.6 Additional Factors

Modal sound synthesis roughly simulates the sounds produced by rigid, vibrating objects, but in the real world more factors influence the final sound we hear; four such examples are acoustic radiance, sound propagation effects, contacts with other objects, and acceleration sound.

Acoustic radiance is the efficiency of propagation for each mode: depending on the shape of an object some modes radiate in different directions with different strengths (James et al., 2006; Li et al., 2015). More generally, any sound source can be directional, requiring additional simulation considerations (Mehra et al., 2014). Once the vibrations transfer to the surrounding air, sound waves bounce around the environment before reaching a listener's ears.

Sound propagation refers to this propagation of sound waves through air. Propagation can be simulated most realistically with wave-based simulation (Raghuvanshi et al., 2009), though for use them in interactive applications these methods have heavy precomputation and storage requirements (Mehra et al., 2015, 2013; Raghuvanshi et al., 2016, 2010). Geometric methods for sound propagation are less accurate for low frequencies, but faster to compute for interactive applications (Savioja and Svensson, 2015; Chandak et al.,

2008; Schissler and Manocha, 2016, 2011), as long as diffraction can be properly simulated (Tsingos et al., 2001; Svensson et al., 1999; Rungta et al., 2018). Hybrid methods use geometric propagation for higher frequencies and wave-based methods for low frequencies heavily affected by wave effects (Hampel et al., 2008; Southern et al., 2011; Yeh et al., 2013). Some work has achieved tight coupling between sound synthesis and propagation (Rungta et al., 2016; Wang et al., 2018).

Contacts with other objects are common as objects rarely float in midair. These contacts with other objects modify the produced sound and can be accounted for with contact models (O'Brien et al., 2002; Zheng and James, 2011). Interactions between a sounding object and a striking tool can be modeled to better simulate the attack of the sound (Avanzini and Rocchesso, 2001; Bilbao et al., 2015). Contact modeling can be exploited to create real objects that vibrate only at desired frequencies. An object can be placed on foam blocks, specifically positioned to damp out the undesired frequencies while leaving the desired frequencies alone (Bharaj et al., 2015).

Continuous interactions between objects, such as sliding and scraping, require additional effort. Fractal noise is a common way of representing the small impacts generated during rolling and scraping (Doel et al., 2001). Ren et al. presented a framework for synthesizing contact sounds between textured objects (Ren et al., 2010). This work introduced a multi-level model for lasting contact sounds combining fractal noise with impulses collected from the normal maps on the surfaces of the objects. However, this application of normal maps to sound generation without similar application to rigid-body dynamics causes noticeable sensory conflict between the produced audio and visible physical behavior.

Acceleration sound is produced when an object is rapidly accelerated through air, and is perceptually noticeable for very small objects such as dice and keys (Chadwick et al., 2012).

2.1.7 Full Physically-Accurate Simulation

To emphasize the restrictions that must be made for real-time sound synthesis, consider the case of dominoes falling on a table, as seen in Figure 1.3. A full and physically-correct simulation would need to consider all of the above factors. Normal modes of vibration are simulated by modal sound synthesis, the perceptually-dominant factor that this dissertation focuses on. Complex modes and acoustic radiance would need to be simulated for a complete model of object vibrations. Acceleration noise may be perceptually noticeable for these small dominoes. Accurate contact modeling for this scene would be important for this

scene given the stacking structure of the fallen dominoes. Sound propagation would be necessary not just to model the acoustics of the room, but also to model the interactions between objects.

This theoretical full simulation would require tight coupling between each objects' interior deformations, inter-object forces, and the air pressure/velocity fields. Some of this could be achieved with a wave-based simulator (Wang et al., 2018) and accurate contact model (Zheng and James, 2011). However, simulation of these factors is too computationally-intensive for real-time sound synthesis. Therefore, for real-time applications such as virtual environments, we are limited to modal sound synthesis and approximate sound propagation (James et al., 2006).

2.2 Multimodal Interaction with Virtual Objects

Multimodal interaction, in the context of this dissertation, refers to interaction using multiple senses simultaneously. The senses of sight, hearing, and touch are each different *interaction modalities*, which have been independently researched. In this section, I discuss prior work related to texture mapping and each of these additional modalities of interaction. As one of the main contributions in this dissertation is a method for multimodal interaction with *textured* surfaces (Chapter 5), much of the background in this section focuses on surface interactions.

2.2.1 Human Auditory Perception

Since this work focuses on audio, rendering for the sense of hearing has been discussed earlier inSection 2.1. However, as sound is inherently perceptual, studies of human auditory perception provide important clues about perceptually important parameters. Studies have evaluated which parameters humans rely on for material identification, finding that damping rate and frequency (i.e. pitch) are particularly important (Klatzky et al., 2000; McAdams et al., 2010). Studies have tested the discriminability of materials and the generalizability of the Rayleigh damping model (Ren et al., 2013a). Similar studies have focused on perception of object size from sound (Giordano and McAdams, 2006; Grassi, 2005). Material perception is also affected by concurrent visual stimuli (Fujisaki et al., 2014).

2.2.2 Visual Rendering

Realistic visual rendering has been the focus of the computer graphics field for many decades, and photo-realistic visual appearances are possible given talented artists and sufficient computational resources. Many books provide an introduction to the field (Akenine-Moller et al., 2002; Foley et al., 1990). Creating realistic visual appearances in interactive environments in real time is more challenging, but can be accomplished using optimizations.

For example, *texture mapping* uses low-resolution 3D triangle meshes with higher-resolution 2D textures to model detailed objects. Normal maps and relief maps are used as representations of fine detail of the surface of objects. Normal maps were originally introduced for the purposes of bump mapping, where they would perturb lighting calculations to make the details more visibly noticeable (Blinn, 1978). Relief mapping uses both depths and normals for more complex shading (Oliveira et al., 2000; Policarpo et al., 2005). Numerous other texture mapping techniques exist as well. Displacement mapping, parallax mapping, and a number of more recent techniques use height maps to simulate parallax and occlusion (Cook, 1984; Kaneko et al., 2001; Tevs et al., 2008). A recent survey goes into more detail about many of these techniques (Szirmay-Kalos and Umenhoffer, 2008). Mapping any of these textures to progressive meshes can preserve texture-level detail as the level-of-detail (LOD) of the mesh shifts (Cohen et al., 1998).

2.2.3 Rigid-Body Simulation

Simulation of the movement and collisions between rigid objects allows virtual environments to simulate gravity and user behavior such as stacking and throwing objects (Featherstone, 2007). Height maps mapped to object surfaces have been used to modify the behavior of simple collisions in rigid-body simulations (Nykl et al., 2013). When height maps are applied to two colliding objects, previous methods can effectively compute and resolve their collision (Otaduy et al., 2004).

2.2.4 Haptic Rendering

Haptics refers to interaction using the sense of touch, and focuses on the textures of surfaces (Loomis and Lederman, 1986). There has been significant work on how humans perceive haptic sensations to recognize shapes and textures (Lederman and Taylor, 1972; Klatzky et al., 1985). In haptic rendering, a 3D object's geometries and textures can be felt by applying forces based on point-contacts with the object (Basdogan et al.,

1997; Ho et al., 1999). Complex objects can also be simplified, with finer detail placed in a displacement map and referenced to produce accurate force *and torque* feedback on a probing object (Otaduy et al., 2004). The mapping of both normal and displacement maps to simplified geometry for the purposes of haptic feedback has also been explored (Theoktisto et al., 2010). Dynamic deformation textures, a variant of displacement maps, can be mapped to create detailed objects with a rigid center layer and deformable outer layer. The technique has been extended to allow for 6-degree-of-freedom (DOF) haptic interaction with these deformable objects (Galoppo et al., 2007). A common approach to force display of textures is to apply lateral force depending on the gradient of a height map such that the user of the haptic interface feels more resistance when moving "uphill" and less resistance when moving "downhill" (Minsky et al., 1990; Minsky, 1995).

2.2.5 Integrated Multimodal Interaction

For realistic multimodal interaction, it is important that content is not only rendered well for individual senses, but that each sense is consistent with one another. Between audio and rigid-body simulation, modal sound synthesis can be coupled with physics simulations to couple the movements of objects and their resulting sounds (O'Brien et al., 2002; Zheng and James, 2011). Depth maps can modify contacts between objects, coupling the visual appearances of the objects with their physical movements (Nykl et al., 2013). When multiple objects are in contact, the long-lasting contacts produce continuous sounds which depend heavily on the objects' textures, further coupling motion and sound (Ren et al., 2010). Between audio and haptics, some work has considered the multimodal aspects of touch-enabled interfaces for sound synthesis (Ren et al., 2012). These methods involve only one or two interaction modalities each, and do not use a single representation of surface detail to inform all modalities.

2.3 Auditory Understanding

The inverse of the modal sound synthesis problem is to use impact sounds to understand the objects that created those sounds. In this dissertation, I present multiple methods for estimating properties of real-world objects from recorded sounds. In this section, I will review the broad area of processing sounds to learn something about the sound's source. I will begin with discussion of general concepts and methodology, then focus in on object sounds.

2.3.1 Environmental Sound Classification

One broad way of approaching sound understanding is to classify sounds into descriptive categories. Multiple datasets have been established for evaluating classification of various environmental sounds (Gemmeke et al., 2017; Piczak, 2015b; Salamon et al., 2014). Traditional techniques use a variety of features extracted from sounds, such as Mel frequency spectral coefficients and spectral shape descriptors (Büchler et al., 2005; Cowling and Sitte, 2003). Similar approaches are used to classify an environment based on the sounds heard within it (Barchiesi et al., 2015).

Convolutional neural networks have also been applied to these problems, producing improved results (Piczak, 2015a; Salamon and Bello, 2017). Recently, some interest has been given to exploring the performance of different network structures (Hershey et al., 2017; Huzaifah, 2017). Impact sounds are a specific category of environmental sounds, which contain fewer cues to differentiate them from one another.

2.3.2 Statistical Sound Modeling

Statistical methods have found applications in summarizing and analyzing sound. The late reverberations of sounds in rooms have been modeled as Gaussian noise, whose summary statistics convey properties of the environment (Traer and McDermott, 2016). It has also been found that humans inherently use summary statistics to understand sounds (McDermott et al., 2013). Previous methods for material parameter estimation assume minimal variable effects in estimation of damping rates. In comparison, the probabilistic damping model presented in Chapter 4 models recorded impact sounds as inherently stochastic.

2.3.3 Object Understanding Through Sound

I now shift the focus to understanding of objects' impact sounds in particular. A common application is to learn properties of a real-world object in order to *resynthesize* similar sounds in a virtual environment. Some methods use a single recorded sound, then apply modifications to create realistic variety in resynthesized sounds. Deterministic features of a sound can be extracted, then stochastic noise can be added to those features to model slight variations (Serra and Smith, 1990). Alternatively, the modal content of a sound can be extracted, then resynthesized, slightly modifying mode amplitudes to create variations (Lloyd et al., 2011). Other methods use multiple input sounds for a single object, generated by striking the object in

known locations. The sounds' spectral content can be interpolated spatially to approximate hit points at new locations (Pai et al., 2001).

However, these methods work by modifying recorded sounds without gaining much fundamental knowledge about the object itself. They do not model the object's material or shape independently from one another. Ideally, sounds from struck real-world objects could be used to recreate the shape and material parameters of the objects. In the rest of this section, I will consider both independent material and shape estimation.

2.3.3.1 Auditory Material Estimation

Material parameters can be estimated experimentally with specialized measurement equipment (E756, 2017), but impact sounds do not require specialized equipment or trained personnel to record. The Young's modulus for small parts of the object can be individually optimized to best match input sounds (Yamamoto and Igarashi, 2016). The most relevant work is that of Ren et al. (Ren et al., 2013b), which performs automatic estimation of material parameters from a single audio sample. This method works by optimizing a synthetic sound to most closely match the recorded sound. The material parameters that optimally match the synthetic and recorded sounds are the most likely material parameters for the real-world object. The estimated parameters can be applied to synthesis of sounds for any object with that material. However, their method is able to estimate damping parameters only for the Rayleigh damping model, which may be limited in its ability to represent diverse materials.

These methods are often limited in their robustness by relying on Rayleigh damping, not accounting for environmental factors, and not using multimodal input. These limitations are addressed as part of this dissertation. Methods that estimate material damping parameters support only the Rayleigh damping model, which I address in Chapter 3. All of these methods assume that properties of the recording environment are known or are assumed to be minimal, which I address in Chapter 4. If both video and audio of the object are available, these methods have no way of using the visual information to improve material estimates, which I address in Chapter 6.

2.3.3.2 Auditory Shape Estimation

The ideal case of using one sound to reconstruct an entire object is known to be underconstrainted (Kac, 1966), but prior research has explored what information can be estimated under different constraints. For example, binary shape attributes such as planarity and mirror symmetry, may be easier to estimate than a full

geometric model (Fouhey et al., 2016; Fouhey et al., 2019). Sound can be used as a source of information for deeper understanding of 3D object structure. Little work has been done in this area, and existing methods limit themselves to estimation of shape attributes (Zhang et al., 2017b). Zhang et al. evaluated the ability of the ShapeNet neural network (Aytar et al., 2016) to identify an object shape out of 14 possible shapes, but these were largely shape primitives that were not representative of real-world object geometries (Zhang et al., 2017a). In Chapter 6, I address these limitations with a method that uses multimodal input and evaluates on datasets of shapes more representative of the real world.

2.4 Visual Understanding

Having discussed machine understanding of sound, I now discuss what information can be gleaned through visual means.

2.4.1 Visual Object Reconstruction

An important step in object virtualization is to obtain the 3D shape and visual surface texture of a real-world object. This information can be obtained by reconstructing the object from a series of images looking at the object from multiple angles. Structure from Motion (SFM) (Westoby et al., 2012; Snavely et al., 2006), Multi-View Stereo (MVS) (Goesele et al., 2007; Seitz et al., 2006), and Shape from Shading (Zhang et al., 1999) are classes of techniques for obtaining 3D shape information from a set of 2D images. Although these methods alone do not achieve a segmented representation of the objects within the scene, they serve as a foundation for many algorithms. Bundle adjustment is used to jointly optimize poses when many images are used as input (Triggs et al., 2000). RGB-D depth-based, active reconstruction methods can also be used to generate 3D geometrical models of static (Newcombe et al., 2011; Golodetz* et al., 2015) and dynamic (Newcombe et al., 2015; Dai et al., 2017) scenes using commodity sensors such as the Microsoft Kinect and GPU hardware in real-time.

2.4.1.1 3D Object Datasets

With the rise of data-driven methods for visual understanding, large and well-annotated datasets have become valuable. Thanks to a plethora of 3D scene and object datasets such as BigBIRD(Singh et al., 2014) and RGB-D Object Dataset (Lai et al., 2011), neural network models have been trained to label objects based

on their visual representation. 3D ShapeNets (Wu et al., 2015) also provides two sets of object categories for object classification referred to as ModelNet10 and ModelNet40, which are common benchmarks for evaluation (Kanezaki et al., 2016). Scene-based datasets have also been built from RGB-D reconstruction scans of entire spaces, allowing for semantic data such as object and room relationships. For instance, NYU Depth Dataset (Silberman et al., 2012) and SUNCG (Song et al., 2017) enable indoor segmentation and semantic scene completion from depth images.

2.4.2 Multimodal Understanding

It is well known that vision alone is limited in its ability to understand scenes. In this dissertation I focus on using audio as a primary cue for improved object understanding, though many other modalities have also been explored. Here, I will discuss prior work on multimodal understanding.

Additional input modalities may improve results for objects and materials that are difficult to reconstruct. Reflective objects have glare which change in location with the movement of the viewer, while transparent objects make it difficult to determine depth. A time-of-flight camera can correct estimated depth of transparent objects (Tanaka et al., 2017). The dip transform for 3D shape reconstruction (Aberman et al., 2017) uses fluid displacement of an object to obtain shape information.

Sound and video are intrinsically linked modalities for understanding the same scene, object, or event. Using visual and audio information, it is possible to predict the sound corresponding to a visual image or video (Owens et al., 2016b; Aytar et al., 2016). Sound prediction from video has also been specifically explored for impact sounds (Owens et al., 2016a).

Impact sound provides an additional input modality, containing cues about the internal structure of an object. Environmental scene classification is a related task approached through spectral analysis (Büchler et al., 2005) or convolutional neural networks (Piczak, 2015a; Salamon and Bello, 2017), but produces broad classifications of an entire environment. However, no current methods use impact sounds in particular to aid in complete shape reconstruction. One goal of this research is to use impact sounds to help determine the object shape and material in cases where visual methods struggle.

2.4.2.1 Multimodal Fusion

Other works have fused audio and visual cues to better understand objects and scenes. Sparse auditory clues can supplement the ability of random fields to obtain material labels and perform segmentation (Arnab

et al., 2015). Neural networks have proven valuable in fusing audio-visual input to emulate the sensory interactions of human information processing (Zhang et al., 2017b). While multimodal methods have succeeded in fusing input streams to capture material and low-level shape properties to aid segmentation, they have not attempted to identify specific object geometries.

Early attempts at multimodal fusion in neural networks focused on increasing classification specificity by combining the individual classification results of separate input streams (Simonyan and Zisserman, 2014). Bilinear modeling can model the multiplicative interactions of differing input types, and has been applied as a method of pooling input streams in neural networks (Tenenbaum and Freeman, 2000; Lin et al., 2015). Bilinear methods have been further developed to reduce complexity and increase speed, while other approaches to modeling multiplicative interactions have also been explored (Gao et al., 2016; Yu et al., 2017; Park et al., 2016). Bilinear methods have not yet been applied to merging audio-visual networks. The ISNN networks I present in Chapter 6 take a step towards combined audio-visual object reconstruction with bilinear methods.

CHAPTER 3: Interactive Modal Sound Synthesis Using Generalized Proportional Damping[1]

3.1 Introduction

Modal analysis requires a model of the rigid object and a set of material parameters. These material parameters are tedious to set by hand, but determine whether the object sounds like glass, metal, or another material. One necessary component of sound synthesis is the damping model, which characterizes how the amplitude of the sound decays over time. Damping is a complex phenomenon, and it can be difficult to determine exactly how the vibrations of a modeled object will decay. Additionally, the presence of damping may give rise to *complex* modes of vibration, which are more difficult to model than *normal* modes (Caughey and O'Kelly, 1965). The most common approach is to assume all damping is viscous and to approximate the decay rate of one part of an object as a linear combination of its mass and stiffness. This model is referred to as Rayleigh damping or linearly proportional damping, and produces only normal modes. It is the de-facto technique for modeling damping using modal sound synthesis. Rayleigh damping uses a simple linear model, but there are known limitations about the damping of sound synthesized even using properly set Rayleigh damping coefficients.

The limitations are: (1) Rayleigh damping is only a first-order approximation and (2) it was originally chosen for its ease of computation, not its physical accuracy. Other damping models are common in material and structural analysis, but have not been thoroughly examined in computer graphics for interactive 3D sound synthesis. The most general damping model to date that limits vibrations to normal modes is *generalized proportional damping* (GPD) (Adhikari, 2006), of which Rayleigh damping is a special case. These alternative damping models may be able to improve sound quality by providing a better fit to the real-world damping behavior. By improving the quality of synthesized sound, we can enhance the immersion in virtual environments to create more effective 3D games, telepresence applications, and training simulations.

[1]This chapter previously appeared as a paper in the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D 2016). The original citation is as follows: Sterling, A. and Lin, M. C. (2016b). Interactive modal sound synthesis using generalized proportional damping. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '16, pages 79–86, New York, NY, USA. ACM

In this chapter, we explore the use of generalized proportional damping for interactive modal sound synthesis. We first present how GPD can be integrated into current methods for modal sound synthesis. We describe a method for deriving damping models from the larger space of GPD functions and propose specific models that may be of interest for modal sound synthesis. We further extend an optimization framework originally designed to compute Rayleigh damping parameters given audio samples to compute material parameters for the GPD model. Finally, we conduct a preliminary user study to evaluate the perceptual differences between multiple damping models in modal sound synthesis.

To sum up, the main results include:

- Investigation of higher-order generalized damping models for modal sound synthesis (Section 3.2);
- Estimation of material parameters for Generalized Proportional Damping in sound rendering (Section 3.3); and
- Evaluation, comparison, and analysis of perceived audio quality using these GPD models (Section 3.4).

3.2 Generalized Proportional Damping for Sound Synthesis

Generalized proportional damping (GPD), introduced by Adhikari (Adhikari, 2006), extends the damping models previously discussed in Section 2.1.2. While Rayleigh and Caughey damping are models parameterized by real valued coefficients α_j, GPD is parameterized by functions. Rayleigh and Caughey damping can both be derived as GPD models, but GPD is also able to capture a wider variety of damping behavior.

3.2.1 Generalized Proportional Damping

Generalized proportional damping is formulated as follows:

$$\mathbf{C} = \mathbf{M}\beta_1(\mathbf{M}^{-1}\mathbf{K}) + \mathbf{K}\beta_2(\mathbf{K}^{-1}\mathbf{M}) \tag{3.1}$$

β_1 and β_2 are matrix valued functions whose only restrictions are that they be analytic near the eigenvalues of their arguments. For example, using $\beta_1(\mathbf{A}) = \alpha_1\mathbf{A}$ and $\beta_2(\mathbf{A}) = \alpha_2\mathbf{A}$ replicates Rayleigh damping. This representation is much more convenient to work with than a Caughey series, as arbitrary functions can be easily plugged in to the β functions. GPD still satisfies the necessary condition of the Caughey series since any continuous function used as a β can be expanded as a power series.

For GPD, the equation for c_i was provided along with a lengthier proof (Adhikari, 2006), which we will omit here:

$$\begin{aligned}\mathbf{\Phi}^T\mathbf{C}\mathbf{\Phi} &= \mathbf{M}\beta_1(\mathbf{M}^{-1}\mathbf{K}) + \mathbf{K}\beta_2(\mathbf{K}^{-1}\mathbf{M}) \\ c_i &= \beta_1(\omega_{in}^2) + \omega_{in}^2\beta_2(\omega_{in}^{-2}) \\ c_i &= \beta(\omega_{in}^2)\end{aligned} \tag{3.2}$$

The final form of the equation can be reached without loss of generality (the second term could be embedded in β_1) and is an even more convenient form to work with.

3.2.2 Modal Sound Synthesis with GPD

The technical change needed to use GPD for modal sound synthesis is conceptually simple: during precomputation of damping coefficients use Equation (3.2) instead of Equation (2.11). GPD's increased flexibility has its downsides: with Rayleigh damping it is tedious, but possible, to select the parameters α_1 and α_2 by hand and fine tune until the resulting sound is acceptable. The challenge now lies in selecting an appropriate β function for the sounding object in question, which covers a much broader space of functions.

β defines a curve in eigenvalue-damping space, which should match as closely as possible to the real-world damping values. Considering damping modeling as a curve fitting problem, Rayleigh damping's linear model is only accurate as long as the true damping curve remains approximately linear.

3.2.2.1 Power Law Model

Our proposed solution is to pick functions parameterized with real-valued coefficients known to provide good fits to damping curves. Rayleigh and Caughey damping use real-valued coefficients and stay in the toolkit, but it opens up the possibility of other models. As one alternative model, in the study of sound attenuation during propagation there is a well-known power law relation between frequency and attenuation (Szabo, 1994). As sound propagates through a material, the pressure of the sound P attenuates depending on the distance traveled Δx and frequency ω according to:

$$P(x + \Delta x) = P(x)e^{-\alpha_1(\omega)_2^{\alpha}\Delta x} \tag{3.3}$$

α_1 and α_2 are real-valued coefficients which vary depending on material. If we assume the physical phenomenon causing attenuation over distance and damping over time are similar, we can use Equation (3.2) to derive a similar damping model based on this power law:

$$c_i = \beta(\omega_{in}^2) = \alpha_1 \omega_{in}^{2\alpha_2} \tag{3.4}$$

α_1 and α_2 are now the real-valued parameters to this power law model. While the 2 in the exponent could be incorporated into α_2, it allows the function to be written in terms of the eigenvalue for clarity. Because this model is a continuous function of the eigenvalue, GPD guarantees that there is a damping matrix $\mathbf{C}$ that diagonalizes to produce these c values and therefore creates only normal modes of vibration.

Empirical findings for the power law model's α_2 in the context of attenuation place it in a range between 0 and 1, with 1 being a common finding for many materials. If damping can be said to be similar, this may provide some physical justification for Rayleigh damping, whose second term fits this model. However, Rayleigh damping could not handle any materials with an exponent not equal to one while a power law damping model could adapt for each material. We use this power law model in later evaluation, but GPD allows for a wide range of models, and we would encourage trying out different models to find optimal fits. We demonstrate one such additional model in Section 4.2.1: a hybrid model combining Rayleigh and power law models, though this chapter focus on the two separately.

3.3 Material Parameter Estimation

Instead of fine-tuning damping model parameters values by hand, we can instead automatically estimate them from recorded audio. Rayleigh damping has been studied to determine that its α_1 and α_2 are geometry-invariant and can be considered as properties of the *material* alone (Ren et al., 2013a). Other damping models have not undergone the same level of rigorous testing, but we hypothesize that for any damping model with real-valued parameters, the parameters will be similar across objects with different shapes and the same material. Ideally, we would like to use the recorded audio to estimate all the material parameters needed to synthesize sound of an object in any shape but made of the same material. The relevant material parameters are Young's modulus, Poisson's ratio, density, and damping model parameters for the chosen damping model.

Ren et al. have suggested an optimization-based framework for estimating these parameters using a Rayleigh damping model (Ren et al., 2013b). We extend the optimization framework to automatically

identify material parameters for any damping model, including both Caughey series and GPD models. In this section, we first review the parameter identification pipeline for turning an audio recording into example-guided, physically-based synthesized sound. We then describe how we generalize such a system to estimate parameters for alternative damping models.

3.3.1 Estimation of Rayleigh Coefficients

3.3.1.1 Feature Extraction

First, the input audio file is processed to extract audio *features*, where each feature represents a single damped mode and consists of a frequency, damping coefficients, and initial amplitude. Multiple power spectrograms of the input audio are constructed with varying temporal and spatial resolution, and frequencies with high power are selected. The spectrograms with high temporal resolution have low spectral resolution and will be useful in different situations than the spectrograms with low temporal resolution and high spectral resolution. Once a peak is identified, an optimizer searches the local variations in frequency, damping, and amplitude to produce the best fit. The power spectrogram of the new peak is subtracted from the current spectrograms and the process repeats until a large enough percentage of the power is accounted for in the extracted features. The remaining audio is the *residual* audio, containing background noise and nonlinear effects such as complex modes.

3.3.1.2 Parameter Estimation

In order to estimate the material parameters of the recorded object, some additional information is needed. Poisson's ratio is not optimized as part of this system, so it must be predetermined before starting. Additionally, eigenvalues scale proportionally with Young's modulus (E) and inversely with density (ρ). The ratio of Young's modulus to density is referred to as the *specific modulus* $\gamma = E/\rho$. If parameter estimation is intended to estimate a Young's modulus, a density value needs to be predetermined in order to get an absolute Young's modulus. Finally, modal analysis is performed on a discretized model of the object with assumed material parameters and it is struck with a unit impulse at the same hit point as the real-world object. The resulting $\mathbf{\Phi}^T f$ contains the initial mode amplitudes of the assumed object, and since the same hit point was used for the recorded object, its amplitudes should be a scaled version of the same. The final set of parameters

used in optimization is γ, scale, and real-value coefficients. Scale is not a *material* parameter, but without it the optimizer would be unable to properly match the volumes of the recorded and reconstructed audio.

The parameters are determined through optimization looking to minimize a similarity metric by varying the parameters. The chosen metric combines both evaluation of differences between power spectrograms and differences between features. Power spectrograms are compared after being transformed based on psychoacoustic principles. Since humans cannot easily distinguish between similar frequencies of sound, and since this effect varies in strength across the range of hearing, the frequency dimension is transformed to the Bark scale which properly accounts for this effect. Perception of loudness also varies based on frequency, so the intensities are converted to the sone scale, in which the loudness is scaled depending on the frequency of the sample. With the spectrograms converted to perceptually-based scales, they can now be compared to one another by finding the squared difference between them:

$$\Pi_{psycho}(\mathbf{I}, \bar{\mathbf{I}}) = \sum_{m,z} \left(\mathbf{T}(\mathbf{I})[m,z] - \mathbf{T}(\bar{\mathbf{I}})[m,z]\right)^2 \tag{3.5}$$

The other part of the metric operates on (frequency, damping, amplitude) features extracted from the recorded audio or taken from an assumed mode of vibration and its corresponding entry in the $\mathbf{\Phi}^T f$ amplitude vector. Once again, the frequency is converted to the Bark scale for psychoacoustic purposes. The damping is also inverted to become duration, which is less sensitive to differences between very short bursts of sound. The sets of features are then matched with one another using the *Match Product Ratio* metric. A single feature f_1 can be compared to the set of possible matches in the other set of features $\bar{\mathbf{f}}$ using the *point-to-set* match ratio:

$$R(f_i, \bar{\mathbf{f}}) = \frac{\sum_j u_{i,j} k(f_i, \bar{f}_j)}{\sum_j u_{i,j}} \tag{3.6}$$

u is a matrix of weights to give higher priority to prominent features, while k is a measure of distance between the two points on $[0, 1]$ such that a 1 means an exact fit. A full set of features $\mathbf{f}$ can be compared to another set of features $\bar{\mathbf{f}}$ using the *set-to-set* match ratio:

$$R(\mathbf{f}, \bar{\mathbf{f}}) = \frac{\sum_i w_i R(f_i, \bar{\mathbf{f}})}{\sum_i w_i} \tag{3.7}$$

w is a vector of weights similar in purpose to u. With these match ratios defined, the Match Ratio Product metric for the extracted audio features $\mathbf{f}_{extract}$ and the assumed audio features $\mathbf{f}_{assume}$ is:

$$\Pi_{MRP} = -R(\mathbf{f}_{extract}, \mathbf{f}_{assume})R(\mathbf{f}_{assume}, \mathbf{f}_{extract}) \tag{3.8}$$

The final metric to optimize is:

$$\Pi_{hybrid} = \frac{\Pi_{psycho}}{\Pi_{MRP}} \tag{3.9}$$

This metric takes into account both the power spectrograms and features, using psychoacoustic scales where it can better match human hearing.

The starting points are generated by choosing multiple pairs of two dominant features extracted from the recorded audio and fitting a line to them to generate starting material parameters. For each mode from the modal analysis on assumed parameters, the eigenvalue and the corresponding amplitude in the $\mathbf{\Phi}^T f$ vector are used to generate the starting γ and scale values. The starting γ is selected as the value that would cause the selected mode to have the same frequency as one of the dominant features. Similarly, the starting scale is the one that would scale the amplitude of the selected mode to the amplitude of the dominant feature. Together, these define a starting point for the optimizer.

By running a non-gradient based optimizer on this metric from each starting point and selecting the best final point, material parameters that best recreate the original sound are selected. The resulting γ can be used to find the Young's modulus, while the material parameters, such as α_1 and α_2 in Rayleigh model, define the damping curve. These parameters can then be transferred to other geometries, effectively applying the material parameters of the original recorded object to different virtual models.

Ren et al. also presented a method for taking the residual sound (anything not captured by the modal feature extraction) and transferring it to alternative shapes, making even the residual somewhat geometry-invariant (Ren et al., 2013b). We focus on the estimation of damping parameters and we do not adopt residuals for sound synthesis.

3.3.2 Estimation of GPD Parameters

In order to estimate damping parameters from an arbitrary damping model, we reformulate the set of optimized parameters to include γ, scale, and all of the damping model parameters (of which there could be

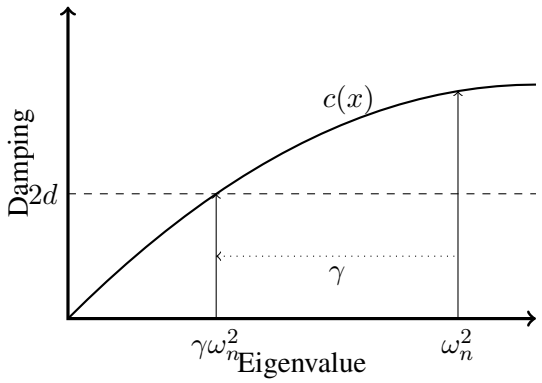


Figure 3.1: Generation of specific modulus γ starting value using a damping value d from an extracted feature, an eigenvalue ω_n^2 from modal analysis on assumed parameters, and a sampled damping model $c(x)$. γ is chosen such that $c(\gamma\omega_n^2) = 2d$.

many). Any instances of Rayleigh damping computation are replaced with a general β function (instead of real values). Feature extraction and metric evaluation are still applicable, as the damping model plays no role there.

The most significant difference lies in generation of the starting points. With Rayleigh damping's linear fit, any two points define a new line and new starting α_1 and α_2, but arbitrary β functions may require many points to define a curve. Instead, we repeatedly sample a customizable percentage of the dominant features—weighted by dominance. On each sample, we perform least-squares nonlinear regression on the damping model to create the starting damping model parameter values. The sampling percentage is ideally set such that there are enough features to get a useful fit, but not so many features that the starting points are tightly clustered.

To generate a starting γ, we pair up each mode's eigenvalue from the modal analysis on assumed parameters with each damping value from extracted features. γ is computed through root-finding as the value that maps the mode's eigenvalue to the feature's damping value through our sampled damping model. We are effectively asking, "If this eigenvalue happened to be damped at this rate, what would γ have to be?" See Figure 3.1 for a visual example. Similarly, the scale is chosen to match the mode's amplitude to the extracted feature's amplitude. This is a fairly exhaustive search and the search space has many local minima, so quite a few starting points are needed to find a nearly-global minima. Once these starting points are generated, the optimizer can proceed to minimizing the metric.

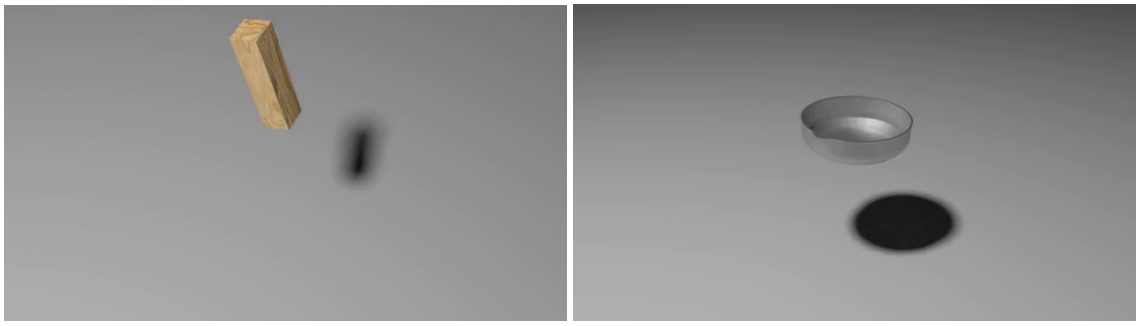


Figure 3.2: Virtual reconstructions of virtual objects, placed in a scene where they fall onto a ground plane to produce impact sounds.

3.4 Results

3.4.1 Sound Synthesis

We implemented Rayleigh, Caughey, and GPD-based sound synthesis using FEM meshes as our discretization. Audio was played using the STK library (Cook and Scavone, 1999; Scavone and Cook, 2005), and videos were created using Blender with Bullet Physics for rigid-body simulation (Coumans, 2015).

Our meshes contain around 10,000–20,000 tetrahedra, resulting in up to 30 minutes of precomputation time on modal analysis using a desktop workstation computer. Run time for material parameter estimation is most dependent on the length of the sound: one starting point for a short impact converges in a few seconds, while a reverberative object requires up to ten minutes. At runtime, sound is synthesized at 44 kHz: the highest frequency we can perceive is around 22 kHz, so there is no benefit in synthesizing sound more often than twice that frequency. The synthesis steps are fast enough that the 44kHz update rate can be easily maintained for a number of sounding objects even on a laptop.

3.4.2 Parameter Estimation

We implemented our extended version of the material parameter optimization process, and have been able to estimate parameters using different damping models. See Figure 1.2 for the full set of objects used, comparing the real objects in the top row to the meshes in the bottom row. Figure 3.2 shows a few of these objects placed into a virtual scene as part of videos for our user study (Section 3.4.3). Figure 1.3 shows a set of dominoes of different materials, which collide with one another to produce a variety of sounds.

Table 3.1 presents some results from performing estimation of material damping parameters given recorded audio, using Rayleigh damping, second-order Caughey damping, and power law damping. These

		Plastic 1	Plastic 2	Porcelain	Wood	Aluminum
Shared	E	8	2.4	20	9.9	0.88
Rayleigh	α_1	125	58	189	35	.225
	α_2	8e-6	1e-6	1.5e-8	4.6e-7	1.45e-6
Caughey	α_1	280	85	420	277	9.7
	α_2	-3.6e-7	6.6e-7	-2.4e-6	-2.0e-6	-3.4e-6
	α_3	2.0e-15	5.6e-14	3.8e-15	4.8e-15	5.8e-13
Power	α_1	1.13	.19	163	6.7	.02
	α_2	.3	.37	.01	.18	.445

Table 3.1: Estimated material parameters for a selection of materials. Young's modulus is given in GPa. See Sections 2.1 and 3.2 for the usage of each damping parameter. These are not necessarily the parameters that minimize the MPR metric, but they are locally optimal and agree on a somewhat physically plausible Young's modulus E across damping models.

parameters may not be the most globally optimal as reported by the optimizer, but they are all parameters that have reasonably low metric values and are at least locally optimal. Plastic 1 comes from the rigid, clear plastic bowl, while Plastic 2 comes from the thin and much more flexible dog food scoop. We can assign some physical meaning to these parameters; for example Porcelain has smaller values for Rayleigh's α_2 and Power's α_2, indicating relatively less damping at higher frequencies. Also note that while most materials are best fit by a Caughey series whose coefficients alternate signs with each term, Plastic 2 was better fit by a set of only positive coefficients. The other damping models are unable to capture this unusual damping behavior as well as the higher-order Caughey series.

3.4.3 User Study

One hypothesis with this work is that alternative damping models can recreate a wider range of more realistic audio with more complex non-linear damping characteristics. In order to evaluate the perceptual realism of the damping models, we conducted a preliminary user study where subjects were asked to compare sound generated with different damping models. This study is a first exploration of the differences between damping models. The study evaluates if subjects can tell the difference between them, and if so, which they find more realistic.

3.4.3.1 User Study Setup

This study was conducted entirely online through the subject's web browser. Subjects were informed about the procedure of the study and instructed to use headphones or earbuds in order to better control the audio environment.

Subjects were presented with a series of pairs of videos of an object being dropped on a flat surface. Refer to Figure 3.2 for images of the objects used in the study. Each pair of videos showed the same visual imagery, but had different audio generated using either Rayleigh damping, a second-order Caughey series, or a power law model. Subjects were asked to rate, on a scale from 1 to 11, which video they perceived as more realistic, with a 1 indicating a strong preference for the video on the left, an 11 indicating a strong preference for the video on the right, and 6 being in the middle. Subjects were also asked to rate the similarity of the sound in the videos, where a 1 is very different and an 11 is indistinguishable. The videos could be watched repeatedly and subjects could return to previously-answered questions in case their opinions change.

3.4.3.2 User Study Results

40 subjects participated in the study, and while little demographic information was collected, the recruitment methods used were likely to attract many subjects with little experience in evaluating sound quality. We can begin by combining data from objects together to get a general sense of the perceived realism ratings as a whole. Recall that perceived realism was rated on a scale from 1 to 11, with 6 being in the middle. In comparisons between Rayleigh damped and Caughey damped audio, a 1 indicates preference for Rayleigh and an 11 indicates preference for Caughey. Across all objects, when subjects compared Rayleigh and Caughey damping, the realism rating was 6.5 ± 3.3, and there was not a significant preference in realism between the two ($p > .05$). When comparing Rayleigh to Power damping, where a 1 again indicates a preference for Rayleigh, the realism rating was 4.78 ± 2.57 and there was a preference for Rayleigh ($p < .0001$). Finally, when comparing Caughey to Power damping, where a 1 indicates a preference for Caughey, the average realism rating was 3.95 ± 2.92 and there was a preference for Caughey ($p < .0001$).

We can also look at the subject-reported similarity values to determine if the subjects could notice a perceptual difference between the models. Similarity was rated on a scale from 1 (very different) to 11 (very similar). In comparisons between Rayleigh and Caughey damping, the similarity was 5.7 ± 3.0. Between

Rayleigh and Power damping, the similarity was 6.9 ± 3.4. Finally, Caughey and Power damping had a similarity of 4.4 ± 2.7.

For more detailed results, realism values for each of the objects individually are laid out in Table 3.2. For simplicity, comparisons between two damping models, say Rayleigh and Power, are abbreviated as R/P in the table. The results for the small floor tile and the long wood block contain some of the most significant results, with Caughey damping greatly preferred over the other two. The plastic scoop was the only object for which Rayleigh was preferred over Caughey, but for most of the objects the difference was not statistically significant. The porcelain bowl is an interesting case where Rayleigh and Caughey are nearly identical in realism, but for once the power model is considered to be nearly as (possibly more) realistic.

3.4.3.3 Discussion

When compared to either of the alternatives, Power was perceptually considered to be less realistic by .47 standard deviations in the case of Rayleigh damping and by .7 standard deviations in the case of Caughey damping. This is only a moderate preference, but enough to be statistically significant. One simple explanation for this result is that a power law may not provide a good curve fit to the data. Despite this, the two most similar sounding damping models were reported to be Rayleigh and Power. The power law model often seems to be perceptually similar to Rayleigh damping at higher frequencies, while having less damping on the lower frequencies. In some cases the amplified lower frequencies sound more realistic, but in most of the cases in this user study it comes across as too strong and unrealistic.

In theory, Caughey damping can only improve upon Rayleigh damping since the higher order terms can simply be set to 0 if the linear model would be optimal. The result that the difference in realism between the two of them was not statistically significant could imply that the benefit gained from the second-order term is not be large enough to be perceptibly noticeable. However, the similarity rating between them is not particularly high, so a better interpretation might be that there *is* a perceptually noticeable difference between the sounds, but subjects had difficulty determining which of the two different sounds was more realistic.

Subjects did not have access to any ground truth sound recordings, which made the task more difficult. However, this is reasonable given that the primary application we are considering is using estimation parameters to synthesize contact sounds in interactive virtual environments. The study focuses on subjects' *perception* of the sounds presented in an entirely virtual environment to understand how they would react to these sounds in a game, virtual teleconference, or training simulation. The subjects only need to perceive

Object	Models	$\bar{x}$	σ	p
Metal Plate	R/C	6.5	3.2	.35
	R/P	*	*	*
	C/P	*	*	*
Plastic Bowl	R/C	6	3.3	1
	R/P	2.9	1.9	.0002
	C/P	3.9	2.0	.01
Plastic Scoop	R/C	3.6	2.2	.0035
	R/P	4.3	2.3	.095
	C/P	5.2	3.4	.294
Porcelain Bowl	R/C	6.1	3	.93
	R/P	6	0	1
	C/P	7.1	3.0	.32
Porcelain Plate	R/C	6.6	3.5	.53
	R/P	4.8	2.9	.16
	C/P	4.2	2.8	.04
Small Floor Tile	R/C	8.9	2.4	.001
	R/P	4.4	3.0	.13
	C/P	2.1	1.5	$<$.0001
Short Wood Block	R/C	6.6	4.2	.63
	R/P	5.2	3.6	.5
	C/P	4.0	2.9	.014
Long Wood Block	R/C	7.8	1.8	.005
	R/P	5.2	2.9	.34
	C/P	2.2	1.6	$<$.0001

Table 3.2: Realism values from the user study. For each object and each pair of damping models (R for Rayleigh, C for Caughey, P for power), the range of realism ratings is shown as a mean $\bar{x}$ and a standard deviation σ. Ratings lower than 6 are a preference for the damping model on the left side of the slash. The p-value evaluates whether there is a significant difference in realism preference from the "no preference" realism rating of 6. *The metal plate power model was not included in the study.

the sounds to be realistic. This perceptual approach also reduces the need for participants to be skilled at evaluating sound; the "perceptual ground truth" is not consistent between subjects.

3.5 Summary

We have presented the integration of generalized proportional damping with modal sound synthesis techniques. We explained how to derive GPD-based damping models, using a power law model as an example. We extended an existing method for estimating material parameters to estimate real-valued parameters from an arbitrary damping model. We conducted a preliminary user study comparing Rayleigh, Caughey, and power law damping models.

While the user study did not find an improvement in perceived realism when using our example of a GPD-based damping model, this result provides other benefits. This study provides additional validation of the popular Rayleigh damping model in that second order Caughey damping models were not always perceptually more realistic than Rayleigh damping and that GPD-based models that provide a perceptual improvement may not be easy to find. In light of this result, future research may find success in using models that encapsulate a larger function space. In Section 4.2.1, we do propose one such higher-order model. Additionally, genetic programming and neural nets can both approximate continuous functions without needing to specify a damping model in advance.

Future work in the area of GPD should likely focus on exploring alternative GPD-based damping models. Additionally, it would be an improvement to incorporate the residual audio after the material parameter estimation process, transferring it to other geometries. There is some uncertainty about the transferability of arbitrary GPD parameters; an analysis similar to the one done for Rayleigh damping (Ren et al., 2013a) could help determine if the real-valued model parameters can all be considered material parameters. Work in these areas would help improve understanding of damping behavior and hopefully lead to more immersive sound.

CHAPTER 4: Audio-Material Reconstruction for Virtualized Reality Using a Probabilistic Damping Model[1]

4.1 Introduction

Modal sound synthesis improves a user's immersion, but it requires accurate real-world material parameters. Damping, which determines the rate at which vibrations and sound decay over time, is crucial in differentiating between different materials. Some parameters, e.g. density and Young's modulus, can be looked up for known materials, but damping behavior can be difficult to identify and parameterize.

Traditionally, damping parameters are selected through laborious human hand-tuning. We present a study evaluating human efficiency and precision at this task in Section 4.3.2. Even with a simple, easy-to-use GUI optimized to minimize during modal analysis, the study shows that significant human effort is needed to select accurate parameters. The study also finds that humans are able to distinguish between sounds with minor differences in material parameters, suggesting that material parameters from a library may not sufficiently reproduce the sound of a specific real-world object.

Figure 4.1: A real-time interactive virtual environment where striking objects produces dynamic sounds using our method (left); a ball striking plates of various sizes plays a melody (middle); and a set of wind chimes blowing in a virtual forest (right).

[1]This chapter previously appeared as a paper in the 26th IEEE Conference on Virtual Reality (IEEE VR 2019) and an article in a special issue of IEEE Transactions on Visualization and Computer Graphics (TVCG). The original citation is as follows: Sterling, A., Rewkowski, N., Klatzky, R. L., and Lin, M. C. (2019). Audio-material reconstruction for virtualized reality using a probabilistic damping model. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):1855–1864

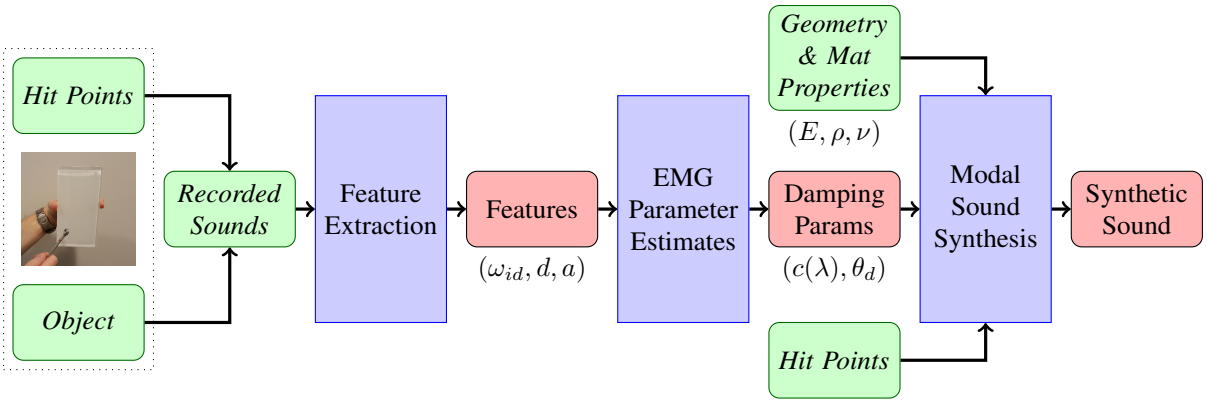


Figure 4.2: Our pipeline for estimating material parameters from recorded audio and using the parameters to synthesize sound for objects of the same material. Inputs are in green with italic text. If the object and hit points are unknown, the pipeline can begin with recorded sounds instead.

Automated material parameter estimation provides a means to estimate the material parameters of a specific object while reducing required human effort. Given an object made of a particular material, we can strike the object and record the resulting sound. Existing methods use the sound, along with mandatory knowledge about the shape and properties of the struck object, to estimate a number of material parameters (Ren et al., 2013b). The material parameters can be applied to sound synthesis, "virtualizing" the audio characteristics of a given material. While recent techniques have been able to estimate material damping parameters, they assume minimal effect on damping from external factors.

For example, an object struck for the purposes of recording either needs to be held by hand or left to rest on another surface. The interface between the object and its *support* will introduce additional damping. To account for this support damping, recordings must be made with supports that introduce minimal damping, requiring a carefully controlled recording environment using special support (Pai et al., 2001), e.g. strings or rubber bands, to suspend the object (Ren et al., 2013a). Other factors that affect estimated damping values, such as complex modes of vibration, background noise, and accumulated error during estimation are assumed by prior work to be minimized. Satisfying all of the assumptions made by prior work requires significant human effort.

In this chapter, we present a practical and efficient probablistic algorithm to estimate material damping parameters directly from recorded impact sounds that accounts for these different factors affecting damping, reducing their effects on the estimated parameters. Unlike previous work (Ren et al., 2013a), this method is fast and requires no prior knowledge about the recorded object's geometry, size, or hit location(s). We are able to virtualize the specific material of a given object. Our method requires significantly less human time

and effort to acquire material damping parameters than previous methods, while producing parameters of similar quality. The key contributions of this work include:

- A new probabilistic material damping model that independently considers each source of damping (Section 4.2.5);
- Application of this probabilistic model to estimation of material damping parameters (Section 4.2.6);
- A study evaluating human effectiveness at manual estimation of material parameters from sound (Section 4.3.2); and
- Quantitative (Section 4.3.3) and perceptual (Section 4.3.4) evaluation of estimated damping parameters.

We validate our method through comparison between estimated and ground-truth damping values, an auditory perceptual study, and comparison against alternative techniques. Figure 4.1 demonstrates our system in several complex virtual environments consisting of real-time interaction with virtual objects of different materials. Figure 4.2 shows the full pipeline for estimating material parameters and using them to synthesize sound.

4.2 Probabilistic Damping Modeling

In order to perform the modal sound synthesis process described previously in Section 2.1, we need to know the object's geometry, Young's Modulus E, density ρ, Poisson's Ratio, and damping parameters α_j for a chosen damping model. We now consider how this information can be obtained in the first place. The geometry can either be taken from a real-world object or designed for a virtual object. Young's Modulus, density, and Poisson's Ratio can be measured from real-world objects, but for many materials these values have been published and approximate values can be selected for synthesis purposes. Damping parameters, on the other hand, are specific to their damping model and are difficult to find for arbitrary materials. In this section, we present our probabilistic damping model for observed damping rates in the presence of external environmental factors.

4.2.1 Hybrid Damping Model

Our probabilistic model extends any of the traditional damping models described in Section 2.1.2. For this work, we consider the Rayleigh and Caughey damping models previously introduced. We also consider

one additional damping model, derived from generalized proportional damping (see Chapter 3). This hybrid model incorporates Rayleigh damping and a power law damping model (Section 3.2.2.1). The damping rates are described according to the function:

$$c_i = \alpha_1 + \alpha_2 \omega_{in}^{2\alpha_3}. \tag{4.1}$$

When α_1 is 0, this becomes the power law damping model, and when α_3 is 1, this becomes the Rayleigh damping model. Since we have found that the optimal damping model varies depending on the object (Section 3.4), this hybrid model can model damping best represented by Rayleigh or power law damping. Using these deterministic damping models, we can now introduce our probabilistic model.

4.2.2 Feature Extraction from Audio

Our technique uses multiple recorded impact sounds to estimate material parameters. A mode that is heavily damped by external factors in one sound may be relatively undamped in another, providing additional information about the range of possible damping values. As damping parameters are geometry-invariant (Ren et al., 2013a) for simple objects often present in virtual enviroments, we do not need to know the object's geometry, its size, or its hit location.

The first step in our approach is to extract the modal components of each input sound. Assuming the sounds come from rigid objects, the sound produced will be mostly modal and can be decomposed into a set of *features*. Each feature corresponds to one mode of vibration and can be parameterized as a damped sinusoid with a damped frequency ω_{id}, an initial amplitude a_i, and an exponential damping coefficient d_i.

We adopt a feature extraction process that identifies likely features, then performs local optimization. This is derived from the feature extraction step of Ren et al. (Ren et al., 2013b), which is described in Section 3.3.1.1. For this work, we again adopt the same feature extraction step, but decouple it from the subsequent Match Ratio Product parameter estimation (Section 3.3.1.2).

We further extend this standalone feature extraction step to improve robustness. As an additional step, we remove features with d_i under a threshold. These low-damping features are likely to be a constant pitched background noise unrelated to the impact sound. We also remove features below an amplitude threshold, as they are more susceptible to noise. The extracted (ω_{id}, a_i, d_i) features can be converted into pairs of (λ_i, d_i)

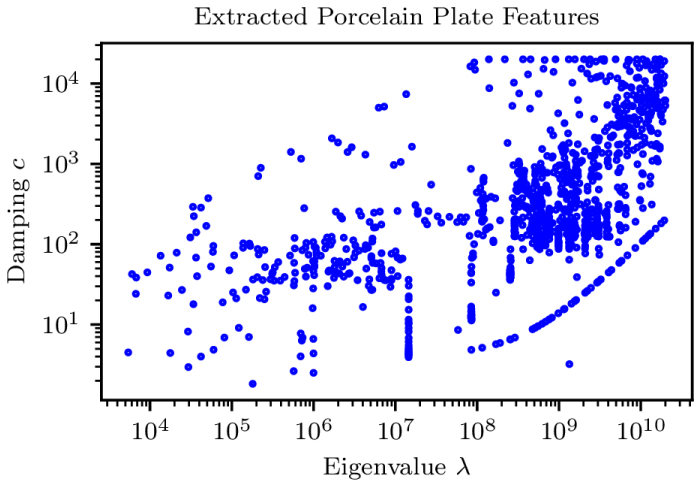


Figure 4.3: Features extracted from multiple impact sounds on a porcelain plate. λ is the eigenvalue of the mode of vibration (related to the frequency), while c is the rate of exponential decay. For any value of λ, there is a range of possible d values, which can be captured in a statistical model.

values by inverting the process in Equation (2.10):

$$\lambda_i = \omega_{in}^2 = \omega_{id}^2 + d_i^2. \tag{4.2}$$

As a result of this feature extraction process, we have a set of features roughly corresponding to the modes of vibration of the object. The most notable modification is that we account for background noise (modeled as additive white Gaussian noise) by estimating the amplitude of the noise floor. The extracted (ω_{id}, a_i, d_i) features are converted into pairs of (λ_i, c_i) values, where λ_i is the eigenvalue corresponding to that mode of vibration and $c_i = 2d_i$.

4.2.3 Distributions of Damping Values

With (λ_i, c_i) features extracted from multiple input sounds, we now interpret the results. Figure 4.3 shows an example of features extracted from impact sounds on a porcelain plate. Note that for any given eigenvalue λ, there exists a range of extracted damping values. This is especially noticeable where feature points appear as a vertical line, showing that even the same mode of vibration may have a variable rate of decay. These results are inconsistent with the damping models in Equations (2.11) and (2.12), which propose a one-to-one mapping between λ and c. Instead, we propose that there is significant error present in the extracted damping value of each feature, and that error can be modeled with a statistical distribution.

The prior work of Ren et al. (Ren et al., 2013b) estimates damping parameters using a least-squares metric to compare spectrograms. Similar results could be produced by fitting a damping model to (λ, c) features using least-squares (e.g. by optimizing all α_j). Statistically, a least-squares fit of a damping model is equivalent to assuming there is normally-distributed error around the model. We will refer to least-squares fitting of damping models as LSQ. However, we have found experimentally that least-squares fits tend to overestimate the material damping parameters, and resynthesized sounds all sound heavily overdamped.

Another notable property of Figure 4.3 is the clear line of points forming a lower bound to the data (with a few outliers). We have found experimentally that a damping model fit to this lower bound curve results in resynthsized sounds much closer to the input sounds. If the damping model should fit the lower bound, then all error is positive and can only increase the extracted damping values. Statistically, this indicates a one-sided error distribution; e.g. half normal, exponential, or chi-square distribution. Computationally, a lower bound Rayleigh damping model can also be found as a line along the lower convex hull of the points. We refer to lower-bound fitting of damping models as LB.

However, as can be seen in Figure 4.3, outliers often appear below the clean LB curve, and for other objects such a clean curve does not appear in the first place. A strict LB fit will be highly sensitive to outliers, as it must assume all error is positive. It is difficult to detect and remove outliers in extracted feature datasets. To solve this problem, we examine the physical sources of error in extracted damping values and construct an appropriate statistical distribution modeling that error. Ideally, this should produce a more robust lower bound fit which handles outliers based on their statistical probability of occurrence.

4.2.4 External Damping Factors

To accurately model error in damping values, we consider a number of physical phenomena that may affect estimates of the material damping values. These *external* damping factors are distinct from the material damping, which occurs due to the internal structure of a material.

4.2.4.1 Support Damping

An object's method of support can be varied; the object could be sitting on a desk, held in a hand, or dangling from a ceiling. We define a *support* broadly as any long-lasting contact with the sounding object of interest, with enough friction to maintain its contact with the object even when the object is struck. Regardless of the form of support, some energy from the object's vibrations will be transferred to the support, causing

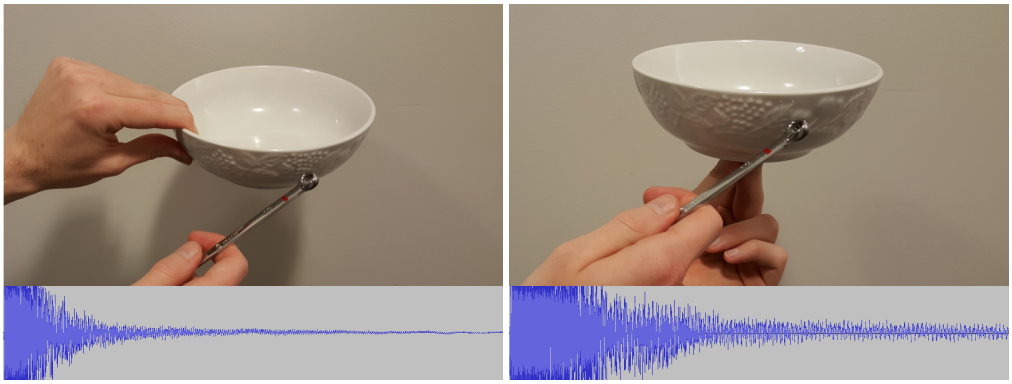


Figure 4.4: A porcelain bowl struck in the same location produces different sound when supported with a tight grip (left) or supported by resting on a single point (right). Without accounting for the effect of the support, prior methods would not be able to estimate accurate material parameters from these sounds.

additional damping. In real-world situations where the object is unlikely to be minimally supported, the additional damping significantly affects the sound.

Refer to Figure 4.4 for an example of the effect of the support on the resulting sound. A tight grip on the bowl's rim produces a more damped sound compared to gentle balancing on fingertips.

4.2.4.2 Complex Modes

Complex modes of vibration are slight deviations from normal mode behavior. Unlike normal modes, complex modes are not linearly separable: energy may be transferred between modes while vibrating. A mode that *loses* energy to others will produce higher damping values, while a mode that *gains* energy from others will produce lower damping values. Most systems have only slightly complex modes (i.e. there is little energy transfer), so normal modes are a close approximation (Imregun and Ewins, 1995), but not an exact one. Since we make the assumption of normal modes, the slight transfers of energy are a source of error in damping value estimates.

4.2.4.3 Background Noise

Background noise in recorded sounds is too variable to realistically model. The feature extraction step of the method is designed to specifically extract *modal* features from the sound. This mostly eliminates persistent "hums" which do not match the modal exponential-decay model. We modify the feature extraction method to account for additive white Gaussian noise (Section 4.2.2). This further removes the influence

of persistent background noise, though there may still be some remaining Gaussian (normal) error in the spectrograms and their resulting extracted damping values.

Acoustic reflections and reverberations from room acoustics are confounding factors. Without knowing the properties of the room acoustics, we cannot separate the effect of a damping material from the effect of the acoustics. For our model, we still assume minimal room reverberations, but some small sources of transient noise or early reflections may be appropriately modeled by normally distributed error.

4.2.4.4 Feature Extraction Error

The feature extraction step itself is not perfect; some error is introduced in the process. For example, spectrograms have limited spectral and temporal resolution, and the Fourier transform's assumption of periodicity in each window is an approximation. The discretization of the spectrogram will produce small amounts of error. Sidelobes resulting from Fourier transforms may appear as separate peaks or affect the estimated damping rate of nearby modes.

4.2.4.5 Acoustic Radiation

Uneven acoustic radiation from the object may mean that different microphone placements will result in different initial mode amplitudes. This can be accounted for by keeping the object and microphone stationary during an impact sound. However, the relative positions of the microphone and object do *not* need to be fixed across all input sounds. Moving the microphone between sounds will not change the frequencies or exponential rates of decay, and thus does not need to be accounted for in our model.

4.2.4.6 External Factor Summary

Current damping parameter estimation techniques do not explicitly consider these factors, instead attributing all damping to the material (as we do in Chapter 3). The resulting damping parameters therefore model the combined effect of the material *and the recording environment*. These parameters may not properly transfer to an object of the same material in a *different environment*. This limits the sounds that can be used for accurate damping parameter estimation: the sounds must be recorded in a carefully controlled setting. With a thoroughly robust technique that can separately model environmental factors, we can reduce the factors' impact on the estimated parameters. The external factors cannot be fully removed, but reducing their impact may result in more physically-accurate material parameters.

4.2.5 Generative Model for Combined Damping

We now introduce a generative model for sampling damping values. The model defines the probability distribution for an extracted damping value c_i, given the eigenvalue λ_i and a set of parameters θ. θ contains parameters representing both the *material* and the *environment*. The material damping parameters, such as α_1 and α_2, are referred to as θ_d for generality. The model can be written as $p(c_i|\lambda_i, \theta)$, and asks, "given a known material and environment, what is the probability of measuring any particular damping value?"

The value c_i is a damping value obtained from the feature extraction step. In the *absence* of any external factors, c_i would consist only of material damping. To account for the external factors, we model c_i as a random variable based on the sum of normally and exponentially distributed random variables.

4.2.5.1 Normal Distribution

A normal distribution models the effect of some external factors. The normal distribution accounts for (1) energy transfer due to complex modes, (2) small sources of background noise, and (3) error in feature extraction due to spectrogram discretization. We assume that each of these factors are an additive, normally distributed random variable. The sum of these normally distributed factors (c_i^n) is also normally distributed:

$$p(c_i^n|\lambda_i, \theta_d, \sigma) = \mathcal{N}\left(c(\lambda_i, \theta_d), \sigma^2\right) \tag{4.3}$$

The distribution is centered on the damping function c evaluated at an eigenvalue λ with damping parameters θ_d, with a standard deviation σ resulting from the combination of factors.

4.2.5.2 Exponential Distribution

An exponential distribution models the effect of the object's support.

$$p(c_i^e|\eta) = \mathrm{Exp}\,(\eta) = \eta e^{-\eta c_i^e}. \tag{4.4}$$

c_i^e is the resulting exponential damping resulting from the object's support, while η is the rate parameter of the exponential distribution. This distribution is an approximation, but in attempting to create a robust lower bound method, it serves the role of a one-sided distribution fitting to the lower bound of damping values.

Zheng and James defined a model to approximate additional per-mode damping based on contacts with other objects (Zheng and James, 2011). However, a statistical analysis of this model is highly dependent on the distribution of elements of the matrix of eigenvectors $\mathbf{\Phi}$. We are not aware of any prior work that has attempted to statistically model the distribution of eigenvector matrix $\mathbf{\Phi}$ elements, and our own analysis using Kolmogorov-Smirnov goodness-of-fit tests found no probable common distributions. In the absence of a more well-defined model and with the main requirement of a one-sided distribution satisfied, the exponential distribution was selected empirically based on extracted feature data.

4.2.5.3 Exponentially Modified Gaussian

The combined damping value c_i can then be modeled as the combination of (1) the normally-distributed factors c_i^n due to complex modes, noises, and other sources of errors, and (2) the exponentially-distributed factor c_i^e due to the support damping. Assuming that the factors are independent (for mathematical feasibility), they can be formulated as two separate sources of exponential decay of the mode amplitude z_i:

$$z_i(t) = a_i e^{-c_i^n t} e^{-c_i^e t} \cos(\omega_{id} t) \tag{4.5}$$

$$= a_i e^{-(c_i^n + c_i^e)t} \cos(\omega_{id} t). \tag{4.6}$$

The probability density function of the sum of the normal and exponential distributions ($c_i^n + c_i^e$) is the convolution of their individual probability density functions. The resulting distribution is an *exponentially modified Gaussian* (EMG) distribution. EMG distributions have been used extensively in chromatography (Grushka, 1972), but have also found uses in other domains. The probability density function for the EMG is:

$$p(c_i|\lambda_i, \theta_d, \sigma, \eta) = \frac{\eta}{2} e^{\frac{\eta}{2}(2c(\lambda_i,\theta_d) + \eta\sigma^2 - 2c_i)} \operatorname{erfc}(s_i) \tag{4.7}$$

$$s_i = \frac{c(\lambda_i, \theta_d) + \eta\sigma^2 - c_i}{\sqrt{2}\sigma}, \tag{4.8}$$

where erfc is the complementary error function, defined as:

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-y^2} dy. \tag{4.9}$$

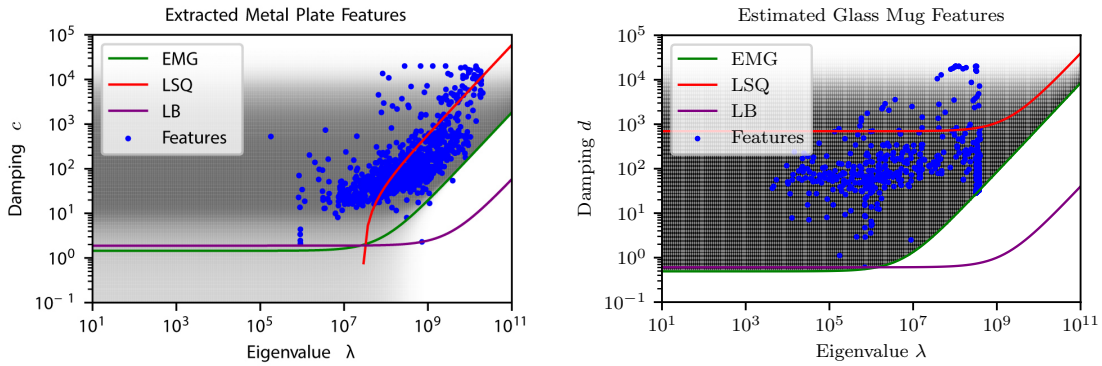


Figure 4.5: Parameter estimation on sound features. Each feature consists of an eigenvalue λ_i and its corresponding damping coefficient c_i. Estimated Rayleigh damping curves are plotted, with the variation from the curve caused by external factors. Our method, using the EMG distribution, provides the closest fit to the lower bound of the data while being relatively unaffected by outliers.

This defines the probability of observing an extracted damping value, given the material damping and environmental damping parameters. This is the complete generative model for damping values, encompassing multiple sources of damping and errors. Since only the frequencies and damping values of the modes are needed for this model, we do not need to assume that the mode shapes remain unchanged. The full set of parameters θ is (θ_d, σ, η), which together define the distribution.

4.2.6 Parameter Estimation

With the generative model established, we now describe the estimation of damping parameters. We estimate the parameters θ through maximum likelihood estimation (MLE). The generative model above uses known parameters to produce data from a distribution. MLE is an optimization method that reverses the process: use known data from a distribution to produce best-fitting parameters. Given a set of extracted (λ_i, c_i) pairs as data and a set of parameters, we can use the generative model to compute the log-likelihood of the data given the parameters:

$$\log p(\mathbf{d}|\boldsymbol{\lambda}, \theta_d, \sigma, \eta) = \sum_i \log\left(\frac{\eta}{2}\right) + \eta c(\lambda_i, \theta_d) + \frac{\eta^2\sigma^2}{2} - \eta c_i + \log\left(\operatorname{erfc}(s_i)\right). \tag{4.10}$$

Using the log-likelihood simplifies computation, removing exponentiation and turning a product of probabilities into a sum of log probabilities. We want to find the parameters that *maximize* this log-likelihood—and hence also maximize the original probability. These maximizing parameters are those that best explain

the extracted data, "fitting" the probability distribution to the data. We compute the analytical gradient of the log-likelihood function and perform gradient ascent to find these optimal parameters.

We compute the full average derivative for the n (λ_i, c_i) samples. We define a term t_i and use the scaled complementary error function $\mathrm{erfcx}(s_i) = \exp(s_i^2)\,\mathrm{erfc}(s_i)$ to simplify notation:

$$t_i = \frac{-2}{\mathrm{erfc}(s_i)\sqrt{\pi}} e^{-s_i^2} = \frac{-2}{\mathrm{erfcx}(s_i)\sqrt{\pi}}. \tag{4.11}$$

The derivatives for η and σ must be computed for all damping models. Their derivatives are as follows:

$$\frac{\partial \log p}{\partial \eta} = \frac{n}{\eta} + n\eta\sigma^2 + \sum_i c(\lambda_i, \theta_d) - c_i + t_i \frac{\sigma}{\sqrt{2}}, \tag{4.12}$$

$$\frac{\partial \log p}{\partial \sigma} = n\lambda^2\sigma + \sum_i t_i \left(\frac{\eta\sigma^2 + c_i - c(\lambda_i, \theta_d)}{\sqrt{2}\sigma^2} \right). \tag{4.13}$$

The derivatives for θ_d will depend on the damping function itself. We will present the derivatives for Rayleigh damping here; derivatives for alternative models are not difficult to compute. For Rayleigh damping's linear $c = \alpha_1 + \alpha_2\lambda$ function, the derivatives for α_1 and α_2 are:

$$\frac{\partial \log p}{\partial \alpha_1} = \eta n + \sum_i \frac{t_i}{\sqrt{2}\sigma}, \tag{4.14}$$

$$\frac{\partial \log p}{\partial \alpha_2} = \sum_i \eta\lambda_i + \frac{t_i\lambda_i}{\sqrt{2}\sigma}. \tag{4.15}$$

With the derivative established, we can perform standard gradient ascent until convergence. The final damping parameters in θ_d are the optimal parameters for the material of the struck object. These damping parameters can be used to represent the recorded material for modal sound synthesis, with other effects (e.g. room acoustics, supports) modeled separately (Zheng and James, 2011).

Section 4.2.6 shows features extracted from 19 impact sounds on a metal plate, while Section 4.2.6 shows features extracted from 40 impact sounds on a glass mug. Similarly, Figure 1.5 shows features extracted from synthetic sounds. The figure compares our EMG fit with MLE optimization against the baseline LSQ and LB methods (see Section 4.2.3). In each case, LSQ overfits the data, while LB is strongly affected by low outliers and underfits the data.

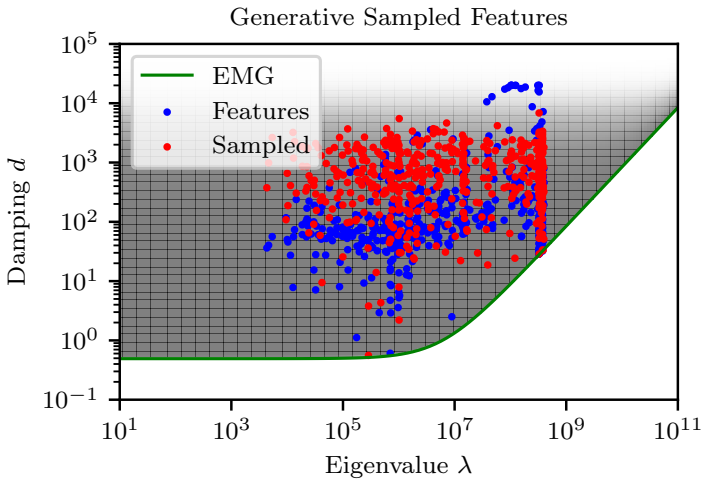


Figure 4.6: Comparison of real-world extracted features (blue) and sampled features from a fitted EMG model (red). The two sets of points are similarly distributed, indicating that the EMG model is properly fit to the real-world data.

4.2.7 Discussion and Analysis

The effect of external damping factors cannot be entirely removed, and in real-world situations the extracted damping values may all be much higher than the material damping function alone. This positively biases the estimator: the estimated parameters will often be larger than the ground truth. By accounting for external factors, this estimator has less bias than other methods, and is therefore more accurate.

Figure 4.6 shows experimental validation of the EMG model. It contains the *same* extracted glass mug features as found in Figure 4.5 in blue, but overlays additional features in red. These red features are sampled from the optimized EMG distribution. They need not correspond one-to-one with the extracted features, but they should follow a similar *distribution*. This shows experimentally that the underlying statistical model is appropriate for capturing the distribution of real-world data.

4.2.8 Sound Synthesis with Estimated Values

The estimated damping parameters should be accurate, having accounted for the effect of the support. However, modal sound synthesis assumes free vibrations (i.e. no support) when in most cases there will be something supporting the object. An additional step is needed to apply support damping to synthesize contact sounds due to support.

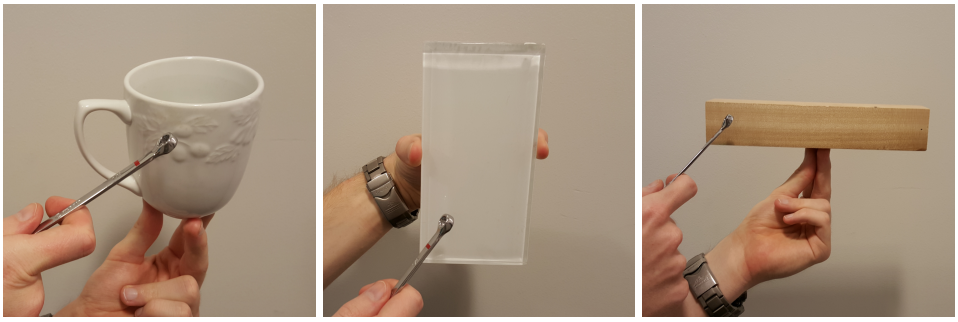


Figure 4.7: Three objects from our impact sound dataset: a porcelain cup (left), a small glass tile (center), and a wood block (right). Note the ways that each object is supported. These supports interfere with damping parameter estimation.

We adopt a contact model for modal sound synthesis introduced by Zheng and James (Zheng and James, 2011). The method uses an additional damping matrix $\mathbf{G}$ to model the additional damping resulting from each contact point k in the set of contact points $\mathcal{C}$:

$$\mathbf{G} = \sum_{k \in \mathcal{C}} c_k \mathbf{\Phi}_k^T (\mu \mathbf{I} + (1 - \mu) \mathbf{n}_k \mathbf{n}_k^T) \mathbf{\Phi}_k, \tag{4.16}$$

where c_k is the magnitude of the contact force for contact k, $\mathbf{\Phi}_k$ is the set of eigenvectors corresponding to the point at contact k, μ is the coefficient of friction, and $\mathbf{n}_k$ is the normal direction at that point. Some mode coupling is introduced since $\mathbf{G}$ is not diagonal, but this coupling was found to be perceptually minor. Therefore, each damping model may be augmented by adding the corresponding diagonal component of $\mathbf{G}$.

4.3 Results

We have implemented the damping parameter estimation method described in Section 4.2.5 and tested its effectiveness through both numerical analysis and perceptual validation. With this method, the process for material damping parameter estimation involves striking an object repeatedly, ideally with varying hit locations and support methods. This approach has less strict requirements about the recording environment than previous work; sounds can be recorded in a quiet room, as long as there are few transient sounds and the room is not heavily reverberative.

We have recorded numerous impact sounds on a set of fifteen rigid objects, where the hit points and the method of support are documented for each impact. Figure 4.7 shows a sample of these objects, with

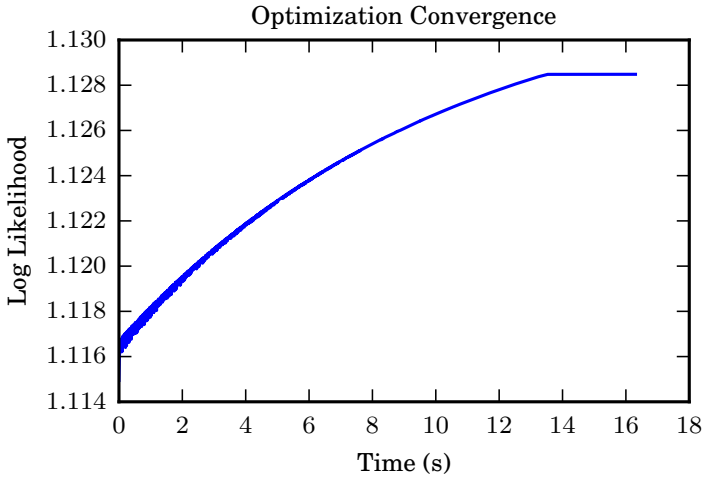


Figure 4.8: Plot of log-likelihood maximization converging over the course of parameter estimation. Optimization was performed on 752 frequency-damping points extracted from porcelain plate impact sounds, and converged after 39,009 iterations for a total of 16.3s in an one-time preprocessing.

various hit locations and methods of support. There are an average of nearly 50 impact sounds sampled per object. All objects were supported by hand, often either with an edge being pinched between two fingers or the center resting on a few fingertips. Audio was recorded using a Zoom H4 in a padded sound recording booth which reduced, but did not eliminate, acoustic effects and background noise. Objects were struck with a small metal wrench, the wrench itself being tightly gripped to minimize its own vibrations.

The eigenvalues and damping values are each normalized, but the data are not shifted or centered. With this normalization, the estimated damping values need to be unnormalized for application to other materials. Although we cannot guarantee that the optimization problem in this context is always convex, especially for higher order damping functions, in multiple runs from different starting points on multiple datasets, all optimization processes have converged on the same parameters. The optimal value of σ tends to be very small, indicating that the distributions of damping values tend to be closer to exponential distributions than to normal distributions. σ and η are used to guide optimization of the damping parameters, but they are not needed for sound synthesis.

We implemented the parameter optimization algorithm in Python and NumPy. On a laptop with a dual core 2.53 GHz Intel Core i5-540M processor, optimization over thousands of features from tens of input sounds and hundreds of thousands of iterations takes 1-5 minutes to complete. See Figure 4.8 to see an example of convergence behavior. Note that we are attempting to *maximize* the log-likelihood, as the

		Porcelain	Travertine	Wood	Steel	Plastic	Glass
Rayleigh	α_1	3.9	1.3	39.0	2.3	39.8	2.0
	α_2	1e-8	2.5e-8	1.3e-7	6.9e-8	1.3e-7	7.8e-8
Hybrid	α_1	3.9	1.3	39.0	2.4	34.83	1.9
	α_2	5.2e-9	2.5e-8	2.1e-7	5.5e-8	4.1e-7	1.5e-7
	α_3	1.027	1.001	0.978	1.011	0.95	0.974

Table 4.1: Damping parameters estimated using our technique. These materials come from a subset of objects in our impact sound dataset. These parameters are described in Sections 2.1 and 4.2.1. When hybrid $\alpha_3 = 1$, the remaining hybrid damping parameters are equivalent to their Rayleigh damping counterparts. These parameters can be used to virtually recreate the material of the real-world object.

parameters that maximize the log-likelihood also maximize the underlying probability. Upon convergence, the optimized θ_d parameters model the damping behavior of the recorded material.

Table 4.1 contains results from estimation on some of the objects. When hybrid $\alpha_3 = 1$, the model is identical to Rayleigh damping. Even small changes in hybrid α_3 can have a large impact on the resulting damping. For example, a 10 kHz mode on the Porcelain plate has a damping coefficient $d = 20$ with the provided parameters (hybrid $\alpha_3 = 1.027$), but changing α_3 to exactly 1 reduces the damping coefficient to $d = 12$.

In general for these damping models, larger parameters create virtual materials with more damping and shorter sounds. For example, the two objects with the most damping are the wood block and plastic bowl, whose materials are known to be naturally heavily damped. The porcelain plate, travertine tile, and glass tile all had similar estimated parameters.

4.3.1 Real-time Synthesis and Rendering

Finally, the optimized parameters are used for sound synthesis. Each sounding object must be preprocessed before running any interactive application. Preprocessing time depends primarily on the number of tetrahedra in the input mesh; a mesh with 2,000 tetrahedra takes under a minute to preprocess while a mesh with 30,000 tetrahedra can take many minutes. Once each sounding object has been preprocessed, modal synthesis is performed in real time at 44 kHz.

Like previous work (Ren et al., 2013b), we are able to synthesize sound using an interactive rigid-body physics simulation in real time. We have implemented our method for sound synthesis with support damping in C++ as a module for Unreal Engine 4. Our demos have been integrated with an HTC Vive headset and

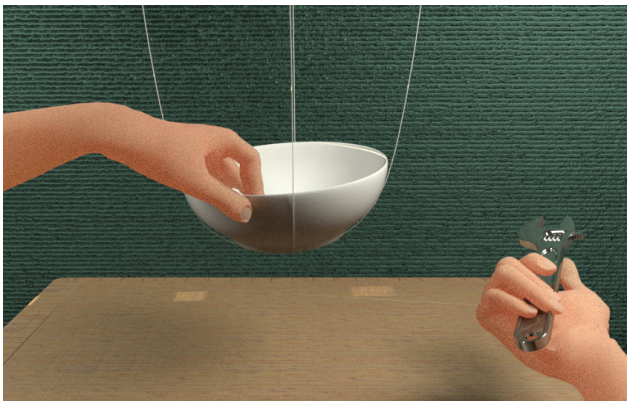


Figure 4.9: A simulated porcelain bowl is struck in multiple locations, with and without a supporting grip.

Leap Motion controller. The user's hands were tracked with the Leap Motion, with the Vive controllers used to represent tools that could be picked up and used to strike objects. Users can walk in the virtual environment and strike objects, immediately hearing the resulting synthesized sound. Figure 4.1 shows multiple scenes from our real-time demo, with multiple objects of various shapes and materials. Figure 4.9 shows another scene, where a bowl is supported by either strings or a hand, producing different sounds depending on the hit point and support type.

4.3.2 Human Hand-Tuning Evaluation

In the absence of an automated method for damping parameter estimation, parameters have traditionally been estimated by hand. We present a study evaluating the effectiveness of human damping parameter estimation, using human subjects to hand-tune material parameters for multiple objects. Specifically, we are interested in the tuning of the damping parameters and the specific modulus γ, defined as the ratio of Young's modulus to density. We seek to evaluate the distributions of subjects' selected material parameters. For example, are subjects able to agree on a single unique set of material parameters, and if so, to what degree of precision? We also seek to determine the time and sound samples needed for subjects to reach their conclusions.

4.3.2.1 Experimental Setup

We constructed an easy-to-use GUI enabling interactive adjustment of material parameters for sounds produced through modal sound synthesis. For each object in the study, we created a corresponding 3D model by hand (a laborious process requiring precise measurements) and performed modal analysis on that model

once (a few hours of computation time). The damping parameters and specific modulus γ for an object can be adjusted as a post-processing step, without needing to repeat the lengthy modal analysis step. With these optimizations, resynthesis with modified parameters took less than two seconds, allowing for rapid iteration. In the interface, each parameter was controlled with a slider, with a range of plausible realistic values presented on normalized scales from 0–100.

Subjects were recruited primarily through mailing lists and were not required to have any background in parameter tuning or impact sound analysis. Subjects were compensated financially for their participation. Subjects were given real-world objects, placed on small foam blocks to reduce support damping. For each object, the subjects' task was to tune material parameters such that the synthesized sound produced by the application most closely matched the sound they heard when striking the real object. Subjects were instructed to find the most accurate parameters possible, regardless of the time needed. Subjects first performed this task with a training object, in order to reduce learning effects. The six objects evaluated were all disk-shaped objects of approximately the same radius and thickness. Every subject hand-tuned material parameters for all six objects in a random order.

The study was divided into two sections. The first 20 subjects hand-tuned three parameters for each object: the two Rayleigh damping parameters and the specific modulus. The following 20 subjects hand-tuned two parameters for each object: just the two Rayleigh damping parameters. For the two-parameter section, the specific modulus was set to the mode of the subject-selected specific moduli from the three-parameter section. The three-parameter section models the real-world case where all three parameters must be picked in order to virtualize an object.

4.3.2.2 Results and Analysis

We first consider the distributions of subjects' selected material parameters. Figure 4.10 shows results from the three-parameter section of the study for a few selected objects: a wood disc and a porcelain disc. For highly reverberant objects such as the porcelain disk, subjects could generally agree on Rayleigh damping's α_1 parameter for each object. However, for highly damped objects such as the wood disk, Rayleigh α_1 responses were less consistent. The distributions of Rayleigh α_2 parameters for each object show agreement between subjects, indicated by the relatively low standard deviations and frequently unimodal distributions. The specific modulus, which modifies the pitch of the synthesized sound, often resulted in multimodal distributions.

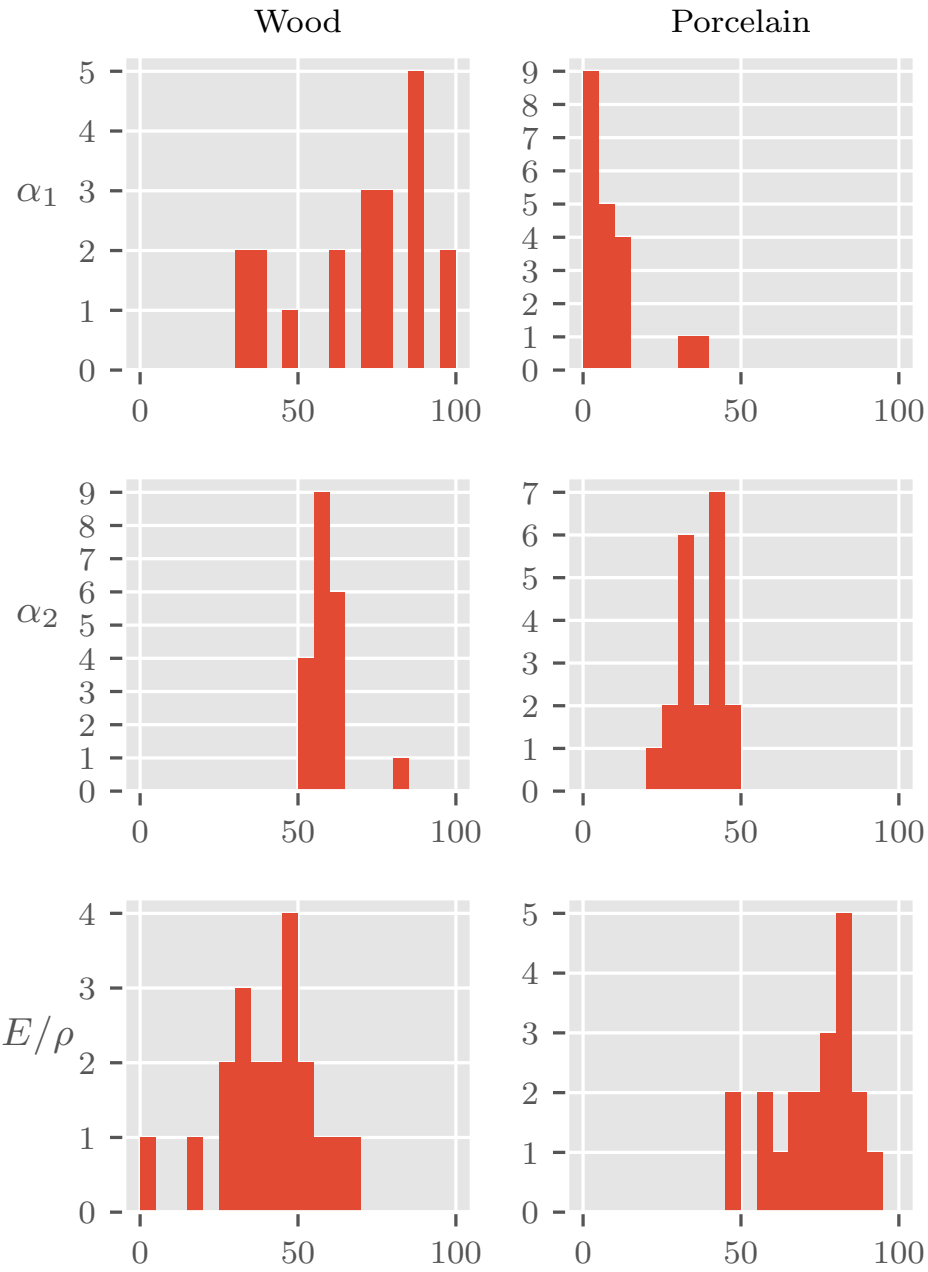


Figure 4.10: Distributions of human-tuned material parameters for wood and porcelain discs. α_1 and α_2 are Rayleigh damping parameters, E/ρ is the specific modulus, and all parameters were tuned on normalized scales from 0–100. The observed distributions indicate that subjects had difficulty finding a unique optimal solution for the specific modulus, and for α_1 in the case of highly-damped objects.

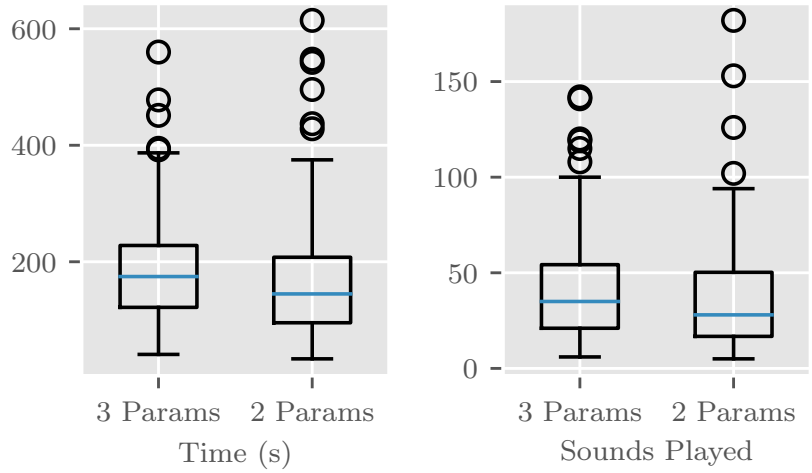


Figure 4.11: Box-and-whisker plots of the time and number of synthesized sounds needed for subjects to reach their final hand-tuned material parameters, with deviant observations plotted as outliers. Between the two versions of the study, the difference in median time is 20 s, and the difference in median sounds played is 7 sounds. Our method is an automated version of the 2-parameter study, and significantly reduces the human labor needed.

We also consider the time and number of sounds needed for subjects to reach their conclusions. Figure 4.11 contains histograms for the amount of time and number of synthesized sounds needed for subjects to finalize their selections. The median time needed was 165 s to tune three parameters, and 145 s to tune two parameters. The median number of synthesized sounds needed was 35 to tune three parameters, and 28 to tune two parameters. The range of times (33–614 s) and sounds (5–182) was highly variable, and the effect of parameter count on times and sounds did not reach statistical significance by t-test.

Our method for parameter estimation is an automated way to perform the parameter selection task. Human hand-tuning requires around 145 s and 28 sounds per object, requiring dedicated human attention for the entire duration. Hand-tuning also requires creating an accurate 3D model of the object and performing modal analysis, possibly adding hours of extra human effort. In contrast, our automated method operates effectively with 10–20 sounds and does not require a 3D model of the object. Parameter estimation then takes a few minutes, during which no human attention is required. Overall, our method significantly reduces the amount of human labor needed to create virtualized objects. Compared to prior work, our method reduces human labor by not requiring carefully controlled recording environments, creation of a 3D model, and knowledge of object geometry and hit points.

4.3.3 Synthetic Validation

Synthetic validation provides a numerical comparison against ground-truth damping parameters. We synthesized a variety of sounds with known damping parameters and passed the resulting sounds through the parameter estimation process to see if the original ground-truth values could be recovered using our algorithm. Sounds were synthesized from the geometry of 18 models, ranging from small, hollow cups to desktop vases and large sculptures. Five materials were chosen by randomly sampling material parameters from a range of realistic values. For each object, ten support points were sampled at random on the surface of the object, each with a random amount of contact force ranging from a light support to a moderate support. Then, 100 sounds were synthesized for each combination of object and material. Each sound sampled its impact point randomly on the exterior surface and picked one support point to be active. The resulting sounds were passed through the feature extraction process for Rayleigh damping, and extracted features from a varying number of sounds were used to estimate the original parameters.

Parameter estimation was performed with three different estimators: EMG (our method, see Section 4.2.5) and the two baselines LSQ and LB (see Section 4.2.3). Direct comparison against the algorithm of Ren et al. (Ren et al., 2013b) is infeasible due to the significant differences in inputs and outputs. However, their method will produce results most similar to the least squares (LSQ) estimator. We compared the error between the ground-truth parameters and the estimated parameters while using a varying number of input sounds. For each tested number of impact sounds, 30 different sets of sounds of that cardinality were sampled, and the resulting errors averaged.

4.3.3.1 Discussion

Figure 4.12 shows the relative error for each parameter and each estimator. For all materials in this synthetic data, both the EMG and LB estimators significantly outperformed the LSQ estimator ($p < .05$). With real-world data, the EMG and LB estimators more frequently decouple, as the EMG estimator's statistical model better adapts to noise and other artifacts of recording. These synthetic sounds, without noise or other effects, are the ideal situation for the LB estimator, and do not leverage the full capabilities of the EMG estimator.

Rayleigh α_1 estimates have minimal error, especially with larger amounts of data. While the error in Rayleigh α_2 is relatively higher, no prior work has performed a similar validation for comparison. Prior work

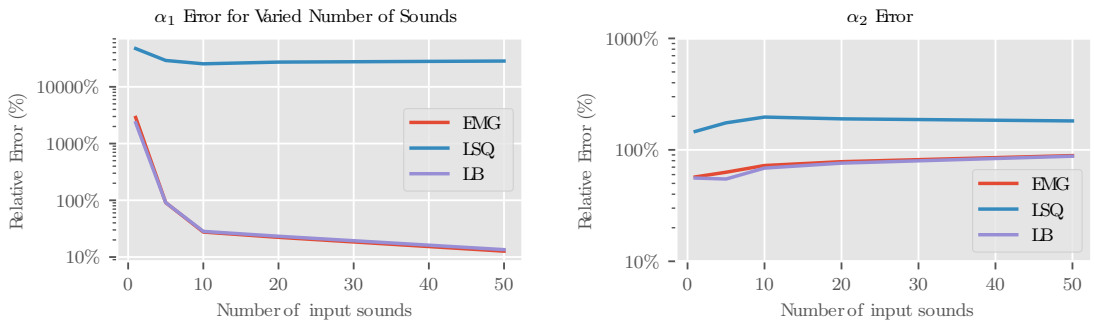


Figure 4.12: Relative error for Rayleigh damping parameters α_1 and α_2 in synthetic validation. As the number of sounds used for parameter estimation increase, error in Rayleigh damping parameter α_1 decreases while α_2 displays some overfitting as number of input sounds increases.

would produce results most similar to the LSQ estimator, which was outperformed by our method. In light of this, our method provides an improvement over previous work while removing the need for knowledge of geometry and hit points. Finally, the error is mostly important as it affects users' perception of the material. Our perceptual evaluation provides an analysis of whether our estimated parameters are accurate when evaluated by humans.

4.3.4 Perceptual Evaluation

Numerical comparisons against previous work are difficult since our method is the first work to estimate damping parameters given only input audio with no knowledge of geometry, size, or hit point. In this study, we considered recorded real-world sounds, and sounds synthesized using three sets of damping parameters: parameters from Ren et al. (Ren et al., 2013b), parameters from the human hand-tuning study (Section 4.3.2), and parameters estimated using our method to create 4 datasets. We sought to evaluate how well the synthesized sounds recreate the real-world sounds. Subjects evaluated sounds individually, answering questions about qualities of the sounds and estimating properties of the object or impact. Synthesized sounds that more accurately recreate qualities of the real-world sounds should produces similar patterns of answers to questions.

4.3.4.1 Experimental Setup

The study was conducted in an online web questionnaire, and subjects were recruited through mailing lists and online posts, but no financial compensation was offered. No prior experience in auditory perception

was expected. Subjects were asked to wear headphones or earbuds to ensure a consistent auditory environment. All sounds were scaled to the same volume, though difference in sound playback devices may have affected perception. Subjects listened to a series of impact sounds, answering questions about each. Variables involved are sound datasets (4: as listed above), object shape (2: disc or rod), and material class (5: wood, metal, plastic, glass, porcelain). All together, this produces a total of 40 sounds to evaluate.

24 subjects participated in the study, but more specific demographic information was not collected. Each subject listened to all 40 impact sounds in randomized order. For each sound, subjects were asked which object shape and material class they suspected created the sound. Subjects also were asked to rate descriptive qualities of the sound—the duration, ringiness, tonality, and pitch—on 7-point ordinal scales. The extreme ends of the scales were descriptively labeled, e.g. tonality ranged from "mixed tones" to "pure tone". Subjects could listen to each sound multiple times as needed. A brief training section at the beginning provided example sounds and definitions for the descriptive qualities.

4.3.4.2 Results: Confusion Matrices

Even with recorded real-world sounds, user identification of material and shape is not always accurate. In evaluation of synthetic datasets, we compare the pattern of errors to those of the real-world sounds, with a closer match suggesting more realistic synthesized sounds. Figure 4.13 shows confusion matrices for material class identification for the disc-shaped objects.

The recorded dataset demonstrates mis-labelings such as heavy confusion between wood and plastic, perception of the glass and ceramic discs as metal, and high accuracy on the metal disc. The hand-tuned dataset differs primarily in perception of its glass and ceramic objects; these differences could be due to human error while hand-tuning or due to inherent assumptions in the underlying modal synthesis model. The Ren dataset largely reproduces the matrix from their original paper (Ren et al., 2013b), although it does not recreate the error patterns (particularly metal) seen in our recorded or hand-tuned datasets. Our dataset (EMG) most closely resembles the hand-tuned results, with the exception of plastic being identified as ceramic by some subjects.

We evaluate the pairwise similarity between these matrices by computing the Frobenius norm of the element-wise difference of the two. The two most similar matrices are the hand-tuned and EMG dataset results, with a difference norm of 16.03. In comparison, between Ren and the hand-tuned data, the norm is 22.09. Against recorded sounds, EMG's norm was 23.11, while ren's norm was 31.85 and hand-tune's norm

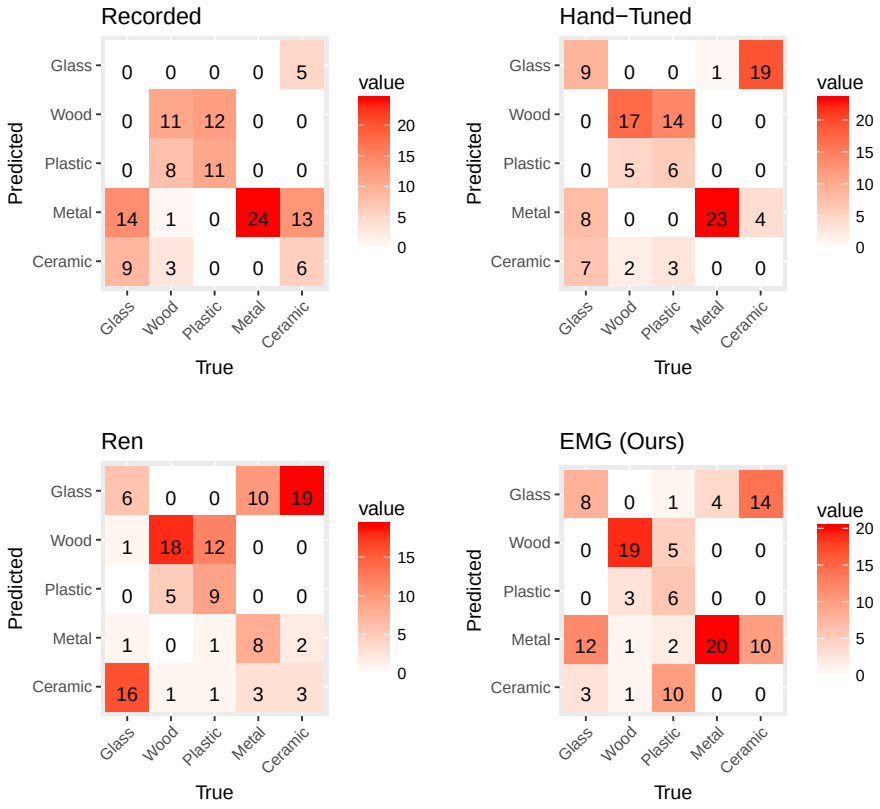


Figure 4.13: Material confusion matrices for the disc-shaped objects in our perceptual study. All four tested datasets (including recorded sounds) show significant labeling errors. However, our method (EMG) replicates the pattern of errors seen in the recorded and hand-tuned datasets more closely than the Ren dataset, suggesting more accurate recreations of the real-world sounds.

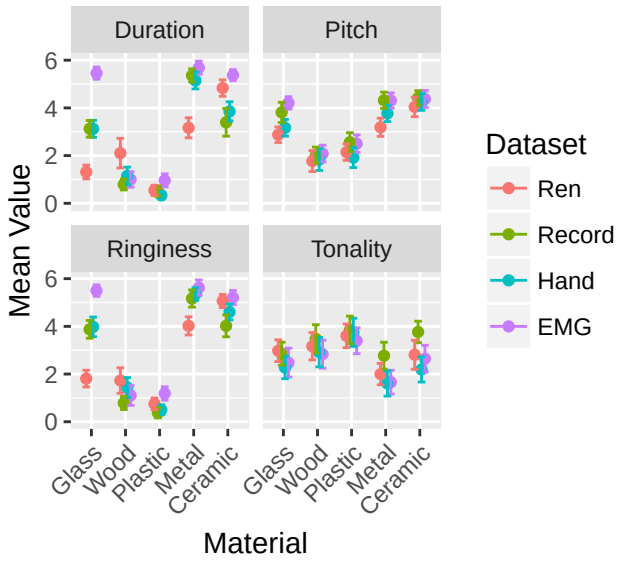


Figure 4.14: The mean selected value for each descriptive quality, material, and dataset. Most datasets are tightly clustered for pitch and tonality, with more differences in duration and ringiness. The Ren dataset contains many statistically-significant differences from the recorded dataset. While our EMG dataset contains some differences in duration and ringiness, it is overall closer to the recorded means.

is 22.83. The high similarity (low difference norm) between our EMG results and the hand-tuned results suggests that our method automatically produces sounds perceptually similar to what would be selected by human hand-tuning.

4.3.4.3 Results: Descriptive Qualities

We evaluate the descriptive quality ratings by performing a multi-factorial repeated measures ANOVA. Each of the variables (sound dataset, object shape, and material) is considered an ANOVA factor, each a repeated measure across subjects. The main effects of dataset, shape, and material are all significant for all four descriptive qualities. For example, for perception of pitch, the effects of material ($F_{4,92} = 95.61, p < .05$), shape ($F_{1,23} = 30.35, p < .05$), and dataset ($F_{3,69} = 22.79, p < .05$) are all significant. This is not surprising, as each of these effects alone can dramatically change the sound. Almost all interaction effects are significant, with the exceptions of material*dataset on tonality ($F_{12,276} = 2.814, p = .107$). Table 4.2 contains the full list of main and interaction effects for each perceptual quality.

Figure 4.14 contains the mean values for each descriptive quality, material, and dataset (using combined results for shapes). For pitch and tonality, most dataset means are closely-clustered. Duration and ringiness

	Df1	Df2	F value	$Pr(> F)$
Duration				
Material	4	92	739.4	$< .001$
Shape	1	23	65.65	$< .001$
Dataset	3	69	81.23	$< .001$
Material*Shape	4	92	52.97	$< .001$
Material*Dataset	12	276	53.56	$< .001$
Shape*Dataset	3	69	43.16	$< .001$
Mat*Shape*Dset	12	276	16.3	$< .001$
Pitch				
Material	4	92	95.61	$< .001$
Shape	1	23	30.35	$< .001$
Dataset	3	69	22.79	$< .001$
Material*Shape	4	92	5.754	$< .001$
Material*Dataset	12	276	2.814	.0012
Shape*Dataset	3	69	69.23	$< .001$
Mat*Shape*Dset	12	276	4.691	$< .001$
Ringiness				
Material	4	92	469.7	$< .001$
Shape	1	23	35.07	$< .001$
Dataset	3	69	51.47	$< .001$
Material*Shape	4	92	40.25	$< .001$
Material*Dataset	12	276	28.45	$< .001$
Shape*Dataset	3	69	7.708	$< .001$
Mat*Shape*Dset	12	276	4.984	$< .001$
Tonality				
Material	4	92	9.325	$< .001$
Shape	1	23	20.97	$< .001$
Dataset	3	69	15.27	$< .001$
Material*Shape	4	92	11.54	$< .001$
Material*Dataset	12	276	1.548	0.107
Shape*Dataset	3	69	35.36	$< .001$
Mat*Shape*Dset	12	276	2.53	.0035

Table 4.2: Significance of fixed effects in our perceptual study, as determined by repeated measures ANOVA. For each of the four perceptual qualities, the resulting degrees of freedom, F score, and p value are listed. All main effects and almost all interactions are significant at the $p < .001$ level.

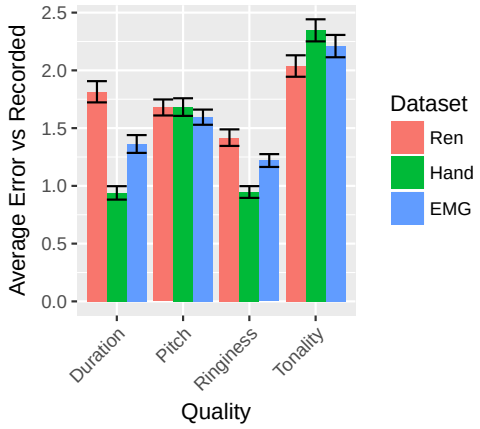


Figure 4.15: Average error between perceptual quality ratings in the recorded dataset versus each of the three synthetic datasets. A lower average error is better, indicating more perceptual similarity with the recorded sounds. Both hand-tuned and EMG perform well.

show more difference: while hand-tuned and recorded are closely aligned, Ren and EMG occasionally display more variance. Ringiness displays nearly the same pattern of significance as duration.

Synthetic datasets that produce more realistic sounds should have descriptive qualities similar to the recorded dataset. To evaluate this, we look at the absolute error between each subject's recorded and synthetic ratings for each sound. Across all materials and shapes, hand-tuned sounds were closest to the recorded sounds for duration and ringiness. For pitch and tonality, all datasets produced more similar results.

Figure 4.15 shows the computed average error values. The hand-tuned dataset is closest for the duration and ringiness qualities, our EMG method is the closest for pitch, Ren is closest for tonality. Our EMG dataset outperforms the Ren dataset on each perceptual quality except for tonality. One-way repeated-measures ANOVAs show a significant effect of dataset on error for duration ($F_{2,46} = 50.03, p < .05$), ringiness ($F_{2,46} = 20.11, p < .05$), and tonality ($F_{2,46} = 5.717, p < .05$). There was no significant effect of dataset on error for pitch ($F_{2,46} = 50.03, p = .347$).

Hand-tuned parameters produce the closest ratings to the recorded dataset on these perceptual scales. The Ren dataset contains many discrepancies from the real dataset, shown in the duration, pitch, and ringiness of the more reverberative materials. Our EMG dataset properly reproduces the perceptions of the pitch and tonality of the recorded objects, but in some cases produces higher duration and ringiness. Our EMG method demonstrates an improvement over the Ren dataset in the fit to recorded sounds.

4.4 Summary

We have presented a method for estimating material damping parameters using recorded impact sounds as the only input. We have validated these contributions through parameter estimation on a new dataset of impact sounds on rigid objects, using both an auditory user study and synthetic validation. These methods can extract real-world material parameters from audio recording and recreate virtualized materials and their rich sound effects arising from dynamic interaction in virtual environments.

4.4.1 Limitations

Our method removes a number of assumptions used by prior damping parameter estimation techniques (Ren et al., 2013b). For example, our method does not require knowledge of the object's geometry, and it reduces the strict assumptions on the object's support and the presence of background noise. However, some common assumptions of prior works remain: (1) application to rigid objects and their vibrations can be accurately modeled by linear analysis. (2) difficulty to fully remove all external damping factors—the presence of loud transient noises, a tightly-coupled support, or a highly reverberative room may still impose residual effects.

4.4.2 Future Work

In general, for parameter estimation, there exists a tradeoff between the amount of assumptions on the required inputs and the quality of outputs. We do not assume prior knowledge on the object geometry, size, material parameters, or the impact location—just audio recording is sufficient. However, this technique currently does not estimate Young's modulus, Poisson's ratio, density, or geometric properties of the object. Generalization of a probabilistic model like this work or use of learning algorithms can potentially estimate these parameters automatically using only a few audio recordings. A method that can optimize all parameters simultaneously would further simplify the pipeline from audio recording to automatic synthesis. Future work may explore if additional inputs can result in a much greater increase in the number of estimated parameters. A single sound is not enough to estimate parameter α_1 with sufficient accuracy; upwards of 10–20 sounds may be needed. In our human parameter tuning study, subjects were initially untrained; experts may produce slightly different parameter distributions.

CHAPTER 5: Integrated Multimodal Interaction Using Texture Representations[1]

5.1 Introduction

In computer graphics, texture mapping has been one of the most widely used techniques to improve the visual fidelity of objects while significantly accelerating the rendering performance. There are several popular texture representations, such as displacement maps (Cook, 1984), bump mapping with normal maps (Blinn, 1978; Cohen et al., 1998), parallax maps (Kaneko et al., 2001; Tevs et al., 2008), relief maps (Oliveira et al., 2000; Policarpo et al., 2005), etc., and they are used mostly as "imposters" for rendering static scenes. These textures are usually mapped onto objects' surfaces represented with simplified geometry. The fine details of the objects are visually encoded in these texture representations. By replacing the geometric detail with a texture equivalent, the resulting rendered image can be made to appear much more complex than its underlying polygonal geometry would otherwise convey. These representations also come with a significant increase in performance: texture maps can be used for real-time augmented and virtual reality (AR/VR) applications on low-end commodity devices.

Sensory conflict occurs when there is a mismatch between information perceived through multiple senses and can cause a break in immersion in a virtual environment. When textures are used to represent complex objects with simpler geometry, sensory conflict becomes a particular concern. In an immersive virtual environment, a user may see a rough surface of varying heights and slopes represented by its texture equivalent mapped to a flat surface. In the real world, objects behave very differently when bouncing, sliding, or rolling on bumpy or rough surfaces than they do on flat surfaces. With visually complex detail and different, contrasting physical behavior due to the simple flat surface, sensory conflict can easily occur—breaking the sense of immersion in the virtual environment. In order to capture such behaviors, the geometry used in a physics simulator would usually require a fine triangle mesh with sufficient surface detail, but in most cases a sufficiently fine mesh is unavailable or would require prohibitive amounts of memory to store.

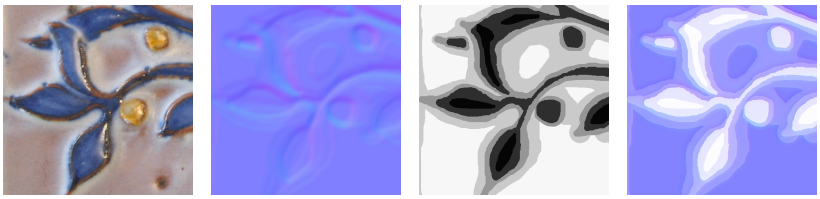
[1]This chapter previously appeared as an article in Computers & Graphics. The original citation is as follows: Sterling, A. and Lin, M. C. (2016a). Integrated multimodal interaction using texture representations. *Computers & Graphics*, 55:118 – 129

Since the given texture maps contain information about the fine detail of the mapped surface, it is possible to use that information to recreate the behavior of the fine mesh. Haptic display and sound rendering of textured surfaces have both been independently explored (Otaduy et al., 2004; Ren et al., 2010), but texture representations of detail have not been previously used for visual simulation of dynamic behavior due to collisions and contacts between rigid bodies. For example, the system for sound rendering of contacts with textured surfaces (Ren et al., 2010) displays a pen sliding smoothly across highly bumpy surfaces. While the generated sound from this interaction is dynamic and realistic, the smooth *visual* movement of the pen noticeably does not match the texture implied by the sound. In order to minimize sensory conflict, it is critical to present a unified and seamlessly integrated multimodal display to users, ensuring that the sensory feedback is consistent across the senses of sight, hearing, and touch for both coarse and fine levels of detail.

Motivated by the need to address the sensory conflict due to the use of textures in a multimodal virtual environment, we explore the use of both normal maps and relief maps as unified texture representations for integrated multimodal display. The main results of this work include:

- A new effective method for visual simulation of physical behaviors for rigid objects textured with normal maps;
- A seamlessly integrated multisensory interaction system using normal maps;
- An extended system using relief maps;
- Evaluation and analysis of texture-based multimodal display and their effects on task performance; and
- Evaluation of perceptual differences between normal and relief map representations.

The rest of the chapter is organized as follows. We first discuss why we have selected normal and relief maps as our texture representations for multimodal display. We then describe how each mode of interaction can specifically use normal maps to improve perception of complex geometry (Section 5.2). We highlight the behavior of virtual objects as they interact with a large textured surface, and describe a new method to improve visual perception of the simulated physical behaviors of colliding objects with a textured surface using normal maps. We also demonstrate how to use the same normal maps in haptic display and sound rendering of textured surfaces. We describe how the additional depth information in relief maps can be used to improve each mode of interaction (Section 5.3).



Color map Normal map Depth map Relief map

Figure 5.1: Texture map example. RGB values encode normal vectors in each texel. In relief maps, the alpha value encodes depth information.

We have implemented a prototype multimodal display system using normal and relief maps and performed both qualitative and quantitative evaluations of its effectiveness on perceptual quality of the VR experience and objective measures on task completion (Section 5.4). A user study suggests that normal maps can serve as an effective, unified texture representation for seamlessly integrated multi-sensory display and the resulting system generally improves task completion rates with greater ease over use of a single modality alone. A second user study suggest that relief maps are also an effective representation of fine detail, with an improvement in sensory consistency over normal maps.

5.2 Overview and Texture Map Representation

Our system uses three main components to create a virtual scene where a user can experience through multiple modalities of interaction. A rigid body physics simulator controls the movement of objects. The only form of user input is through a haptic device, which also provides force feedback to stimulate the sense of touch. Finally, modal sound synthesis is used to dynamically generate the auditory component of the system. In this section, we briefly cover the details of texture mapping, discuss haptic illusions and justify the use of texture representations, then describe each of these components using normal maps as the representation of detail. The relief map representation is covered in greater detail in Section 5.3.

5.2.1 Normal and Relief Maps

Normal maps are usually stored as RGB images, with the color values encoding vectors normal to the details of the surface they are mapped to. Refer to Figure 5.1 for an example. It is common practice to create normal maps directly corresponding to a color map, such that the color map can be referenced at a location to get a base color and the normal map can be referenced at the same location for the corresponding normal vector.

Relief mapping is a technique for rendering textured surfaces using additional depth information. It is usually implemented on GPUs and can be briefly described as computing intersections with the height-field defined by the depth values using rays from the camera to each pixel (Policarpo et al., 2005). Ray casting lets relief-mapped surfaces properly handle self-occlusion, and extra ray casts from a light source enable self-shadowing. Since rays are cast from the camera, proper perspective is maintained as the camera looks at the textured surface from different angles. Our surfaces are rendered using relief mapping, so we refer to their textures as "relief maps", though the same texture could be used for parallax occlusion mapping or for displacements on GPU-tessellated surfaces.

Our relief maps contain their depth information in the alpha channel of the image. In the alpha channel, a value of zero (black, entirely transparent) means the texel is at its highest, exactly along the geometry of the mapped object. Larger values (tending towards white/visible) indicate that the texel is recessed inside the object. Much like sculpted relief artwork, relief maps can only cut into the surface; they cannot raise a texel outside the object's geometry. The maximum depth as a percentage of mapped object dimensions can be set individually for each relief map.

Depending on the resolution of the texture image and the surface area of the object it is mapped to, a normal or relief map can provide very fine detail about the object's surface. This detail—while still an approximation of a more complex surface—is sufficient to replicate other phenomena requiring knowledge of fine detail.

5.2.2 Design Consideration

Next we discuss various consideration in choosing texture maps as our representation of fine detail, beginning with a discussion on haptic perception.

5.2.2.1 Haptic Illusions

Perceptual illusions, including visual, haptic and auditory, have been explored in virtual reality for immersing users in computer generated environments through multi-sensory display. For example, bump mapping can be regarded as a *visual* illusion where a user who is expecting to see depth in a bump-mapped surface may interpret the shading as depth. Haptic illusions can be roughly defined as when a haptic stimulus is applied under specific conditions that change the perception of that stimulus. A classic example is the size-weight illusion in which a participant lifts two boxes of equal weight and unequal sizes and perceives the

smaller box to be heavier. There are many types of haptic illusions, which have been well documented and catalogued (Hayward, 2008).

There are some real-world examples of haptic illusions that are relevant for simulating slope and depth. In the "curved plate" illusion, a flat edge rolled over a fingertip at about 1 Hz produces the sensation that the edge is curved. As described earlier, previous work on simulating haptic textures also relies on haptic illusions: applying only lateral forces to a haptic probe can create the sensation of a vertical height difference.

In these illusions, the changing direction of normal force creates the illusion of curvature. That is, *the normal vector is an important haptic cue for curvature*. Texture maps with normal vectors provide exactly that information, and therefore should be able to simulate the curvature of a more complicated surface through haptic illusions. This observation forms the hypothesis of our exploration of texture representations.

5.2.2.2 Choice of Representation

On top of providing an important haptic cue, normal vectors have additional advantages over alternative options. Using very high-resolution geometry would automatically produce many of the desired effects, but the performance requirements for *interactive* 3D applications significantly reduces their viability in our early deliberation. This is especially important to consider in AR and VR applications, where real-time performance must be maintained while possibly operating on a low-end mobile phone or head mounted display.

Other texture map information may also be considered, such as height (or displacement) maps. For sound, Ren et al. (Ren et al., 2010) used normal maps because the absolute height does not affect the resulting sound; it is the change in normal that causes a single impulse to produce meso-level sound. With regard to force display of textured surfaces, the Sandpaper system (Minsky, 1995) has been a popular and efficient method for applying tangential forces to simulate slope based on a height map. Using normal vectors we can instead scale a sampled normal vector to produce the same normal and tangential forces. Rigid body collision response also depends entirely on normal vectors.

Since each component of the system depends directly on the normals, a normal map representation emerges as the natural choice. An added convenience is that normal maps are widely supported (including mobile games) and frequently included alongside color maps. Although normal maps contain the most important cues for multimodal interaction, we would like to evaluate how much benefit is gained from combining normals with depth information. Relief mapping uses both for visual rendering and has become

more common alongside GPUs, so relief maps provide a useful starting point for considering depth in multimodal interaction with textures. The application needs, the performance requirement, and the wide availability and support on commodity systems all contribute to our adoption of normal maps and relief maps as the mapping techniques in this work.

5.2.3 Rigid Body Dynamics

In order to simulate the movement of objects in the virtual scene, we use a rigid body dynamics simulator. These simulators are designed to run in real time and produce movements of rigid objects that visually appear believable.

Rigid body dynamics has two major steps: collision detection and collision response. Collision detection determines the point of collision between two interpenetrating objects as well as the directions in which to apply force to most quickly separate them. Modifying the normals of an object, as we do with normal maps, does not affect whether or not a collision occurs. This is a significant limitation of a normal map representation without any height or displacement information.

There are numerous algorithms for collision resolution, which determines how to update positions and/or velocities to separate the penetrating objects. In impulse-based approaches, collisions are resolved by applying an impulse in the form of an instantaneous change in each objects' velocity. Considering a single object's velocity vector $\mathbf{v}$, $\Delta\mathbf{v}$ is chosen to be large enough so that the objects separate in the subsequent timesteps. The change in velocity on an object with mass m is computed by applying a force f over a short time Δt in the direction of the geometric normal $\mathbf{n_g}$ of the other colliding object:

$$\Delta\mathbf{v} = \frac{f\Delta t}{m}\mathbf{n_g} \tag{5.1}$$

This process is highly dependent on the normal vectors of each object, and other collision resolution approaches have this same dependency.

5.2.3.1 Modifying Collision Behavior with Normal Maps

We focus on simulating collisions between small dynamic objects and large textured surfaces whose details would have a large effect on the dynamic object. To get an intuitive understanding of the behavior we seek to replicate, imagine a marble rolling on a brick-and-mortar floor. When the marble rolls to the edge of a

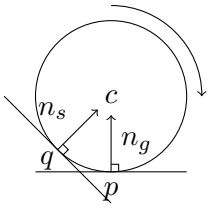


Figure 5.2: Contact point modification on a rolling ball: given the contact point $\mathbf{p}$ and sampled normal $\mathbf{n_s}$, we want to simulate the collision at point $\mathbf{q}$.

brick, the expected behavior would be for it to fall into the mortar between bricks and possibly end up stuck at the bottom.

The level of detail needed to accurately recreate these dynamics with a conventional rigid body physics engine is too fine to be interactively represented with a geometric mesh, especially with large scenes in real-time applications. A normal map contains the appropriate level of detail and is able to represent the flat brick tops and rounded mortar indentations.

In order to change the behavior of collisions to respect fine detail, our solution is to modify the contact point and contact normal reported by the collision detection step. This is an extra step in resolving collisions, and does not require any changes to the collision detection or resolution algorithms themselves.

The contact normal usually comes from the geometry of the colliding objects, but the normal map provides the same information with higher resolution, so our new approach uses the normal map's vectors instead. Given the collision point on the flat surface, we can query the surface normal at that point and instruct the physics engine to use this perturbed normal instead of the one it would receive from the geometry. One side effect of using the single collision point to find the perturbed normal is that it treats the object as an infinitely small probe.

5.2.3.2 Rolling Objects and Collision Point Modification

There is a significant issue with this technique when simulating rolling objects. Refer to Figure 5.2 for an example. Two planes are shown, the horizontal one being the plane of the coarse geometry and the other being the plane simulated by the perturbed normal. Note that the contact points with the rolling ball differ when the plane changes. The vector $\mathbf{n_s}$ shows the direction of the force we would ideally like to apply. If we were to apply that force at the original contact point $\mathbf{p}$, the angular velocity of the sphere would change and the ball would begin to roll backwards. In practice, this often results in the sphere rolling in place when

it comes across a more extreme surface normal. Instead, we use the sphere radius r, the perturbed surface normal $\mathbf{n_s}$, and the sphere center $\mathbf{c}$ to produce the modified contact point $\mathbf{q}$:

$$\mathbf{q} = \mathbf{c} - (r\mathbf{n}) \tag{5.2}$$

This modification applies the force directly towards the center of mass and causes no change in angular velocity, but is less accurate for large spheres and extreme normal perturbations.

This contact point modification is important for perceptually believable rolling effects. Shapes other than spheres do not have the guarantee that the contact point will be in the direction of the $\mathbf{c} - \mathbf{n}$ vector, so this does not apply in the general case. Generally, we can simply modify the normal without changing the contact point. In the case of relief maps, the true collision points and contact normals can be determined, so this correction is unnecessary.

5.2.4 Haptic Interface

We have designed our system to use a PHANToM Desktop haptic device (Massie and Salisbury, 1994). This device can measure 6-DOF motion: three translational and three rotational, but display only 3-DOF forces (i.e. no torques). We have chosen to represent the PHANToM as a pen inside the virtual environment, which matches the scale and shape of the grip. While we could use forces determined by the rigid-body physics engine to apply feedback, the physics update rate (about 60 Hz) is much lower than the required thousands of Hz needed to stably simulate a hard surface.

We simulate the textured surface by projecting the tip of the PHANToM Desktop grip onto the surface in the direction of the coarse geometry's normal. The fine surface normal is queried and interpolated from nearby normal map vectors. The PHANToM simulates the presence of a plane with that normal and the projected surface point. Given the normal vector sampled from the normal map $\mathbf{n_s}$ and pen tip position projected onto the surface $\mathbf{p}$, the equation modeling this plane is:

$$(\mathbf{n_s} \cdot (x, y, z)) - (\mathbf{n_s} \cdot \mathbf{p}) = 0 \tag{5.3}$$

The PHANToM now needs to apply the proper feedback force to prevent the pen's tip from penetrating into the plane. This is accomplished using a penalty force, simulating a damped spring pulling the point

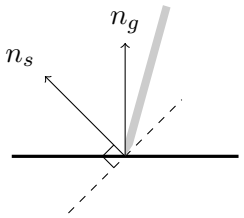


Figure 5.3: Haptic force is applied in the direction of the sampled normal $\mathbf{n_s}$ instead of the geometric normal $\mathbf{n_g}$.

back to the surface. Using the modified normal vector, the simulated plane serves as a local first order approximation of the surface. Note that while the slopes of the planes produced by the PHANToM can vary significantly based on the normal map, at the position of the pen the plane will coincide with the surface. This is illustrated in Figure 5.3, where the simulated plane intersects the geometric plane at the collision point. This creates an illusion of feeling a textured surface while keeping the pen in contact with the flat underlying surface geometry.

With this technique, stability can be concern in some cases. Most noticeably, in steep and narrow V-shaped valleys, a user pushing down on the surface might cause the tip of the pen to oscillate between the valley sides. Users sliding the pen rapidly across bumpy surfaces may also feel forces that are stronger and more abrupt than they would expect. We have mainly mitigated these concerns by smoothing the normal maps and scaling down the penalty forces. A side effect is that the surfaces end up feeling slightly smoother and softer, though we have found this an acceptable tradeoff for improved stability.

We use a simplified model to interact with dynamic objects. The PHANToM's corresponding pen appearance in the environment is added as an object in the rigid-body physics simulator. Whenever this pen comes in contact with a dynamic object, the physics simulator computes the forces on the objects needed to separate them. We can directly apply a scaled version of this force to the haptic device. This ignores torque as our 3-DOF PHANToM can apply only translational forces. This approach is fast, simple, and lets the user push and interact with objects around the environment.

5.2.5 Sound Synthesis

Sound is created due to a pressure wave propagating through a medium such as air or water. These waves are often produced by the vibrations of objects when they are struck, and human ears can convert these waves into electrical signals for the brain to process and interpret as sound. One of the most popular

physically-based approaches to modeling the creation of sound is modal sound synthesis, which analyzes how objects vibrate when struck at different locations to synthesize contact sounds. I provide a comprehensive description of the process of modal sound synthesis in Section 2.1.

5.2.5.1 Textures and Lasting Sounds

Modal synthesis works well for generating sound that varies for each object, material, and impulse. However, for long-lasting collisions such as scraping, sliding, and rolling, the sound primarily comes from the fine details of the surface, which are not captured in the geometry of the input mesh when using texture maps. We adopt the method by Ren et al. (Ren et al., 2010), which uses three levels of detail to represent objects, with normal maps providing the intermediate level of detail.

At the macro level, the object is represented with the provided triangle mesh. The first frame in which a collision is detected, it is considered transient and impulses are applied according to conventional modal synthesis. If the collision persists for multiple frames, we instead use the lower levels described below.

Even surfaces that look completely flat produce rolling, sliding, and scraping sounds during long-lasting collisions. The micro level of detail contains the very fine details that produce these sounds and are usually consistent throughout the material. Sound at this level is modeled as fractal noise. Playback speed is controlled by the relative velocity of the objects, and the amplitude is proportional to the magnitude of the normal force.

The meso level of detail describes detail too small to be efficiently integrated into the triangle mesh, but large enough to be distinguishable from fractal noise and possibly varying across the surface. Normal maps contain this level of detail, namely the variation in the surface normals. This sound is produced by following the path of the collision point over time. Any time the normal vector changes, the momentum of the rolling or sliding object must change in order to follow the path of that new normal. This change produces an impulse which can be used alongside the others for modal synthesis. This can be mathematically formulated as follows.

Given an object with mass m moving with tangent-space velocity vector $\mathbf{v_t}$ along a face of the coarse geometry with normal vector $\mathbf{n_g}$ whose nearest normal map texel provides a sampled normal $\mathbf{n_s}$, the component of the momentum orthogonal to the face $\mathbf{p_n}$ is:

$$\mathbf{p_n} = m\left(-\frac{\mathbf{v_t} \cdot \mathbf{n_s}}{\mathbf{n_g} \cdot \mathbf{n_s}}\right)\mathbf{n_g} \tag{5.4}$$

This momentum is calculated every time an object's contact point slides or rolls to a new texel, and the difference is applied as an impulse to the object. More extreme normals or a higher velocity will result in higher momentum and larger impulses. Whenever objects are in collision for multiple frames, both the micro-level fractal noise and the meso-level normal map impulses are applied, and the combined sound produces the long-lasting rolling, sliding, or scraping sound.

5.3 Relief Map Representation

As an extension to the modalities described above which rely solely on the surface's normal vectors, we have also explored how a relief map's depth information can be incorporated to improve each component. In this section, we explain these differences.

5.3.1 Modifying Collision Behavior with Relief Maps

When discussing rigid body physics with a normal map, we mentioned that collision *detection* remained unchanged while collision *resolution* required modification. With relief maps' depth information, collision *detection* now requires additional steps, as now objects may penetrate inside the geometry of a surface as long as they stay outside the recessed relief surface. Again focusing on the situation where a small object collides with a large textured surface, the problem is collision detection between an object and a height map. We adopt a similar approach described by Otaduy et al. for computing directional penetration depth between two textured objects (Otaduy et al., 2004).

In general, the penetration depth between two colliding objects is the shortest distance one of the objects would have to move in order to separate themselves. The *directional* penetration depth is the penetration depth where the objects can move along only one specified axis. Computing the general penetration depth between finely-detailed objects can be prohibitively slow for interactive applications. Directional penetration depth can be used in place of general penetration depth, sacrificing accuracy for speed, which is more appropriate for our goals.

The GPU-based method proposed by Otaduy et al. for computing directional penetration depth is to represent each colliding object as a height map perpendicular to the specified direction. These height maps

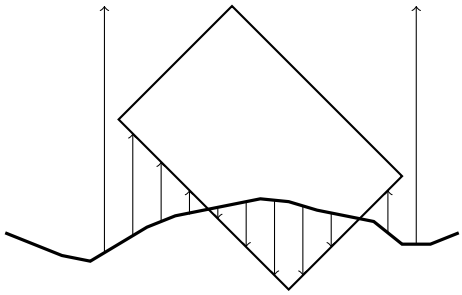


Figure 5.4: A rectangle colliding with a 1D relief map. Wherever arrows point downwards, the distance is negative and there is a collision.

are aligned with one another so that the distance between the objects at some point is the difference in height between two matching height map texels. Wherever the distance between objects is negative, there is a collision. The most negative distance value can then be reported as the directional penetration depth.

In our case, the large plane textured with a relief map is already a height map perpendicular to the normal vector of the plane. In order to adopt a similar technique on any CPU (and GPU), we need to convert the colliding object into a height map of its own. We primarily accomplish this by projecting the object onto the plane and rasterizing the result with the same resolution as the relief map. The depth information from that process can then be used as the object's height map. The difference between the relief map's depth and the object's height map is the distance between them, and one or more collision points can be found by searching for negative distances. The collision points and the normal vectors sampled from the relief map at the same locations can then be passed to the collision resolution solver.

A simple example is illustrated in Figure 5.4, where a rectangular object is colliding with a 1D relief map. Each arrow points from a relief map texel to the corresponding texel of the rasterized object height map, where upwards arrows are positive distance values and downwards arrows are negative. The most negative distance values would be reported as collision points. Since the points are found through a sampling process, there is naturally a tradeoff between speed and accuracy: each sample takes time to compute but contributes to finding a more accurate collision point.

5.3.2 Haptic Interface with Relief Maps

For haptic interaction through the PHANToM, as with rigid body physics, the change is in collision detection and not resolution. The tip of the pen is projected down in the direction of the surface normal, but collision is reported only if the pen's tip is below the relief map depth value. If there is a collision, the

simulated plane is created in exactly the same way as described in the normal map section. With depth information, the pen can follow the actual contours of the surface.

5.3.3 Sound Synthesis with Relief Maps

With normal maps, it is necessary to track the change in the sampled normal vector to estimate the impulses felt by a rolling, sliding, or scraping object for the purposes of sound synthesis. In the case of a relief map with depth information, we can compute significantly more accurate collision information, and with that comes significantly more accurate impulse information. With the relief map collision detection described previously, we can directly take the impulses reported by the physics engine and apply them to the bank of modes of vibration to synthesize sound.

Since the physics engine properly takes into account the normal and depth information from the relief map, the resulting impulses already account for the texture detail. Adding in the same fractal noise to account for surface variations too small to be captured by either texture representation produces realistic long-lasting contact sounds.

5.4 Implementation and Results

We have described each component of our multimodal system using texture maps. We implemented this prototype system in C++, using NVIDIA's PhysX as the rigid body physics simulator, OGRE3D as the rendering engine, VRPN to communicate with the PHANToM (Taylor II et al., 2001), and STK for playing synthesized sound (Cook and Scavone, 1999).

Objects can be discretized using spring-mass systems to perform modal analysis for sound synthesis, but for this work we instead use a finite element method representation using tetrahedral meshes. The difference between the representations is primarily that the spring-mass model represents objects as hollow shells with a given shell thickness, while using tetrahedral meshes properly represents the full volume of objects. With either representation, the system of equations in Equation (2.1) is used, but matrices are constructed differently. This provides an improvement in accuracy over spring-mass discretizations and negatively impacts computation time during the precomputation step only. All scenarios we created contained at least one textured surface acting as the ground of the environment, using its texture maps to modify collision response, haptic display, or sound rendering.

	Memory Requirements			Time Requirements		
	Mesh	**Offline**	**Runtime**	**Physics**	**Visual**	**Haptic**
Normal Map	10KB	2.7 MB	270 KB	175 μs	486 μs	60 μs
Relief Map	110KB	1 GB	18 MB	2.2 ms	900 μs	60 μs
Coarse Mesh	4.5 MB	288 GB*	450 MB*	3.0 ms	2.1 ms	–**
Fine Mesh	19 MB	4500 GB*	1700 MB*	4.9 ms	7.0 ms	–**

Table 5.1: Memory and timing results for our (texture-based) methods compared to a similarly detailed coarse mesh (66,500 vertices) and fine mesh (264,200 vertices). Entries marked with * are extrapolated values, since the memory requirements are too high to run on modern machines. Haptic time (**) was not measurable for triangle meshes due to an API limitation. Normal maps are able to achieve up to **25 times** of runtime speedup and up to **6 orders of magnitude** in memory saving.

5.4.1 Performance Analysis

The sound synthesis module generates samples at 44100Hz, the physics engine updates at 60Hz, and while the PHANToM hardware itself updates at around 1000Hz, the surface normal is sampled to create a new plane once per frame. On a computer with an Intel Xeon E5620 processor and 24GB RAM, the program consistently averages more than 100 frames per second. This update rate is sufficient for real-time interaction, with multi-rate updates (Otaduy et al., 2004; Ren et al., 2010).

A natural comparison is between our texture-based method and methods using meshes containing the same level of detail. Most of our texture maps are around 512×512, so recreating the same amount of detail in a similarly fine mesh would require more than $512^2 = 262114$ vertices and nearly twice as many triangles. As a slightly more realistic alternative, we also compare to a relatively coarse 256×256 mesh with more than $256^2 = 65536$ vertices. For a discussion of LOD representations and the challenges in simplifying meshes for multimodal systems, refer to Section 5.4.4.2.

Table 5.1 presents memory and timing information when comparing our method to methods using the equivalent geometry meshes instead. The coarse mesh used for modal analysis is greatly reduced in size compared to the finer meshes. We generated these finely-detailed meshes for the sake of comparison, but in practice, neither mesh would be available to a game developer and they would have to make do with the constraints considered in our method.

Modal analysis for audio generation on the finer meshes requires significantly more memory than is available on modern machines, so a simplified mesh is required. The listed runtime memory requirement is for modal sound synthesis and primarily consists of the matrix mapping impulses to modal response. The

listed memory requirements are based on a spring-mass discretization for normal maps and the FEM-based discretization for relief maps.

Our method is faster than using fine meshes in each mode of interaction. Haptic rendering time using our method took merely 60 μs per frame. The listed visual time requirement is the time taken to render the surface, either as a flat texture mapped plane, or as a color-mapped mesh without normal mapping. The PHANToM's API integrated with VRPN does not support triangular meshes, and we could not test performance of collision detection and haptic rendering manually, though the time needed to compute collision with an arbitrary triangular mesh would have been significantly longer (at least by one to two orders of magnitude based on prior work, such as H-COLLIDE).

The main sound rendering loop runs at 44.1 kHz regardless of the chosen representation of detail. The only difference comes from the source of sound-generating impulses: our method for normal maps collects impulses from a path along the normal map while a relief map or mesh-based approach collects impulses reported by the physics engine. Applying impulses to the modal synthesis system is very fast relative to the timed modes of interaction.

5.4.2 Normal Map Texture Identification User Study

In order to evaluate the effectiveness of this multimodal system, we conducted a user study consisting of a series of tasks followed by a questionnaire. One objective of this user study was to determine the overall effectiveness of our system. For this study, only the normal map representation was used. A subject is interacting with the normal map through sight, touch, and sound. If each of these components are well designed and implemented, a subject should be able to identify the material by multimodal interaction. The other goal is to see how well the use of multiple senses helps to create a consistent recognition of the material being probed. Even if subjects find the haptic display alone is enough to understand the texture of the material being probed, does adding sound cues speed up their process of identifying textures or instead cause sensory conflict?

5.4.2.1 Set-up

Twelve participants volunteered to take part in this study experiment. Each subject was trained on how to use the PHANToM and was given some time to get used to the system by playing in a test scene (see Figure 5.7, top row). The subject then completed a series of six trials. In each trial, a material for the surface

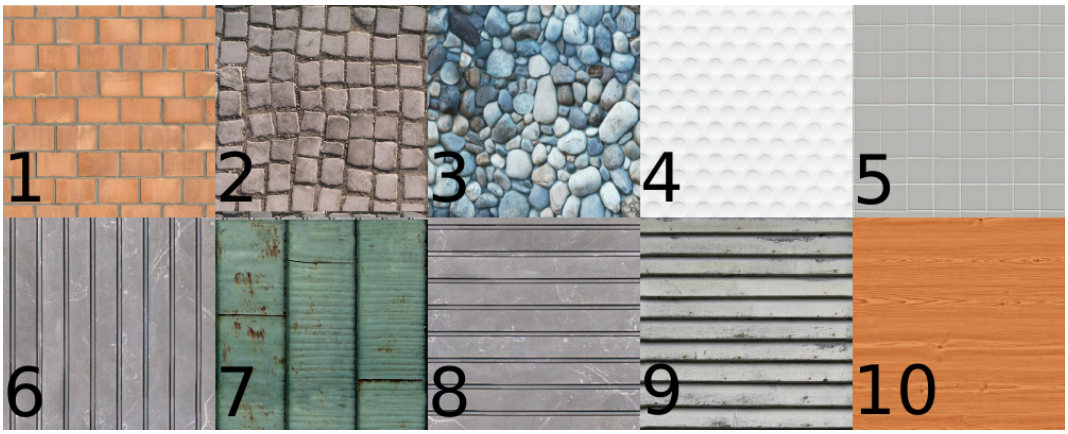


Figure 5.5: The available materials for the texture identification user study. 1–3 sounded like bricks, 4–5 sounded like porcelain, 6–8 sounded like metal, and 9–10 sounded like wood.

was chosen at random, and all aspects of it *except* its visual appearance were applied. That is, the subject would be able to feel the surface's texture with the PHANToM, hear the sound generated from ball and PHANToM pen contacts, and see the rolling ball respond to ridges and valleys on the surface. The subject was able to cycle through each material's visual appearance (in the form of a texture) by pressing the button on the PHANToM's grip. Their task was to select the material's unknown visual appearance based on the multimodal cues received.

The first three trials provided all three cues—sound, ball, and pen—but in each of the remaining three trials only two of the three cues would be available. The subject would be informed before the trial began if any cues were missing. The subjects were recommended to use all available cues to make their decision, but were otherwise unguided as to how to distinguish the materials. After the trials were completed, a short questionnaire was provided for subjective evaluation and feedback.

This study utilizes sensory conflict to guide the subjects to correctly identify the visual appearance. If the multimodal cues present the sounds, haptic texture, and physical response of a metal surface with regular grooves, but the subject has currently selected the visual appearance of a flat, smooth wooden surface, they should recognize the sensory conflict and reject the wooden surface as the answer. Once the subject has selected the correct visual appearance (grooved metal in this example), they should feel relatively little sensory conflict and from that realize they have found the answer.

Figure 5.5 shows the materials chosen for the user study. The subjects were allowed to look at each of these textures before the trials began, but were not able to feel or hear them. Some of these were specifically chosen to be challenging to distinguish.

	ID rate	Time (s)	Ease (1–10)
All modes	78%	38 ± 18	7.9 ± 1.3
No sound	81%	46 ± 45	4.9 ± 2.2
No haptics	54%	41 ± 23	3.6 ± 1.8
No physics	72%	47 ± 58	6.4 ± 2.6

Table 5.2: Results comparing modality effectiveness when limiting the available modes of interaction in the texture identification user study. "Ease" is evaluated by the subjects where 1 is difficult and 10 is easy. When using all modes of interaction, subjects were generally able to identify the material more frequently than when only using two modes and reported that they found identification to be easiest when all modalities of interaction were engaged.

	Always	Frequently	Occasionally	Rarely	Never	Reported accuracy (1–10)
Haptics	88%	0%	6%	0%	6%	9.3 ± 0.9
Sound	34%	22%	22%	11%	11%	7.6 ± 1.4
Physics	29%	6%	47%	6%	12%	7.3 ± 2.6

Table 5.3: Texture identification study: Results from question asking how often subjects used each mode of interaction and question asking how well each mode represented the materials (10 is very accurate).

5.4.2.2 Experimental Results

In Table 5.2, we compare the results when varying which modes of interaction are available to subjects. The ID rate is the percentage of trials in which the subject was able to correctly identify the material, and the mean time takes into account time for correct guesses only. The "ease" was provided by the subjects on the questionnaire, where they were asked to rate on a scale from 1–10 how easy they found it was to identify the material for each combination of modes of interaction. Higher "ease" scores mean the subject found it easier to identify the material.

In all cases, the identification rate was higher than 50%, and usually much higher than that. The loss of haptic feedback caused the largest drop in ID rate and ease. The loss of sound actually improved material identification—although the difference is not statistically significant—but subjects still found identification to be much more perceptually challenging.

Two more noteworthy results were gathered from a subjective questionnaire, with results shown in Table 5.3. Subjects were asked how frequently they used each of the modes in identifying the material. The subjects were also asked how well each mode of interaction represented how they would expect the materials to sound or feel. These results could help explain the low identification rate when haptics are disabled: most

	Guesses (%)									
ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	**50**	0	33	0	0	17	0	0	10	0
2	0	**80**	0	20	0	0	0	0	0	0
3	0	0	**100**	0	0	0	0	0	0	0
4	0	0	0	**83**	17	0	0	0	0	0
5	0	13	25	0	**50**	0	12	0	0	0
6	0	0	17	0	0	**83**	0	0	0	0
7	8	0	8	0	0	8	**60**	8	8	0
8	0	0	0	0	0	0	0	**75**	25	0
9	0	0	17	0	0	0	0	16	**67**	0
10	0	0	0	0	0	0	0	0	12	**88**

Table 5.4: Confusion matrix showing the guesses made by subjects in the texture identification study. For all materials, a significant majority of subjects were able to identify the right materials.

subjects both relied heavily on tactile senses and found it be the most accurate mode. The subjects considered the sound and physics somewhat less accurate but still occasionally useful for determining the materials.

More detailed results from the study are presented in Table 5.4. An entry in row i and column j is the percentage of times the subject was presented material i and guessed that it was material j. The higher percentages along the diagonal demonstrate the high correct identification rate. Also note that in most categories there is no close second-place guess. The largest exception is that 33% of the time material 1 (brick grid) was mistakenly identified as material 3 (pebbles), likely due to similarity in both material sounds and patterns.

5.4.2.3 Analysis

Our analysis is largely based on comparing the results from interactions with different sets of modalities using a t-test to analyze the difference between the modalities. In addition to the p value for statistical significance, we also use Cohen's effect size d, defined as the difference between the means of two samples divided by their pooled standard deviation (Nakagawa and Cuthill, 2007). Effect size is an important factor to consider alongside statistical significance, explaining not just if there is a difference, but explaining (in units of standard deviations) how large that difference actually is.

Due to the relatively low sample size in the study of each material, many of the possible direct comparisons would not be statistically significant. Therefore, for this study the reported statistics are based on combined data from all study materials; we do not compare the result on each material to one another.

Between identification rates, there was no statistically significant change when removing a mode ($p > .05$), but the removal of haptics came close with $p = .066$. The subjective subject-reported values of ease and accuracy were generally more significant. Subjects reported that they found material identification to be more difficult when either sound or haptics were removed in comparison to having all modes available ($p < .05$), but did not find identification more difficult when the physics modification was removed ($p > .05$). Cohen's effect size values (d) of 1.66 for the removal of sound and 2.79 for the removal of haptics suggest a very large change in perceptual difficulty when removing these modes. Subjects also reported that they found the haptics to be more accurate than physics or sound ($p < .05$), but did not find a significant difference in accuracy between physics and sound ($p > .05$). Cohen's effect size values of 1.02 comparing haptics to physics and 1.36 comparing haptics to sound suggest a large difference in the perception of how accurate these modes are.

Overall, these results demonstrate that each mode of interaction is effectively enabled through use of normal maps. Combining multiple modes increases accuracy, which suggests that the subjects are receiving consistent, non-conflicting information across their senses. This was a deliberately challenging study, using materials that sounded similar and had similar geometric features and patterns. Furthermore, the task asked subjects to carefully consider properties of materials not often noticed. Not many people take the time to consider the difference in frequency distributions between the sounds of porcelain and metal, but that distinction could have been important for these tasks. Within such a context, a 78% rate for identifying the correct material out of ten options appears rather promising, and significantly better than random selection.

5.4.3 Normal and Relief Comparison User Study

We now move on to discuss a second, separate user study. In order to evaluate the effectiveness of the relief map representation, we conducted another user study where subjects compared normal mapped surfaces to relief mapped surfaces. Since the previous study found most of the benefit in the subjects' perception of the surface, this study was largely designed to test the perceptual aspects of these representations.

Figure 5.6: The available materials for the normal and relief map comparison user study. Material 2 and 5 sounded like stone; 3 sounded like ceramic tile; 4 sounded like metal; 1 and 6 sounded like wood.

5.4.3.1 Set-up

Twenty-two subjects volunteered to participate in this study, primarily students with computer literacy in the age between 20 to 30. The subjects were allowed to interact with six textured surfaces, where, for each subject, three textures were randomly selected to use the normal map representation and the remaining three used the relief map representation. Much like in the previous user study, subjects controlled the PHANToM, which corresponded to a virtual pen that could strike the surface or a rolling ball. Through this interaction the subjects would feel the surface, watch the ball roll across the surface, and hear sound synthesized from the surface. Subjects were given as much time as needed to interact with the textured surfaces, and were able to switch between textures at will. Feedback was obtained through a questionnaire in which subjects evaluated each texture, rating the perceived realism of the visual appearance, how well each mode of interaction matched what they would expect from the visual appearance, and the overall quality of interaction.

Figure 5.6 shows the relief map versions of each surface chosen for the user study. These were selected to provide a range of complexity, depth, and materials. The subjects were allowed to spend as much time as needed to properly evaluate each surface.

The subjects were not informed that some surfaces would have relief maps and some would have normal maps, nor were they specifically told to consider the depth of the surface. Furthermore, no subject ever saw

both the normal and relief versions of the same surface, always one or the other. With the subjects largely going into the study unaware of the multiple representations, we pose the following questions:

- With this scenario, do the subjects find the relief maps more accurate and realistic? If not, do they instead significantly prefer the normal maps, or are the two representations indistinguishable?
- Do subjects interacting with a relief mapped surface rate it more highly than the subjects interacting with its normal map equivalent?
- How much, if any, does depth information help with reduction of sensory conflict?

5.4.3.2 Experimental Results

A general way to look at the results is to, for each question, compare all responses (across all surface materials) to use of normal maps vs. use of relief maps. This way can provide a general idea of which texture representation was preferred for each mode of interaction. When subjects were asked how realistic the surfaces appeared, how much the ball physics matched their expectations, and how much the synthesized sound matched their expectations, there was no significant difference between normal maps and relief maps ($p >> .05$). Cohen's effect size for each of these was no greater than 0.11, further indicating little distinction between the texture representations.

When subjects were asked how well the haptics matched their expectations, there was weak evidence showing that subjects preferred the relief maps ($p \approx .053$), and Cohen's effect size of .34 indicates some moderate preference of relief maps. However, when subjects reported their overall perceived quality of interaction, they significantly favored relief maps over normal maps ($p < .05$), with Cohen's effect size of .36 further suggesting a moderate preference of relief maps.

In Table 5.5, we show the results from comparing the two versions of each texture to one another. For each of the six surfaces, the ratings from the subjects who were given the normal map version are compared to the ratings from the subjects who were given the relief map version, and the table presents the p values and effect sizes for each category the subjects were questioned about. See the beginning of Section 5.4.2.3 for a brief description of effect size. Notice that the results vary largely from surface to surface.

Recall that, out of the six surfaces each subject experienced, three at random were chosen to be normal maps and the other three were relief maps. Comparing each subject's average normal map rating to that same

		Surface					
		1	**2**	**3**	**4**	**5**	**6**
Visuals	p	.03	.61	.96	.14	.21	.66
	d	.84	-.21	.03	.65	-.57	.18
Physics	p	.80	.64	.38	.83	.08	.56
	d	-.1	.20	-.4	.09	-.78	.25
Sound	p	.31	.84	.47	.27	.07	.14
	d	-.45	-.09	-.34	.49	-.83	.65
Haptics	p	.03	.70	.77	.03	.002	.002
	d	.9	.16	.16	1.03	-1.42	1.44
Overall	p	.2	.68	.92	.08	.14	.02
	d	.52	.18	.05	.80	-.65	1.02

Table 5.5: For each of the six surfaces, subjects interacted with either the normal or relief map version of that surface's texture. This table contains results of t-tests for each surface and each modality determining whether there are significant differences between the subjects' responses for each texture representation. A small p indicates a statistically significant difference. A positive d value indicates that subjects prefer the relief map version; negative indicates a preference for the normal map.

subject's average relief map rating, we found that each subject tended to prefer their three relief maps over their three normal maps ($p < .05$).

5.4.3.3 Analysis

We can now revisit our originally posed questions, which each involve different means of analyzing the data:

Accuracy and realism of relief maps In order to assess the overall quality of interaction with relief maps, we can consider the data in aggregate, regardless of surface or user. Based on the subjects' ratings of the surfaces' overall quality across all surfaces, on average subjects preferred relief maps over normal maps. We also know that, despite not being informed of the multiple representations, subjects significantly preferred their three randomly selected relief maps over their three normal maps. This neglects the subjects' opinions on individual modes of interaction, but that will be discussed later in the context of sensory conflict. When considered as a whole, relief maps were considered to be of somewhat better overall quality.

Comparisons between normal and relief map versions of the same surface In order to see how subjects compared different versions of the same surface, we now focus on the data in Table 5.5, which groups ratings

by surface. When broken up in this way, we now see that results varied greatly from surface to surface. For most surfaces and most modes of interaction, the differences in ratings were not statistically significant, and the effect sizes ranged from medium preference of the normal map to medium preference of the relief map. Certain textures therefore may be more suitable for representation as relief maps than others. For example, subjects often commented that haptics and ball physics were unrealistic near vertical edges in a relief map (likely due to limitations of directional penetration depth). Surface five contained many prominent near-vertical edges, and subjects strongly preferred the normal map version. Even though there is an average preference for relief maps across all surfaces, this and other situational reasons for preferring a particular representation mean that the choice of representation may need to be considered on a case-by-case basis.

Reduction of sensory conflict In order to assess sensory conflict, we now see if the results indicate that the experience as a whole was more appealing than each separate modality would indicate. Preferences were mixed when subjects were told to rate a specific mode of interaction, but they rated the overall quality of relief maps to be significantly higher than normal maps. This suggests that when interacting with multiple modes of interaction simultaneously, relief maps appear to produce more consistent multimodal interaction than normal maps. Normal vectors already provided most of the cues for depth and curvature, so adding depth information in the form of a relief map had only a small effect on any one mode of interaction. It is only when all modes are considered together that the combined effect is significantly larger. While the overall quality of interaction with reliefs maps may be only moderately better on average and dependent on traits of the surface itself, this reduction in sensory conflict provides its own, possibly subconscious, advantages.

5.4.4 Discussion

5.4.4.1 Applications

We demonstrate several possibilities on the potential use of normal and relief maps as unified representations for accelerating multimodal interaction. See Figures 1.6 and 5.7 for examples applications. Given the prevalence of texture mapping in numerous interactive 3D graphics applications (e.g. games and virtual environment systems), our techniques enable the users to interact with textured objects that have extremely simple underlying geometry (such as flat surfaces) so that they would be able to observe *consistent* dynamic behaviors of moving textured objects, hear the resulting sounds from collisions between them, and feel the object contacts, as shown in Figure 5.7 (left). The example of the simplified pinball game in Figure 1.6 (right),

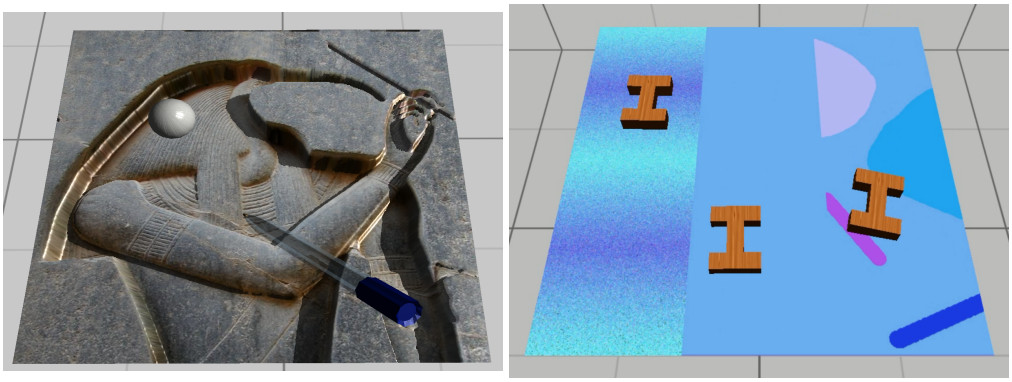


Figure 5.7: A selection of applications based on our system: a virtual environment with multimodal interaction with a relief map used in the normal and relief map comparison user study (top left), and letter blocks sliding down a normal-mapped surface (bottom left).

Figure 5.8: Lombard street color map with normal map (left) and mapped to a plane with rolling balls (right).

balls rolling down Lombard Street in San Francisco City in Figure 5.8, and letter blocks sliding down sloped surfaces with noise or obstacles in Figure 5.7 (right) are a few additional examples, where texture maps can be incorporated into physics simulation with multimodal display to provide a more consistent, immersive experience without sensory disparity.

5.4.4.2 Comparison with Level-of-Detail Representations

While we have shown comparisons between normal maps and high-resolution meshes as representations of fine detail, using multiple levels-of-detail when appropriate can also improve runtime performance (Otaduy and Lin, 2003b,a; Yoon et al., 2004). These LOD meshes can also reduce the complexity of the geometry

while trying to retain the most important features, as determined by perceptual metrics. Since human perception is limited, there may be no significant perceptual benefit in using meshes past a certain quality, in which case the simplified version could be used throughout for significant performance gain.

However, there would be a number of challenges to overcome in designing a multimodal LOD system. The metrics defining important visual features are known to be different than the metrics defining important haptic features (Otaduy and Lin, 2005). It remains an open problem to create metrics for selecting important audio features for switching between LODs in a multimodal system. Furthermore, the haptic LOD meshes are different from LOD meshes for visual rendering (Otaduy and Lin, 2005), leading to significantly higher memory requirements than texture-based representation in general.

5.5 Summary

In this chapter, we presented an integrated system for multimodal interaction with textured surfaces. We demonstrated that normal maps and relief maps can be used as unified representations of fine surface detail for visual simulation of rigid body dynamics, haptic display and sound rendering. We showed that in a system that uses normal maps to present fine detail to subjects through multiple modes of interaction, subjects are able to combine this information to create a more consistent mental model of the material they are interacting with. Our first user evaluation result further provides validation that our system succeeded in reducing sensory conflict in virtual environments when using texture maps. Our second user evaluation result demonstrates that relief maps, when chosen carefully, may produce a further reduction in sensory conflict.

We have now explored two different texture representations of fine detail, but some limitations should be addressed. Our current implementation and studies limited the texture-mapped surfaces to single flat planes and we assume our multimodal method would translate gracefully to more complex shapes, as techniques exist for *visually* rendering relief maps on arbitrary polygonal surfaces (Policarpo et al., 2005). We have also been detecting collisions only between static relief-mapped surfaces and dynamic *non-relief-mapped* objects. A more generalized and versatile system could consider the texture of both colliding textured objects, even if both are dynamic, although performance may become more of a limitation. Vectorial textures may be used to help reducing the aliasing artifacts of relief maps in better rendering sharp edges. Additionally, our choice of haptic device has limited our results to 3-DOF force feedback, though it should be possible to compute torques with a slight extension of our method.

For future research, it may be possible to explore the integration of material perception (Ren et al., 2013a,b) for multimodal displays. Future work may also attempt to generalize this system by addressing the limitations described. We hope this work will lead to further interest in development of techniques on minimizing sensory conflicts when using texture representations for interactive 3D graphics applications, like AR and VR systems.

CHAPTER 6: ISNN: Impact Sound Neural Network for Audio-Visual Object Classification[1]

6.1 Introduction

The problem of object detection, classification, and segmentation are central to understanding complex scenes. Detection of objects is typically approached using visual cues (Girshick et al., 2014; Ren et al., 2015). Classification techniques have steadily improved, advancing our ability to accurately label an object by class given its depth image (Wu et al., 2015), voxelization (Maturana and Scherer, 2015), and/or RGB-D data (Socher et al., 2012). Segmentation of objects from scenes provides contextual understanding of scenes (Golodetz* et al., 2015; Valentin et al., 2015). While these state-of-the-art techniques often result in high accuracy for common scenes and environments, there is still room for improvement when accounting for different object materials, textures, lighting, and other variable conditions.

The challenges introduced by transparent and highly reflective objects remain open research areas in 3D object classification. Common vision-based approaches cannot gain information about the internal structure of objects, however audio-augmented techniques may contribute that missing information. Sound as a modality of input has the potential to close the audio-visual feedback loop and enhance object classification. It has been demonstrated that sound can augment visual information-gathering techniques, providing additional clues for classification of material and general shape features (Zhang et al., 2017b; Arnab et al., 2015). However, previous work has not focused on identifying complete object geometries. Identifying object geometry from a combined audio-visual approach expands the capabilities of scene understanding.

In this chapter, we consider identification of rigid objects such as tableware, tools, and furniture that are common in indoor scenes. Each object is identified by its geometry and its material. A discriminative factor for object classification is the sound that these objects produce when struck, referred to as an *impact sound*. This sound depends on a combination of the object's material composition and geometric model. Impact

[1]This chapter previously appeared as a paper in the European Conference on Computer Vision (ECCV 2018). The original citation is as follows: Sterling, A., Wilson, J., Lowe, S., and Lin, M. C. (2018). ISNN: Impact sound neural network for audio-visual object classification. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 578–595, Cham. Springer International Publishing

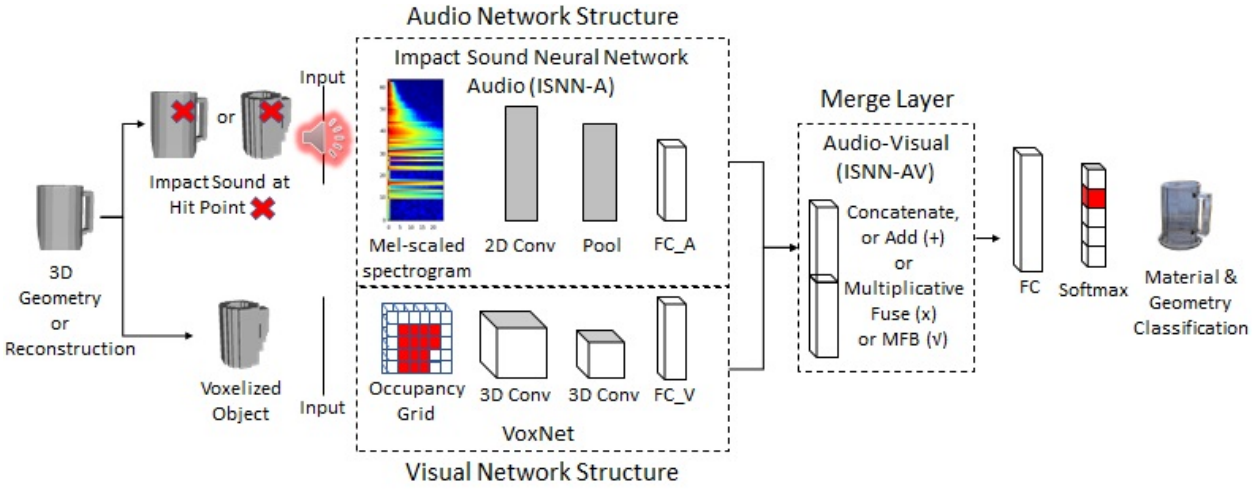


Figure 6.1: Our Impact Sound Neural Network - Audio (ISNN-A) uses as input a spectrogram of sound created by a real or synthetic object being struck. Our audio-visual network (ISNN-AV) combines ISNN-A with VoxNet to produce state-of-the-art object classification accuracy.

sounds are distinguished as object discriminators from video in that they reflect the internal structure of the object, providing clues about parts of an opaque or transparent object that cannot be seen visually. Impact sounds, therefore, complement video as an input to object recognition problems by addressing the some inherent limitations of incomplete or partial visual data.

Main Results: We introduce an audio-only Impact Sound Neural Network (ISNN-A) and a multimodal audio-visual neural network (ISNN-AV). These networks:

- Are the first networks to show high classification accuracy of both an object's geometry and material based on its impact sound;
- Use impact sound spectrograms as input to reduce overfitting and improve accuracy and generalizability;
- Merge multimodal inputs through bilinear models, which have not been previously applied to audio-visual networks yet result in higher accuracy as demonstrated in Table 6.4;
- Provide state-of-the-art results on geometry classification; and
- Enable real time, interactive scene reconstruction in which users can strike objects to automatically insert the appropriate object into the scene.

Figure 6.2: We use various datasets for training and testing: (1) our RSAudio dataset with real and synthesized impact sounds from objects of varying shapes and sizes and (2) voxelized ModelNet objects. (3) Audio inputs are formatted as spectrograms.

6.2 Audio and Visual Datasets

To perform multimodal classification of object geometries, we need datasets containing appropriate multimodal information. Visual object reconstruction can provide a rough approximation of object geometry, serving as one form of input. *Impact audio produced from real or simulated object vibrations provide information about internal and occluded object structure, making for an effective second input.* Figure 6.2 provides examples of object geometries, while the corresponding spectrograms model the sounds that provide another input modality.

Appropriate audio can be found in some existing datasets, but the corresponding geometries are difficult to model. AudioSet contains impact sounds in its "Generic impact sounds" and "{Bell, Wood, Glass}" categories (Gemmeke et al., 2017), while ESC-50 has specific categories including "Door knock" and "Church bells" (Piczak, 2015b). The *Greatest Hits* sound dataset comes closest to our needs, containing impact sounds labeled according to the type of object (Owens et al., 2016a). However, many of the categories do not contain rigid objects (*e.g.*, cloth, water, grass) or contain complex structures that cannot be represented with one geometric model of one material (*e.g.*, a stump with roots embedded in the ground).

We want to use an impact sound as one input to identify a specific geometric model that could have created that sound. A classifier for this purpose could be trained on a large number of recorded sounds produced from struck objects. However, it is difficult and time-consuming to obtain a representative sample of real-world objects of all shapes and sizes. It is much easier to create a large dataset of synthetic sounds using geometric shapes and materials which can be applied to the objects. We now describe our methodology for generating the data used for training, as visualized in Figure 6.3.

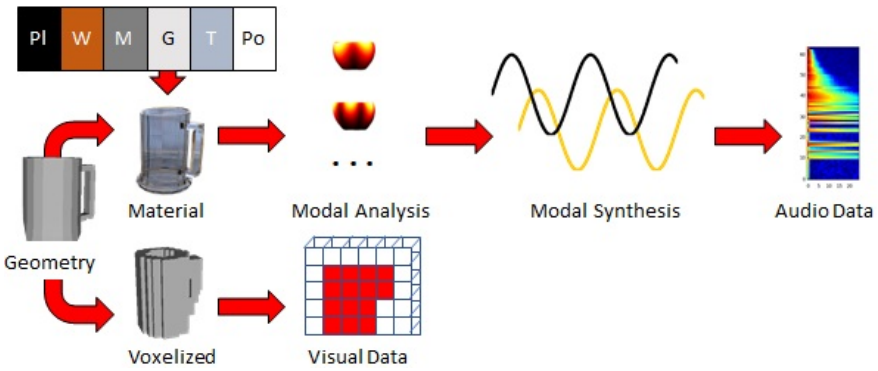


Figure 6.3: We build multimodal datasets through separate processing flows. Modal sound synthesis produces spectrograms used for audio input. Voxelization as another modality provides a first estimate of shape. Incorporating audio features improves classification accuracy through understanding of how objects vibrate.

6.2.1 Audio Data

We create a large amount of our training data by simulating the vibrations of rigid-body objects and the sounds that they produce. We use the established process of modal sound synthesis to create synthetic sound datasets from 3D models. The process of modal sound synthesis is described in detail in Section 2.1.

6.2.2 Audio Augmentations

Modal sound synthesis produces the set of frequencies, damping rates, and initial amplitudes of an object's surface vibrations. However, since we are attempting to imitate real-world sounds, there are some additional auditory effects to take into account: acoustic radiance, room acoustics, background noise, and time variance.

Acoustic Radiance: Sound waves produced by the object must propagate through the air to reach a listener or microphone position. Even in an empty space, the resulting sound will change with different listener positions depending on the vibrational mode shapes; this is the acoustic radiance of the object (James et al., 2006). This effect has a high computational cost for each geometric model, and since we use datasets with relatively large numbers of models, we do not include it in our simulations.

Room Acoustics: In an enclosed space, sound waves bounce off walls to produce early echo-like reflections and noisy late reverberations; this is the effect of room acoustics. We created a set of room impulse responses in rooms of different sizes and materials using a real-time sound propagation simulator, GSound (Schissler and Manocha, 2011). Each modal sound is convolved with a randomly selected room impulse response.

Background Noise: In most real-world situations, background noise will also be present in any recording. We simulate background noise through addition of a random segment of environmental audio from the DEMAND database (Thiemann et al., 2013). These noise samples come from diverse indoor and outdoor environments and contain around 1.5 hours of recordings.

Time Variance: Finally, we slightly randomize the start time of each modal sound. This reflects the imperfect timing of any real-world recording process. Together, these augmentations make the synthesized sounds more accurately simulate recordings that would be taken in the real world.

6.2.3 Visual Data

Our visual data consists of datasets of geometric models of rigid objects, ranging from small to large and of varying complexity. Given these geometric models, we can simulate synthesized sounds for a set of possible materials. During evaluation, object classification results were tested using multiple scenarios of voxelization, scale, and material assignment (Section 6.4.2).

6.3 Impact Sound Neural Network (Audio & Audio-Visual)

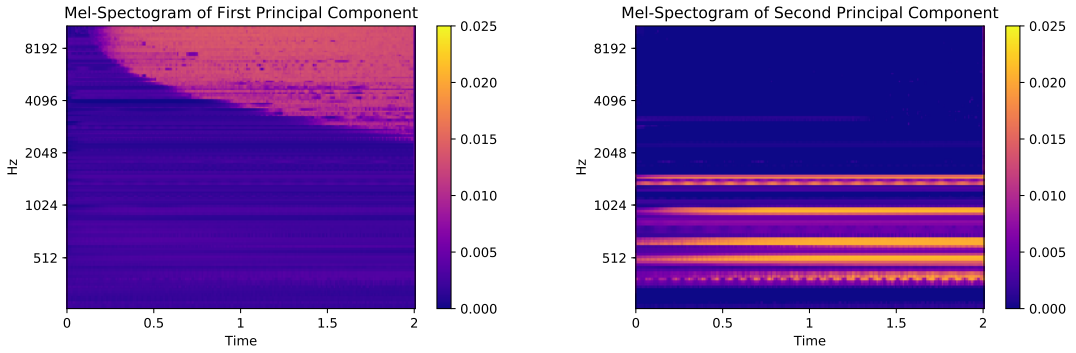
Given the impact sounds and representation described in Section 6.2, we now examine their ability to identify materials and geometric models. We begin with an analysis of the distributions of the features themselves as proper feature selection is a key component in classifier construction.

6.3.1 Input Features and Analysis

6.3.1.1 Audio Features

In environmental sound classification tasks, classification accuracy can be affected by the input sound's form of representation (Cowling and Sitte, 2003; Huzaifah, 2017). A one-dimensional time series of audio samples over time can be used as features (Aytar et al., 2016), but they do not capture the spectral properties of sound. A frequency dimension can be introduced to create a time-frequency representation and better represent the differentiating features of audio signals.

In this work, we use a mel-scaled spectrogram as input. Spectrograms have demonstrated high performance in CNNs for other tasks (Huzaifah, 2017). A given sound, originally represented as a waveform of audio samples over time, is first trimmed to one second in length since impact sounds are generally transient.



(a) (b)

Figure 6.4: The first two principal components of 420 synthesized sounds demonstrate that the key differentiating factors between sounds and models are the presence of high-frequency damping (first component) and the presence of specific frequency bins (second component).

The sound is resampled to 44.1 kHz, the Nyquist rate for the full range of audible frequencies up to 22.05 kHz. We compute the short-time Fourier transform of the sound, using a Hann window function with 2048 samples and an overlap of 25 %. The result is squared to produce a canonical "spectrogram", then the frequencies are mapped into mel-scaled bins to provide appropriate weights matching the logarithmic perception of frequency. Each spectrogram is individually normalized to reduce the effects of loudness and microphone distance. To create the final input features for the classifier, we downsample the mel-spectrogram to a size of 64 frequency bins by 25 time bins.

We performed principal component analysis on a small sample of synthesized impact sounds to demonstrate the advantage of mel-spectrograms as input features for audio of this type. We used 70 models and 6 materials with a single hit per combination to synthesize a total of 420 impact sounds for this analysis. Figure 6.4 displays the first two principal components as mel-spectrograms, describing important distinguishing factors in our dataset. The first component identifies damping in higher frequencies, while the second component identifies specific frequency bins.

Figure 6.5 contains a scatter plot of material classes on the axes of the first two principal components. The first principal component explains much of the variation between material classes, as there is clear horizontal delineation—albeit with overlap. This is consistent with the expectation of damping as a material-dependent property. The presence of specific frequency bins that comprise the second component likely delineates model more than material.

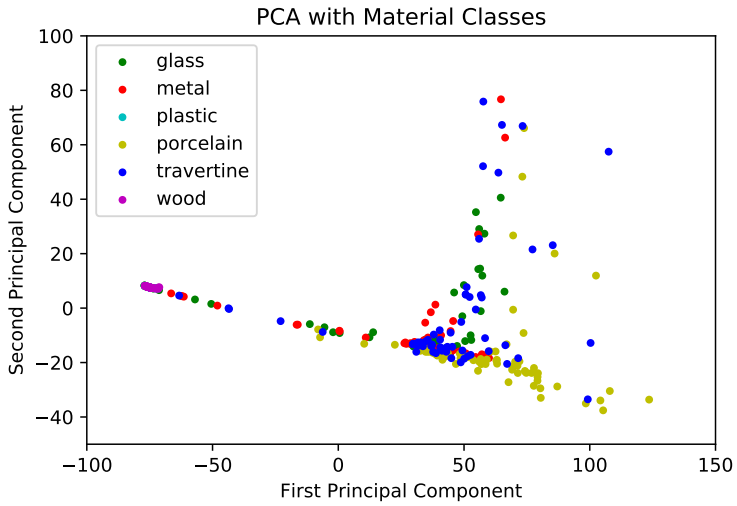


Figure 6.5: A scatter plot of material classes on the first two principle component axes. While the horizontal delineation of materials is useful in characterizing those sounds, a full understanding of the relationships between materials and models necessitates a deeper classification scheme.

6.3.1.2 Visual Features

As in VoxNet (Maturana and Scherer, 2015), visual data serves as an input into classification models based on a 30x30x30 voxelized representation of the object geometry. We voxelize models from our real and synthetic dataset and ShapeNets ModelNet10 and ModelNet40. All objects were voxelized using the same voxel and grid size. We generated audio and visual data for our dataset and up to 200 objects (train and test) per ModelNet class.

6.3.2 Model Architecture

Using our audio and visual features, our approach to performing object geometry classification uses convolutional neural networks (CNNs) due to their high accuracy in a wide variety of tasks, with the specific motivation that convolutional kernels should be able to capture the recurring patterns underlying the structure of our sounds.

6.3.2.1 Audio-Only Network (ISNN-A)

We first developed a network structure to perform object classification using audio only. Our audio Impact Sound Neural Network (ISNN-A) is based on optimization performed over a search space combining general network structure, such as the number of convolutional layers, and hyperparameter values. This

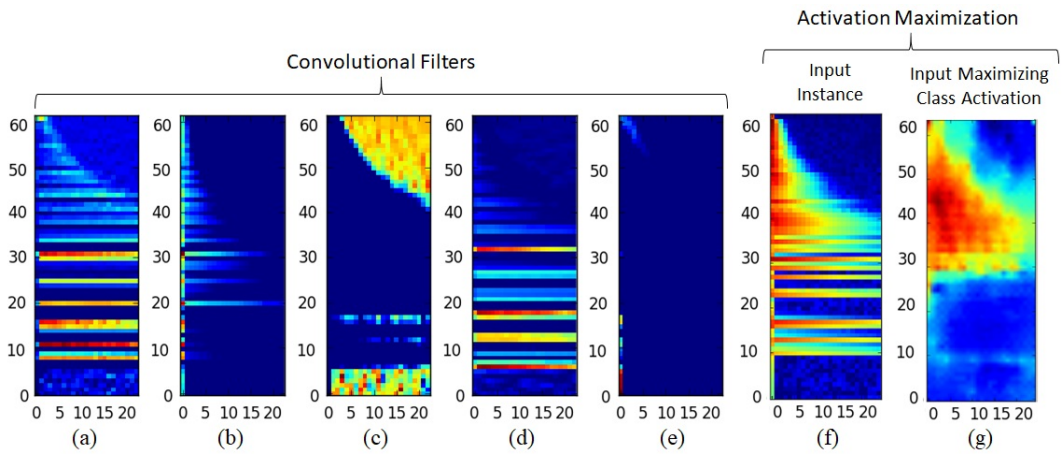


Figure 6.6: Sample activations (a-e) of ISNN convolution layer. Filters identify characteristic patterns in frequencies (a) (d), damping rates (b) (c), and high-frequency noise (e). The distinguishing characteristics in these activations match the expected factors discovered in the PCA analysis in Figure 6.4. An audio input spectrogram (f) and activation maximization (g) learned by the ISNN network for the toilet ModelNet10 class show correctly-learned patterns.

optimization was performed using the TPE algorithm (Bergstra et al., 2011). We found a single convolutional layer followed by two dense layers performs optimally on our classification tasks. This network structure utilizes a convolution kernel with increased frequency resolution to more effectively recognize spectral patterns across a range of frequencies (Piczak, 2015a). Our generally low number of filters and narrower layer sizes aim to reduce overfitting by encouraging the learning of generalizable geometric properties.

Figure 6.6 shows sample activations of a convolutional layer of the ISNN-A network. Based on the PCA and modal analysis we performed, we expect that the differences between geometries primarily manifest as different sets of modal frequencies, as well as different sets of initial mode amplitudes and damping rates. These activations corroborate our expectations. In Figure 6.4(a), we see that damping is an important discriminating feature, which has been learned by filters (b) and (c) in Figure 6.6. Similarly, the frequency patterns that we expected because of Figure 6.4(b) can be seen in filters (a) and (d). This demonstrates that our model is learning statistically optimal kernels with high discriminatory power.

6.3.2.2 Multimodal Audio-Visual Network (ISNN-AV)

Our audio-visual network, as shown in Figure 6.1, consists of our audio-only network combined with a visual network based on VoxNet (Maturana and Scherer, 2015) using either a concatenation, addition, multiplicative fusion, or bilinear pooling operation. Concatenation and addition serve as our baseline

operations, in which the outputs of the first dense layers are concatenated or added before performing final classification. These operations are not ideal because they fail to emulate the interactions that occur between multiple forms of input. On the other hand, multiplicative interactions allow the input streams to modulate each other, providing a more accurate model.

We evaluate two multiplicative merging techniques to better model such interactions. Multiplicative fusion calculates element-wise products between inputs, while projecting the interactions into a lower-dimensional space to reduce dimensionality (Park et al., 2016). Multimodal factorized bilinear pooling takes advantage of optimizations in size and complexity, and is our final merged model (Yu et al., 2017). This method builds on the basic idea of multiplicative fusion by performing a sequence of pooling and regularization steps after the initial element-wise multiplication.

6.4 Results

We now present our training and evaluation methodology along with final results. For each of the datasets, we evaluate the network architectures described in Section 6.3.2. We compare against several baselines: a K Nearest Neighbor classifier, a linear SVM trained through SGD (Bottou, 2010), VoxNet (Maturana and Scherer, 2015), and SoundNet (Aytar et al., 2016). Our multimodal networks combined VoxNet with either ISNN-A or SoundNet8 and were merged through either concatenation (MergeCat), element-wise addition (MergeAdd), multiplicative fusion (MergeMultFuse) (Park et al., 2016), or multimodal factorized bilinear pooling (MergeMFB) (Yu et al., 2017). Training was performed using an Adam optimizer (Kingma and Ba, 2015) and run with a batch size of 64, with remaining hyperparameters hand-tuned on a validation set before final evaluation on a test set.

6.4.1 RSAudio Evaluation

Our "RSAudio" dataset was constructed from real and synthesized sounds. When performing geometry classification, each geometric model is its own class; given a query sound, the network returns the geometric model that would produce the most similar sound. RSAudio combines real and synthetic sounds to increase dataset size and improve accuracy.

Geometry Classification Accuracy: RSAudio and Related Work Datasets (ISNN-A Ours)

Method	Input	RSA S	RSA R	RSA Merged	Sound-20K*	Arnab A	ImageNet
Nearest Neighbor	A	**96.92%**	68.63%	97.59%	95.54%	87.50%	N/A
Linear SVM (Bottou, 2010)	A	2.31%	2.30%	3.20%	82.07%	7.14%	N/A
SoundNet5 (Aytar et al., 2016)	A	94.74%	16.10%	**97.70%**	58.81%	23.21%	N/A
SoundNet8 (Aytar et al., 2016)	A	83.83%	4.24%	89.62%	71.43%	58.93%	N/A
ISNN-A	A	96.74%	**92.37%**	97.07%	**99.52%**	**89.29%**	N/A
Pre-Trained VGG16	V	N/A	N/A	N/A	N/A	N/A	73.27%

Table 6.1: For real sounds, ISNN-A significantly outperforms all other methods, with an accuracy up to 92.37 %. For some synthetic datasets, ISNN-A produces results competitive with the top-performing methods. *Based on a subset of Sound-20K.

62 400 synthesized sounds come from a set of 59 geometric models and 11 sets of material parameters categorized into 6 classes of materials. For each model and material pairing (with a few exceptions), 100 sounds with random hit points were synthesized.

1183 real impact sounds come from a set of 24 struck rigid objects. These objects are each made of one homogeneous material and primarily consist of dining dishes, utensils, tools, and material samples used for building construction. A majority of the sounds were recorded in a padded sound booth using a Zoom H4 microphone to reduce background noise and room acoustics. The remaining sounds were recorded in a wider set of environments ranging from small offices to large outdoor areas. Each recording contains one impact in isolation from other impacts.

Objects were either struck with a small metal wrench or a rubber-headed drumstick, and in most cases, both. In either case, the striking tool was tightly gripped in a hand while striking in order to minimize its vibrations while the main struck object could vibrate freely. No post-processing was performed to attempt to remove the remaining sound from the striking tool.

The results for geometry classification are presented in Tables 6.1 to 6.4. For RSAudio synthetic (S) and real (R), ISNN-A provides competitive results with all other tested algorithms. For real sounds, where issues of recordings are most problematic, ISNN-A significantly outperforms all other algorithms, with an accuracy of 92.37 %. On the merged RSAudio dataset of real and synthetic sounds, all models actually produce *higher* accuracy than on either synthetic or real alone, indicating that training on both sets improves generalizability. As an additional baseline, we classified 100 ImageNet RGB transparent object images based on the VGG16 pre-trained model and obtained 73.27% accuracy with top 5 labels and an average confidence of 46.64%.

Geometry Classification Accuracy: Audio Methods (ISNN-A Ours), ModelNet

Method	Input	MN10o	MN10os	MN10om	MN10osm	MN10	MN40o	MN40osm
Nearest Neighbor	A	40.73%	32.42%	62.81%	67.97%	—	26.55%	54.41%
Linear SVM	A	16.67%	7.81%	28.85%	15.63%	11.73%	3.97%	12.18%
SoundNet5	A	16.96%	10.00%	10.70%	11.00%	—	4.10%	10.95%
SoundNet8	A	10.64%	19.50%	20.74%	29.67%	—	5.73%	49.27%
ISNN-A	A	**43.35%**	**56.50%**	**68.00%**	**71.50%**	**42.90%**	**32.51%**	**65.07%**

Table 6.2: Our audio-only ISNN-A outperforms other audio-only baselines.

Geometry Classification Accuracy: Visual Methods (All Baselines), ModelNet

Method	Input	MN10o	MN10os	MN10om	MN10osm	MN10	MN40o	MN40osm
Nearest Neighbor	V	83.11%	72.57%	82.62%	72.96%	—	65.72%	67.23%
Linear SVM	V	74.06%	66.80%	68.65%	77.34%	35.39%	51.15%	12.06%
VoxNet (Maturana and Scherer, 2015)	V			**89.47%**			**80.17%**	

Table 6.3: VoxNet achieves the highest level of accuracy compared to other alternative methods for geometry classification with visual input only.

While the accuracy is not directly comparable with ModelNet and RSAudio results, it provides a preliminary suggestion that a second modality could further improve results.

6.4.2 ModelNet Evaluation

In Tables 6.2 to 6.4, ModelNet results are categorized by input: audio (A), voxel (V), or both (AV). The "MN10" dataset consists of 119.620 total synthetic sounds: multiple sounds at different hit points for each geometry and material combination. The "o" suffix (*e.g.*, "MN10o") indicates that only one sound per model was produced, and all models were assigned one identical material. The "s" suffix (*e.g.*, "MN10os") indicates that each ModelNet class was assigned a realistic and normally distributed scale before synthesizing sounds. The "m" suffix (*e.g.*, "MN10om") indicates that each ModelNet class was assigned a realistic material.

By assigning a material and scale to each ModelNet10 class (MN10osm), classification performance achieved 71.50% for ISNN-A. Real-world objects within a class will tend to be made of a similar material and scale, so MN10osm is likely more reflective of performance in real-world settings where these factors provide increased potential for classification. However, for the multimodal ISNN-AV, material and scale assignments do not improve accuracy. In MN10o, larger geometric features will correspond to lower-pitched sounds (*i.e.*, a large object will produce a deeper sound than a small object), and the multimodal fusion of those cues produces higher accuracy. However, when models are given materials and scales in MN10o{s,m,sm}, the

Geometry Classification Accuracy: Audio-Visual Methods (ISNN-AV Ours), ModelNet

Method	Input	MN10o	MN10os	MN10om	MN10osm	MN10	MN40o	MN40osm
Nearest Neighbor	AV	82.91%	72.57%	83.40%	74.05%	—	65.84%	71.25%
Linear SVM	AV	80.63%	73.44%	82.50%	81.64%	36.70%	54.93%	66.15%
MergeCat (ISNN-AV)	AV	86.25%	78.50%	88.96%	88.50%	87.40%	79.93%	92.30%
MergeCat (SoundNet8)	AV	88.14%	52.50%	72.80%	54.50%	—	79.56%	56.39%
MergeAdd (ISNN-AV)	AV	88.91%	80.00%	88.52%	86.00%	88.27%	79.40%	90.43%
MergeAdd (SoundNet8)	AV	88.58%	50.50%	72.91%	64.33%	—	79.89%	24.43%
MergeMultFuse (ISNN-AV)	AV	89.14%	84.00%	89.41%	86.24%	87.51%	81.35%	**93.24%**
MergeMultFuse (SoundNet8)	AV	83.48%	66.00%	71.79%	51.67%	—	61.44%	38.97%
MergeMFB (ISNN-AV)	AV	**91.80%**	**84.50%**	**89.97%**	**90.12%**	**89.16%**	**82.04%**	92.51%
MergeMFB (SoundNet8)	AV	88.69%	76.50%	73.02%	42.00%	—	80.90%	91.33%

Table 6.4: Our merged networks produce accuracy upto 90.12 % on MN10osm and upto 93.24 % on MN40osm. Please visit ModelNet for more information on other methods and results.

voxel inputs remain unchanged, weakening the relationship between voxel and audio inputs. Scaling the voxel representation as well as the model used for sound synthesis may reduce this issue.

Assigning scale and material improve ModelNet40 accuracy (MN40osm) because its object classes differ more in size and material than ModelNet10. The merged audio-visual networks outperform the separate audio or visual networks in every case except for MN10os, as discussed above. Across all ModelNet10 datasets, ISNN-AV with multimodal factorized bilinear pooling produces the highest accuracy on MN10o, at 91.80 %. Similarly, ModelNet40 produces optimal results using ISNN-AV with multiplicative fusion on MN40osm, at 93.24 %. Entries with a "—" were not completed due to prohibitive time or memory costs when using the large MN10 dataset.

6.4.3 Additional Evaluations

We evaluated on additional audio-only datasets such as Arnab et. al (Arnab et al., 2015) and Sound-20K (Zhang et al., 2017b), with results displayed in Table 6.1. The Arnab dataset consists of audio of tabletop objects being struck, with ground-truth object labels provided. ISNN-A produces 89.29 % geometry classification accuracy, the highest of all evaluated algorithms. This accuracy is slightly lower than ISNN-A's accuracy on RSAudio's real sounds, likely due to the loosened constraints on the recording environment and striking methodology.

Material Classification Accuracy

Model	RSAudio S	RSAudio R	Arnab Audio (Arnab et al., 2015)
ISNN-A	98.69%	**95.76%**	**71.86%**
SoundNet5 (Aytar et al., 2016)	**99.97%**	29.66%	43.11%
SoundNet8 (Aytar et al., 2016)	92.66%	30.51%	43.11%

Table 6.5: Material classification accuracy on subsets of the RSAudio dataset. Our ISNN-A network produces the highest accuracy on the two real-world datasets, with competitive accuracy on the synthetic RSAudio dataset

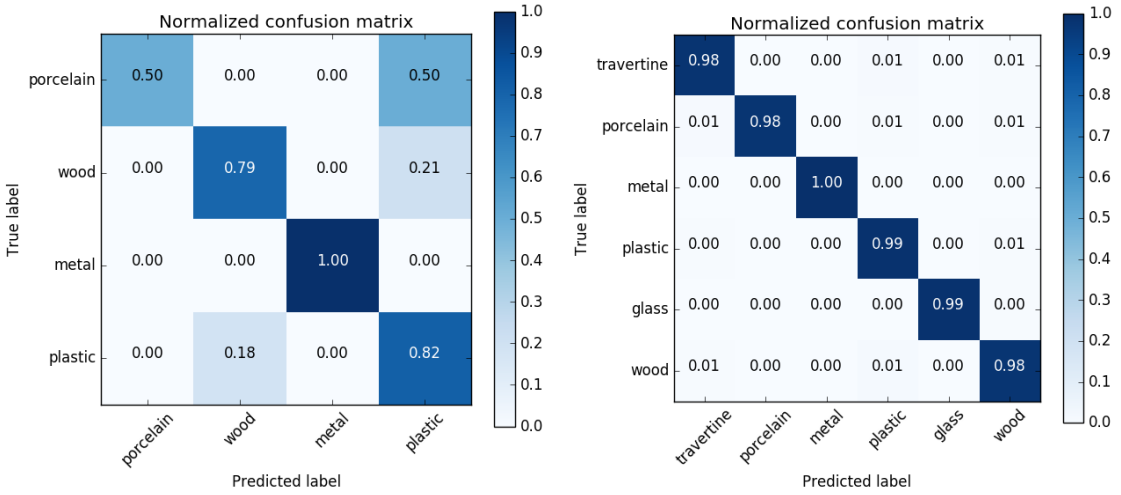
The Sound-20K dataset consists of impact sounds produced from a physics-based simulation, which may produce multiple impacts spread over time. ISNN-A produces 99.52 % geometry classification accuracy, again the highest accuracy of all evaluated algorithms.

6.4.3.1 Material Classification

The ISNN networks can also be trained for the task of material classification. That is, given an input impact sound, ISNN trained in this way will produce an estimate of the material class of the object. This is a task that has been more thoroughly evaluated by previous work, but we are still interested in the performance of ISNN on this same task.

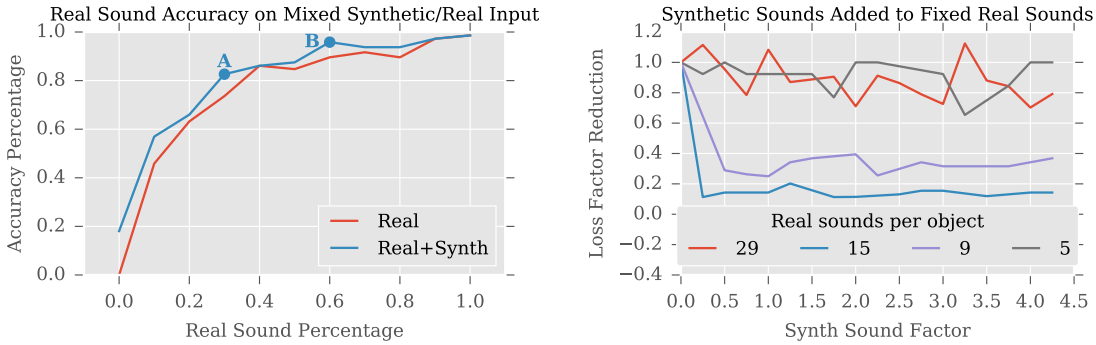
In Table 6.5, we compare the material classification accuracy of various classification models on multiple datasets. ISNN-A produces consistently high accuracy, up to 98.69 %, and is either competitive with or or outperforms SoundNet. The material labels provided with the Arnab dataset are not consistent with those listed in their publication, but we selected a subset of those labels with clearly-distinct material names for this test. In comparison to geometry classification (Table 6.2), material classification accuracies are a few percent higher on the RSA datasets, but somewhat lower on the Arnab dataset, likely due to the labeling discrepancies.

In Figure 6.7, we look at a breakdown of the classifications performed by ISNN on RSAudio's synthetic sounds and Arnab sounds. While RSAudio produces consistently accurate classifications with only minor error, the majority of misclassifications on the Arnab dataset come from porcelain classified as plastic.



(a) (b)

Figure 6.7: Material classification confusion matrices produced by ISNN-A on (a) the Arnab audio dataset and (b) the synthetic subset of RSAudio. In both cases, there is high accuracy with only a minimal amount of confusion.



(a) (b)

Figure 6.8: Classification accuracy on a test set of real sounds using ISNN trained on a combination of real and synthetic sounds. (a) When trained on combined real and synthetic sounds (Real+Synth), classification accuracy is upto 11% higher than when trained on the real sounds alone (Real). (b) When insufficient real sounds are provided, synthetic sounds further reduce loss.

6.4.3.2 Combined Real and Synthetic Training

We also evaluate the ability of synthetic sounds to supplement a smaller number of real sounds for training, which would reduce necessary human effort in obtaining sounds. Figure 6.8 shows classification accuracy on a real subset of our RSAudio dataset for ISNN-A trained on a combination of real and synthetic sounds. The training sets have identical total sizes but are created with specific percentages of real and synthetic sounds, then networks are trained on either the combined dataset or the real sounds independently. We find that the addition of synthetic sounds to the dataset improves accuracy by up to 11 %. With only 30 % real sounds (Point A), accuracy begins to plateau, reaching over 90 % accuracy with only 60 % real sounds (Point B). These indicate that synthetic audio can supplement a smaller amount of recorded audio to improve accuracy.

Augmentations in Section 6.2.2 were designed to enhance the realism of synthetic audio for improved transfer learning from synthetic to real sounds. However, we were unable to find an instance when these augmentations significantly improved test accuracy of RSAudio real when trained on RSAudio synthetic. This indicates that *modal* components of sounds (frequencies, amplitudes) are sufficient and most critical in object classification, and that acoustic radiance, noise, and propagation effects produce little, if any, impact on accuracy.

6.4.3.3 Activation Maximization

We additionally use activation maximization to visualize the spectrogram inputs that would produce the highest activation for a given ModelNet class. Figure 6.9 shows how the result of activation maximization changes as different modifications to ModelNet sounds are performed. When no scale or material are applied, the maximized spectrogram demonstrates a need for robustness to variance in frequency and damping. When scale is fixed, so is the fundamental frequency, as can be seen by the single active region and lower overall activation weights. When material is fixed, so are the damping rates, which become recognizable identifiers for this particular class of object.

6.4.4 Application: Audio-Guided 3D Reconstruction

A primary use case of the ISNN networks is to improve reconstruction of transparent, occluded, or reflective objects. Existing methods have become very effective at 3D scene reconstruction from RGB-D

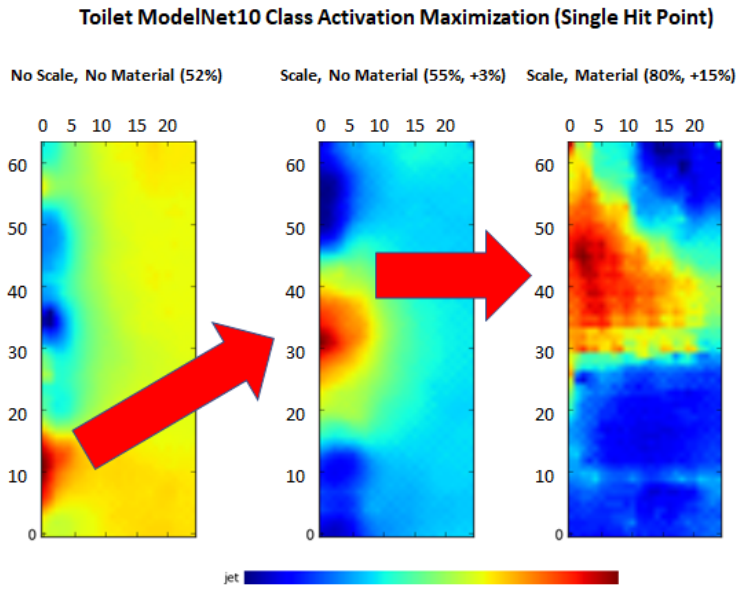


Figure 6.9: Activation maximization results for the Toilet class of ModelNet10 as different modifications are made to the model for sound synthesis. When both scale and material are not fixed as distinguishing factors, the network must be general and robust to differences (left). When both are fixed, the network clearly identifies a recognizable pattern (right).

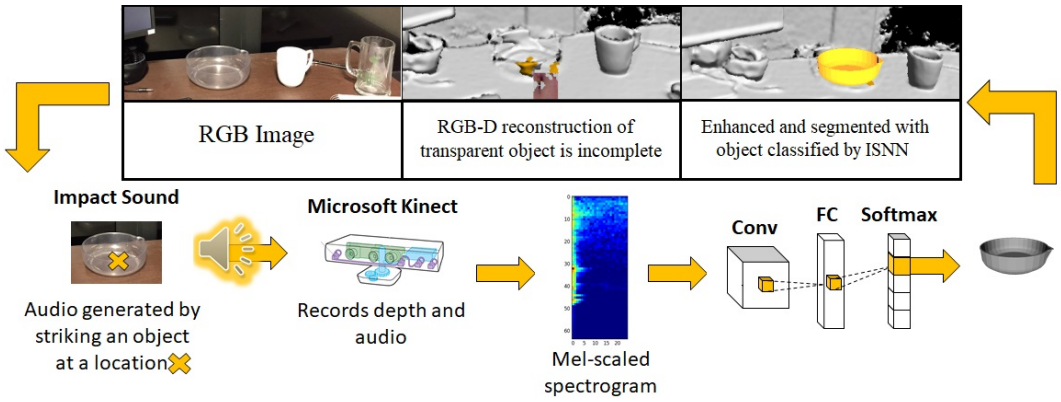


Figure 6.10: A user strikes a real-world object to generate sound as an input into our ISNN network which returns material and object classification. Based on these, the real-time 3D reconstruction is enhanced and segmented.

video, even in real-time. However, due to limitations of vision-based methods, transparent and occluded objects are still a challenge for these methods. We have constructed a demo utility in which our method enables real-time scene reconstruction and augmentation. Figure 6.10 illustrates the application pipeline.

6.4.4.1 Algorithm

The utility at its simplest provides a system for real-time scene reconstruction, based on previous real-time RGB-D work (Golodetz* et al., 2015; Valentin et al., 2015). Using the RGB-D camera of a Kinect, a user scans the scene from multiple angles until estimations have sufficiently converged. At this point, transparent objects may be incomplete or missing. The user interacts with the application to select one of these objects, then physically reaches into the scene to strike the corresponding object.

The Kinect's microphone array records the impact sound, identifies the time of impact, and extracts a 1-second clip containing the sound and its decay. The recorded audio waveform is converted to the form of input to the ISNN-A network: a downsampled mel-scaled spectrogram. This spectrogram is passed through ISNN-A trained on the full RSAudio dataset. One network trained to perform geometric model classification identifies the closest matching geometry to the recorded sound, while another network trained to perform material classification identifies the closest matching material class.

The full object can then be inserted into the reconstructed scene. The object is inserted at the position earlier selected, using the classified geometric model. In our reconstruction utility, the object is textured with a different color than that of the original geometry, indicating the segmentation of the object from the rest of the scene. Alternatively, the material classification could correspond to a texture which could be applied to the object. As a result of this process, the transparent object that had previously been incomplete or missing, has been both completed and segmented.

6.4.4.2 Utility Limitations

ISNN's geometric model classification cannot interpolate or extrapolate geometry given new sounds. When ISNN is trained on the RSAudio dataset, each individual geometric model is considered to be its own class, and classification of a test sound is selection of the closest *training* geometry to that sound. For the utility, this means that the inserted geometric model may be similar to the ground truth object, but not match exactly. Shape optimization from sound is still an open area of research. We have also tested pose estimation methods based on RGB (Brachmann et al., 2016) and RGB-D (Lysenkov and Rabaud, 2013; Lysenkov et al.,

2013); however, future work is needed to extend these to accept asymmetric transparent objects as input and integrate into our application.

6.5 Summary

We presented a novel approach for improving the reconstruction of 3D objects using audio-visual data. Given impact sound as an additional input, ISNN-A and ISNN-AV have been optimized to achieve high accuracy on object classification tasks. The use of spectrogram representations of input reduce overfitting by directly inputting spectral information to the networks. ISNN has further shown higher performance when using a dataset with combined synthetic and real audio. Sound provides additional cues, allowing us to estimate the object's material class, provide segmentation, and enhance scene reconstruction.

Limitations and Future Work: While VoxNet serves as a strong baseline for the visual component of ISNN-AV, different visual networks in its place could identify more optimal network pairings. As with existing learning methods, VoxNet is limited to performing classifications of known geometries. However, impact sounds hold potential of identifying correct geometry, even when a model database is not provided, allowing for accurate 3D reconstructions or hole-filling.

CHAPTER 7: SUMMARY AND CONCLUSIONS

My thesis statement states that "interaction with objects in virtual environments can be made more perceptually realistic by using expressive object material models that account for real-world phenomena and by reducing sensory conflict." To improve the expressiveness of object material models, I presented methods for performing modal sound synthesis with Generalized Proportional Damping and automatically estimating material damping parameters for any GPD-derived damping model from a single impact sound (Chapter 3). To account for real-world phenomena that impact estimation of material damping parameters, I presented a method for automatic material parameter estimation using a probabilistic damping model that encodes these phenomena (Chapter 4). To reduce sensory conflict during interaction with virtual objects, I presented methods for multimodal interaction with textured surfaces using unified texture representations of detail (Chapter 5). To further reduce sensory conflict, I presented a method for estimating object shape and material using joint audio and visual input (Chapter 6). In this chapter, I will summarize these topics.

7.1 Summary of Results

In this section, I summarize the results found by myself and my collaborators. Modal sound synthesis simulates impact sounds produced when a rigid object is struck by modeling the object's vibrations. However, it is limited by the Rayleigh damping model, which is a linear approximation to a more complex phenomenon. Our method for deriving additional damping models for modal sound synthesis increases the expressiveness of material models by better capturing nonlinear damping behavior that Rayleigh damping would only approximate. Our single-sound damping parameter estimation method works for all damping models, making it easy to create virtual objects using these models. In our perceptual study, no damping model was consistently superior, demonstrating that Rayleigh damping cannot express the full variability in damping behavior.

This single-sound damping parameter estimation method is limited in that it requires significant knowledge of the object in addition to the recorded sound, and that the sound must be recorded in a carefully-

controlled environment. These limitations are addressed by our probabilistic damping model, which models real-world phenomena that influence damping estimates and enables robust damping parameter estimation. The only inputs are impact sounds recorded in less-controlled environments, making preparation very simple. Our human hand-tuning study establishes a baseline for human performance on the damping parameter estimation task, which future automated methods may compare against. Our perceptual evaluation found that sounds synthesized using our automatically-estimated parameters produce a pattern of errors similar to that of sounds synthesized using hand-tuned parameters, indicating high perceptual similarity with less human effort. Compared to parameters from Ren et. al (Ren et al., 2013b), our synthesized sounds are perceptually more similar to recorded sounds on three out of four quality metrics.

These methods for damping parameter estimation focus on the perceptual realism of auditory object interactions, but do not account for object interaction's inherently multimodal nature. Our method for multimodal interaction with textured surfaces focuses on the perceptual realism of object interaction through reduced sensory conflict. Our texture identification study found that users perceived texture identification to be easiest when all modalities of interaction were provided. Our study comparing normal and relief maps found that the perceived realism of interaction with relief-mapped surfaces was higher than that of normal mapped surfaces when considering all modalities of interaction. When each modality of interaction was considered independently, normal maps alone were sufficient. These results can guide the design of interactions with textured surfaces, and suggest that multimodal interaction using unified representations of detail can reduce sensory conflict.

Damping parameter estimation methods are also limited by operating in isolation from visual object understanding methods. Our Impact Sound Neural Networks reduce multimodal sensory conflict by using joint multimodal inputs. ISNN-A (audio-only input) and ISNN-AV (combined audio-visual input) provide accurate identification and classification of object shapes and material. ISNN networks are particularly useful when estimating properties of transparent or occluded objects. Our ISNN-A network outperforms models such as SoundNet (Aytar et al., 2016) on audio-only object identification tasks, while our multimodal ISNN-AV network outperforms the visual-only VoxNet (Maturana and Scherer, 2015) on the ModelNet dataset. These results suggest that joint audio-visual estimation of object properties can improve multimodal interaction with virtual objects created based on those properties.

7.2 Limitations

My proposed methods improve both the expressiveness of object damping modeling and multimodal interaction with virtualized objects, but some limitations remain. While the limitations of each method are discussed in its respective chapter, this section relates to the general methodology and assumptions made across my work.

7.2.1 Rigid Object Modeling

First, my proposed methods use linear models of object vibrations. While linear models are critical for real-time synthesis, impact sounds involve significant nonlinear effects. One example is the nonlinear interaction between two object in collision. This interaction is short compared to the total duration of the resulting impact sounds, however, it can significantly affect the *attack* of the sounds. Another example is acceleration noise produced when an object is rapidly accelerated through air. Acceleration noise is nonlinear and perceptually significant for small objects such as shards of broken glass or ceramic (Chadwick et al., 2012). Prior work has attempted to model these nonlinear components of impact sounds as a residual (Ren et al., 2013b), though further analysis of these effects may improve understanding of real-world sounds and synthesis of virtual sounds.

Current damping models are also limited. My presented studies found that no current damping model optimally represents all rigid-object materials Section 3.4.3. The damping models proposed in this dissertation provide more accurate modeling of a subset of materials, but it is a challenge to identify which damping model is ideal for a given real-world material. My methods for damping parameter estimation (Chapters 3 and 4) easily extend to estimate material damping parameters for future damping models.

7.2.2 Object Virtualization

To create realistic virtual versions of real-world objects (object virtualization), virtualizing only an object's audio-material is insufficient. In my work, I often neglect that an object's shape, surface appearance, tactile roughness, and even smell are important attributes of a virtualized object. My parameter estimation methods produce audio-material parameters which directly translate to a virtual object, but must rely on other methods for virtualization of any other object attributes. The ISNN networks take a step towards a

more unified multimodal approach to virtualization, but sacrifice the ability to estimate quantitative material parameters.

I have focused on virtualization of rigid objects, but that is only one category of objects that may be expected to produce sound in virtual environments. My proposed methods will struggle to virtualize deformable objects, thin-shell objects, and objects with heterogeneous material—mugs and bottles are rigid objects, but produce different sounds depending on how much liquid they contain (Wilson et al., 2017). Recent methods propose generalizable wave-based frameworks for simulating a wider variety of physical sounds, though these methods are time-consuming and would need to be significantly accelerated for interactive sound synthesis (Wang et al., 2018).

7.2.3 Multimodal Interaction

Without a complete multimodal object virtualization pipeline, virtualized objects are not ideal recreations of their real-world analogues. Many of the objects virtualized in this dissertation were carefully measured by hand to manually construct a 3D model, and surface textures were often selected from existing datasets as approximate matches. These virtual objects may have had accurate virtualized audio-material parameters from real-world objects, but the properties that had been picked by hand may have caused sensory conflict. Some of my perceptual studies (but not all) had subjects performing multimodal interaction with virtual objects; these studies may have been impacted by this sensory conflict.

Furthermore, interaction may be limited by hardware and software constraints. For hardware, there are two devices for object interaction across this dissertation, both with limitations. First, the PHANToM haptic device provides force feedback but covers a small working area, limiting object size. Second, the HTC Vive greatly expands the working area to the size of a small room, but only simulate haptics with vibrations. For software, many demos were implemented in game engines, which often sacrifice physical accuracy for computation speed, affecting rigid-body collision dynamics and surface appearances. With these and other technical constraints, even an object that has been perfectly virtualized may still not reproduce the same interactions in a virtual environment.

7.3 Future Work

In this section, I discuss five research directions for future work. These are: sound synthesis with nonlinear models, probabilistic modeling for other physical phenomena, learning-based methods for sound synthesis, automatic audio-visual object reconstruction, and extension of my methods to an augmented reality setting.

To improve the realism of synthesized impact sounds, one direction of research would be to use *nonlinear* models for object vibrations. Nonlinear models could more accurately simulate complex modes and nonlinearities during object collisions, leading to richer attack at the start of synthesized impact sounds. One of the primary challenges would be maintaining sufficient runtime performance. A sound synthesis module must provide samples at 44 kHz to be applicable to interactive virtual environments, but nonlinear methods tend to be computationally intensive. Parameter estimation methods would also need to be adapted for nonlinear models, and although they do not have real-time requirements, performance is still important to be practically useful for object virtualization.

Probabilistic models such as the one described in Chapter 4 can improve the robustness of estimation tasks in the presence of error or confounding factors. One application is estimation of object parameters relating to rigid-body dynamics, such as the coefficients of friction and restitution. More applications are estimation of liquid parameters (*e.g.*, viscosity) or deformable object parameters (*e.g.*, stiffness). Many optimization methods for these parameters seek to minimize a least-squares metric (Yang et al., 2016), which implicitly assumes error is normally distributed. If error can be more accurately modeled using a different statistical distribution, then a probabilistic maximum likelihood method may be ideal.

Learning-based methods may provide further improvement through two options. One option is applying learning-based methods to the task of material parameter identification, though it remains to be seen how their results would compare against current methods. The ISNN networks may serve as a starting point if their outputs can be adapted to produce quantitative parameters rather than classifications. The other option is performing sound synthesis directly through a learned generative model. Sound synthesis is primarily still performed through physical simulation, but there have been recent advances in generative neural network design. Work such as "Visual to Sound" (Zhou et al., 2018) and "Visually Indicated Sound" (Owens et al., 2016a) suggest that data-driven approaches may be able to directly synthesize high-quality impact sounds. One current challenge to address is the lack of an ideal dataset for rigid-object impact sounds; the Greatest

Hits dataset (Owens et al., 2016a) covers a breadth of sounds but does not have the depth for a method focused on rigid objects.

Virtualization would ideally be performed through automatic audio-visual *reconstruction* of scenes possibly containing multiple objects. My methods provide a step in this direction, but are limited to producing *classifications* pertaining to object geometry. The first challenge is performing full object reconstruction: producing the complete 3D geometry of a novel object while leveraging audio-visual inputs. Reconstruction from audio alone is an underconstrained problem (Kac, 1966), but using vision as a regularizer may allow complete and accurate reconstructions. The second challenge is reconstructing multiple objects in the same scene—differentiating and segmenting them from one another. However, being able to virtualize an entire scene would greatly extend the applications of these methods.

Finally, there are unrealized applications of this work to augmented reality (AR) settings. Augmented reality requires more understanding of the real world than virtual reality does, so the ability to estimate properties of real-world objects is key. If a real-world object near an AR user can be virtualized on the fly, it opens multiple possibilities. The virtualized object may be duplicated to other locations or visually modified for interior design visualization (Choo and Phan, 2010), and virtual agents could produce realistic interactions with the original object. One challenge in adapting my methods to AR is that there are known to be differences in human perception between the real world, virtual reality, and augmented reality (Jones et al., 2008). The user studies presented in this dissertation all focus on virtual settings, and it remains to be seen how the results would translate to augmented reality. Another challenge is that sensory conflict may be more difficult to avoid, as any virtual objects will be directly contrasted against real-world objects present in the scene.

BIBLIOGRAPHY

Aberman, K., Katzir, O., Zhou, Q., Luo, Z., Sharf, A., Greif, C., Chen, B., and Cohen-Or, D. (2017). Dip transform for 3D shape reconstruction. *ACM Transactions on Graphics (Special Issue of SIGGRAPH)*, 36(4):Article No. 79.

Adhikari, S. (2001). *Damping models for structural vibration*. PhD thesis, University of Cambridge.

Adhikari, S. (2006). Damping modelling using generalized proportional damping. *Journal of Sound and Vibration*, 293(1-2):156–170.

Adhikari, S. and Woodhouse, J. (2001). Identificaion of damping: Part 1, viscous damping. *Journal of Sound and Vibration*, 243(1):43 – 61.

Akenine-Moller, T., Moller, T., and Haines, E. (2002). *Real-Time Rendering*. A. K. Peters, Ltd., Natick, MA, USA, 2nd edition.

Arnab, A., Sapienza, M., Golodetz, S., Valentin, J., Miksik, O., Izadi, S., and Torr, P. H. S. (2015). Joint object-material category segmentation from audio-visual cues. In Xianghua Xie, M. W. J. and Tam, G. K. L., editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 40.1–40.12. BMVA Press.

Avanzini, F. and Rocchesso, D. (2001). Modeling collision sounds: Non-linear contact force. In *In Proc. COST-G6 Conf. Digital Audio Effects (DAFx-01*, pages 61–66.

Aytar, Y., Vondrick, C., and Torralba, A. (2016). SoundNet: Learning sound representations from unlabeled video. In *Advances in Neural Information Processing Systems*, pages 892–900.

Barchiesi, D., Giannoulis, D., Stowell, D., and Plumbley, M. D. (2015). Acoustic scene classification: Classifying environments from the sounds they produce. *IEEE Signal Processing Magazine*, 32(3):16–34.

Basdogan, C., Ho, C.-H., and Srinivasan, M. A. (1997). A ray-based haptic rendering technique for displaying shape and texture of 3D objects in virtual environments. In *Proc. ASME Dynamic Systems and Control Division*, pages 77 – 84.

Bergstra, J. S., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 2546–2554. Curran Associates, Inc.

Bharaj, G., Levin, D. I. W., Tompkin, J., Fei, Y., Pfister, H., Matusik, W., and Zheng, C. (2015). Computational design of metallophone contact sounds. *ACM Trans. Graph.*, 34(6):223:1–223:13.

Bilbao, S., Hamilton, B., Torin, A., Webb, C., Graham, P., Gray, A., Kavoussanakis, K., and Perry, J. (2013). Large scale physical modeling sound synthesis. In *Proceedings of the Stockholm Musical Acoustics Conference/Sound and Music Computing Conference*.

Bilbao, S., Torin, A., and Chatziioannou, V. (2015). Numerical modeling of collisions in musical instruments. *Acta Acustica united with Acustica*, 101(1):155–173.

Blinn, J. F. (1978). Simulation of wrinkled surfaces. *SIGGRAPH Comput. Graph.*, 12(3):286–292.

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 (SIGGRAPH Conference Proceedings)*, 27(3).

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In Lechevallier, Y. and Saporta, G., editors, *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010)*, pages 177–187, Paris, France. Springer.

Brachmann, E., Michel, F., Krull, A., Yang, M. Y., Gumhold, S., and Rother, C. (2016). Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3364–3372.

Büchler, M., Allegro, S., Launer, S., and Dillier, N. (2005). Sound classification in hearing aids inspired by auditory scene analysis. *EURASIP Journal on Advances in Signal Processing*, 2005(18):387845.

Caughey, T. (1960). Classical normal modes in damped linear dynamic systems. *Journal of Applied Mechanics*, 27(2):269–271.

Caughey, T. and O'Kelly, M. (1965). Classical normal modes in damped linear dynamic systems. *Journal of Applied Mechanics*, 32(3):583–588.

Chadwick, J. N., An, S. S., and James, D. L. (2009). Harmonic shells: A practical nonlinear sound model for near-rigid thin shells. In *ACM SIGGRAPH Asia 2009 Papers*, SIGGRAPH Asia '09, pages 119:1–119:10, New York, NY, USA. ACM.

Chadwick, J. N. and James, D. L. (2011). Animating fire with sound. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 84:1–84:8, New York, NY, USA. ACM.

Chadwick, J. N., Zheng, C., and James, D. L. (2012). Precomputed acceleration noise for improved rigid-body sound. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012)*, 31(4).

Chandak, A., Lauterbach, C., Taylor, M., Ren, Z., and Manocha, D. (2008). Ad-frustum: Adaptive frustum tracing for interactive sound propagation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1707–1722.

Choo, S. Y. and Phan, V. T. (2010). Interior design in augmented reality environment. *International Journal of Computer Applications*, 5(5):16–21. Published By Foundation of Computer Science.

Cohen, J., Olano, M., and Manocha, D. (1998). Appearance-preserving simplification. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 115–122, New York, NY, USA. ACM.

Cook, P. R. and Scavone, G. P. (1999). The synthesis toolkit (STK). In *In Proceedings of the International Computer Music Conference*.

Cook, R. L. (1984). Shade trees. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, pages 223–231, New York, NY, USA. ACM.

Coumans, E. (2015). Bullet physics simulation. In *ACM SIGGRAPH 2015 Courses*, SIGGRAPH '15, New York, NY, USA. ACM.

Cowling, M. and Sitte, R. (2003). Comparison of techniques for environmental sound recognition. *Pattern Recognition Letters*, 24(15):2895 – 2907.

Dai, A., Niessner, M., Zollhöfer, M., Izadi, S., and Theobalt, C. (2017). BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph.*, 36(3).

Doel, K. V. D., Kry, P. G., and Pai, D. K. (2001). FoleyAutomatic: Physically-based sound effects for interactive simulation and animation. In *in Computer Graphics (ACM SIGGRAPH 01 Conference Proceedings*, pages 537–544. ACM Press.

E756, A. (2017). Standard test method for measuring vibration-damping properties of materials.

Featherstone, R. (2007). *Rigid Body Dynamics Algorithms*. Springer-Verlag, Berlin, Heidelberg.

Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. (1990). *Computer Graphics: Principles and Practice (2Nd Ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Fouhey, D. F., Gupta, A., and Zisserman, A. (2016). 3D shape attributes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1516–1524.

Fouhey, D. F., Gupta, A., and Zisserman, A. (2019). From images to 3D shape attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):93 – 106.

Fujisaki, W., Goda, N., Motoyoshi, I., Komatsu, H., and Nishida, S. (2014). Audiovisual integration in the human perception of materials. *Journal of Vision*, 14(4):12.

Galoppo, N., Tekin, S., Otaduy, M. A., Gross, M., and Lin, M. C. (2007). Interactive haptic rendering of high-resolution deformable objects. In *Proceedings of the 2Nd International Conference on Virtual Reality*, ICVR'07, pages 215–233, Berlin, Heidelberg. Springer-Verlag.

Gao, Y., Beijbom, O., Zhang, N., and Darrell, T. (2016). Compact bilinear pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 317–326.

Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. (2017). Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA.

Giordano, B. L. and McAdams, S. (2006). Material identification of real impact sounds: Effects of size variation in steel, glass, wood, and plexiglass plates. *The Journal of the Acoustical Society of America*, 119(2):1171–1181.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 580–587, Washington, DC, USA. IEEE Computer Society.

Goesele, M., Snavely, N., Curless, B., Hoppe, H., and Seitz, S. M. (2007). Multi-view stereo for community photo collections. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8.

Golodetz*, S., Sapienza*, M., Valentin, J. P. C., Vineet, V., Cheng, M.-M., Arnab, A., Prisacariu, V. A., Kähler, O., Ren, C. Y., Murray, D. W., Izadi, S., and Torr, P. H. S. (2015). SemanticPaint: A Framework for the Interactive Segmentation of 3D Scenes. Technical Report TVG-2015-1, Department of Engineering Science, University of Oxford. Released as arXiv e-print 1510.03727.

Grassi, M. (2005). Do we hear size or sound? Balls dropped on plates. *Perception & Psychophysics*, 67(2):274–284.

Grushka, E. (1972). Characterization of exponentially modified gaussian peaks in chromatography. *Analytical Chemistry*, 44(11):1733–1738. PMID: 22324584.

Hampel, S., Langer, S., and Cisilino, A. P. (2008). Coupling boundary elements to a raytracing procedure. *International Journal for Numerical Methods in Engineering*, 73(3):427–445.

Hayward, V. (2008). A brief taxonomy of tactile illusions and demonstrations that can be done in a hardware store. *Brain Research Bulletin*, 75(6):742 – 752. Special Issue: Robotics and Neuroscience.

Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R. J., and Wilson, K. (2017). CNN architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135.

Ho, C.-H., Basdogan, C., and Srinivasan, M. A. (1999). Efficient point-based rendering techniques for haptic display of virtual objects. *Presence: Teleoper. Virtual Environ.*, 8(5):477–491.

Huzaifah, M. (2017). Comparison of time-frequency representations for environmental sound classification using convolutional neural networks. *CoRR*, abs/1706.07156.

Imregun, M. and Ewins, D. J. (1995). Complex Modes - Origins and Limits. In *Proceedings of the 13th International Modal Analysis Conference*, volume 2460, page 496.

James, D. L., Barbič, J., and Pai, D. K. (2006). Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 987–995. ACM.

Jones, A., Swan, J. E., Singh, G., and Kolstad, E. (2008). The effects of virtual reality, augmented reality, and motion parallax on egocentric depth perception. In *2008 IEEE Virtual Reality Conference*, pages 267–268.

Kac, M. (1966). Can one hear the shape of a drum? *The American Mathematical Monthly*, 73(4):1–23.

Kaneko, T., Takahei, T., Inami, M., Kawakami, N., Yanagida, Y., Maeda, T., and Tachi, S. (2001). Detailed shape representation with parallax mapping. In *In Proceedings of the ICAT*, pages 205–208.

Kanezaki, A., Matsushita, Y., and Nishida, Y. (2016). RotationNet: Learning object classification using unsupervised viewpoint estimation. *CoRR*, abs/1603.06208.

Karplus, K. and Strong, A. (1983). Digital synthesis of plucked-string and drum timbres. *Computer Music Journal*, 7(2):43–55.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Klatzky, R. L., Lederman, S. J., and Metzger, V. A. (1985). Identifying objects by touch: An "expert system". *Perception & Psychophysics*, 37(4):299–302.

Klatzky, R. L., Pai, D. K., and Krotkov, E. P. (2000). Perception of material from contact sounds. *Presence: Teleoperators and Virtual Environments*, 9(4):399–410.

Lai, K., Bo, L., Ren, X., and Fox, D. (2011). A large-scale hierarchical multi-view RGB-D object dataset. In *2011 IEEE International Conference on Robotics and Automation*, pages 1817–1824.

Langlois, T. R., An, S. S., Jin, K. K., and James, D. L. (2014). Eigenmode compression for modal sound models. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014)*, 33(4).

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

Lederman, S. J. and Taylor, M. M. (1972). Fingertip force, surface geometry, and the perception of roughness by active touch. *Perception & Psychophysics*, 12(5):401–408.

Lee, K. M. (2006). Presence, Explicated. *Communication Theory*, 14(1):27–50.

Li, D., Fei, Y., and Zheng, C. (2015). Interactive acoustic transfer approximation for modal sound. *ACM Trans. Graph.*, 35(1).

Lin, T., RoyChowdhury, A., and Maji, S. (2015). Bilinear CNN models for fine-grained visual recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1449–1457.

Lloyd, D. B., Raghuvanshi, N., and Govindaraju, N. K. (2011). Sound synthesis for impact sounds in video games. In *Symposium on Interactive 3D Graphics and Games*, I3D '11, pages 55–62, New York, NY, USA. ACM.

Lombard, M. and Jones, M. T. (2015). *Defining Presence*, pages 13–34. Springer International Publishing, Cham.

Loomis, J. M. and Lederman, S. J. (1986). Tactual perception. *Handbook of perception and human performances*, 2:2.

Lysenkov, I., Eruhimov, V., and Bradski, G. (2013). Recognition and pose estimation of rigid transparent objects with a kinect sensor. In *Robotics: Science and Systems Conference (RSS)*, volume 273.

Lysenkov, I. and Rabaud, V. (2013). Pose estimation of rigid transparent objects in transparent clutter. In *2013 IEEE International Conference on Robotics and Automation*, pages 162–169.

Massie, T. H. and Salisbury, J. K. (1994). The PHANToM haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, volume 55, pages 295–300. Chicago, IL.

Maturana, D. and Scherer, S. (2015). VoxNet: A 3D convolutional neural network for real-time object recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, page 922 – 928.

McAdams, S., Roussarie, V., Chaigne, A., and Giordano, B. L. (2010). The psychomechanics of simulated sound sources: Material properties of impacted thin plates. *The Journal of the Acoustical Society of America*, 128(3):1401–1413.

McDermott, J. H., Schemitsch, M., and Simoncelli, E. P. (2013). Summary statistics in auditory perception. *Nature neuroscience*, 16(4):493.

Mehra, R., Antani, L., Kim, S., and Manocha, D. (2014). Source and listener directivity for interactive wave-based sound propagation. *IEEE Transactions on Visualization and Computer Graphics*, 20(4):495–503.

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

Mehra, R., Rungta, A., Golas, A., Lin, M., and Manocha, D. (2015). Wave: Interactive wave-based sound propagation for virtual environments. *IEEE Transactions on Visualization and Computer Graphics*, 21(4):434–442.

Michon, R., Martin, S. R., and Smith, J. O. (2017). Mesh2faust: a modal physical model generator for the faust programming language – application to bell modeling. In *Proceedings of the International Computer Music Conference (ICMC-17)*, Shanghai, China.

Michon, R. and Smith, J. O. (2011). Faust-STK: a set of linear and nonlinear physical models for the faust programming language. In *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, pages 19–23, Paris, France.

Minsky, M., Ming, O.-y., Steele, O., Brooks, Jr., F. P., and Behensky, M. (1990). Feeling and seeing: Issues in force display. *SIGGRAPH Comput. Graph.*, 24(2):235–241.

Minsky, M. D. R. R. (1995). *Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force-feedback Display*. PhD thesis, Cambridge, MA, USA. Not available from Univ. Microfilms Int.

Morrison, J. D. and Adrien, J.-M. (1993). MOSAIC: A framework for modal synthesis. *Computer Music Journal*, 17(1):45–56.

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

Nakagawa, S. and Cuthill, I. C. (2007). Effect size, confidence interval and statistical significance: a practical guide for biologists. *Biological Reviews*, 82(4):591–605.

Nashif, A. D., Jones, D. I., and Henderson, J. P. (1985). *Vibration damping*. John Wiley & Sons.

Newcombe, R. A., Fox, D., and Seitz, S. M. (2015). DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 343–352.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011). KinectFusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136.

Nykl, S., Mourning, C., and Chelberg, D. (2013). Interactive mesostructures. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '13, pages 37–44, New York, NY, USA. ACM.

O'Brien, J. F., Shen, C., and Gatchalian, C. M. (2002). Synthesizing sounds from rigid-body simulations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '02, pages 175–181, New York, NY, USA. ACM.

Oliveira, M. M., Bishop, G., and McAllister, D. (2000). Relief texture mapping. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 359–368, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

Otaduy, M., Jain, N., Sud, A., and Lin, M. (2004). Haptic display of interaction between textured models. In *IEEE Visualization Conference*, pages 297–304.

Otaduy, M. A. and Lin, M. C. (2003a). CLODs: Dual hierarchies for multiresolution collision detection. *Eurographics Symposium on Geometry Processing*, pages 94–101.

Otaduy, M. A. and Lin, M. C. (2003b). Sensation preserving simplification for haptic rendering. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*, pages 543–553.

Otaduy, M. A. and Lin, M. C. (2005). Sensation preserving simplification for haptic rendering. In *ACM SIGGRAPH 2005 Courses*, SIGGRAPH '05, New York, NY, USA. ACM.

Owens, A., Isola, P., McDermott, J., Torralba, A., Adelson, E. H., and Freeman, W. T. (2016a). Visually indicated sounds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2405–2413.

Owens, A., Wu, J., McDermott, J. H., Freeman, W. T., and Torralba, A. (2016b). *Ambient Sound Provides Supervision for Visual Learning*, pages 801–816. Springer International Publishing, Cham.

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*, pages 87–96. ACM.

Park, E., Han, X., Berg, T. L., and Berg, A. C. (2016). Combining multiple sources of knowledge in deep CNNs for action recognition. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–8. IEEE.

Piczak, K. J. (2015a). Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6.

Piczak, K. J. (2015b). ESC: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pages 1015–1018, New York, NY, USA. ACM.

Policarpo, F., Oliveira, M. M., and Comba, J. a. L. D. (2005). Real-time relief mapping on arbitrary polygonal surfaces. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games*, I3D '05, pages 155–162, New York, NY, USA. ACM.

Raghuvanshi, N., Allen, A., and Snyder, J. (2016). Numerical wave simulation for interactive audio-visual applications. *The Journal of the Acoustical Society of America*, 139(4):2008–2009.

Raghuvanshi, N. and Lin, M. C. (2006). Interactive sound synthesis for large scale environments. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, I3D '06, pages 101–108, New York, NY, USA. ACM.

Raghuvanshi, N., Narain, R., and Lin, M. C. (2009). Efficient and accurate sound propagation using adaptive rectangular decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):789–801.

Raghuvanshi, N., Snyder, J., Mehra, R., Lin, M., and Govindaraju, N. (2010). Precomputed wave simulation for real-time sound propagation of dynamic sources in complex scenes. *ACM Trans. Graph.*, 29(4):68:1–68:11.

Rayleigh, J. W. S. B. (1896). *The Theory of Sound*, volume 2. Macmillan.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc.

Ren, Z., Mehra, R., Coposky, J., and Lin, M. (2012). Designing virtual instruments with touch-enabled interface. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, pages 433–436. ACM.

Ren, Z., Yeh, H., Klatzky, R., and Lin, M. C. (2013a). Auditory perception of geometry-invariant material properties. *Visualization and Computer Graphics, IEEE Transactions on*, 19(4):557–566.

Ren, Z., Yeh, H., and Lin, M. (2010). Synthesizing contact sounds between textured models. In *Virtual Reality Conference (VR), 2010 IEEE*, pages 139–146.

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

Rungta, A., Schissler, C., Mehra, R., Malloy, C., Lin, M., and Manocha, D. (2016). Syncopation: Interactive synthesis-coupled sound propagation. *IEEE Transactions on Visualization and Computer Graphics*, 22(4):1346–1355.

Rungta, A., Schissler, C., Rewkowski, N., Mehra, R., and Manocha, D. (2018). Diffraction kernels for interactive sound propagation in dynamic environments. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1613–1622.

Salamon, J. and Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3):279–283.

Salamon, J., Jacoby, C., and Bello, J. P. (2014). A dataset and taxonomy for urban sound research. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 1041–1044, New York, NY, USA. ACM.

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

Scavone, G. P. and Cook, P. R. (2005). RTMidi, RTAudio, and a synthesis toolkit (STK) update. In *In Proceedings of the International Computer Music Conference*.

Schissler, C. and Manocha, D. (2011). GSound: Interactive sound propagation for games. In *Audio Engineering Society Conference: 41st International Conference: Audio for Games*.

Schissler, C. and Manocha, D. (2016). Adaptive impulse response modeling for interactive sound propagation. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '16, pages 71–78, New York, NY, USA. ACM.

Schreck, C., Rohmer, D., James, D. L., Hahmann, S., and Cani, M.-P. (2016). Real-time sound synthesis for paper material based on geometric analysis. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '16, pages 211–220, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 519–528.

Serra, X. and Smith, J. (1990). Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Computer Music Journal*, 14(4):12–24.

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012). Indoor segmentation and support inference from RGBD images. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, ECCV'12, pages 746–760, Berlin, Heidelberg. Springer-Verlag.

Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 568–576. Curran Associates, Inc.

Singh, A., Sha, J., Narayan, K. S., Achim, T., and Abbeel, P. (2014). BigBIRD: A large-scale 3D database of object instances. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 509–516. IEEE.

Slater, J. C., Keith Belvin, W., and Inman, D. J. (1993). A survey of modern methods for modeling frequency dependent damping in finite element models. In *PROCEEDINGS-SPIE THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING*, pages 1508–1508. SPIE INTERNATIONAL SOCIETY FOR OPTICAL.

Slater, M., Antley, A., Davison, A., Swapp, D., Guger, C., Barker, C., Pistrang, N., and Sanchez-Vives, M. V. (2006a). A virtual reprise of the stanley milgram obedience experiments. *PLOS ONE*, 1(1):1–10.

Slater, M., Khanna, P., Mortensen, J., and Yu, I. (2009). Visual realism enhances realistic response in an immersive virtual environment. *IEEE Computer Graphics and Applications*, 29(3):76–84.

Slater, M., Pertaub, D.-P., Barker, C., and Clark, D. M. (2006b). An experimental study on fear of public speaking using a virtual environment. *CyberPsychology & Behavior*, 9(5):627–633. PMID: 17034333.

Smith, J. O. (1992). Physical modeling using digital waveguides. *Computer Music Journal*, 16(4):74–91.

Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: Exploring photo collections in 3D. *ACM Trans. Graph.*, 25(3):835–846.

Socher, R., Huval, B., Bhat, B., Manning, C. D., and Ng, A. Y. (2012). Convolutional-recursive deep learning for 3D object classification. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 656–664, USA. Curran Associates Inc.

Song, S., Yu, F., Zeng, A., Chang, A. X., Savva, M., and Funkhouser, T. (2017). Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*.

Southern, A., Siltanen, S., and Savioja, L. (2011). Spatial room impulse responses with a hybrid modeling method. In *Audio Engineering Society Convention 130*.

Sterling, A. and Lin, M. C. (2016a). Integrated multimodal interaction using texture representations. *Computers & Graphics*, 55:118 – 129.

Sterling, A. and Lin, M. C. (2016b). Interactive modal sound synthesis using generalized proportional damping. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '16, pages 79–86, New York, NY, USA. ACM.

Sterling, A., Rewkowski, N., Klatzky, R. L., and Lin, M. C. (2019). Audio-material reconstruction for virtualized reality using a probabilistic damping model. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):1855–1864.

Sterling, A., Wilson, J., Lowe, S., and Lin, M. C. (2018). ISNN: Impact sound neural network for audio-visual object classification. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 578–595, Cham. Springer International Publishing.

Svensson, U. P., Fred, R. I., and Vanderkooy, J. (1999). An analytic secondary source model of edge diffraction impulse responses. *The Journal of the Acoustical Society of America*, 106(5):2331–2344.

Szabo, T. L. (1994). Time domain wave equations for lossy media obeying a frequency power law. *The Journal of the Acoustical Society of America*, 96(1):491–500.

Szirmay-Kalos, L. and Umenhoffer, T. (2008). Displacement mapping on the gpu — state of the art. *Computer Graphics Forum*, 27(6):1567–1592.

Tanaka, K., Mukaigawa, Y., Funatomi, T., Kubo, H., Matsushita, Y., and Yagi, Y. (2017). Material Classification using Frequency- and Depth-dependent Time-of-Flight Distortion. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 79–88.

Taylor II, R. M., Hudson, T. C., Seeger, A., Weber, H., Juliano, J., and Helser, A. T. (2001). VRPN: a device-independent, network-transparent VR peripheral system. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 55–61. ACM.

Tenenbaum, J. B. and Freeman, W. T. (2000). Separating style and content with bilinear models. *Neural Comput.*, 12(6):1247–1283.

Tevs, A., Ihrke, I., and Seidel, H.-P. (2008). Maximum mipmaps for fast, accurate, and scalable dynamic height field rendering. In *Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games*, I3D '08, pages 183–190, New York, NY, USA. ACM.

Theoktisto, V., González, M. F., and Navazo, I. (2010). Hybrid rugosity mesostructures (HRMs) for fast and accurate rendering of fine haptic detail. *CLEI Electron. J.*, pages –1–1.

Thiemann, J., Ito, N., and Vincent, E. (2013). The diverse environments multi-channel acoustic noise database (DEMAND): A database of multichannel environmental noise recordings. *Proceedings of Meetings on Acoustics*, 19(1):035081.

Traer, J. and McDermott, J. H. (2016). Statistics of natural reverberation enable perceptual separation of sound and space. *Proceedings of the National Academy of Sciences*, 113(48):E7856–E7865.

Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (2000). Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, pages 298–372, London, UK, UK. Springer-Verlag.

Tsingos, N., Funkhouser, T., Ngan, A., and Carlbom, I. (2001). Modeling acoustics in virtual environments using the uniform theory of diffraction. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 545–552, New York, NY, USA. ACM.

Valentin, J., Vineet, V., Cheng, M.-M., Kim, D., Shotton, J., Kohli, P., Niessner, M., Criminisi, A., Izadi, S., and Torr, P. H. S. (2015). SemanticPaint: Interactive 3D Labeling and Learning at your Fingertips. *ACM Transactions on Graphics*, 34(5).

van den Doel, K. and Pai, D. K. (1996). The sounds of physical shapes. *Presence*, 7:382–395.

van Walstijn, M. and Mehes, S. (2017). An explorative string-bridge-plate model with tunable parameters. In *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17)*, Edinburgh, UK.

Wang, J.-H., Qu, A., Langlois, T. R., and James, D. L. (2018). Toward wave-based sound synthesis for computer animation. *ACM Trans. Graph.*, 37(4):109:1–109:16.

Webb, C. J. (2014). *Parallel computation techniques for virtual acoustics and physical modelling synthesis*. PhD thesis, The University of Edinburgh.

Westoby, M., Brasington, J., Glasser, N., Hambrey, M., and Reynolds, J. (2012). 'Structure-from-Motion' photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179:300 – 314.

Wilson, J., Sterling, A., Rewkowski, N., and Lin, M. C. (2017). Glass half full: sound synthesis for fluid–structure coupling using added mass operator. *The Visual Computer*, 33(6):1039–1048.

Woodhouse, J. (1998). Linear damping models for structural vibration. *Journal of Sound and Vibration*, 215(3):547–569.

Wu, Z., Song, S., Khosla, A., and Xiao, J. (2015). 3D ShapeNets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920.

Yamamoto, K. and Igarashi, T. (2016). Interactive physically-based sound design of 3D model using material optimization. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '16, pages 231–240, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Yang, S., Jojic, V., Lian, J., Chen, R., Zhu, H., and Lin, M. C. (2016). Classification of prostate cancer grades and t-stages based on tissue elasticity using medical image analysis. In Ourselin, S., Joskowicz, L., Sabuncu, M. R., Unal, G., and Wells, W., editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 627–635, Cham. Springer International Publishing.

Yeh, H., Mehra, R., Ren, Z., Antani, L., Manocha, D., and Lin, M. (2013). Wave-ray coupling for interactive sound propagation in large complex scenes. *ACM Trans. Graph.*, 32(6):165:1–165:11.

Yoon, S., Salomon, B., Lin, M. C., and Manocha, D. (2004). Fast collision detection between massive models using dynamic simplification. *Eurographics Symposium on Geometry Processing*, pages 136–146.

Yu, Z., Yu, J., Fan, J., and Tao, D. (2017). Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. *IEEE International Conference on Computer Vision (ICCV)*, pages 1839–1848.

Zhang, R., Tsai, P.-S., Cryer, J. E., and Shah, M. (1999). Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706.

Zhang, Z., Li, Q., Huang, Z., Wu, J., Tenenbaum, J., and Freeman, B. (2017a). Shape and material from sound. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 1278–1288. Curran Associates, Inc.

Zhang, Z., Wu, J., Li, Q., Huang, Z., Traer, J., McDermott, J. H., Tenenbaum, J. B., and Freeman, W. T. (2017b). Generative modeling of audible shapes for object perception. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1260–1269.

Zheng, C. and James, D. L. (2011). Toward high-quality modal contact sound. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011)*, 30(4).

Zhou, Y., Wang, Z., Fang, C., Bui, T., and Berg, T. L. (2018). Visual to sound: Generating natural sound for videos in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zimmons, P. and Panter, A. (2003). The influence of rendering quality on presence and task performance in a virtual environment. In *IEEE Virtual Reality, 2003. Proceedings.*, pages 293–294.