REAL-TIME PHYSICALLY BASED SOUND SYNTEHSIS AND APPLICATION IN MULTIMODAL INTERACTION

Zhimin Ren

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
2014

Approved by:

Ming C. Lin

Dinesh Manocha

Gary Bishop

Roberta Klatzky

Nikunj Raghuvanshi

ABSTRACT

Zhimin Ren: Real-Time Physically Based Sound Synthesis and Application in Multimodal Interaction
(Under the direction of Ming C. Lin)

An immersive experience in virtual environments requires realistic auditory feedback that is closely coupled with other modalities, such as vision and touch. This is particularly challenging for real-time applications due to its stringent computational requirement. In this dissertation, I present and evaluate effective real-time physically based sound synthesis models that integrate visual and touch data and apply them to create richly varying multimodal interaction. I first propose an efficient contact sound synthesis technique that accounts for texture information used for visual rendering and greatly reinforces cross-modal perception. Secondly, I present both empirical and psychoacoustic approaches that formally study the geometry-invariant property of the commonly used material model in real-time sound synthesis. Based on this property, I design a novel example-based material parameter estimation framework that automatically creates synthetic sound effects naturally controlled by complex geometry and dynamics in visual simulation. Lastly, I translate user touch input captured on commodity multi-touch devices to physical performance models that drive both visual and auditory rendering. This novel multimodal interaction is demonstrated in a virtual musical instrument application on both a large-size tabletop and mobile tablet devices, and evaluated through pilot studies. Such an application offers capabilities for intuitive and expressive music playing, rapid prototyping of virtual instruments, and active exploration of sound effects determined by various physical parameters.

To my dearest,

Dad, Mom, and Feng

ACKNOWLEDGEMENTS

During my PhD education, I have received tremendous help and support from a large number of people, who guided me through challenging times and accompanied me along my fun and exciting PhD journey. To all of them, I wish to extend my sincerest gratitude!

I thank Prof. Ming C. Lin for being my advisor and constantly offering guidance to me throughout my doctoral studies. I am sincerely grateful for her patience when I was taking my time, her encouragement when I doubted myself, and her advice whenever I needed it. Her hard work and zeal for research always inspire me and push me to do better. Her genuine care for me and interest in my growth constantly remind me how lucky I am to have her as my advisor. It has truely been a great honor of mine to have worked so closely with Prof. Lin, and I owe a great deal of gratitude for all the support, help, and guidance she has offered me so tirelessly.

I am grateful to Prof. Roberta Klatzky for a wonderful collaboration opportunity on one of my papers. I thank her for broadening my research interest and unselfishly spending a lot of time on helping me with my doctoral studies. I appreciate her traveling from Pittsburgh to attend my defense and provide the extremely helpful and thoughtful feedback to my defense and dissertation. I thank Prof. Dinesh Manocha for his early support when I first started graduate school and introducing me to the field of sound simulation. His mentorship throughout my entire doctoral studies in the GAMMA research group has helped me significantly. I thank Dr. Nikunj Raghuvanshi. His research inspired me to pursue the area of physically-based sound synthesis. He is the top expert in sound simulation research and is always willing to help me get over technical difficulties. He often joins our discussions remotely, answers my technical questions with great expertise and insights, and urges me to do solid research. I also thank Prof. Gary Bishop for the invaluable feedback and support he has provided me as my committee member. His encouragement has helped me stay optimistic and enthusiastic about my research.

In addition, I hope to thank my peers in the GAMMA research group, who provide the best feedbacks on my research and many great friendships. I am extremely grateful to have had the

chance to work with Hengching Yeh, Anish Chandak, Micah Taylor, Ravish Mehra, Lakulish Antani, Christian Lauterbach, Jason Coposky, James Norton, and Maggie Zhou. Thanks to them for teaching me so much and being great friends of mine.

I thank the staff in the Computer Science Department at UNC. I especially thank Janet Jones and Jodie Turbull for helping me cope with so many logistic details related to getting my degree.

I owe my every little accomplishment to my parents. Their love and support motivate me to do my best. I thank them for always believing in me and supporting me in anything my heart desires. They are my heaven, my harbor, my protecting shield, and always the home that I return to when I need strength and love.

Last but not least, I thank my dearest husband, Feng. His love and support immeasurably help me grow and go through more difficult times. He is the pillar of my life, and I am forever grateful for having him.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

In our everyday life, we are constantly engaged in interactions that involve different senses, e.g. sight, hearing, and touch. In order to create an immersive experience in virtual environment (VE) applications, generating synchronous multiple sensory feedback is essential. In particular, *auditory* feedback plays a vital part. In our real-world experiences, humans are constantly submerged in a large variety of audible sounds. When we type on a keyboard, we expect varying clicking sounds based on how fast or hard we strike the keys. When we walk in the streets, we expect our shoes to rub against different materials on the ground and make richly detailed sounds depending on this complex interaction. When we throw a bowling ball, we expect a series of loud impact sounds coming in synchrony with the dynamic collision among the ball, the pins, and the surrounding environment. My thesis focuses on studying physically based techniques that automatically synthesizes richly varying *auditory* feedback in *realtime* and utilizing detailed user interaction information to drive this sound synthesis process. I aim to design and evaluate dynamic *multimodal* applications that generate sounds corresponding closely with users' visual and touch sensory.

Recently, visual simulation and rendering have been immensely developed and studied, in comparison, auditory simulation has been largely overlooked. The computer-generated imagery in modern movies, video games, simulators, and other VE applications are often simulated with physically based methods and then rendered at photo-realistic quality with advanced ray tracing or rasterization techniques. Meanwhile, the accompanying audio component is usually *manually* recorded, edited, and then synchronized with the visual by foley artists and sound designers. Unfortunately, this is a labor-intensive practice. More importantly, it cannot be applied to all interactive applications, in which it is still challenging, if not infeasible, to produce sound effects that precisely capture complex interactions that cannot be predicted in advance. On the other hand, *physically based sound synthesis* is capable of reflecting the variations and diverse configurations at run-time. With such methods, geometry and interaction dependent and highly dynamic sound effects can be automatically generated. While all VE applications can benefit from such auditory simulation,

real-time applications in particular demand responsive and richly-varying auditory feedback that closely corresponds to information in other senses. The first part of my thesis presents techniques that advance real-time physically based sound synthesis for rigid bodies and proposes a novel framework that facilitates this process.

The sense of touch is also ubiquitous in our daily-life. Haptic research has long been studying generating feedbacks in the sense of touch through various novel materials and devices. However, very few haptic devices have gained mainstream popularity. With *multi-touch* displays becoming prevalent instead, much richer real-time input and control data from users' touch interaction are readily available for VE applications compared with the traditional cursor or joystick based interfaces. My thesis examines how users' active touch and contact that are captured by consumer multi-touch devices can be translated into physical performance models that control the sound simulation. Applications for virtual musical instruments are studied. Such an interactive, multimodal system would offer capabilities for expressive music playing, rapid prototyping of virtual instruments, and active exploration of sound effects determined by various physical parameters. Moreover, I demonstrate that through effective algorithms this is feasible on mobile hardware, where computing resources is limited.

1.1 Challenges

With state-of-the-art real-time sound synthesis techniques and current multi-touch enabled devices, I have identified the key challenges for real-time physically based sound synthesis and its adoption for multimodal interaction.

Real-Time Performance: Human's audible frequency range is between 20Hz and $22,000$Hz, so the audio refresh rate has to be as high as $44,000$Hz. This directly translates into extremely small time steps for real-time sound simulation. Within such a short time step, faithfully solving for the complex dynamics that drives the sound synthesis is infeasible. Moreover, on mobile multi-touch devices, where computing resources are scarcer, with touch event tracking and visual renderings, very limited resources are left for user interaction modeling and sound simulation.

Material Parameter Model in Sound Synthesis: Due to the real-time performance requirement, the widely adopted physically based sound synthesis techniques approximate real-world physical

materials with a much simpler model. Firstly, no previous work has studied if this simple model is sufficient to be considered geometry-invariant so that it is usable across virtual objects of different shapes and sizes. Without this property, it is infeasible to adopt this material model for complex applications. Moreover, it is painstakingly difficult to obtain high-quality material parameters for this model. While pre-processing takes minutes to hours depending on the complexity of the geometry, so far it has been challenging and laborious to explore and identify material parameters users desire.

Limitations on Multi-Touch Devices: With visual, auditory, and touch interactions happening at the same time, one of the biggest challenges is how to diminish any perceptible latency among these components. Moreover, multi-touch devices usually only have sensors for tracking interaction on the touch surface, so accurately modeling user interaction dynamics in 3D space is a hard problem to solve. On mobile devices like tablets and phones, the above challenges coupled with limited computing power make it even more difficult to create a highly responsive and integrated multimodal experience.

1.2 Previous Work

1.2.1 Sound Synthesis

In the last couple of decades, there has been strong interest in digital sound synthesis in both computer music and computer graphics communities due to the needs for auditory display in virtual environment applications. The traditional practice of Foley sounds is still widely adopted by sound designers for applications like video games and movies. Real sound effects are recorded and edited to match a visual display. More recently, *granular synthesis* became a popular technique to create sounds with computers or other digital synthesizers. Short grains of sounds are manipulated to form a sequence of audio signals that sound like a particular object or event. Roads (2004) gave an excellent review on the theories and implementation of generating sounds with this approach. Picard et al. (2009) proposed techniques to mix sound grains according to events in a physics engine.

Another approach for simulating sound sources is using physically based simulation to synthesize realistic sounds that automatically synchronize with the visual rendering. Generating sounds of interesting natural phenomena like fluid dynamics and aerodynamics have been proposed (Dobashi et al., 2003, 2004; Zheng and James, 2009; Moss et al., 2010; Chadwick and James, 2011). The

ubiquitous rigid-body sounds play a vital role in all types of virtual environments, and these sounds are what I focus on in this chapter. O'Brien et al. (2001) proposed simulating rigid bodies with deformable body models that approximates solid objects' small-scale vibration leading to variation in air pressure, which propagates sounds to human ears. Their approach accurately captures surface vibration and wave propagation once sounds are emitted from objects. However, it is far from being efficient enough to handle interactive applications. Adrien (1991) introduced *modal synthesis* to digital sound generation. For real-time applications, *linear modal sound synthesis* has been widely adopted to synthesize rigid-body sounds (van den Doel and Pai, 1998a; O'Brien et al., 2002b; Raghuvanshi and Lin, 2006b; James et al., 2006a; Zheng and James, 2010a; Ren et al., 2012a). However, despite its extensive adoption, the Rayleigh damping model has never been formally evaluated for its transferability across varying shapes and sizes. In other words, it has not been formally studied and validated that the same set of Rayleigh damping coefficients along with the intrinsic material parameters, i.e. density and elasticity, preserve the same sense of material perception, if they are applied to objects made of the same materials but different shapes and sizes. This method acquires a modal model (i.e. a bank of damped sinusoidal waves) using *modal analysis* and generates sounds at runtime based on excitation to this modal model. Moreover, sounds of complex interaction can be achieved with modal synthesis. van den Doel et al. (2001a) presented parametric models to approximate contact forces as excitation to modal models to generate impact, sliding, and rolling sounds. More recently, Zheng and James (2011) created highly realistic contact sounds with linear modal synthesis by enabling non-rigid sound phenomena and modeling vibrational contact damping. Moreover, the standard modal synthesis can be accelerated with techniques proposed by (Raghuvanshi and Lin, 2006b; Bonneel et al., 2008a), which make synthesizing a large number of sounding objects feasible at interactive rates.

The use of linear modal synthesis is not limited to creating simple rigid-body sounds. Chadwick et al. (2009) used modal analysis to compute linear mode basis, and added nonlinear coupling of those modes to efficiently approximate the rich thin-shell sounds. Zheng and James (2010a) extended linear modal synthesis to handle complex fracture phenomena by precomputing modal models for ellipsoidal sound proxies. However, few previous sound synthesis work addressed the issue of how to determine material parameters used in modal analysis to more easily recreate realistic sounds.

1.2.2 Multimodal Applications

Multi-touch hardware and software have been actively studied and innovated for many years. As early as 1985, Buxton et al. (1985) analysed touch-input devices and compared them with conventional mice and joysticks. Nowadays, multi-touch technologies are mainly categorized into three types of devices: *capacitive sensing*, *resistive sensing*, and *optical sensing*. Han (2005) invented a low-cost optical multi-touch sensing technology that made fast multi-touch on a large surface more practical. Rosenberg and Perlin (2009) designed an inexpensive and lightweight multi-touch input pad that provides pressure-sensing, and this device can be attached to assorted displays for a direct touch experience. With the prevalence of capacitive sensing devices like the iPhone and iPad, average digital device users have become familiar, comfortable, and even used to interacting with multi-touch. Such expressive interfaces encourage much more intuitive and natural interactions from users and also offer applications additional dimensions for input information.

One prominent format of such interface is the multi-touch tabletop, which has low cost and the capability for multiple-user collaboration. It is a great candidate for building multi-modal interactive systems. Researchers have employed such technology for creating music and sounds in general. Davidson and Han (2006) employed their multi-touch tabletop to control widgets that modify sound synthesis. Kaltenbrunner et al. (2006) designed a tangible multi-touch interface that allows both local and remote collaboration on synthesizing audio. Hochenbaum and Vallis (2009) built a multi-touch table and applied it to generating parametric sounds and remotely controlling real drums. However, none of these works attempts to virtually simulate musical instruments. Various techniques for finger tracking are surveyed by Schöening et al. (2009). These methods facilitate higher-level touch interpretation and gesture recognition. Nevertheless, none of those techniques handles percussive interactions, in which case striking velocity is required to be estimated.

Moreover, it is more challenging to implement a real-time and richly responsive multimodal experience on mobile devices, which have become ubiqutous. Given the limited computing resources, efficient algorithms that utilizes the mobile multi-touch hardware are required.

1.3 Thesis Statement

Through studying the geometry-invariant property, a novel example-guided framework makes real-time physically based sound synthesis feasible and easy to adopt for virtual environment applications. Combined with an efficient contact sound synthesis model and expressive multi-touch handling, the richly varying synthetic sound effects provide a responsive and immersive multimodal interaction that closely couples visual rendering, auditory simulation, and touch input.

1.4 Main Results

The goal of my work is to develop techniques that advance *real-time* physically based sound synthesis, make it feasible and easy-to-adopt for VE applications, and finally apply such sound simulation approaches to creating richly varying multimodal interaction:

1.4.1 Evaluation of the real-time sound synthesis material model:

Through an empirical analysis and a psychoacoustic study, I evaluate and conclude that the widely adopted real-time sound synthesis model, which assumes Rayleigh damping, can be largely considered geometry-invariant. As a result, the same set of material parameters can be applied to objects of various geometry while the same sense of material is generally preserved.

1.4.2 Example-guided framework for material parameter estimation:

A novel framework that automatically identifies material parameters for sound synthesis based on one audio recording example is presented. With this framework, adopting real-time physically based sound synthesis for various VE applications that involve a large number of objects is now feasible.

1.4.3 Efficient contact sound computation:

Through a three-level surface representation that takes advantages of textures, I am able to synthesize complex contact sounds in real-time that closely correspond to the visual renderings in VE applications.

1.4.4 Real-time sound synthesis driven by multi-touch:

Rich and detailed modeling of user interaction based on the tracked multi-touch input is presented. With this information, physically based sound synthesis closely respond to users' rich and dynamic input. A multimodal interaction coupling visual rendering, auditory simulation, and multi-touch input is presented on both large-size tabletop and mobile tablet devices.

1.5 Organization

The rest of this dissertation is organized as follows:

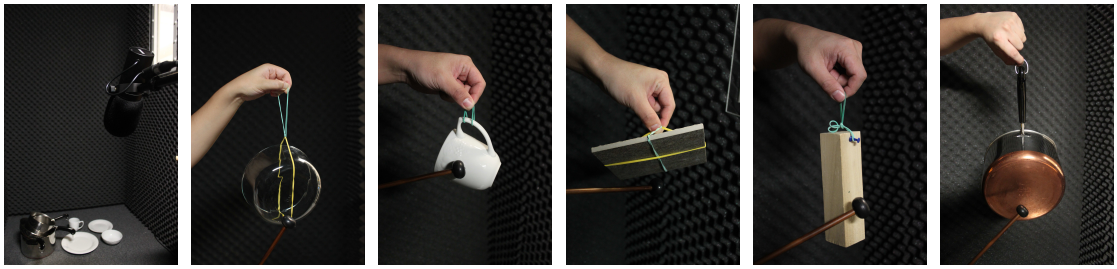
Chapter 2 describes the widely-adopted modal sound synthesis model and presents both an empirical and a psychoacoustic study on the material model assumed in modal sound synthesis. Based on findings in this chapter, I propose a novel example-guided material estimation framework for modal sound synthesis in Chapter 3. Chapter 4 introduces an efficient contact sound synthesis model that allows modal sound synthesis model to handle complex interactions like continuous sliding in real-time. Chapter 5 shows virtual musical instrument systems using modal sound synthesis model on multitouch devices, i.e. a tabletop and a tablet. Such systems provide users with multimodal interaction that couples visual rendering, auditory feedbacks, and touch input. Finally, Chapter 6 presents a similarly responsive and multimodal experience on consumer mobile hardware, where user are also allowed to customize the virtual musical instruments on the fly.

CHAPTER 2: AUDITORY PERCEPTION OF MATERIAL PROPERTIES

2.1 Introduction

Realistic sound effects that closely correlate with visual stimulus play a vital role in many virtual environment (VE) systems and interactive 3D graphics applications, e.g. video games, immersive simulators, and special effects. With recent advances in high-quality audio generation, physically-based sound synthesis is gradually becoming a feasible and suitable approach for automatic incorporation of convincing sound effects in 3D graphics applications. These methods offer synthesized sounds based on material properties, object geometries, and physical contacts that excite the resonant objects.

Among various physically-based sound synthesis methods, *modal synthesis* (Adrien, 1991; Shabana, 1997) is one of the most widely used real-time techniques in VE applications. It is highly efficient because it reduces complex vibrations of arbitrary geometries and materials to a linear combination of decoupled resonance modes. The geometry, characterized typically by shape and size, along with material parameters, determines the resonance modes obtained in the preprocessing step called *modal analysis*. When modeling resonant materials using modal sound synthesis, the *damping* component has always been a challenging issue, largely because the mechanism of energy dissipation for vibration is complex and not well understood. Moreover, modal decoupling is only feasible under



(a) Sound booth (b) Glass (c) Porcelain (d) Ceramic (e) Wood (f) Metal

Figure 2.1: Recording setup: (a) the sound booth where recordings take place. Other figures (b) - (f) the setups for recording impact sounds from real-world materials: glass, porcelain, ceramic, wood, and metal, respectively.

certain damping models. *Rayleigh damping* (Rayleigh, 1945) is one of the approximation models that enable such decoupling. As a result, it has been commonly adopted in rigid-body sound synthesis. However, to the best of our knowledge, though widely used in engineering applications, there has not been a formal analysis or rigorous evaluation of the Rayleigh damping model's transferability across different geometry (i.e. shapes and sizes). In other words, it is unknown if a single set of Rayleigh damping model parameters is sufficient for an arbitrary space of geometries or if the parameters would have to be "tuned" for changing geometry.

Without such an assumption, the Rayleigh damping model can only be applied on a per-object basis and a new set of damping parameters must be selected and tuned for every unique geometry – even *with the same materials*. This greatly limits the use of this approximation model and the adoption of modal sound synthesis in general, since finding appropriate Rayleigh damping parameters *per object* is usually non-trivial, tedious, and time-consuming. This process of material parameter tuning can quickly become prohibitively expensive for even a slightly complex VE scenario, where objects of different shapes with the same material are simulated. For example, the virtual fracture sound simulated by Zheng and James (2010b) is only feasible when assuming the same material parameters, including Rayleigh damping parameters, for the hundreds of fractured pieces.

In this chapter, I examine the Rayleigh damping model's transferability across different shapes and sizes, using both real-world audio recordings and synthesized sounds to perform both objective and subjective analysis of this approximation model. Our goal is to determine if auditory perception of material under Rayleigh damping assumption is "geometry-invariant", i.e. if this model is transferable across different shapes and sizes. To achieve this goal, I have conducted an empirical analysis and a number of psychoacoustic studies in exploring human auditory perception of materials using the Rayleigh damping model across different geometric variations, as well as crossmodal perception of material under the influence of geometry.

The rest of the chapter is organized as follows. In Sec. 2.2, I briefly describe the formulation of Rayleigh damping and related work on material perception in visual rendering and sound synthesis. Sec. 2.3 introduces our empirical study with real-world audio recordings. I analyze the recorded impact sounds of five sets of real objects. Each set contains several objects of the *same* material but *different* shapes or sizes. I verify if these recordings of the same material can be fitted to the same Rayleigh damping parameters with relatively small errors. Sec. 2.4 presents a psychoacoustic study

to evaluate material similarity. Based on the responses from the subjects, I analyze the transferability of Rayleigh damping model with respect to variation in shapes and sizes. In Sec. 2.5 and 2.6, I discuss our findings, the application of these findings, limitations, and possible future directions of this work.

2.2 Background

2.2.1 Rayleigh Damping Model

Sound from rigid bodies is generated due to resonant objects' vibration. In order to model this process accurately and efficiently, *linear modal synthesis* methods (Adrien, 1991; Shabana, 1997) are commonly adopted. It assumes small deformations during object vibration, thus its dynamics can be modeled as a linear system described by:

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

where $\mathbf{x} \in \mathbb{R}^{3N}$ is the displacement vector of the system, and $\mathbf{M}$, $\mathbf{C}$, $\mathbf{K}$ represent the mass, damping, stiffness matrices, respectively. $\mathbf{M}$ and $\mathbf{K}$ can be acquired through finite element analysis (O'Brien et al., 2002a), simple mass-spring formulation (Raghuvanshi and Lin, 2006b) and so on. In an *undamped* system, $\mathbf{M}$ and $\mathbf{K}$ can be diagonalized, and through generalized eigen-decomposition the solution of Eqn. 3.1 can be obtained, which is a series of *decoupled* harmonic oscillators, or *resonance modes*. Therefore, the complex dynamics of resonant objects are simplified and can now be computed efficiently. This process is called *modal analysis*, which is a standard structural analysis technique in engineering. However, if the *damping* term is present, the vibration dynamics can be reduced to a decoupled linear system only if $\mathbf{C}$ can be diagonalized as well as $\mathbf{M}$ and $\mathbf{K}$. Rayleigh (1945) proposed a formulation for the damping matrix:

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K}, \tag{2.2}$$

which is a linear combination of mass and stiffness matrices, where α and β are Rayleigh damping coefficients. Given this simplification, solutions to the linear system in Eqn. 3.1 are:

$$q_i = a_i e^{-d_i t} \sin(\omega_i t + \theta_i). \tag{2.3}$$

In this equation, ω_i and d_i are respectively the angular frequency and the decay rate of the ith mode, while a_i and θ_i are the excited amplitude and initial phase determined by runtime excitation.

I further observe that the Rayleigh damping assumption (Eqn. 2.2) and solutions to the dynamics formulation (Eqn. 3.4) define a frequency-decay relationship as a circle determined by Rayleigh damping coefficients α and β:

$$\omega_i{}^2 + \left(d_i - \frac{1}{\beta}\right)^2 = \left(\frac{1}{\beta}\sqrt{1-\alpha\beta}\right)^2. \tag{2.4}$$

This frequency-dependent decay rate model is a simplification of the complex mechanism of real internal material friction.

This simple damping formulation allows modal decoupling, and, therefore, it has been extensively used in rigid-body sound synthesis (Cook, 2002b; O'Brien et al., 2002a; Raghuvanshi and Lin, 2006b; James et al., 2006b; Ren et al., 2010; Zheng and James, 2010b; Ren et al., 2012a).

2.2.2 Related Work

Human hearing and auditory perception have been widely studied by researchers. Among them, Gaver (1988) designed experiments to study the perception of everyday sounds, more particularly in sonic events, such as struck bars of wooden and metallic materials, and went on to apply his results to designing user interface with auditory icons. Wildes and Richards (1988) studied recording audio of anelastic solids and determined that the *angle of internal friction*, $tan(\phi)$, is constant throughout all geometries of the same material. This work essentially defines a simple damping model, in which decay rate is linearly dependent on frequency. This damping model has been adopted by previous sound synthesis work (e.g. (Doel and Pai, 1998; Takala and Hahn, 1992)).

Klatzky et al. (2000) designed perceptual experiments with synthetic sounds using the same damping formulation and studied the relationship between perceived resonant materials and the

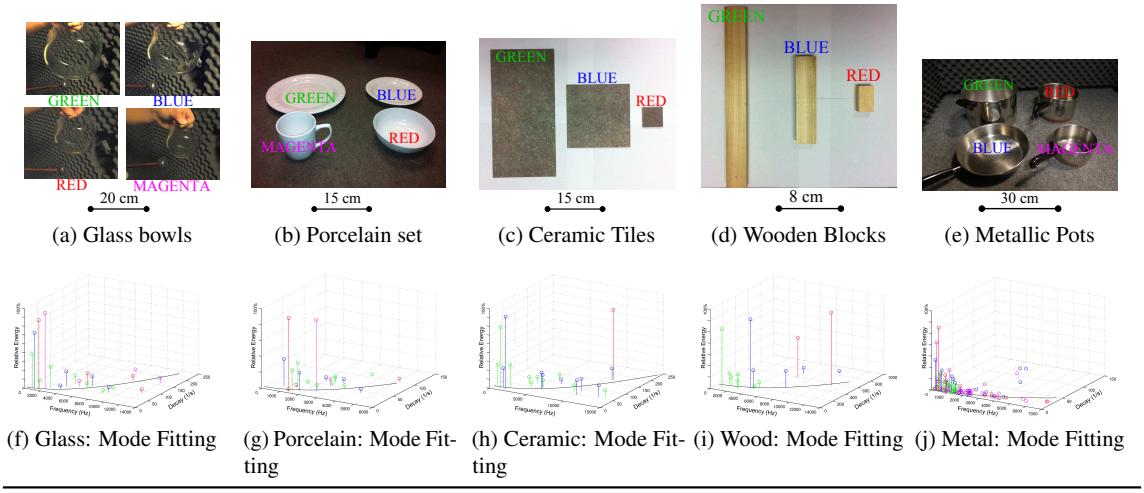


Figure 2.2: Fitting objects' resonance modes to the Rayleigh damping model. The top row shows real-world objects used in this experiment. The bottom row presents the fitting results, where the bottom plane represents the *frequency-decay* plane, the values in the height axis are relative energy, and the black curves on the frequency-decay plane visualize the fitted Rayleigh damping model. The color codes on the real objects match their extracted resonance modes in the same color.

parameters in this sound synthesis model. In particular they found that the decay parameter τ_d, or equivalently, the internal friction coefficient $tan(\phi)$, is a better indicator than frequency alone in determining material similarity. This work suggests that the decay parameter can be used as a shape-invariant material property for synthesizing sounds. They also found that when subjects were asked to directly assign a gross material category for a given synthetic sound, it is the combination of both the frequency and decay parameter that determines their categorization.

However, the constant internal friction model is not sufficient. Krotkov et al. (1996, 1997) analyzed the recordings of hitting real world objects of different materials and observed that for a given material, the internal friction is not a constant but instead a function of frequency. They suggest that the shape invariance may be encoded in the functional form of the relation of $tan(\phi)$ and frequency, and proposed that a quadratic function appears to be a possible fit. In fact, the *Rayleigh damping* model is one such quadratic formulation for relationship between damping and frequency.

Giordano and Mcadams (2006) studied synthesized, impacted xylophone bars with varying material and geometric properties. In their physical model, two viscoelastic damping coefficients were used to describe a material, which is similar to Rayleigh damping. The relation of these

properties to perceptual dissimilarity of the resulting sounds were studied, and a two-dimensional perceptual space was found to correlate with the material properties, namely the density and one of the two viscoelastic damping coefficients. Their result attests to the perceptual salience of energy-loss phenomena in sound source behavior. In another study, McAdams et al. (2004) studied material categorization of recorded impact sound, and a large set of acoustic descriptors related to frequency, damping, and loudness. They found that a slightly modified measure, $tan(\phi_{aud})$, of damping is sufficient for recognition of gross material categories. For example, they combined steel with glass as a "gross" category steel-glass. They also combined wood and plexiglass, a special type of plastic, as plastic.

Multi-modal interaction in material perception involving both audio and visual was studied by Bonneel et al. (2010). They varied the quality of synthesized sound and visual animation and studied subjects' material discrimination ability. Their study shows that high-quality audio rendering improves material perception, even when the visual rendering is low-quality. However, they did not show any correlation between visual and audio in material perception when virtual geometry vary. Visual perception of material reflectance is first studied through an exploratory psychophysical experiment in (Vangorp et al., 2007) to understand various influences on material discrimination in a realistic rendering setting. Their statistical analysis suggests that the accuracy of material perception is influenced by the geometrical shape of the object rendered with that particular material model. Nordahl et al. (2010) synthesized footstep sounds in real-time for both solid and aggregate materials. They performed a perceptual study of floor material recognition for three groups of subjects. One group listened to real-world recorded footsteps, another group interactively generated footstep sounds by themselves and listened to the real-time synthesized sounds produced by the proposed system, and the third group listened to pre-recorded footstep sounds produced by the same system. Their study show that, in the *interactive* setup, subjects were able to identify synthesized floor materials at a comparable accuracy with real-world recordings, while the performance with pre-recorded sounds was significantly worse than the other two. This work provides interesting insights in how multi-modal interaction affects auditory material perception. However, visual elements are not included in this study.

2.3 Empirical Analysis of Real-World Recordings

In this experiment, I use recorded audio from real-world objects to evaluate the transferability of Rayleigh damping model across different geometry. To verify if the Rayleigh damping model is capable of capturing the intrinsic material damping that does not vary with the object's shape and size, I fit the recorded audio to a sound synthesis model using the Rayleigh damping assumption. If impact sounds from *same-material* objects in *different shapes and sizes* can be well approximated with the same Rayleigh damping model, this material model can be considered geometry-invariant across these objects. Five sets of real-world objects are selected for this experiment. Each set consists of three to four items made of the same material but with different geometry, i.e. varying shapes and/or sizes. The five sets are glass bowls (Fig. 2.2a), a set of porcelain dinnerware (Fig. 2.2b), ceramic tiles (Fig. 2.2c), wooden blocks (Fig. 2.2d), and metallic pots (Fig. 2.2e). The legend under these figures indicate the sizes of these experimental objects. In this section, I first describe the setup of our recording sessions. Then, I use an existing method to extract the *resonance modes* from the original recordings, and the summation of these key features accurately represent the recorded audio. Finally, I present the results for fitting these resonance modes to corresponding Rayleigh damping models. The fitting results are shown in the bottom row of Fig. 2.2, where the resonance modes' colors match the color codes of the real objects shown in the top row.

Here, I show feature extraction results for materials: porcelain, ceramics, wood, and metal. The objects used in this empirical study are shown in Fig. 2.2b, 2.2c, 2.2d, and 2.2e. The power spectrograms of the feature extraction results are Fig. 2.5, 2.4, 2.6, 2.7.

2.3.1 Recording Setup

Recordings were performed in a professional-quality sound booth, where all walls are padded with absorption materials to reduce reverberation effects, as shown in Fig. 2.1a. In order to generate impact sounds that best capture the intrinsic resonance properties of objects, I try to minimize their contacts with other articles. In most cases, rubber bands are used to suspend the object of interest, allowing the object to vibrate with minimum external damping due to contacts. The metallic pots are suspended by the attached metal loops (Fig. 2.1f). To reduce sounds coming from the striker during the impact motion, I adopt a mallet with a hard rubber head as the striking object. Special

care is taken during the striking motion to minimize the swinging of the struck object, so that ringing sound effects are reduced. In order to limit the variation to only geometry and material, I manually control the striking motion's magnitude and direction to be as consistent as possible throughout all recordings. To diminish the hit point variation, all strikes are aimed at the center position of objects, for example the center point on the bottom of the glass bowls, metallic pots, and porcelain set. The recording setups for some examples are shown in Fig. 2.1.

2.3.2 Resonance Mode Extraction

Recorded audio is complex and high-dimensional data, which are difficult to directly map to any simple material model. As shown by van den Doel et al. (2001a) and Corbett et al. (2007), many rigid-body impact sounds can be well approximated with the summation of a bank of *damped sinusoids* with different frequencies, decay rates, and amplitudes. Each damped sinusoid is considered one *resonance mode*, whose frequencies and decay rates are intrinsic to the particular object, while the amplitudes vary with the magnitude and location of an impact applied to the object. I adopt these *modes* as a high-level representation for the original sound.

I use the feature extraction method in (Ren et al., 2013) to determine the resonance mode representation of the recorded impact sound clips. This method uses an optimization framework that extracts modes from the original audio in a greedy fashion. *Power spectrograms* of the original recorded audio, the audio from mixing only the extracted resonances modes, and the absolute difference, i.e. error, between the two are shown for the glass bowls in Fig. 2.3, while the data for the experimental objects of other materials are included in Fig. 2.5, 2.4, 2.6, and 2.7. The error plots show the extracted modes accurately capture the frequencies and decay rates of all prominent components in the recordings. Therefore, it is appropriate to use these modes to represent the original recordings for the purpose of studying the damping model, which defines a frequency-decay relationship for objects' vibration. For many objects, noticeable error appears in the range 0 - 1000Hz, which is quite possibly due to sound of the striker, i.e. the hard rubber ball, and the impact motion. How to separate the sound of striker and the struck object is still an open problem, which introduced error to the resonance mode analysis process.

2.3.3 Fitting Modes to the Rayleigh Damping Model

Once a resonance mode representation of a recording is acquired, I study how well the Rayleigh damping model can approximate these modes. I do so by fitting a curve following the Rayleigh damping model to these collected mode data points. For each material, the resonance modes of objects with different geometry are fitted to the same curve defined by one set of Rayleigh damping parameters. As shown in Fig. 2.2, I fit the curves on the 2D bottom plane to the observed (frequency, decay) pairs of modes. The values in the height axis represents relative energy of modes under a certain excitation. The relative energy values are used as weights in the least square regression, where the residual is defined as the difference between the observed and the predicted decay values. I weight the residual with relative energy because I want the fitted Rayleigh damping model to predict the more important modes (i.e. the higher energy modes) better than the less important ones. The fitting results for the five materials are shown as the black curves in Fig. 2.2f, 2.2g, 2.2h, 2.2i, and 2.2j. In Sec. 2.3.3, I statistically analyze the quality of the fit.

Quantitative Analysis of Goodness of Fit: In order to evaluate how well the curves fit the data, I compute the *coefficient of determination*, R^2, which is a widely used measure for assessing the *goodness* of regression using least squares techniques (Steel and Torrie, 1960). I adopted the standard weighted R^2 formulation,

$$R^2 = 1 - \frac{\Sigma w_i \times (y_i - \hat{y}_i)^2}{\Sigma w_i \times (y_i - \bar{y})^2}, \tag{2.5}$$

where $\{y_i\}$ are the decay values of the observed resonance modes, $\{\hat{y}_i\}$ are the decay values predicted by the Rayleigh damping model given the resonance modes' frequencies, $\bar{y}$ is the mean of $\{y_i\}$, i.e. the average value of observed decays, and $\{w_i\}$ are the weights, which are the relative energies of modes. Based on the standard interpretation of R^2 measure, an R^2 of 1 means the curve model perfectly fits the observed data, and the closer the value to 1 the better the fitting. The R^2 measures of the fitted Rayleigh damping models for the five materials in our experiment are listed in Table 2.1 ($p < 0.0001$ for all materials). This indicate the Rayleigh damping model generate predictions that are strongly and significantly correlated with the observed models of all materials. In four out of the five materials, the model accounts for approximately 75% of the observed variance in modes.

Notice the R^2 measure is noticeably lower for the wooden material compared with that of other materials. I believe that the anisotropy and other complex properties (e.g. heterogeneity of micro-

Table 2.1: Goodness of Fit for the Rayleigh Damping Model

	Glass Bowls	Porcelain Set	Ceramic Tiles	Wood Blocks	Metallic Pots
R^2 Measure	0.77	0.78	0.74	0.63	0.77

structures) of the wooden material contribute to the fact that the simple Rayleigh damping model cannot fully reflect the damping phenomena of wood, hence the resonance modes fitted relatively poorly to the Rayleigh damping model. The relatively higher decay rates of the modes of wooden blocks may have also led to the poorer fitting. Nonetheless, the R^2 measures for all materials are reasonably high, indicating that in our experiment the Rayleigh damping approximation is accounting for a substantial, and highly significant, amount of the variance in the observed modes.

2.4 Perceptual Study on Material Similarity

In addition to the empirical experiment described in Sec. 2.3, I also conduct a psychoacoustic study where, in each trial, I ask subjects to determine if two sound clips played side-by-side are coming from objects made of the same *material*, while the objects can be of the same or different *geometry*. The study objective is to determine if the Rayleigh damping model can indeed capture the perceived material property sufficiently well to achieve transferability across different geometry.

Throughout this perceptual study, the independent variables are material and geometry (i.e. shape and size). The dependent variables that I measure as results are accuracy and confidence for experiments using recordings and consistency and confidence for those using synthetic sounds. Sec. 2.4.1 introduces what independent variables are used, and Sec. 2.4.2.1 describes how the study is designed to reasonably sample all independent variable combinations. Finally, Sec. 2.4.4 presents a detailed definition for the dependent variables and their values for the studies. I perform *within subject* study, where a single subject answer trial questions covering different combinations of independent variables, and in the end a *within subject* analysis of dependent variables is presented. In order to counterbalance, all the trial questions in this study appear in randomized order for every subject. In addition, the number of different-material synthetic sounds is very comparable to the number of same-material synthetic sounds, I also did not inform the subjects of the ratio of same material versus different material, and they go through the study treating each trial question as an independent

incidence. Combining these factors, I believe our subjects do not have any assumption about the material identities before hand and are not biased to give a same-material or different-material answer in either way.

2.4.1 Audio Stimuli

In this experiment, subjects' perceived sense of materials is directly used as the indicator for determining whether Rayleigh damping model can be considered transferable across different geometry. However, human perception of materials is not solely dependent on the intrinsic material itself. It can also be affected by objects' geometry (Klatzky et al., 2000; Vangorp et al., 2007). I hope to study to what extent this effects the perception of real-world materials, and this finding serves as the baseline for interpreting the results from synthetic sound. Therefore, both *recorded* and *synthetic* audio clips are used as stimuli in our perceptual study.

For *recorded* audio stimuli, I use all the recordings acquired in Sec. 2.3. The first row in Fig. 2.2 shows pictures of the 18 objects for which impact sounds are used.

As to *synthetic* sounds, to explore the wide range of geometry and material variations, I selected a representative set of variations for generating the audio stimuli.

Shape variation: stick, cube, bunny, sphere, plate, and torus. They are shown in Fig. 2.8. These six sample shapes are chosen to represent shape variations such as complexity, dimensionality, and genus. For example, the simple cube shape is used, while the bunny shape is much more complicated. The plate is flat and circular, while the stick is much larger in one dimension than the other two. The sphere is a closed shape, while the torus has genus one. In addition, all shapes are solids that contain no cavity.

Size variation: small, medium, and large. I also vary the size of our sample shapes in order to study potential size-induced change in material perception. Three-size variations are adopted and illustrated on the example of bunny in Fig. 2.9. The smallest bunny is about 6cm tall, while the medium and large ones are respectively 2x and 4x the size of the small one. The same size variation is applied to all other shapes.

Material variation: metal, wood, glass, plastic, and porcelain. These five synthetic resonant materials are chosen to represent a variety of materials, and they are visualized on the sample shapes in Fig. 2.8.

In total, there are 90 variations arising from the combinations of the six shapes, three sizes, and five materials. Synthetic impact sounds for these 90 variations are generated using modal synthesis with the Rayleigh damping assumption, and they serve as the synthetic audio stimuli in our psychoacoustic experiments.

2.4.2 Study Design

In designing these experiments, I face two major challenges. Firstly, as described in the previous subsection, I have 18 recorded and 90 synthetic audio stimuli. If I aim to cover all variations in the stimulus space, picking two stimuli to form a question results in a huge number (nearly $12,000$) of combinations which is infeasible for the study questionnaire. Secondly, human perception of material is inevitably affected by geometry variation. It is difficult to separate such effects in our study. Moreover, most people probably do not pay enough attention to auditory sensations in their daily lives to have closely observed the geometry effects in perceiving materials. Therefore, it is challenging to study auditory perception of materials across different geometry due to subjects' inability to distinguish variation in sound caused by geometry or material variation.

In this section, I first present an efficient stimulus sampling scheme that systematically picks pairing of audio stimuli to sample the combination space with a relatively small number of questions. Then, I describe our three-segment study procedure as an effort to better understand the perceived material variation due to geometric effects.

2.4.2.1 Stimulus Sampling

I randomly sample the complete stimulus combination space in the approach described below, where each subject is asked to complete a total of 56 trial questions. In particular, each subject judges six pairs from the recorded and 50 pairs from the synthetic stimulus set. I categorize our stimulus combinations based on their *material* and *geometry* configurations: identical or different material and identical or different geometry. This high-level grouping allows us to control the sample counts in each category and guarantees well distributed sample points that help us observe major trends.

Recorded stimulus sampling: Six trials are performed by each subject, and they are randomly selected from the 153 combinations made possible by picking any two from the 18 recorded stimuli. The random sampling follows the grouping listed in Table 2.2.

Table 2.2: Recorded stimulus sampling

	Material	Geometry	Count	
Group 1	Identical	Different	5	Total: 6 Trials
Group 2	Different	Different	1	

Table 2.3: Synthetic stimulus sampling

	Material	Geometry	Count	
Group 3	Identical	Different	30	Total: 50 Trials
Group 4	Different	Identical	10	
Group 5	Different	Different	10	

In Group 1, identical material and different geometry, one sample is selected for each real-world material out of the five I have, and the geometry combination is randomly selected. Group 2 is randomly selected following the constraint of different material and different geometry. I pick more samples for Group 1 because I hope to gather more data with real-world *recordings* on how geometry affects material perception with the same material.

Synthetic stimulus sampling: Each subject is asked to complete a total of 50 trials for this category. The proposed sampling is outlined in Table 2.3.

Group 3 is the focus of this study, since it evaluates if the same sense of material is preserved across geometry variation when the synthetic stimuli are generated with the same Rayleigh damping material model. Geometry variation comes in two forms: shape and size. Therefore, Group 3 can be decomposed into three subgroups: different in both shape and size, only different in shape, and only different in size. 10 trials are performed respectively for each of these three subgroups. Particularly, the combination space is huge for the subgroup that is different in both shape and size. I propose the following scheme that achieves effective sampling for this subgroup. Fig. 2.10 illustrates the sampling scheme. First, 18 sample pairings are chosen from all shape-size combinations, and these samples satisfy that each chosen object is strictly selected twice in all combinations, and each pair is strictly different in both size and shape. Three such 18-combination groups are designed and color coded respectively in red, green, and black in Fig. 2.10. It appears these three groups evenly cover most combinations in the space. In each round of the study, one of the three groups is randomly selected, and 10 of the chosen group's 18 pairs are then randomly selected to represent the shape and size variation combination. Finally, for these 10 fixed geometry configurations, I randomly

assign each of the five material choices to two of them. Thus, the 10 sample pairs for this subgroup are decided. For the subgroup of 10 pairs only different in shape, I fix the size configuration to be medium, randomly assign each of the five materials twice, and randomly combine shapes from the six options. Similarly, for the 10 pairs only different in size, I fix the shape configuration (five are fixed to be plates, and five are torus), two pairs for each material, and randomly combine sizes from the three options.

The 10 pairs in Group 4 and 5 follow the constraint of covering all possible material combinations (i.e. select any two out of five). For Group 4, an identical geometry configuration is randomly drawn for each pair, while for Group 5, two different geometry configurations are randomly selected for each trial.

With the above described sampling scheme, I define an approach that generates pairings in a random yet controlled fashion that provides us with experiments that cover a wide range of variants and focus on specific configurations that are central to our study (i.e. Group 3 in Table 2.3). Note that I did not include the group of identical material and identical geometry in either the recorded audio or the synthetic audio samplings. This is due to the subject's perfect identification rate for such pairings in our preliminary studies.

2.4.2.2 Study procedure

Our perceptual study is conducted in the format of online surveys. The interface of the study is shown in the accompanying video, and the study is designed to consist of the following three major parts, where each subject takes a 7-trial training session and then judges 67 stimulus pairs.

Training session: Geometry variation in objects leads to different qualities in sounds, which in turn affect subjects' auditory perception of material. It is challenging to separate the geometry and material influence in auditory perception. In order to take the geometry effect into consideration, our material similarity experiment includes a short training session, which shows subjects real-world sounds from objects in various materials and geometry. Subjects are firstly instructed to be aware that same-material objects can sound differently due to geometry variation. A video of impact sounds coming from four glass bowls that vary in shape and size are shown. Then they are asked to complete a seven-trial training, where each trial consists of two side-by-side audio clips. Subjects are asked to decide if the two clips are from the same material. Immediately after answering each training

question, images of the resonance objects are revealed to subjects, which show audio and visual renderings of both the geometry and material of the experiment objects.

Material discrimination: The second part is an audio-only material discrimination study. Subjects are presented with two side-by-side audio clips and asked two questions for each trial. First, they are asked if the two audio clips come from objects made of *the same material*. Radio buttons for yes and no are provided for subjects to input their answers. Second, they are asked to rate how *confident* they are with their answer. Scores ranging from 0 to 10 represent 'not confident at all' to 'very confident'. The 56 trials sampled as described in Sec. 2.4.2.1 are conducted in this part of experiments.

Material discrimination with geometry visualization: The final part of this experiment is an audio-visual material discrimination study. The questionnaire is the same as the previous part, except that two side-by-side *images* corresponding to the two audio stimuli are also shown to subjects. These images are visual renderings of the resonance objects' *geometry*, and subjects are informed that they only carry geometry information and no texture or material clues. This allows us to explore how the added geometry visualization affects the subject's auditory perception of materials. A total of 11 trials are conducted, and they are sampled as shown in Table 2.4.

Table 2.4: Synthetic stimulus sampling with **geometry visualization**

	Material	Geometry	Count	
Group 6	Identical	Different	7	Total: 11 Trials
Group 7	Different	Identical	2	
Group 8	Different	Different	2	

A focus study: While the above study is relatively thorough, I hope to obtain more data in the category that is the most interesting to this study, i.e. Group 3 in Table 2.3, since I aim to evaluate if *Rayleigh damping* parameters transfer across geometry. Therefore, I also provide a focus perceptual study that only asks 21 trial questions after the 7-step training. 15 of the the 21 questions are subsamples chosen from Group 3 in Table 2.3, and the other 6 are randomly sampled in different material combinations, so that the study is more balanced with both same and different material pairs.

2.4.3 Participants

A total of 42 volunteer subjects, age between 21 and 45, were recruited for this perceptual study. 20 of them finished the full study, and among them 6 were female. The average age for this group

is 28.40. The other 22 subjects completed the focus study, 8 are female, and the group average age is 32.27. All subjects reported normal hearing and performed the study at their own pace on a personal computer. All of them used headphones in the study for better audio quality, since frequency components at the high and low ends of audible spectrum might be inaudible through some consumer speakers.

2.4.4 Results and Analysis

The result of each trial is measured by the following two variables.

Consistency: Subjects are asked to answer if two audio clips are from the same material in each trial. For recorded stimuli, the concept of *accuracy* is directly adopted, since the ground truth of same or different materials for each trial pair can be determined, and an answer is correct or incorrect can be decided. For synthetic stimuli, I define *consistency*, which is analogous to *accuracy* for recorded stimuli. If subjects' answer is consistent with the material model assumption, I define consistency as 1.00. If not, it is defined to be 0.00. For example, if two audio clips in one trial are synthesized using the same material parameters, and the subject consider them the same material, I assign 1.00 to the consistency of this trial. The mean consistency is essentially the proportion of subjects' answers consistent with the tested material model assumption.

Confidence: Besides the yes and no material discrimination question, subjects are also asked to rate their confidence with their decision. This 0 - 10 value indicates how confident the subject is, while 0 means not confident at all, and 10 is very confident. In other word, if the subject has difficulty or uncertainty in answering the material discrimination question, the confidence value of this trial should be low.

Results from the full-length studies are used in the analysis below. The focus study is solely designed to provide more samples for Group 3 in Table 2.3, so its results are only used in analyzing the across-shape and across-size cases described in Sec. 2.4.4.2. In all the results, I report both *means* and 95% *confidence intervals* (CI) for the accuracy/consistency and confidence values for each group, presented as CI centered around means. Where appropriate, *paired t-tests* (Howell, 2009) are performed to test the statistical significance of hypotheses on comparisons between two groups, and the *p-value*, which represents the probability of the observed result occurring by chance, is reported.

I adopt .05 as the p-level for significance. The rest of this subsection includes the results and analysis of each data group. More observation and discussion are presented in Sec. 2.5.

2.4.4.1 Recorded stimulus trials

Table 2.5 shows the 95% confidence intervals of accuracy and confidence for all trials of *recorded* stimulus respectively in Group 1 and 2. Notice in Table 2.5, the accuracy rate is only 84.75% for Group 1. This indicates even with real-world recordings, when sounds of objects from identical materials are presented, subjects can be affected by the geometry variation and mistake identical materials as different. Perfect material discrimination across geometry variation is improbable. The variance in accuracy values is relatively large for Group 2, I suspect it is due to the small number of trials I performed for this particular category.

Table 2.5: 95% CI for accuracy and confidence for recorded audio trials

	Material	Geometry	Accuracy	Confidence
Group 1	Identical	Different	84.75% ± 6.95%	7.46 ± 0.52
Group 2	Different	Different	85.00% ± 16.06%	7.70 ± 0.77

2.4.4.2 Synthetic stimulus trials

Table 2.6 presents the 95% CI centered around mean consistency and confidence for each group throughout all synthetic audio trials. Notice the consistency rate for Group 3 is quite high, which indicates subjects perceive Rayleigh damping as transferable across geometries in a large proportion (around 76%) of the study trials. A paired two-tailed t-test between Group 3 and 5 indicates subjects are more capable of detecting mismatches than matches in material, when geometry differs ($t_{consistency}(20) = -3.01$, $p_{consistency} < 0.007$; $t_{confidence}(20) = -2.43$, $p_{confidence} < 0.025$). The same type of t-test between Group 4 and 5 fails to support the hypothesis that a geometric mismatch heightens reports of material mismatch ($t_{consistency}(20) = -1.94$, $p_{consistency} < 0.068$; $t_{confidence}(20) = -1.23$, $p_{confidence} < 0.233$).

In order to evaluate the differences among all materials, I also categorize the results based on the five materials in the study. Table 2.7 presents this result for each material throughout all synthetic

Table 2.6: 95% CI for consistency and confidence for synthetic audio trials

	Material	Geometry	Consistency	Confidence
Group 3	Identical	Different	76.73% ± 4.29%	7.25 ± 0.57
Group 4	Different	Identical	81.17% ± 4.97%	7.65 ± 0.53
Group 5	Different	Different	87.50% ± 4.69%	8.07 ± 0.46

Table 2.7: 95% CI for consistency and confidence for synthetic materials

	Wood	Plastic	Porcelain	Metal	Glass
Consistency	70.77% ± 7.92%	90.81% ± 4.82%	77.64% ± 7.23%	81.69% ± 6.84%	80.38% ± 7.49%
Confidence	7.26 ± 0.52	7.54 ± 0.62	7.25 ± 0.71	7.87 ± 0.60	7.24 ± 0.62

audio trials. For all five materials, consistency and confidence are relatively high. Notice that the material of wood leads to the lowest performance.

Synthetic stimulus trials - same material, across shapes and sizes respectively: The focus of our study is to test if the same sense of material is preserved across different geometry (i.e. shapes and sizes), if the same material parameters including the same Rayleigh damping coefficients are assumed. Below, I present results categorized respectively into different shapes while sizes are fixed (Fig. 2.11) and different sizes while shapes are fixed (Fig. 2.12). All results in this part are calculated from the identical material trials in both the full-length and the focus study. Therefore, a total of 42 subjects' results are included. Fig. 2.11 shows results across all shapes: stick, cube, bunny, sphere, plate, and torus. Fig. 2.11a, 2.11b, 2.11c, 2.11d, and 2.11e present data for materials: wood, plastic, porcelain, metal, and glass, respectively. Fig. 2.12 shows results across all sizes: small, medium, and large. Fig. 2.12a, 2.12b, 2.12c, 2.12d, 2.12e present data for materials: wood, plastic, porcelain, metal, and glass, respectively. Once again, trials of wooden material yield one of the worst consistency rates. Additionally, the *small* objects in general appear to be identified as inconsistent with the material model more often than the other sizes. It also appears consistency varies more with shapes than sizes, which means, compared with sizes, a drastic shape change is more likely to lead subjects to identify sounds produced by the same material parameters as coming from different materials.

Synthetic stimulus trials - with geometry visualization: Table 2.8 shows results of the trials in which subjects are provided with visualization of the resonance objects' geometry. The consistency and confidence values are remarkably high. Group 7 has the highest consistency, and it is mainly

due to that the geometry is identical. When a subject is shown the visualization of two identical geometries, it is clear the only variable is material. In this case, subjects can judge material similarity purely based on the variation in the perceived audio and not be affected by geometry variation at all. The results with geometry visualization are also categorized into different materials and shown in Table 2.9. The mean consistency and confidence values are generally larger than those of the audio only results (Table 2.7), while the standard deviations are also larger, which can be due to the smaller number of trials performed. In fact, in the comments left by several subjects, they specifically pointed out that the geometry visualization made the material discrimination task easier for them.

Table 2.8: 95% CI for consistency and confidence for synthetic audio trials with *geometry visualization*

	Material	Geometry	Consistency	Confidence
Group 6	Identical	Different	87.14% ± 7.57%	7.62 ± 0.62
Group 7	Different	Identical	95.00% ± 6.74%	8.53 ± 0.71
Group 8	Different	Different	82.50% ± 12.87%	7.85 ± 0.74

2.5 Discussion

Through the empirical experiment in Sec. 2.3 and the perceptual study in Sec. 2.4, I make the following key observations.

The Rayleigh damping model can be considered geometry-invariant: In the empirical study, the Rayleigh damping model appeared to serve as a reasonably good approximation for five real-world resonance materials, based on the observed fitting results (i.e. R^2 measure in Table 2.1) for the experimental materials across different geometries. In addition, synthetic audio generated with the Rayleigh damping model were tested in our perceptual study. High consistency between these adopted synthetic materials and subjects' material discrimination were recorded (76.73% for Group 3 synthetic audio only trials in Table 2.6 and 86.14% for Group 6 synthetic audio with geometry visualization trials in Table 2.8). The consistency rates indicate that synthetic sounds of various

Table 2.9: 95% CI for consistency and confidence for synthetic materials with geometry visualization

	Wood	Plastic	Porcelain	Metal	Glass
Consistency	92.65% ± 7.72%	92.11% ± 7.20%	83.33% ± 13.60%	87.72% ± 11.31%	78.24% ± 15.89%
Confidence	7.78 ± 0.66	8.11 ± 0.69	7.53 ± 0.94	8.10 ± 0.63	7.49 ± 0.83

geometry (i.e. sizes and shapes) using the same Rayleigh damping model are perceived as the same material at a very high percentage. In addition, the broken down across-shape and across-size consistency rates shown in Fig. 2.11 and Fig. 2.12 and the average consistency rates for each material recorded in Table 2.7 and Table 2.8 (with geometry visualization) are also relatively high, especially the ones with geometry information. Moreover, I need to consider that subjects are not capable of perfectly discriminating materials due to geometry variation. Evidence for this is shown in the recorded audio trials. Even if the underlying material is identical (no approximation with any model), subjects can mistake them for different materials. In fact, the mean consistency values for synthetic stimuli in Group 3 and 5 are not significantly smaller than those of recorded stimuli in Group 1 and 2, respectively. It suggests that synthetic stimuli with Rayleigh damping assumptions can be considered good approximations in terms of preserving the sense of materials that is comparable with that of real-world audio. From the results in our experiments, I verified when applying the same set of Rayleigh damping parameters across different geometries, the same sense of material is preserved to a large extent.

Multi-modal effects in auditory material perception: Respectively compare results in Table 2.6 and Table 2.8, and Table 2.7 and Table 2.9. I observe, with the added visualization of object geometry, subjects' material perception shows significantly higher agreement with the Rayleigh damping model. In fact, a paired two-tailed t-test between Group 3 and 6 has $t_{consistency}(20) = -3.34$, $P_{consistency} < 0.003$, and $t_{confidence}(20) = -2.29$, $P_{confidence} < 0.033$, and same type of t-test between Group 4 and 7 show $t_{consistency}(20) = -3.09$, $P_{consistency} < 0.006$, and $t_{confidence}(20) = -2.42$, $P_{confidence} < 0.026$. This strongly indicates that when visual geometry information is present, which is the case for most graphics and virtual environment applications, the Rayleigh damping model is perceived as geometry-invariant at an even higher rate. Therefore, Rayleigh damping assumptions should be readily adopted as a geometry-invariant material approximation model in most virtual environment applications. In scenarios, where multiple objects of various geometry are present, I can apply the same set of material parameters in Rayleigh damping model to them, and users would generally perceive them as bearing the same auditory material.

Rayleigh damping's limitations: Notice in Table 2.1, the fitting result is the poorest for the wooden blocks in this study. With synthetic audio samples (results in Table 2.7, Fig. 2.11 and Fig. 2.12), the wooden material also seems to be perceived as the least consistent with Rayleigh damping model. I

posit that the Rayleigh damping model is not ideal for approximating anisotropic materials like wood, which display complex energy decay effects. In addition, the high decay rates of wood are possibly pushing the limits of Rayleigh damping assumption. Lastly, human's auditory perception in high frequency range is poor, and I believe it largely contributes to the worse agreement for smaller objects (as shown in Fig. 2.12), which generally have resonance modes of higher frequencies. The synthetic audio of porcelain and glass in our experiments also have higher frequencies, and their discrimination rates appear less consistent with the Rayleigh damping assumption (Fig. 2.11c and Fig. 2.12c, and Fig. 2.11e and Fig. 2.12e). Therefore, based on our studies, for relatively extreme cases like highly complex decay effects, large decay rates, and high frequency range, Rayleigh damping model is not ideal.

2.6 Conclusion and Future Work

In this chapter, I have presented a number of experiments in which I examine the auditory perception of material across different geometry using the Rayleigh damping model for interactive sound synthesis in VR applications. I perform these studies both quantitatively and qualitatively by analyzing the real-world audio recordings and the synthetic sound clips generated by the Rayleigh damping model to determine if the material perception under this model is *geometry invariant*, i.e. does not vary across shapes and sizes.

Statistical analysis shows that the auditory perception of materials under the Rayleigh damping model for *homogeneous materials* is not influenced much by variation in shapes and/or sizes. However, our study results suggest that the Rayleigh damping model does not provide equally good approximation for materials with heterogeneous micro-structures, such as wood. Other more complex (perhaps more general but likely more compute-intensive) damping models (Adhikari and Woodhouse, 2001) for capturing the material properties of sounding objects should be investigated and evaluated.

Reinforcing expectations based on well-known principles in crossmodal perception, our psychoacoustic experiments indicate that visual perception of geometry has noticeable effects on auditory perception of materials. This result is also consistent with study results in crossmodal perception

and earlier study by (Vangorp et al., 2007) in which they found the visual perception of material is influenced by the geometry of objects.

These findings enable the wide adoption of Rayleigh damping in virtual environment applications for real-time modal sound synthesis and efficient reuse of material parameters under this approximation model across different geometry, thereby alleviating time-consuming per-object material parameter tuning.

In the future, I hope to perform analytical and qualitative comparisons between the Rayleigh damping model and other damping models of higher degrees, as well as how different models affect sound synthesis algorithms both in rendered sound quality and computation costs. In addition, how to design perceptual studies to reduce the geometry variation effects in material discrimination tasks is worth studying. Perceptual studies on crossmodal (esp. auditory-visual) perception in virtual reality also demand more exploration.

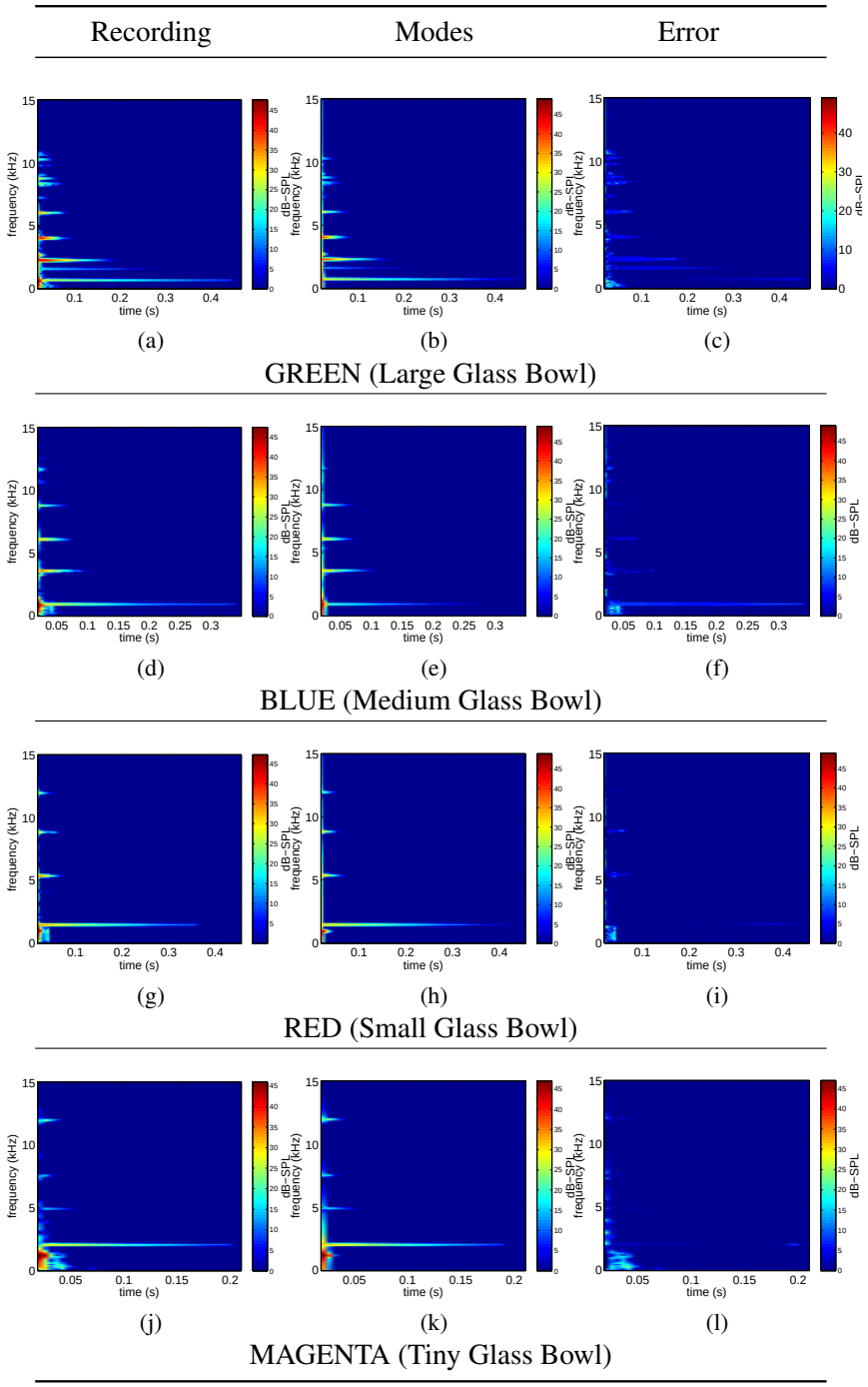


Figure 2.3: Resonance mode extraction results for **glass** bowls of different shapes and sizes, as shown in Fig. 2.2a. Fig. 2.3a - Fig. 2.3l show, for each object, the power spectrograms of the recorded audio, the extracted resonance mode mixed audio, and the absolute error between the two.

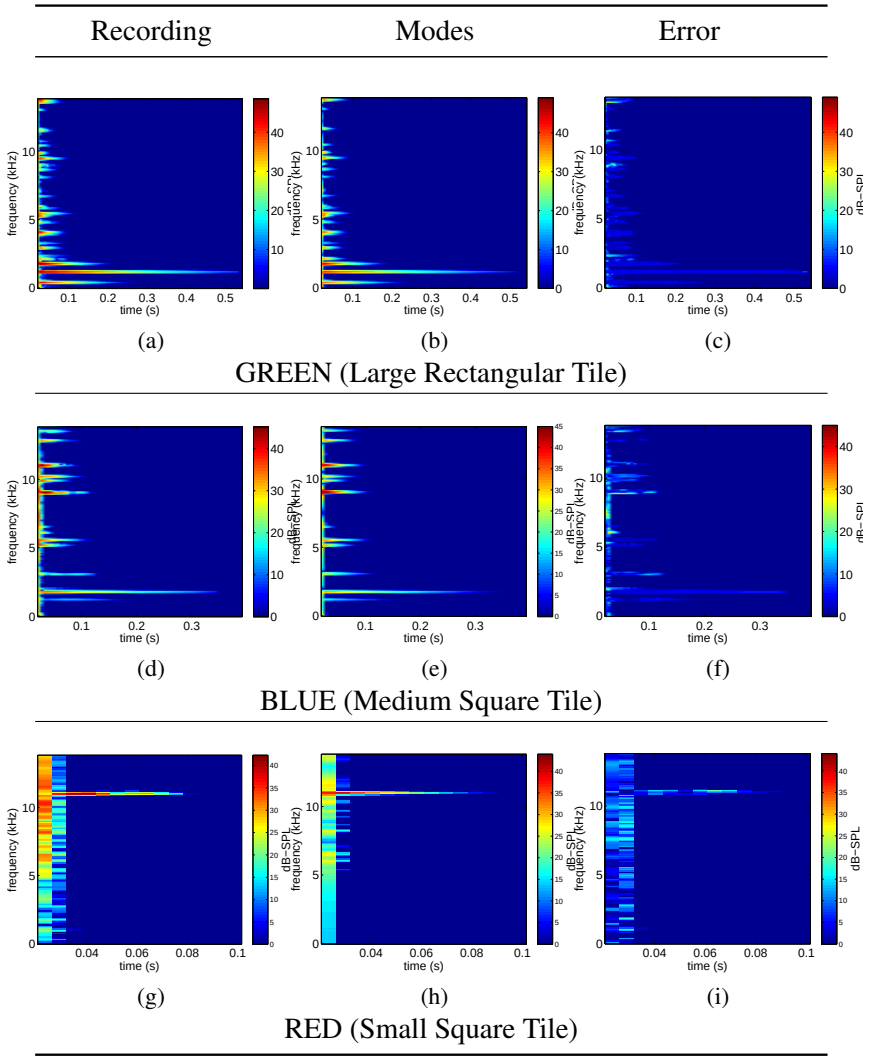


Figure 2.4: Resonance mode extraction results for **ceramic** material, as shown in Fig. 2.2c. For example, GREEN object in this figure represents the object labled GREEN in Fig. 2.2b. Fig. 2.4a - Fig. 2.4i show, for each object, the power spectrograms of the recorded audio, the extracted resonance mode mixed audio, and the absolute error between the two.

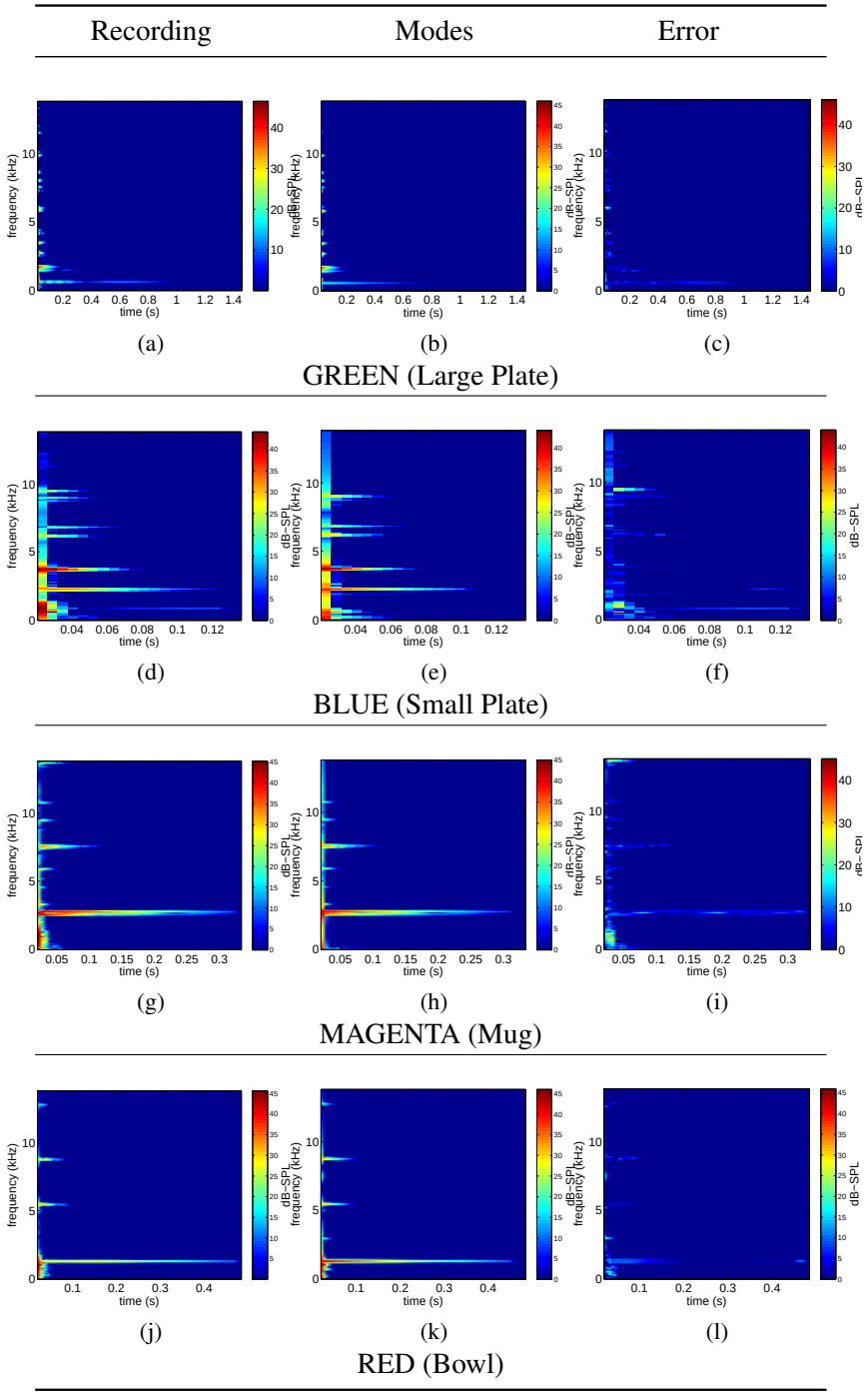


Figure 2.5: Resonance mode extraction results for **porcelain** material, as shown in Fig. 2.2b. For example, GREEN object in this figure represents the object labled GREEN in Fig. 2.2b. Fig. 2.5a - Fig. 2.5i show, for each object, the power spectrograms of the recorded audio, the extracted resonance mode mixed audio, and the absolute error between the two.

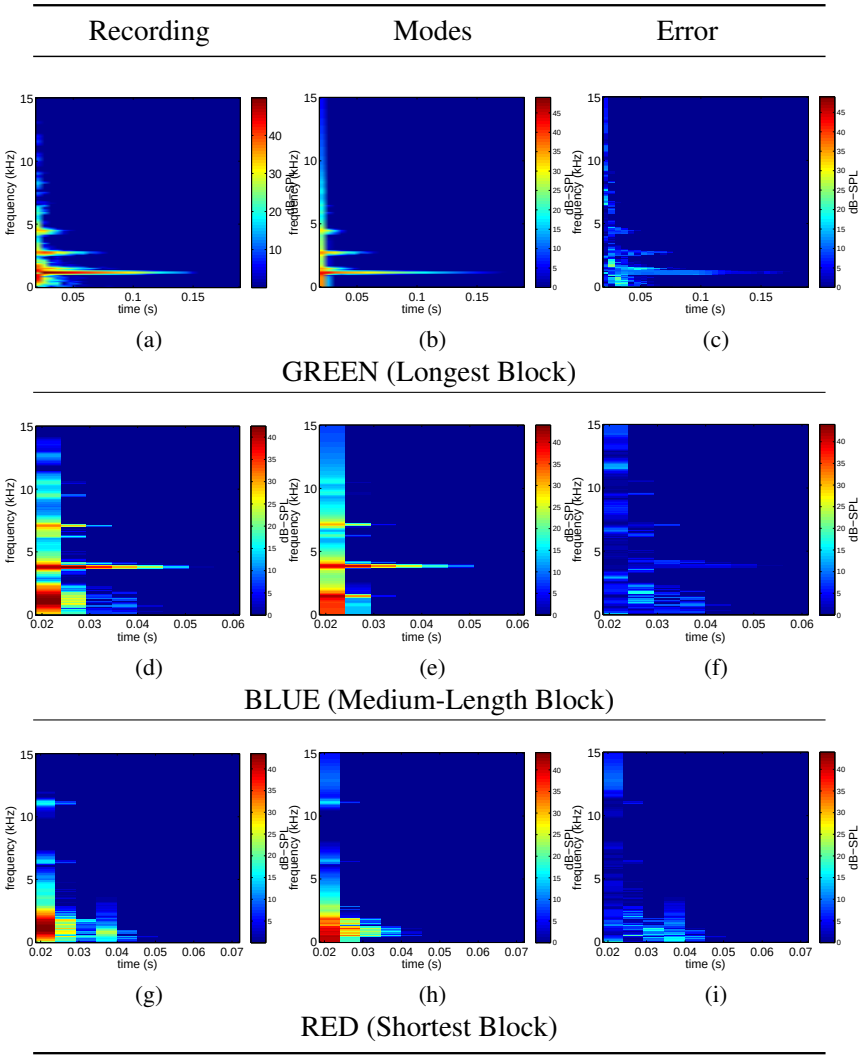


Figure 2.6: Resonance mode extraction results for **wooden** material, as shown in Fig. 2.2d. For example, GREEN object in this figure represents the object labled GREEN in Fig. 2.2d. Fig. 2.6a - Fig. 2.6i show, for each object, the power spectrograms of the recorded audio, the extracted resonance mode mixed audio, and the absolute error between the two.

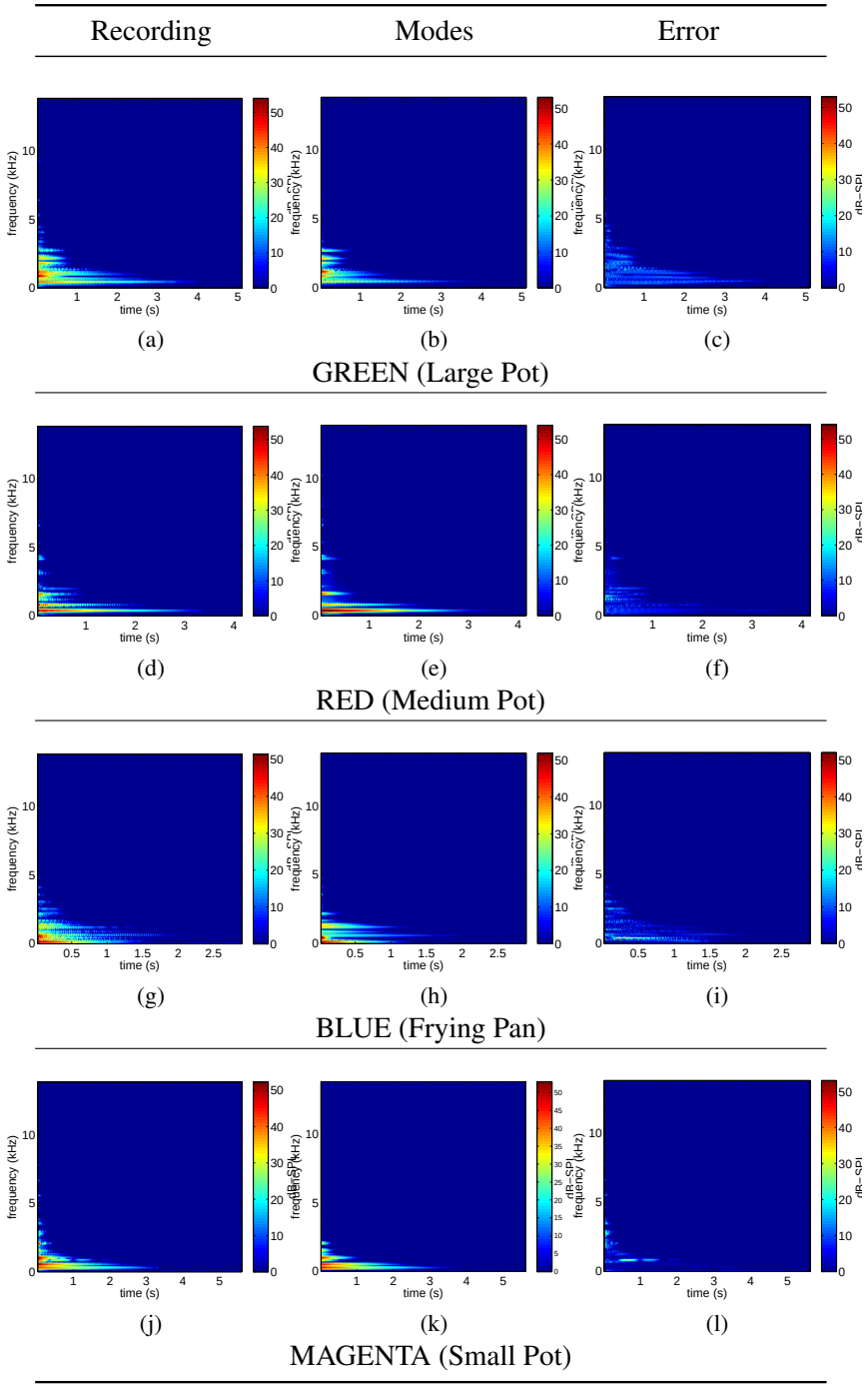
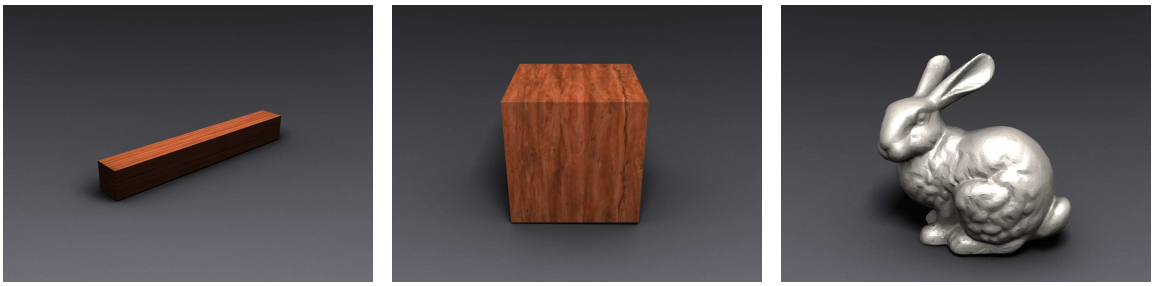
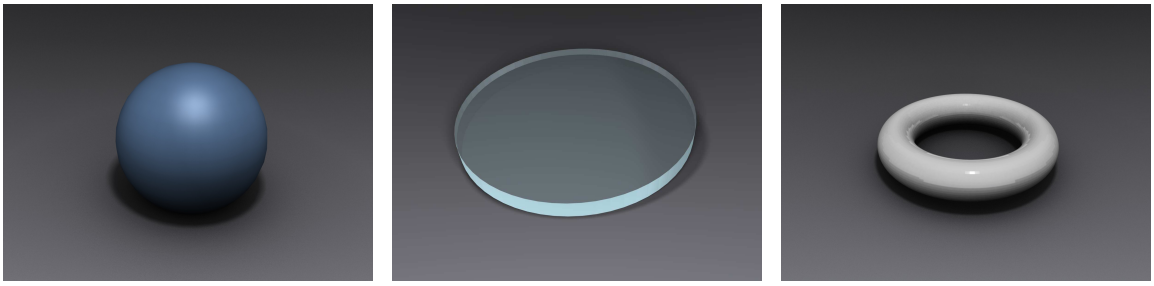


Figure 2.7: Resonance mode extraction results for **metallic** material, as shown in Fig. 2.2e. For example, GREEN object in this figure represents the object labled GREEN in Fig. 2.2e. Fig. 2.7a - Fig. 2.7l show, for each object, the power spectrograms of the recorded audio, the extracted resonance mode mixed audio, and the absolute error between the two.



(a) A wooden stick (b) A wooden cube (c) A metallic bunny



(d) A plastic sphere (e) A glass plate (f) A porcelain torus

Figure 2.8: Various **shapes** and **materials** used in the material similarity perceptual study described in Sec. 2.4. Six representative shapes: stick, cube, bunny, sphere, plate, and torus; five synthetic materials modeled with Rayleigh damping: metal, wood, glass, plastic, and porcelain.

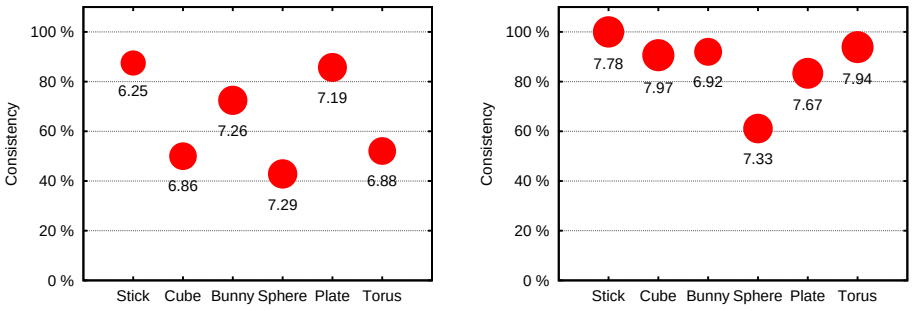
Figure 2.9: Three different **sizes** (1x, 2x, and 4x) for each shape, as shown on the metallic bunny example in this figure.

Shape/Size IDs:						
Shape:	1. bunny	2. cube	3. plate	4. sphere	5. stick	6. torus
size:	1. small	2. medium	3. large			

Color Codes:	
Group 1	Black
Group 2	Red
Group 3	Green

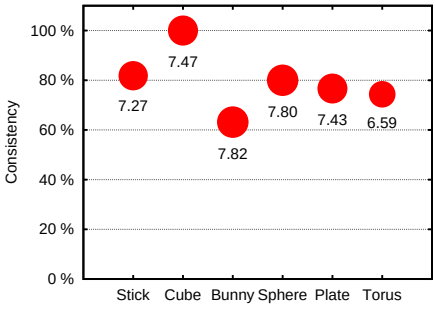
	(1, 1)	(1, 2)	(1, 3)	(2, 1)	(2, 2)	(2, 3)	(3, 1)	(3, 2)	(3, 3)	(4, 1)	(4, 2)	(4, 3)	(5, 1)	(5, 2)	(5, 3)	(6, 1)	(6, 2)	(6, 3)	row sum
(1, 1)																			0
(1, 2)																			0
(1, 3)																			0
(2, 1)		1																	1
(2, 2)	1		1																2
(2, 3)	1																		1
(3, 1)		1				1													2
(3, 2)																			0
(3, 3)	1			1															2
(4, 1)		1			1			1	1										4
(4, 2)	1		1	1		1	1												5
(4, 3)		1		1	1		1	1											5
(5, 1)			1					1	1		1								4
(5, 2)			1			1	1		1	1									5
(5, 3)	1	1		1	1			1											5
(6, 1)			1		1	1		1	1			1							6
(6, 2)	1		1			1				1			1		1				6
(6, 3)		1		1			1	1					1	1					6
column sum	6	6	6	5	4	5	4	6	4	2	1	1	2	1	1	0	0	0	
row + column sum	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	
sample sum	54																		

Figure 2.10: Sampling schemes for subgroups: The number 1 marked in the spreadsheet cells indicates a selected combination. The shape and size IDs are listed in the top left corner of this table. The three different colors represent three different sampling subgroups. In each subgroup, each geometry of a distinctive size and shape is selected exactly twice, and all the combination pairs are different in both size and shape among the selected two geometry instances. Notice the combined three subgroups appear to randomly sample all possible pairings.

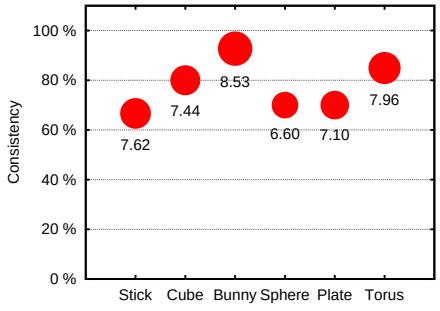


(a) Wood

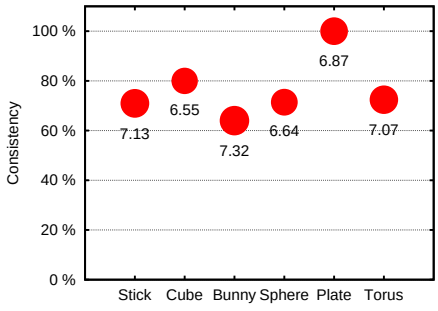
(b) Plastic



(c) Porcelain

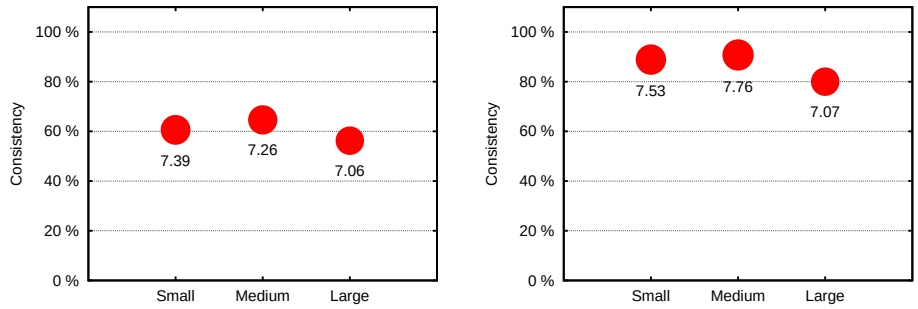


(d) Metal



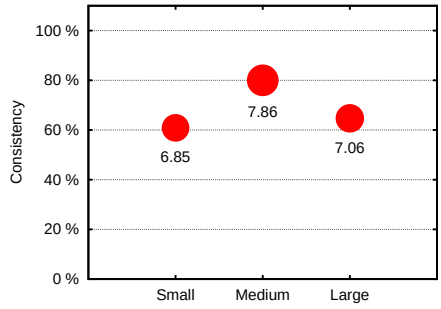
(e) Glass

Figure 2.11: Consistency and confidence levels for synthetic audio trials across shapes for all materials. The radii of the disks represent confidence levels, which are also shown as the numbers below the disks.

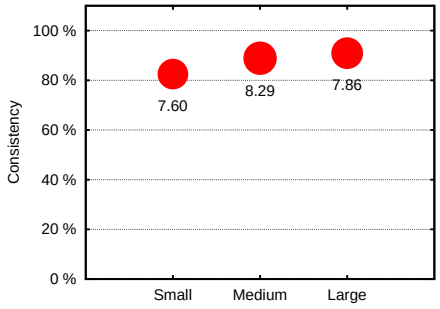


(a) Wood

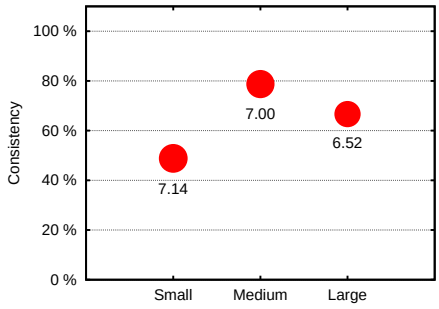
(b) Plastic



(c) Porcelain



(d) Metal



(e) Glass

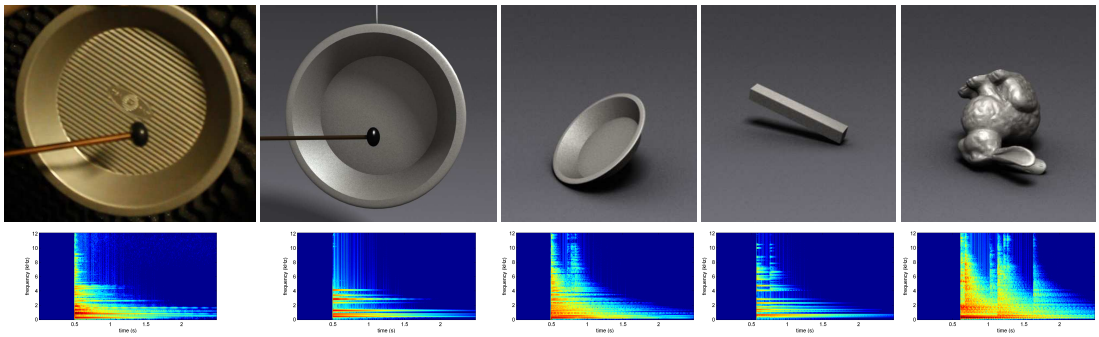
Figure 2.12: Consistency and confidence levels for synthetic audio trials across sizes for all materials. The radii of the disks represent confidence levels, which are also shown as the numbers below the disks.

CHAPTER 3: EXAMPLE-GUIDED PHYSICALLY BASED SOUND SYNTHESIS

3.1 Introduction

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

The traditional sound effect production for video games, animation, and movies is a laborious practice. Talented Foley artists are normally employed to record a large number of sound samples in advance and manually edit and synchronize the recorded sounds to a visual scene. This approach generally achieves satisfactory results. However, it is labor-intensive and cannot be applied to all



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

Figure 3.1: From the recording of a real-world object (a), our framework is able to find the material parameters and generates similar sound for a replicate object (b). The same set of parameters can be transfered to various virtual objects to produce sounds with the same material quality ((c), (d), (e)).

interactive applications. It is still challenging, if not infeasible, to produce sound effects that precisely capture complex interactions that cannot be predicted in advance.

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

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

As a result, for objects with different geometries and run-time interactions, different sets of modes are generated or excited differently, and different sounds are produced. However, if the material properties are the same, they should all sound like coming from the same material. For example, a plastic plate being hit, a plastic ball being dropped, and a plastic box sliding on the floor generate different sounds, but they all sound like 'plastic', as they have the same material properties. Therefore, if I can deduce the material properties from a recorded sound and *transfer* them to different objects with rich interactions, the *intrinsic quality* of the original sounding material is preserved. Our method can also compensate the differences between the example audio and the modal-synthesized sound. Both the material parameters and the residual compensation are capable of being transfered to virtual objects of varying sizes and shapes and capture all forms of interactions. Fig. 3.1 shows an example of our framework. From one recorded impact sound (Fig. 3.1a), I estimated material

parameters, which can be directly applied to various geometries (Fig. 3.1c, 3.1d, 3.1e) to generate audio effects that automatically reflect the shape variation while still preserve the same sense of material. Fig. 3.2 depicts the pipeline of our approach, and its various stages are explained below.

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

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

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

Our key contributions are summarized below:

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

- An effective residual compensation method that accounts for the difference between the real-world recording and the modal-synthesized sound.
- A general framework for synthesizing rigid-body sounds that closely resemble recorded example materials.
- Automatic transfer of material parameters and residual compensation to different geometries and runtime dynamics, producing realistic sounds that vary accordingly.

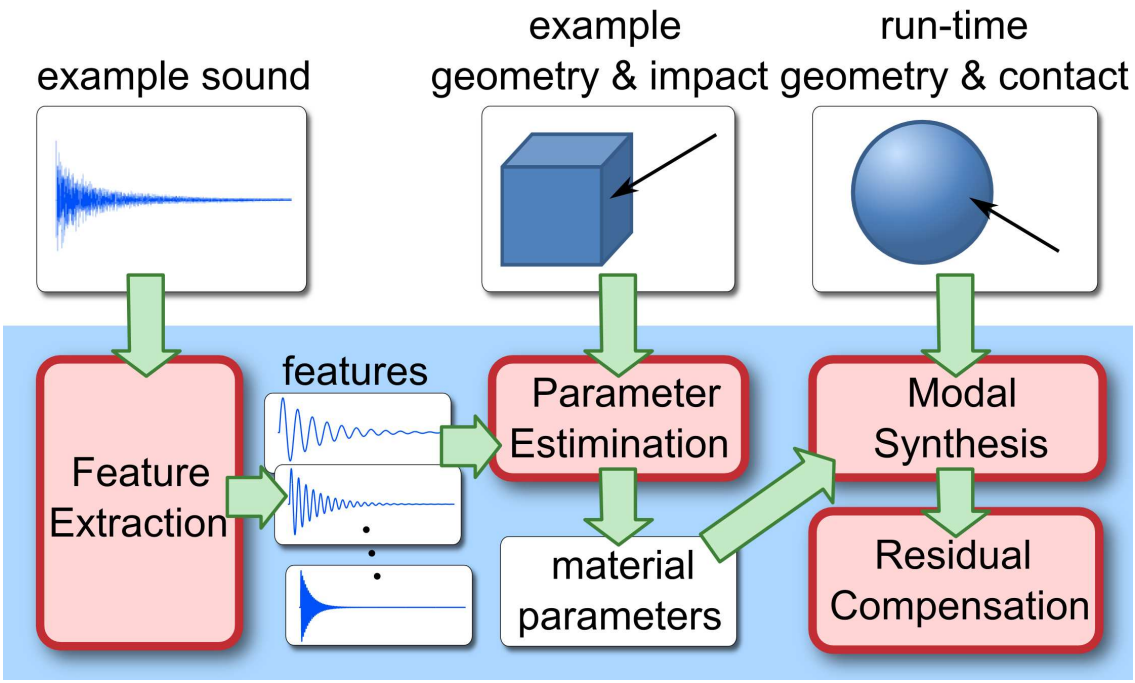


Figure 3.2: Overview of the example-guided sound synthesis framework (shown in the blue block): Given an example audio clip as input, features are extracted. They are then used to search for the optimal material parameters based on a perceptually inspired metric. A residual between the recorded audio and the modal synthesis sound is calculated. At run-time, the excitation is observed for the modes. Corresponding rigid-body sounds that have a similar audio quality as the original sounding materials can be automatically synthesized. A modified residual is added to generate a more realistic final sound.

3.2 Related Work

Spring-mass (Raghuvanshi and Lin, 2006b) and finite element (O'Brien et al., 2002b) representations have been used to calculate the modal model of arbitrary shapes. Challenges lie in how to choose the material parameters used in these representations. Pai et al. (2001) and Corbett et al.

(2007) directly acquires a modal model by estimating modal parameters (i.e. amplitudes, frequencies, and dampings) from measured impact sound data. A robotic device is used to apply impulses on a real object at a large number of sample points, and the resulting impact sounds are analyzed for modal parameter estimation. This method is capable of constructing a virtual sounding object that faithfully recreates the audible resonance of its measured real-world counterpart. However, each new virtual geometry would require a new measuring process performed on a real object that has exactly the same shape, and it can become prohibitively expensive with an increasing number of objects in a scene. This approach generally extracts hundreds of parameters for one object from many audio clips, while the goal of our technique instead is to estimate the few parameters that best represent one *material* of a sounding object from only *one* audio clip.

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

In computer music and acoustic communities, researchers proposed methods to calibrate physically based virtual musical instruments. For example, Välimäki et al. (1996); Välimäki and Tolonen (1997) proposed a physical model for simulating plucked string instruments. They presented a parameter calibration framework that detects pitches and damping rates from recorded instrument

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

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

3.3 Background

Modal Sound Synthesis: The standard linear modal synthesis technique (Shabana, 1997) is frequently used for modeling of dynamic deformation and physically based sound synthesis. I adopt tetrahedral finite element models to represent any given geometry (O'Brien et al., 2002b). The

displacements, $\mathbf{x} \in \mathbb{R}^{3N}$, in such a system can be calculated with the following linear deformation equation:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{f}, \tag{3.1}$$

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

$$\mathbf{K}\mathbf{U} = \lambda\mathbf{M}\mathbf{U}, \tag{3.2}$$

the system can be decoupled into the following form:

$$\ddot{\mathbf{q}} + (\alpha\mathbf{I} + \beta\lambda)\dot{\mathbf{q}} + \lambda\mathbf{q} = \mathbf{U}^T\mathbf{f}, \tag{3.3}$$

where λ is a diagonal matrix, containing the eigenvalues of Eqn. 3.2; $\mathbf{U}$ is the eigenvector matrix, and transforms $\mathbf{x}$ to the decoupled deformation bases $\mathbf{q}$ with $\mathbf{x} = \mathbf{U}\mathbf{q}$.

The solution to this decoupled system, Eqn. 3.3, are a bank of *modes*, i.e. damped sinusoidal waves. The i'th mode looks like:

$$q_i = a_i e^{-d_i t} \sin(2\pi f_i t + \theta_i), \tag{3.4}$$

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

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

$$\phi_i = (f_i, d_i, a_i) \tag{3.5}$$

and will be used throughout the rest of the chapter. I ignore θ_i in Eqn. 3.4 because it can be safely assumed as zero in our estimation process, where the object is initially at rest and struck at $t = 0$. f and ω are used interchangeably to represent frequency, where $\omega = 2\pi f$.

Material properties: The values in Eqn. 3.4 depend on the material properties, the geometry, and the run-time interactions: a_i and θ_i depend on the run-time excitation of the object, while f_i and d_i depend on the geometry and the material properties as shown below. Solving Eqn. 3.3, I get

$$d_i = \frac{1}{2}(\alpha + \beta\lambda_i), \tag{3.6}$$

$$f_i = \frac{1}{2\pi}\sqrt{\lambda_i - \left(\frac{\alpha + \beta\lambda_i}{2}\right)^2}. \tag{3.7}$$

I assume the Rayleigh damping coefficients, α and β, can be transfered to another object with no drastic shape or size change. Empirical and psychoacoustic experiments were carried out to support this assumption. Please refer to Chapter 3 for more detail. The eigenvalues λ_i's are calculated from **M** and **K** and determined by the geometry and tetrahedralization as well as the material properties: in our tetrahedral finite element model, **M** and **K** depend on mass density ρ, Young's modulus E, and Poisson's ratio ν, if I assume the material is *isotropic* and *homogeneous*.

Constraint for modes: I observe modes in the adopted linear modal synthesis model have to obey some constraint due to its formulation. Because of the Rayleigh damping model I adopted, all estimated modes lie on a circle in the (ω, d)-space, characterized by α and β. This can be shown as follows. Rearranging Eqn. 3.6 and Eqn. 3.7 as

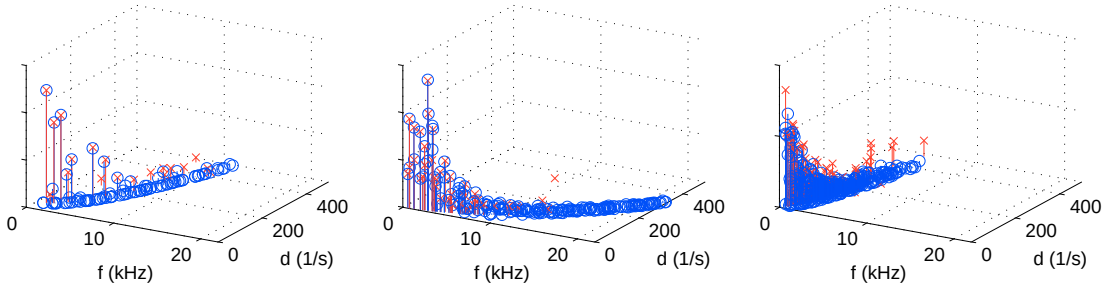
$$\omega_i{}^2 + \left(d_i - \frac{1}{\beta}\right)^2 = \left(\frac{1}{\beta}\sqrt{1 - \alpha\beta}\right)^2 \tag{3.8}$$

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

3.4 Results and Analysis

Parameter estimation: Before working on real-world recordings, I design an experiment to evaluate the effectiveness of our parameter estimation with synthetic sound clips. A virtual object with known

material parameters $\{\alpha, \beta, \gamma, \sigma\}$ and geometry is struck, and a sound clip is synthesized by mixing the excited modes. The sound clip is entered to the parameter estimation pipeline to test if the same parameters are recovered. Three sets of parameters are tested and the results are shown in Fig.3.3.



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

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

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

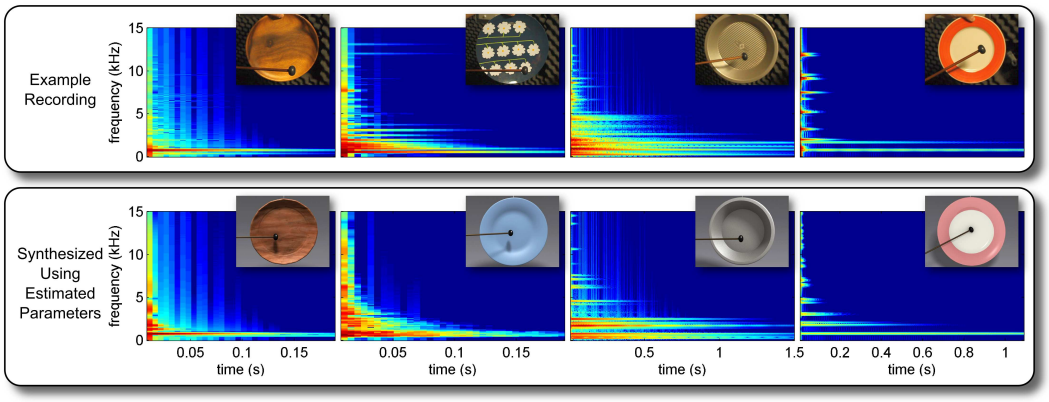
Figure 3.3: Results of estimating material parameters using synthetic sound clips. The intermediate results of the feature extraction step are visualized in the plots. Each blue circle represents a synthesized feature, whose coordinates (x, y, z) denote the frequency, damping, and energy of the mode. The red crosses represent the extracted features. The tables show the truth value, estimated value, and relative error for each of the parameters.

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

I estimate the material parameters from various real-world audio recordings: a wood plate, a plastic plate, a metal plate, a porcelain plate, and a glass bowl. For each recording, the parameters are estimated using a virtual object that is of the same size and shape as the one used to record the audio clips. When the virtual object is hit at the same location as the real-world object, it produces a sound similar to the recorded audio, as shown in Fig. 3.4 and the supplementary video.

Fig. 3.5 compares the refenece features of the real-world objects and the estimated features of the virtual objects as a result of the parameter estimation. The parameter estimated for these materials are shown in Table. 3.1.

Refer to Sec. 3.3 for the definition and estimation of these parameters.



(a) *wood plate* (b) *plastic plate* (c) *metal plate* (d) *porcelain plate*

Figure 3.4: Parameter estimation for different materials. For each material, the material parameters are estimated using an example recorded audio (top row). Applying the estimated parameters to a virtual object with the same geometry as the real object used in recording the audio will produce a similar sound (bottom row).

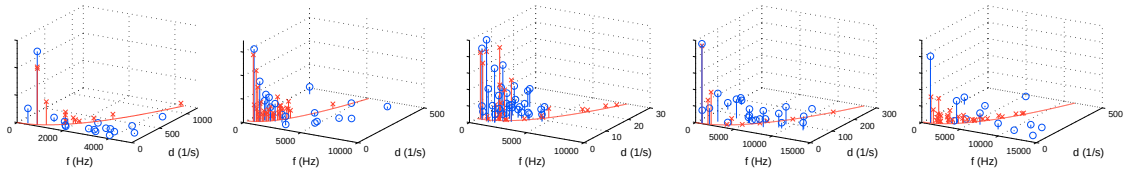
Table 3.1: Estimated parameters

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

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

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

Fig. 3.6 gives an example of this transferring process. From an example recording of a porcelain plate (a), the parameters for the porcelain material are estimated, and the residual computed (b). The parameters and residual are then transfered to a smaller porcelain plate (c) and a porcelain bunny (d).



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

Figure 3.5: Feature comparison of real and virtual objects. The blue circles represent the reference features extracted from the recordings of the real objects. The red crosses are the features of the virtual objects using the estimated parameters. Because of the Rayleigh damping model, all the features of a virtual object lie on the depicted red curve on the (f, d)-plane.

Comparison with real recordings: Fig. 3.7 shows a comparison of the transferred results with the real recordings. From a recording of glass bowl, the parameters for glass are estimated (column (a)) and transfered to other virtual glass bowls of different sizes. The synthesized sounds ((b) (c) (d), bottom row) are compared with the real-world audio for these different-sized glass bowls ((b) (c) (d), top row). It can be seen that although the transfered sounds are not identical to the recorded ones, the overall trend in variation is similar. Moreover, the perception of material is preserved, as can be verified in the accompanying video. More examples of transferring the material parameters as well as the residuals are demonstrated in the accompanying video.

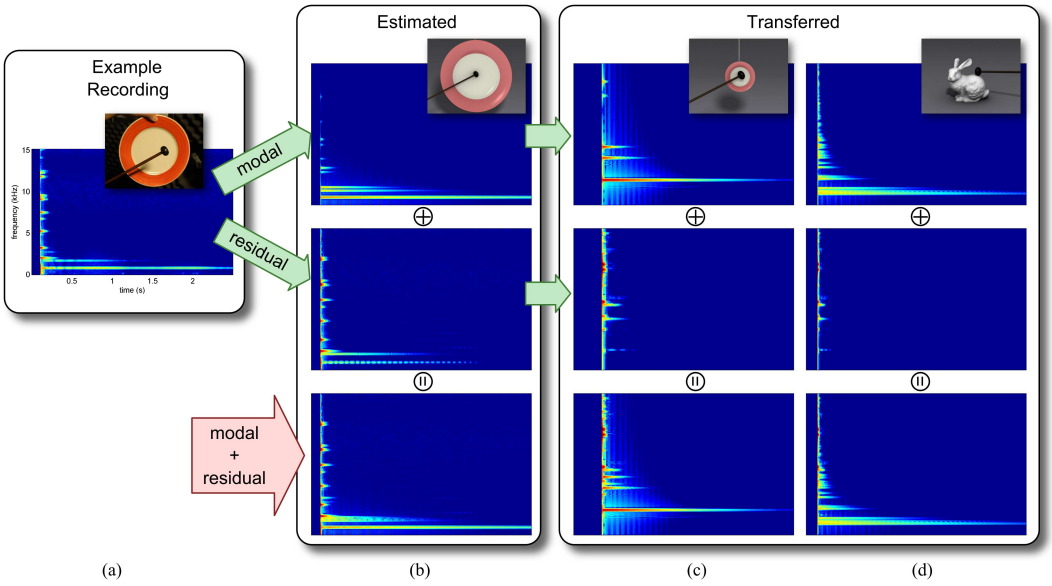


Figure 3.6: Transfered material parameters and residual: from a real-world recording (a), the material parameters are estimated and the residual computed (b). The parameters and residual can then be applied to various objects made of the same material, including (c) a smaller object with similar shape; (d) an object with different geometry. The transfered modes and residuals are combined to form the final results (bottom row).

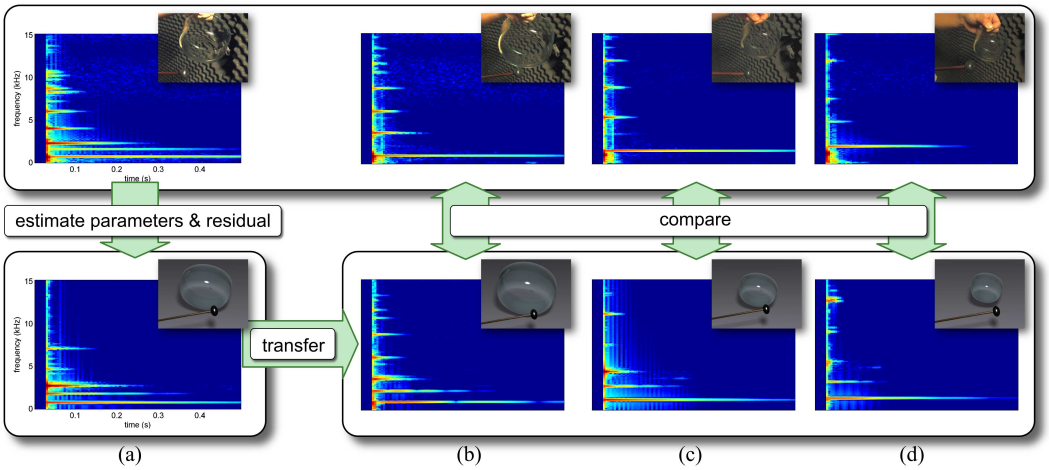


Figure 3.7: Comparison of transfered results with real-word recordings: from one recording (column (a), top), the optimal parameters and residual are estimated, and a similar sound is reproduced (column (a), bottom). The parameters and residual can then be applied to different objects of the same material ((b), (c), (d), bottom), and the results are comparable to the real-world recordings ((b), (c), (d), top).

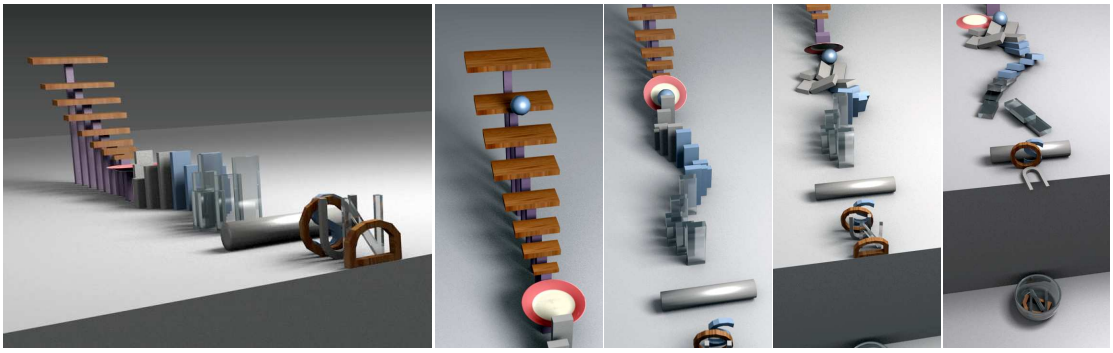


Figure 3.8: The estimated parameters are applied to virtual objects of various sizes and shapes, generating sounds corresponding to all kinds of interactions such as colliding, rolling, and sliding.

Example: a complicated scenario I applied the estimated parameters for various virtual objects in a scenario where complex interactions take place, as shown in Fig. 3.8 and the accompanying video.

Performance: Table 3.2 shows the timing for our system running on a single core of a 2.80 GHz Intel Xeon X5560 machine. It should be noted that the parameter estimation is an offline process: it needs to be run only once per material, and the result can be stored in a database for future reuse.

For each material in column one, multiple starting points are generated, and the numbers of starting points are shown in column two. From each of these starting points, the optimization process runs for an average number of iterations (column three) until convergence. The average time taken for the process to converge is shown in column four. The convergence is defined as when both the

Table 3.2: Offline Computation for Material Parameter Estimation

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

step size and the difference in metric value are lower than their respective tolerance values, Δ_x and Δ_{metric}. The numbers reported in Table 3.2 are measured with Δ_x = 1e-4 and Δ_{metric} = 1e-8.

3.5 Perceptual Study

To assess the effectiveness of our parameter estimation algorithm, I designed an experiment to evaluate the auditory perception of the synthesized sounds of five different materials. Each subject is presented with a series of 24 audio clips with no visual image or graphical animation. Among them, 8 are audio recordings of sound generated from hitting a real-world object, and 16 are synthesized using the techniques described in this chapter. For each audio clip, the subject is asked to identify among a set of 5 choices (wood, plastic, metal, porcelain, and glass), from which the sound came. A total of 53 subjects (35 women and 18 men), from age of 22 to 71, participated in this study. The 8 real objects are: a wood plate, a plastic plate, a metal plate, a porcelain plate, and four glass bowls with different sizes. The 16 virtual objects are: three different shapes (a plate, a stick, and a bunny) for each of these four materials: wood, plastic, metal, and porcelain, plus four glass bowls with different sizes.

I show the cumulative recognition rates of the sounding materials in two separate matrices: Table 3.3 presents the recognition rates of sounds from real-world materials, and Table 3.4 reflects the recognition rates of sounds from synthesized virtual materials. The numbers are normalized with the number of subjects answering the questions. For example, Row 3 of Table 3.3 means that for a given *real-world* sound recorded from hitting a metal object, none of the subjects thought it came from wood or plastic, 66.1% of them thought it came from metal, 9.7% of them thought it came from porcelain and 24.2% of them thought it came from glass. Correspondingly, Row 3 of

Table 3.3: Material Recognition Rate Matrix: Recorded Sounds

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

Table 3.4: Material Recognition Rate Matrix: Synthesized Sounds Using Our Method

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

Table 3.4 shows that for a sound *synthesized* with our estimated parameters for metal, the percentage of subjects thinking that it came from wood, plastic, metal, porcelain or glass respectively.

I found that the successful recognition rate of virtual materials using our synthesized sounds compares favorably to the recognition rate of real materials using recorded sounds. The difference of the recognition rates (recorded minus synthesized) is close to zero for most of the materials, with 95% confidence intervals shown in Table 3.5. A confidence interval covering zero means that the difference in recognition rate is not *statistically significant*. If both endpoints of a confidence interval are positive, the recognition rate of the real material is significantly higher than that of the virtual material; if both endpoints are negative, the recognition rate of the real material is significantly lower.

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

Table 3.5: 95% Confidence Interval of Difference in Recognition Rates

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

3.6 Conclusion and Future Work

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

Although our experiments show successful results in estimating the material parameters and computing the residuals, it has some limitations. Our model assumes linear deformation and Rayleigh damping. While offering computational efficiency, these models cannot always capture all sound phenomena that real world materials demonstrate. Therefore, it is practically impossible for the modal synthesis sounds generated with our estimated material parameters to sound exactly the same as the real-world recording. Our feature extraction and parameter estimation depend on the assumption that the modes do not couple with one another. Although it holds for the objects in our experiments, it may fail when recording from objects of other shapes, e.g. thin shells where nonliear models would be more appropriate (Chadwick et al., 2009).

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

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

requirements regarding the geometry of the virtual object may affect the accuracy of the results using this method.

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

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

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

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

When transferring residual computed from impacts to continuous contacts (e.g. sliding and rolling), there are certain issues to be considered. Several previous work have approximated continuous contacts with a series of impacts and have generated plausible *modal* sounds. Under this

approximation, our proposed feature-guided residual transfer technique can be readily adopted. However, the effectiveness of this direct mapping needs further evaluation. Moreover, future study on continuous contact sound may lead to an improved modal synthesis model different than the impact-based approximation, under which our residual transfer may not be applicable. It is then also necessary to reconsider how to compensate the difference between a real continuous contact sound and the modal synthesis sound.

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

In summary, data-driven approaches have proven useful in areas in computer graphics, including rendering, lighting, character animation, and dynamics simulation. With promising results that are transferable to virtual objects of different geometry, sizes, and interactions (e.g. (Ren et al., 2012a)), this work is the first rigorous treatment of the problem on automatically determining the material parameters for physically based sound synthesis using a single sound recording, and it offers a new direction for combining example-guided and modal-based approaches.

CHAPTER 4: SYNTHESIZING CONTACT SOUNDS OF TEXTURED MODELS

4.1 Modal Analysis for Sound Synthesis

Sound is a traveling wave produced by the variation of medium pressure, which is caused by the vibration of objects. The pressure oscillation at frequencies between about 20 and 20K Hz can be heard by human auditory systems. To simulate the physical process of sound generation, I need to model the mechanical vibration of sounding objects. This vibration may not be visually noticeable, but can make a considerable difference to human ears.

Many recent real-time physics-based sound synthesis methods adopt the modal synthesis approach for discretely approximating the vibration of sounding objects (O'Brien et al., 2002c; Raghuvanshi and Lin, 2006a; van den Doel et al., 2001b; van den Doel and Pai, 1998b). The complete process is composed of two stages: modal analysis (in pre-processing) and modal synthesis (during run-time). Modal analysis represents the vibration of an arbitrarily shaped object with a bank of damped harmonic oscillators. In this process, the amplitude, damping, and decay coefficients are extracted from the mesh geometry for each sounding object. The next process, modal synthesis, approximates the vibration caused by external force applied on the object using a linear combination of the damped oscillators determined by modal analysis. Next, I briefly introduces modal analysis and synthesis, as well as its application in our sound synthesis system.

4.1.1 Modal Analysis

Each sounding object can be viewed as a continuous system. To represent its vibration for sound synthesis, model. Different discretization approaches can be adopted for models of different shapes to obtain the parameters for the modal representation. Modal analysis (Shabana, 1997) is a well-known technique in computational mechanics for modeling the structural vibration of objects and I adopt this technique to model the surface vibration leading to sound generation. For some simple shapes, first principles can be used to solve for the parameters (van den Doel and Pai, 1998b).

For an arbitrary shape, finite element methods (FEM) can be used to discretize the objects (O'Brien et al., 2002c). The physics properties of this geometry can also be modeled with a spring-mass system (Raghuvanshi and Lin, 2006a). Finally, the parameters can be fitted to recordings of real objects (Pai et al., 2001).

In our sound synthesis system, I adopt the mass-spring representation for modal analysis. This representation is less accurate compared to FEM, but it is much faster. Therefore, it is more suitable for real-time VR applications, because the materials and shapes of the objects can be changed on the fly. In the mass-spring representation, each vertex of the input triangle mesh is considered as a particle mass, and each edge between two vertices is considered as a damped spring. Different parameters used in the mass-spring system construction creates different modal models (i.e. frequencies, damping, and mode shapes) that sound like different materials. I refer the readers to (Raghuvanshi and Lin, 2006a) for more details on the input processing.

The mass-spring system created from the input mesh forms an ordinary equation (ODE) system as below:

$$M\frac{d^2r}{dt^2} + C\frac{dr}{dt} + Kr = f \tag{4.1}$$

where M, C, and K are respectively the mass, damping, and stiffness matrix. If there are N vertices in the triangle mesh, r in Eqn. 4.1 is a vector of dimension N, and it represents the displacement of each mass particle from its rest position. Each diagonal element in M represents the mass of each particle. In our implementation, C adopts Rayleigh damping approximation, so it is a linear combination of M and K. The element at row i and column j in K represents the spring constant between particle i and particle j. f is the external force vector. The resulting ODE system turns into:

$$M\frac{d^2r}{dt^2} + (\gamma M + \eta K)\frac{dr}{dt} + Kr = f \tag{4.2}$$

where M is diagonal, and K is real symmetric. Therefore, Eqn. 4.2 can be simplified into a decoupled system after diagonalizing K with $K = GDG^{-1}$, where D is a diagonal matrix containing the eigenvalues of K. The diagonal ODE system that I eventually need to solve is:

$$M\frac{d^2z}{dt^2} + (\gamma M + \eta D)\frac{dz}{dt} + Dz = G^{-1}f \tag{4.3}$$

where $z = G^{-1}r$, a linear combination of the original vertex displacement. The general solution to Eqn. 4.3 is:

$$z_i(t) = c_i e^{\omega_i^+ t} + \bar{c}_i e^{\omega_i^- t}$$

$$\omega_i^{\pm} = \frac{-(\gamma\lambda_i + \eta) \pm \sqrt{(\gamma\lambda_i + \eta)^2 - 4\lambda_i}}{2} \tag{4.4}$$

where λ_i is the i'th eigenvalue of D. With particular initial conditions, I can solve for the coefficient c_i and its complex conjugate, $\bar{c}_i$. Therefore, the vibration of the original triangle mesh is now approximated with the linear combination of the mode shapes z_i.

4.1.2 Impulse Response and Modal Synthesis

When an object experiences a sudden external force f that lasts for a duration of time, Δt, I say that there is an *impulse* $f\Delta t$ applied to the object. This impulse either causes a resting object to oscillate, or changes the way it oscillates, I say that the impulse *excites* the oscillation. Mathematically, since the right-hand side of Eqn. 4.3 changes, the solution of coefficients c_i and $\bar{c}_i$ also changes in response, which is called the *impulse response* of the model.

The impulse response, or the update of c_i and $\bar{c}_i$, for an impulse $f\Delta t$ follows the rule (Raghuvanshi and Lin, 2006a):

$$c_{i,t_0+\Delta t} = c_{i,t_0} e^{\omega^+ t_0} + \frac{g_i}{m_i(\omega_i^+ + \omega_i^-)}$$

$$\bar{c}_{i,t_0+\Delta t} = \bar{c}_{i,t_0} e^{\omega^+ t_0} - \frac{g_i}{m_i(\omega_i^+ + \omega_i^-)} \tag{4.5}$$

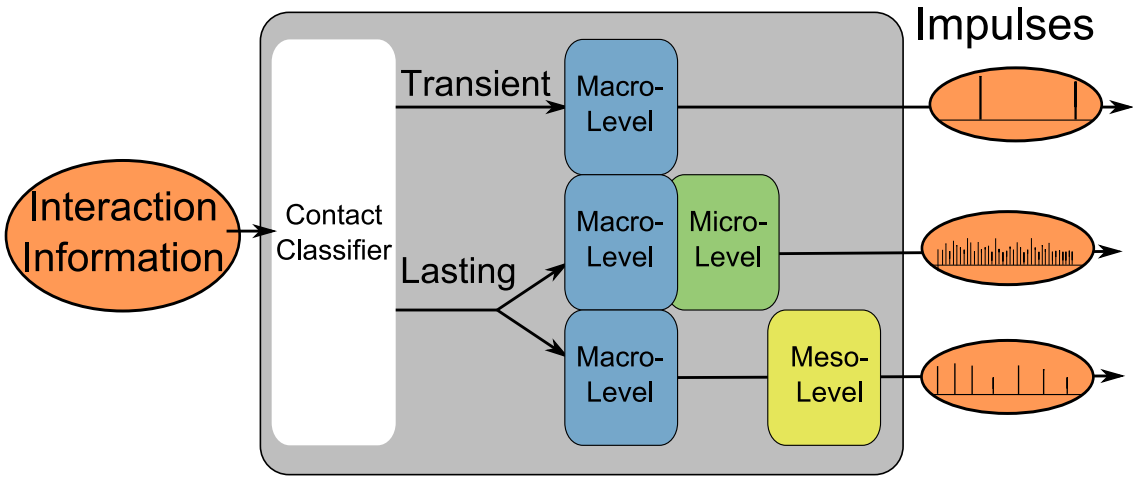


Figure 4.1: **Interaction Handling:** Given contact information, this module will classify the type of contacts based on velocity and contact normals. It then uses the three-level surface representation for contact handling to generate impulses that drive the sound synthesis module.

where g_i is the i'th element in vector $G^{-1}f$. Whenever there is an impulse acting on an object, I can quickly compute the approximated displacement of the mesh representing the object at any time instance onwards by plugging Eqn. 4.5 to Eqn. 4.4.

4.2 Interaction Handling

In the previous section I have discussed how to generate sounds, once the impulses applied to the object are given. In this section I will explain how to actually produce these impulses from the complex interactions that take place in the VE application. Due to performance constraints of real-time sound synthesis, these impulses approximate the complex interactions but still retain the characteristics.

I present a novel *three-level interaction handling* approach that models various interactions. The pipeline of this approach is shown in Figure 4.1. The approach requires first categorizing the interaction among objects into *lasting contact* and *transient contact*. These contacts are then handled by three-level surface representation for contact handling to generate sound. Sounds generated using this representation have contributions from different levels of surface details for different types of contacts: transient contacts can be sufficiently handled using the macro-level geometric representation alone, while lasting contacts are handled using all three-levels of surface representation. The micro-level geometry aims at simulating the friction interaction at audio sampling rate and provides the overall roughness of the contacting material. The meso-level representation provides the

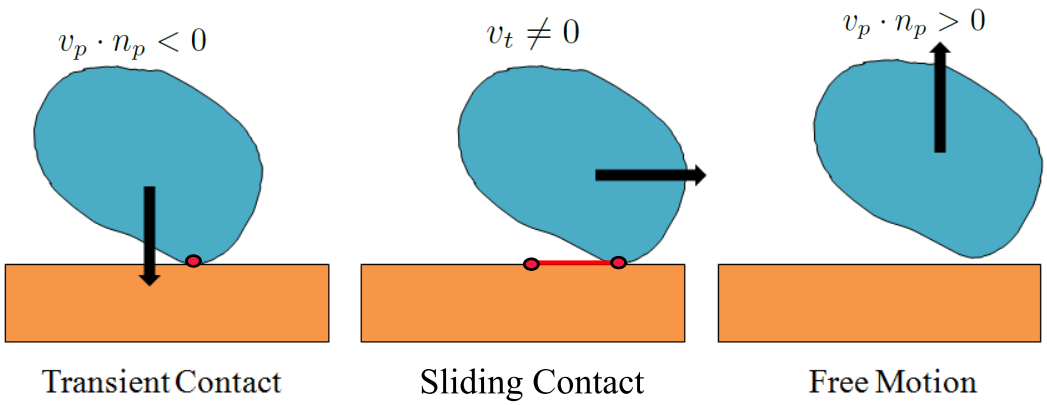


Figure 4.2: **Different Contact States.** The arrows indicates the linear velocity of the object. The dots indicate the contact point, and the line between them indicates the contact area.

variation of sound caused by the bumpiness of the material that is typically encoded in some forms of texture maps for visual rendering. The ridges and troughs at this level are both visible from the screen and perceivable from the synthesized sound using our new representation for contact-handling. The macro-level simulation is updated at the physics engine's time step, so it can provide the shape and contact information on the scale that the rigid-body simulator can handle. The three-level representation for simulating contact sounds is illustrated in Figure 4.3 and I will elaborate it next.

4.2.1 Contact Categorization

I adopt the state and event computation from the event-based approach developed by Sreng et al. (2007) to identify and categorize contacts, using the position, velocity and geometry information of the objects.

Two objects are said to be *contacting* if their models overlap in space at a certain point p, and if $\mathbf{v_p} \cdot \mathbf{n_p} < 0$, where $\mathbf{v_p}$ and $\mathbf{n_p}$ are their relative velocity and contact normal at point p.

Two contacting objects are said to be in *lasting contact* if $\mathbf{v_t} \neq 0$, where v_t is their relative tangential velocity. Otherwise they are in *transient contact*. The process is illustrated in Figure 4.2.

Lasting Contacts: Sliding contacts are ubiquitous. When any two solid objects scrub against each other, there is a sliding contact. However, it is a very difficult task to simulate the micro-level collision of objects, which is essential for modeling the friction forces that actually excite the vibration of surface resonators during a sliding contact.

There are mainly two different approaches to simulate the friction interaction between two objects: physics-based and parametric models. Each has its strength and issues. The physics engine normally has a simulation rate on the order of 100Hz, which is much lower than the audio sampling rate (i.e. 44100Hz). If I choose to simulate the physics of friction faithfully, it would be impossible to achieve real-time simulation rate. In addition, it is infeasible to obtain the roughness geometry at such a micro level. The fractal noise friction model introduced by van den Doel et al. (2001b) is a good approximation of the friction force at the micro level. However, the method only emulates the micro-level interaction between materials that are visually smooth. Some intermediate-level details of the object cannot be simulated with only fractal noise excitation.

Transient (Impact and Rolling) Contacts: For simulating the impact and rolling sound, I adopt the method of Raghuvanshi and Lin (2006a). When the interaction handling module detects a transient contact, an impulse is added to the sound synthesis module. The magnitude of the impulse is modulated by the magnitude of the relative velocity between the two colliding objects. Rolling sound is generated by adding a sequence of impulses to the sound synthesis engine. This is feasible due to the geometry tessellations of models used in graphics applications. Normally, a number of discrete geometries are used to approximate the smooth curvature of objects. The rigid-body simulator automatically reports contacts between the tessellated geometries, and corresponding impulses are added to the sound synthesis module.

4.2.2 Three-Level Surface Representation

In this section, I describe our novel three-level representation for contact handling to synthesize sounds.

The Macro Level: Geometry

The macro shapes are represented by the input triangle meshes of objects. These macro-level geometries are used for handling collision and computing forces in the rigid body simulator.

The Micro Level: Friction

The roughness of the contacting material is reflected by the micro-level simulation of friction sound. I use the method proposed by Van den van den Doel et al. (2001b) to generate an approximated force profile at this fine level. A fractal noise is used as the force profile, and the spectrum of the

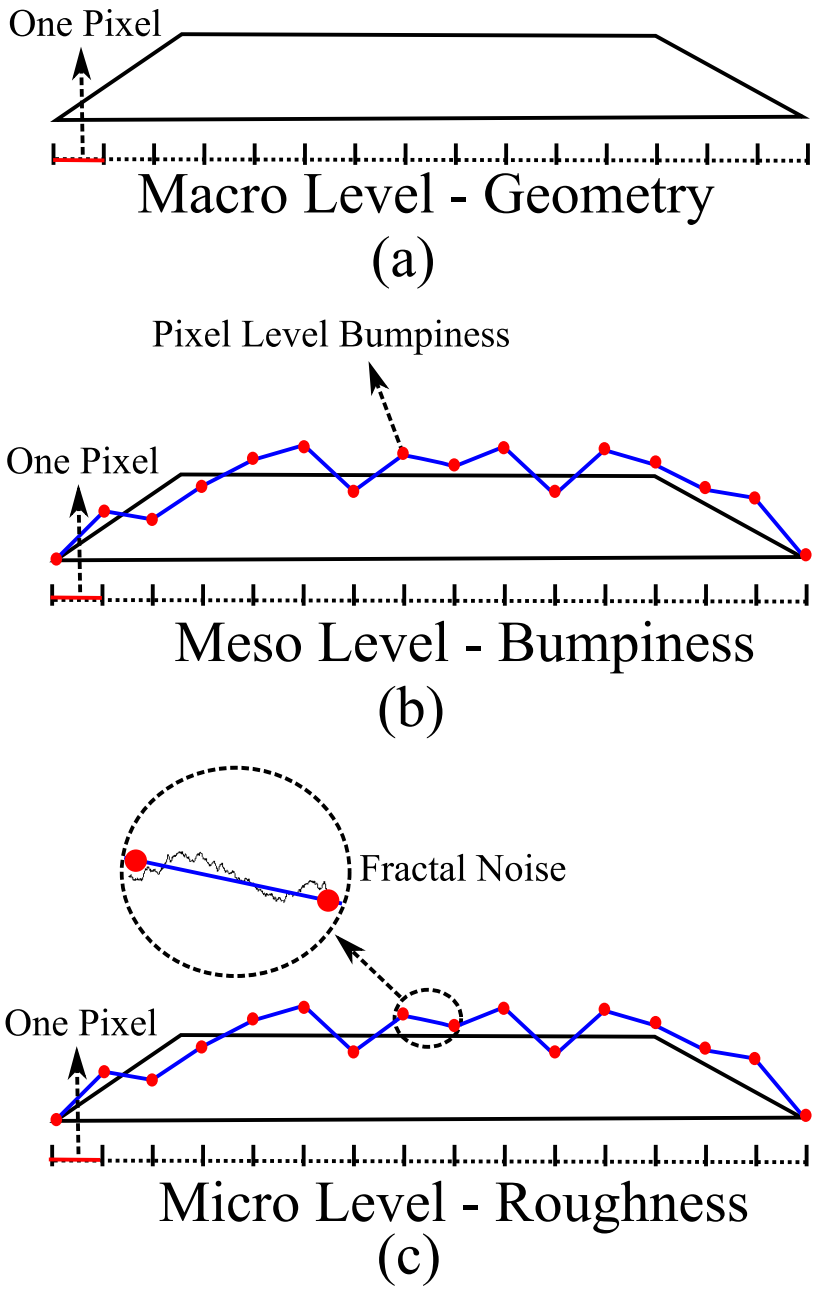


Figure 4.3: **The Three-level Contact Surface Representation.** (a) The trapezoid conceptualizes the geometry of the object. (b) The wiggly curve represents the surface of the geometry after the surface normals being changed by a normal map. (c) Within one pixel, the roughness of the surface is represented by a fractal noise. The geometry, bumpiness, and roughness models all contribute to various levels of frictional interaction.

fractal noise varies with the auditory roughness of the material. The force profile is stored in a wave-table and played back to give users the sound that varies at audio sampling rate. The wave-table play-back speed is governed by the contact speed to give users the feeling of scratching through the grainy material fast or slowly. The magnitude of the impulse added also linearly varies with the normal force between the two objects scrubbing against each other. In summary, this parametric model reflects the contact speed, contact normal force, and the roughness of the material at the micro level.

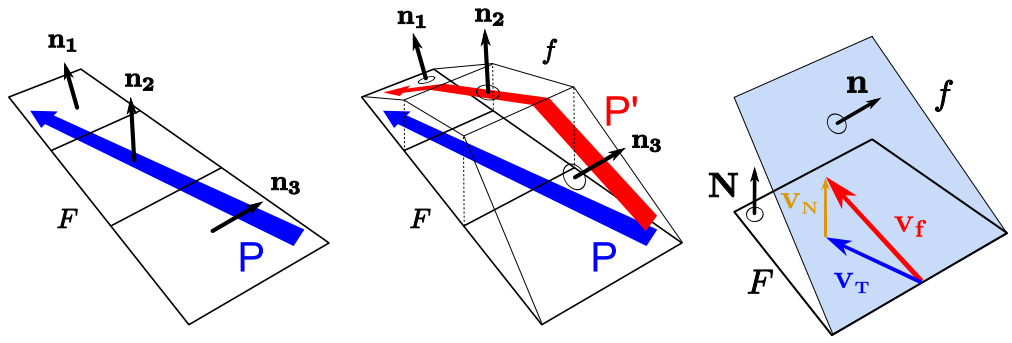
The Meso Level: Bumpiness

Solely using the micro-level force profile generated by a fractal noise to excite the resonators does not render any information for the bumpiness or *heterogeneous* variation of the contacting geometry at the meso level. Many graphics applications use bump mapping, normal mapping, and height mapping for rendering the complicated bumpiness of materials, using image-based representations. This level of details is clearly visible to the users but transparent to the rigid-body simulator; in contrast, the micro-level details are neither seen by the users nor by the physics engine.

Barrass and Adcock considered using bumpiness as a single surface-level granular synthesis to generate sound due to granular interaction (Barrass and Adcock, 2002). In contrast, our synthesis method takes the normal map from the visual rendering and considers this pixel-level information as small geometries.

Imagine an object in sliding contact with another object, whose surface F are shown in Figure 4.4a, the contact point traverses the path P within a time step. I look up the normal map associated to F and collect those normals around P. The normals suggest that the high resolution surface looks like f in Figure 4.4b, and that the contact point is expected to traverse a path P' on f. Therefore, besides the momentum along the tangential direction of F, the object must also have a time-varying momentum along the normal direction of F, namely, $\mathbf{p_N}$, where $\mathbf{N}$ is the normal vector of F. From simple geometry (Figure 4.4c), I compute its value

$$\mathbf{p_N} = m\mathbf{v_N} = mv_N\mathbf{N} = m\left(-\frac{\mathbf{v_T}\cdot\mathbf{n}}{\mathbf{N}\cdot\mathbf{n}}\right)\mathbf{N},$$



(a) (b) (c)

Figure 4.4: **Impulse Computation.** (a) The path P traced by an object sliding against another object within a time step, and the normals stored in the normal map around the path. The path lies on the surface F, which is represented coarsely with a low-resolution mesh (here a flat plane). (b) The normal map suggests that the high-resolution surface looks like f, and the object is expected to traverse the path P'. (c) The impulse along the normal direction can be recovered from the geometry configuration of $\mathbf{n}$, $\mathbf{N}$, and $\mathbf{V_T}$.

where m is the object's mass, $\mathbf{v_T}$ is the tangential velocity of the object relative to F. The impulse along the normal direction $\mathbf{J_N}$ that applies on the object is just the change of its normal momentum:

$$\mathbf{J_N} = \mathbf{p_N}(i) - \mathbf{p_N}(j),$$

when the object moves from pixel i to pixel j on the normal map. With this formulation, the impulses actually models the force applied by the bumps on the surface of one object to another, generating sound naturally correlated with the visual appearance of bumps from textures.

4.3 Implementation and Results

I have implemented the method described in this chapter using C++ and integrated it with OGRE3D, an open-source graphics rendering engine (Streeting et al., 2005).

4.3.1 User Interface

In designing the user interface to our sound synthesis system, I attempt to minimize the need for key-presses, mouse input, and any complex control that are required from non-technical users. Inspired by the intuitive user interface provided by the virtual painting system (Baxter et al., 2001), our system also takes user input from a Wacom Intuos tablet. Users can create sounds by simply

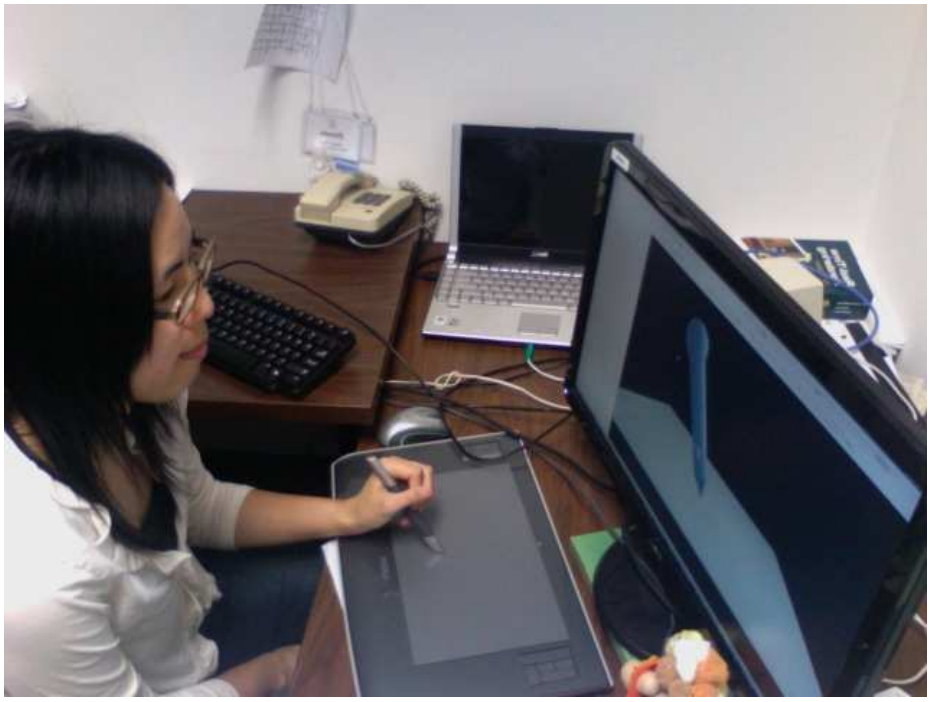


Figure 4.5: **The System Setup.** A user is synthesizing sound using a tablet connected to our sound rendering system by moving the stylus to interact with the virtual environment.

moving the stylus on the tablet with very minimal keyboard input. Figure 4.5 shows an user using the system to synthesize sound of a pen scrubbing against a surface. This simple interface allows users to intuitively interact with the virtual objects in the synthetic environment.

Users also have the flexibility to change the material parameters to design and synthesize the sounds that they desire to closely match the graphics rendering. By giving users the freedom to choose material parameters, I also introduce some difficulty in how to select the right parameters for some inexperienced users. I reduced this difficulty by providing the users a repository of materials. The sound synthesis parameters for many representative and normal materials in everyday life are given to the users. Based on these pre-selected material parameters, it should be much easier for users to *design* the material that sounds right to them. For now, I use trial-and-error method to find the parameters that generate the modal models that corresponds to the materials in our repository.

4.3.2 Results

In this section, I demonstrate some of the results produced by our sound synthesis system and enumerate its possible applications.

Surface Scrapping: This scene shows a user scrapping a pen on surfaces textured with normal maps, generating contact sounds that highly correlates with the visual cues. It also shows the ability to handle different materials. Since impulses are universally handled in our sound module, if I change the material property of the object, the change is automatically reflected in the resulting scrapping and impact sound.

In Figure 4.6, our method successfully captures the characteristics of the bumpiness. Scrapping on various surfaces using only fractal noises approximating frictional contact sounds is distinctively different (can also been seen in the wave plots) from scrapping textured surfaces using our sound synthesis method (please also view the accompanying video).

Virtual Instruments: Our system can be used to construct virtual instruments for education and entertainments. Users can build virtual instruments out of their designed sound by changing the material properties, and play them with our tablet user interface. With our interaction model, users are allowed to have complicated interaction with the instrument like scraping at various speed and tapping with different forces. Figure 4.7 shows a marimba-like virtual instrument with a user controlled mallet. Figure 4.7(b)-(d) show the different wave patterns generated by hitting the same bar made of different materials.

Add-on to Game Engines: Our sound synthesis system is able to synthesize sound from physics-based simulation in real time. This capability makes it a great add-on to applications like games, virtual environment and simulators. I integrated our sound synthesis system with a general graphics engine: Open Source 3D Graphics Engine (OGRE) (Streeting et al., 2005) and with a physics engine: NVidia's PhysX (NVIDIA, 2013). In the scenes shown in Figure 4.8 and Figure 4.9, I are able to easily achieve real-time performance with graphical rendering, physics simulation, and sound synthesis all running at the same time, which makes our approach a good candidate for sound synthesis in games.

Performance: In all the benchmarks mentioned above, impulses are generated by our method at faster than real-time rates: micro-level at about 5000 samples per second, meso-level at 1000 samples per second, and macro-level at 100 samples per second or higher. For all the scenes, the sound synthesis module runs at about 100 frames per second (fps) or higher; while the entire system, including visual rendering, sound synthesis, and physics simulation, typically runs at 30 to 60 fps, depending on the events in the scene.

4.4 Preliminary User Study

To assess the effectiveness of our approach, I have designed a set of simple experiments to solicit user feedback on the perceived difference of the auditory experiences accompanying a series of video clips. I have focused on two key aspects: (a) Does the addition of sound synthesized by our approach offer a more immersive experience than the visual simulation alone; (b) Does the sound synthesized by our approach offer a more immersive experience than the sound generated by the existing technique (van den Doel et al., 2001b) that simulates sliding sounds with only the micro-level information?

4.4.1 Procedure

The participants consist of 19 volunteers: 6 women and 13 men, in the age of 8 to 43. For each subject, six sets of video clips were presented. For each set of video clips, all video clips have the same visual simulation but with different sound effects.

In the first three sets of video clips, I show several boxes falling down a ramp and sliding down to the same surface with three different textures: (1) cobblestone, (2) rough, mud terrain, and (3) gridded floor (see Figure 4.9). For each set of video clips, one video is completely silent and the other has impact and sliding sounds generated by our method. For each set of two clips, I asked the user study participants which one offers a more immersive experience over the other.

In the last three sets of video clips, I hope to in addition compare the sense of immersion between our method and an existing method for simulating sliding contacts. The video clips show a pen scraping (4) a brick surface, (5) a ceramic tiled surface, and (6) a wooden, textured surface (see Figure 4.6). In each set, there are three videos. One video has no sound, one video has sound generated using existing technique (i.e. the parametric method for sliding contact sounds (van den Doel et al., 2001b)), and one video with sound generated by our technique (i.e. three-level simulation). The modal basis in (van den Doel et al., 2001b) was constructed based on measurements using a robotic arm which is not available commercially. For a fair comparison, I used the same mass-spring formulation for constructing the modal models and the same transient contact handling (Raghuvanshi and Lin, 2006a) in both our method and the parametric method (van den Doel et al., 2001b). So, the only difference in the two methods in our user study is how each method handle lasting contacts, i.e.

sliding contacts, which is the only variable factor our study focuses on. For each set, these video clips were presented in random orders and I asked the participants which one offers a more immersive experience over the other two.

Some of the video clips used in this preliminary user study are included in the supplementary video accompanying this chapter and the entire study can be found at: `http://gamma.cs.unc.edu/SlidingSound/UserStudy`.

4.4.2 Statistics

In Table 4.1 I summarize the results for the experiment using the set of video clips as described in (1), (2) and (3). In Table 4.2 I summarize the results for the experiment using the set of video clips as described in (4), (5), and (6).

It is well known that good auditory display reinforcing the visual experience can enhance the sense of immersion; similarly unrealistic sound effect that is poorly synchronized with visual cues can disrupt the sense of presence in a VE. Therefore, the addition of auditory cues would not automatically improve the sense of immersion in a VE, unless the added sound effects are realistic and correlate with the visual events well. In all six sets of our experiments, the participants clearly prefer the same video clip with sound over without, indicating that the sounds generated by our method has achieved a satisfactory level of realism to reinforce the visual experience of nearly all subjects.

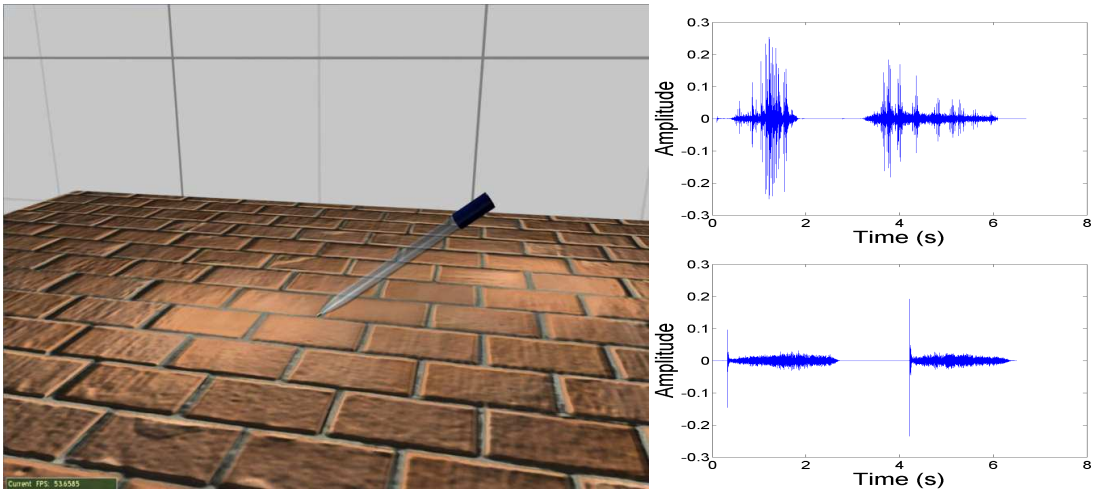
It has been reported in (van den Doel et al., 2002) that individual's ability to perceive sound may vary significantly from subject to subject. However, they overwhelmingly and consistently found the sliding sounds generated by our method offer more immersive experiences than the sounds synthesized by only using the parametric technique.

Experiment	No Sound	Our Method
(1) Cobblestone	1	18
(2) Mud Terrain	0	19
(3) Gridded Floor	2	17

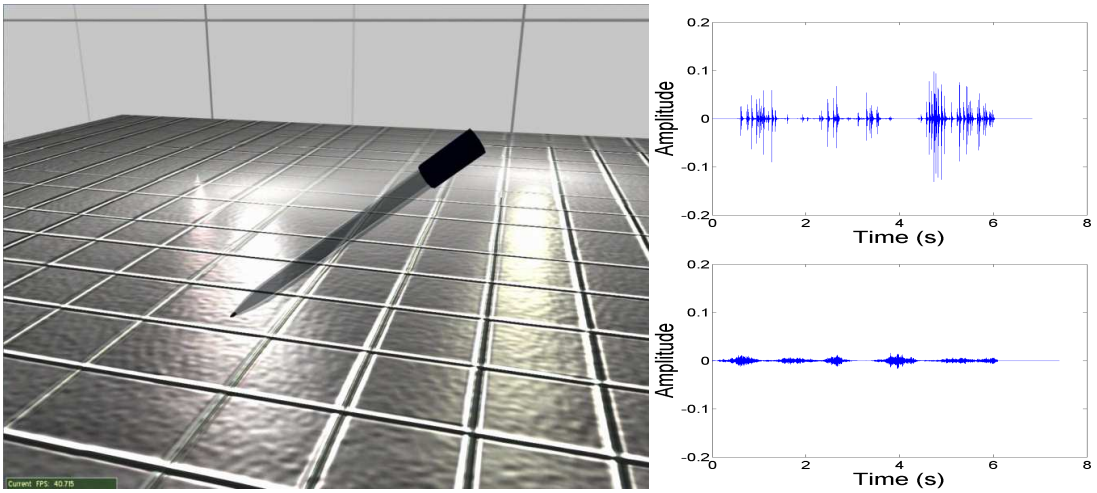
Table 4.1: **Results of User Study:** the number of subjects who feel either no audio or the addition of contact and sliding sounds generated by our method make the video more immersive for each scenario shown.

Experiment	No Sound	Parametric Method	Our Method
(4) Brick	0	0	19
(5) Ceramic	0	0	19
(6) Wood	0	0	19

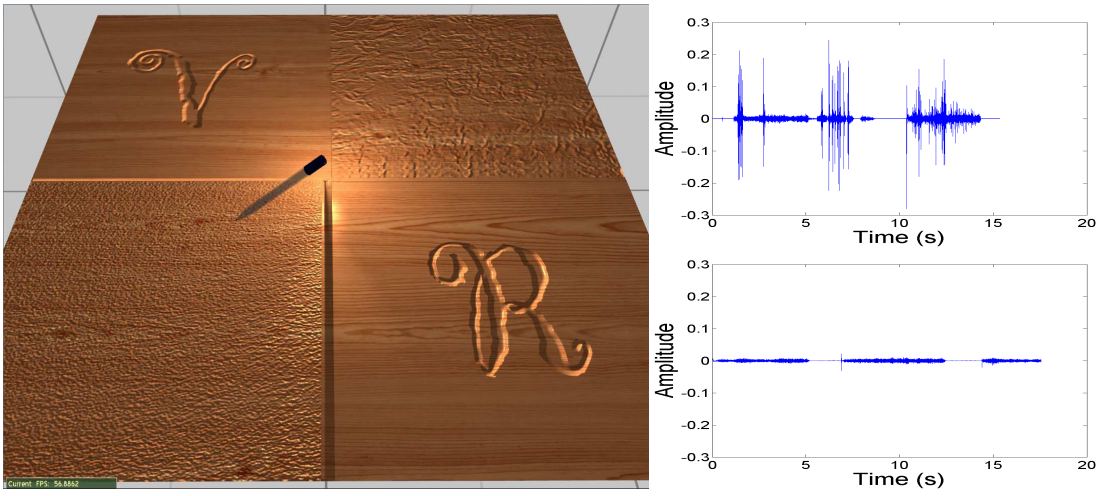
Table 4.2: **Results of User Study:** the number of subjects who feel no audio, or the addition of sliding sounds using only the parametric method, or using our method offers more immersive experiences.



(a) Brick



(b) Ceramic tiles



(c) Wood

Figure 4.6: **Comparison:** Snapshot images of a pen scrapping on three surface textures with different normal maps. The wave plots to the right show the sounds generated by our method (upper) and those generated from previous methods with only contact and friction sounds (lower).

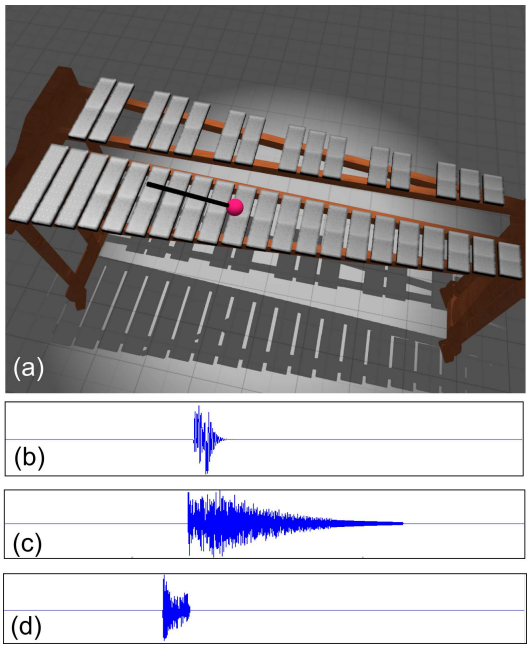


Figure 4.7: An example of a contact sound generated from the virtual marimba-like instrument. The bars are set to have different material parameters. In the three wave files shown above, sound waves correspond to marimba (b: wood), xylophone (c: metal), and a user designed material (d).

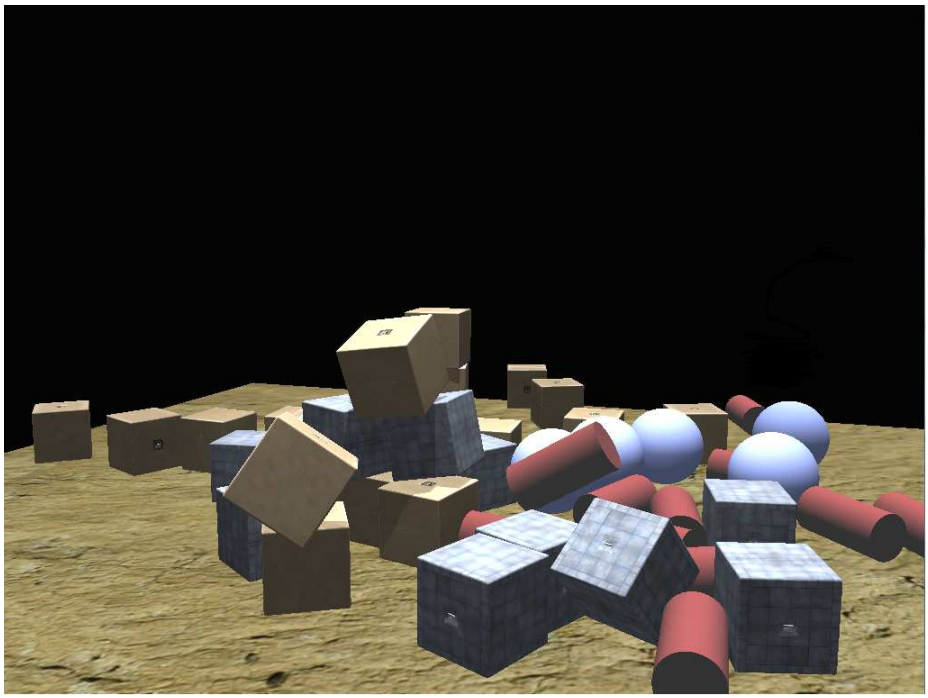
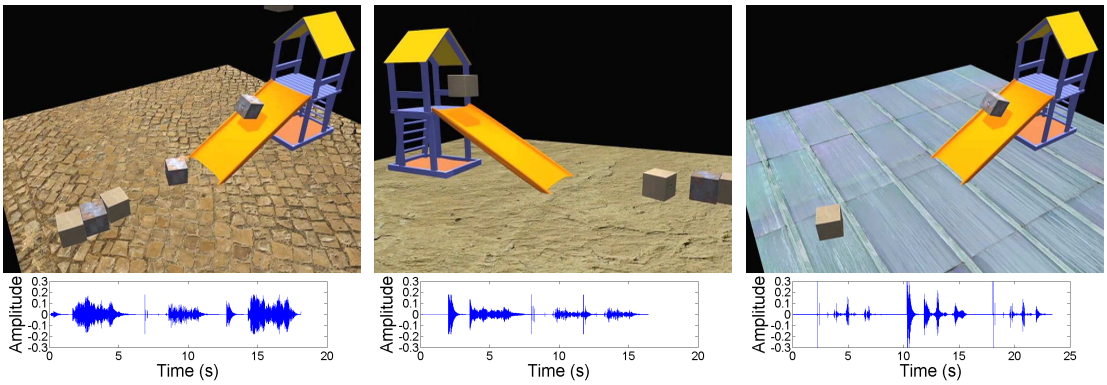


Figure 4.8: Many objects interacting with each other, making colliding, rolling and sliding sounds.



(a) Cobblestone (b) Rough Mud Terrain (c) Gridded Floor

Figure 4.9: Contact sounds (shown in wave plots below each image) generated by our method by the objects moving in a game-like environment, where boxes slide through the same surface with three different textures.

CHAPTER 5: MULTITOUCH VIRTUAL MUSICAL INSTRUMENTS

5.1 Introduction

Music is an integral part of our artistic, cultural, and social experiences and an important part of our life. With the recent advances in computing, scientists and engineers have created many digital musical instruments and synthesizers to perform, edit, record, and play back musical performances. Meanwhile, inventions of novel human computer interaction systems show a new dimension for computer applications to evolve. Particularly, multi-touch interfaces in many forms have been well studied and become prevalent among average users. These devices enable expressive user controls which are suitable for digital music playing. However, it still remains a challenge to build a virtual musical instrument system that allows users to intuitively perform music and generates life-like musical sounds that closely corresponds to user interaction in real time.

In this chapter, I present a virtual percussion instrument system using coupled multi-touch interfaces and fast physically-based sound simulation techniques that offer an alternative paradigm that allowing users to play several different virtual musical instruments *simultaneously* on the same platform with no overhead. The optical multi-touch table provides an interface for users to intuitively interact with the system as they would with real percussion instruments. The proposed system setup accurately captures users' performance actions, such as striking position, striking velocity, and time of impact, which are then interpreted with our *input handling* module and used to control the simulated sounds accordingly.

In addition, the size of this tabletop system enables multiple users to collaboratively participate in the musical performance simultaneously. The sound synthesis, acoustic effect simulation, and the coupling scheme between the two presented in this chapter can generate rich and varying sounds for multiple sounding objects in real time. This feature also makes a collaborative and realistic music playing possible. In addition, these sound simulation techniques preserve the flexibility for easily creating new instruments of different materials, shapes, and sizes. Figure 5.1 shows multiple users

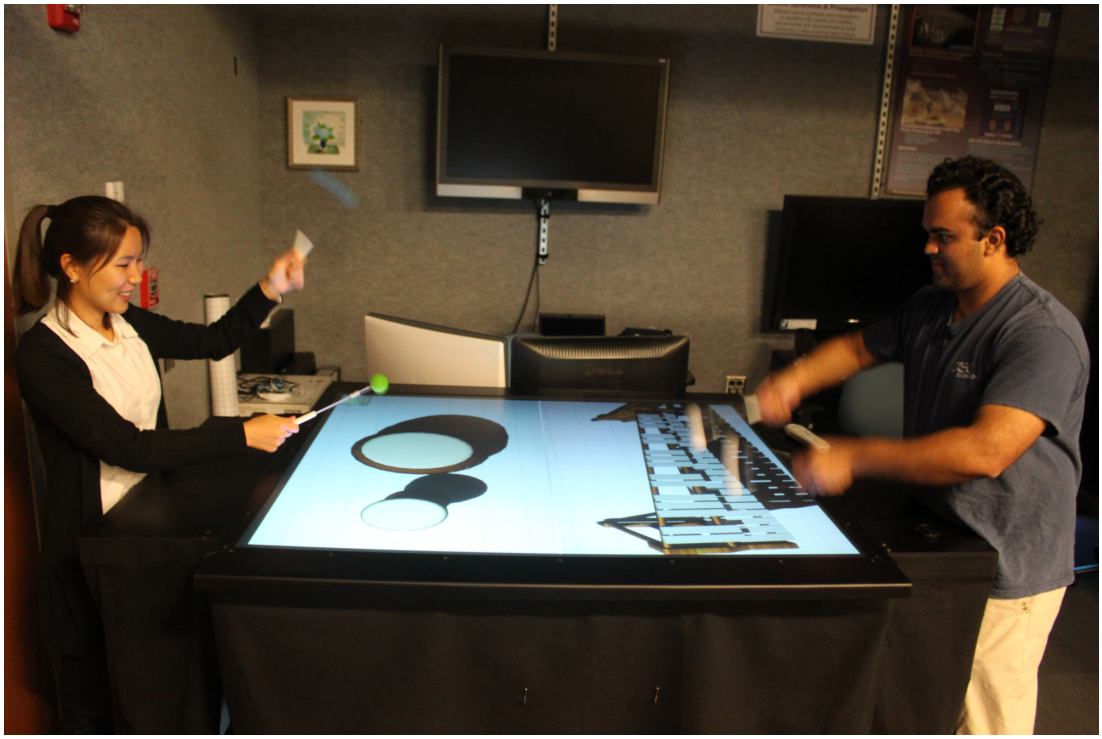


Figure 5.1: **Tabletop Ensemble** Multiple players performing music using our virtual percussion instruments.

playing virtual percussion instruments on our tabletop system. A xylophone and a set of drums of various sizes, shapes, and materials are implemented to demonstrate the system.

Main Contribution: This work is the first known system that uses physically-based sound synthesis and propagation algorithms to simulate virtual percussion instruments on an optical multi-touch tabletop. It offers the following unique characteristics over existing digital instruments:

- **Direct and Intuitive Multi-Modal Interface and Handling Suitable for Percussion Instruments** The multi-touch tabletop user interface enables users to intuitively control a virtual percussion instrument. Novice users can interact with the system with no learning curve. A novel algorithm for mapping touches on optical touch-sensing surfaces to percussion instrument controls is proposed to accurately interpret users' interaction with the tabletop. (Section 5.4)

- **Physically-Based Sound Generation** A physically-based sound synthesis technique is adopted to generate instrument sounds given interpreted user interactions. A numerical sound propagation simulation algorithm is used to model the acoustic effects of the instrument's air cavity. I propose a simple yet effective system integration setup between synthesis and propa-

gation to enable real-time simulation of dozens of sounding instruments. The generated sound closely corresponds to users' interaction with the system, e. g. striking position, velocity etc. (Section 5.5 and Section 5.6)

- **A Reconfigurable Platform for Different Instruments and Multiple Players** Physically-based sound simulation offers the ease of employing various sounding materials, sizes, and shapes for easily creating new virtual instruments, thus making the system reconfigurable with little overhead. Our system setup is capable of accommodating and handling multiple users' simultaneous interaction with the virtual instruments and simulate the musical tunes for many sounding objects at interactive rates. It enables multiple users to collaborate for performing music on a single, portable platform.

I have also conducted early pilot study to solicit qualitative feedback and suggestions from users with various music background and skills. I briefly discuss the results and limitations of this system in Section 5.8.

5.2 Previous Work

This work builds upon two distinct large bodies of research: one in user interfaces for virtual musical instruments and the other in sound simulation for digital music generation.

5.2.1 Multi-Touch Interfaces for Musical Instruments

Electronic musicians have long adopted Musical Instrument Digital Interface (MIDI) protocols for creating digital music. A plethora of MIDI controllers have been built that enable users to perform music. For example, there are MIDI keyboards and MIDI drum pads. Moreover, other novel interfaces have also been explored for virtual instruments (Miranda and Wanderley, 2006; Chuchacz et al., 2007; Weinberg and Driscoll, 2007). However, none of them is as intuitive or easy-to-use for average users as multi-touch interfaces.

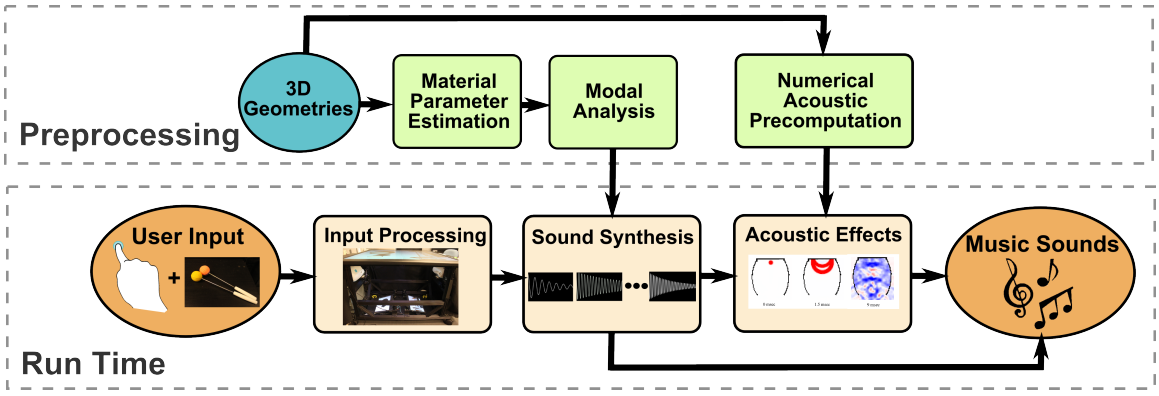


Figure 5.2: **The system pipeline of Tabletop Ensemble**. During the preprocessing stage, our system automatically extracts the material parameters from a sample audio recording for a musical instrument. Given the geometry of each virtual instrument and its material parameters, I can precompute the acoustic effects due to the instrument's body cavity. At run time, user interaction with the multi-touch table is first interpreted by the input processing module and forwarded to sound synthesis engine. Synthesized sounds for instruments with cavity structures are modulated by the precomputed acoustic effects to generate the final audio.

5.2.2 Sound Simulation for Musical Instruments

5.2.2.1 Sound Synthesis

Sound synthesis methods are well studied and applied to digitally generating music. The most realistic yet fast approach is sample-based methods, which process recorded audio samples with parametric models. Some of these models related to music instrument synthesis are presented by Cook (2002a). However, sample-based methods do not offer intuitive or rich control that closely maps to real-world sound generation mechanisms. On the other hand, physical models are superior in terms of natural and expressive controls, and they promise easy flexibilities for creating artificial instruments with expected audio effects. Numerical methods by Bilbao (2009) produce high-quality music instrument sounds. However, like other time-domain wave-equation based approaches, they are not fast enough for real-time applications that demand synthesizing multiple instruments simultaneously.

To physically-based synthesize sound in real time, van den Doel and Pai (1998b) introduced a general framework using resonance modes, i.e. *modal synthesis* ((Adrien, 1991; Shabana, 1997)). This approach generates sound dependent on the materials, shapes, and strike positions of the simulated sounding objects, while it assumes linear dynamics for the vibrating objects. Modal

synthesis applied to simple shapes (e.g. strings, tubes, membranes, and more) with analytical modal analysis results can be found in (Cook, 2002a). Bruyns (2006) showed modal synthesis on arbitrary shapes and compared the synthesized sounds' spectral contents with real-world recordings. Modal synthesis is a suitable approach for our purposes, due to its low run-time costs and flexibility as a physical model approach.

5.2.2.2 Acoustic Effects

The techniques to capture acoustic effects of a space can be classified into two categories - *geometric acoustics* (GA) and *numerical acoustics* (NA). GA approaches are based upon the geometric approximation of rectilinear propagation of sound waves. A large variety of methods have been developed starting from ray-tracing and image source methods in early days to current techniques that include beam tracing (Funkhouser et al., 2004), frustum tracing (Chandak et al., 2008), phonon tracing (Deines et al., 2006) and many more. A more detailed survey can be found at (Funkhouser et al., 2003).

NA techniques solve the wave equation of the sound propagation and therefore capture all the wave effects of sound. Typical numerical techniques include Finite Element Method (FEM) (Thompson, 2006), Boundary Element Method (BEM) (Brebbia, 1991), Finite Difference Time Domain (FDTD) (Sakamoto et al., 2006), spectral methods (Boyd, 2001) and more recently Adaptive Rectangular decomposition (ARD) technique (Raghuvanshi et al., 2009). NA techniques have high computational cost and used only in offline simulations.

Recently, a new wave-based acoustic simulation technique has been proposed by (Raghuvanshi et al., 2010) for performing real-time sound propagation in complex static 3D scenes for multiple moving sources and listener. This technique captures all the acoustic wave effects like diffraction, reverberation, etc., and exploits human perception to efficiently encode the acoustic response reducing the overall memory requirements. It divides the computation into three stages: an off-line simulation to compute acoustic response of the scene, an off-line perceptually-motivated encoding of this response, and a fast run-time system to perform auralization. I adopt this method to introduce acoustic effects to our percussion instruments.

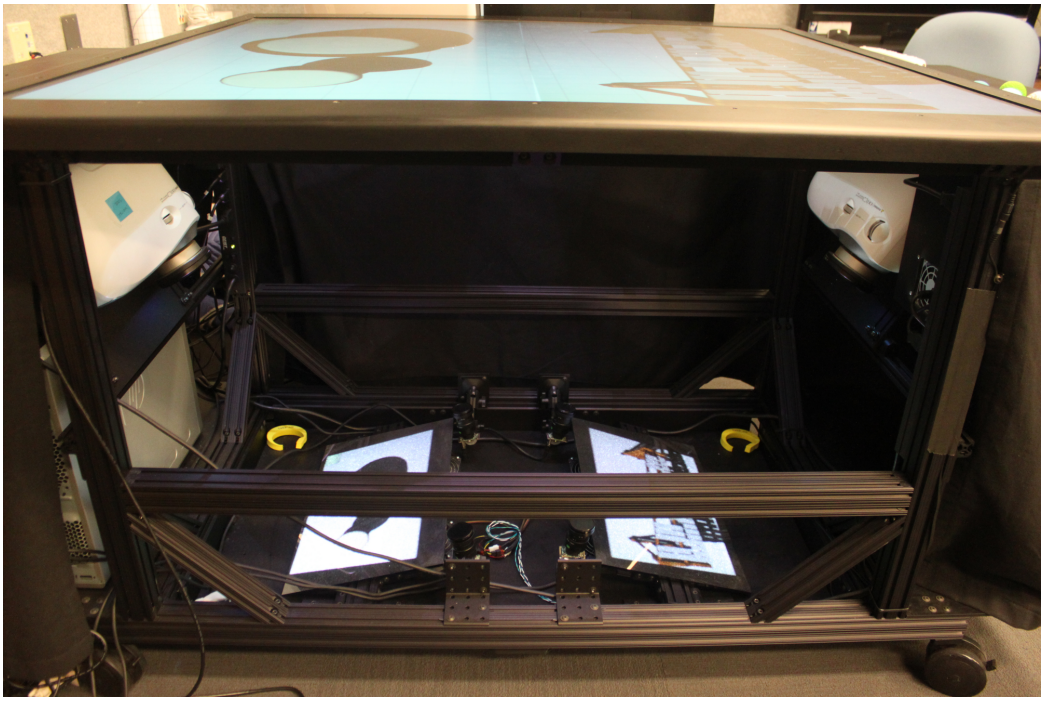


Figure 5.3: The optical multi-touch table with diffuse side illumination, upon which our virtual percussion instruments are built.

5.3 System Overview

This section gives an overview on the hardware configuration and the software modules that make up our touch-enabled virtual instrument system.

5.3.1 Hardware Apparatus

Our application is developed on top of a custom-built optical multi-touch table, using the diffuse side illumination technique. By employing a sheet of $CyroAcryliteEndLighten^{TM}$ with polished edges and a 5-foot strip of LED Edge-View Ribbon Flex from $EnvironmentalLights^{TM}$, I are able to distribute the infrared (IR) illumination more evenly. The touch detection for our tabletop is handled by four Point Grey Firefly MV FMVU-03MTM cameras. For the projection surface, I use a thin, 3mm-sheet of $AcryliteRP7D513$ rear projection acrylic. This design works out well since the thin sheet protects the more expensive Endlighten material and the projection surface has a nice touch. The table has a 62" diagonal work surface and is 40" tall (see Figure 5.3) with two high-definition rear-projection display (1920×2160 pixels), driven by a 3.2 GHz quad-core Xeon processor.

The entire table was designed through an architecture of commodity-level components and custom software. This high-resolution interactive display provides an effective means for multiple users to directly interact with their application system, data, and peripherals. It comfortably allows 4 to 6 people to work at the table simultaneously. The table allows tracking multiple (up to 20 or so) interactions on its surface by properly-sized objects that are infrared reflective. Tracked touch events' IDs, timestamps, contact positions, and contact area information are provided for application development. The size of this multi-touch tabletop and its interaction mechanism make it an attractive, intuitive physical interface for playing virtual percussive instruments.

Optical multi-touch interfaces like this multi-touch table accurately tracks touch points' spatial information on the 2D touch plane. However, without additional ceiling mounted camera or tracking, it is difficult to obtain information on how fast an object is approaching the table surface, i. e. hitting velocity, which is one of the most important control parameters in playing percussive instruments. In order to obtain this parameter, I propose using deformable bodies as the hitting object, and I design a velocity estimation algorithm based on this deformation data. As the input to the system (as shown in Figure 6.2 and supplementary video), soft sponge balls of roughly four centimeters in diameter are used as the mallet heads for exciting the virtual instruments. Users can also play the instruments with fingers, which are tracked in the same way as the sponge balls with a slightly different configuration. The input handling process is explained in detail in Sec. 5.4.

5.3.2 Algorithmic Modules

Figure 6.2 illustrates the overall algorithmic pipeline. The virtual percussion instrument implementation depends on two separate stages. One is the preprocessing phase, where an instrument's 3D geometry and a recorded impact sound of an example material are analyzed. In *modal analysis* and *numerical acoustic precomputation*, *resonance* and *wave simulation* data are generated respectively, which are later used in sound synthesis and acoustic effect modules in the next phase. During runtime, touch messages from tabletop hardware are interpreted by the *input processing* module, and excitation to virtual sounding instruments are generated accordingly. Given the excitation, *sound synthesis* module generates sound using modal synthesis techniques. For instruments with air cavities, their synthesized sounds are further processed by *acoustic effects* (i.e. sound propagation) module for

adding important audio effects due to resonance in their cavities. Details on the preprocessing and runtime modules are presented in the following sections.

5.4 Touch Input Processing

The impulse applied by the striking body to the virtual instrument is directly used as excitation to our sound synthesis engine. I choose impulse over pressure, because pressure at one instance does not necessarily reflect how hard users are hitting, e.g. users can be statically pressing against the surface but this should not excite the vibration of the virtual sounding objects. Therefore, how accurately I can capture the impulse information determines how well I can model a performer's control over the generated music sounds. Impulses are proportional to the change of velocity, and derived by estimating the rate of change of the striking body's velocity.

Without the loss of generality, let us assume the touch-surface is the X-Y plane. It is relatively easy and already a standard technique to track the velocity in the X-Y plane on an optical multi-touch tabletop. However, velocity along the Z-axis (perpendicular to the tabletop) cannot be directly acquired. While systems with extra cameras mounted perpendicular to X-Y plane or full 3D motion capture are feasible for tracking velocity in Z-axis, such a set up adds additional hardware and calibration overhead. More importantly, with camera-based tracking, when multiple users are interacting with the system, multiple interaction points (e.g. hands) occlude one another, which might greatly impact the accuracy of the tracking. For processing multiple (and possibly simultaneous) touch inputs, I propose to use soft bodies, representing either sponge balls as the mallet heads or user's finger tips, as an input device for the multi-touch table. I describe how velocity perpendicular to the touch surface can be tracked through the deformation of the soft bodies. This approach involves simple and easy computations that can be adopted to add velocity tracking for the third dimension beyond the touch surface for any type of optical touch-enabled device – *with or without pressure sensing*.

5.4.1 Z-Velocity Tracking

In order to track Z-velocity, a sequence of occlusion information registered by the multi-touch system is recorded for each striking soft body. The recorded occlusion information includes the touch

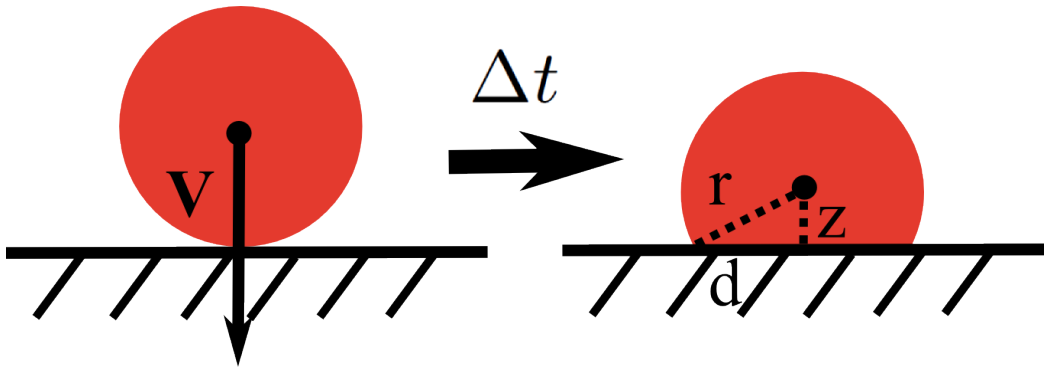


Figure 5.4: A snapshot of the cross section of a soft ball striking against X-Y plane at a velocity V. After time Δt, the ball is deformed, and its center position in Z-direction can be calculated. Velocity in Z-axis can be derived from this position and the elapsed time information as discussed in Sec. 5.4.1.

center position, the occluded area, and the time stamp of this occlusion event. Figure 5.4 illustrates a snapshot in time of a squeezed soft sphere after it strikes against the X-Y plane. By using this snapshot of occlusion data, i.e. the radius of the occluded circle (denoted as d_n) and given the striking soft sphere's radius (denoted as r), I can derive the striker's center distance from the X-Y plane with $z_n = r - \sqrt{r^2 - d_n^2}$. Therefore, the average Z-velocity v_z in one time step Δt can be quickly calculated with

$$v_z = \frac{z_n - z_{n-1}}{\Delta t}. \tag{5.1}$$

With a single strike, one or more time steps may have elapsed, I need to use the *average* velocity throughout all time steps of the whole sequence to more accurately estimate the Z-velocity for this one strike. However, with slower strikes, the whole sequence can span a long interval. Computing the average velocity at the end of these strikes would introduce a significant latency between the hit motion and the generated audio to users. In order to eliminate perceptible latency, I adopt a temporal window. Velocity values from the initial contact time to the last time step within this window are averaged to approximate the Z-velocity of this strike. According to a perceptual study by Guski and Troje (2003), 200ms of latency is the tolerance for human to reliably perceive an audio signal and a visual signal as a unitary event, and this temporal window size is also adopted and verified by Bonneel et al. (2008b) for plausible sound simulation. In our case, I employ an even smaller temporal window of 100ms for average Z-velocity estimation, which gives us good results. When the occlusion area of one touch sequence decreases with time, that touch is considered a release, and

the buffer associated with this touch is cleared. At each time instance, the occlusion centered within a small spatial range near a touch in the last time step is considered coming from the same soft body and therefore stored in the same buffer for velocity estimation. K-d tree is used for nearest neighbor search to accelerate this clustering process. With this approach, I can easily track a large number of simultaneous touches from multiple soft bodies, sponge balls and/or user fingers, to estimate their Z-velocity and generate corresponding excitation to our sound simulation.

5.4.2 Implementation and Results

Theoretically, the higher the cameras' frame rate, the more accurate the Z-velocity estimation. However, increasing frame rate also lowers cameras' resolution, which undermines the accuracy for tracking occlusion area. Through experiments, I decided on adopting 504×480 as the region of interest (ROI) resolution with the cameras in our system, and under such configuration, the frame rate is roughly 60 frames per second. Better camera hardware is likely to increase the accuracy of the proposed velocity estimation heuristic. In our implementation, the radius parameters for the sponge balls are 15 pixels, while the radius for hand contacts are set as 10 pixels.

Using a metronome, I repeatedly strike the table from 30cm above at four different tempos, namely 60, 100, 150, and 200 strikes per minute. For each tempo, 100 such strikes are performed. The mean Z-velocity of all those strikes estimated with our method along with their standard deviation are shown as red in Fig. 5.5, while the real values directly computed from dividing the strike distance and the strike interval are shown as blue. With our method, I can accurately estimate the Z-velocity with only the deformation data.

5.5 Sound Synthesis

As mentioned in Sec. 5.2, *modal synthesis* technique has been employed for sound synthesis in our system. It is one of the most widely adopted approaches for generating sounds based on the first principle of physics in graphics, game, and music communities (O'Brien et al., 2002c; Raghuvanshi and Lin, 2006b). *Modal analysis* is performed during the preprocessing stage to analyze an arbitrary 3D geometry and its material parameters to compute the resonance modes of that object. The output is a bank of damped sinusoidal waves, i. e. *modes*. At run time, different excitations to the model trigger

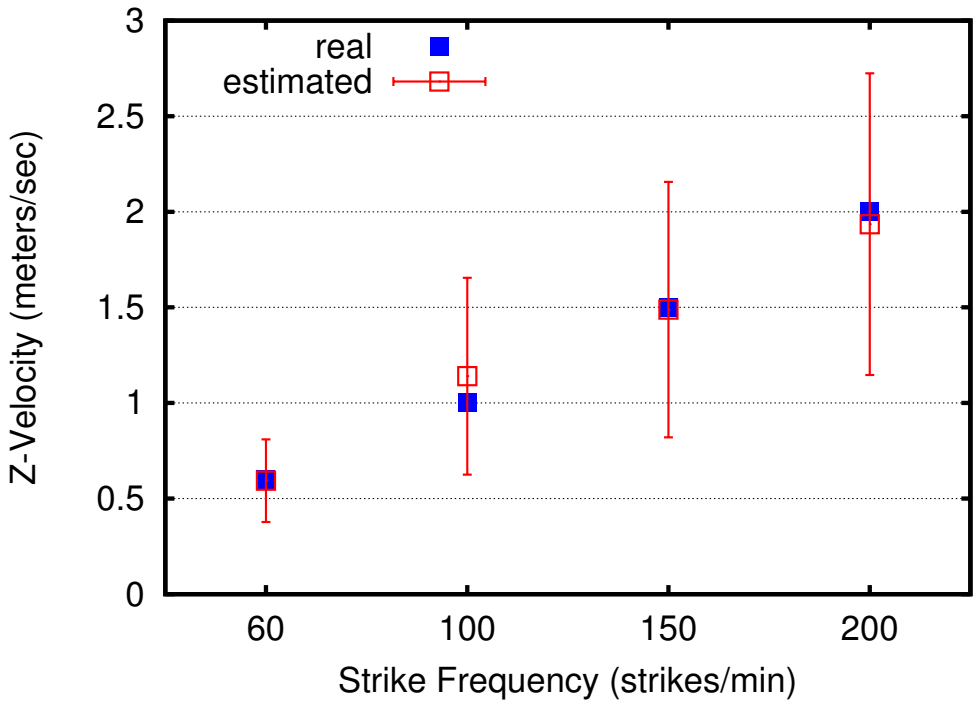


Figure 5.5: Estimated Z-velocity vs. real velocity values: This experiment is performed under four different tempos for strikes, i.e. 60, 100, 150, and 200 strikes per minute.

different modal responses. This approach assumes linear dynamics for the vibration of sounding objects with proportional damping, which is also often adopted for simplification. These assumptions make this method less suited for modeling some highly complex sound phenomena, yet sufficiently (physically) correct for our purpose in real-time synthesis of musical tunes that closely correspond to user interaction. It also offers the flexibility of changing instruments' physical properties for rapid prototyping.

One of the challenging and important elements for high-quality synthesis of modal sounds is to acquire appropriate material parameters for modal analysis. This process is time-consuming when 3D model meshes are not sufficiently fine and detailed for directly using real physical parameters. More importantly, parameters like proportional damping coefficients do not directly map to real-world materials, therefore impossible to look up for modal analysis. In our preprocessing, I adopt the example-guided parameter estimation algorithm introduced in Chapter 3. Guided by a sample audio recording of a xylophone bar and of a drum, this automatic, offline process facilitates quick

determination of material parameters of realistic sounding materials the simulated xylophone and a drum set.

In our system, I modulate synthesized audio for instruments that have notable acoustic effects, in addition to vibration sounds. This process is presented next in Section 5.6. Real-time synthesized audio samples for each sounding object are formated into buffers of size 2048 and then passed on to the acoustic simulation module in a separate interprocess communication pipe.

5.6 Acoustic Effects

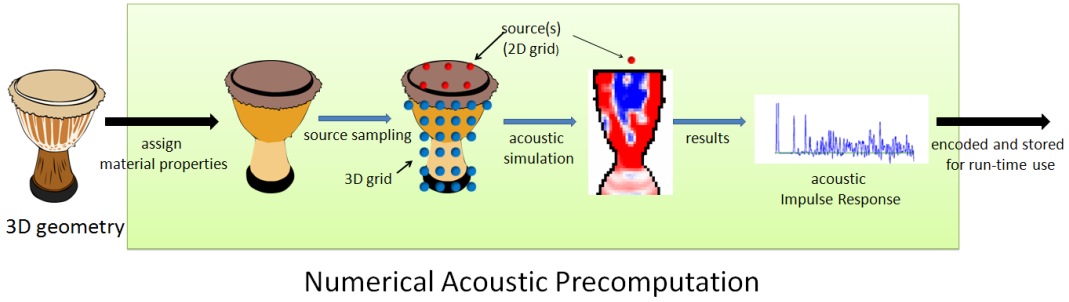


Figure 5.6: **Numerical acoustics precomputation pipeline**: The input to our system is a 3D model of the virtual instrument. I assign material properties to its different parts based on the type of percussion instrument I want to model. Next, I place impulsive sound sources (red spheres) at sampled positions on its sound generating surface, run the numerical simulation and collect impulse responses at 3D grid positions (blue spheres) corresponding to each source. This impulse response is encoded and stored for run-time use.

In most musical instruments, especially percussion instruments, sound produced by a generating surface (membrane or string) propagates inside the cavity of the instrument and gets modulated due to its shape, size and the material. The propagation of sound inside the air cavity produces resonance resulting in amplification of certain frequencies and loss of others. This vibration of the generating surface along with the acoustic effect of the air cavity, gives the musical instrument its characteristic sound. Therefore, while designing virtual music instruments, it is critical to properly model acoustics of the instrument i.e. the way its shape, size and material changes the sound. In our system, I perform *one-way coupling* of the sound synthesis and acoustic simulation stages. The sound generated by the synthesis stage enters the instrument cavity after which the acoustic simulation stage propagates this sound inside the cavity to model its acoustic effects. The final propagated sound then leaves

the music instrument towards the surroundings. This one-way coupling is a good approximation for percussion instruments that are open at one end like congo drums.

In order to capture the acoustics inside the instrument's cavity in a physically-accurate way, I chose wave-based simulation technique of (Raghuvanshi et al., 2010). This technique captures all the wave-effects of sound including diffraction, interference, scattering and reverberation. It performs real time sound propagation for multiple sources in a static environments in a fast and memory efficient manner. This capability enables us to handle multiple percussion instruments and their acoustics at interactive rates and maintain low latency in our system, a critical requirement for satisfactory user-experience.

In the pre-processing stage (see Figure 5.6), I start with a 3D model of the instrument and assign material absorption coefficients to its various parts. I then sample positions on the sound generating surface (2D membrane in case of drum), place an impulsive sound source at each position, run an acoustics simulation (Raghuvanshi et al., 2009) and determine the sound propagated inside the instrument cavity including reflection, diffraction and interference of sound waves. This propagated sound produced by the impulsive source is called acoustic *impulse response* (IR). It completely determines the acoustics of the instrument cavity. IRs are recorded at sampled 3D locations inside the cavity and stored in a highly compact representation as discussed in (Raghuvanshi et al., 2010).

At run-time, the hit position and the corresponding sound produced by the synthesis module is passed as input to the propagation technique. The listeners are placed on the sound generating surface to capture the sound emitted by it. Based on the hit position, this technique performs a look-up of the IR corresponding to the nearest sound source at the given listener position and performs an interpolation to produce the correct IR for that hit position. This IR is convolved with the synthesized sound in real time to produce the final propagated sound capturing the acoustics of the instrument.

In our test scenarios, I have simulated the acoustics of five drums with different shapes, sizes and materials. Large drums trap the sound more effectively and hence have longer reverbs (more echoing). On the other hand, small drums placed high above the ground are less reverberant. I have also tested two different material properties - metallic and wooden. Since the metallic drums have low absorption coefficient, they have longer reverberation times compared to the highly absorbing wooden drums.

(a) Xylophone (b) Drum Set

Figure 5.7: 5.7a shows a virtual metallic xylophone, and 5.7b shows a five-piece drum set.

5.7 Instrument Modeling and Implementation

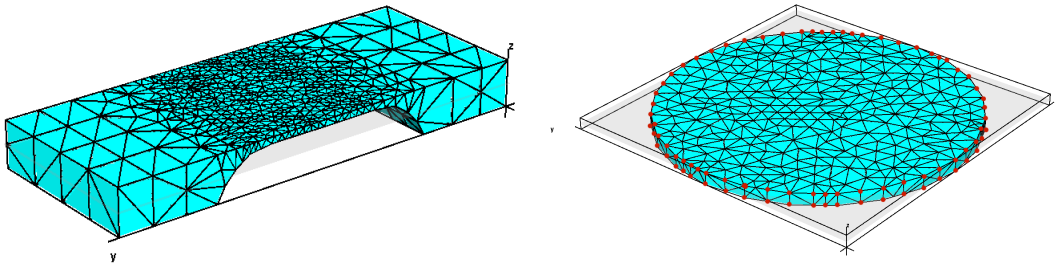
Two types of representative percussion instruments are simulated: a xylophone (Fig. 5.7a) and a set of five drums (Fig. 5.7b) with various membrane sizes, cavity shapes, and drum wall materials.

5.7.1 Sound Generation

I model xylophone bars with arched curve at the bottom just like real xylophone bars (see Figure 5.8a). Real xylophone bars are normally strung to their nodal points to minimize the damping from external attachment. Therefore, in our simulation, I simplified the mechanism by allowing xylophone bars to freely vibrate for sound generation purposes. For drum heads, I model them as circular plate with a small thickness. The rim vertices of a real drum are firmly attached to a drum body. Therefore, I specify all rim nodes of our virtual drum as fixed nodes in modal analysis (shown in red dots in Figure 5.8b). In our implementation, I first discretize 3D geometries into tetrahedra with TetGen (Si, 2011), with no tetrahedron's radius-edge ratio greater than 2.0 (shown in Figure 5.8). I then perform finite element analysis on the discretized geometries to acquire the simulation meshes which are used in modal analysis.

5.7.2 Acoustic Simulation

I now discuss the implementation details of acoustic simulation in our multiple-drum scenario. In the pre-processing stage, the sampled sound sources are placed on membrane of each drum at 20 cm distance resulting in, typically, $5 - 10$ sampled sources per drum. I tested two material properties



(a) Xylophone Bar Mesh (b) Drum Head Mesh

Figure 5.8: Discretized mesh representation for the xylophone bar and drum head models used in this system. The red dots in 5.8b indicate fixed nodes.

for the drums - metallic and wooden, having absorption coefficients of 10% and 30% respectively[1] (Fig. 5.9). The run-time propagation system can handle a maximum of 10 instruments playing at the same time. Since I are mainly interested in the sound propagating from the drum membrane, our listeners are placed at the center of each drum's membrane. For the multi-drum scenario, the final auralized sound is a mix of the sounds received at the listener of each instrument.

5.7.3 System Integration

In order to capture the acoustics of percussion instruments, I propose a simple and efficient method to couple the synthesis and acoustic simulation systems. Our sound synthesis pipeline performs *modal analysis* to generate sound due to the vibration of the drum membrane. This sound is packaged in audio buffers and transferred to the acoustics simulation system over the Windows interprocess communication (IPC) framework called *Named pipe* (framework Named pipes, 2011). To avoid any communication delay between the two systems, I use asynchronous data transfer over the pipes. At the acoustic simulation side, the audio buffers are convolved with the appropriate impulse response to generate the auralized audio. This auralized audio is sent to the sound card for playback using the XAudio2 (XAudio2, 2011).

The size of the buffer and number of pipes used depends on the latency requirement of the application. Small buffer size implies low latency between the two applications but higher communication cost per byte due to large number of buffers transferred. Large buffers have low communication cost per byte but high latency. I found out experimentally that a buffer size of 2048, corresponding to

[1]Absorption coefficient of 10% means the surface will absorb 10% of the incoming acoustic energy at each interaction with the sound wave

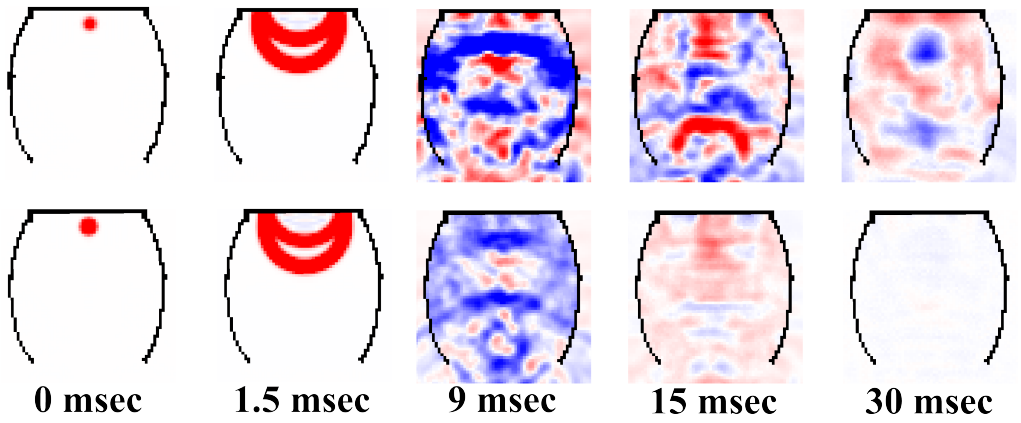


Figure 5.9: Acoustic simulation results for metallic (top row) vs wooden (bottom row) drum at different time-steps with absorption coefficient of 10% and 30% respectively.

50ms at sampling rate of 44.1kHz, satisfied our latency requirements. Since all the instruments can potentially be played at the same time by multiple users, I create a dedicated pipe for each sounding object in the musical instrument to transfer data from the synthesis to the acoustic simulation system. Therefore, the number of pipes is equal to number of sounding objects in the system.

5.8 Results and Discussions

I present a multi-modal interaction system that allows users to intuitively perform percussion music on a xylophone and a drum set. I achieve interactive handling of multiple users' touch inputs, sound synthesis, sound modulation using physically-accurate acoustic simulation, and final auralization – *all in real time*.

5.8.1 Results

Our multi-touch interface tracks touches on the tabletop surface and also estimates hit velocity perpendicular to the tabletop. This capability allows us to model the musical performance of the user over the percussive instrument. In the supplementary video, I show how volume of the generated sound is directly modulated by performers' hit vigor, i. e. the faster the strike against the instrument, the louder the simulated audio. Users' hit position is also accurately tracked and transformed into corresponding excitations to the system, resulting in generation of position-dependent musical sounds based on user interaction.

The sound simulation engine implemented in this system efficiently couples a sound synthesis module and a numerical acoustics simulation. Generated audio captures essential reverberation effects due to air cavity in the instrument. The coupled system handles multiple numbers of sounding objects and adds prominent acoustic effects to the synthesized sounds all in real time without perceptible latency. The complete sound simulation effects are shown on the five-piece drum set in the accompanying video. Note how added acoustic effects instantly change the overall sound quality of the drum simulation. The physically-based sound synthesis and propagation models adopted by the system also provide the flexibility for changing the simulated instruments based on their shapes, sizes, and materials easily. Results of instruments with different physical properties are also shown in the supplementary video.

I invited people of different age groups and various music playing backgrounds, from novice players to professional musicians, to play our virtual instrument system. All the users were able to interact with the system and play the instrument easily without any learning curve or significant familiarization with the setup. Multiple users were able to collaborate naturally on the tabletop. Users also appreciated the fact that the size of the virtual instruments on the touch table resembled the real xylophones and drums, which made it easy to play.

5.8.2 Limitations

Although the proposed hit velocity tracking and estimation method works well with the sponge balls adopted in our implementation, the estimation for direct interaction with hands is not as accurate. Limited deformation of fingers and palms, and the small variation in occluded area data make it hard to accurately estimate velocity along the direction perpendicular to the touch surface. Moreover, human hands vary in size from person to person. Even for the same person, the size of a finger contact is very different from a palm contact. Without a fixed, known deformation model for hands, it is difficult to provide correct velocity estimation purely based on the tracked deformation. One possible solution is to incorporate a user-specific hand deformation model. However, such a computation may be too costly for interactive user experiences. For accurate control over hit vigor, users employ the sponge ball mallets. Due to limited frame rates and resolution of the cameras used in our system (discussed in Sec. 5.4.2), the implemented system has an upper limit for strike velocity that it can

handle. Therefore, when users are attacking the touch surface at a very high speed and lifting up immediately, it is likely that those touches are not registered, resulting in missed strikes.

Our user input processing through the touch-enabled interface provides users an experience that emulates performing on real instruments. Synthesized sounds correspond to users' performance and also the intrinsic characteristics of the instrument itself. However, the current system does not incorporate all user controls. For example, users cannot damp the resonance bodies with contacts to achieve articulation like staccato. Additionally, in more complex musical instruments, the propagated sound can in turn affect the vibrations on the sound generating surface resulting in a reverse feedback that has to be modeled as *two-way coupling*, which is not simulated in our current implementation.

5.9 Conclusions and Future Work

In conclusion, I present a virtual instrument system that enables multiple users to simultaneously perform musical pieces together on a single platform. It uses an efficient and responsive approach that interprets the user inputs from an optical multi-touch interface and generates excitation information for real-time sound simulation to create realistic sounds depending on striking position, impact vigor, instrument shapes, and instrument cavity. While our current hardware setup suits collaborative purposes for scenarios like museums and schools, these design principles can be easily adopted to run on other input devices, such as multiple tablet PC, iPad, or other commercial multi-touch displays. Based on early user feedback, this multi-modal system is intuitive, easy-to-use, and fun to play with for novice users and experienced musicians alike.

For future work, I plan to introduce new interfaces for users to change instrument parameters and properties on the fly and experience the fun of building their own virtual instruments. In addition, more accurate physical models for sound simulation can be explored to achieve richer and more realistic sounds, especially nonlinear effects in synthesis and two-way coupling between synthesis and propagation.s With further algorithmic advances and novel features, I hope to provide users with more forms of virtual instruments in the future.

CHAPTER 6: VIRTUAL MUSICAL INSTRUMENTS ON MOBILE DEVICES

6.1 Introduction

Mobile devices with multi-touch hardware like smart phones and tablets have brought a disruptive change to our daily interaction with computing devices. Such mobile devices have become widely adopted for both professional and personal uses in all aspects of our lives. Nowadays many more mobile devices are sold than desktop and laptop computers, and mobile systems are becoming more dominant and accessible than conventional computing platforms. Compared with desktops and latptops, these mobile devices present a more natural and easy-to-adopt user interaction pattern that can be immediately picked up with little to nearly no training. Therefore, this type of user interface is more suitable for consumer multimedia applications that involve multiple sensories and more complex interactions. Virtual music playing systems are among such applications.

A plethora of music related applications specifically made for mobile platforms can be found. Currently, almost all of them use only pre-recorded music or audio sample playbacks. However, this setup presents considerable drawbacks to provide a natural and expressive music playing experience. First, most of them only allow simple interactions like a single touch or tap, which triggers a single sound playback. More complex user interactions, which can be easily captured by professional human-computer interfaces like motion trackers and digital gloves, are difficult to handle on consumer multi-touch screens. Moreover, due to limited computing power, current applications cannot provide

Figure 6.1: **Virtual Musical Instruments:** The two images on the left show a user editing virtual musical instruments and a screenshot of the mobile application in **editing mode**, while the two on the right show **playing mode**.

realistic and dynamic synthetic sound effects, which are usually achieved by physical modeling and directly driven by the user motion.

In this chapter, I propose a reconfigurable virtual musical instrument system on a consumer mobile device that is responsive to rich user interactions and capable of generating dynamically varying sound based on real-time user control. The main results are:

- **A flexibly configurable virtual musical instrument system,** with which users can create their own musical instruments by choosing from two basic types of instruments and multiple materials, and editting the instruments based on pitches. This capability allows quick prototyping of virtual instruments.

- **An effective interaction processing algorithm** that handles strike and slide actions. It tracks in real time multiple finger inputs and scales to a large number of virtual objects.

- **A fast physical sound synthesis model** to generate sound effects that are dynamically varying and truly reflecting user's physical interaction with the device.

- **A real-time multimodal application** built upon both interaction processing and physically-based sound synthesis on a consumer mobile device with very limited computing resources. Auditory, visual, and haptic feedbacks are all efficiently computed on the fly.

6.2 Previous Work

In the last decade, multi-touch hardware has become widely available. Han (2005) first designed a low-cost optical multi-touch tabletop and made multi-touch interaction accesible on a large-size display. However, its application has been mostly limited to professional visualization uses. Since the introduction of multi-touch screens on consumer smartphones and tablets, capacitive multi-touch hardware has become commonplace. On such devices, music applications like the GarageBand (Apple, 2014) and djay (Algoriddim, 2014) have become popular. Nonetheless, these applications all store static sound samples and play them at run time when triggered by user touches. Some of them are configurable, but they are all limited to simple selection of instruments and do not offer variations that match an instrument's physical properties like materials and shapes. Real-time physically-based sound synthesis have been a focus of recent interest. The most widely adopted

algorithm is *modal synthesis* (Adrien, 1991; Shabana, 1997). Unfortunately, this approach requires expensive pre-computation called *modal analysis*, which renders it infeasible for applications that alters object geometry and materials at run time.

6.3 User Interface Design

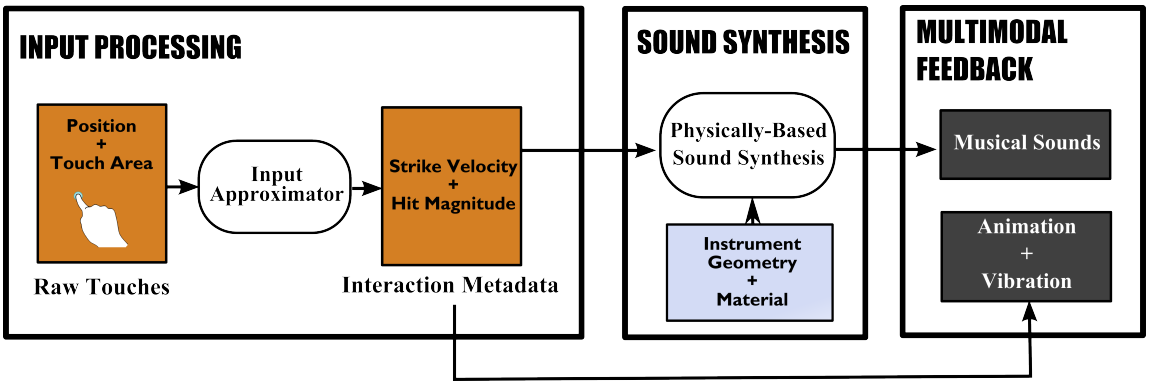


Figure 6.2: **Playing Mode System Pipeline:** Raw multi-touch events registered by touch screen are processed by the **Input Approximator** and interpreted as meta interaction data. These data are used to drive the efficient physically-based sound synthesis module, which takes the instrument geometry and materials defined in **editting mode**. The interpreted interaction data also determine the dynamic animation and vibration the user experiences. Together richly varying multimodal feedback that corresponds to the user input is computed in real time.

Our goal is to design and implement a virtual musical instrument system that is easily reconfigurable and when being played dynamically and realistically responds to user interactions in real time. The mobile application I present provides two distinctive modes for these two different uses, namely *editing mode* and *playing mode*. On application startup, users are presented with the editing mode. A mode switching button is shown on the buttom left of the screen (see application screenshots in Figure 6.1). The button label tells users which mode they are in, and when tapped, the application is quickly toggled into the other mode. This button is also properly sized, so that users can easily tap on it while focused on editing or playing the instruments. For example, when users are editing instruments, they get the visual feedback for what the instruments' shape and materials are, but they might also want auditory feedback on what the instruments sound like. In this case, users can focus on the instruments, easily tap the mode switching button, enter into playing mode, hit the instrument for sounds, and finally tap back into editing mode and continue with editing without much cognititive overhead. To further facilitate easy mode, I assign different background colors for the two modes:

BLACK for editing and GREY for playing. As users spend more time with this application, they naturally associate one mode with one background. Once that association is established, without even looking at the text on the button, users can smoothly distinguish and switch between the two modes, resulting in even smaller cognitive overhead.

6.3.1 Editing Mode

Editing mode is for users to create and configure their own customized virtual musical instruments to their liking. On application startup, an empty screen is presented and waiting for users' creation. Users are allowed to select from two types of instruments, namely *string* and *bar* instruments. In addition, for each type, users can choose a material. Two dropdown menus at the bottom of the screen are shown for choosing the *type* and *material*. From the first menu, users are presented with the option to add a copper string or a nylon strying, while the second menu presents the ability to add a wooden bar, a metallic bar, and a plastic bar. When one of these menu items is tapped, a musical instrument of the corresponding type and material is created. Both the sound and the visual renderings of the instruments match the chosen type and material. Users can combine these instrument types and materials and create a variety of instrument configurations.

Moreover, users can edit any existing instrument on the screen. By touching an instrument, users express the intent to *select* it. When an instrument is selected, it is animated, and the device also vibrates to notify users of a successful selection. Users can perform two types of editing on the selected instrument, namely *pitch configuration* and *deletion*. A third dropdown menu lists musical notes from C_3 to B_4, a total of 14 pitches. Users can select any pitch from the list and assign it to the selected instruments. The length of the instrument also automatically changes corresponding to the chosen pitch. This realistically matches the physical properties of the virtual instruments' real-world counterparts.

6.3.2 Playing Mode

Playing mode is intentionally designed to be simple, so that users can focus on interacting with the instruments they created and configured. To provide a multimodal experience while still giving users control, I allow users to toggle the animation of instruments (visual feedback) and vibration of the device (haptic feedback) when an instrument is being played. Therefore, the user interface of

playing mode only consistis of the instruments and two small buttons shown at the buttom right of the screen for toggling animation and vibration (see the screenshot of playing mode in Figure 6.1). Despite the simple user interface, playing mode offers users rich and detailed multimodal feedback that is completely computed on the fly to match users' multi-touch inputs. In order to achieve this goal, I designed and adopted specific algorithms that are elaborated in Section 6.4.

6.4 Algorithmic Design

To provide richly, detailed, and dynamic responses that are driven by user interaction in three-dimensional space and realistically match the geometries and materials of the musical instruments, real-time physical modeling is the ideal choice. Physical modeling realistically reflects real-world phenomena which are the most familiar to users and matches their expections. However, faithful physical modeling requires intense computing power, which is usually far from what is currently available on mobile devices. Two categories of challenges are present in the playing mode. The first is how to capture and process complex user input, and the second is how to compute the rich multimodal feedback in real time. I present an algorithmic system that addresses both challenges. Figure 6.2 shows the pipeline of the system, in which the *input processing* module converts raw multi-touch inputs into interaction metadata that are used to drive the *sound synthesis* module, as well as animation and vibration as part of the *multimodal feedback*.

6.4.1 Input Processing

When users interact with musical instruments in real world, complex dynamics happen in three-dimensional (3D) space. However, a multi-touch screen can only capture interactions in 2D. To create a responsive and immersive experience, I need to close the gap between the two. Moreover, unlike traditional virtual applications on personal computers, where mouse and keyboard are usually the input devices, on multi-touch enabled devices, I are required to simultaneously track interaction of up to 10 fingers (i.e. 10 contact areas) with virtual objects, not to mention this all needs to happen with much more limited computing resources. Therefore, I are present with the challenge to effectively reconstruct, approximate, and represent 3D interaction in real time.

6.4.1.1 Reconstruct 3D Interaction

I reconstruct 3D interaction solely based on the limited 2D touch-screen events exposed by consumer tablets. Such touch events are usually registered as *touch begin*, *touch continue*, and *touch end*. Accompanied with these events are *touch position* (an obsolute position x pixels right and y pixels down from the upperleft corner of the screen) and *touch area* (the square area in pixels that approximates a finger touch area on the screen). With these limited 2D data, I derive 3D interaction data critical to generating sound effects as follows.

Position of the contact. Hitting an instruments at different positions generates different sound effects. This data is retrieved by shooting a ray from the camera position in the virtual 3D world to the touch position mapped on the viewport and computing the intersection between this ray and virtual instruments. The intersection point is used as the contact position.

Type of the interaction. An interaction can be either a *strike* or a *slide*. A *strike* has a single contact position, while a *slide* is a continuous contact. They have different computation complexities. If I treat a slide the same as a strike, I would be constantly tracking and computing intersections with virtual objects, which is expensive. In addition, the problem is exacerbated as I support ten fingers interacting with a large number of virtual objects.

Based on the *touch begin* event, I distinguish between the two types of interactions. On a touch begin event, if the touch position is intersecting with any virtual object, I identify it as a strike interaction. If the device sees a *touch continue* event, it treats the interaction as a sliding event. For a sliding event, the intersection is computed very efficiently with the *input approximation* described in Section 6.4.1.2.

Force of the contact. The maginitude of the force is an important variable, because a light touch should only induce a low volume musical tone, while a forceful strike or slide action should produce a loud sound. However, multi-touch screens on consumer mobile devices usually do not have capabilities to capture force or pressure. For strike interaction, I use the *touch area* returned by touch screens to emulate the magnitude of the applied force. This is similar to the input handling introduced in (Ren et al., 2012b). For a slide action, I use the *velocity* derived from the traveled distance of a finger and scale the magnitude of force proportionally.

Table 6.1: Are responses as expected? Scale: 0 (No) to 10 (Yes)

String: sound variation with different hit points	Bar: sound variation with different hit points	String: shape change when changing pitch	Bar: shape change when changing pitch	String: sound variation with different strike velocities	Bar: sound variation with different strike velocities
7.33 ± 1.86	8.00 ± 0.89	9.17 ± 0.75	9.17 ± 0.75	6.50 ± 3.02	5.50 ± 3.02

6.4.1.2 Input Approximation

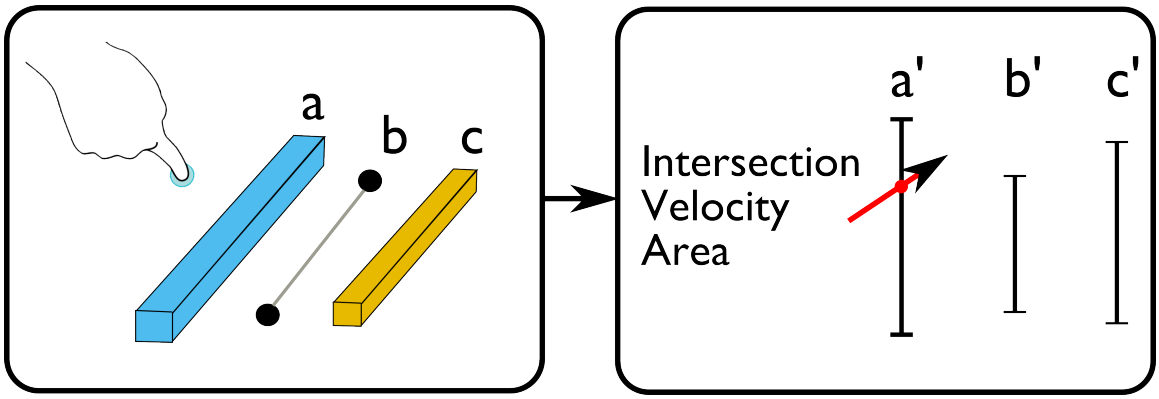


Figure 6.3: **Input Approximation:** Dimentionality reduction that abstracts and represents a 3D space configuration with a 2D one.

As alluded in Section 6.4.1.1, faithfully computing continuous intersections between touch events from ten fingers and all the virtual objects is expensive. Moreover, this computation has to happen in real time (30 frames per second) with only a small percentage of the available processing resources. In this case, directly computing these intersections in 3D is infeasible, because the user experience is unacceptable due to lag and unresponsiveness.

I propose an input approximation that reduces dimentionality and abstracts and represents the 3D interactions effectively with a 2D configuration. Figure 6.3 illustrates an example of such approximation process. On the left, a user is interacting with the multi-touch screen, which displays the virtual instruments, namely Bar *a*, String *b*, and Bar *c*. Without any approximation, in each frame, I would be casting a ray from the camera position to the touch point in the viewport and then compute the intersection between the ray with all three objects in 3D space. On the right, I present the approximation which speeds up this intersection computation. I first compute the bounding box of a virtual instrument and then represent the original instrument geometry with a line segment that spans

the longest dimension of its bounding box. As shown in the image on the right, Bar a, String b, and Bar c are respectively approximated by Line a', b', and c'. When a *touch continue* event happens, I identify it as a sliding action as described in Section 6.4.1.1 and compute an *interaction line segment*, which starts at the projected touch position in last time step and ends at the projected touch position in the current time step (shown as the red arrow in Figure 6.3, where the arrow indicates the direction of this interaction). Once I represent 3D instruments and touch events as line segments, I compute inteserction between the instrument and interaction line segments, and the intersections in 3D space are approximated as contact locations in the 2D configuration (shown as the red dot in the right image in Figure 6.3). Based on the sliding velocity and contact area, I also scale the applied force appropriately. The described input approximation significantly reduces computation complexity and guarantees performant response computation that is critical to real-time applications.

6.4.2 Sound Synthesis

In order to build a virtual musical instrument system that automatically and correctly responds to users' dynamic and rich interaction, I need to compute sound samples in real time with physical modeling. This is not the case for most existing mobile music applications, which usually play recorded or pre-computed sounds at run-time. Moreover, to render sound effects that are pleasant to human ears, the sampling rate needs to be $44000Hz$, which means a sound sample is computed every 0.023 milli-second! Given the stringent compute requirement and tight resource budget, extremely fast sound synthesis algorithms are required. I adopt the *waveguide synthesis* methods described in (Cook, 2007). Specifically, string instruments are modeled with the bowed string physical model introduced in (Jaffe and Smith, 1995). For bar instruments, a banded waveguide model for bowed bars (Essl and Cook, 1999) is used. When a strike happens, one single impulse is exerted to the sound synthesis model. For a slide, a series of impulses are applied corresponding to a series of contact points.

6.4.3 Implementation Details

The virtual musical instrument system described in this chapter is implemented on a consumer mobile device, the Google Nexus 10 tablet with a Dual-core A15 mobile CPU and 2GB of RAM. Android 4.3 Jelly Bean (Google, 2013) is the mobile operating system. To render sound, I use the

audio library in Android SDK and render audio at 44,100 Hz. Graphical rendering and animation are run at 30 frames per second.

6.5 Pilot Study

In order to evaluate the effectiveness of the proposed virtual musical instrument system, I performed a pilot study with six subjects: five male, one female, and age between 28 and 34. All results presented in this section are data averaged over the six subjects with the standard deviation appended. At the start the study, I briefly demonstrate the functionality of the mobile application and present subjects with a questionnaire. Subjects can answer the questions at any time in the duration of the study. I observed all subjects quickly learned how to use the application immediately and started reconfiguring and playing the virtual musical instruments right away. Specifically, I asked the subjects to evaluate how easy the process was, and the result is shown in Table 6.2, and subjects generally considered it easy to operate the application.

Table 6.2: Is it easy to do the following? 0 (Difficult) to 10 (Easy)

Easy to generate sounds?	Easy to pick the desired instrument?	Easy to modify the chosen instrument?
9.17 ± 0.98	7.67 ± 2.25	8.33 ± 2.25

I evaluate how realisitc and natural the generated responses are by asking our subjects to score if the sound and geometry change match what they expect when they vary their interaction with the instruments. These questions and results are listed in Table 6.1. Sound variation with different hit points and shape change when changing pitch both scored high. While subjects still considered the sound variation with different strike velocities as expected, the score is lower. I suspect, due to the small-screen real estate on mobile devices, it is difficult for users to vary strike velocities considerably.

I also hope to measure the quality of the synthesized sounds. I asked if the generated sounds were realistic. 0 means ‘no’, and 10 means ‘yes’. For string instruments, subjects responded with 8.50 ± 0.84, and for bar instruments, 7.50 ± 1.38. In addition, our subjects observed low latency when using the application. When asked ”can you observe any latency” (0 means obsolutely yes, and 10 means obsolutely no latency), our subjects rated this 7.38 ± 2.93. Last but not least, I evaluate the

multimodal experience. Our subjects are able to toggle on and off the visual feedback (animation of the instruments) and the haptic feedback (vibration of the device). I asked subjects which mode of experience do they prefer. Four out of the six subjects prefer the experience with only auditory and visual feedback, one likes all modals turned on, and one likes only auditory feedback on. I further asked the four subjects why they preferred the haptic feedback off, and they all responded that the device's vibration is too strong and generates a buzzing sound that interferes with the auditory feedback.

6.6 Conclusion and Future Work

I present a real-time physically based virtual musical instrument system that is configurable and reallistically responds to user interaction. With efficient input handling and sound synthesis algorithms, I achieved this on a consumer mobile device with limited computing resources. The presented system allows intuitive and expressive music playing and rapid prototyping of sounding virtual instruments. The real-time interaction also encourages users to actively explore sound effects determined by physical parameters like materials and geometry.

Next, I hope to add fully-featured and more complex shape editing, so that users can scale and sculpt virtual sounding objects on the fly. With an arbitrary geometry created by users, modal synthesis will be adopted for generating sounds. As a result, I plan to study efficient modal analysis methods that can compute and represent sound models for arbitrary shapes at interactive rates. Providing user with an intuitive way of editing material can also be explored. Last but not least, I plan to perform a more thorough user study that evaluates the effectiveness of future virtual musical instrument systems that I present here.

CHAPTER 7: CONCLUSION

Throughout this dissertation, I have addressed challenges in multiple aspects regarding real-time physically based sound synthesis and building a richly responsive multimodal experience on both multi-touch tabletops and mobile devices. To summarize the main results presented here:

Evaluation of geometry-invariant property In modal sound synthesis, the Rayleigh damping model is assumed to achieve real-time performance. I present both an empirical and a psychoacoustic study that evaluate the geometry-invariant property of this model and show that Rayleigh damping model can largely be considered geometry invariant. With this discovery, modal synthesis can be more widely adopted for real-time sound synthesis.

Example-guided modal sound synthesis I propose a novel framework that takes on example audio clip of an impact sound and automatically estimates the material parameters under modal sound synthesis model using Rayleigh damping. The estimated material parameters can be directly applied to different geometries, and the resulting sound automatically reflect the geometry change and interaction variation at runtime. This framework simplifies the process of adding sound effects driven by visual simulation in games, animation, and other virtual environment applications. With this framework, it is now feasible to extend physically-based sound synthesis algorithms like modal synthesis to complex scenarios that involve a big number of sounding objects of various materials and geometries, which is always the case for real-world applications like movies, video games, and simulators.

Efficient contact sound synthesis Through taking into consideration texture information in visual rendering, I proposed a three-level surface representation. With simulation on all three levels, I achieved real-time synthesis of continuous contact sounds. The resulting sound effects closely correlates with the visual rendering of the contacting surfaces and largely diminishes the discrepancy between visual and audio feedbacks, which was not addressed by previous

work. With this proposed method, real-time physically-based simulation of complex contact interaction is feasible.

Real-time sound synthesis driven by multi-touch Virtual musical instrument applications have been designed and implemented on multi-touch-enabled devices, i.e. both a tabletop system and commodity tablet devices. Touch input is interpreted and used to expressively model users' musical performances and used to drive both visual rendering and sound simulation. A multimodal interaction that couples feedbacks in multiple senses (i.e. sigh, hearing, and touch) is achieved.

7.1 Future Work

There are many exciting avenues for future work. First, it is worthwhile to investigate in other damping models beside the Rayleigh damping model. A more general or high-degree damping model can potentially simulate more complex internal friction in materials and create more sophisticated sound effects. Secondly, I have only looked at simulating sounds with linear dynamic formulation. It would be important to study if nonlinear effects can be approximated and still achieve real-time performances. For the example-guided material estimation framework, currently only one audio clip is used as the example. However, if multiple audio clips of the same object excited at different locations are provided, it would be meaningful to extend the current framework and evaluate if even better estimation can be acquired. In the virtual musical instrument setup, I am currently modeling the hit action in percussion instrument playing. Other types of user performance behaviors like damping and rubbing can be modeled. Lastly, more thorough user evaluation on the effectiveness of the provided multimodal interaction can be done. It would be important to further study how humans perceive visual, audio, and touch feedback in a multimodal virtual environment setup.

REFERENCES

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

Adrien, J.-M. (1991). Representations of musical signals. chapter The missing link: modal synthesis, pages 269–298. MIT Press, Cambridge, MA, USA.

Algoriddim (2014). djay for ipad. `http://www.algoriddim.com/djay-ipad`.

Apple (2014). Garageband for ios. `http://www.apple.com/ios/garageband/`.

Audiokinetic (2011). Wwise SoundSeed Impact.

Barrass, S. and Adcock, M. (2002). Interactive granular synthesis of haptic contact sounds. In *AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio, Helsinki University of Technology, Espoo, Finland, 15th-17th June*.

Baxter, B., Scheib, V., Lin, M. C., and Manocha, D. (2001). Dab: interactive haptic painting with 3d virtual brushes. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 461–468. ACM.

Bilbao, S. (2009). *Numerical Sound Synthesis*. Wiley Online Library.

Bonneel, N., Drettakis, G., Tsingos, N., Viaud-Delmon, I., and James, D. (2008a). Fast modal sounds with scalable frequency-domain synthesis. *ACM Transactions on Graphics (TOG)*, 27(3):24.

Bonneel, N., Drettakis, G., Tsingos, N., Viaud-Delmon, I., and James, D. (2008b). Fast modal sounds with scalable frequency-domain synthesis. In *ACM SIGGRAPH 2008 papers*, SIGGRAPH '08, pages 24:1–24:9, New York, NY, USA. ACM.

Bonneel, N., Suied, C., Viaud-Delmon, I., and Drettakis, G. (2010). Bimodal perception of audio-visual material properties for virtual environments. *ACM Transactions on Applied Perception (TAP)*, 7(1):1.

Boyd, J. P. (2001). *Chebyshev and Fourier Spectral Methods: Second Revised Edition*. Dover Publications, 2 revised edition.

Brebbia, C. A. (1991). *Boundary Element Methods in Acoustics*. Springer, 1 edition.

Bruyns, C. (2006). Modal synthesis for arbitrarily shaped objects. *Computer Music Journal*, 30(3):22–37.

Buxton, W., Hill, R., and Rowley, P. (1985). Issues and techniques in touch-sensitive tablet input. In *ACM SIGGRAPH Computer Graphics*, volume 19, pages 215–224. ACM.

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

Chadwick, J. N. and James, D. L. (2011). Animating fire with sound. In *ACM Transactions on Graphics (TOG)*, volume 30, page 84. ACM.

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*, 14:1707–1722.

Chuchacz, K., O'Modhrain, S., and Woods, R. (2007). Physical models and musical controllers: designing a novel electronic percussion instrument. In *NIME '07: Proceedings of the 7th international conference on New interfaces for musical expression*, pages 37–40, New York, NY, USA. ACM.

Cook, P. (2002a). *Real sound synthesis for interactive applications*. AK Peters, Ltd.

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

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

Cook, P. R. (2002b). *Real Sound Synthesis for Interactive Applications*. A. K. Peters, Ltd., Natick, MA, USA.

Cook, P. R. (2007). Real sound synthesis for interactive applications (book & cd-rom).

Corbett, R., van den Doel, K., Lloyd, J. E., and Heidrich, W. (2007). Timbrefields: 3d interactive sound models for real-time audio. *Presence: Teleoperators and Virtual Environments*, 16(6):643–654.

Davidson, P. and Han, J. (2006). Synthesis and control on large scale multi-touch sensing displays. In *Proceedings of the 2006 conference on New interfaces for musical expression*, pages 216–219. IRCAMCentre Pompidou.

Deines, E., Michel, F., Bertram, M., Hagen, H., and Nielson, G. (2006). Visualizing the phonon map. In *Eurovis*.

Dobashi, Y., Yamamoto, T., and Nishita, T. (2003). Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics. *ACM Trans. Graph.*, 22(3):732–740.

Dobashi, Y., Yamamoto, T., and Nishita, T. (2004). Synthesizing sound from turbulent field using sound textures for interactive fluid simulation. In *Computer Graphics Forum*, volume 23, pages 539–545. Wiley Online Library.

Doel, K. and Pai, D. (1998). The sounds of physical shapes. *Presence*, 7(4):382–395.

Essl, G. and Cook, P. R. (1999). Banded waveguides: Towards physical modeling of bowed bar percussion instruments. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 321–324.

Fontana, F. (2003). *The sounding object*. Mondo Estremo.

framework Named pipes, M. I. (2011). Named pipes.

Funkhouser, T., Tsingos, N., Carlbom, I., Elko, G., Sondhi, M., West, J. E., Pingali, G., Min, P., and Ngan, A. (2004). A beam tracing method for interactive architectural acoustics. *The Journal of the Acoustical Society of America*, 115(2):739–756.

Funkhouser, T., Tsingos, N., and Jot, J.-M. (2003). Survey of methods for modeling sound propagation in interactive virtual environment systems. *Presence and Teleoperation*.

Gaver, W. (1988). *Everyday listening and auditory icons*. PhD thesis, University of California, San Diego.

Giordano, B. 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*, 119:1171.

Google (2013). Android 4.3 jelly bean operating system. `http://www.android.com/about/jelly-bean/`.

Guski, R. and Troje, N. (2003). Audiovisual phenomenal causality. *Perception & psychophysics*, 65(5):789.

Han, J. (2005). Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118. ACM.

Hochenbaum, J. and Vallis, O. (2009). Bricktable: A musical tangible multi-touch interface. *Proceedings of Berlin Open Converence 09*.

Howell, D. (2009). *Statistical methods for psychology*. Wadsworth Pub Co.

Jaffe, D. A. and Smith, J. O. (1995). Performance expression in commuted waveguide synthesis of bowed strings. In *In ICMC*, pages 343–346.

James, D., Barbič, J., and Pai, D. (2006a). Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. In *ACM SIGGRAPH 2006 Papers*, page 995. ACM.

James, D. L., Barbič, J., and Pai, D. K. (2006b). Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 987–995, New York, NY, USA. ACM.

Kaltenbrunner, M., Jorda, S., Geiger, G., and Alonso, M. (2006). The reactable*: A collaborative musical instrument.

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

Krotkov, E., Klatzky, R., and Zumel, N. (1996). Analysis and synthesis of the sounds of impact based on shape-invariant properties of materials. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 1, pages 115–119. IEEE.

Krotkov, E., Klatzky, R., and Zumel, N. (1997). Robotic perception of material: Experiments with shape-invariant acoustic measures of material type. *Experimental Robotics IV*, pages 204–211.

Lakatos, S., McAdams, S., and Caussé, R. (1997). The representation of auditory source characteristics: Simple geometric form. *Attention, Perception, & Psychophysics*, 59(8):1180–1190.

Lloyd, D. B., Raghuvanshi, N., and Govindaraju, N. K. (2011). Sound Synthesis for Impact Sounds in Video Games. In *Proceedings of Symposium on Interactive 3D Graphics and Games*.

McAdams, S., Chaigne, A., and Roussarie, V. (2004). The psychomechanics of simulated sound sources: Material properties of impacted bars. *Journal of The Acoustical Society of America*, 115.

Miranda, E. and Wanderley, M. (2006). *New digital musical instruments: control and interaction beyond the keyboard*. AR Editions, Inc.

Moss, W., Yeh, H., Hong, J., Lin, M., and Manocha, D. (2010). Sounding Liquids: Automatic Sound Synthesis from Fluid Simulation. *ACM Transactions on Graphics (TOG)*.

Nordahl, R., Serafin, S., and Turchet, L. (2010). Sound synthesis and evaluation of interactive footsteps for virtual reality applications. In *Virtual Reality Conference (VR), 2010 IEEE*, pages 147 –153.

NVIDIA (2013). Nvidia physx. `http://www.nvidia.com/object/physx_new.html`.

O'Brien, J., Shen, C., and Gatchalian, C. (2002a). Synthesizing sounds from rigid-body simulations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 175–181. ACM.

O'Brien, J. F., Cook, P. R., and Essl, G. (2001). Synthesizing sounds from physically based motion. In *Proceedings of ACM SIGGRAPH 2001*, pages 529–536. ACM Press.

O'Brien, J. F., Shen, C., and Gatchalian, C. M. (2002b). Synthesizing sounds from rigid-body simulations. In *The ACM SIGGRAPH 2002 Symposium on Computer Animation*, pages 175–181. ACM Press.

O'Brien, J. F., Shen, C., and Gatchalian, C. M. (2002c). Synthesizing sounds from rigid-body simulations. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 175–181. ACM.

Pai, D., Van Den Doel, K., James, D., Lang, J., Lloyd, J., Richmond, J., and Yau, S. (2001). Scanning physical interaction behavior of 3D objects. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 87–96.

Picard, C., Tsingos, N., and Faure, F. (2009). Retargetting example sounds to interactive physics-driven animations. In *AES 35th International Conference-Audio for Games, London, UK*.

Quatieri, T. and McAulay, R. (1985). Speech transformations based on a sinusoidal representation. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '85.*, volume 10, pages 489 – 492.

Raghuvanshi, N. and Lin, M. (2006a). Interactive sound synthesis for large-scale environments. *ACM Symposium on Interactive 3D Graphics and Games*, pages 101–108.

Raghuvanshi, N. and Lin, M. C. (2006b). 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. (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.*, 29:68:1–68:11.

Rayleigh, B. (1945). *The theory of sound*, volume 2. Reprinted: Dover, New York.

Ren, Z., Mehra, R., Coposky, J., and Lin, M. C. (2012a). Tabletop ensemble: touch-enabled virtual percussion instruments. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '12, pages 7–14, New York, NY, USA. ACM.

Ren, Z., Mehra, R., Coposky, J., and Lin, M. C. (2012b). Tabletop ensemble: touch-enabled virtual percussion instruments. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 7–14. ACM.

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

Ren, Z., Yeh, H., and Lin, M. C. (2013). Example-Guided Physically Based Modal Sound Synthesis. *ACM Transactions on Graphics*, 32(1).

Roads, C. (2004). *Microsound*. The MIT Press.

Rosenberg, I. and Perlin, K. (2009). The unmousepad: an interpolating multi-touch force-sensing input pad. In *ACM SIGGRAPH 2009 papers*, pages 1–9. ACM.

Sakamoto, S., Ushiyama, A., and Nagatomo, H. (2006). Numerical analysis of sound propagation in rooms using the finite difference time domain method. *The Journal of the Acoustical Society of America*, 120(5):3008.

Schöening, J., Hook, J., Motamedi, N., Olivier, P., Echtler, F., Brandl, P., Muller, L., Daiber, F., Hilliges, O., Loechtefeld, M., Roth, T., Schmidt, D., and von Zadow, U. (2009). Building interactive multi-touch surfaces. *Journal of Graphics, GPU, and Game Tools*, 14(3):35–55.

Serra, X. (1997). Musical sound modeling with sinusoids plus noise. *Musical signal processing*, pages 497–510.

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

Shabana, A. (1997). *Vibration of discrete and continuous systems*. Springer Verlag.

Si, H. (2011). TetGen: A Quality Tetrahedral Mesh Generator and a 3D Delaunay Triangulator.

Sreng, J., Bergez, F., Legarrec, J., Lécuyer, A., and Andriot, C. (2007). Using an event-based approach to improve the multimodal rendering of 6dof virtual contact. In *VRST '07: Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, pages 165–173. ACM.

Steel, R. and Torrie, J. (1960). *Principles and procedures of statistics: with special reference to the biological sciences*. McGraw-Hill Companies.

Streeting, S., Muldowney, T., O'Sullivan, J., van der Laan, W. J., Doyle, J., and Xie, J. (2005). Ogre: Object-oriented graphics rendering engine. `http://www.ogre3d.org/`.

Takala, T. and Hahn, J. (1992). Sound rendering. In *ACM SIGGRAPH Computer Graphics*, volume 26, pages 211–220. ACM.

Thompson, L. L. (2006). A review of finite-element methods for time-harmonic acoustics. *The Journal of the Acoustical Society of America*, 119(3):1315–1330.

Trebien, F. and Oliveira, M. (2009). Realistic real-time sound re-synthesis and processing forinteractive virtual worlds. *The Visual Computer*, 25:469–477.

Välimäki, V., Huopaniemi, J., Karjalainen, M., and Jánosy, Z. (1996). Physical modeling of plucked string instruments with application to real-time sound synthesis. *Journal of the Audio Engineering Society*, 44(5):331–353.

Välimäki, V. and Tolonen, T. (1997). Development and calibration of a guitar synthesizer. *PREPRINTS-AUDIO ENGINEERING SOCIETY*.

van den Doel, K., Kry, P., and Pai, D. (2001a). FoleyAutomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 537–544. ACM New York, NY, USA.

van den Doel, K., Kry, P. G., and Pai, D. K. (2001b). Foleyautomatic: physically-based sound effects for interactive simulation and animation. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 537–544. ACM.

van den Doel, K., Pai, D., Adam, T., Kortchmar, L., and Pichora-Fuller, K. (2002). Measurements of perceptual quality of contact sound models. In *Proc. of the International Conference on Auditory Display (ICAD 2002), Kyoto, Japan*, pages 345–349.

van den Doel, K. and Pai, D. K. (1998a). The sounds of physical shapes. *Presence: Teleoper. Virtual Environ.*, 7:382–395.

van den Doel, K. and Pai, D. K. (1998b). The sounds of physical shapes. *Presence*, 7(4):382–395.

Vangorp, P., Laurijssen, J., and Dutré, P. (2007). The influence of shape on the perception of material reflectance. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, New York, NY, USA. ACM.

Weinberg, G. and Driscoll, S. (2007). The interactive robotic percussionist: new developments in form, mechanics, perception and interaction design. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, HRI '07, pages 97–104, New York, NY, USA. ACM.

Wildes, R. and Richards, W. (1988). Recovering material properties from sound. *Natural computation*, pages 356–363.

XAudio2, M. (2011). XAudio2.

Zheng, C. and James, D. L. (2009). Harmonic fluids. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pages 1–12, New York, NY, USA. ACM.

Zheng, C. and James, D. L. (2010a). Rigid-body fracture sound with precomputed soundbanks. *ACM Trans. Graph.*, 29:69:1–69:13.

Zheng, C. and James, D. L. (2010b). Rigid-body fracture sound with precomputed soundbanks. In *ACM SIGGRAPH 2010 papers*, SIGGRAPH '10, pages 69:1–69:13, New York, NY, USA. ACM.

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