Expressive Sound Synthesis for Animation

Cécile Picard-Limpens

UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS

ÉCOLE DOCTORALE STIC

SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

T H È S E

pour obtenir le titre de

Docteur en Sciences

de l'Université de Nice - Sophia Antipolis

Mention : INFORMATIQUE

Présentée et soutenue par

Cécile PICARD LIMPENS

Expressive Sound Synthesis for Animation

Thèse dirigée par George DRETTAKIS, François FAURE, et Nicolas TSINGOS

préparée à l'INRIA Sophia Antipolis, Projet REVES
et à l'INRIA Rhône-Alpes, Projet EVASION

soutenue le 4 décembre 2009

Jury :

Rapporteurs :	Davide Rocchesso	-	Università IUAV, Venezia
	Xavier Serra	-	Universitat Pompeu Fabra, Barcelona
Directeurs :	George DRETTAKIS	-	INRIA Sophia Antipolis (Reves)
	François FAURE	-	INRIA Rhône-Alpes (Evasion)
	Nicolas TSINGOS	-	DOLBY Laboratories, CA, USA
President :	René Caussé	-	IRCAM, Paris
Examinateurs :	Sylvain Marchand	-	LaBRI Université Bordeaux I

Quand on écoute le tambour, on entend le silence.
Braque, Le jour et la nuit - Cahiers, 1917-1952.

A i Miei.

Acknowledgments

I would like to thank my thesis supervisors, George Drettakis, Nicolas Tsingos, and François Faure, for their support and their confidence.

I am grateful to Marie-Paule Cani for having me allowed to conduct my research project in the EVASION team. I would like to thank the members team for their warm welcome and their very fruitful interactions.

I have to acknowledge Paul G. Kry for having me welcomed in his lab at the School of Computer Science, McGill University in Montreal. I had the great opportunity to collaborate with him, which was a very productive period of time.

I appreciate feedback and comments from my reviewers Davide Rocchesso and Xavier Serra.

This work was partially funded by *Eden Games*[1], an *ATARI* game studio. We would like to thank David Alloza, Xavier Guillaume, Jean-Yves Geffroy, and *Eden Games* for their feedback at various stages of this project.

Sophia-Antipolis, October 2009.

Cécile Picard-Limpens.

[1] Eden Games: http://www.eden-games.com/

Contents

Chapter 1

Introduction

Contents

The network of computers that surrounds us is above all a means of information. The codified information can be used to control machines, as for commercial and educational purposes. However, with games, training simulations, and other interactive virtual environment, this network becomes effectively a network of interactions, and space of immersion. Besides, it sometimes evolves into a space (and non-space) of interpersonal actions such as in collaborative games. The main goal of virtual environments is to convey the full experience of being *there* to the user. This generally involves considering the *action-perception loop* from which a perception space originates. Virtual scene should consequently exhibit convincing graphics, audio and behavior.

From the perspective of audio perception, current virtual environments generally use inadequate audio-visual representations. Audio rendering is often limited to the playback of prerecorded samples, possibly processed with amplitude, pitch or filter-envelopes. Due to their static character, prerecordings can not manage sound rendering of the large variety of situations that are usually dealt with, and particularly in current video games. Recent work has concentrated more on visual rendering, addressing both expressiveness and efficiency. Little effort has been made to perceptually optimize audio rendering. When objects are outside the view frustum, sound cues might become even more important in providing information about the circumstances of the virtual environment. Until recently, the primary focus for sound generation in virtual environments has been in spatialized sound effects.

Synthesizing a virtual environment in its entirety and for interactive applications requires a large amount of computational effort that should be divided among the underlined perceptual attributes. Accuracy, efficiency and flexibility are consequently the main requirements for synthesis engines. Physics-driven animation and interaction are becoming fundamental, especially for game engines. To ensure audio-visual synchrony and consistency for persuasive virtual scenes, related auditory events have to be dealt with. This involves the development of appropriate audio synthesis engines.

This thesis is concerned with the real-time synthesis of sounds resulting from physical interactions of various objects in a 3D virtual environment. These sounds are highly dynamic and vary notably according to the interaction type and the object properties. Their unpredictability makes them problematic to generate in a pre-production process, and implies specific synthesis algorithms that depend on real-time parameters. Virtual environments are, moreover, becoming extensively complex with a large number of simultaneous sound sources which may simulate even more precise physical behavior. Sound rendering consequently results in a high computational expense which cannot be supported by current solutions.

Starting from the observation that physics-driven animation and interaction are more and more central to virtual environments, the first goal of this thesis is to develop new physically based algorithms, which enable the sound rendering of a variety of interactions. In particular, we aim at efficient sound modeling depending on the properties of the force that arises between interacting objects and on the attributes of the resonating objects. The second goal of this thesis is to address sound design. For this purpose, flexibility of sound modeling is addressed, in order to allow the combination of sound design and consistency between the actions of the player. Procedural audio is adopted which consists in generating the audio at the position of use according to runtime parameters from the context and object behaviors. Finally, the third goal of this thesis aims at suggesting an adequate combination of physically based and empirical models, namely *hybrid* models, to add more realism in timbre especially when conditions of validity for a physically based model are no more fulfilled, such as in the case of nonlinear behavior.

1.1 Thesis Outline

This thesis is divided into four parts. The first part will present the fundamental principles of the physics of wave sound and the mechanisms of sound perception (Chapter 2), and will introduce the basic concepts related to audio for virtual reality and computer games (Chapter 3). Traditional approaches and possible strategies to render sound for virtual environments will be reviewed. The main techniques involved in physically based animation will then be presented. Finally, we will examine how control of sound synthesis can be efficiently managed, which will open the directions pursued in this thesis.

The second part will address sound source rendering with physically based models. Sound sources are studied by separating the source into the excitation which supplies energy to the system and the resonator which receives the energy. Previous work in contact modeling will be first reviewed (Chapter 4), and we will present a new synthesis method for complex contact sounds in the context of interactive simulations (Chapter 5). Past research in resonator modeling (Chapter 6) will be examined and we will introduce our robust and multi-scale approach for modal analysis relative to sound modeling (Chapter 7).

The third part will direct sound source rendering with an example-based model. Signal-based techniques will first be reviewed (Chapter 8). We will then present a novel signal-based approach for sound rendering that narrows the division between direct playback of recordings and physically based synthesis. The technique consists in retargetting audio grains extracted from recordings according to the output of a physics engine (Chapter 9).

The fourth part will introduce our prospects for a hybrid model. Our purpose will be to appropriately combine a physically based approach and an empirical approach. We will first present the motivation that guided this model (Chapter 10). We will then apply our perspectives onto a specific scenario, fracture sounds, and we will detail the parametrization of its hybrid model for sound rendering (Chapter 11).

Finally, we will conclude on the presented contributions and we will propose extensions and possible applications of this research.

1.2 Publications

The body of this thesis is part of three publications:

- *Audio texture synthesis for complex contact interactions* [Picard 2008]. Cécile Picard, Nicolas Tsingos and François Faure. The Fifth Workshop On Virtual Reality Interaction and Physical Simulation, VRIPHYS 08, Grenoble, France.
- *Retargetting Example Sounds to Interactive Physics-Driven Animations* [Picard 2009b]. Cécile Picard, Nicolas Tsingos and François Faure. AES 35th International Conference - Audio for Games, London, United Kingdom.
- *A Robust And Multi-Scale Modal Analysis For Sound Synthesis* [Picard 2009a]. Cécile Picard, François Faure, George Drettakis and Paul G. Kry. The International Conference on Digital Audio Effects (DAFx-09), Como, Italy.

Part I

Sound and Virtuality

Chapter 2

Physical and Perceptual Approaches for Sound

Contents

Sound modeling starts with the understanding of what is sound and how it is perceived. In this Chapter, we present the main principles of the physics of wave sound and the mechanisms of sound perception. This chapter is mainly based on the book from Crowell [Crowell 1998].

2.1 Physics of Sound

Sound is a traveling wave conveyed by an oscillation of pressure that is transmitted through a solid, liquid, or gas. In gases, plasma, and liquids, sound travels as longitudinal waves, also called compression waves, that is, the local oscillations always move in the same direction as the wave. Through solids, however, it can be transmitted as both longitudinal and transverse waves. Longitudinal sound waves are waves of alternating pressure deviations from the equilibrium pressure, while transverse waves are waves of alternating shear stress at a right angle to the direction of propagation.

Sound waves in gases are well described by the experiment of the piston, see Figure 2.1. When the piston vibrates, the air particles move back and forth about their equilibrium position. This creates alternating zones of compression and rarefaction where the pressure is less than the normal undisturbed atmospheric pressure, denoted P_{atm}, in the rarefied region, whereas the pressure is greater in the compressed region. The disturbance travels while the individual particles do not.

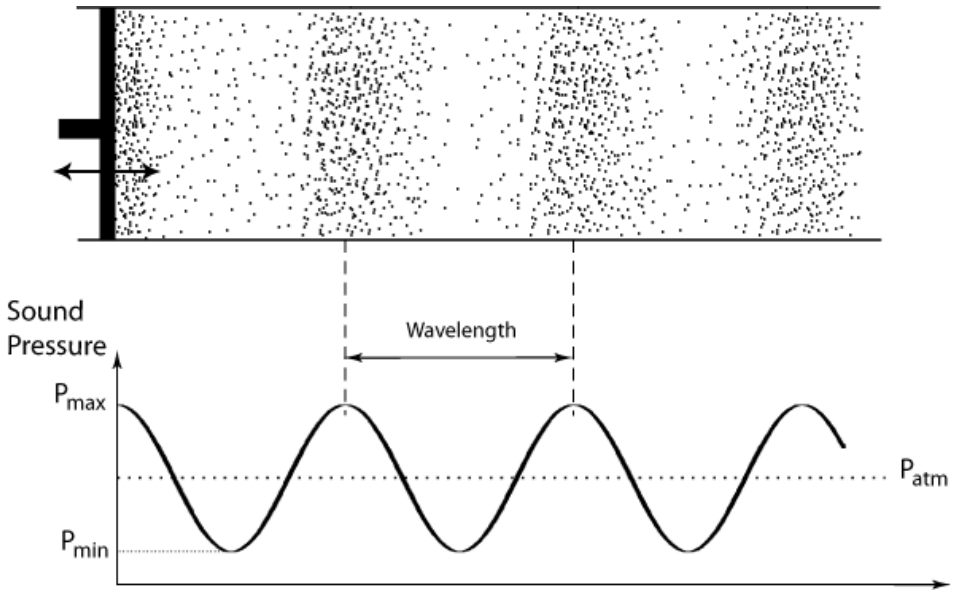

Figure 2.1: *A vibrating piston creates alternating zones of compression and rarefaction. This experiment illustrates sound waves.*

2.1.1 At the Origin of Sound, A Vibration

When hitting a teacup with a spoon, the teacup vibrates freely and creates a periodic motion in the medium corresponding to a pressure wave. Our eardrum is driven by the force of the wave and we hear a sound. The sound is actually composed of a number of tones whose frequencies are the natural frequencies of the teacup. The vibrations of the teacup after the excitation by an impact are called free vibrations.

Conservation laws make periodic motion a common observation in the world around us. Figure 2.2 shows the most basic example of a vibration, a spring attached to a wall on the left and to a mass on the right. If we assume friction is negligible and the motion of the mass is one-dimensional, only potential energy and kinetic energy are involved. Conservation of energy requires that the mass repeats its motion. In the same way, the energy carried by the sound wave converts back and forth between the potential energy of the extra compression (in the case of longitudinal waves) or lateral displacement strain (in the case of transverse waves) of the matter and the kinetic energy of the oscillations of the medium.

Small-amplitude vibrations are always sinusoidal. This type of vibration is called simple harmonic motion. In simple harmonic motion, the period do not relate to the amplitude, and is given by $\mathrm{T} = 2\pi\sqrt{m/k}$ where m and k are respectively the mass and the stiffness of the vibrator. Sinusoidal waves are the most important special case of periodic waves. Since the French mathematician Fourier demonstrated that any periodic wave with frequency f can be modeled as a superposition of sine waves with frequencies *f*, *2f*, *3f*, ..., sine waves are, by definition, the fundamental elements

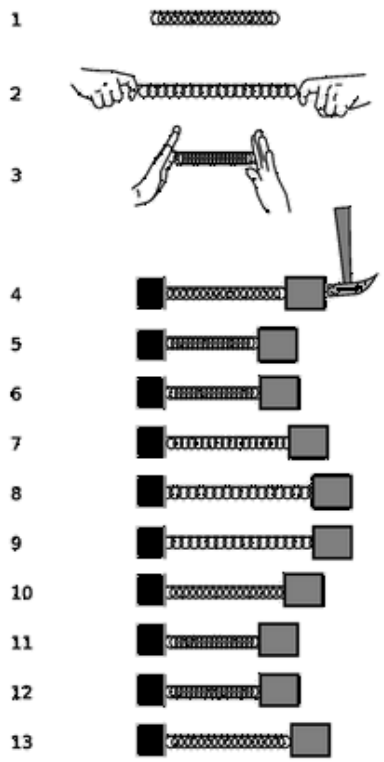

Figure 2.2: *A spring has an equilibrium length, 1, and can be stretched, 2, or compressed, 3. A mass attached to the spring can be set into motion initially, 4, and will then vibrate, 4-13. Image courtesy of Benjamin Crowell [Crowell 1998].*

of all waves.

Given that the amplitude is small, the energy of a vibration is always proportional to the square of the amplitude. A vibrating system loses energy for various reasons, such as the emission of sound. This effect, called damping, induce the exponential decay of the vibrations unless energy is newly injected into the system to replace the loss. A driving force that supply energy into the system may animate the system at its own natural frequency or at some other frequency, according to the type of excitation. When the driving force fits the natural frequency of vibration, the amplitude of the steady-state response is greatest in proportion to the amount of driving force. Finally, when a system is driven at resonance, the steady-state vibrations have an amplitude that is proportional to its quality factor Q, defined as the number of cycles required for the energy to fall off by a factor of 535.

The Wave Motion

The propagation of sound in the atmosphere is described by the wave equation:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u \tag{2.1}$$

where ∇^2 is the Laplacian and where c is a fixed constant equal to the propagation speed of the wave.

The magnitude of a wave's velocity depends on the properties of the medium. As an example, sound waves travel at about 340 m/s in air and 1000 m/s in helium. The energy of the wave relates to its amplitude and frequency, not to its speed. Thus, the sound waves from an exploding stick of dynamite carry a lot of energy, but are no faster than any other waves. Also, the wave velocity is related to frequency and wavelength by the equation v = λ f, and a wave emitted by a moving source will be shifted in wavelength and frequency, referred to as the Doppler effect. The shifted wavelength is given by the equation:

$$\lambda' = (1 - \frac{v_s}{v})\lambda \tag{2.2}$$

where v is the velocity of the waves and v_s is the velocity of the source. A similar shift occurs if the observer is moving, and in general the Doppler shift depends approximately only on the relative motion of the source and observer if their velocities are both small compared to the waves' velocity.

Waves spread out in all directions from every point on the disturbance that created them. If the dimensions of the sound source are much smaller than the wavelength of the emitted sound, the source can be represented by a *point source* or *monopole*. It will tend to radiate sound equally in all directions, that is to say, with

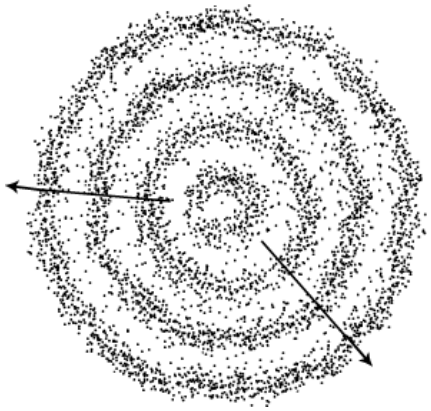

Figure 2.3: *An acoustic monopole generates sound by rhythmically expanding and contracting. The point source is a good approximation for the sound field radiated by a loudspeaker.*

spherical symmetry, see Figure 2.3. In practice, the point source model is a good approximation for the sound field radiated by a loudspeaker in a sealed box at low frequencies. When moving further from a source of spherical waves, the amplitude of the sound declines. The intensity I is the power W in the wave divided by the area A over which it is spread. Assuming that none of the sound wave power is absorbed as it propagates from the source to the listener, the area of the imaginary sphere over which the spherical wave is spread is A = $4\pi r^2$, r being the distance of the listener from the source. Thus, the intensity I of the source is related to the

power W by $I = W/(4\ \pi\ r^2)$. Thus, the intensity decreases as an *inverse-square law* with distance r, that is like $1/r^2$.

For a three-dimensional wave such as a sound wave, the wave patterns would be spherical waves and plane waves. Infinitely many patterns are possible, but linear or plane waves are often the simplest to analyze, because the velocity vector is the same in all directions. Since all the velocity vectors are parallel to one another, the problem is effectively one-dimensional.

2.1.2 Elaborate Sounds

Speech makes humans stand out from animals most decisively. Complex speech sounds are experimented even from birth. How are these sound waves controlled? Mostly this is done by changing the shape of a connected set of hollow cavities in our chest, throat, and head. By moving the boundaries of this space in and out, and due to specific properties of sound waves, many complex sounds can be produced. Indeed, sound waves can be subject to:

- Reflection, Transmission and Absorption. Whenever a wave encounters the boundary between two media in which its speeds are different, part of the wave is reflected and part is transmitted. During travel, the energy of the sound wave is also gradually converted into heat.
- Diffraction. The waves bend as they interact with obstacles in their path, especially when wavelengths are on the order of the diffracting object size.
- Interference. Two wave patterns can overlap in the same region of space and they add together where they coincide.

Standing Waves

Standing waves are induced in a solid medium at its resonant frequencies. They are of great importance since they are the free vibrations an object may emit after an impact. They may be created from two waves traveling in opposite directions. Unlike traveling waves, they do not cause a net transport of energy. These waves freely propagate in the surrounding medium producing the characteristic sound of the vibrating object. In music, they have been extensively studied. Indeed, to be a musical tone, that is, a sound with at particular pitch, a group of sound waves has to be very regular, all exactly the same distance apart. Thus, standing waves are produced in or on the musical instrument. In addition, considering that atoms are actually standing-wave patterns of electron waves, and that atoms make up all matter in existence, we are ourself standing waves!

By shaking a rope, standing waves arise, see Figure 2.4. The sine wave of a specific vibration mode is just automatically created when the right frequency is found. Figure 2.4 shows the four first vibration patterns of the rope. Each mode has a wavelength and a corresponding frequency related to the length of the string.

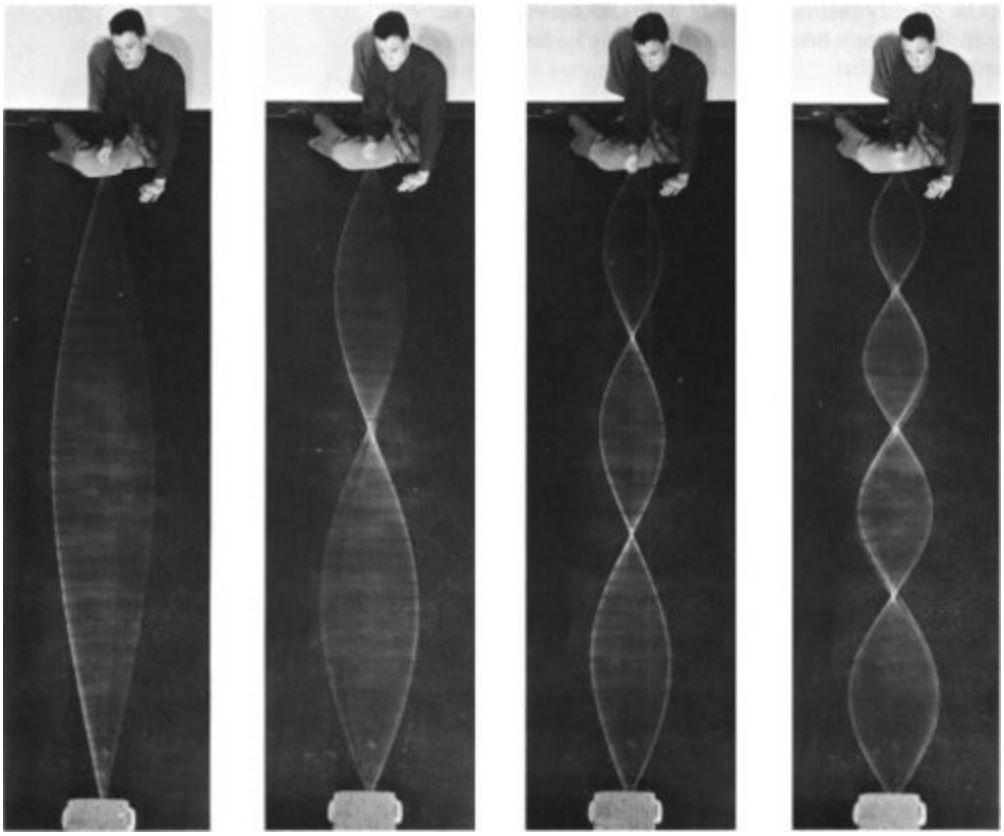

Figure 2.4: *Sinusoidal wave patterns made by shaking a rope. Image courtesy of Benjamin Crowell.*

In the same way, a rectangular membrane with fixed edges has specific vibration modes, see Figure 2.5. It suggests that the vibration modes can be seen as two-dimensional string modes since standing waves in one direction appear to be independent of standing waves in the other direction.

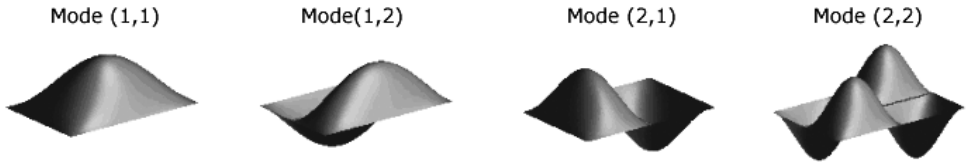

Figure 2.5: *The 4 first vibration modes for a rectangular membrane with fixed edges (Image courtesy of the ISVR).*

Each object can potentially vibrate and present vibration modes. Solutions can

be found for some simple systems. However, unlike the string, the natural modes are not always simply related to each other. As an example, membranes with fixed edges also vibrate in modes but the natural frequencies are unrelated. Standing wave phenomena can occur in space, such as in a room at specific frequencies called the resonance frequencies of the room. These depend on the dimension and shape of the room. At resonance, the acoustic response of the room will be enhanced.

2.2 Sound Perception

When an acoustic traveling wave generated by a vibration propagates through a medium, the auditory system perceives sound. The human auditory apparatus detects the vibration and transmits the stimuli to the brain which informs us about the nature of the sound and its localization. Thus, the goal of audio rendering is to create the illusion of a real sound scene for the listener. Research in human hearing provides us with the knowledge needed to focus on specific cues for sound illusion.

In this section, the anatomy of human hearing is first presented. Then, the physiological mechanism of the auditory system is described. Finally, multisensory integration is introduced.

2.2.1 Human Hearing

The ear is composed of three basic parts - the outer ear, the middle ear, and the inner ear, see Figure 2.6. Detection and interpretation of sound is allowed by the specific function of each part of the ear. Sound is collected and transmitted from the outer ear to the middle ear. The energy of the sound wave is transformed into the internal vibrations of the bone structure of the middle ear. A longitudinal wave is then produced in the inner ear from the vibrations of the bone structure. Finally, the energy of the longitudinal wave within the inner ear fluid is converted into nerve impulses which can be transmitted to the brain. The audible range of frequencies for human ears lies between about 20 and 20 000 cycles per second (20 Hz to 20 kHz). This corresponds in air to wavelength that range between roughly 17 mm (at 20 kHz) and 17 m (at 20 Hz), under room conditions.

The large dynamic range of human hearing is due to the non-linear behavior of the ear, and more specifically the inner ear non-linearity. This produces distortions that can be heard and measured in the ear canal. As a consequence, from two frequencies, *f1* and *f2*, present in a sound, the ear non-linearity would theoretically generate intermodulation products according to the sum and difference of the two frequencies, that is $m\ f1 \pm n\ f2$, for any whole numbers m and n. This is perceived when two sounds are played together at sufficient volume, the ear hears a fictitious difference frequency. In particular, this phenomenon allows psychoacousticians to investigate the nonlinearities of the ear and to check the healthiness of the auditory system.

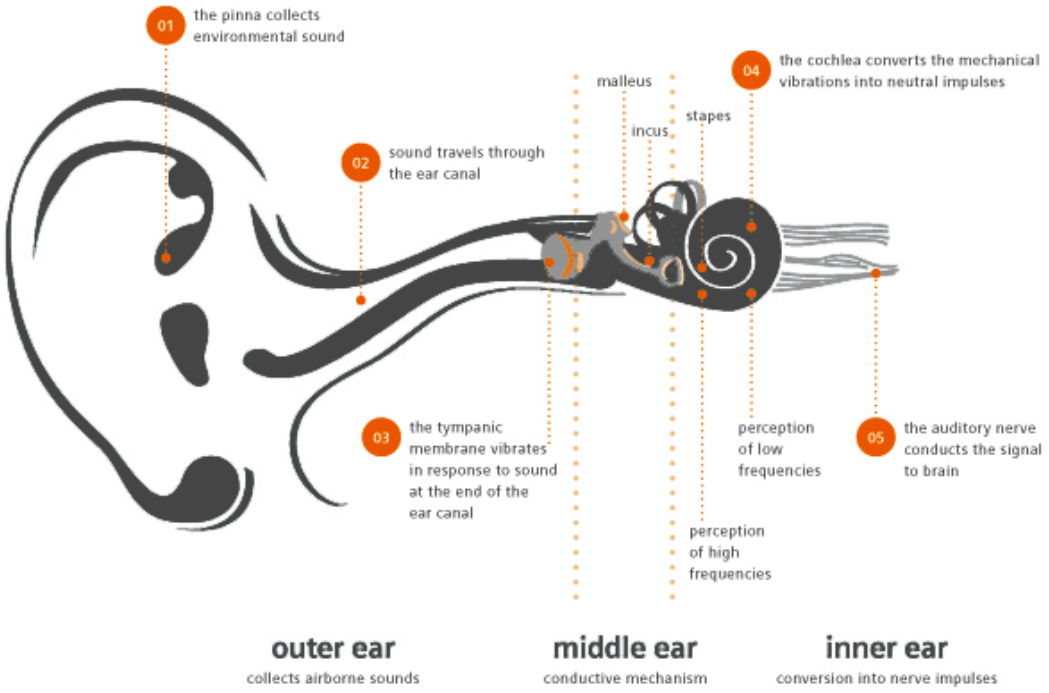

Figure 2.6: *The ear and its three parts: the outer ear, the middle ear, and the inner ear. Image courtesy of SIEMENS.*

2.2.2 Sound Pressure Level

Sound is an oscillation of pressure. Sound pressure refers to the the difference between the average local pressure of the medium outside of the sound wave, at a given point and a given time, and the pressure found within the sound wave itself within that same medium. Sound pressure is often expressed on a logarithmic decibel scale, due to the very wide range of amplitudes that the human ear can perceive. The sound pressure level (SPL) or Lp is defined as:

$$L_p = 10log_{10}(\frac{p^2}{{p_{ref}}^2}) = 20log_{10}(\frac{p}{p_{ref}})dB \tag{2.3}$$

where p is the root-mean-square sound pressure and *pref* is a reference sound pressure. Commonly used reference sound pressures, defined in the standard ANSI S1.1-1994, are 20 μPa in air and 1 μPa in water. To be a pressure level, a value in decibels automatically implies a reference sound pressure.

The human ear does not have a flat spectral response, and for this reason, sound pressures are often frequency weighted in order to adapt the measured level to the perceived levels. Several weighting schemes have been defined by the International Electrotechnical Commission (IEC). Among them, A-weighting attempts to match

the response of the human ear to noise and A-weighted sound pressure levels are labeled dBA.

The root mean square (RMS) value is also used to evaluate the sound pressure field. It is calculated by taking the square of the deviation from the equilibrium pressure and averaged over time and/or space. For example, 1 Pa RMS sound pressure (94 dB SPL) in atmospheric air implies that the actual pressure in the sound wave oscillates between (1 atm -$\sqrt{2}$ Pa) and (1 atm +$\sqrt{2}$ Pa), that is between 101323.6 and 101326.4 Pa. Although very small, this variation in air pressure at an audio frequency will be perceived as quite a deafening sound, and can cause hearing damage.

2.2.3 Auditory Masking

The mechanism of the basilar membrane produces an interesting psycho-acoustic phenomenon called auditory masking which appears when a sound becomes inaudible due to the presence of another sound. Two types of auditory masking exist : simultaneous (or frequency) masking and temporal masking.

Simultaneous masking occurs when two concurrent sounds emit at different frequencies but in the same critical band. One of them, namely the maskee, can be totally inaudible due to the presence of the other, the masker. If the vibration in regions of the basilar membrane produced by a masker sound is large, the vibration of the maskee sound in the same regions will not be perceived unless the excitation produced by the maskee exceeds that of the masker by a given minimum threshold. This masking threshold depends on the sound pressure level, the frequency and some characteristics of the masker.

Temporal masking occurs when two sounds are close in time. There is a loss of sensitivity around the sound's frequency that lasts a few milliseconds. Thus, the ear does not perceive the sounds preceding or immediately following a sound of strong intensity. The premasking is short, about 5 milliseconds, contrary to the postmasking which persists longer, depending of the duration on the sound.

2.2.4 Hearing and Seeing: Multisensory Integration

The familiar experience of both hearing another person speak in natural conversation, and seeing the speaker's lip movements while they speak, is an everyday example of multisensory integration or multimodal integration [Callan 2004]. This phenomenon involves both low-level perceptual features, such as detecting sounds and lip movements, as well as higher-level linguistic and semantic factors. Multimodal integration refers to the neural integration or combination of information from different sensory modalities. The neural association of the different stimuli gives rise to changes in behavior according to the perception of and reaction to those stimuli. It finally appears that the world is not perceived through distinct consideration of each sensory modality but rather through integrative and selective evaluation of the sensory impressions.

When sensory stimuli that normally evoke different subjective experiences are received, they are treated as incongruent by the sensory system [Callan 2004]. In such cases, crossmodal effects may occur in which interactions appear between sensory modality information. Previous work has revealed that sensory impression coupling is integrated in a much more complex way than a simple superposition of the distinct sensory modalities. This indicates that crossmodality has to be considered when designing virtual reality displays. Especially for our study case, the interaction between auditory and visual qualities is an important topic. To illustrate the interplay between auditory and visual features in a audio-visual display, Begault reports [Begault 1994]: *Dr. Laurel went on to comment how, while in the video game business, she found that really high-quality audio will actually make people tell you that games have better pictures, but really good pictures will not make audio sound better; in fact, they make audio sound worse.* And the author adds, as an explanation of the related scene: *So in the (virtual) world we're building, one of the features is a rich auditory environment.* Storms et al. [Storms 2000] mention that if the picture resolution in a video display system is held at a constant level, the visual quality can be rated higher if the audio bandwidth is increased. Similarly, the studies from Hollier and Voelecker [P. 1997] conclude that audio quality is improved in the presence of visual stimuli and that a decrease in video quality is followed by a corresponding decrease in audio quality. On the other hand, Beerends and De Caluwe [Beerends 1999] find an asymmetrical effect of auditory and visual influences; visual quality has a large significant effect on rated sound quality, whereas sound quality affects rated visual quality in a less extent. One general consensus is that perception of visual quality affects the impression of sound quality and vice versa. Research results are however divided in setting the extent of influence for both stimuli. This observation can be explained by the ambiguity of the term quality which implies subjective interpretation more than purely technically determined properties. Although quality can be usually defined for VR simulations as the property of being close to reality, the possible meanings of lack of quality are hard to define. Quality or effectiveness of a VR has been commonly measured by the amount of *presence* it evokes in users [Slater 2000].

For an extensive review of multimodal and cross-modal techniques for control of sound processing and synthesis, we refer the reader to the study of Camurri et al. [Camurri 2008]. The work surveys some relevant aspects of current research in control of interactive systems, putting into evidence research issues, achieved results, and problems that are still open for the future.

Chapter 3

Audio in Computer Games and Virtual Reality

Contents

The target of virtual environments is to simulate a situation where user behavior can be comparable with his/her response in reality. In this Chapter, we review the state of the art on sound simulation for VR and video games and the different strategies used for sound rendering. Then, the main principles and methods used in physically based simulation are presented, with specific emphasis on relevance for sound modeling. Finally, we discuss one of the essential goals of sound synthesis, that is, control, which will open the fundamental research directions taken in this thesis.

3.1 Sound Simulation for Virtual Reality and Games

As noticed by Raghuvanshi et al. [Raghuvanshi 2007], graphics, behavior and sound are the three main ingredients for a computer game to induce believability. Research in computer graphics and modern graphics hardware have made it feasible for many of today's games to render near-photorealistic images at interactive rates.

More recently, *NaturalMotion*[1], the creator of Dynamic Motion Synthesis (DMS), a break-through in 3D character animation, and *NVIDIA*[2], which acquired *PhysX* physics acceleration technology, have decided to gather their resources to provide an integrated way for developers to create games with fluid animation and simulated physics, which would further increase immersive gameplay. However, in contrast to graphics and behavior engine advances, sound generation and propagation in gameplay are not so extensively considered, mainly due to the computation expense that realistic sounds require for simulation.

3.1.1 The Traditional Approach

The traditional approach to creating soundtracks for interactive physically based animations is to directly play-back pre-recorded samples, for instance, synchronized with the contacts reported from a rigid-body simulation, see Figure 3.1. Looping and pitch shifting audio recordings are the methods of choice for more realistic continuous contact, where the velocity and the intensity of the normal force influence the parameters. Additional precomputed effects are implemented for spatial rendering when playing back the sound, namely the reflections of sound waves in the virtual scene that causes echoes.

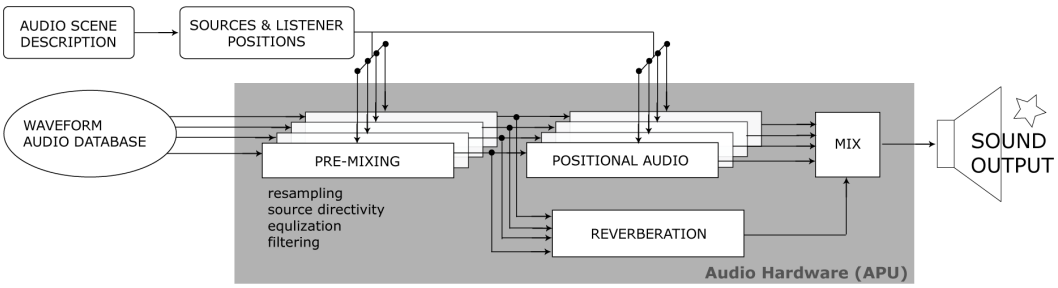

Figure 3.1: *A traditional hardware-accelerated audio rendering pipeline. Audio rendering operations are usually performed for every sound source. Pre-mixing can usually be implemented with few operations while positional audio and reverberation rendering require heavier processing (Diagram inspired from [Tsingos 2004]).*

The signal is represented by a digital sample. The values are stored at time intervals of $1/S_R$, where S_R is the sampling rate. The values can be given as floats or as 16 or 8 bit signed integers. To represent sounds with frequencies up to f Hz, the *Shannon Sampling Theorem* has to be fulfilled for an accurate representation of the signal, that is, $S_R \succ 2f$. Otherwise, the frequencies above f will *fold back* into low frequencies. Since the human ear can perceive frequencies up to 20 kHz, a

[1] http://www.naturalmotion.com/
[2] http://www.nvidia.com/page/home.html

common sampling rate for sound is 44.1 kHz. Depending on the available hardware, it is possible to dynamically manipulate a sample and change certain aspects of it in real time.

3.1.2 Limitations

Play-back of prerecorded samples obviously appears simple and fast, but it also has noticeable disadvantages. First, due to memory constraints, the number of samples is limited, leading to repetitive audio. Moreover, matching sampled sounds to interactive animation is difficult and often leads to discrepancies between the simulated visuals and their accompanying soundtrack. In addition, it requires each specific contact interaction to be associated with a corresponding prerecorded sound, resulting in a time-consuming authoring process. Finally, the technique can not capture all the physical effects of the objects and little flexibility is provided for authoring. Thus, other strategies have been explored.

3.2 From Playback of Sound Samples to Synthesis

By using recorded sound clips for object interactions, single sounds are associated with objects. This method however, fails, for example, to account for the variation in loudness and sound timbre which depend respectively on the magnitude and location of the impact forces. Realism of a scene depends on such subtle effects, avoiding repetitive effects of recorded sounds. In contrast to prerecorded samples, physically based sound synthesis is able to render the slight shifts in tone and timbre according to the impact location, material property, object geometry, and other factors.

3.2.1 Concept

Work by Adrien [Adrien 1991] describes how effective digital sound synthesis can be used to reconstruct the richness of natural sounds. In theory, to compute the sound perceived by an observer, that is, to generate a signal that drives speakers or headphones that lead to a correct perception of the sound, the following steps have to be performed:

- Modeling of the sound generating objects,
- Computation of the resulting sound field at the eardrums of the simulated observer,
- Generation of a signal at the headphones or speakers such that the eardrum of the human subject is exposed to the sound field computed.

For simplicity, it is usually assumed that the sources of the sound are sufficiently far from the observation point that we can approximate the acoustic wave by a plane wave. We can then replace the source with a suitable sound source. The simulation

is then divided in three parts: source modeling, sound propagation (reverberation), and sound reception. The work presented in this thesis focuses on the modeling of the point source.

3.2.2 Approaches

The problem of generating effective sounds in VEs has been addressed in early work by Hahn et al. [Hahn 1998], who identify three sub-problems: sound modeling, sound synchronisation, and sound rendering. Sound modeling has been studied extensively in the field of computer music [Cook 2002b, Iovino 1997] and computer graphics [O'Brien 2002b, Raghuvanshi 2006, van den Doel 2001]. Sound rendering consists in creating sound signals from models of objects and their movements within a given environment. This process is somehow similar to the process of generating images from their geometric models. Above all, the main concern in VE applications is the adequate parametrization of sound models that map the motion parameters, such as user's interaction, to the sound control parameters in order to effectively synchronized the visual and auditory displays [Avanzini 2008]. Automatic generation of a large and important subset of Foley sounds, namely the sounds made by the contact interactions between solid objects, has been addressed [van den Doel 2001, van den Doel 1999, van den Doel 1996]. Other work that extend techniques for computing sound effects includes [Cook 2002b, O'Brien 2001, Hahn 1998, Gaver 1993b, Takala 1992].

3.3 Physics Models for Interactive Sound Synthesis

Virtual environments increasingly rely on physics-driven animation and interaction. In order to synthesize the related auditory events, the physics engine must properly communicate, for example, the exact collision geometry, as well as the forces involved for every collision in the scene, to a sound system. Many of nowadays commercial games already perform this condition due to the use of appropriate physics engine such as *Havok*[3]. This section covers the main principles and methods used in physically based simulation with specific emphasis on relevance for sound modeling, that is, differential equations, particle systems, constraints, rigid bodies and contact detection. This section is mainly based on the course by Witkin and Baraff [Witkin 2001] on physically based modeling.

3.3.1 The Starting Point: Differential Equations

Differential equations emerge whenever a deterministic relationship involving some continuously changing quantities and their rates of change is identified or assumed. Many real world problems involve solving differential equations and one simple ex-

[3] www.havok.com/

ample is the determination of a ball falling through the air, considering only gravity and air resistance.

Numerical Solutions for Differential Equations

Computer animation is concerned with numerical solutions for solving differential equations, in which discrete time steps are taken starting with the initial value. The simplest numerical method is Euler's method. However, Euler's method is not accurate and can be unstable for time steps exceeding a certain threshold. Consequently, Euler's method is not very efficient. To improve the accuracy of numerical solution, the *midpoint method* can be used. A popular procedure for doing this is a method called Runge-Kutta of order 4 and has an error per step of $O(h^5)$. In videogames, the favored integration methods are simple symplectic schemes such as Verlet, leap-frog or symplectic Euler.

Efficiency and Accuracy

Whatever the underlying method, a major problem lies in determing a good stepsize. By using the method of adaptive stepsizing, the step can be varied accordingly, that is the step is large when it does not incur too much error and it is reduced when preventing excessive error.

Sometimes, an ODE can become *stiff*, preventing the use of Euler's method or the midpoint method, which are *explicit* methods. In this case, *implicit* ODE solution methods are used [Baraff 1998]. Implicit methods find a solution by solving an equation involving both the current state of the system and the latter one. Even if implicit methods require extra computation and can be much harder to implement, they take, however, much less computational time to use with larger time steps.

3.3.2 Particle Systems Dynamics

Reeves [Reeves 1983] introduces particle systems for explosion and further expanding fire simulation, presented in the movie *Star Trek II: The Wrath of Khan*. Other fuzzy objects such as clouds and water, namely objects that do not present a well-defined surface, can also be modeled by the same technique. Even if particles are usually basic graphical primitives such as points or spheres, they can also reproduce complex group dynamics such as emergent crowd behaviors [Heigeas 2003]. Each particle embeds its relative features, such as position, velocity or lifetime; these features determine the dynamical behavior of the particle over time, which can be altered due to procedural stochastic processes. Particles experience three different phases: generation, dynamics and death. Particles interact with each other. Modeling the interaction potential between pairs of atoms has long been addressed in molecular dynamics and is commonly used to characterize mass-spring systems. Adapting the dissociation energy and the potential energy width enables to model a large variety of physical properties, from stiff to fluid-like behavior.

With the work of Florens et al. [Florens 1991], and the more recent work of Raghuvanshi and Lin [Raghuvanshi 2006], particle systems, and in particular mass-spring systems, are adopted for sound rendering. The object's surface deformation is approximated with a mass-spring model, based on the geometry and a few material parameters. Masses and damped springs substitute respectively the vertices and edges of the input shell. The material properties of the object, namely Young's modulus, thickness of the object surface, and material density, are used to adjust the spring constants and masses. Assuming that the deformations are small, a coupled linear system of ordinary differential equations (ODEs) is deduced by linearizing about the rest positions. Providing the necessary conditions regarding sound percepton, the sounding modes of the input shell are determined. Therefore, during collision events, the sound system is acquainted with the object(s) involved, together with the magnitude and location of the impact that determine the gains for the different vibration modes. The final sound is computed as the sum of the pressure contribution of each particle, the pressure being determined by the velocity of the particle.

Particle systems can be integrated in game engines, digital content creation systems, and effects applications. Several implementations are ready for use, such as *The Particle Systems API* or the *Dynamic Particle System Framework* for the *XNA Framework*. Also, *Havok* and *Ageia*, bought in 2008 by *NVIDIA*, provide multiple particle system APIs that are used in many games. Finally, *Magic Particles* is another advanced particle solution (API + Editor), which is used by game developers in PC and console games.

3.3.3 Rigid Body Simulation

In physics, a rigid body is a simplification of a solid body of finite size in which deformations are neglected. The principal advantage of representing portions of a model with rigid bodies, rather than deformable finite elements, is computational efficiency. Indeed, although the update of nodes' motion and the assemble of concentrated and distributed loads require computational effort, the motion of the rigid body is determined completely by a maximum of six degrees of freedom at the rigid body reference frame.

Rigid bodies can be used to model very stiff components that are either fixed or undergoing large rigid body motions. They can also be applied to model constraints between deformable components, and they provide an appropriate approach for specifying certain contact interactions.

Unconstrained Rigid Body

In the case of unconstrained rigid bodies, the motion is simulated in response to the external forces acting on the body. The position of the whole body is represented by linear position, namely the position of one of the particles of the body, for instance, its center of mass or its centroid, together with angular position, or

orientation. When considering a rigid body in motion, both its position and orientation generally change in time. These variations are referred to as translation and rotation, respectively. Velocity, also called linear velocity, and angular velocity are determined with respect to a reference frame.

Non-penetration Constraints and Contacts

For constrained rigid bodies, the non-penetration constraints are enforced by computing appropriate contact forces between bodies in contact. Considering values for these contact forces, simulation carries on exactly as in the unconstrained case, that is, all the forces are simply applied to the bodies and the simulation evolves as if the motions of bodies were completely unconstrained. The computation of these contact forces is the most critical step of the entire simulation process.

Dealing with rigid bodies that are totally non-flexible, implies disallowing interpenetration. Two types of contact are dealt with. First, when two bodies are in contact at some point, and they have a velocity towards each other, such as a particle striking the floor, this case is called the *colliding contact*. Colliding contact requires an instantaneous change in velocity at any time a collision occurs and implies stopping the ODE solver.

Second, whenever bodies are resting on one another at some point, such as a particle in contact with the floor with zero velocity, the bodies are said to be in *resting contact*. In this case, a force is computed that prevents the particle from accelerating downwards; basically, this force is the weight of the particle due to gravity, or whatever other external forces exerted on the particle. Resting contact definitely does not involve to stop and restart the ODE solver at every instant. If we assume a physics engine with a routine that computes the force and torque acting on the rigid body, from the ODE solver's point of view, contact forces are just a part of the force returned by this routine.

The Finite Element Method

Finite elements and rigid bodies are the fundamental components for a model in animation. Rigid body idealization implies that only negligible deformations are undertaken by the object when responding to typical interactions. However, althoug very small, these deformations may be noticeable by hearing, provided that vibrations are between 20Hz and 20 KHz. In contrast to rigid bodies that move through space without changing shape and are described completely by no more than six degrees of freedom at a reference node, finite elements are deformable and have many degrees of freedom. Finite elements require expensive calculations to determine the deformations, preventing them from being used for real-time manipulation. For this reason, computation is performed in a preprocessing step.

3.4 Controlling the Sound Simulation

Physically based sound models can generally allow the creation of dynamic virtual environments in which the characteristics for sound rendering are included into general data structures for multimodal encoding of object properties, such as geometry, dimensions, or material. Thus, the physical properties of an object are gathered in a unified description that can be used to manage the rendering of visual, haptic and sound, in the same effort.

Sound design and creation of an auditory ambience remains a key feature for specific applications such as video games. The sound designer is usually responsible for these aspects and aims at providing a virtual environment with its specific identity. Sound simulation should allow coherence between sound and visuals, but also control of audio rendering.

3.4.1 A Sound in Coherence with Visual

To generate animated motions in real-time, interactive applications are more and more built upon physically based simulation techniques. Automatic handling of related auditory events is thus becoming a key factor in order to ensure audio-visual synchrony and consistency. This requires developing dedicated audio synthesis engines. However, the interface between physics engine and audio synthesis engine is not obvious and has to be defined.

Simulating sounds which are at the same time realistic and compelling for complex audio-visual scenes is challenging due to the trade-off between quality and efficiency. The physical simulation used to drive the sound synthesis would require a sampling rate of 44.1 KHz, that is about 1000 times higher than the one used for graphics (60Hz), since many surface properties produce force variations that have to be captured at a proper simulation rate. As a consequence, real-time sound synthesis cannot be managed by brute-force sound simulation. Real-time synthesis of sounds resulting from physical interaction of various objects in a 3D virtual environment, namely collisions, sliding, rolling, etc., are difficult to create in a pre-production process because they are highly dynamic and vary drastically depending on objects and interaction types. Since the main resources of the system are dominated by graphics and physically based simulation, synthesizing the sound efficiently using only a few part of the running time remains the critical challenge.

3.4.2 Parametrization and Expressiveness

Until recently the primary focus for sound generation in VEs has been in spatial localisation of sounds. Techniques have been developed to efficiently process the 3D audio data in order to allow real-time applications. Wand and Straßer [Wand 2004] propose a multi-resolution approach to 3D audio rendering where an importance-based sampling strategy is used to randomly select a sub-set of all the sound sources to be rendered at each processing frame. Also, in the work from Tsingos

at al. [Tsingos 2004], a framework for 3D audio rendering of complex virtual environments is proposed, in which sound sources are first sorted by the loudness level of the sound signals, which is efficiently updated in real-time using pre-computed descriptors stored with small chunks of the input audio signals.

Other scalable approaches based, for instance, on modal synthesis, have been proposed for real-time modal synthesis of multiple contact sounds in virtual environments [Lagrange 2001, van den Doel 2002, van den Doel 2004, Bonneel 2008]. Similarly, parametric audio representations enable for scalable audio processing, such as pitch shifting or time-stretching [McNally 1984, Wayman 1989, Moulines 1989], or frequency content alteration [Smith 1987, Serra 1997], without increasing significantly the computational effort, since processing would only concern a limited number of parameters. However, research is far less developed about parametrization techniques for sound source models and object motion/interaction.

In general, the fundamental objective of sound design might not be necessary physical realism but rather a physical *hyper-realism* that permits to combine sound design, consistency between the actions of the player, and the generated soundtrack. There has been much work in computer graphics addressing control or expressiveness. Expressive rendering is a new extension for computer graphics whose main objective is not to conceive images based on realistic physical phenomena, but to preferably convey knowledge using different styles (drawing, watercolor, etc.). In particular, *Non-Photorealistic Rendering* (NPR) involves any technique that produces images of a simulated 3D world in a style other than realism. For sound rendering, similar attempts to address control and expressiveness are not so numerous, even if Gaver [Gaver 1993b] already mentioned that the nonlinearity of the relationship between physical parameters and perceptual results should be neglect through the use of more simple models. As an example, the study from Rath [Rath 2003b] focuses on an efficient acoustic expression, namely, cartoonification, of a real-time rolling sound model, and proposes to combine physics-based models with more perception-oriented structures.

So far, physical models of sound have been acknowledged for their appropriateness in terms of control, efficiency, simplicity of implementation and sound quality. For interactive simulation, the method of choice has been to directly use the vibrational parameters. However, methods for control of physical parameters need to be extended [Erkut 2008]. An optimal model should demonstrate, at the same time, suitability to illustrate the underlined phenomenon and enough flexibility to perform the processing in a natural way. Thus, performance of a model can be evaluated in terms of its accuracy to model the physical process, or in terms of its parametric control properties. It is also generally accepted that physical models hardly agree with real observed data due to the numerous parameters that are involved and to the fact that control parameters are not easily correlated to the attributes of the final sound. Adequate parametrization is the main problem and several studies have investigated this problem to identify structural and control parameters. Model fitting to real data should be further explored for physical sound modeling not only in

terms of parametric control but also in terms of system structure design. Above all, the fundamental question is how to make controllability and interactivity dominant directions for design of physics-based sound synthesis.

In the first part, we have introduced the main principles for physics of sound and sound perception, and the fundamentals of audio in the context of virtual reality and computer games. We have seen that virtual reality aims at simulating an environment where users feel immersed and behave almost as in a real situation. Physics-driven animation and interaction are increasingly major factors for virtual environments, especially for games. The traditional approach to creating soundtracks for interactive animations is to use recorded sound clips triggered by events in the virtual scene. Since this approach has major limitations, audio synthesis techniques have been developed. We will now investigate physically based synthesis for sound interactions of various objects in a virtual scene. We propose to address control of sound modeling through the properties of the force that arises between the interacting objects, and the attributes of the resonating objects.

Part II

Physics-Based Sound Synthesis

Chapter 4

Contact Modeling

Contents

According to Gaver [Gaver 1993a], physical analysis of an acoustic event is beneficial, especially considering that it is often challenging to determine the acoustic information from acoustic analysis alone. By investigating the physics of sound producing events, significant source characteristics can be extracted. In addition, resynthesis of the audio event can be then directed by the resulting physical simulations.

This chapter starts with general concepts about physics-based models. Then, previous work about contact modeling is reviewed with an emphasis on interactive applications. Finally, first investigations in complex contact interactions are presented, which opens the way to our work presented in Chapter 5.

4.1 Generalities About Physics-Based Models

The purpose of physics-based models for sound synthesis is to develop efficient algorithms based on the fundamentals of sound production mechanisms. In this section, we present the main principles of physics-based sound models, and we introduce the decomposition upon which our modeling system is built, that is the *exciter - resonator* system.

The Importance of the Model

Erkut et al. [Erkut 2008] emphasize how a model can be beneficial in dealing with the elaborate mechanisms of sound production. Indeed, a model enables us to focus

on the essential features of the phenomenon, leaving aside the irrelevant details. The abstraction upon which the model is built, is closely related to the context; for instance, modeling for immersive virtual environment implies specific choices. Models just approximate real physical phenomena. For instance, a sound synthesis model may not need to be very precise, since the listeners cannot hear the difference to the exact solution. A mathematical model is deduced from the abstraction of the physical mechanism, usually through rules that relate measurable quantities, such as force and velocity in the mechanical domain. Finally, by using a physics-based sound model, a huge amount of audio data is described by a small number of significant parameters.

The Sound Synthesis System: Exciter - Resonator

Our study focuses on the sound source modeling. Source modeling consists in recreating sound production properties. The model denotes the specific features of the mechanisms related to sound rendering. One method for source modeling is to underline the functional elements *exciter* and *resonator*. We consider that the signal goes from the exciter to the resonator and is unidirectional. Despite its simplification, source modeling allows modularity to a certain extent. As an example, van den Doel et al. [van den Doel 2001] adopt a similar approach where their audio toolkit consists of three layers of software objects, a filter-graph layer which provides objects to build unit-generator graphs, a generator layer with basic audio processing blocks such as wave-tables and filters, and a model layer which contains implementations of the physical models for excitation types.

In source modeling, the exciter provides the energy for the vibration of the sounding object, whereas the sounding object or resonator receives the energy from the exciter. As an example, when a glass impacts the floor, its potential energy is partly transformed into acoustic energy, whereas some portion is turned into kinetic energy as the brittle objects bounce, and the rest is dissipated. We assume that most of the time, sound arises from contacts between objects. Thus, by modeling the contacts in a physics based sound synthesis approach, we model the excitation that gives rise to the sound.

4.2 Previous Work for Contact Modeling

Contacts between bodies have been extensively investigated for sound rendering [Avanzini 2002a, Avanzini 2002b, Pai 2001]. In their study of bouncing and breaking events, Waren and Verbrugge [Warren 1984] underline the existence of two classes of invariants that are relevant in the perception and estimation of contact sounds. The *structural invariants* are related to the attributes of involved objects such as size, shape, mass, elasticity, surface properties or material, whereas the *transformational invariants* specify object interactions and changes, namely velocities, forces and position of interaction points. With contact modeling, the purpose is

to identify the transformational invariants that are characteristic to a specific contact type. According to [Pai 2001], realistic contact sounds imply adequate physically based models not only for the resonators, but also for the contact interactions.

In this section, we first shortly introduce the two schemes, according to which the contact force modeling can be performed, namely a feed-forward scheme and a direct computation of non-linear contact forces. Contacts are divided into two parts: impacts, and continuous contacts, i.e., scraping, sliding and rolling. We then present the previous work related to modeling of impacts and continuous contacts.

4.2.1 Two Modeling Schemes

In a feed-forward scheme, the physical mechanism at the origin of the contact sounds is not directly considered due to its intricate character, and the audio algorithms are not determined using the fundamentals of known physical laws. The contact forces that induce the vibrations of the interacting objects, are externally computed or recorded instead. This technique is the most commonly used and is illustrated for example in the work from van den Doel et al. [van den Doel 2001]. In contrast with this method, the excitation models can enclose direct computation of non-linear contact forces. Although this implies more complexity in the synthesis algorithms, the method has major benefits. The quality is improved due to the accuracy of the audio-rate computation. This is especially true for impacts where contact time is very short. In addition, better interactivity and receptiveness are achieved, and this is particularly noticeable for continuous contacts, such as in the study from Avanzini et al. [Avanzini 2005] on stick-slip friction.

4.2.2 Impact Modeling

During a collision between two solid bodies, an impact force arises at the contact point and is characterized by a large amplitude for a short period of time. The specific features of the contact force depend on the surfaces in contact, namely their shape and the material properties. Foley sounds, that is, the sounds that arise when solid objects interact, are introduced in the work from van den Doel et al. [van den Doel 2001]. Impacts, or impulsive contacts, are shown to be mainly influenced by the energy transfer during object interaction and the hardness of the contact. The duration of the force is closely related to the hardness of contact, whereas the magnitude of the force profile is affected by the energy transfer. The Dirac, commonly used to model impact events, appears inappropriate for modeling of *very hard* collisions such as a marble on a stone floor. Based on experimental data that show sequences of very fast contact separations and collisions, van den Doel et al. [van den Doel 2001] estimate that the micro collisions are caused by the modal vibrations and propose to model hard impacts with a brief distribution of impulse trains at the dominant modal frequencies. The nonlinearity of contact forces has been especially addressed in the work from Avanzini et al. [Avanzini 2001]. Contact time appears as a substantial cue in the perception of collision, since it affects the

spectral characteristics of the initial transient. In particular, a short contact time creates an impulse-like transient with a rich spectrum whereas a long contact time produces a damped transient where the high frequency content is narrow.

For a more perceptual point of view, Aramaki et al. [Aramaki 2006a, Aramaki 2006b] introduce an efficient hybrid synthesis technique for percussive sounds, based on a combination of physical and perceptual considerations. Their real-time implementation proposes a wide variety of impact sounds. The excitation type and especially its mechanical characteristics, such as the impact velocity and the attack time, can be adjusted. Tuning is allowed by controllers on different parameters, such as pitch and damping coefficients. The control mapping is based on natural sounds and allows different types of tuning regarding the tonal and noisy parts of the input signal for example. The audio algorithm includes the auditory sensitivity by the use of critical bands of hearing, the Bark bands, and by extracting the pitch of the sound from emergent spectral components.

4.2.3 Continuous Contacts Modeling

Continuous contacts are generally divided into two types according to the amount of frictional force involved: scraping, sliding with a non-zero frictional force vs rolling with no or a very small frictional force.

Modeling continuous contacts for interactive applications is particularly challenging. Accuracy in sound modeling implies tracking the surface variation at a sufficient high detail level to ensure appropriate audio rendering. In practical terms, the sampling rate would need to be increased about 1000 times that of graphics since many surface properties induce force variations. Given the complexity of the task, most approaches that have been used for real-time rendering of continuous contacts are based on perceptual features that largely simplify the acoustic phenomenon and can propose continuous contact forces which are externally computed [Hahn 1998, van den Doel 2001, van den Doel 2003, Rath 2003b]. Simplifications may be based, for example, on the role of surfaces irregularities in the production of noise in sliding and rolling contacts. Based on noise spectra recordings of those contacts, Ananthapadmanaban and Radhakrishnan[Ananthapadmanaban 1982] show that discrete frequencies produced by surface irregularities are overshadowed by the frequencies of the excited system, whereas they are significant in the case of specific periodicity. In the following, we detail these methods based on a feed-forward scheme. Although less common, we also specify the technique used by Avanzini et al. [Avanzini 2005] to directly compute the contact force for continuous contact interactions.

4.2.3.1 Frictional Interactions: Scraping, Sliding

The early work of Hahn et al. [Hahn 1998] introduces a number of synthesis algorithms for contacts, and in particular, for scraping. Sounds generated from the surface textures of objects are modeled through a characteristic signal that corresponds to the microscopic grain texture. The signal is filtered by the geometry and

material characteristics of the object that is used to scrape the surface. Based on the metaphor of the phonograph needle, smaller, harder objects extract smaller microscopic surface features whereas larger, softer objects behave as low-pass filters. A timbre tree is implemented to model this process and is evaluated at the sampling rate of the sound signal, that is, 22 kHz in order to avoid extreme aliasing. This method involves a large computational effort. Another solution is to directly synthesize the contact force from a stochastic noise model [van den Doel 2001], which is also quite involved. Scraping and sliding are modeled as a combination of an effective surface roughness and an interaction model. If sliding involves multiple micro-collisions at the contact area, scraping is characterized by noise with an overall spectral shape on which one or more peak are superimposed. The frequency of the reson filter is scaled with the contact velocity to render the sensation of scraping at different speeds.

For more simple formulations, van den Doel and Pai [van den Doel 2003] propose the use of surface profiles by simply scraping a real object with a contact microphone. However, this technique implies that the considered objects/surfaces are available which is not necessary the case for all objects of a virtual scene. In addition, extracted profiles are dependent on experiment conditions and the main features of the surface may not be modeled. Huang et al. [Huang 2003] propose a system that models the audio and haptic interactions with a fabric, combining a stylus and an audio-haptic interface. The stylus rubbing interaction model relates natural frequencies of the fabric sound and the fabric surface roughness. They assume that the acoustic energy is proportional to the frictional power loss during the rubbing motion, and state that sound is proportional to the square root of the frictional power loss from the rubbing force. The sound appears louder as the pressure force increases and it is sharper as the speed of interaction increases. In a similar way, the work from Essl et al. [Essl 2005] introduce the Scrubber for controlling the friction-induced sound where users experience a realistic relationship between gesture and sound. The Scrubber allows a variety of sound synthesis algorithms to be handled, based on granular synthesis, wavetable synthesis and physically informed modeling.

Unlike feed-forward scheme approaches, Avanzini et al. [Avanzini 2005] present a sound synthesis algorithm which embeds direct computation of non-linear contact forces for frictional interactions. The model is based on an elasto-plastic friction formulation and aims at providing a general description of the non-linear friction between two resonating objects. The friction is modeled as a large number of bristles, namely stiff hair, each contributing to a fraction of the total friction load. The friction force is parametrized thanks to a pseudo-random value so that a broader set of frictional interactions including scraping and sliding can be simulated. By direct manipulation and listening, users perceive that the normal force is related to the roughness of the interaction, whereas the bristle stiffness and damping respectively affect the evolution of mode locking and the sound bandwidth. However, since control parameters are numerous, controlling the sound module can be problematic.

4.2.3.2 Rolling

In contrast to scraping or sliding interactions, the surfaces involved in rolling have no relative speed at the contact point leading to a difference in sound rendering. In addition, it has been shown that the radius of the interacting object has an effect, and only the low frequency content of the effective profile impacts the sound. Van den Doel et al. [van den Doel 2001] propose a model similar to their scraping model, where an additional low-pass filter with adjustable cutoff frequency defines the rolling quality. Based on the analysis of recorded rolling sounds which suggests a stronger coupling with the modes than for the sliding force, a gamma-tone model driven by noise is implemented. The spectral envelope is further improved near the object's resonance modes. In their illustrated example, where a little rock is thrown in a wok, the normal force, the sliding speed (or relative surface velocity), the rolling speeds (speed of contact point with respect to the surface) on both objects, and the impact force are dynamically tracked to drive the appropriate audio.

Instead of aiming at ideal realism, Rath [Rath 2003b] explores an efficient acoustic expression, namely *cartoonification*, and he proposes the combination of physics-based models with perception-oriented structures. The basis of the algorithm is a physical model of an impact interaction force without additional perpendicular friction forces. A dynamic offset signal is created from the interacting surface profiles, that is further input to the impact model. Rath underlines that rolling sounds are mostly characterized by periodic patterns of timbre and volume, which is of high perceptual significance. In the case of object asymmetries, the gravity acting on the rolling object is modulated, a phenomenon that intensifies as the velocity increases. An adequate model should take into account this effect with appropriate parameter modulations.

For a more perceptual approach, the studies from Houben et al. [Houben 2004, Houben 2005] investigate the auditory perception of the size and speed of rolling balls. Auditory perception experiments show that listeners generally concentrate on spectral information for assessing the size or speed of rolling balls, and only to a slight extent on temporal information for assessing the speed of rolling balls. Results also show an interaction effect when both size and speed of the rolling ball are varied.

4.3 Initial Investigations for Contact Modeling

As mentioned before, capturing the variations of the surface for sound modeling is not a feasible task due to constraints on the sampling rate. On the other hand, contact modeling has to render the audio effects caused by specific visible features. Almost regular patterns as observed on many ground surfaces, produce macro-temporal periodicities that prove to be of high perceptual significance [Ananthapadmanaban 1982, Rath 2005]. Starting from these observations, this section presents our investigations on contact modeling, from which the origi-

nal work presented in Chapter 5 originates.

4.3.1 Texture Modeling

Texture for graphics and haptics can inspire sound rendering. In computer graphics, Perlin introduced a type of coherent noise, based on a fractal observation of natural phenomena,that is the sum of several coherent-noise functions of increasing frequencies and decreasing amplitudes [Perlin 1985]. Given this property, Perlin noise is optimal for producing natural, self-similar textures such as granite, wood, marble, and clouds.

The survey by Strobl et al. [Strobl 2006] points out the existence of methods that try to transfer existing techniques from computer graphics for modeling sound textures. As an example, Filatriau et al. [Filatriau 2006] propose different strategies to link visual and sonic textures using similar synthesis processes. For music applications, visual textures are exploited for sonic texture production. Methods to induce both visual and sound mechanisms by a common gestural control are proposed.

4.3.2 An Audio Force Driven by Perlin Approach

Method

The scenario we want to address is an object interacting continuously with a plane surface by sliding or scraping. Previous work from van den Doel et al. [van den Doel 2001] has shown that directly synthesizing the contact force from a stochastic noise model is quite involved. Thus, we investigate a noise-based approach for modeling contact interactions based on Perlin noise for computer graphics. Indeed, Perlin noise has the main advantage of low-memory usage, and for this reason, it is frequently used to generate textures when memory is extremely limited, and is increasingly finding use in Graphics Processing Units (GPU) for real-time graphics in computer games. A Perlin noise function is merely made of several interpolated noise functions added together. It consists in first creating a noise function, that is a random number generator. Then, an interpolation function, usually a cosine, is used to smooth out the values it returns. Finally, aside from interpolation, the output of the noise function can also be smoothed but it really becomes useful in two or three dimensions, where the effect is to reduce the squareness of the noise.

During sliding or scraping, a force results from the surfaces in contact. Our goal is to easily generate this force according to specific parameters concerning the object and the surface in interaction. For this purpose, we propose an interface to construct a synthetic force, or *audio force* as defined by van den Doel et al. [van den Doel 2001], see Figure 4.1. Our model uses a simple 1D Perlin function, since the audio force is one dimensional. The main part of the Perlin function is the loop, as shown in the code below. Each successive noise function that is added is known as an octave, and persistence determines the amplitude of the octave. If i is the ith octave being added, frequency and amplitude of the noise function

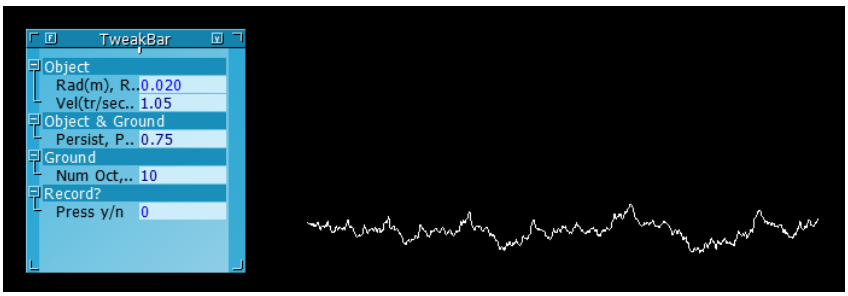

Figure 4.1: *An audio force generator based on Perlin noise function for computer graphics.*

are determined according to the definition for Perlin noise, that is, $frequency = 2^i$ and $amplitude = persistence^i$. In particular, persistence inferior to 1 leads to small amplitudes for high frequencies, typically *smooth rolling hills*, whereas persistence superior to 1 implies lower amplitudes for low frequencies, typically a *flat but very rocky plane*.

```
function PerlinNoise_1D(float x)
      total = 0
      p = persistence
      n = Number_Of_Octaves - 1

      loop i from 0 to n
          frequency = 2i
          amplitude = pi
          total = total + InterpolatedNoisei(x * frequency) * amplitude
      end of i loop

      return total
end function
```

Our implementation relates the number of octaves and the persistence to parameters of the surfaces in contact. The formulation of the 1D Perlin function is further developed to include influence of object's parameters such as its mean radius and velocity. Thus, it conveys the idea that the audio force is filtered by the geometry attributes of the object interacting with the surface, namely smaller objects are able to pick up smaller microscopic surface features and larger objects act as low-pass filters.

Results

The resulting audio force is then convolved with the impulse response of the objects in interaction, see Figure 4.2 and Figure 4.3.

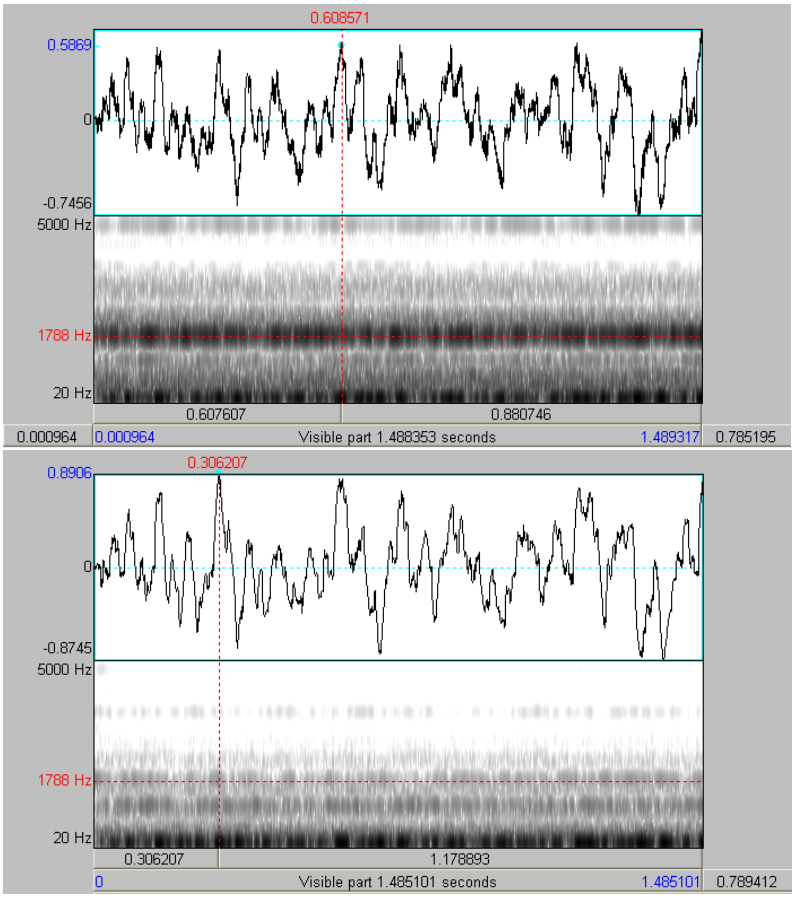

Figure 4.2: *A synthetic audio force is generated, based on the Perlin noise function: persistence = 1.5, number of octaves = 10, mean radius of the interacting object = 0.005 (top) and 1 (bottom). Resulting sounds (time signal and corresponding spectrogram) are obtained by convolving the audio force with the impulse response of the objects in interaction, here a ceramic ball.*

In Figure 4.2, we simulate an interaction where the changing parameter is the mean radius of the object, that is from 5 mm (top) to 1 cm (bottom), persistence and number of octaves being set to 1.5 and 10 respectively. Since smaller objects are more likely to catch microscopic surface features than larger objects, a smaller object is more frequently excited by a rough surface and the spectrogram of the resulting sound consequently has a larger high-frequency content, see top of Figure 4.2.

In Figure 4.3, persistence and number of octaves are set to 0.7 and 10 respectively, whereas the changing parameter is the mean radius of the object, that is from 5 mm (top) to 1 cm (bottom). Since the roughness of the surfaces is less pronounced compared to the case in Figure 4.2 (persistence = 0.7 vs 1.5), the sounding object is not so broadly excited, which is noticeable by the spectrogram of the resulting

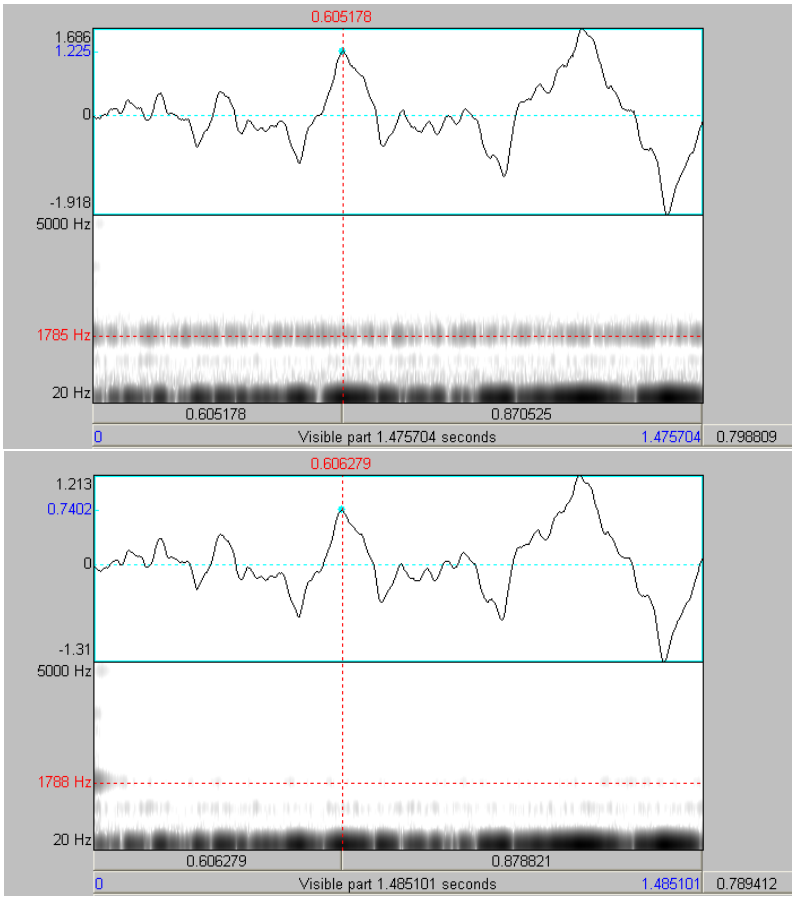

Figure 4.3: *A synthetic audio force is generated, based on the Perlin noise function: persistence = 0.7, number of octaves = 10, mean radius of the interacting object = 0.005 (top) and 1 (bottom). Resulting sounds (time signal and corresponding spectrogram) are obtained by convolving the audio force with the impulse response of the objects in interaction, here a ceramic ball.*

sounds.

Discussion

The approach presented here has the main advantages of having low-memory usage and simple to implement. However, since it is based on the Perlin noise function, it does not account for strong repetitive patterns that may improve sound perception of continuous contact. Thus, in the following Chapter we present a method to address audible position-dependent variations during continuous contacts by using visual textures of objects in the environment.

Chapter 5

Audio Texture Synthesis for Complex Contact Interactions

Contents

We present a new synthesis approach for generating contact sounds for interactive simulations. To address complex contact sounds, surface texturing is introduced. This work was presented at *the Fifth Workshop On Virtual Reality Interaction and Physical Simulation, VRIPHYS 08*, in Grenoble, France [Picard 2008]. This Chapter first introduces the motivations for this approach, and an overview of the technique is given. Synthesis of excitation patterns is presented. We then describe our flexible audio pipeline which controls the real-time time audio rendering. Finally, we discuss the relevance of the method and the limitations.

5.1 Introduction

Compelling audio rendering is becoming a key challenge for interactive simulation and gaming applications. Matching sound samples to interactive animation is difficult and often leads to discrepancies between the simulated visuals and their soundtrack. Furthermore, sounds of complex contact interactions should be consistent with visuals which is even more difficult when a small number of pre-recorded samples is used. Increasing the number of samples is not always possible due to the cost of recording samples. Alternatively, contact sounds can be automatically generated using sound synthesis approaches. Convincing continuous contact such as rolling or sliding requires an appropriate contact interaction model. To date, the proposed

models appear quite involved and can make authoring and control challenging for a sound designer.

Our approach proposes a technique for rendering continuous contact sounds that automatically derive excitation profiles from the analysis of a surface image. The image textured onto the surface of interacting object is considered to model the force causing the sound. Contact features are modeled as a discontinuity map computed in a pre-process, making data available on the fly for real-time audio rendering. An implementation of a modal model allows for separating the material properties from the force characteristics. A sound material database is then processed with the excitation profiles resulting in the subtle audio sensation of interaction with textured or rough surfaces. Our flexible audio pipeline further proposes different levels of detail which can be chosen according to the desired granularity of the sound interactions. Our contributions are:

- a contact interaction model suitable for audio rendering,
- a solution for generating on-line audio of subtle sounding events,
- a control mechanism for the resolution of sound interactions.

5.2 Overview

Our approach borrows from physically based sound synthesis and textured-based modeling. Contact sounds result from the combination of the material property of the objects in contact and the characteristics of the interaction force .

The proposed solution consists in deriving the force transmitted between interacting objects from the textured surfaces used for visual rendering. Thus, the method allows to render the sound emitted from real and virtual objects. The force interaction causing the sound is extracted as a height map. Modal parameters, i.e frequencies, gains and decays, are then processed with the excitation profiles. In addition, the excitation profiles can be of different levels of detail to achieve the desired granularity of the rendered sound. Figure 5.1 illustrates the modal model used for sound rendering of impacts. Figure 5.2 illustrates our method for sound rendering of continuous contacts with the use of synthetic excitation profiles.

5.3 Synthesis of Excitation Patterns

Our goal is to model the excitation force causing the object to vibrate and to output a sound comparable to a real-life situation. It has been shown that the precise details of the contact force will depend on the shape of the contact areas [Pai 2001]. The visual texture image is used both for visual rendering and to determine the potential excitation profile of the interacting object, in the case of rigid body interaction. This approach can be compared to Shape from Shading approach [Zhang 1999]

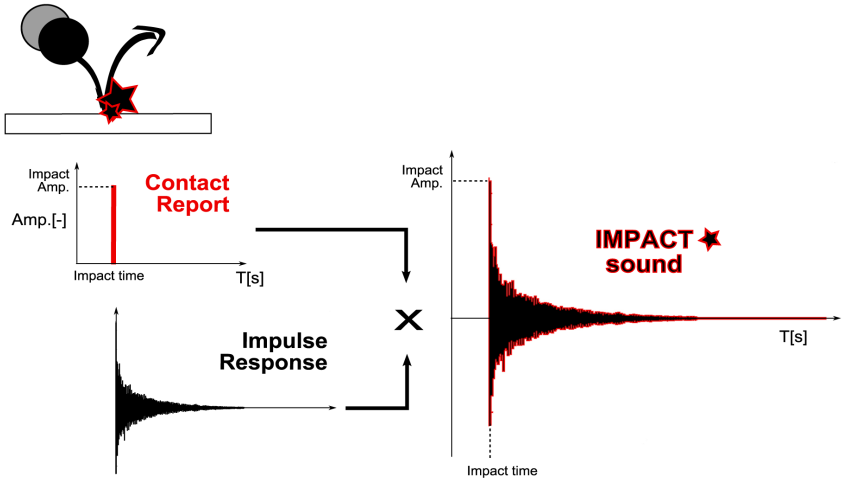

Figure 5.1: *Sound rendering method for impacts.*

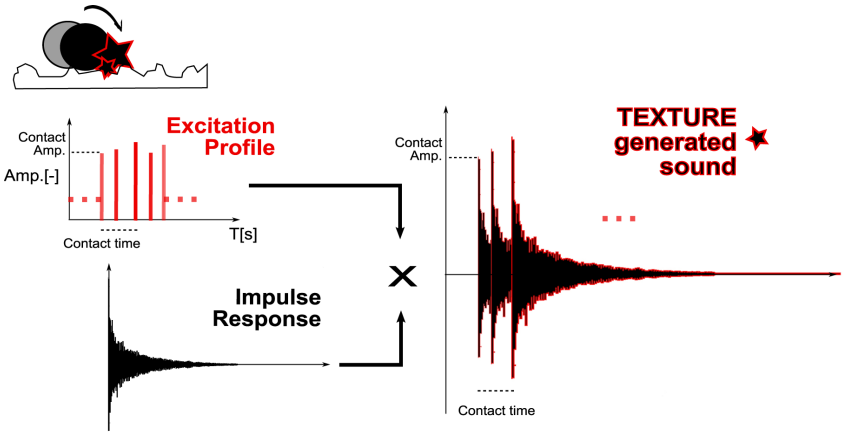

Figure 5.2: *Sound rendering method for continuous contacts (for simplicity, we only show the resulting sound of the localized excitation profile.). Our contribution lies in the creation of an excitation profile.*

which computes the three-dimensional shape of a surface from one image of that surface. As ambiguities exist when interpreting the surface, the main features of excitation profiles are considered to be independent from the light source, the surface reflectance or the camera position in the image. As a result, synthetic *audio excitation* profiles created directly from an image surface represent the potential audio effect resulting from the interaction between this surface and another smooth one. As an example, if an image of a tiled floor is simply mapped on a groundplane, no contact information about a smooth ball colliding with the gaps is given. Thanks our approach, the synthetic *audio excitation* profiles created from the tiled floor picture provides the missing pieces of information and the sound resulting from the interaction can be rendered.

The simulation of the excitation profiles needs to advance at the audio sampling rate or greater, i.e. 44kHz which is much higher than the physics simulation rate and the graphics frame-rate. The simulation of audio excitation profiles does not need to be of high auditory quality since profiles are used to excite resonance models and will not be heard directly. Thus, profiles are generated by re-sampling a *discontinuity map* extracted in a pre-processing task, along the trajectory of the contact interaction.

Finally, the synthesis of excitation textures can create the audio sensation of interacting with regularly featured or textured surfaces and also rough surfaces. Regularly featured surfaces have been shown as significant for the sensation of rolling objects [van den Doel 2001], i.e. the resulting sound should be noticeably repetitive. As a consequence, our method of extracting the prominent features from visual textures appears relevant.

5.3.1 Extracting the Discontinuity Map

Extracting the discontinuities of a textured surface with an edge detection filter allows us to render the resulting sound from the interaction with this surface. Different methods of edge detection in images were tested for the modeling of the excitation profiles. The requirements for edge detection in vision are not the same as for audio. It is likely that audibly speaking, main/strong features of a surface would be prominent and should be enhanced in comparison with other "noisy" parts of the map. We used a method for discontinuity extraction which depends on the image texture appearance. We distinguished between "simple" and "complex" image textures depending on the feature content. This is analyzed in a pre-processing task using the histogram of the image.

"Simple" image textures are characterized by a narrow-band histogram whereas "complex" image textures demonstrate a broad-band histogram, as seen in Figure 5.3. We used the *CImg*library[1].

"Simple" image textures have prominent features without pronounced noise. Because of its efficiency in terms of computation, the Sobel filter [Chien 1974] is used

[1]CImg Library - C++ toolkit for image processing: http://cimg.sourceforge.net/

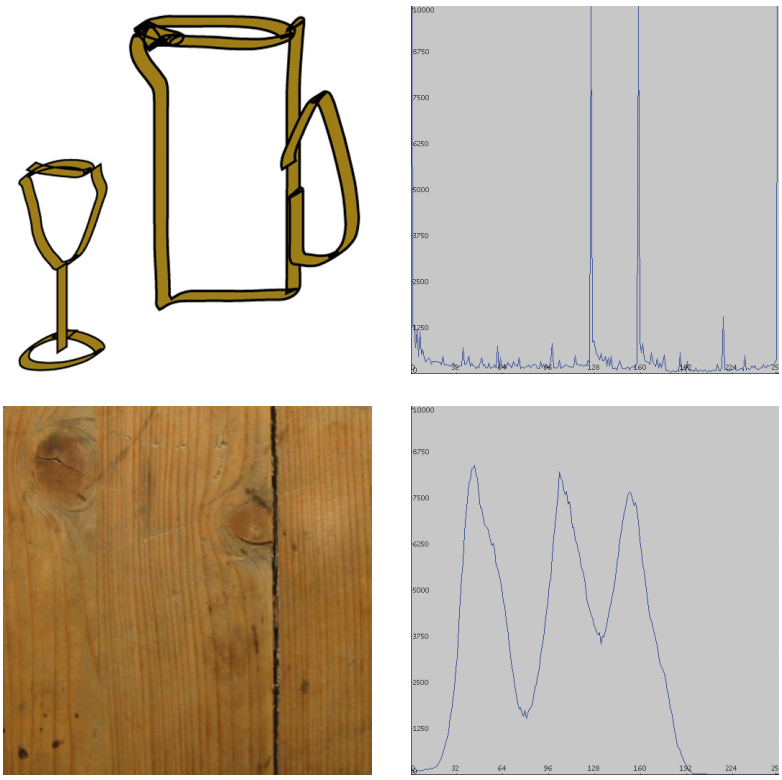

Figure 5.3: *"Simple"(top) and "complex" (bottom) image textures and their respective histogram.*

to detect the discontinuities which may produce sound. The Sobel operator is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function.

This filter approximates high frequency variations but is adequate for enhancing the main features of a "simple" image texture. The *audio excitation* profile is stored as an image, an example is given in Figure 5.4. Knowing the positions of the objects in interaction, an excitation force is created based on the interpolated value of the closest pixel values. In order to guarantee audio quality, the discontinuity map has to be of high resolution. However, since the information is binary, the size is moderate.

"Complex" image textures are analyzed in terms of isophotes, i.e. lines drawn through areas of constant brightness. This is computed using a marching squares algorithm. The isocurves are saved as a set of points. Figure 5.5 shows that this technique preserves the subtle discontinuities of the tiles which are not completely smooth. In comparison, the Sobel filtered image enhances the main features only and the fine parts of the *audio excitation* profiles are not adequately rendered. An excitation force is created according to the proximity of the contact interaction to

Figure 5.4: *"Simple" image texture: original image and discontinuity map by Sobel filtering.*

an isocurve. The difference between the current elevation gradient value and the previous elevation gradient value of the isophotes are used to modulate the amplitude of the excitation.

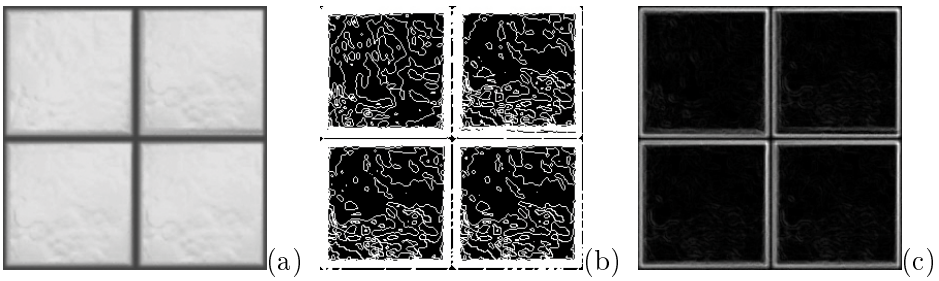

Figure 5.5: *"Complex" image texture: original image texture (a), corresponding isocurves image texture (b), to be compared with the Sobel filtered image (c).*

5.3.2 Coding the Discontinuity Map

When the rigid-body simulation reports a continuous contact, the discontinuity map is inspected to access the excitation profile for adequate sound rendering. In the case of discontinuity maps with high frequency content, i.e. with a highly noticeable noisy part, the high resolution of the isocurves might not be needed and an approximation would be sufficient. For this reason we propose encoding the texture as two maps corresponding to the main features and a noise map, as seen in Figure 5.6.

In order to extract the main features, the original image texture is filtered with a "Difference of Gaussians" (DOG) filter [Young 1987]. The DOG filter computes two different Gaussian blurs on the image, with a specific blurring radius for each, and subtracts them to yield the result. This filter offers more control parameters than the Sobel filter. The most important parameters are the blurring radii for the two Gaussian blurs. Increasing the smaller radius tends to give thicker edges, and

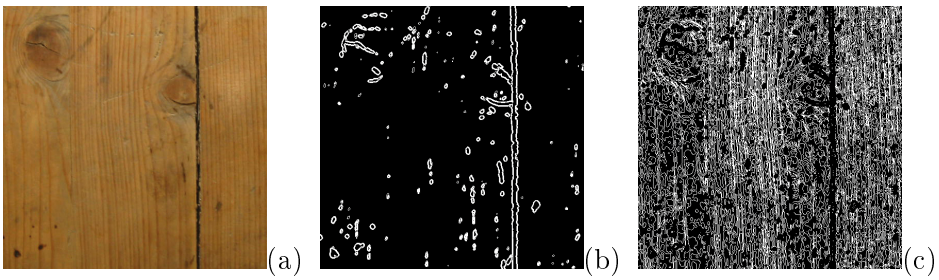

Figure 5.6: *Coding a texture (a) as the main features (b) and the noise part (c).*

decreasing the larger radius tends to increase the threshold for recognizing something as an edge. The blurring radii were set to 1.0 and $\sqrt[2]{1.6}$. The sensitivity threshold and the sharpness of edge representations were respectively set to 0.998 and 4.0. Our method combines this with a pre-process of bilateral filtering [C.Tomasi 1998] in order to smooth out the noise while maintaining edges.

The noise map is coded statistically. Figure 5.7 shows that random excitation profiles have similar behaviors and consequently, the noisy part of the texture can be considered uniformly distributed. The noise map is approximated by one of the excitation profiles, where excitation velocity and scale of the texture are known. During the real-time animation, the noise excitation profile is resampled according to the interaction velocity and the scale of the surface texture. Then, it is combined with the component corresponding to the main features during sound rendering.

Type	Original Image	Isophote Vectors	Feat.+Noise Map Coding
Size	786Ko	1.09Mo	544Ko

Table 5.1: *Statistics for size of the discontinuity map for an original image of 512x512 pixels (seen in Figure 5.6).*

Table 5.1 shows that the coding for discontinuity map is efficient. Moreover, vectorization allows trivial scaling of the excitation profiles making them independent of the size of the original image texture. Thus, our method provides two levels of detail which can be used according to the desired precision of the rendered sound. The resolution of the excitation profile can be modulated according to the viewpoint in the scene: as an example, when the interaction is far from the listener, a low resolution is sufficient to provide realistic sound interaction. Moreover, complex scenes with multiple objects/surfaces interactions can be synthesized in a more computationally efficient manner using this technique.

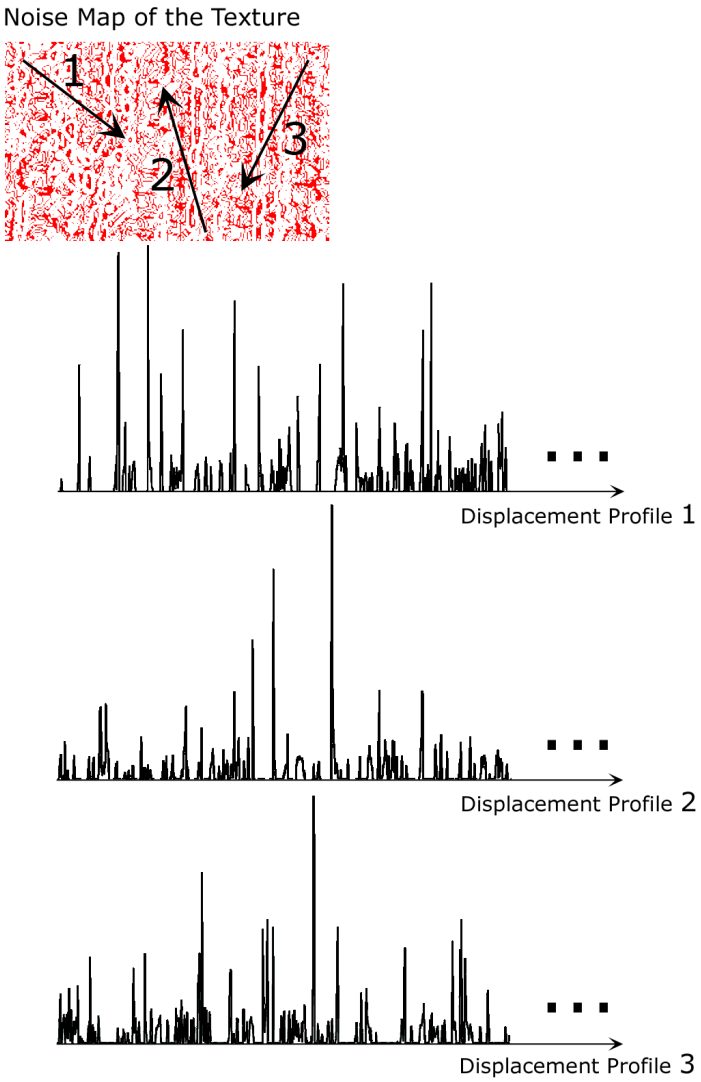

Figure 5.7: *Three trajectories of a smooth ball interacting with the noise map (top) and the corresponding excitation profiles extracted (bottom).*

5.4 Real-Time Audio-Visual Animations

Our approach proposes a flexible audio pipeline specifically adapted for real-time audio-visual animations focused on quality and variety. Sound rendering can be seen as a flexible *audio shading* (see Figure 5.8) allowing procedural choice of the parameters of the sound material, i.e modal parameters, and the excitation profiles for synthesis, driven by the contact report of the rigid-body simulation.

In [Takala 1992], a modular sonic extension of the image rendering pipeline is introduced where analogies between sound and texture are drawn. Similar to our

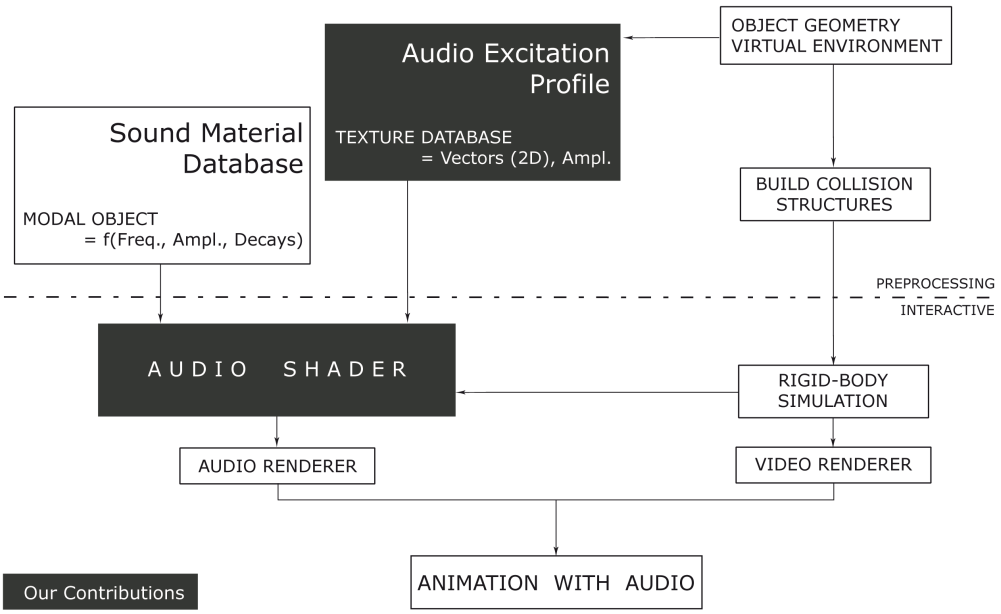

Figure 5.8: *Overview of the audio shader integration. Our contributions are underlined in dark gray boxes.*

approach, the sound transformations correspond to shaders[2] in a shade tree during its traversal from a source to a camera. However, the modular architecture aims at providing a general methodology for sound rendering for animations and does not detail methods for audio resulting from complex interactions between objects/surfaces. Our audio pipeline gathers data from sound materials and excitation profiles. In our modal model implementation, modal parameters, i.e frequencies, gains and decays, encode the geometry, dimensions and materials of the interacting actors. The modal description represents any sound as a sum of oscillators characterized by their frequency, amplitude and exponential decay. This recursive representation is efficient for real-time sound synthesis. In our case, modal parameters are extracted from impulse response recordings similar to [van den Doel 1996, van den Doel 2001]. The modal data is then made available on the fly for real-time sound synthesis. According to [van den Doel 2001], the main characteristics of the impact forces are the energy transfer and the hardness which affect duration and magnitude of the force respectively. We experimented with a number of force profiles and the exact details of the shape were found to be relatively unimportant, the hardness being well conveyed by the duration. Our method modulates the duration of the force by

[2] A shader in the field of computer graphics is a set of software instructions, which is used primarily to calculate rendering effects on graphics hardware with a high degree of flexibility. Shaders are used to program the graphics processing unit (GPU) programmable rendering pipeline, which has mostly superseded the fixed-function pipeline that allowed only common geometry transformation and pixel shading functions; with shaders, customized effects can be used (Source: Wikipedia).

the kinetic energy of the interaction.

An arbitrary number of simulated textures, or *patches*, leading to complex contact interactions, (see Section 5.3), are also maintained for real-time sound rendering. This organization is suggested by object-oriented programming systems, in which objects, i.e classes of texture patches, maintain their state. Each patch is associated with a block of code that implements its particular dynamics model and exports a set of editable parameters to the user interface so that the model may be varied interactively. During real-time processing, data from the rigid-body simulation such as velocity, force and positions, modal parameters and discontinuity maps of available textures are gathered to render the resulting sound. With our method, time performances for sound rendering during impact and continuous contact are 0.01msec and 0.3msec respectively. The time increase is acceptable when compared to the prohibitive cost of generating contact reports from complex geometries.

We implemented an interactive system based on our approach (see Figure 5.9). The simulation was driven by the physics engine *Ageia's PhysX*[3]. Our test application allows the user to interact with an object, (a capsule), making it roll or jump on a floor. The texture of the floor, its scale and the material of the objects interacting can be modified in real-time in order to experience the differences in the resulting sound rendering.

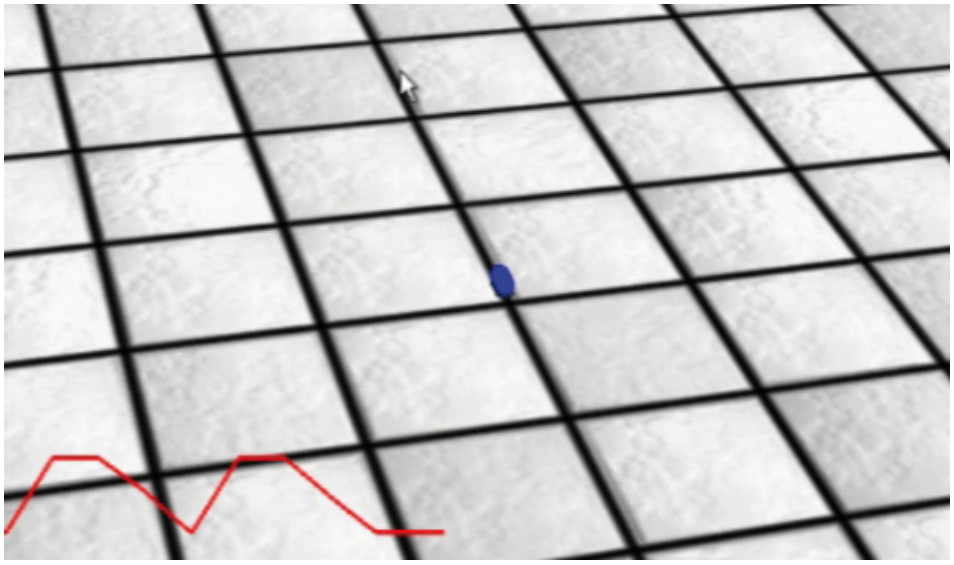

Figure 5.9: *Audio-visual interface: a user controls a capsule interacting with a tiled floor to experiment with the resulting sound. The amplitude of the excitation force is computed and rendered to the screen in real-time (red curve).*

[3]AGEIA PhysX SDK 2.7: http://www.ageia.com/

5.5 Discussion and Limitations

We demonstrated a number of applications of our approach that can be used in the context of interactive sound rendering for animations. Our approach shares a number of similarities with computer graphics and haptic techniques and in particular with the approach of [van den Doel 2004] where an *impact map* is presented for a two-dimensional parameterization of the impact locations and characteristics. We also introduce a two-scale approach for capturing and storing the geometric details for sound synthesis under contact situations. Our study is analogous to the approach of [Otaduy 2005] where the haptic texture rendering is synthesized based on the gradient of the directional penetration depth. Similarly, our method proposes adding more details to the coarse representation of the texture which are not sufficient for sound rendering. Fine geometric details are coded as discontinuity maps and are related to the excitation profiles. Finally, our approach is related to the work of Lécuyer et al. [Lécuyer 2000] that presents interaction techniques for simulating contact without a haptic interface. They give the illusion of perceiving the relief of an image when moving a mouse cursor over it. The speed and movement of the cursor are artificially modified.

Our approach presents one main limitation: it does not yet consider the case of two interacting textures, where features intersect creating specific excitation profiles. This will be examined in future work.

5.6 Conclusion

We have presented a novel approach for generating contact sounds for interactive simulations. This was achieved by considering the two-dimensional visual textures of objects in interaction as roughness maps to create audible and position-dependent variations during rolling and sliding. This approach allows us to guarantee coherence between visual and sound rendering. The physics engine is used to compute the motion of the objects, and the excitation force can be synthesized based upon the position of the contact in the texture-space. The force is used to excite the modal resonances of the sounding objects. A flexible audio pipeline was introduced, proposing different levels of detail which can be chosen according to the desired granularity of the rendered impacts.

We believe that improving contact sounds for interactive virtual environments is relevant and has many applications. Our technique is useful since video games increasingly incorporate procedurally-generated content to increase variety and realism but also to address memory savings, as the complexity of the environments grows.

Chapter 6

Resonator Modeling

Contents

In a sound source modeling paradigm, the physics-based sound synthesis system can be decomposed in two functional elements, exciter and resonator. The sounding object or resonator receives the energy from the exciter. The resonator vibrates, creating a sound with a specific timbre. In this chapter, we present the fundamentals of a physics based model for the resonator, that is the vibration model. We explain the motivation for such a model, and we give the main principles to create it. Finally, we present methods that have been developed for controlling vibration models.

6.1 General Concepts

Vibrating bodies are sources of acoustic waves and produce sound. A vibrating system can be characterized by a discrete set of eigenvalues which correspond to the natural frequencies and their associated decay rates. When the system is excited by an external force of finite duration, some of these frequencies will be stimulated. The relative amplitudes after the application of the force depend on the nature of the excitation. Typically, when the system is struck, the applied force is of very short duration and contains many frequencies. The emitted sound is perceived as containing a click of very short duration, being a mixture of many frequencies, and a sustained part, which contains only a few frequencies and is the characteristic pitch of the system.

For precise modeling of surface vibrations, the best method is to apply classical mechanics to the problem while considering the object as a continuous entity. This method yields equations for which solutions can be anatically computed [van den Doel 1999], allowing for computational efficiency and real-time

sound synthesis. However, the approach can only handle simple systems, such as bars or plates, for which the analytical solution is known. Previous research on real-time sound synthesis have proposed to model the modes of acoustical systems using resonant filters [Adrien 1991, Doel 1998, Cook 1999] or additive sinusoidal synthesis [Iovino 1997, van den Doel 2001].

Advantages for Interactive Synthesis

Vibration models are beneficial mainly due to their ease of control, efficiency, and simplicity of implementation, that are especially suited for interactive sound design. The vibrational data, namely the eigenvalues and the eigenvectors, allow the representation of an extremely large amount of data, that can be directly used for interactive simulations.

Interactive applications require responsiveness and flexibility in associating motion to sound control. Using vibration models, the parameters needed to characterize collision sounds, that is, relative velocity at collision and penetration force, are computed in the physical simulation engine and can be directly mapped into control parameters of the model. Therefore, this allows the creation of a sound feedback that responds in a natural way to gestures and actions.

6.2 Creating Vibration Models

There has been much work in computer music [Cook 2002b, Iovino 1997] and computer graphics [O'Brien 2002b, Raghuvanshi 2006, van den Doel 2001] exploring methods for generating sound based on physical simulation. Most approaches target sounds emitted by vibrating solids and consist in rendering the microscopic deformations of the sounding object. In the following, we review the methods used for numerically solving the sound emission from vibrating objects. We present the Finite Element Method (FEM), the Boundary Element Method (BEM), and mass-spring systems, which are Lagrangian mesh based methods, i.e., the model consists of a set of points with varying locations and properties. Finally, we introduce modal synthesis which belongs to reduced deformation models. For an extensive review of physically based deformable models in computer graphics, we refer the reader to the article [Nealen 2006].

6.2.1 The Finite Element Method

Instead of directly applying classical mechanics to the continuous system, suitable discrete approximations of the object geometry can be performed, making the problem more manageable for mathematical analysis. The first step of any finite element simulation is to discretize the actual geometry of the structure using a collection of finite elements. Each finite element represents a discrete portion of the physical structure and the finite elements are joined by shared nodes. The collection of nodes

and finite elements is called a mesh. The nodal displacements can be obtained using implicit or explicit methods (see Section 3.3).

O'Brien et al. [O'Brien 2001] introduces a nonlinear Finite Element Method (FEM) to explicitly model the response of an object to external forces. Audio is generated by analyzing the computed surface behavior. They apply a set of filters to the computed motion for extracting frequency components that fall within the audible range. The Rayleigh method, that is simulated for the first time in computer graphics, is used to simulate the radiated sound field. In this work, each element of the vibrating surface is treated independently, summing each element's contribution together. The use of a nonlinear finite element method allows the modeling of sounds that arise from nonlinear behaviors. The formulation is general but its main limitation is the computational effort that it requires, preventing its application to real-time manipulation. In addition, considerable inaccuracy can come with speed since the method completely ignores reflections and diffractions.

6.2.2 The Boundary Element Method

The Boundary Element Method (BEM) offers an alternative to the Finite Element approach by computing the equations on the surface (boundary) of the elastic body instead of on its volume (interior), allowing reflections and diffractions to be modeled. For real-time sound synthesis, Doug L. James et al. [James 2006] investigate diffraction and interreflection effects by realistic sound radiation from a rigid body. The technique consists in preprocessing the linear vibration modes of an object and in associating each mode to its sound pressure field or acoustic transfer function. The Precomputed Acoustic Transfer (PAT) functions are computed based on accurate approximations to Helmholtz equation solutions using the BEM. At runtime, simulation of new interaction sounds is directly performed by adding contributions from each mode's equivalent multipole sources. In contrast with the work of O'Brien et al. [O'Brien 2001], where artifacts appear in the radiated sound due to the use of polygons, the PAT method allows the rendering of the sound's timbre changes due to interaction with the object's own geometry. However, the method becomes inaccurate when frequencies increase due to the the use of multi-point multipole expansions which are difficult to evaluate for higher frequency radiation. To address this problem, Chadwick et al. [Chadwick 2009] introduce Far-Field Acoustic Transfer (FFAT) maps which direct low- to high-frequency far-field radiation complexity by applying (1) fast multipole boundary element methods from acoustics and (2) texture-based far-field expansions that are adequate to modeling rapid angular variations with simple radial structure.

6.2.3 Mass-Spring Systems

Similar to finite elements, mass-spring systems, also referred to as mass-interaction, cellular or particle systems, discretize an object by decomposing the system in its small pair-like elements. Mass-spring systems are arguably the simplest and most

intuitive of all deformable models. It simply consist of point masses connected together by a network of massless springs and the motion of each particle is then governed by Newton's second law. Therefore, mass-spring systems only require the solution of a system of coupled ordinary differential equations (ODEs) (see Section 3.3).

Modeling the surface vibrations of objects using discretized physical models in real time was introduced by Florens and Cadoz [Florens 1991], who proposed a system of masses and damped springs to model 3D shapes. The CORDIS-ANIMA system was then developed for physically based sound synthesis. A renewed interest, probably due to the intuitiveness of the mass-spring metaphor, was presented by Raghuvanshi and Lin [Raghuvanshi 2006] for sound vibrations. A spring-mass model is constructed to approximate an object's surface deformation based on the geometry and a few material parameters. It appears that despite their coarser approximation than FEM models used in prior approaches [O'Brien 2001, O'Brien 2002b], mass-spring systems appropriately model the micro vibrations of surfaces that produce sound. The technique similarly offers discretization of the object geometry but in a more straightforward way and with an easier implementation. In compare to OBrien et al. [O'Brien 2002b] who manage to handle a maximum of only 10 sounding objects and to synthesize mainly impacts, the method extends to hundreds of objects in real-time with acceleration techniques based on auditory perception, and is also capable of producing realistic rolling sounds in addition to impact sounds.

6.2.4 Modal Synthesis

Modal synthesis, as any kind of *additive* synthesis, consists in describing a source as many components which are added. A vibrating object is modeled by a bank of damped harmonic oscillators which are excited by an external stimulus. The modal model consists in the vector of the modal frequencies, the vector of the decay rates and the matrix of the gains for each mode at different locations on the surface of the object. The frequencies and dampings of the oscillators are governed by the geometry and material properties of the object, whereas the coupling gains of the modes are determined by the mode shapes and are dependent on the contact location on the object [van den Doel 2001].

Because the object being analyzed can be of arbitrary shape, the Finite Element Method (FEM) is commonly used to perform modal analysis, which in general gives, satisfactory results. The natural frequencies are determined assuming the dynamic response of the unloaded structure, with the equation of motion. The system has n eigenvalues, where n is the number of degrees of freedom in the finite element model. Let λ_j be the jth eigenvalue, its square root, ω_j , is the natural frequency of the jth mode of the structure, and ϕ_j is the corresponding jth eigenvector. The eigenvector, also known as the mode shape, is the deformed shape of the structure as it vibrates in the jth mode. The natural frequencies and mode shapes of a structure can be used to characterize its dynamic response to loads in the linear regime. The deformation

of the structure can be calculated from a combination of the mode shapes of the structure using the modal superposition technique. The vector of displacements of the model, u, is defined as

$$u = \sum_{n}^{1} \alpha_i \phi_i \tag{6.1}$$

where α_i is the scale factor for mode ϕ_i. For more details on modal superposition, we refer the reader to Appendix A.

The response of a system is usually governed by a relatively small part of the modes, which makes modal superposition a particularly suitable method for computing the vibration response. Thus, if the structural response is characterized by n modes, only n equations need to be solved. In addition, in contrast with the original equations that are coupled, the modal equations are uncoupled. Finally, the initial computational expense in calculating the modes and frequencies is largely offset by the savings obtained in the calculation of the response.

Modal synthesis is valid only for linear problems, that is, simulations with small displacements, linear elastic materials, and no contact conditions. If the simulation presents nonlinearities, significant changes in the natural frequencies may appear during the analysis. In this case, direct integration of the dynamic equation of equilibrium is needed, which requires much more computational effort. Basically, efficiency of modal analysis relies on neglecting the spatial dynamics and modeling the actual physical system by a corresponding mass-spring system which has the same spectral response. However, the spatial dynamics, and in particular the propagation of disturbances, can be preserved if the modal shapes are known.

Computing Modal Parameters by Simulation

In contrast with the previous work [O'Brien 2001], O'Brien et al. [O'Brien 2002b] present a real-time method that can accurately model sounds produced by linear phenomena. Modal analysis is applied to produce realistic and compelling sounds of rigid objects. Given that the amount of elastic deformation experienced is small, the authors consider that there is no interaction between the rigid body modes for an object and its deformation modes. They present a method that can discretize any given arbitrarily-shaped object with tetrahedral volume elements. The finite element representation is then used to numerically evaluate the object's deformation modes with an eigen-decomposition of the system matrices. In order to decouple the damped system into a single degree-of-freedom oscillator, Rayleigh damping is assumed (see, for instance, [Bathe 1982]). The general form of the system for eigen-decomposition is then obtained, from which the modal parameters, i.e., frequencies, dampings, and corresponding gains are extracted. To approximate the coupling between vibrations on the surface of the object and vibrations in the surrounding medium, a coupling coefficient is evaluated for each mode by adding the amount of normal displacement produced by that mode multiplied by the mode's frequency. The coefficient obtained for each mode then scales the response of the corresponding mode's oscillator. The final sound produced by the system is obtained by summing

the scaled oscillators, treating all objects as omni-directional sources. Finally, by evaluating the modal decomposition in a preprocess and storing it with the corresponding object, efficient runtime computation for vibrational response to contact forces can be easily managed.

Fitting Modal Parameters to Empirical Data

Modal data can be extracted from recorded sounds of real objects. As an example, Pai et al. [Pai 2001] estimate the modal model by exciting the object with an arbitrary force. The audio response and the input force are then both measured at the same high rate, and the input force is deconvolved from the audio signal. However, measuring forces at audio frequencies requires very stiff force sensors to avoid the influence of the resonances of the sensor itself, which may make the approach difficult to handle. Deconvolution is also a delicate inverse problem. In the study, a device is proposed to apply a light, highly peaked force, that adequately approximates an impulsive force. The work from Corbett et al. [Corbett 2007] broaden modal synthesis to enable the spatial variations of timbre in the 3D space around the object, as well as variation due to the changes in contact point (2D) and contact type. They extract model parameters from measurements using an automated robotic remote controlled measuring system, coupled with an original parameter fitting algorithm. The parameters are retrieved from a variety of sound recordings around objects and a continuous timbre field is produced by interpolation, leading to accurate values for the modal frequencies, dampings, and gains. The method handles efficient real-time audio synthesis in multimodal interactive simulations. However, the main drawback of extracting modal parameters from empirical data is that arbitrary 3D models have to be physically procured.

6.3 Control for Vibration Models

Modal sound synthesis can adequately manage complex impact sounds, such as those caused by falling objects or explosion debris. However, mode-based computations can become extremely expensive when numerous objects, each with many modes, come into contact simultaneously. Van den Doel et al. [van den Doel 2002] point out that accurate selection of the modal parameters may significantly increase the efficiency of the synthesis. In later work [van den Doel 2004], they propose a novel method that eliminate modes based on human auditory perception, and in particular, auditory masking (see Section 2.2.3). The technique enables a large number of modes to be removed without any decline in sound quality. Auditory masking is a highly non-linear phenomenon and is dependent on the excitations and the observation point. The audible modes are dynamically chosen at a moderate rate compared to the audio sampling rate; only the modes above the upper envelope of all the masking curves are maintained. The computational expense is reduced by first ordering the modes by loudness and then removing modes that are masked

by the loudest mode. This enables speed improvement of the audio synthesis by a factor of 7-9. In a similar approach, *mode compression* and *mode truncation* based on auditory perception are introduced by Raghuvanshi et al. [Raghuvanshi 2006] to accelerate the sound simulation. *Mode compression* is based on the observation that humans have a limited ability to distinguish between nearby frequencies, which is different from frequency masking where two sounds are considered simultaneously. *Mode truncation* is related to the structure of the sound of a typical object. They also point out that the transient attack of the signal is fundamental to the quality of sound, which is related to the perception of the timbre. In order to ensure simultaneously tight time constraints and an immersive sound experience, a priority-based scheme dynamically manages sound quality and the associated computational cost for each object. In particular, the foreground sounds which much more easily attract the user's attention are rendered with higher quality than background sounds.

Tsingos et al. [Tsingos 2004] propose an approach based on the preprocess of perceptual data for culling, masking and prioritizing sounds in real-time. The follow-up work [Tsingos 2005, Moeck 2007] extend the approach to a entirely scalable processing pipeline that takes advantage of the sparseness of the input audio signal in the Fourier domain to handle complex mixture with scalable or progressive rendering. Audio spatialization of several thousands of sound sources is made available via clustering. However, since the method is based on precomputed data, sounds synthesized in real-time, such as modal sounds, are not dealt with. To address this, a fast sound synthesis approach that exploits the inherent sparsity of modal sounds in the frequency domain has recently been introduced by Bonneel et al. [Bonneel 2008]. The technique consists in performing frequency-domain modal synthesis by fast summation of a few Fourier coefficients using a formulation that efficiently approximates the short-time Fourier Transform (STFT) for modes. Compared to time-domain modal synthesis [van den Doel 2004], the audio synthesis is speeded up by a factor of 5-8, with minor degradation in quality.

Given the widespread use and availability of computer graphics hardware, several studies investigate graphics processing units (GPU) for computationally efficient rendering of sound. The study of van den Doel et al. [van den Doel 2004] introduces an *impact map* to reduce the amount of computation on the computer's CPU: the detailed simulation of impact events usually supported using FEM is formulated in order to be supported by the GPU, and the pixels are read from the rendering surface to detect impacts. . The work from Zhang et al. [Zhang 2005] addresses the difficult task of synthesizing numerous modal sounds in real-time by introducing parallelism and using programmability in graphics pipeline. More recently, Trebien et al. [Trebien 2009] propose a method to efficiently implement general filtering on the GPU and in particular solve the problem of recursive filters, based on recursive feedback coefficients. On the other hand, Gallo et al. [Gallo 2004] investigate the use of GPU for 3D audio rendering applications, addressing dynamic and interactive games and virtual environments.

The computation time required by current methods to preprocess the vibration

models prevents it from being used for real-time rendering, except for mass-spring systems. Maxwell and Bindel [Maxwell 2007] study how the change of the shape of a finite element model affects the sound emission. The study shows that it is possible to avoid recomputing the eigen-decompositions only for moderate changes.

There has been much work in controlling the computational expense of modal synthesis, allowing the simultaneous handling of a large variety of sounding objects. However, to be even more efficient, flexibility should be included in the design of the model itself, in order to control the processing. Thus, modal synthesis should be further developed in terms of parametric control properties.

Chapter 7

A Robust and Multi-Scale Modal Analysis for Sound Synthesis

Contents

In order to improve current modal analysis for sound rendering, we propose a new approach that preserves sound variety across the surface of an object, at different scales of resolution and for a variety of complex geometries. This work was published at *the International Conference on Digital Audio Effects (DAFx-09)*, in Como, Italy [Picard 2009a]. This Chapter first introduces the context in which the method is being applied. Our approach for modal analysis is then described in detail. A validation process is performed to provide objective evidence of the accuracy of the model. Results on robustness and multi-scale modeling are then presented. Finally, we discuss on the relevance of the sound synthesis model for virtual applications.

7.1 Introduction

Our goal is to realistically model sounding objects for animated real-time virtual environments. To achieve this, we propose a robust and flexible modal analysis

approach that efficiently extracts modal parameters for persuasive sound synthesis while also focusing on efficient memory usage.

Modal synthesis models the sound of an object as a combination of sinusoids, each of which oscillates independently of the others. Modal synthesis approaches are only accurate for sounds produced by linear phenomena, but they can compute these sounds in real-time. Modal synthesis requires the computation of a partial eigenvalue decomposition of the system matrices, which is relatively expensive. For this reason, modal analysis is performed in a preprocessing step. The eigenvalues and eigenvectors strongly depend on the geometry, material and scale of the sounding object. Therefore, modeling numerous sounding objects can rapidly become prohibitively expensive. In addition, this processing step can be subject to computation problems; in particular, when the geometries are non-manifold.

We propose a new approach to efficiently extract modal parameters for any given geometry, overcoming many of the afore mentioned limitations. Our method employs bounding voxels of a given shape at arbitrary resolution for hexahedral finite elements. The advantages of this technique are the automatic voxelization of a surface model and the automatic tuning of the finite element method (FEM) parameters based on the distribution of material in each cell. A particular advantage of this approach is that we can easily deal with non-manifold geometry which includes both volumetric and surface parts. These kinds of geometries cannot be processed with traditional approaches which use a tetrahedralization of the model (e.g., [O'Brien 2002b]). Likewise, even with solid watertight geometries, complex details often lead to poorly shaped tetrahedra and numerical instabilities; in contrast, our approach does not suffer from this problem. Our specific contribution is the adaptation of the multi-resolution hexahedral embedding technique to modal analysis for sound synthesis. Most importantly, our solution preserves variety in what we call the *Sound Map*, that is, the changes in sound across the surface of the sounding object.

7.2 Method

In the case of small elastic deformations, rigid motion of an object does not interact with the objects's vibrations. On the other hand, we assume that small-amplitude elastic deformations will not significantly affect the rigid-body collisions between objects. For these reasons, the rigid-body behavior of the objects can be modeled in the same way as animation without audio generation.

7.2.1 Deformation Model

Our implementation uses the *Sofa Framework*[1] for rigid-body simulation. This choice was motivated by the ease with which it could be extended for our purpose.

[1] Simulation Open Framework Architecture http://www.sofa-framework.org/

The main feature of SOFA compared with other libraries is its high flexibility while maintaining efficiency. SOFA is an open-source C++ library for physical simulation. It can be used as an external library in another program, or using one of the associated GUI applications. It allows the use of multiple interacting geometrical models of the same object, typically, a mechanical model with mass and constitutive laws and a collision model with simple geometry. A visual model with detailed geometry and rendering parameters is also integrated, where each model can be designed independently of the others. During run-time, consistency is maintained using mappings. Additionally, SOFA scenes are modeled using a data structure similar to hierarchical scene graphs which allows the physical objects to be split easily into collections of independent components, each describing one feature of the model. Moreover, simulation algorithms are also modeled as components in the scene graph, providing us with the same flexibility for algorithms as for models.

Elastic deformations are used to generate the audio signal. Before performing modal decomposition, we must first select a deformable modeling method that can be used to generate the stiffness and the mass matrices of the mechanical system. A variety of methods could be used, including particle systems [Raghuvanshi 2006] or finite differences methods. The tetrahedral finite element method has also been used [O'Brien 2002b]. However, tetrahedral meshes are computationally expensive for complex geometries, and can be difficult to tune. As an example, in the tetrahedral mesh generator *Tetgen*[2], the mesh element quality criterion is based on the minimum radius-edge ratio, which limits the ratio between the radius of the circumsphere of the tetrahedron and the shortest edge length.

Our method is inspired from work by Nesme et al. [Nesme 2006]. It uses hexahedral finite element for computing the mass and stiffness matrices of the mechanical system. The technique can be summarized as follows. An automatic high-resolution voxelization of the geometric object is first built. The voxelization initially concerns the surface of the geometric model, while the interior is automatically filled when the geometry represents a solid object. The voxels are then recursively merged up to an arbitrary coarser mechanical resolution. The merged voxels are used as hexahedral finite elements embedding the detailed geometrical shape. At each level, the mass and stiffness of a merged voxel are deduced from its eight children, using a weighted average that takes into account the distribution of material. With this method, we can handle objects with geometries that simultaneously include volumetric and surface parts; thin or flat features will occupy voxels and will thus result in the creation of mechanical elements that approximate their shape (see Section 7.4.1).

We extend the method for microscopic deformations that allows sound rendering. Thus, in order to compute the modal parameters, we compute the assembled mass and the assembled stiffness matrices for the object by summing the contribution of each cell. Then, we solve the decoupled system to extract the modal parameters as explained in Section 6.2.4. Our preprocessing step that performs modal analysis

[2] http://tetgen.berlios.de/

can be summarized as follows.

Algorithm for modal parameters extraction.	
1.	Compute mass and stiffness at desired mechanical level
2.	Assemble the mass and the stiffness matrices
3.	Modal analysis: solve the eigenproblem
4.	Store eigenvalues and eigenvectors for sound synthesis

The model approximates the motion of the embedded mesh vertices. That is, the visual model with detailed geometry does not match the mechanical model on which the modal analysis is performed. The motion of the embedding uses a trilinear interpolation of the mechanical degrees of freedom (DOFs), so we can nevertheless compute the motion of any point on the surface given the mode shapes.

7.2.2 Sound Generation

When rendering the sound with a modal synthesis approach, we do not solve the emission problem, but instead we consider the sound to be simply a sum of damped sinusoids. The activation of this model depends on where the object is hit. If we hit the object at a vibration node of a mode, then that mode will not vibrate, but others will. This is what we refer to as the *Sound Map*, which could also be called a sound excitation map as it indicates how the different modes are excited when the object is struck at different locations. For our approach, the calculations for modal parameters are similar to the ones presented in the paper of O'Brien et al. [O'Brien 2002b] and we refer the reader to this work for additional information.

The sound resulting from an impact on a specific location on the surface is calculated as a sum of n damped oscillators:

$$s(t) = \sum_{n}^{1} a_i \sin(w_i t) e^{-d_i t} \tag{7.1}$$

where w_i, d_i, and a_i are respectively the frequency, the decay rate and the gain of the mode i. In our method, we synthesize the sounds via a reson filter (see, for example, Van den Doel et al. [van den Doel 2001]). This choice is made based on the effectiveness for real-time audio processing. No radiation properties are considered; our study focuses specifically on effective modal synthesis. However, radiation can be computed in a number of ways [O'Brien 2001, James 2006]. As the motions of objects are computed with modal analysis, surfaces can be easily analyzed to determine how the motion will induce acoustic pressure waves in the surrounding medium. Finally, our study does not consider contact-position dependent damping or changes in boundary constraints, as might happen during moments of excitation. Instead we use a uniform damping value for the sounding object.

7.3 Validation of the Model

7.3.1 A Metal Cube

In order to globally validate our method for modal analysis, we study the sound emitted when impacting a cube in metal. This example is interesting due to its symmetry. In particular, the cube should sound the same when impacting normal to the face at the eight corners. The sound emitted should also be similar when hitting with forces normal to the pair of the cube faces.

We suppose the cube is made of steel with density 7850 kg/m^3, the Raleigh coefficients $\alpha 1$ and $\alpha 2$ are equal to 3x10^{-7} and 10 respectively. A Dirac is chosen for the excitation force. From the results, see Appendix B, the frequency content is shown to be dependent on the resolution of the hexahedral finite elements. A 1×1×1 or 2×2×2 grid represents an extremely coarse embedding, and consequently may not be accurate to properly synthesize the sound of objects. The higher models have a wider range of frequencies because of the supplementary degrees of freedom. There is a frequency centroid shift as the FEM resolution increases. Most importantly, the frequency content converge as the FEM increases. The resulting sounds when impacting on one corner with three perpendicular forces, each normal to one pair of cube faces, are quite similar. In addition, the resulting sounds when impacting on different corners of a cube are similar.

7.3.2 Position Dependent Sound Rendering

To properly render impact sounds of an object, the method must preserve the sound variety when hitting the surface at different locations. We consider a metal bowl, modeled by a triangle mesh with 274 vertices. We take 3 different locations, i.e.,

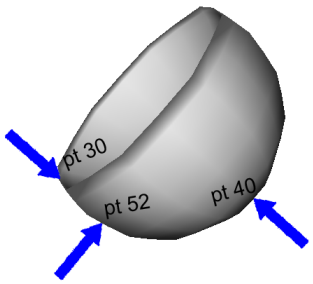

Figure 7.1: *A sounding metal bowl: sound synthesis is performed for excitation on specific locations on the surface: points 30, 40 and 52.*

top, side and bottom, on the surface of the object where the object is impacted, see Figure 7.1. The excitation force is modeled as a Dirac, such as a regular impact. The material of the bowl is aluminium, with the parameters 69×10^9 for Young Modulus, 0.33 for Poisson coefficient, and 2700 kg/m^3 for the volumic mass. The Rayleigh

damping parameters for stiffness and mass are set to 3×10^{-7} and 10. The use of a constant damping ratio is a simplification that still produces good results.

We compare our approach to modal analysis performed first using tetrahedralization with *Tetgen*[3] with 822 modes. Our method uses hexahedral finite elements and is applied with a grid of 2×2×2 cells, leading to 81 modes. However, to adapt the stiffness of a cell according to its content, the mesh is refined more precisely than desired for the animation. The information is propagated from fine cells to coarser cells. For this example, the elements of the 2×2×2 coarse grid resolution approximates mechanical properties propagated from a fine grid of 4×4×4 cells.

The frequency content of the sound resulting from impact at the 3 locations on the surface is shown in Figure 7.2. Each power spectrum is normalized with

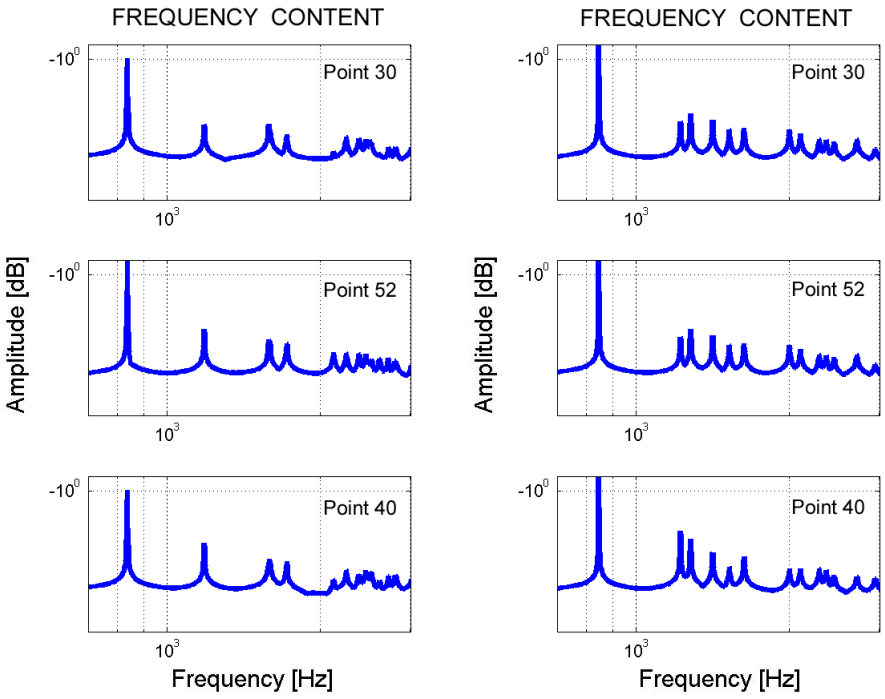

Figure 7.2: *Sound synthesis with a modal approach using classical tetrahedralization with 822 modes (left) and our method with a 2×2×2 hexahedral FEM resolution, leading to 81 modes (right): power spectrum of the sounds emitted when impacting at the 3 different locations shown in Figure 7.1, (from top to bottom) points 30, 52 and 40.*

the maximum amplitude in order to factor out the magnitude of the impact. The eigenvalues that correspond to vibration modes will be nonzero, but for each free body in the system there will be six zero eigenvalues for the body's six rigid-body

[3] Tetrahedral Mesh Generator: http://tetgen.berlios.de/

freedoms. Only the modes with nonzero eigenvalue are kept. Thus, 816 modes are finally used for sound rendering with the tetrahedralization method and 75 with our hexahedral FEM method.

The movie provided[4] compares the sounds synthesized with the tetrahedral FEM and the hexadedral FEM approaches. While Figure 7.2 highlights the visual differences in the frequency content, we notice in listening to the synthesized sounds that those generated by our method are quite similar to those created with the standard tetrahedralization, even when significantly fewer vibration modes are used (i.e., 75 in contrast to 816).

7.4 Robustness and Multi-Scale Results

Computing modes for complex geometries can become prohibitively expensive especially when numerous sounding objects have to be processed. As an example, the actual cost of computing the partial eigenvalue decomposition using a tetrahedralization in the case of a bowl with 274 vertices and generating 2426 tetrahedras is 5 minutes with an Intel Core Duo with 2.33 GHz and 2 GB of memory. The number of tetrahedras determine the dimension of the system to solve. To avoid this expense, we provide a method that greatly simplifies the modal parameter extraction even for non-manifold geometries that include both volumetric and surface parts. Our technique consists of using multi-resolution hexahedral embeddings.

7.4.1 Robustness

Most approaches for tetrahedral mesh generation have limitations. In particular, an important requirement imposed by the application of deformable FEM is that tetrahedra must have appropriate shapes, for instance, not too flat or sharp. By far the most popular of the tetrahedral meshing techniques are those utilizing the Delaunay criterion [Shewchuk 1998]. When the Delaunay criterion is not satisfied, modal analysis using standard tetrahedralization is impossible. In comparison with tetrahedralization methods, our technique can handle complex geometries and adequately performs modal analysis. Figure 7.3 gives an example of problematic geometry for tetrahedralization because of the presence of very thin parts, specifically the blades that protrude from either side.

We suppose the object is made of aluminium (see Section 7.3.2 for the material parameters). We apply a coarse grid of 7×7×7 cells for modal analysis. The coarse level encloses the mechanical properties of a fine grid of 14×14×14 cells. Figure 7.5 shows the power spectrum of the sounds resulting from impacts, modeled as a Dirac, on 5 different locations. Each power spectrum is normalized with the maximum amplitude of the spectrum in order to factor out the magnitude of the impact.

[4] Additional material: http://www-sop.inria.fr/reves/Cecile.Picard

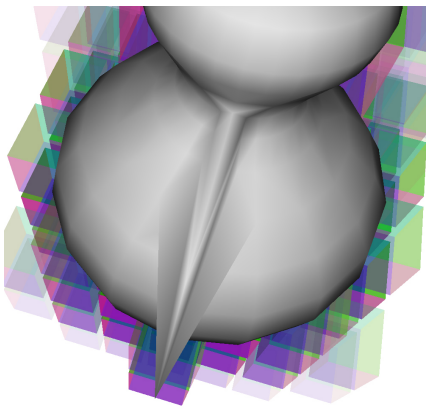

Figure 7.3: *An example of a complex geometry that can be handled with our method. The thin blade causes problems with traditional tetrahedralization methods.*

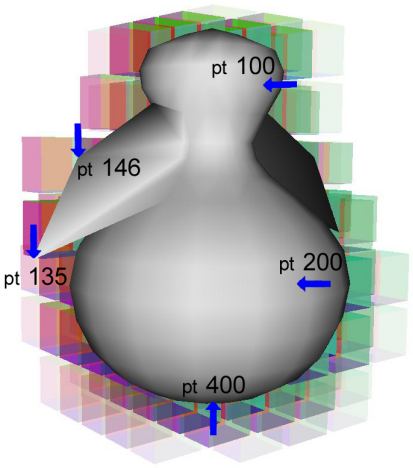

Figure 7.4: *Test impacts for sound generation are simulated on 5 different locations on the surface of the complex geometry: points 100, 135, 146, 200 and 400.*

Figure 7.5 shows that the *Sound Map* is preserved; we can observe that the different modes have varying amplitude depending on the location of excitation. It is interesting to examine the quality of the sound rendered when hitting the wings. Because this part is lightweight compared to the rest of the object, the amplitude of higher frequencies is more pronounced than at other locations.

7.4.2 A Multi-Scale Approach

To study the influence of the number of hexahedral finite elements on the sound rendering, we model a sounding object with different resolutions of hexahedral finite elements. We have created a squirrel model with 999 vertices which we use as our test sounding object. Its material is pine wood, which has parameters 12×10^9 for Young

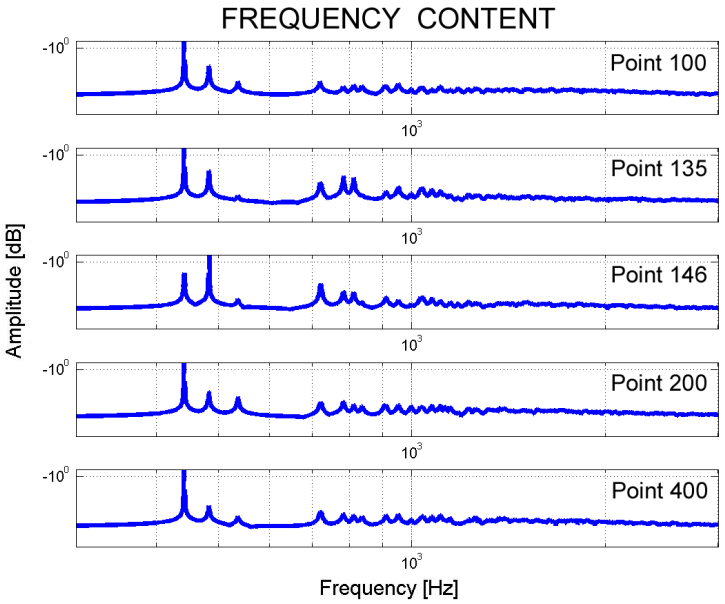

Figure 7.5: *The power spectrum of the sounds resulting from impacts at the 5 different locations shown in Figure 7.4: (from top to bottom) points 100, 135, 146, 200 and 400. Note that the audible response is different based on where the object is hit.*

Modulus, 0.3 for Poisson coefficient, 750 kg/m^3 for the volumetric mass. Rayleigh damping parameters for stiffness and mass are set to 8×10^{-6} and 50 respectively. Sound synthesis is performed for 3 different locations of excitation, see Figure 7.6 (top left). The coarse grid resolution for finite elements is set to 2×2×2, 3×3×3, 4×4×4, 8×8×8 and 9×9×9 cells. In this example, the finer grid resolution is one level up to the one of coarse grid, that is, a coarse grid of 2×2×2 cells has a fine level of 4×4×4 cells.

Results show that the frequency content of sounds depend on the location of excitation and on the resolution of the hexahedral finite elements. The higher resolution models have a wider range of frequencies because of the supplementary degrees of freedom. We also observe a frequency shift as the FEM resolution increases. Note that a 2×2×2 grid represents an extremely coarse embedding, and consequently it is not surprising that the frequency content is different at higher resolution. Nevertheless, there are still some strong similarities at the dominant frequencies. Above all, an important feature is the convergence in frequency content as the FEM increases. According to Figure 7.6, a grid of 4×4×4 cells may be sufficient to properly render the sound quality of the object.

7.4.3 Limitations

The *Sound Map* is influenced by the resolution of the hexahedral finite elements. This is related to the way stiffnesses and masses of different elements are altered based on their contents. As a consequence, a 2×2×2 hexahedral FEM resolution would show much less expressive variation than higher FEM resolution. This is shown in the movie[5]. One approach to improving this would be to use better approximations of the mass and stiffness of coarse elements [Nesme 2009].

Nevertheless, based on the quality of the resulting sounds, and given that increased resolution for the finite elements implies higher memory and computational requirements for modal data, finite elements resolution can be adapted to the number of sounding objects in the virtual scene.

7.5 Discussion

Table 7.1 gives the computation time and the memory usage of the modal data when computing the modal analysis with different FEM resolution on the squirrel model. In this example, the finer grid resolution is one level up to the one of coarse grid, that is, a coarse grid of 4×4×4 cells has a fine level of 8×8×8 cells. These are

Coarse Grid Resolution (cells)	Computation Time (seconds)	Memory Usage (MB)
7×7×7	115.11	11.309
6×6×6	39.14	5.698
5×5×5	12.98	2.663
4×4×4	5.35	1.042

Table 7.1: *Computation time and memory usage for different grid resolutions.*

computation times of our unoptimized initial implementation on a 2.33 GHz Intel Core Duo.

Despite the fact that audio is considered a very important aspect in virtual environments, it is still considered to be of lower importance than graphics. We believe that physically modeled audio brings a significant added value in terms of realism and the increased sense of immersion.

7.6 Sound Synthesis for Virtual Environments

The use of physics engines is becoming much more widespread for animated interactive virtual environments. The study of [Menzies 2007] address the pertinence of physical audio within physical computer game environment. He develops a library

[5] Additional material: http://www-sop.inria.fr/reves/Cecile.Picard

whose technical aspects are based on practical requirements. Menzies emphasizes broader issues concerning the uptake of audio modeling within industry. Indeed, the interface between physics engines and audio has often been one of the obstacles for the adoption of physically based sound synthesis in simulations. This is often due to the lack of appropriate design choices in the two interfaces that prevent them from working together effectively.

O'Brien et al. [O'Brien 2001] employed finite elements simulations for generating both animated videos and audio. However, the method requires large amounts of computation, preventing from using it for real-time manipulation.

The method presented in [Picard 2009a] is built on a physically based animation engine, *Sofa Framework*. As a consequence, problems of coherence between physics simulation and sound synthesis are avoided by using exactly the same model for simulation and sound modeling. In contrast to the approach of O'Brien et al. [O'Brien 2001], the sound can be processed in real-time knowing the modal parameters of the sounding object.

7.7 Conclusion

We propose a new approach to modal analysis using automatic voxelization of a surface model and automatic tuning of the finite elements parameters, based on the distribution of material in each cell. Our goal is to perform sound rendering in the context of an animated real-time virtual environment, which has specific requirements, such as real-time processing and efficient memory usage.

We have shown that in simple cases our method gives similar results as traditional modal analysis with tetrahedralization for simple cases. For more complex cases, our approach provides persuasive results. In particular, sound variety along the object surface, the *Sound Map*, is well preserved.

Our technique can handle complex non-manifold geometries that include both volumetric and surface parts, which can not be handled by previous techniques. We are thus able to compute the audio response of numerous and diverse sounding objects, such as those used games, training simulations, and other interactive virtual environment.

Our solution allows a multi-scale solution because the number of hexahedral finite elements only loosely depends on the geometry of the sounding object.

Finally, since our method is built on a physics animation engine, the *Sofa Framework*, problems of coherence between simulation and audio can be easily addressed, which is of great interest in the context of interactive environment.

In addition, due to the fast computation time, we are hopeful that real-time modal analysis will soon be possible on the fly, with sound results that are approximate but still realistic for virtual environments.

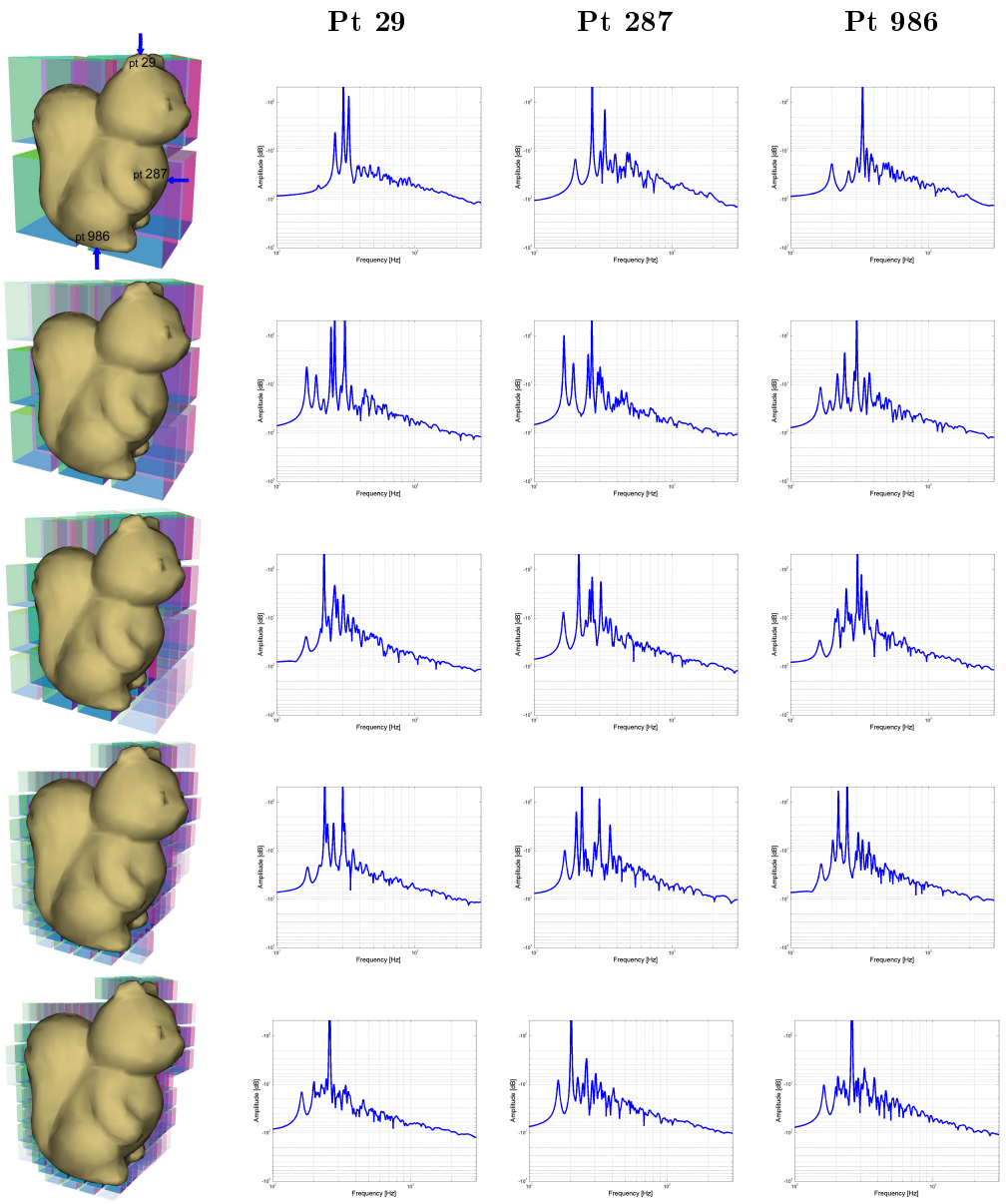

Figure 7.6: *A squirrel in pine wood is sounding when impacting on 3 different locations: points 29, 287 and 986 (from left to right). Frequency content of the resulted sounds with 5 different resolutions for the hexahedral finite elements: (from top to bottom), 2×2×2, 3×3×3, 4×4×4, 8×8×8 and 9×9×9 cells.*

In the previous part, we presented new physically based algorithms which enable sound rendering of a variety of interactions. In particular, we first presented an approach to direct excitation force that arises between objects when a sound interaction occurs. To address complex contact sounds, we introduced surface texturing. We then focused on sound modeling of resonating objects and we proposed an efficient technique that especially targets robustness and scalability of the audio representation. However, physical sound synthesis sometimes lacks realism. Much effort has been spent on making example-based synthesis more expressive. In the next part, we will investigate example-based techniques to extend the framework of standard sample-based audio generation.

Part III

Example-Based Sound Synthesis

Chapter 8

Techniques for Signal-Based Models

Contents

A *signal* typically means a measurable or observable quantity as a function of time and possibly as a function of place. Numeric computation in digital signal processing enables easy modification of signals, leading to great simplification for the design of algorithms compared to physical systems. Since digital signal processing is an advanced theory, many tools have been developed, especially addressing computational issues, and in particular maximal efficiency. Therefore, discrete-time signal processing approach is often preferred, particularly for real-time simulation and sound synthesis where efficiency is crucial. This Chapter provides the methods for representation, and transformation of audio signals. Content exploitation is addressed and various applications of signal-based sound models are presented.

8.1 Audio Content Representation

This section provides an overview of audio content representation according to low-level attributes and diverse audio meaningful tools such as segmentation or audio fingerprinting. Given that this section covers general introduction, we mainly based it on the review of Gouyon et al. [Gouyon 2008].

Before audio attribute extraction, the audio signal is first digitized, if necessary, and converted to a general format, e.g. mono Pulse Code Modulation (PCM) (e.g.,

16 bits) with a fixed sampling rate (ranging from 5 to 44.1 KHz). Basically, the signal is assumed to be stationary over intervals of a few milliseconds, and is then divided into frames of, for example, 10 ms. The frame rate defines the number of frames computed per second. To reduce discontinuities at the beginning and end of each segment obtained, a tapered window function, e.g., a Gaussian or Hanning window, is applied; further overlap for consecutive frames usually guarantees smoother analysis. The analysis step, namely the hop size, is equivalent to the frame rate minus the overlap.

8.1.1 Low-Level Audio Attributes

A low-level descriptor, also referred to as *signal-centered descriptor*, can be evaluated from the audio signals in a direct or derived way, as for example, after signal transformations like Fourier or Wavelet transforms, or after statistical processing like averaging. Although low-level descriptors are usually meaningless for the majority of users, their utilization by computing systems has many advantages.

Many different low-level attributes can be extracted from audio signals. Numerous techniques have been developed for signal modeling, mostly in computer music and speech processing, allowing for signal representations over which features can be computed. As an example, the Computer Audition Toolbox (CATbox)[1] provides us with various functions that are of interest to audition and related fields.

Temporal Attributes

A large variety of audio attributes can be supplied directly by the temporal representation of the frames, for instance, the mean (but also the maximum or the range) of the amplitude of the samples in a frame, the energy, and auto-correlation coefficients. Some low-level characteristics demonstrate significant correlation with perceptual attributes, as for example, amplitude is closely related to loudness.

Spectral Attributes

Audio attributes can also be extracted from the spectral representation of the frames. A spectrum is computed from each signal frame by applying a Discrete Fourier Transform (DFT), usually based on the Fast Fourier Transform (FFT), and this procedure is labeled Short-Time Fourier Transform (STFT) (for more details on formulas, we refer to the Appendix C). Other transforms can be applied instead of the DFT such as the Wavelet [Kronland-Martinet 1987] or the Wigner-Ville transforms [Cohen 1989]. Occasionally, the time-frequency representation can be refined with more perceptual motivated attributes that deal with the human auditory perception, such as the filtering performed by the middle-ear, loudness perception, temporal integration or frequency masking [Moore 1995]. From the representation

[1] Computer Audition Toolbox (CATbox): (available under the GNU license)
http://cosmal.ucsd.edu/cal/projects/CATbox/catbox.htm

obtained, a large variety of attributes can be deduced, for instance, the spectrum energy, energy values in several frequency sub-bands, the mean, geometric mean, spread, centroid, flatness, spectral slope and spectral flux. In addition, modeling of the spectral representation can be performed, as for example, through sinusoidal plus residual modeling [Serra 1989].

Cepstral Attributes

Mel-Frequency Cepstrum Coefficients (MFCCs) are extensively applied in speech research. MFCCs are coefficients that collectively formulate the mel-frequency cepstrum (MFC), that is, a representation of the short-term power spectrum of a sound based on a linear cosine transform of a log power spectrum, on a nonlinear mel scale of frequency. The relevance of this characteristic cepstrum is that the frequency bands used in the computation are equally spaced on the mel scale, more closely approximating the human auditory system's response. In particular, the cepstral representation enables us to nicely isolate the voice excitation (the higher coefficients) from the successive filtering carried out by the vocal tract (the lower coefficients). MFCCs are also increasingly finding uses in audio similarity measures. MFCCs can allow for better representation of sound, for example, in audio compression.

Temporal Evolution of Frame Attributes

In contrast to instantaneous, or frame, attributes, the temporal evolution of attributes can be of great interest to qualify an audio signal. It can simply be evaluated by deriving feature values, i.e. a first-order differentiator. The amount of change can also be computed with the differential of the attribute normalized with its magnitude.

8.1.2 Segmentation

Thanks to frame attributes, dimensionality with respect to the audio signal is significantly reduced. Further reductions can be performed by considering attributes on groups of consecutive frames, often called regions. In this way, detection of relevant region boundaries is of great importance for segmentation.

There are different ways to perform segmentation. Our study is more closely related to an adaptation of a classic definition coming from the field of visual segmentation [Pal 1993], and mentioned by [Gouyon 2008]: [*sound*] *segmentation is a process of partitioning* [*the sound stream*] *into non-intersecting regions such that each region is homogeneous and the union of no two adjacent regions is homogeneous. Homogeneity* may refer here to a property of signal or feature stationarity in agreement with a perceptual grouping process. The main idea is to use the amount of attribute variation vector as a boundary detector. The segmentation can also

be performed based on muldimensional vectors, and the distance between consecutive frames can be computed through different measures [Tzanetakis 1999]. The attribute choice determines the level of abstraction that can be attributed to the resulting regions. As an example, if the attributes relate to human percept, as the energy in Bark bands relate to loudness, the regions obtained may infer some perceptual generalization about the signal regions, such as individual impacts in a recording of breaking object.

8.1.3 Audio Fingerprints

Audio fingerprints are compact content-based signatures that describe audio recordings. They can be computed from an audio chunk/stream and kept in a database. Fingerprint extraction deduces a set of features which requires discrimination, invariance to distortions, compactness and computational simplicity. Most fingerprint extraction systems consists of an initial stage and a fingerprint modeling block. The initial stage involves different efforts, mainly dimensionality reduction, perceptually meaningful parameters and temporal correlation. After audio digitization, redundancy reduction is usually performed through linear transforms, such as Singular Value Decomposition (SVD) or DFT for lower complexity transforms. The main goal is to decrease the dimensionality and, at the same time, to increase the invariance to distortions. More perceptual attributes are generally extracted by involving knowledge of the transduction stages of the human auditory system. In order to gain robustness against distortions and reduce the memory requirements, some systems condense the feature vector representation using additional transforms such as Principal Component Analysis (PCA) or perform a very low resolution quantization to the features. The fingerprint modeling block consists then in the reduction of the features sequence, usually suggested by statistics of frame values and redundancies in frame vicinity. Clustering of feature vectors is also an efficient method for compact representation. Thus, a much lower number of representative code vectors, the *codebook*, represent the sequence of feature vectors.

8.2 Content-Based Audio Transformations

This section presents the main principles for signal-based transformations. Our review is mainly derived from Cook's book [Cook 2002b] about signal digital processing.

As mentioned in Chapter 3, soundtracks for interactive animations are usually carried out via the playback of stored Pulse Code Modulation (PCM) waveforms. Most PC software systems for sound synthesis use prestored PCM properly manipulated to yield the final output sound(s). For physically based animations, prestored PCM are for instance, synchronized with the contacts reported from a rigid-body simulation. On the other hand, musical sound synthesis usually store just a loop,

or table, of the periodic component of a recorded sound waveform and trigger it repeatedly; this is referred to as *wavetable* synthesis.

The necessity and wish to exploit PCM generally appears for artistic reasons. In games and other interactive virtual environments, the available PCM rarely agree with the scene that has to be rendered. For instance, a sound of striking a metal plate available in the sound database may not exactly match any given virtual strike event on a metal plate. Thus, the temporal, spectral characteristics, etc., should be adjustable in order to create appropriate sounds in agreement with the virtual rendered scene. This section presents some of the current manipulations and methods.

8.2.1 Time Domain Approaches

One simple approach to manipulate sound is to alter the pitch by dynamic sample-rate conversion [McNally 1984]. However, shifting the sampling rate of a PCM sound causes the time of the sound to be modified too. This is similar to an increase or decrease in the velocity on a mechanical playback device such as a turntable or a tape machine. Speeding down with a time twice as long, gives a sound that has a pitch one octave lower. This transformation affects the quality of the perceived sound, referred to as *timbre*.

To independently control time and pitch of PCM, we must refer to *Psychoacoustics*. Psychoacoustics relates sound stimuli to information that is sensitive, musical, threatening, emotional, moving, etc., to our brains. An interesting aspect is that repeating, or roughly repeating, events at rates much lower than 30 Hz are perceived as time events. On the other hand, sounds at frequencies 35, 60, and 100 Hz, are identified increasingly as pitch, namely perceptual quantity related to the height of a tone. This perceptual effect can be referred to as the *30 Hz transition* and is the crucial feature as to how PCM samples have to be manipulated in time without changing pitch. There has been much work about time shifting, mostly in music computing. *Overlap-Add* (OLA) methods take advantage of the *30 Hz transition* by segmenting, repeating, overlapping and adding an existing waveform to produce a new waveform that has the same perceived pitch, but a modified time. However, overlap and add of waveforms could produce audible cancellations or reinforcements, in other words destructive and constructive interferences. The *Synchronous Overlap-Add* (SOLA) and *Pitch Synchronous Overlap-Add* (PSOLA) allow to reduce the modulations due to interference from overlap-addby using a cross-correlation approach to determine where to place the segment boundaries [Wayman 1989] (SOLA), and by moving the windows around dynamically based on the pitch of a region waveform [Moulines 1989] (PSOLA). Another artifact may appear from overlap-add time scaling due to certain parts of sounds that are more consistently enlarged or compressed in time, whereas others are not. By detecting the transient regions and just translating them into a new time position instead of timescale, better results can be achieved.

8.2.2 Signal Based Approaches

Signal models allow the input signal to be split into different components which can be parameterized and processed independently, providing flexibility for transformations. Typically, these components are sinusoids, transients and noise.

Spectral Modeling and Additive Synthesis

In sinusoidal modeling [McAulay 1986], the input signal is represented as a sum of sinusoids with time-varying amplitude, phase and frequency. Obviously, the major part of the sounds we perceive are not solely sinusoidal, even, for example, our voiced speech vowels. By Fourier analysis, we can determine the spectral features of a sound and the components that can be represented by sinusoids. With additive synthesis, sinusoids and other components are added to form a final wave with specific spectral properties.

Representing a signal with estimated sinusoids and a residual signal has been posed and implemented by Serra and Smith [Smith 1987, Serra 1997] in the *Spectral Modeling Synthesis* (SMS) system. We refer the reader to Appendix D for more details. This method allows for dividing, for example, a flute sound into the air flow and the harmonics components. In addition, many interesting modifications can be made to the signal on resynthesis. For instance, removing the harmonics from voiced speech, followed by resynthesizing with a scaled version of the noise residual, can result in the synthesis of whispered speech.

Vocoders and Decomposition in Subbands

Similar to the auditory system, a typical signal processing technique consists in splitting the sound into separate frequency bands, also referred to as filterbank decomposition. Due to the correlation with the source-filter model of the human vocal system, spectral subband *vocoders* (VOice CODERs) have shown their efficiency in coding and compressing speech. Vocoders decompose the spectrum into sections called subbands and analyze the information in each subband. The extracted parameters can be manipulated in various ways, producing transformations, such as pitch or time shifting, spectral shaping, cross synthesis, and other effects.

Substractive Synthesis and Linear Prediction

Subtractive synthesis consists in using a complex sound source, such as an impulse, a periodic train of impulses or a white noise, to excite resonant filters. Decomposing a sound into a source and a filter can be automatically achieved by linear prediction, or Linear Predictive Coding (LPC). This mathematical technique works particularly well for speech, due to the source-filter nature of the human vocal tract. However, LPC can also be used for analysis and modeling other types of sounds whenever an interesting spectral resonant structure appears. It is also useful when there is a significant time variation in the timbral content of a sound.

8.3 Implementation of Signal-Based Sound Models

If we consider sound rendering for a virtual environment using prerecorded events, we can imagine a content-based retrieval system as a search engine at the interface of organized database of prerecorded events. Typically, it first collects a request, defined by means of audio strategies, for instance, some information from the physics engine, or user's textual queries that describe some rendering attribute like event type, object material, etc., referring to audio descriptors. The system is in relation to a set of attributes extracted from the audio files in the database and it sends back a list of files or excerpts, ranked or not, that are all relevant to the demand. Alternatively, the system can refine user-feedback information in order to improve its execution in the future. The sound synthesis system then manage the audio files appropriately to render the final sound(s).

In this section, we present related work on methods for flexible playback of sound samples, that is, sound texture modeling. Then, some studies that particularly address sound authoring and controlling are introduced.

8.3.1 Sound Texture Modeling

Sound texture modeling allows for flexible playback of sound samples. Among the large number of sound texture generation methods, *concatenative synthesis*, sometimes referred to as *mosaicing*, [Roads 1991, Schwarz 2006] is closest in spirit to our method of example-based sound synthesis [Picard 2009b], described in the next Chapter (Chapter 9). Concatenative synthesis approaches aim at generating a meaningful macroscopic waveform structure from a large number of shorter waveforms. They typically use databases of sound snippets, or grains, to assemble a given target phrase.

As mentioned by Shwarz [Schwarz 2005], the first investigations for concatenative synthesis were made by the Groupe de Recherche Musicale (GRM) of Pierre Schaeffer, which introduced the use of recorded segments of sound to create their pieces of *Musique Concrète*. Schaeffer specifies the concept of sound object as a clearly delimited segment in a source recording, and since then, sound texture is a widely used concept in computer music. Later, Roads [Roads 1988] introduced *granular synthesis* as a elementary datadriven process. No analysis is performed on the audio units, the unit size is defined arbitrarily, and the choice is restricted to setting the position of the audio unit in the sound file. On the contrary, concatenative synthesis selects the audio units according to audio descriptors, such as those defined in Section 8.1. The probability to find the matching audio unit increases as the database becomes larger. Audio transformation can also be performed in order to completely match the selected units to the target specification.

In the work of Keller and Truax [Keller 1998], modeling of environmental-like sounds is addressed using sampled sound grains and meso-time control functions. The procedure is based on a database of several samples of everyday sounds gener-

ated by objects. The latter have been excited by physical mechanisms from which temporal patterns and spectral characteristics are derived. From the samples, grains that will further be used in the synthesis algorithm are extracted and the meso-scale temporal behavior of the simulation is defined. Finally, the sounds are synthesized and compared to the original samples. Different examples such as bouncing, water stream and texture scraping are given. In a quite similar approach, Cook states that many sound events, such as, for example, walking around on gravelor playing a game of dice, induce energy that feeds a particle system [Cook 2002b]. Based on the observation that most physics interactions are modeled by particles interacting with each other, Cook introduces the particles of sound production. His related work PhISEM [Cook 1999], namely Physically Informed Stochastic Event Modeling, is based on pseudo-random overlapping and adding of small grains of sound. The algorithms are derived from particle models characterized by basic Newtonian equations that control the motion and collisions of point masses.

Miller Puckette [Puckette 2004] explores the technique of controlling synthesis and proposes real-time audio streams as a source of timbral control over a synthetic audio stream produced by a computer. Additionally, in the work from Kobayashi [Kobayashi 2003] a sound is used as a target for resynthesis, starting from a pre-analyzed and pre-clustered sound base. Resynthesis is done FFT-frame-wise, conserving the association of consecutive frame clusters. This leads to high consistency in the development of the synthesized sound, even if spectral continuity is not necessarily preserved. On the other hand, Reck et al. [Reck 2005] propose a technique for granular synthesis that control the progression of the stream and the content of audio grains. They use Markov chains to control the evolution of the sound in time and fuzzy sets to define the internal structure of the sound grains. The major feature of the model leads in the coupling between the spectral components of the grains and the state transition probability through grain membership vectors.

For a more physically informed model, Dobashi et al. [Dobashi 2003] address aerodynamic sounds by first preprocessing sound textures based on computational fluid dynamics. During real-time process, sound textures are rendered according to the motion of objects or wind velocity creating plausible aerodynamic sounds. More recently, Zheng and James [Zheng 2009], propose physically based algorithms to synthesize fluid sounds for computer animation and virtual environments. Actual fluid solvers are extended with particle-based models, avoiding audio-rate time stepping. A time-varying linear superposition of bubble oscillators is applied to model the sound radiation from harmonic fluid variations and each oscillator is weighted by its bubble-to-ear acoustic transfer function. A large variety of fluid sound examples are presented, generally with thousands of acoustic bubbles. However, the numerous bubble sound sources cannot currently be processed in real-time.

8.3.2 Authoring and Interactive Control

Vocem introduced by Lopez et al. [Lopez 1998], is one of the first graphical interfaces for real-time granular synthesis, with high-quality audio output and very short latencies. Parameters allow the user to control the creation and the distribution of the grains with simplicity and precision. With *MoSevius*, Lazier et al. [Lazier 2003] first attempt to apply unit selection to real-time performance-oriented synthesis with direct intuitive control. A unit is played when its descriptor values lie within ranges controlled by the user, and the features used range from energy, spectral flux or spectral centroid, to voicing and instrument class. For a more musical context, Misra et al. [Misra 2006] focus on a single framework that starts with recordings and proposes a flexible *workbench* for sonic sculpting in general. The synthesis phase is controlled by real-time manipulation of parameters. Finally, for a virtual reality-related application, Cook [Cook 2002a] introduces automatic analysis and parametric synthesis of walking and other (gesture-rate) periodically modulated noisy sounds. The method consists in evaluating recordings of walking by extracting characteristic features as tempo, basic resonances and control envelopes. However, the approach remains limited in the classes of sounds it can handle.

Chapter 9

Retargetting Example Sounds to Interactive Physics-Driven Animations

Contents

We present a new method to generate audio in the context of interactive animations driven by a physics engine. Our approach extends current capabilities of audio sample playback techniques by retargetting audio grains extracted from recordings according to the output of a physics engine. This work was published at *the AES 35th International Conference - Audio for Games*, in London, United Kingdom [Picard 2009b](JAES, 57(6), June 2009). This Chapter first introduces the reasons for our approach. We then decribe the automatic analysis of recordings from which audio grains are extracted. Finally, we present our flexible synthesis of soundtracks in the context of physics-driven animations.

9.1 Introduction

Audio rendering of complex contact interactions between objects animated with physics engines is becoming a key challenge for interactive simulation and gaming applications. Contact sounds should not appear repetitive, as it is often the case when a small number of pre-recorded samples is directly used. Increasing the number of samples is not always possible due to memory constraints. Furthermore, matching sampled sounds to interactive animation is difficult and often leads to discrepancies between the simulated visuals and their soundtrack. Alternatively, contact sounds can be automatically generated using sound synthesis approaches. However, interactive physically based synthesis currently targets limited classes of sounds and, to date, cannot render convincing breaking, tearing or non-rigid interaction sounds.

Our approach aims at bridging the gap between direct playback of sound samples and physically based synthesis by automatically analyzing audio recordings so that they can be retargetted to interactive animations. It combines an off-line analysis process with an interactive on-line resynthesis as illustrated in Figure 9.1. In the off-line process, we automatically segment audio recordings into atomic time-slices or *grains* and build a shared, compact dictionary. Based on a correlation estimation, we represent each original recording as a series of grains. During interactive animations, the audio grains are triggered individually or in sequence according to parameters reported from the physics engine and/or user-defined procedures.

Our specific contributions are:

- a method for automatic analysis of audio recordings, and the generation of compact dictionaries of audio grains and correlation patterns.
- a specific technique for analysis of recordings of continuous contact events such as rolling or sliding, that separately encodes the steady-state and transient parts.
- a solution for generating on-line audio for interactive animations by retargetting the audio grains in order to match inter-object contacts and interaction state.

The proposed framework allows the generation of compelling and controllable soundtracks in the context of physics-driven animations. Key aspects of our approach are its low memory usage and the different levels of authoring available to both audio programmers and designers.

9.2 A Generic Analysis of Pre-Recordings

In this section, we detail the components of our automatic analysis, performed off-line. The analysis process consists in segmenting the available audio recordings into audio grains and generating corresponding re-synthesis information.

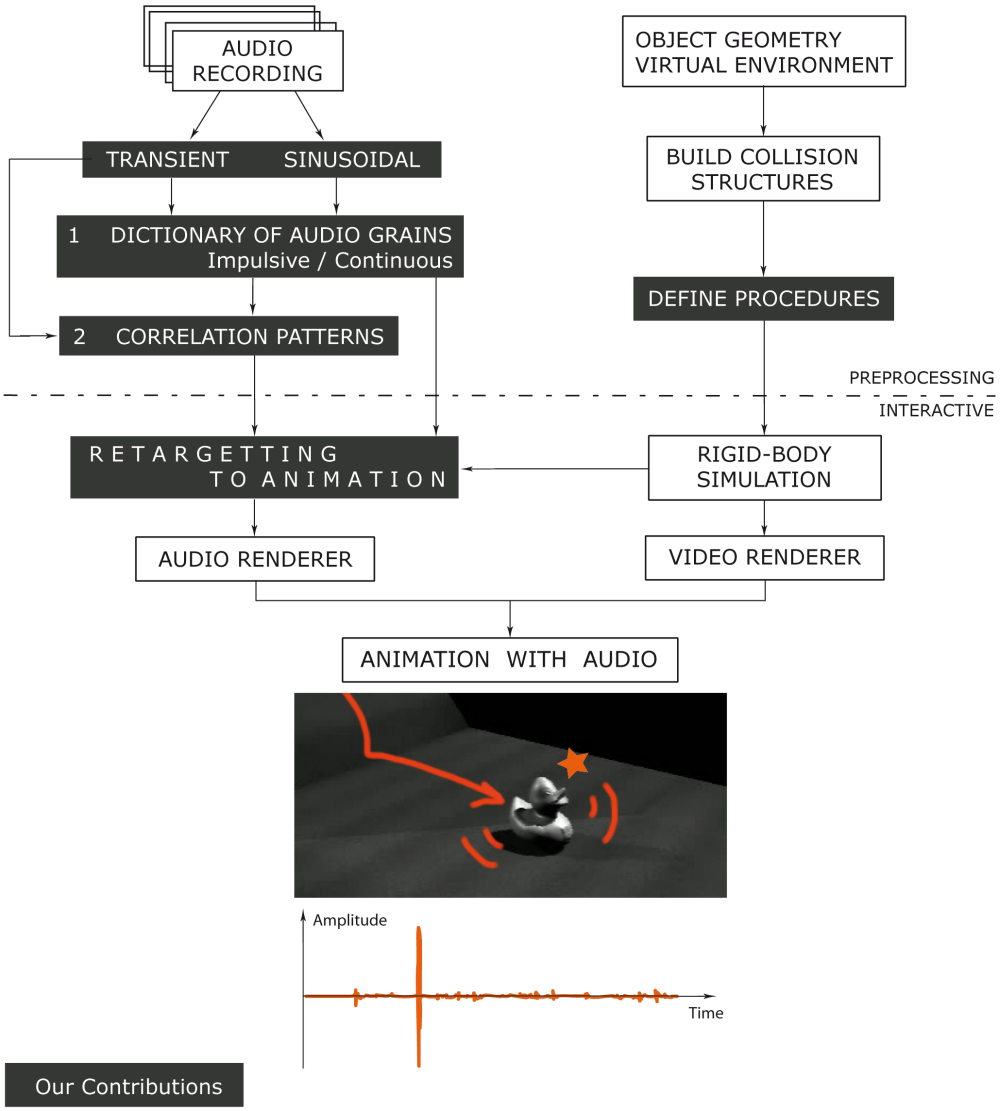

Figure 9.1: *Overview of the approach combining off-line analysis of recorded sounds with interactive retargetting to motion. Our contributions are underlined in dark gray boxes.*

9.2.1 Impulsive and Continuous Contacts

We distinguished between impulsive and continuous contacts. Audio recordings of impulsive contacts, such as impact, hitting or breaking sounds, are characterized by the preponderance of transients. Conversely, recordings of continuous contacts, such as rolling or sliding, contain a significant steady-state part. This steady-state part is characteristic of the objects being continuously excited during contact and should be preserved during resynthesis. Continuous contact sounds are thus segmented into a steady-state and a noise/transient part. Segmentation is carried out using

the *Spectral Modeling Synthesis* (SMS) approach introduced by Serra [Serra 1997]. SMS is based on modeling sounds as stable sinusoids (sinusoidal part) plus noise (residual part). The analysis procedure extracts the steady-state part by tracking the time-varying spectral characteristics of the sound and represents them with time-varying sinusoids. These sinusoids are then subtracted from the original sound. The residual part mainly contains energy at high-frequencies and may represent the micro-transient character of the event. Thus, we will refer to the residual part as the *transient part*. We refer the reader to the Appendix D for more details about the method. As a result, recordings of continuous-contacts are first decomposed into two recordings, prior to grain extraction while recordings of impulsive contact are directly segmented.

9.2.2 Automatic Extraction of Audio Grains

In the off-line analysis, audio recordings are fragmented into syllable-like audio grains. Audio grains are elementary signal components (typically 300 to 3000 samples long, i.e. 0.01 to 0.1 s @ 44.1KHz) and, for instance, may correspond to discrete impacts in the more complex recording of a breaking glass.

Audio recordings are by nature non-stationary. Thus, their amplitude and frequency are time-varying and the variations are characteristic of the signal. In signal processing, the *spectral flux* is a measure of how fast the power spectrum of a signal is changing and can be calculated as the Euclidean distance between the two normalized spectra of consecutive frames. The spectral flux is typically used for onset detection [Dixon 2006], and we conjecture that its variations are appropriate to extract the audio grains. In our implementation, the spectral flux is calculated as in [Dubnov 2006] and is applied to all the input recordings. For recordings of continuous contacts, we calculate spectral flux on both sinusoidal and transient parts. The recordings are then segmented into grains at the inflection points of the spectral flux, as shown in Figure 9.2.

Extracted audio grains are windowed with a Tukey window, i.e. a cosine-tapered window, with r=0.02 for the ratio of taper and normalized for power conservation. To limit the number of extracted grains, we discard the grains with low energy according to a user-defined threshold. The remaining audio grains are labeled either continuous or transient depending on their source recording and stored away.

As a result, we obtain a dictionary of grains that can be retargetted to parameters of the physics engine and/or user-defined procedures. Note that the dictionary is non-redundant since the segmentation occurs exactly at the inflection points of the spectral flux. Audio grain extraction can be carried out on an entire database of audio recordings, leading to a large dictionary that offers more possibilities than the original recordings alone.

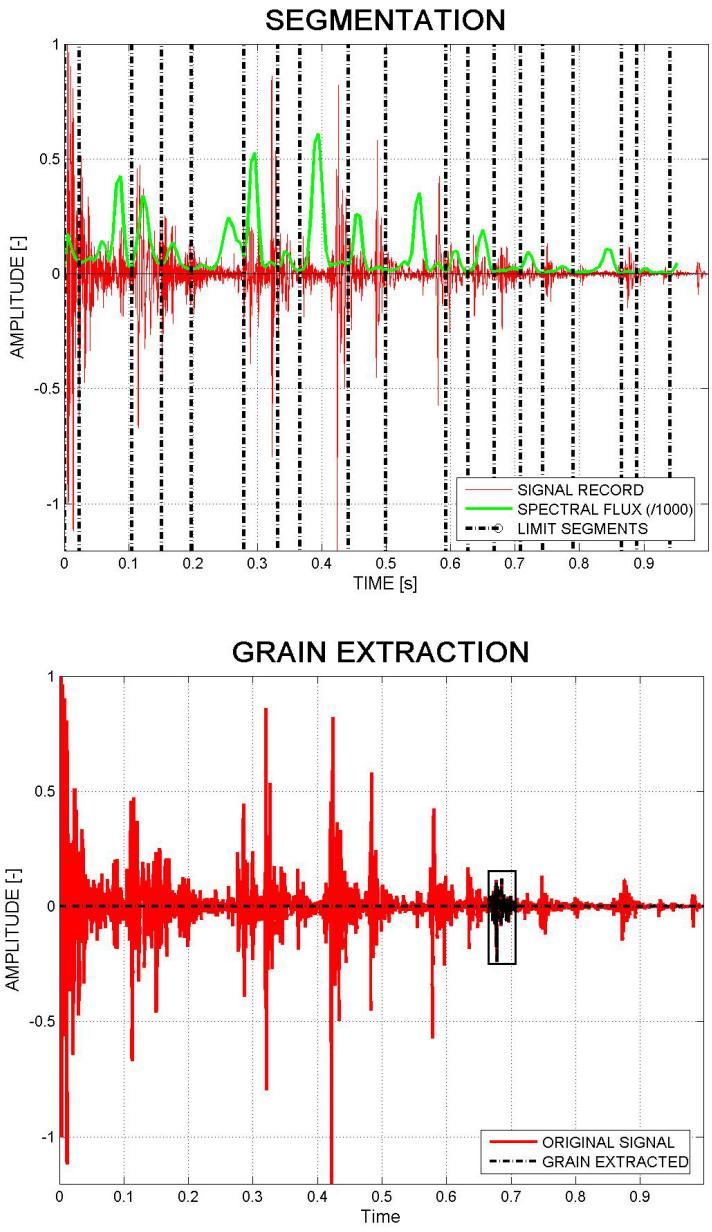

Figure 9.2: *Segmentation of an impulse-like sound event according to a spectral flux estimation (top) and example of an extracted audio grain (bottom).*

9.2.3 Generation of Correlation Patterns

In the second step of the analysis process, original audio recordings are decomposed onto the "'basis"' formed by the previously extracted audio grains to obtain correlation patterns. The correlation patterns will be used to encode each original recording as a series of audio grains.

Our method is similar in spirit to matching pursuit (MP) ap-

proaches [Mallat 1993] that decompose any signal into a sparse linear expansion of waveforms selected from a redundant dictionary of functions [Kling 2004] by performing iterative local optimization. MP approaches are traditionally considered too slow for high-dimensional signals, with impractical runtimes. In comparison with traditional MP approaches, we introduce a greedy grain selection approach avoiding the iterative process.

The cross-correlation between each available recording and each grain is calculated. Normalized vectors are used for the calculation. We greedily identify major peaks of the cross-correlation function using a peak picking step. A user-defined number of peaks is then preserved for each recording (for instance the k largest). A correlation pattern is stored as the list of retained cross-correlation peaks across all grains, with the corresponding time indices: they represent the appropriateness of each grain to synthesize the considered original source. Cross-correlation calculation is performed for all recordings, including the sinusoidal and transient part of continuous contact sounds.

A direct benefit of our analysis step is a compact representation of the original sound database since the dictionary of grains and the correlation patterns typically have a smaller memory footprint than the original assets.

9.3 Flexible Sound Synthesis

Once the analysis has been performed, we can resynthesize an infinite variety of sounds similar but not identical to the source recordings. We introduce a flexible *audio shading* approach [Takala 1992] allowing us to render contact sounds for animation based on collision reporting from the real-time physics engine and/or user-defined procedures.

9.3.1 Resynthesis of the Original Recordings

The correlation patterns obtained in the off-line analysis process can be directly used to reconstruct the original recordings at run-time. The resynthesis is performed by choosing from the correlation pattern the grains presenting the maximum correlation amplitude (see Figure 9.3). Similar to concatenative synthesis, our approach considers candidate grains one at a time, in a time-interval starting from the end of the previously added grain. In order to take into account the effect of windowing (see Section 9.2.2), the look-up starts slightly before the end of the previously added grain. In our case, this offset is consistent with the overlap of the Tukey window and equals (r**len*/2) where *len* is the length of the previously added grain and r is the taper of the window (here, r=0.02). The candidate grain is searched within an interval of duration equal to half of the median length of the available grains. Recall from Section 9.2.3 that several correlation peaks might have been stored within that time interval. To reconstruct a signal closest to the original recording, the grain with maximum correlation value is chosen. We then concatenate the grain

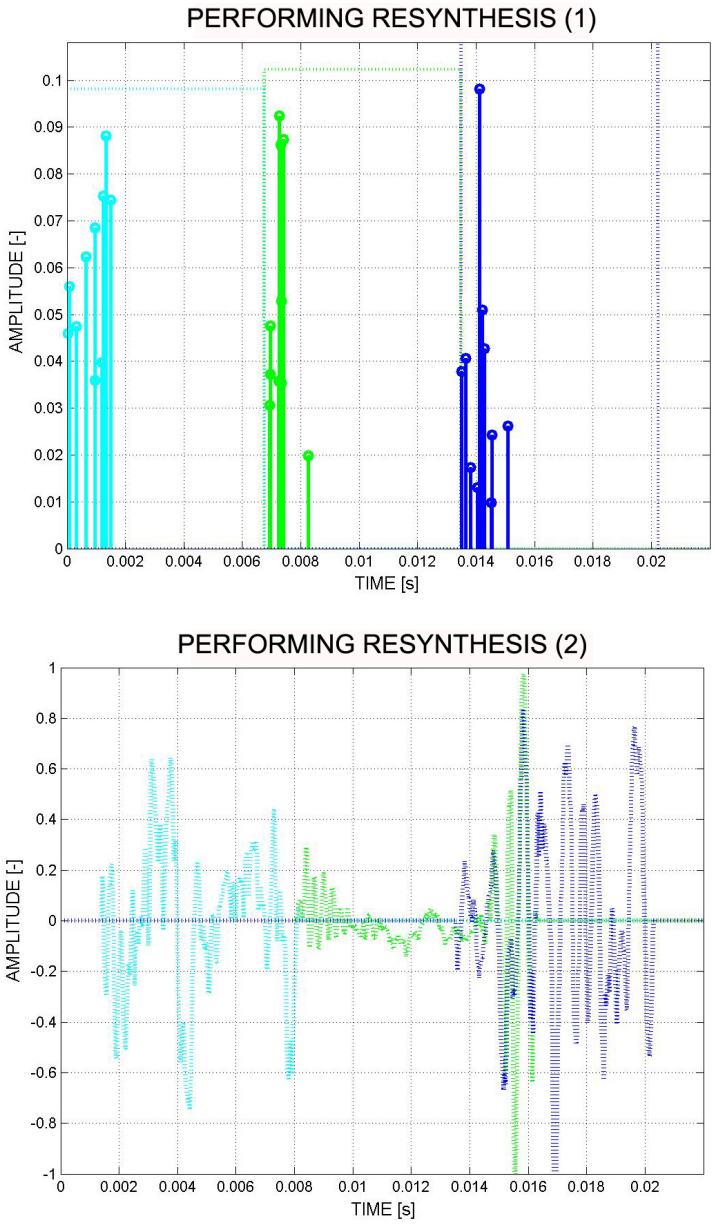

Figure 9.3: *Concatenating grains for resynthesis. Top: example correlation peaks for the considered recording. Bottom: Reconstructed recording. In this case, only the grains presenting the maximum correlation amplitude with the considered original recording are chosen.*

using overlap-add blending at the specific correlation time index, with an amplitude equal to the power of the original audio recording in the considered time-window normalized by the power of the grain.

9.3.2 Physics Parameters for Retargetting

For a compelling re-synthesis, it is necessary to match the audio content to the contact parameters reported by the physics engine. In particular, we need to detect and distinguish impulsive and continuous contact, including rolling and sliding. While existing physics engines, such as *Havok* or *Nvidia's PhysX*, do report contacts, they generally fail to distinguish between rolling and sliding, making it challenging to trigger consistent sound effects.

Contacts between bodies have been extensively investigated for sound rendering [Avanzini 2002a, Pai 2001]. Sliding involves multiple micro-collisions at the contact area. For rolling, the surfaces have no relative speed at the contact point and this difference in contact interaction leads to an audible change. Based on previous studies on contact simulations, we analyze penetration forces and relative velocities across time in order to detect impulsive and continuous contacts, as seen in Figure 9.4.

When a contact is reported, grains are appropriately retargetted according to the labeling of the grain obtained during the analysis (see Section 9.2). For impacts, impulsive grains are retargetted to penetration force peaks, knowing that relative velocity peaks occur at the same time instant as penetration force peaks. Continuous contacts are detected when the penetration force and the relative velocity are constant. Rolling is identified when the relative velocity is equal or close to zero. Audio grains can be chosen either randomly or retargetted from a consistent correlation pattern, for instance a pattern extracted from a rolling sound if rolling was detected. They are concatenated with an amplitude proportional to the power of the contact interaction, i.e. the dot product of the penetration force and the relative velocity. For continuous contacts, spectral domain modifications can also be easily achieved, thanks to the SMS approach (see Section 9.2.1). For instance, the sound can be adapted to the velocity of the objects in interaction. In addition, the SMS approach enables the modification of the continuous and transient parts separately. In [Stoimenov 2007] the effect of surface roughness on the frequency of non-squealing frictional sound generated in dry flat/flat sliding contact was studied. It was found that rubbing frequency or load does not qualitatively change the spectrum of the sound but high rubbing speed generates higher levels of power spectral density and high load gives much broader peaks. Thus, transient grains are played back without any modification in frequency content. In contrast, continuous grains are frequency-scaled according to the velocity of the interaction, shifting toward higher frequencies as the velocity increases. Examples of modified recordings of continuous contact events are provided as additional material[1].

[1] Additional material: http://evasion.inrialpes.fr/Membres/Cecile.Picard/SupplementalAES/

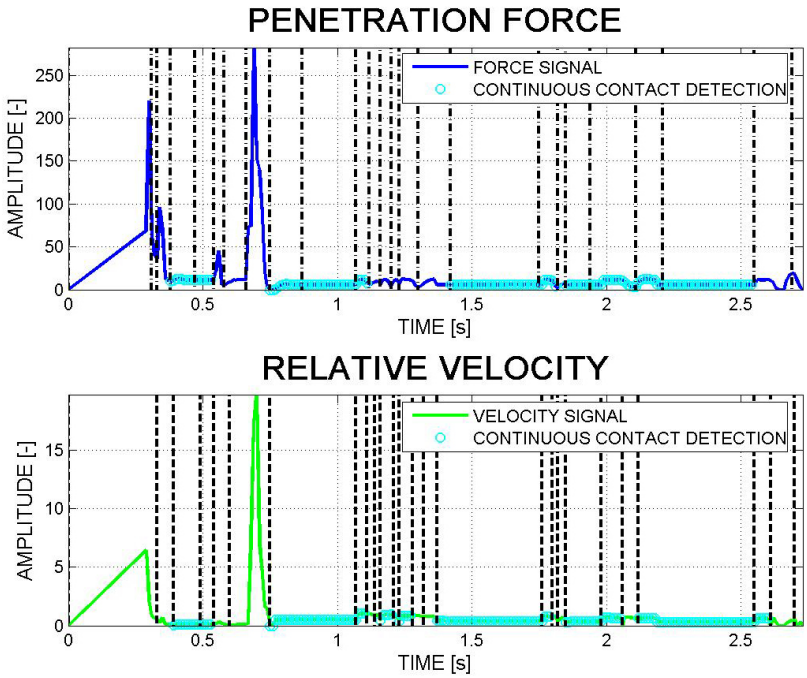

Figure 9.4: *Detection of continuous contact based on the evaluation of the penetration force and the relative velocity reported by the physics engine.*

9.3.3 Flexible Audio Shading Approach

Our approach also supports additional, user-defined, resynthesis schemes. For instance, time-scaling of the original audio recordings can be easily implemented using the previously calculated correlation patterns and an approach similar to Section 9.3.1. First, a time-scale modification factor α is set: $\alpha < 1$ to reduce the length of the original recording and $\alpha > 1$ to increase it. The length of the time-scaled recording is α times the length of the original audio recording. The correlation patterns that code the original recordings as series of audio grains (see Section 9.3.1) are used to construct the modified recording. Series of grains are displaced in time by a factor of α. Note that, in the case of time-stretching, this approach does not only shift original grains in time but also synthesizes additional material to fill in the gaps introduced by the stretching process.

The method for concatenating grains is flexible and allows the synthesis of an infinity similar audio events. Similar audio events can be built using a variety of approaches. For instance, variable audio content can be matched to the characteristic rhythmic pattern of a source recording using its correlation pattern. Time-scaled audio recordings and synthesis of similar audio events are provided for listening as additional material[2].

[2]Additional material: http://evasion.inrialpes.fr/Membres/Cecile.Picard/SupplementalAES/

Considering that the audio grains are enough small, and also that a windowing is performed after extraction, we can assume that grains do not contain medium or listener information such as reverberation or Doppler effects. Thus, those effects can be easily added, as an additional procedure in the audio shader.

9.4 Results

We tested our approach using a complete database of recorded contact interaction events used in a commercial video game. The database comprises various material types, e.g., plastic, metal, glass, and interaction types, such as hitting, breaking or rolling. It consists in 855 audio files (44.1 KHz, 16 bit PCM data), for a total recording time of 18 minutes ($\approx$ 48 million samples) and a memory footprint of 95 Mb. We assumed that those recordings are free of outside noise and unwanted echoes in order to extract audio grains that can be retargetted to a wide range of animations. The analysis of the recordings required to extract the grains and generate the correlation patterns was performed in an off-line process using a *MATLAB* implementation (see Section 9.2).

9.4.1 Dictionary-based compression

A first application of our approach is simply to reduce the memory footprint of the original assets. We applied our approach to a single group of recordings, the stone interaction events with 94 recordings in total, representing 14,6Mb of data. We extracted 5171 audio grains from which 3646 were impulsive and 1525 were continuous, and computed the correlation patterns for all input recordings. The extraction of the grains for the 94 recordings took 11 minutes while the correlation patterns were computed in 4 hours. In this case, we only coded into the correlation patterns the grains with maximum correlation amplitude. A single grain was encoded for each time-interval multiple of the average grain length. As an example, if we consider an original recording of 3 sec. and grains with an average length of 3000 samples (0.1 s @ 44.1KHz), the correlation pattern consists in about 45 peaks, taking into account the overlap of the grains. The memory footprint was 11,64Mb for the audio grains and 29Kb for the correlation patterns, leading to a 20% gain over the original assets in this case. The gain in memory footprint is not high but still significant and the approach allows us to create more content than the original database. In addition, to more efficiently manage the memory usage, only a subset of the grain dictionary can be made available, such as the grains showing the highest correlation with the type of recordings we consider. A subsequent lossy audio compression step on the obtained dictionaries of grains would lead to an additional 5:1 (e.g., 128 kbps mp3) to 10:1 (e.g., mp3 VBR) compression ratio.

We provide as additional material a number of comparisons between original audio recordings and reconstructed versions in order to evaluate the quality of the

approach[3].

9.4.2 Interactive physics-driven animations

Retargetting of extracted audio grains was used to generate interactive audio in the context of simulations driven by the *Sofa*[4] physics engine. We used the precomputed correlation patterns generated in the analysis step. The penetration force and the relative velocity of the objects in interaction were studied across time in order to detect impacts and continuous contacts. The grains were chosen from an appropriate correlation pattern and triggered, in a random manner or in series following the approach described in Section 9.3.2.

Physics engines typically report contact information at a framerate of 30 to 60Hz, whereas sound has to be synthesized at 44kHz. In the case where the grains are of minimum length, i.e. 300 samples, the physics framerate is not sufficient to guarantee a continuous sound rendering. A benefit of our approach is that we can continuously select and play-back grains from a given correlation pattern to ensure a continuous result, regardless of the frame-rate of the physics simulation.

Example movie files demonstrating our approach are provided as additional material[5]. The result shows that our approach can be effectively used for plausible soundtracks consistent with visual rendering, although more work is required to perfectly interpret the parameters delivered by the physics engine.

9.5 Extensions

The actual implementation in *MATLAB* implies an extensive preprocessing time. The analysis algorithm is currently being implemented in C++. This will allow, in addition, a more straightforward integration in the audio pipeline of current video games.

The method has been essentially applied on recording events, which are often short and can be easily related to a physical quantity. We believe that the technique can be extended to speech or ambiance recordings. However, more research is needed. First, the analysis step has to be further developed to handle the audio sample classes and to appropriately extract the corresponding audio grains. Second, information from the physics engine has to be better studied in order to identify the parameters that would trigger the audio grains.

In order to offer a more compact representation of the data, further reduction of the audio grains and the correlation patterns can be investigated. Clustering of feature vectors can be an efficient method for compact representation of audio grains. This has been explored by the use of MFCCs. However, further work is needed to precisely set the parameters. LPC might also be a good candidate to

[3] Additional material: http://evasion.inrialpes.fr/Membres/Cecile.Picard/SupplementalAES/
[4] Simulation Open Framework Architecture: http://www.sofa-framework.org/
[5] Additional material: http://www-sop.inria.fr/reves/Cecile.Picard

model audio grains. Reduction of the correlation patterns can be suggested by statistics of the values and redundancies detection. Additional transforms such as Principal Component Analysis (PCA) could be applied to condense the feature vector representation. In addition, statistical analysis could provide us with typical event schemes.

9.6 Conclusion

We have proposed a technique to generate in real-time plausible soundtracks, synchronized with the animation, in the context of simulations driven by a physics engine. Our approach dynamically retargets audio grains to the events reported by the engine. We proposed an off-line solution to automatically extract grains and correlation patterns from a database of recordings.

Our method can be applied to reduce the footprint of the original assets and can further be combined with traditional compression techniques while maintaining acceptable audio quality. In addition, the approach allows trading memory footprint for variety at synthesis time and frees the sound designer from the hassle of having to create a variety of similar sounds by hand. By combining audio content and information that describes how that content is to be used, our approach might be a good candidate for the Interactive XMF Audio File Format (eXtensible Music Format)[6].

The main limitation of our technique is the preprocessing time required to calculate the correlation between the original recordings and the grains. However, this could be improved by implementing this step in C++.

We believe our work addresses important issues for both audio programmers and designers by offering extended sound synthesis capabilities in the framework of standard sample-based audio generation. In the future, higher level statistical analysis of the patterns obtained for similar recordings could lead to more efficient and controllable resynthesis. Finally, clustering of similar grains could be investigated to further reduce the size of the data structures and the pre-processing time.

[6]Interactive XMF Audio File Format (eXtensible Music Format):
http://www.iasig.org/wg/ixwg/

In the previous part, we introduced example-based techniques and we proposed an approach for sound rendering of object interactions in the context of physics-driven animations. Our approach extends current capabilities of audio sample playback by retargetting audio grains extracted from recordings according to the output of a physics engine. In particular, we proposed automatic analysis of audio recordings, and the generation of compact dictionaries of audio grains and correlation patterns. A flexible sound synthesis solution was then introduced to generate online audio according to parameters reported from the physics engine. However, these parameters are sometimes difficult to interpret, especially when the physical mechanism that leads to sound is complex. In the last part, we will propose a prospective hybrid model that accordingly combines a physical model and an example-based model for sound rendering of specific physical interactions.

Part IV

Perspectives on a Hybrid Model for Sound Synthesis

Chapter 10

Motivation for a Hybrid Model

Contents

Virtual environnements increasingly rely on physics simulations. Physically based approaches for sound rendering sometimes lack accuracy due to simplified models. On the other hand, example-based techniques rely on the physics engine report and its interpretation. Sound for complex physical mechanisms may consequently be difficult to parametrize. In this Chapter, we discuss the possible strategies to combine a physically based approach with an empirical approach, thus building a hybrid model. We mainly focus on a hybrid model that can extend a linear system to simulate specific sound properties, leading to more realism in the rendering. This Chapter starts with typical limitations of the physical-only model and the empirical-only model which prevents them from being used for specific audio rendering tasks. We then present previous work addressing sound modeling of nonlinear object vibrations, and previous work related to sound simulation with mixed techniques. Finally, we propose our approach for hybrid model, the embodied techniques and its parametrization, and we introduce our application case, the fracture event, which will be further developed in Chapter 11.

10.1 Problems of Single Models

10.1.1 Modeling Nonlinearity With A Simple Vibration Model

As an illustration of modal analysis for one dimension, the extremely simple mechanical system consisting of a mass attached to a spring, the *mass/spring/damper* system, can still illustrate physical nonlinearity [Cook 2002b]. It is usually accepted

that the force applied by the spring is linearly related to the displacement. However, this simplification is valid only for ideal springs or small displacements of real springs. Actual springs exert more force for larger displacements. Similarly, nonlinearity can be observed for a plucked string. In the case of large displacements, the tension is amplified twice each cycle. As a consequence, extra even harmonics could arise in the spectrum. These nonlinearities are of continuous nature, and are continuous functions of displacement. A more extreme type of nonlinearity, discontinuous nonlinearity, has, in general, a larger effect on the system and spectrum. This is the case, for example, when considering transient or sustained sounds that result from deterministic coupling interactions between exciting and resonating objects, or successive excitations of the same set of resonators.

A nonlinear system is a system which does not satisfy the superposition principle, or whose output is not proportional to its input. Nonlinearities in the simulation imply critical changes for the natural frequencies extracted during analysis. Modal superposition is consequently no longer valid. This is the case, for instance, when the energy suddenly increases such as during shocks. In these cases, direct integration of the dynamic equation of equilibrium is required, which cannot be processed in real-time.

10.1.2 The Embedded Signature of an Example-Based Sound

If playback of sampled sound files, possibly processed with amplitude, pitch or filter-envelopes, is unsuitable to render the large variety of sounds in typical virtual scenes, sound texture modeling tends to adapt audio samples to the highly dynamic content. In the approach described in Chapter 9, the flexible sound synthesis is depending on the physics engine report. This implies non ambiguous interpretation of the parameters.

Parametrization is important for designing effective mappings between user gestures and sound control parameters. However, the problem of parameterizing audio grains and correlations patterns might be similar in essence to a sort of *reverse engineering* problem. Indeed, the purpose would be to deduce how audio is produced starting from the resulting sounds themselves. This would be especially problematic for complex physical mechanisms where the variables are not completely defined or can not easily be obtained by the physics engine.

10.2 Previous Work

Modeling Nonlinearities

O'Brien et al. address for the first time in graphics sound rendering from nonlinear object vibrations [O'Brien 2001]. This is performed by explicitly integrating a nonlinear FEM with small time-step sizes and by simulating the radiated sound field with time-domain ray-based Rayleigh method. However, the method prevents

real-time simulation due to its large computational cost. In addition, the radiation model presents limited accuracy. Cook [Cook 2002b] tackles the problem of nonlinear systems and suggests that linear systems can be adopted with careful additional modeling blocks that target nonlinear phenomenon. In this way, a linear system is extended to simulate nonlinearities. As an example, a model of a waveguide plucked string can be improved with an additional algorithm that represents the traveling waves reflected back into the junctions when the displacement on the string exceeds a given threshold. These nonlinearities can also be modeled with careful lookup tables (see Section 8.2), also referred to as waveshaping synthesis. This consists in accessing a wavetable (see Section 8.2) with another, usually simple, waveform.

Bilbao [Bilbao 2008] focus on constructing a numerical method that offers reduced stability analysis, leading to simplified computer implementation. He proposes to model sounds from nonlinear plates with energy conserving finite difference discretizations and time-stepping schemes. More recently, Chadwick et al. [Chadwick 2009] present an efficient model for nonlinear thin shells vibrations. They extend linear modal analysis with nonlinear thin shell forces that represent the modes coupling. A reduced-order dynamics model allows for audiorate time-stepping of mode amplitudes. Their nonlinear modal model is guaranteed to perform in a valid energetic range, preventing from *mode locking*. The method allows for more convincing *crashing* and *rumbling* sounds than when using linear modeling. However, modeling any given nonlinear vibration-based sound remains the question. Models that target nonlinearities in real-time should be developed.

Hybrid Approaches

Hybrid approaches have been studied in sound rendering in order to maximize strengths and minimize weaknesses of each technique. As an example, Castle et al. [Castle 2002] develop a *TapAndScratch node* that automatically synthesizes the sounds of tapping and scratching interactions with haptic objects in the *Reachin* haptic API. Scratching sounds are generated by granular synthesis. The method supplies adequate information about the surface of an object/variations in force, rate of scratching, duration of forces, as well as properties of the surface such as stiffness, friction and bumpiness. On the other hand, tapping sounds are generated through modal synthesis that relates more closely to the material and shape of the object. The authors observe, however, that both techniques synthesize sound separately at the same time, and the progression from hitting to scratching lacks coherence due to the change in timbre from one algorithm to the other. They finally suggest, for future work, new hybrid algorithms that merge the surface and shape properties in a single coherent sound, allowing smooth changes from impacts to scrapes, rather than mixing.

Some interesting approaches propose a hybrid model where one model part is integrated according to the dimension of the other one. Keller and Truax introduce a granular synthesis technique that is related to physical modeling and tra-

ditional granular synthesis [Keller 1998]. Environmental-like sounds are produced using sampled sound grains and meso-time control functions. They focus on the grain distributions for ecologically meaningful sound events. The technique aims at filling some of the gaps along the continuum between stable resonant modes and completely stochastic clouds. With PhISEM, Cook [Cook 1999] introduces physical modeling of granular synthesis. In the approach, basic Newtonian equations govern particle models behavior, creating pseudo-random arrangements of small grains of sound.

10.3 Our Approach for a Hybrid Model

A physically based sound model is still beneficial to generate a coherent sound feedback with user interactions because mapping control parameters to a physics engine is simple. However, extra design considerations must be taken into account in sound modeling, in order to appropriately simulate nonlinearity and other complex physical phenomena that may alter the sound, and to ensure coherence in sound rendering of user interactions.

10.3.1 Selection Criteria

We suggest that our hybrid model is applied only when nonlinearity occurs. By checking noncritical parameters, the physically based model is guaranteed to be valid for rendering the sound. If the physical model is no longer accurate, our alternative hybrid model takes over. Noncritical parameters are intimately related to the phenomenon being observed. In the case of the plucked string for example, the parameter would be the displacement on the string.

10.3.2 Techniques and Mutual Connection

The presence of nonlinearity widens the frequency content of even the most simple vibration systems. Starting from this observation, Frequency Modulation (FM) synthesis appears adequate to model the wider frequency range. FM synthesis relies on modulating the frequency of a sine wave of average f_c, the carrier frequency, by another sine wave f_m, the modulator:

$$y(t) = sin(2\pi t f_c + \Delta f_c cos(2\pi t f_m)) \tag{10.1}$$

Bessel functions provide us with a mathematical formulation of FM synthesis and allow to represent the harmonic distribution of the obtained sound.

We propose a hybrid model that gathers Frequency Modulation (FM) synthesis with additional audio grains that aim to supply more expressiveness to the model. We suggest that audio grains be extracted from recordings of similar sound events. Triggering of audio grains could be managed through dynamic parameters of the physics engine or user-defined procedures, similar to the approach in Chapter 9.

Parametrization

A basic problem when using a hybrid approach is to understand how different discrete-time modeling paradigms, are complementary and how they can be associated in a clever way. Audio grains aim at bringing more texture to the final sound(s) and are surimposed according to specific parameters reported from the physics engine and/or user-defined procedures. To ensure coherence inside the model itself, spectral attributes of FM synthesis and audio grains should be consistent. For this purpose, we suggest to choose audio grains that are more closely related to FM synthesis in terms of spectral attributes.

Adequate smoothing in sound should be provided when nonlinearity occurs, that is when switching from vibration model to hybrid model, from modal synthesis to FM synthesis with additional audio grains. Physical models usually fail to fit real data. Most of the time, the involved parameters are not related to the final sound in an intuitive way. This is the main problem when considering smooth transition between a vibrational model and a model that includes audio grains extracted from recordings. To avoid audible artifacts during transition, we propose to tune the parameters of the FM synthesis according to the resonant parameters of the vibration model.

By, on one hand, directing the connection between the vibrational model and the FM synthesis of the hybrid model, and on the other hand, addressing the coherence inside the hybrid model, we assume that audible artifacts would be avoided.

Very Local Nonlinearities

Previous work has shown that when an object is struck, the resulting sound shows a very local nonlinearity in its early part [van den Doel 1996]. This is similar to shock events. We assume that our algorithm may be able to predict such an event, based on physical parameters involved in the phenomenon. Because the nonlinear phenomenon is restricted to a very short period of time, our hybrid model is not very pertinent. To appropriately render the sound of shocks or impacts, we propose to use granular synthesis for the transient part then mix in modal synthesis for the longer ringing part. This may also overcome the perceptible lag between granular synthesis and modal synthesis [Castle 2002], and the synthesis integration would be thus along the dimension of sound itself.

10.3.3 A Hybrid Model for Fracture Event

We presently introduce our application case for hybrid model. Fracture events are widely used in virtual environments, especially in games. Our purpose is to propose a model that is simple enough to be incorporated into interactive game pipelines, and that offers a larger variety for sound than was proposed until now via the playback of stored PCM waveforms. Fracture has specific mechanical properties that would make a hybrid approach adequate to render the corresponding sound(s):

1. Fracture can have very different forms according to the material involved. Indeed, ceramic produce brittles (brittle fracture) whereas metal is subjected to deformation (ductile fracture) and wood is much similar to cloth due its fiber-like structure. We intend to provide a general model for fracture in terms of structure design, general enough to capture the main attributes of this broad family of sounds.

2. Fracture cannot be modeled by an event per se. Indeed, well-defined steps/phases emerge, namely crack initiation, crack propagation, and creation of fragments. These phases are fundamentally different due to the mechanical process that takes place from a more microscopic point of view. Consequently, control parameters may vary substantially.

3. Micro-events of fracture could be simply modeled by micro-impacts, or more precisely, by micro-shocks due to a large release of energy over a short period of time. In this case, the frequency content of the response suddenly increases leading to inaccuracy when using modal superposition.

In Chapter 11, we develop the details of our proposed hybrid model for fracture events.

Chapter 11

A Hybrid Model for Fracture Events

Contents

Fracture events are a common experience in virtual environments and especially in video games. Most of the time, sound rendering is performed through prerecorded sounds simply triggered according to specific dynamic information. Due to memory constraints for sound databases, the variety of these sound events is mostly limited to the use of amplitude-, pitch- and filter-envelopes. This commonly induces very repetitive sound rendering. In this Chapter, we propose a hybrid model for realistic fracture sound events, combining a physically based approach and an example-based approach. Our main goal is variety and flexibility of sound rendering for interactive applications. We first present basic mechanical principles of fracture events. We then review previous work in fracture modeling, mainly concerned with visual rendering. We finally introduce our hybrid model and present prospects for its implementation.

11.1 Basic Fracture Mechanics Concepts

In this section, we review the main concepts for fracture mechanics, with a specific focus on properties that may affect the emission of sounds. This summary is mainly based on lecture notes of Zehnder [Zehnder 2009].

We consider a sheet with an initial crack length a, that is loaded with tensile stress σa, see Figure 11.1. If the material behaves linearly, the stress can be expressed

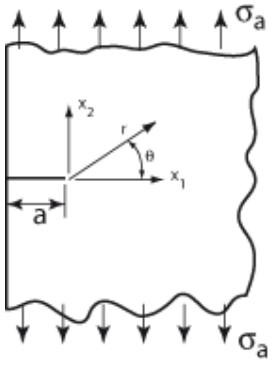

Figure 11.1: *Edge crack in a plate in tension. Mode-I stress intensity factor (Image courtesy of Zehner).*

as:

$$\sigma_{22} = \frac{K_I}{\sqrt{2\pi r}} \tag{11.1}$$

where r is the distance from the crack tip and K_I is related to the stress applied by:

$$K_I = 1.12\sigma_a\sqrt{\pi a} \tag{11.2}$$

The stress is well approximated by eq. 11.1, except close to the tip, namely the *yielding zone*, where inelasticity and non-linearly deformations occur to eliminate the predicted infinite stress. We assume the *small scale yielding* (SSY) assumption, that is, the size of the zone near the crack tip r_p is small relative to a. In a small scale yielding model, the stresses in an annulus $r > r_p$ and $r \ll a$ are well approximated by:

$$\sigma = \frac{K_I}{\sqrt{2\pi r}} f(\theta) \tag{11.3}$$

with respect to polar coordinates, where f is a universal function of θ. All of the loading and geometry of loading are reflected in the single quantity K_I, known as the *stress intensity factor*. The *autonomy* principle states that the distribution of stress around the crack tip has a universal spatial distribution with magnitude given by K_I. The size of the inelastic zone at the crack tip, namely the *plastic zone* or *process zone* is scaled as:

$$r_p \ \frac{K_I^2}{\sigma^2} \tag{11.4}$$

where σ_y is the tensile stress at which inelastic deformation begins to occur.

Fracture Criteria

In SSY, all crack tip deformations and failures are solely driven by K_I. The characteristic resistance to fracture known as *fracture toughness* K_{IC} determines the crack growth. In other words, the crack will grow, if:

$$K_I \geq K_{IC} \tag{11.5}$$

An alternate criterion for fracture is based on the *energy release rate*, G, or energy dissipated per unit area of new fracture surface. As the crack grows in a component, work done on the component by the externally applied forces and strain energy stored in the part prior to fracture, provide energy to the crack. The physical mechanisms of energy dissipation due to fracture include plastic deformation ahead of the crack in metals, microcracking in ceramics, fiber pull out and other frictional processes in composite materials. Other mechanisms are surface energy in all materials, which is generally small relative to the other components, except in glassy materials. In the energy approach the criterion for fracture can be given as:

$$G \geq G_C \tag{11.6}$$

where G is the available energy release rate and G_C is the toughness of the material, or energy per area required to propagate a crack. In SSY, the energy release rate G scales with K_I as:

$$G = \frac{K_I^2}{E} \tag{11.7}$$

where E is the Young's modulus of the material. However, when SSY is violated, the formula no longer applies, which is generally the case for tearing fracture of ductile materials (as mentioned by [O'Brien 2002a]).

For cyclic loadings with $\mathrm{K}_I < \mathrm{K}_{IC}$, the material ahead of the crack will undergo fatigue deformation and eventually failure, and the crack will grow a small amount at each cycle of loading. The rate of crack growth typically scales as $\Delta {K_I}^n$ where ΔK is the difference between the maximum and minimum stress intensity factors due to the cyclic loads, and n is an exponent that must be experimentally determined, but typically $2 \leq \mathrm{n} \leq 4$.

Modes of Fracture

Fracture mechanics mainly relies on the linear analysis of the crack tip field and the calculation of the stress and deformation fields near the tips of the cracks. Two dimensional linear elastic stress is assumed. As sketched in Figure 11.2, the stress field at the crack tip can be broken up into three components, called Mode-I , Mode-II and Mode-III. Mode-I causes the crack to open orthogonally to the local fracture surface and results in tension or compressive stresses on surfaces that are normal to *x2*. Mode-II causes the crack surfaces to slide relative to each other in the *x1* direction and results in shear stresses in the *x2* direction ahead of the crack. Mode-III causes the crack surface to slide relative to each other in the *x3* direction and results in shear stresses in the *x3* direction ahead of the crack. With this idealization, the solution of the crack tip fields can be broken down into three problems; Mode-I and II are found by the solution of either a plane stress or plane strain problem and Mode-III by the solution of an anti-plane shear problem. In many solid mechanics problems the anti-plane shear problem is the simplest to solve.

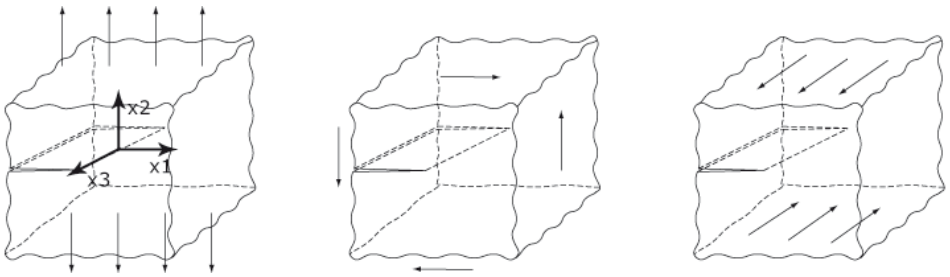

Figure 11.2: *Representation of the state of stress for a cube of material surrounding part of a crack tip. The actual crack is a mix of Mode-I,II,III loadings and this mix varies along the crack front. The tractions on the front and back faces of Mode-III cube are not shown (Image courtesy of Zehner [Zehnder 2009]).*

The displacement fields along the crack faces determine the fracture mode. As an example, in Mode-I there is no relative sliding of the crack faces. Close enough to the crack tip, the stress and displacement fields are completely determined by the values of K_I, K_{II} , and K_{III}. If a crack is loaded in a way that both Mode-I and Mode-II are present, the crack tip stresses will be a superposition of the two solutions of Mode-I and Mode-II.

Three dimensional cracks can be deduced from the two dimensional case study. In Figure 11.1, the crack front is straight through the thickness of the plate and this illustrates a pure Mode-I problem. Studies show that the in-plane stresses, σ_{11}, σ_{22} and σ_{12} are nearly constant through the thickness with the normal stresses dropping off by approximately 25% at the free surfaces. Thus the 2D stress fields provide an accurate description of the 3D problem. However the out-of-plane stress, σ_{33} has considerable variation through the thickness; very close to the crack tip plane strain predominates except in a boundary layer near the free surfaces. For real cracks in three dimensional objects, the stresses will be given by a superposition of the Mode-I, Mode-II and Mode-III fields with the values of K_I, K_{II} , and K_{III} varying at different locations along the crack line.

11.2 State of the Art on Fracture Rendering

Fracture, cracking and tearing have been previously addressed for virtual purposes, mainly concerning the visual rendering of the phenomena. Little research relates to sound modeling emitted during fracture events. In this section, we review previous work on fracture modeling, regarding sound and visual rendering.

11.2.1 Sound Rendering for Breaking and Tearing

In the work from Warren and Verbrugge [Warren 1984], breaking and bouncing sounds are studied in order to extract specific auditory patterns. Based on acoustic

analysis and a qualitative physical analysis, they postulate that breaking and bouncing events are essentially characterized by temporal properties, whereas static spectral properties have only little influence. Experiments with synthetic sounds are conducted to verify the hypothesis. Starting from this analysis, Rath et al. [Rath 2003a] propose sound models for bouncing, breaking, and crumpling, using only temporal organization and dynamic control of micro-impacts. Crumpling events are created by triggering impacts according to stochastic laws, based on dynamic and temporal statistics.

Salminen et al. [Salminen 2002] address the acoustic emission from paper fracture. Paper shows local fluctuations of areal mass density. Starting from this observation, they assume that the inhomogeneity of material should have consequences in crack propagation and acoustic emission. The average strength can follow scaling laws due to the presence of disorder in the material. It appears that paper failure shows unusual physical properties that makes it difficult to model. First, one can not clearly detect a *critical point*, namely the point at which the crack propagates, or phase transition. Second, a complex time-dependent temporal behavior is observed, which is not directly related to the fast relaxation of stress. This might be due to the viscoelastic nature of the wood fibers. Third, power-laws do not agree with the predictions of simple fracture models.

Concerning sound modeling for tearing, the only related study is the approach of Terzopoulos and Fleischer [Terzopoulos 1988] on modeling tearing cloth. The sound rendering consists in playing a pre-recorded sound whenever a connection in a spring mesh fails.

11.2.2 Visual Modeling for Fracture

O'Brien and Hodgins [O'Brien 1999] address cracks for three-dimensional objects. Based on stress tensor computation over a finite element model, crack initiation is localized and the directions in which cracks should propagate are determined. This model is further developed for ductile fracture [O'Brien 2002a] by appending a plasticity model that expresses the interactions between plastic yielding and fracture process. The approach allows realistic animation of ductile fracture in common solid materials such as plastics and metals, but not in real-time. Specific materials such as woven fabrics can not be handled, because of the several cycles of elastic and plastic behavior they may endure. On the other hand, Pauly et al. [Pauly 2005] propose a meshless method for elastic and plastic materials with a highly dynamic surface and a volume sampling method that can support arbitrary crack initiation, propagation, and termination. The method overcomes several stability problems that arise when using traditional mesh-based techniques. Continuous propagation of cracks is achieved with highly detailed fracture surfaces, independent of the spatial resolution of the simulation nodes. The simulation is not performed in real-time, and it takes, in one example, about 22 seconds per frame for the computation of brittle fracture of a stone sculpture.

According to Oda et al. [Oda 2005], an optimal model for fracture should preferably be fast and respond to user actions in a natural way rather than being too realistic. Starting from this idea, they introduce a method based on multi-stage dynamic refinement, achieving realistic results with moderate computational cost. The method operates in 2D, and it has not been tested on 3D objects. For a more expressive approach, Mould proposes image-guided fracture and combines NPR rendering approaches with fracture [Mould 2005]. An input line is altered through an image filter to generate an image of a fractured surface. Cracks are rendered either based on image analogies or by modulation of an uncracked texture.

More recently, Parker et al. [Parker 2009] have addressed deformation and fracture of solid objects, especially in the context of real-time games. They combine previous research in graphics and computational physics to create an engine that targets real-time user interactions, persuasive situations, appropriateness for the design of a game pipeline, and flexibility for authoring. Using a similar but less advanced approach, a Maya plugin *Fracture*[1] for procedural destruction of 3D objects is currently under development, with a specific focus on an easy-to-use solution.

11.3 Overview of Our Hybrid Model

When material breaks into pieces the sound corresponding to the fracture and to each specific sounding piece have to be appropriately rendered. In this section, we develop our approach to address sound of fracture events.

Figure 11.3 underlines the main characteristics of a fracture sound event we will be focusing on, namely timbre, rhythm, structure and human knowledge. However, a perceptually based model will not be directly investigated, but still stays in mind since we aim at expressiveness. Timbre refers to the spectral distribution of the fracture sound event, whereas rhythm and structure deal respectively with the underlying temporal regularity and the segmentation of the sound sequence into temporal groups on the basis of duration [Peretz 2003] (see Section 8.1).

We propose a hybrid model for fracture sound synthesis based on the underlying characteristic attributes. Our goal is an expressive and efficient model for audio-visual animations focused on quality and variety. We propose a combination of physically based and example-based approaches. We focus on a representation that is general enough to render the large variety of sounds emitted from brittle and ductile fractures, but also from fracture of fiber-like structure such as wood.

11.4 Parametrization of the Model

Fracture events can be described by three distinct stages, namely crack initiation, crack propagation, and creation of fragments. Fracture stages are different due to the mechanical process which takes place from a more microscopic point of view,

[1] Fracture Demolition Software http://www.fracture-fx.com/

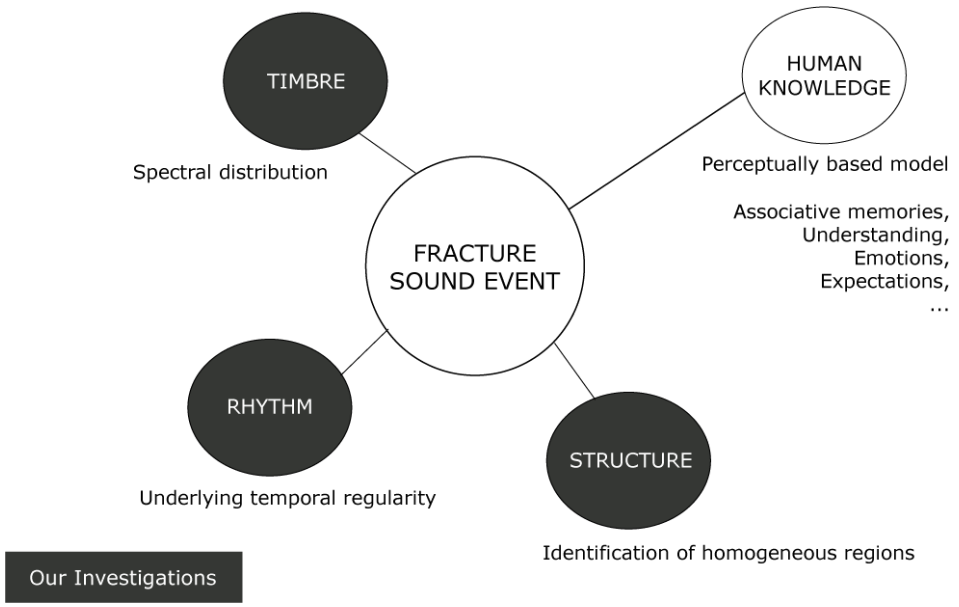

Figure 11.3: *A model strategy for fracture sound event. Our proposals are underlined in dark gray.*

which in consequence, affects the resulting sound. It is better to achieve integration between different modeling approaches in a single coherent sound than independently synthesizing audio streams corresponding to each fracture stage. In the virtual scene, a smooth sound transition should be experienced when moving from object interactions to object fracture, but also from one fracture stage to another in fracture sound modeling. Parameters of the hybridization have to be determined and we suggest synthesis algorithms governed along dimensions of the fracture event, similar to the approach of Gaver [Gaver 1993b].

Figure 11.4 presents an analysis of the fracture event. Procedures for sound effects are proposed according to the different fracture stages. Synthesis approaches are closely related to the linearity/nonlinearity characteristic of these stages. Sound rendering can be seen as flexible audio shading allowing procedural choice of sound parameters. These sound parameters could be driven by information reported by the physics engine, such as stress tensor values, or according to user-defined procedures. In order to ensure sound coherence between object interaction and fracture, as well as through fracture stages, some of the sound parameters are maintained through the algorithms.

Note that the approach presented is particularly well adapted for brittle fracture, similar to that one represented in Figure 11.5, where brittles have been created using the *Power Booleans 3.0* software[2], and the *Power Cutter tool*. However, we believe that this model can be easily extended to ductile fracture or fracture from fiber-like

[2]Power Booleans 3.0 http://www.npowersoftware.com/booleans/pboverview.htm

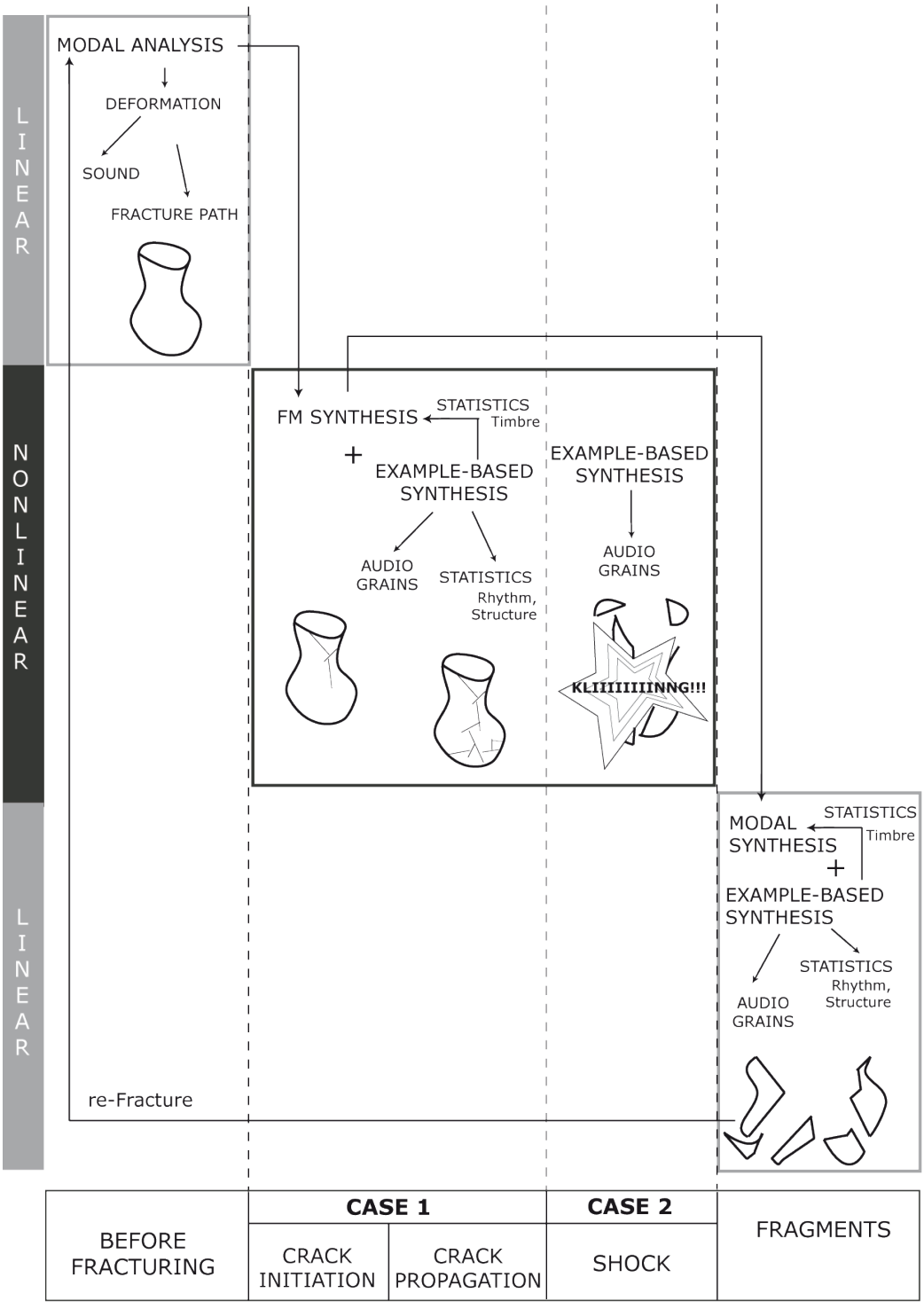

Figure 11.4: *Our hybrid model for fracture sound events*

Figure 11.5: *A fracture example: a breaking glass. (Pieces have been created using the Power Booleans 3.0 software, and the Power Cutter tool)*

materials.

11.4.1 Before Fracturing

Fractures result from internal stresses generated as the material deforms. Modeling fracture with a physically based technique implies modeling the deformations that induce fractures. An adequate deformation method must provide information about the magnitude and orientation of the internal stresses, and notify if the stresses are tensile or compressive [O'Brien 1999]. Deformation techniques commonly consist in defining a set of differential equations that represent the behavior of the object material in a continuous fashion. A finite element discretization is then performed for computer simulation. Several approaches can be formulated in this way. However, our main goal is to design a model that is simple and fast for fracture modeling.

The goal of modal analysis is to determine the natural mode shapes and frequencies of an object or structure during free vibration. By performing modal analysis, fracture simulation and sound rendering can be addressed jointly. Modal synthesis has been often adopted for sound synthesis because of its control and simplicity of implementation, in conjunction with its compact formulation of the audio data. Our modal analysis approach presented in Chapter 7 [Picard 2009a] can be particularly appropriate to efficiently model the patterns of fracture. Indeed, the technique is flexible by proposing different scales of resolution and can be used for a variety of complex geometries. In addition, since it is built on a physically based animation engine, the *Sofa Framework*[3], the same model for simulation and sound modeling is used and problems of coherence between physics simulation and audio are avoided.

For the sake of efficiency for real-time rendering, we may prefer an approximate model which is still realistic and expressive. The work of Gladden et

[3]Simulation Open Framework Architecture http://www.sofa-framework.org/

al [Gladden 2005] on dynamic buckling and fragmentation in brittle rods derives a preferred wavelength λ for the buckling instability, and experimentally verifies the resulting scaling law for a range of materials including teflon, dry pasta, glass, and steel. For brittle materials in 1D, fragmentation appears mainly near $\lambda/2$ and $\lambda/4$. These results for 1D can inspire a simple but realistic fracture model, that generate random fracture patterns around $\lambda/2$ and $\lambda/4$ of the first vibration modes, assuming that these modes contain the main part of the energy of the structure.

11.4.2 During Fracture Event

Due to the sudden release of energy, fracture events can present nonlinearities in their mechanical behavior. To be valid, a pure physically based approach would require direct integration of the dynamic equation of equilibrium. However, this approach is too expensive for real-time sound synthesis. For this reason, we propose a hybrid model combining physical and example-based approaches. A suitable parametrization of the hybrid model is the main focus.

During a fracture event, two main scenarios can be considered, that is, the case when cracks arise and propagate in an observable time, and the shock case.

Case 1: Crack Initiation - Crack Propagation

In the first phase of the fracture process, changes of the sounding object are assumed to be moderate. In particular, we can assume that modifications mainly concern the stiffness of the material, the shape and size remaining the same. Structural damage detection has shown that dynamic responses of a structure vary according to its inherent damage [He 2001]. Damage detection formulates relationships between the damage and modal parameter changes of a structure. Our approach can use these results to deduce plausible evolution of modal parameters. Furthermore, the work from Maxwell et al [Maxwell 2007] can guide us to avoid the recomputation of the eigen-decompositions. They address estimation of frequencies for nearby geometries by using modes computed for one geometry. In Maxwell's PhD thesis [Maxwell 2008], numerical tools and software systems are developed to interact with geometric shapes that synthesize sound. One of the purposes of the work is to rapidly formulate the updated modal parameters of a geometric model as the shape is changing. The study shows that for moderate changes, it is possible to avoid recompution of the eigen-decompositions and the resonant frequencies can be deduced quite easily. However, as cracks propagate, critical changes appear for the natural frequencies extracted during analysis. Modal superposition is consequently no longer valid and we propose to apply our hybrid model.

We first assume that the modifications of the object being fractured can be tracked during the event. We need a method that appropriately renders the sounds in correspondence to the modifications of the sounding object. The abrupt release of energy tends to increase the frequency content of the response of the sounding object. FM synthesis may appropriately model the increase of the frequency content

as the fracture progress. Figure 11.6 illustrates the method for a coherent hybrid sound synthesis. We propose to parameterize the FM algorithm with the modal parameters of the object before fracturing. More precisely, modal frequencies are used as a bank of carrier frequencies for FM synthesis. To corroborate the frequency

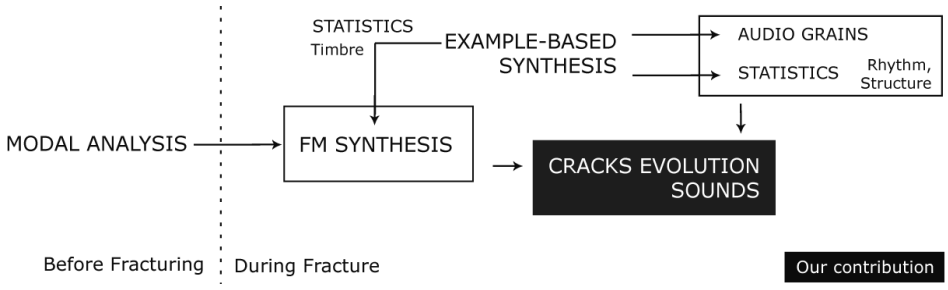

Figure 11.6: *During fracture, crack initiation and propagation are addressed through a hybrid model.*

content evolution, statistical analysis of a fracture sound database can be used to extract the timbre and to further adjust the FM synthesis. For more perceptually based modeling, spectral analysis can take into account the processing of the human auditory system, as for instance, auditory filtering or frequency masking. We propose to choose the index modulation of the FM synthesis, $\Delta f_c/f_c$, and its modulator frequency f_m according to the extracted spectral features. For this purpose, we introduce a graphical interface, see Figure 11.7. It takes a sound as input and with adequate tools can easily deduce FM parameters.

We now assume that crack propagation cannot be tracked from the physics engine. Thus, we have to model the rhythm and the structure of the sound fracture event. Spectral structure of the fracture event can be extracted from statistical analysis of a sound database. In Figure 11.8, a method for extraction of spectral structure is proposed. A breaking glass event is analyzed for the temporal evolution of its frequency content. The novelty of the method is the analysis per audio grain. Indeed, we assume that spectral evolution across grains has higher perceptual significance than a simple spectrogram. Audio grains are extracted with a prior computation of spectral flux, similar to the method presented in Chapter 9. To reproduce the distribution of crack propagation, we suggest a method related to previous work from Avanzini et al. [Avanzini 2002a] on bouncing and breaking events. Their sound models are derived from temporal organization and dynamic control of micro-impacts, based on loss of macro-kinetic energy. We assume that this approach is suitable to model the sudden release of energy during crack propagation. We propose a simple prototype that models the temporal distribution of cracks, or micro-events, with a Poisson distribution. Micro-events are rendered with audio grains extracted from a sound database. In our example, see Figure 11.9, only two audio grains are handled. Low-pass filtering is applied to render the smothered effect of crack sound. Despite its simplicity, the model synthesizes a plausible crack

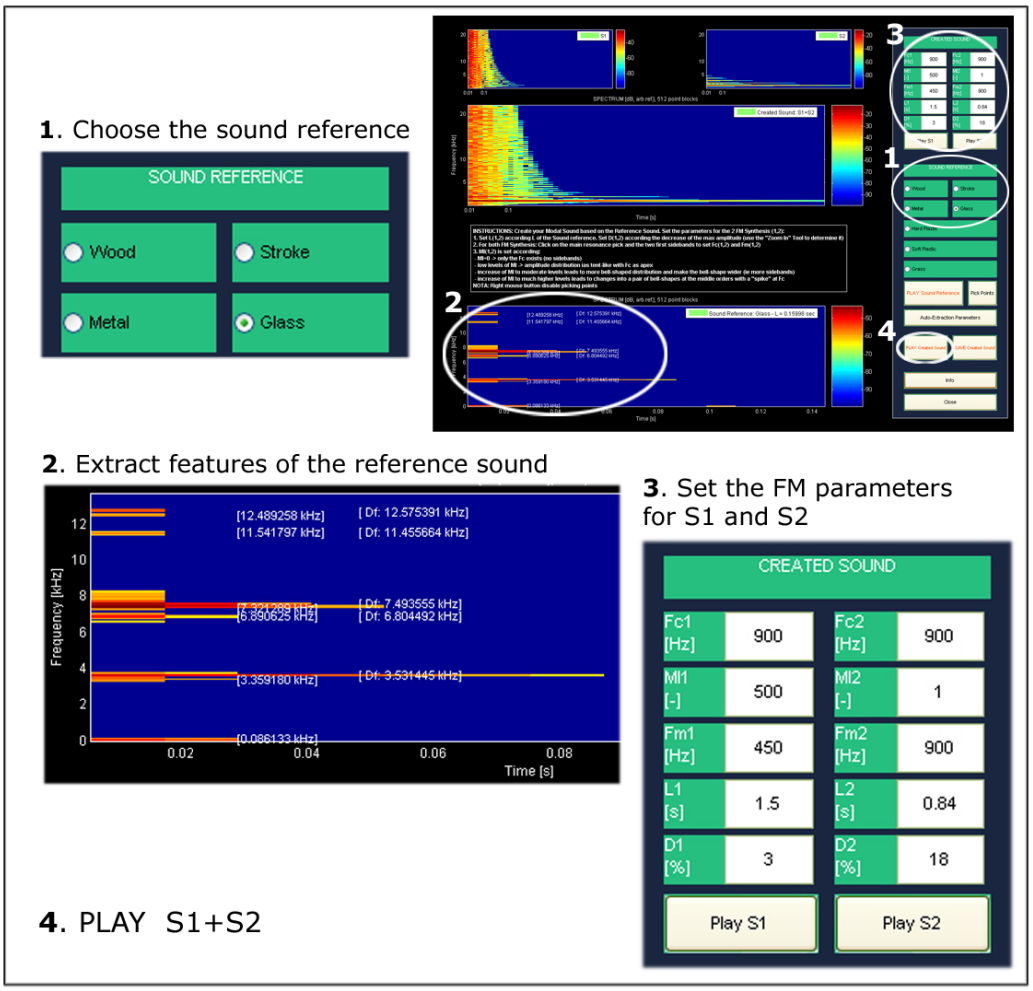

Figure 11.7: *Guiding FM synthesis with a graphical interface. Rich spectral content of a sound recording is modeled with the sum of two FM signals*

sound.

FM synthesis may sound too *clean* in practice and removes some properties of the sound emission. In particular, it does not take into account the propagation of disturbances. Example-based sound synthesis can address expressiveness of the sound event. Audio grains can target the noticeable texture that may be difficult to compute. However, this implies adequate parametrization. The method developed by Picard et al. [Picard 2009b] aims at effectively solving this problem by retargetting audio grains to physical parameters and/or user-defined procedures. To ensure coherence between FM synthesis and audio grains, spectral attributes of FM synthesis and audio grains should be consistent. For this purpose, we suggest to choose audio grains that are more closely related to FM synthesis in terms of spectral attributes.

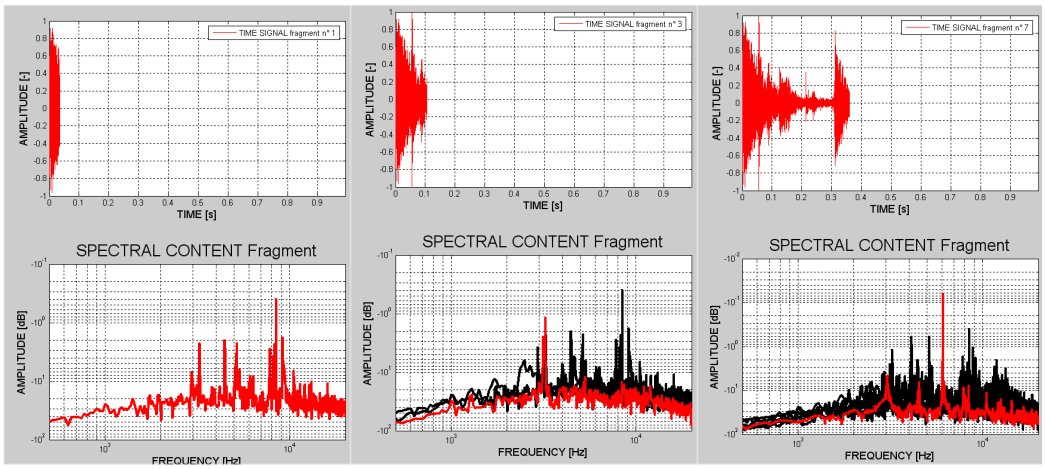

Figure 11.8: *Spectral analysis on a breaking glass event. The novelty of the approach is the analysis per audio grain.*

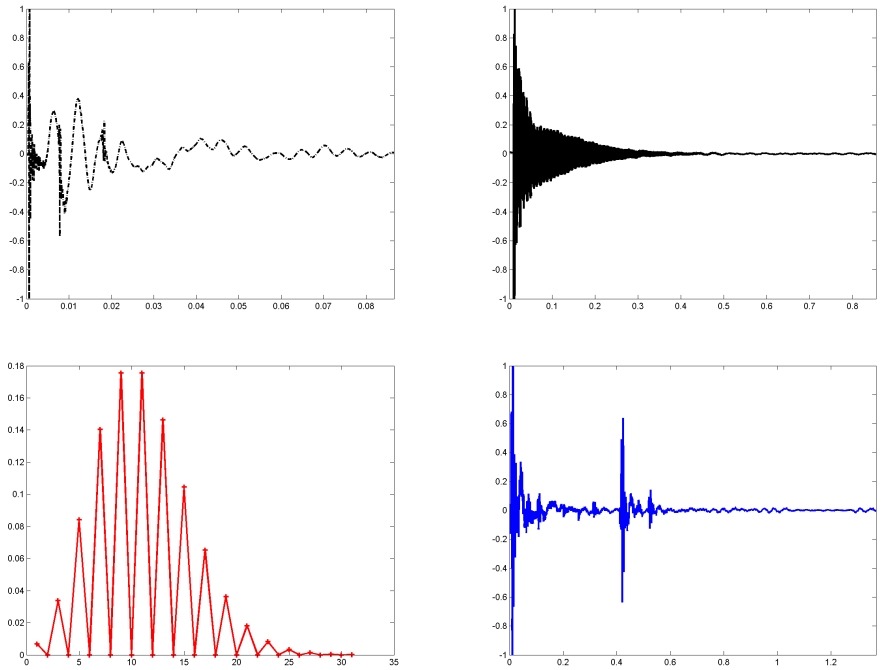

Figure 11.9: *A simple crack propagation model using Poisson distribution. From top to bottom, and left to right, the two audio grains used for the sound synthesis, the Poisson distribution for temporal distribution of cracks, and the resulting crack sound event.*

Case 2: Shock

In the case of shock, frequency content of the response suddenly increases, and techniques such as modal superposition become less effective. Because of the short

duration of the event and the wide-band spectral content, we suggest an example-based approach. We propose to use audio grains appropriately retargetted to velocity and force reported from the physics engine. To address coherence between the vibrational model and the hybrid model, we propose to select the audio grains that correlate more strongly the natural frequencies of the vibrational model.

11.4.3 After Fracturing, Brittles

After fracturing, we obtain fragments of the initial sounding object. Sound is still related to the original object, mainly due to the material of the resulted fragments, but the shape and dimensions have changed. Modal parameters for the created fragments can not be predefined due to their highly dynamic character. For this reason, we propose a modal synthesis triggered by the spectral distribution of the FM synthesis algorithm used in the crack propagation stage. Sound interactions of brittles are modeled using audio grains for the early part of the sound and modal synthesis for the sustained part.

If the fragments obtained are sufficiently large, they may be able to refracture. In this case, the all cycle is processed again.

11.5 Discussion

The model presented is prospective and further work should be done. A complete implementation of the fracture sound simulation framework may reveal the limitations of the method and further improvements. In particular, tracking the relevant physical data related to the crack propagation could be a difficult task, especially for real-time applications. Even if spectral structure and temporal distribution of the crack propagation can be modeled using a statistical approach, values that inform fracture stages are needed. In this case, prediction values should be investigated.

Most importantly, fracture is an appropriate application for a hybrid model, since the physically based sound model cannot adequately represent the phenomenon that produces sound. The difficult part in hybrid modeling is to ensure, on one hand, a smooth sound transition between object interactions in the virtual scene and object fracture stages, and on the other hand, the coherence between components of the hybridization. Simplicity of the tools allows efficient sound rendering for real-time applications.

In this part, we have presented our prospective research on a hybrid model that accordingly combines a physical model and an example-based model for sound rendering of physical interactions. We have applied our ideas on the fracture event, since this physical phenomenon is widely used in games, training simulations, and other interactive virtual environment. We proposed procedures for sound rendering of fracture, depending on the stage of the fracture process, that is, crack initiation, crack propagation, and creation of fragments. The techniques adopted are intimately depending on the linearity/nonlinearity of the stage in process. We presented prototypes for sound modeling, according to spectral and temporal features. Although this study is at prospective stage, we believe that these early results are promising and should be extended in future work.

Chapter 12

Conclusion

Contents

The work presented in this thesis is an effort to tackle the large variety of sounds resulting from physical interaction of objects in a typical virtual scene, such as in video games. Four main reasons make them difficult to be appropriately rendered with traditional sound rendering solutions, which generally use playback of prerecorded samples.

1. Audio-visual coherence has to be preserved.
2. Sounds from interactions are extremely dynamic and change drastically in time. This makes them hard to precompute efficiently.
3. Sounds from highly detailed physical behavior have to be precisely rendered.
4. A large variety of objects have to be simultaneously processed.

In this thesis, we proposed solutions to the problems outlined above.

12.1 Contributions

Complex Contact Modeling

We focused on providing several detail-layer mechanisms that allow for realistic contact sound modeling while maintaining simple solid dynamic simulations. We proposed a novel approach for generating complex contact sounds in the context of interactive simulations. The two-dimensional visual textures of objects in interaction are used as roughness maps to create audible and position-dependent variations during rolling or sliding. A flexible audio pipeline enables the choice of different detail levels according to the desired granularity of the rendered impacts.

The approach presented unifies and develops previous work from different fields, in the same framework. We believe that our technique is useful due to the increasing use of procedurally-generated content especially in games, and to the memory savings it offers, as the complexity of the environments grows.

Improved Modal Analysis for Resonator Modeling

In order to improve current modal analysis for sound rendering, we proposed a new approach which specifically addresses virtual environments with a large variety of objects and efficient memory usage. The technique performs automatic voxelization of a surface model and automatic tuning of the finite elements parameters, based on the distribution of material in each cell. In contrast to previous methods, we can handle complex non-manifold geometries that include both volumetric and surface parts. Our approach thus allows the computation of the audio response of numerous and diverse sounding objects, such as those used in games, training simulations, and other interactive virtual environments.

Since the number of hexahedral finite elements only loosely depends on the geometry of the sounding object, the technique affords a multi-scale solution. Problems of coherence between simulation and audio are finally easily addressed considering that the technique is built on a physics animation engine, the *Sofa Framework*[1].

Flexibility of The Sound Design

In the context of simulations driven by a physics engine, we proposed a technique which creates convincing soundtracks in real-time, synchronized with the animation. Our approach dynamically retargets audio grains to the events reported by the simulation engine. Audio grains and correlation patterns are automatically extracted from a database of recordings, with a novel method which operates off-line.

The approach finds a compromise between memory footprint and variety at synthesis time, and discharges the sound designer from the task of having to create a large variety of sounds by hand. We believe our work addresses important issues for both audio programmers and designers by offering extended sound synthesis capabilities in the framework of standard sample-based audio generation.

Perspectives on a Hybrid Model for Complex Physical Phenomena

We extended the capabilities of a purely physical model by proposing a hybrid approach that appropriately combines physically based and example-based methods. The hybrid model especially targets nonlinearity and other complex physical phenomena where a single model would fail to properly render the sound.

Due to their extensive use in games, fracture sounds are treated specifically and prospects for an efficient and flexible hybrid model are presented. The parametrization of the model is defined according to the underlined stages of the fracture event. Smooth sound transition between object interactions and object fracture stages, and coherence between components of the hybridization are managed with adequate parametrization of a waveshaping synthesis determined by physical and example-based parameters. The simplicity of the formulation makes the model convenient

[1] Simulation Open Framework Architecture http://www.sofa-framework.org/

for situations where numerous objects have to be rendered simultaneously such as in typical fracture scenes.

12.2 Extensions and Applications

Besides the proposed solutions, this dissertation offers many promising directions for future work.

Our modeling of complex contact interactions does not take the case of two interacting textures into account, where features intersect creating specific excitation profiles. Additional consideration of the interaction modeling is consequently needed. The position-dependent variations have been considered according to a point-based computation, and interaction with precise surfaces in contact should be now examined. Finally, adequate perceptual experiments should be undertaken to test the relevance of our approach. Audio quality should be evaluated in comparison to reference sequences where contact sounds are modeled using the technique of van den Doel [van den Doel 2001] for scraping and rolling. In their model, sound resulting from the interaction with surface features is rendered only if these features are detectable by the physics engine. When visual textures are used to render the surfaces, no geometric information is provided. We believe that quality ratings will reveal the relevance of our approach on sound rendering of prominent features.

Our method for modal analysis allows the reduction of computational expense. Further investigations with graphics processing units (GPU) may improve the efficiency of the technique. Thus, real-time modal analysis could become possible on the fly, with sound results that are approximate but still realistic for virtual environments. This would, in particular, enable us to address modal synthesis for fracturing objects, since they are highly dynamic sound events.

The example-based technique we proposed can be improved with higher level statistical analysis of the patterns obtained for similar recordings. In addition, efficient clustering of similar grains could be investigated to further reduce the size of the data structures and the pre-processing time. Due to the extensive use of physics-driven animation, the design structure of the physics engine itself could be explored in terms of physical sound modeling. Thus, information collected from the physics engine would be more easily exploited for sound rendering.

Our hybrid model for fracture sound is prospective and further research should be conducted to construct the entire fracture sound simulation framework. The main issue is to track the relevant physical data related to crack propagation and this is probably a difficult task, in particular for real-time applications. In this case, further research is required to extract other prediction values.

To conclude, we believe that the goals set out in the introduction have been achieved. Indeed, new physically based algorithms for sound rendering have been

developed, flexibility of sound modeling has been addressed, and ideas on an adequate hybrid sound model have been proposed. We believe that the favorable results and the directions for future work described above reveal the strong potential of this research.

Appendix A

Modal Superposition An Overview

The purpose of this Appendix is to present how sound impulse response of an arbitrary object can be calculated by means of modal superposition. This will give the mathematical background behind modal superposition for discrete Multi-DOF (MDOF) systems with proportional damping.

To apply modal superposition, we assume the steady state situation. In other words, we consider the sustained part of the impulse response of an object being strucked. Indeed, the early part, which is of very short duration, contains many frequencies and is consequently not well described by a discrete set of frequencies.

Modal superposition uses the Finite Element Method (FEM) and determine the impulse response of vibrating objects by means of a superposition of eigenmodes.

A.1 Derivation of the equations

We first consider the undamped MDOF system; its equation of motion is expressed by:

$$[M]\ddot{x} + [K]x = f \tag{A.1}$$

where [M] and [K] are respectively the mass and stiffness matrices of the discrete MDOF system. The mass matrix is typically a diagonal matrix, its main diagonal being populated with elements whose value is the mass assumed in each DOF. The stiffness matrix is symmetric (often a sparse matrix, i.e. only a band of elements around the main diagonal is populated and the other elements are zero). In finite elements, these matrices are assembled based on the element geometry and properties.

Our study is in the frequency domain, and for this reason, the displacement vector x and the force vector f are based on harmonic components, that is, x = X $e^{j\omega t}$, $\dot{x}$ = j ω X $e^{j\omega t}$, $\ddot{x}$ = - ω^2 X $e^{j\omega t}$ and f = F $e^{j\omega t}$. The equation of motion can be rewritten:

$$X = ([K] - \omega^2[M])^{-1}F \tag{A.2}$$

This form is the direct frequency response analysis. The term [K]-$\omega^2[M]$ needs to be calculated for each frequency. To calculate the response to any excitation force *F(ω)*, we need to solve the eigenvalue problem:

$$([K] - \omega^2[M])X = 0 \tag{A.3}$$

or

$$([M]^{-1}[K])X = \omega^2 X = \lambda X \tag{A.4}$$

This equation says that each sounding object has a structure-related set of eigenvalues λ, which are simply connected to the system's eigenfrequencies. To extract the eigenvalues, the following condition has be fulfilled:

$$det([K] - \omega^2[M]) = 0 \tag{A.5}$$

Solving Equation A.5 implies finding the roots of a polynomial, which correspond to the eigenvalues λ. The latter can then be replaced in the Equation A.3:

$$([K] - \lambda[M])[\Psi] = 0 \tag{A.6}$$

Ψ is the matrix of eigenvectors, or eigenfunctions, where the column r is the vector related to the eigenvalue ω_r^2. The eigenvectors define the mode shapes linked to the corresponding frequency of the system.

If the eigenfrequencies are unique, many eigenvectors can be extracted for a given eigenvalue and all are proportional. Thus, the information enclosed in the eigenvectors is not the absolute amplitude but a ratio between the amplitudes in the degrees of freedom. For this reason, the eigenvectors are often normalized according to a reference. Due to the orthogonal property of the eigenvectors, $[\Psi]^T[\Psi] = [\mathrm{I}]$. Consequently, $[\Psi]^T[\mathrm{M}][\Psi]$ and $[\Psi]^T[\mathrm{K}][\Psi]$ are diagonal matrices, and are respectively called the modal mass and the modal stiffness of the system, because the ratio between modal stiffness and modal mass give the matrix of eigenvalues. A very suitable reference choice is to scale the eigenvectors so that the modal mass matrix becomes an identity matrix. Back to the Equation A.2, we can write:

$$\begin{aligned} [\Psi]^T([K] - \omega^2[M])[\Psi] &= [\Psi]^T \frac{F(\omega)}{X(\omega)}[\Psi] \\ ([\lambda] - \omega^2[I]) &= [\Psi]^T \frac{F(\omega)}{X(\omega)}[\Psi] \end{aligned} \tag{A.7}$$

and finally:

$$X(\omega) = [\Psi]([\lambda] - \omega^2[I])^{-1}[\Psi]^T F(\omega) \tag{A.8}$$

Equation A.8 simply expresses that the response *X(ω)* can be calculated by surimposing a set of eigenmodes weighted by the excitation frequency, multiplied with an excitation load vector $\mathrm{F}(\omega)$.

Properties of eigenvalues and eigenvectors

The orthogonality of modes expresses that each mode contains information which the other modes do not have, and consequently a given mode can not be built from the others. On the other hand, solutions of geometrically symmetric systems often give pairs of or multiple eigenmodes.

Boundary conditions are settled simply by prescribing the value of certain degrees of freedom in the displacement vector. As an example, a structure being screwed infinitely rigid to the ground will show null DOFs around the support point. In consequence, the elements in the mode shapes corresponding to these DOFs will always be zero and will not need to be solved.

A.2 Damping

We now consider a damped system, and in particular the proportional damping model which assumes that the damping can be expressed proportional to the stiffness and mass matrix (Raleigh damping), that is, [C]= β[K] + γ[M]. In consequence, the eigenvalues of the proportional damped system are complex and can be expressed according to the eigenvalues of the undamped one:

$$\lambda'_r = \omega_r^2(1 + i\eta_r) \tag{A.9}$$

where the imaginary part contains the loss factor η_r.

The modal superposition is thus given by:

$$X(\omega) = [\Psi]([\lambda] - \omega^2[I] + i[\eta][\lambda])^{-1}[\Psi]^T F(\omega) \tag{A.10}$$

Equation A.10 enables us to determine entire response velocity fields that causes the surrounding medium to vibrate and to generate sound.

Appendix B

Validation of our Modal Analysis on a Metal Cube

In order to globally validate our modal analysis [Picard 2009a], described in Chapter 7, we study the sound synthesized when impacting a cube in metal. This example is interesting due to its symmetry. In particular, the cube should sound the same when impacting normal to the face at the 8 corners, as well as when hitting with a force normal to a pair of the faces of the cube.

The chosen cube is made of steel with density 7850 kg/m^3, the Raleigh coefficients $\alpha 1$ and $\alpha 2$ are equal to 3x10^{-7} and 10 respectively. The excitation force is modeled with a Dirac.

B.1 Frequency Content

Our new approach for modal analysis is applied on the cube. Figure B.1 shows the distribution of the extracted modes.

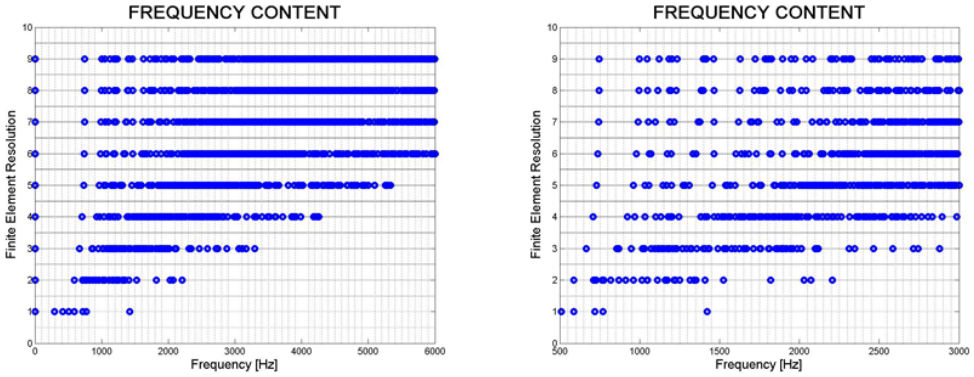

Figure B.1: *Frequency modes for the cube. From left to right, the frequency modes are focused in the range [0;6000Hz] and [500;5000Hz]. Modal analysis is performed with different resolutions for the finite elements: 1×1×1, 2×2×2, 3×3×3, 4×4×4, 5×5×5,6×6×6, 7×7×7, 8×8×8, and 9×9×9.*

The frequency modes depend on the resolution of the hexahedral finite elements. The higher models have a wider range of frequencies because of the supplementary

degrees of freedom. We also observe a frequency shift as the FEM resolution increases. Since 1×1×1 or 2×2×2 grid represents an extremely coarse embedding, it is not surprising that the frequency content is very different in compare to higher resolutions. Above all, an important feature is the convergence in frequency content as the FEM increases. Listening tests might allow us to detect the resolution which offers a sufficient quality.

B.2 Excitation Direction

We compare the sound created when impacting on one corner with three perpendicular forces, each normal to one pair of the cube faces. The sound synthesis is modeled with the reson filter approach [van den Doel 2001], and all the modes extracted from the modal analysis are kept for the sound synthesis.

Figure B.2 shows that the sound is almost the same when impacting with the three forces. The differences in the frequency content might be due to the non-perfect homogeneity of the matter distribution across the three perpendicular directions.

B.3 Excitation Position

We compare the sound created when impacting on the different corners of the metal cube with a force normal to one face. The sound synthesis is modeled with the reson filter approach, using all the modes extracted from the modal analysis.

For the sake of room, we just give the results for impacts on four corners, with different resolutions of the hexahedral finite elements (see Figure B.3). The frequency content of the sounds is shown to be similar within a given resolution. We also notice that very coarse resolution, such as 1×1×1 grid resolution, give very different results in comparison to higher resolution and the results converge as the FEM increases.

B.4 Conclusion

The results demonstrate that:

1. The frequency content of the sounds depend on the resolution of the hexahedral finite elements.

2. The higher models have a wider range of frequencies because of the supplementary degrees of freedom.

3. There is a frequency shift as the FEM resolution increases.

4. a 1×1×1 or 2×2×2 grid represents an extremely coarse embedding, and consequently may not be accurate to synthesize the sound of objects.

5. There is a convergence in frequency content as the FEM increases.

6. The resulting sounds when impacting on one corner with three perpendicular forces, each normal to one pair of cube faces, are quite similar.

7. The resulting sounds when impacting on different corners of a cube are similar.

This almost validates our method for modal analysis. Moreover, better approximations of the mass and stiffness of the finite elements would probably improve the results, in particular using the recent work from Nesme et al. [Nesme 2009].

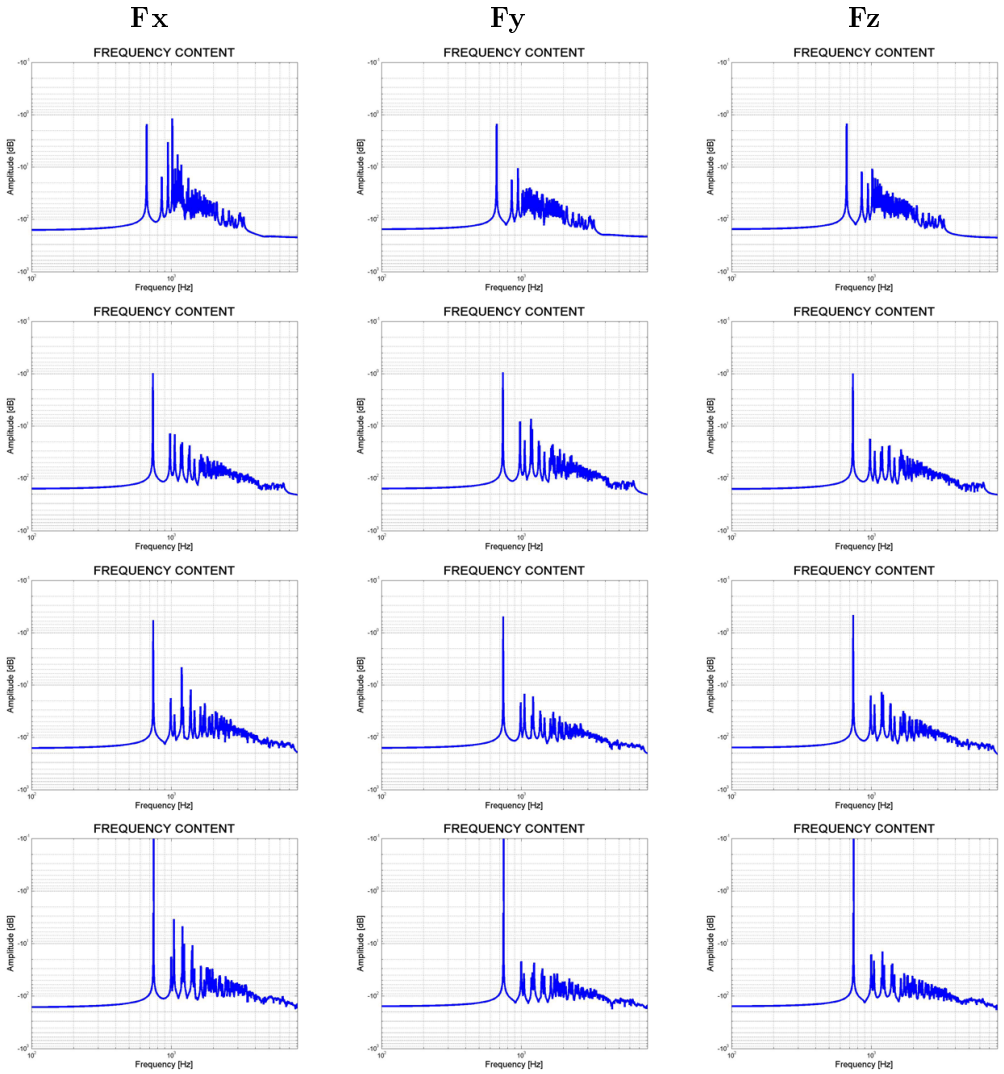

Figure B.2: *A cube in metal is sounding when impacting on one corner with three perpendicular forces, each normal to one pair of the cube faces, from left to right. Modal synthesis is performed with two different resolutions for the hexahedral finite elements: 3×3×3 (192 modes), 6×6×6 (1029 modes), 7×7×7 (1536 modes) and 9×9×9 (3000 modes), from top to bottom.*

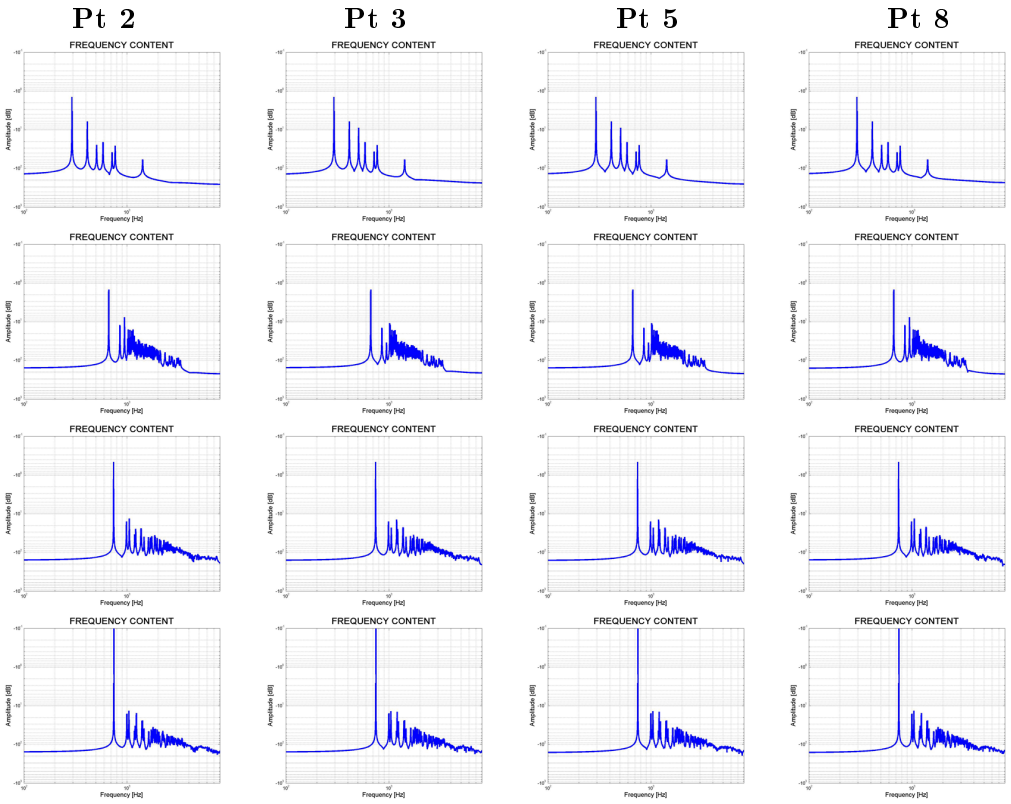

Figure B.3: *Frequency content of the impulse response of the cube when impacting on four corners, from left to right. Modal synthesis is performed with 2 different resolutions for the hexahedral finite elements: 1×1×1 (24 modes), 3×3×3 (192 modes), 7×7×7 (1536 modes) and 10×10×10 (3000 modes), from top to bottom.*

Appendix C

Signal Processing Formulas

Since the auditory system behaves as a spectrum analyzer, spectral representations, also known as Fourier representations, have been well studied and widely used in sound modeling.

This Appendix present the main principles of Fourier-based analysis and we summarize the signal processing formulas we used in our approach of example-based sound synthesis (Chapter 9).

C.1 Basics

The Fourier Transform

The Fourier transform converts a time domain waveform into a frequency domain spectrum, that is, the waveform sinusoidal components. The Fourier transform is commonly defined as:

$$X(\omega) \triangleq \int_{+\infty}^{-\infty} x(t) e^{-j\omega t} dt \tag{C.1}$$

where t and ω are respectively, the continuous time index (in seconds) and the continuous frequency index (in radians per second).

X(ω) is a complex number *(a+ib)* for every value of ω. The magnitude,$\sqrt{a^2+b^2}$, defines the amplitude of a sinusoidal component, and the angle, $tan^{-1}(\frac{b}{a})$, defines its phase. *X(ω)* is a periodic function of ω with period 2π and the original signal *x(t)* can be retrieved by means of the inverse Fourier transform:

$$x(t) \triangleq \frac{1}{2\pi} \int_{+\infty}^{-\infty} X(\omega) e^{j\omega t} d\omega \tag{C.2}$$

The Discrete Fourier Transform (DFT)

In computers, a signal waveform is sampled according to the sampling frequency f_s and the continuous Fourier transform is formulated into the discrete Fourier transform (DFT):

$$X(k) \triangleq \sum_{N/2-1}^{n=-N/2} x(n) e^{-j\omega_k n} \tag{C.3}$$

where n is the discrete time index in samples, *x(n)* is the original signal N samples long, ω is the discrete radian frequency, and k the discrete frequency index in bins.

The DFT considers *x(n)* to be represented by a finite number of sinusoids, which implies that the signal *x(n)* is bandlimited in frequency. The DFT converts the time sequence of length N into a frequency domain sequence of length N, equally spaced between 0Hz and f_s.

The main advantage of DFT is its computational efficiency through the use of the fast Fourier transform (FFT). FFT is applied on a signal length that is power of 2, and allows to reduce the computational time from one proportional to N^2 for the DFT to one proportional to $N\log N$ for the FFT.

The Short Time Fourier Transform (STFT)

In general, the input signal is non-periodic and time-varying, which makes the Fourier transform and the DFT not adequate. *Leakage* is a consequence of the DFT's assumption periodicity, which can cause energy to appear at frequencies in the spectrum where there is none. For this reason, the short-time Fourier transform (STFT) applies the Fourier transform per portion of signal with a jump-reducing window *w(n)*:

$$X_l(k) \triangleq \sum_{N-1}^{n=0} w(n)x(n+lH)e^{-j\omega_k n}, \qquad l = 0, 1, ... \tag{C.4}$$

where H is the hopsize, that is, the time from which the window advances along the signal *x(n)* when computing the spectrum.

For reduced leakage, the window should be zero at the edges and smooth. Windowing is always a trade-off between good frequency resolution and low leakage. These properties are determined by the spectral characteristics of the window, that is, the width of the main lobe and the highest side-lobe level, respectively. A Hanning window is commonly used. In addition, windowing causes the spectral amplitudes to be reduced. This is compensated by dividing by the window sum.

C.2 Signal Operators

Zero-Padding

Zero padding consists of extending a signal (or spectrum) with zeros. This is generally performed in the case of FFT computation which requires the length of the analyzed signal to be a power of two. Zero-padding only increases the apparent resolution but creates more spectral frequency points, which gives a smoother spectrum computation.

Convolution

At first, convolution is a mathematical operation on two functions, producing a third function that is typically viewed as a modified version of one of the original

functions. Convolution is a very important operation in signal processing, an in particular, the output of a LTI system is obtained by convolving the input with the impulse response. Convolution between *x(n)* and *h(n)* is defined as:

$$\sum_{\infty}^{k=-\infty} x(n-k)h(k) = x(n) * h(n) \tag{C.5}$$

Convolution is commutative. We may interpret either x or y as a filter that operates on the other signal which is in turn interpreted as the filter's input signal. In addition, according to the convolution theorem:

$$x(n) * h(n) \leftrightarrow X.H \tag{C.6}$$

where X and H are respectively the Fourier transform of x and h. Thanks to the convolution theorem, FFT allows fast convolution, and the savings compared with direct convolutions, become especially relevant for long signals.

Convolution is similar to cross-correlation. In signal processing, cross-correlation is used to measure the similarity of two waveforms as a function of a time-lag applied to one of them. It is described as a sliding dot product or inner-product. Cross-correlation has applications in pattern recognition.

Spectral Measures

Power spectral density (PSD) is commonly used to describe how a signal is distributed with frequency. The PSD is the Fourier transform of the autocorrelation function, R(τ), of the signal if the signal can be treated as a wide-sense stationary random process. This results in the formula:

$$X(f) = \int_{-\infty}^{\infty} R(\tau) e^{-2\pi\ if\tau} d\tau \tag{C.7}$$

The power of the signal in a given frequency band can be calculated by integrating over positive and negative frequencies:

$$P = \int_{F_1}^{F_2} S(f)df + \int_{-F_2}^{-F_1} S(f)df. \tag{C.8}$$

If the signal is not stationary, then the autocorrelation function must be a function of two variables, so no PSD exists, but similar techniques may be used to estimate a time-varying spectral density.

The spectral flux (used in Chapter 9) has been introduced for onset detections [Bello 2005]. Spectral flux measures the change in magnitude in each frequency bin:

$$SF(n) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} H(|X(n,k)| - |X(n-1,k)|) \tag{C.9}$$

where k and n represent respectively the frequency bin and the time index, and *H(x)* $= \frac{x+|x|}{2}$ is the half-wave rectifier function.

Appendix D

Spectral Modeling Synthesis (SMS)

We summarize the Spectral Modeling Synthesis (SMS) method developed by Serra and Smith [Smith 1987, Serra 1997] that has been used in our approach detailed in Chapter 9, prior to audio grains extraction from continuous contact recordings.

SMS belongs to the spectrum models, which address the sound parametrization at the basilar membrane of the ear, that is, by keeping only spectrum information that is perceive by the ear. SMS represents time-varying spectra as a finite number of stable sinusoids (partials or *deterministic* part) plus noise (residual or *stochastic* part). This method originates from the need to represent noise-like signals, such as transients.

D.1 The Deterministic plus Stochastic Model

The input sound *s(t)* is modeled assuming it to be composed of a deterministic part (limited here to a collection of stable and quasi-sinusoidal components) plus a stochastic part:

$$s(t) = \sum_{R}^{r=1} A_r(t) cos[\theta_r(t)] + e(t) \tag{D.1}$$

where $A_r(t)$ and $\theta_r(t)$ are the instantaneous amplitude and phase of the r^{th} sinusoid, and $e(t)$ is the noise component at time t (in seconds). The instantaneous phase is computed through the integral of the instantaneous frequency $\omega_r(t)$:

$$\theta_r(t) = \int_t^0 \omega_r(\tau) d\tau + \theta_r(0) \tag{D.2}$$

D.2 Description of the Analysis Steps

Figure D.1 gives us an overview of the SMS analysis process. The first step consists in extracting a series of magnitude spectra from the input sound with a STFT computation. The STFT is performed with careful choice of the analysis window in order to control the smoothness of the spectrum and the detectability of different sinusoidal components.

The main peaks in the series of magnitude spectra are identified. By identifying any pitch characteristic, that is, any periodicity, partials tracking is simplified and

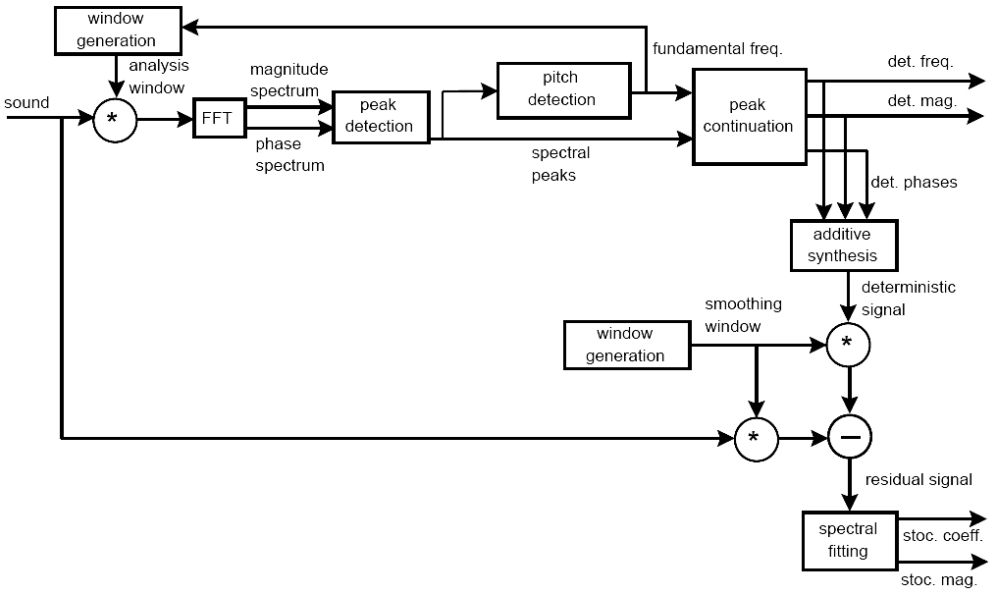

Figure D.1: *Block diagram of the SMS analysis process. Image courtesy of Serra [Serra 1997]*

enhanced. The main peaks are then arranged into frequency trajectories using a peak continuation algorithm which relies on a set of *guides* progressing in time through spectral peaks.

The deterministic part is created as the sum of sine waves (i.e additive synthesis), each resulted from magnitude and frequency trajectories. The STFT of the deterministic part is calculated and each magnitude spectrum is subtracted from the original waveform. A line-segment approximation is applied to deduce the envelope of each *residual* spectrum. The stochastic part is generated from a complex spectra which compiles every spectral envelope of the residual. It is assumed that the residual is fully described by its amplitude and its general frequency characteristics, and the instantaneous phase is not necessary. The stochastic part is described as a filtered white noise, where the filter has a frequency response equal to the spectral envelope:

$$e(t) = \int_{t}^{0} h(t,\tau)u(\tau)d\tau \tag{D.3}$$

where u(τ) is the white noise and h(t,τ) is the impulse response of the filter. A random number generator is appended in the stochastic representation in order to define the phase.

D.3 Modification of the Analysis Data

The benefit of extracting an adequate representation of the input sound is to simplify the manipulation and transformation of the analysis data. The deterministic part is represented by a set of amplitude and frequency functions, whereas the stochastic part is given by a set of spectral envelopes.

Time-scale modifications are performed by modifying the synthesis frame size, producing slow down or speed up while maintaining pitch and formant structure. The separation of the stochastic and the deterministic components give better results than traditional approaches.

In the deterministic representation, a partial is fully defined with its function pair, amplitude and frequency, which can be easily manipulated. The stochastic part representation can be transformed by acting on the shape of the envelopes. Modifying the envelope shape is more obvious than manipulating filter coefficients, for example.

Problems of blending could however arise from the use of two representations where the modifications are unrelated. For this reason, practical experimentation of both the representations is required.

List of Figures

List of Tables

Bibliography

[Adrien 1991] Jean-Marie Adrien. *The missing link: modal synthesis.* Representations of musical signals, pages 269–298, 1991. 19, 54

[Ananthapadmanaban 1982] T. Ananthapadmanaban and Radhakrishnan. V. *An investigation on the role of surface irregularities in the noise spectrum of rolling and sliding contacts.* WEAR, vol. 83, pages 399–409, 1982. 34, 36

[Aramaki 2006a] Mitsuko Aramaki and Richard Kronland-Martinet. *Analysis-Synthesis of Impact Sound by Real Time Dynamic Filtering.* IEEE transactions on Speech and Audio Processing, pages 695–705, 2006. 34

[Aramaki 2006b] Mitsuko Aramaki, Richard Kronland-Martinet, Thierry Voinier and Solvi Ystad. *A Percussive Sound Synthesizer Based on Physical and Perceptual Attributes.* Comput. Music J., vol. 30, no. 2, pages 32–41, 2006. 34

[Avanzini 2001] Federico Avanzini and Davide Rocchesso. *Modeling Collision Sounds: Non-Linear Contact Force.* In In Proc. COST-G6 Conf. Digital Audio Effects (DAFx-01, pages 61–66, 2001. 33

[Avanzini 2002a] F. Avanzini, M. Rath and D. Rocchesso. *Physically-based audio rendering of contact.* In Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on, volume 2, pages 445–448 vol.2, 2002. 32, 94, 119

[Avanzini 2002b] F. Avanzini, D. Rocchesso and S. Serafin. *Modeling Interactions between Rubbed Dry Surfaces Using an Elasto-Plastic Friction Model.* In Digital Audio Effects (DAFx) Conference, volume 2, pages 445–448, 2002. 32

[Avanzini 2005] F. Avanzini, S. Serafin and D. Rocchesso. *Interactive Simulation of Rigid Body Interaction With Friction-Induced Sound Generation.* Speech and Audio Processing, IEEE Transactions on, vol. 13, no. 5, pages 1073–1081, Sept. 2005. 33, 34, 35

[Avanzini 2008] Federico Avanzini. Interactive sound, chapitre 3, pages 83–140. Logos Verlag Berlin GmbH, Berlin, 2008. 20

[Baraff 1998] David Baraff and Andrew Witkin. *Large steps in cloth simulation.* In SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pages 43–54, New York, NY, USA, 1998. ACM. 21

[Bathe 1982] Klaus-Juergen Bathe. Finite element procedures in engineering analysis. Prentice-Hall, New Jersey, 1982. 57

[Beerends 1999] J.G. Beerends and F.E. de Caluwe. *The influence of video quality on perceived audio quality and vice versa.* Journal of the Audio Engineering Society, vol. 47, no. 5, pages 355–362, 1999. 16

[Begault 1994] Durand R. Begault. 3-d sound for virtual reality and multimedia. Academic Press Professional, Inc., San Diego, CA, USA, 1994. 16

[Bello 2005] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies and M. B. Sandler. *A Tutorial on Onset Detection in Music Signals.* Speech and Audio Processing, IEEE Transactions on, vol. 13, no. 5, pages 1035–1047, 2005. 141

[Bilbao 2008] Stefan Bilbao. *A family of conservative finite difference schemes for the dynamical von Karman plate equations.* Numerical Methods for Partial Differential Equations, vol. 24, no. 1, pages 193–216, 2008. 105

[Bonneel 2008] Nicolas Bonneel, George Drettakis, Nicolas Tsingos, Isabelle Viaud-Delmon and Doug James. *Fast modal sounds with scalable frequency-domain synthesis.* In SIGGRAPH '08: ACM SIGGRAPH 2008 papers, pages 1–9, New York, NY, USA, 2008. ACM. 25, 59

[Callan 2004] Daniel E. Callan, Jeffery A. Jones, Kevin Munhall, Christian Kroos, Akiko M. Callan and Eric Vatikiotis-bateson. *Multisensory Integration Sites Identified by Perception of Spatial Wavelet Filtered Visual Speech Gesture Information.* J. Cognitive Neuroscience, vol. 16, no. 5, pages 805–816, 2004. 15, 16

[Camurri 2008] Antonio Camurri, Carlo Drioli, Barbara Mazzarino and Gualtiero Volpe. Sound to sense - sense to sound: A state of the art in sound and music computing, chapitre Controlling Sound with Senses: Multimodal and Cross-Modal Approaches to Control of Interactive Systems, pages 243–278. Logos Verlag, Berlin, may 2008. 16

[Castle 2002] Greg Castle, Matt Adcock and Stephen Barrass. *Integrated Modal and Granular Synthesis of Haptic Tapping and Scratching Sounds.* In Eurohaptics 2002 Conference proceedings Edinburgh, pages 99–102, 2002. 105, 107

[Chadwick 2009] Jeffrey N. Chadwick, Steven S. An, and Doug L. James. *Harmonic Shells: A Practical Nonlinear Sound Model for Near-Rigid Thin Shells.* In ACM Transactions on Graphics (SIGGRAPH ASIA Conference Proceedings). ACM, December 2009. 55, 105

[Chien 1974] Y. Chien. *Pattern classification and scene analysis.* Automatic Control, IEEE Transactions on, vol. 19, no. 4, pages 462–463, Aug 1974. 44

[Cohen 1989] L. Cohen. *Time-Frequency distributions: a review.* Proc. of IEEE, vol. 77, pages 941–979, July 1989. 78

[Cook 1999] P. R. Cook. *Toward Physically-Informed Parametric Synthesis of Sound Effects.* In In Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA-99), pages 1–5, oct 1999. 54, 84, 106

[Cook 2002a] Perry R. Cook. *Modeling Bill's gait: Analysis and parametric synthesis of walking sounds.* In AES 22nd Intl. Conf. Virtual, Synthetic and Entertainment Audio, pages 73–78, Espoo, Finland, 2002. 85

[Cook 2002b] Perry R. Cook. Real sound synthesis for interactive applications. A. K. Peters, 2002. 20, 54, 80, 84, 103, 105

[Corbett 2007] Richard Corbett, Kees van den Doel, John E. Lloyd and Wolfgang Heidrich. *Timbrefields: 3d interactive sound models for real-time audio.* Presence: Teleoper. Virtual Environ., vol. 16, no. 6, pages 643–654, 2007. 58

[Crowell 1998] Benjamin Crowell. *Light and Matter: Vibrations and Waves Textbook*, August 3, 2007 1998. 7, 9

[C.Tomasi 1998] C.Tomasi and R. Manduchi. *Bilateral Filtering for gray and color images.* In Proceedings of the Sixth International Conference on Computer Vision, pages 839–46, New Delhi, India, 1998. 47

[Dixon 2006] S. Dixon. *Onset Detection Revisited.* In Proceedings of the 9th International Conference on Digital Audio Effects (DAFx-09), Montreal, Canada, sept 2006. 90

[Dobashi 2003] Yoshinori Dobashi, Tsuyoshi Yamamoto and Tomoyuki Nishita. *Real-time rendering of aerodynamic sound using sound textures based on computational fluid dynamics.* In SIGGRAPH '03: ACM SIGGRAPH 2003 Papers, pages 732–740, New York, NY, USA, 2003. ACM. 84

[Doel 1998] Kees van den Doel and Dinesh K. Pai. *The Sounds of Physical Shapes.* Presence: Teleoper. Virtual Environ., vol. 7, no. 4, pages 382–395, 1998. 54

[Dubnov 2006] Dubnov. *CATbox: Computer Audition Toolbox in Matlab V0.1*, 2006. online web resource `http://music.ucsd.edu/~sdubnov/ComputerAudition.htm`. 90

[Erkut 2008] Cumhur Erkut, Vesa Välimäki, Matti Karjalainen and Henri Penttinen. Physics-based sound synthesis, chapitre 8, pages 303–343. Logos Verlag Berlin GmbH, Berlin, 2008. 25, 31

[Essl 2005] Georg Essl and Sile O'Modhrain. *Scrubber: an interface for friction-induced sounds.* In NIME '05: Proceedings of the 2005 conference on New interfaces for musical expression, pages 70–75, Singapore, Singapore, 2005. National University of Singapore. 35

[Filatriau 2006] Jehan-Julien Filatriau, Daniel Arfib and Jean-Michel Couturier. *Using Visual Textures for Sonic Textures Production and Control.* In Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06), pages 31–36, Montreal, Quebec, Canada, Sept. 18–20, 2006. http://www.dafx.ca/proceedings/papers/p_031.pdf. 37

[Florens 1991] Jean-Loup Florens and Claude Cadoz. *The physical model: modeling and simulating the instrumental universe.* pages 227–268, 1991. 22, 56

[Gallo 2004] Emmanuel Gallo and Nicolas Tsingos. *Efficient 3D Audio Processing on the GPU.* In Proceedings of the ACM Workshop on General Purpose Computing on Graphics Processors. ACM, August 2004. 59

[Gaver 1993a] William W. Gaver. *How Do We Hear in the World? Explorations in Ecological Acoustics.* Ecological Psychology, vol. 5, no. 4, pages 285–313, 1993. 31

[Gaver 1993b] William W. Gaver. *Synthesizing auditory icons.* In CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems, pages 228–235, New York, NY, USA, 1993. ACM. 20, 25, 115

[Gladden 2005] J. R. Gladden, N. Z. Handzy, A. Belmonte and Emmanuel Villermaux. *Dynamic Buckling and Fragmentation in Brittle Rods.* Physical Review Letters, vol. 94, no. 3, 2005. 118

[Gouyon 2008] Fabien Gouyon, Perfecto Herrera, Emilia Gomez, Pedro Cano, Jordi Bonada, Alex Loscos, Xavier Amatriain and Xavier Serra. Content processing of music audio signals, chapitre 3, pages 83–140. Logos Verlag Berlin GmbH, Berlin, 2008. 77, 79

[Hahn 1998] James K. Hahn, Hesham Fouad, Larry Gritz and Jong Won Lee. *Integrating Sounds and Motions in Virtual Environments.* Presence: Teleoper. Virtual Environ., vol. 7, no. 1, pages 67–77, 1998. 20, 34

[He 2001] Jimin. He and Zhi-Fang. Fu. Modal analysis. Butterworth-Heinemann, Oxford ; Boston :, 2001. 118

[Heigeas 2003] Laure Heigeas, Annie Luciani, Joëlle Thollot and Nicolas Castagné. *A Physically-Based Particle Model of Emergent Crowd Behaviors.* In Graphicon, 2003. 21

[Houben 2004] M. M. J. Houben, A. Kohlrausch and D. J. Hermes. *Perception of the size and speed of rolling balls by sound.* Speech Communication, vol. 43, no. 4, pages 331–345, sept 2004. 36

[Houben 2005] M. M. J. Houben, A. Kohlrausch and D. J. Hermes. *The contribution of spectral and temporal information to the auditory perception of the size and speed of rolling balls.* Acta Acustica, vol. 91, no. 6, pages 1007–1015, 2005. 36

[Huang 2003] G. Huang, D. Metaxas and M. Govindaraj. *Feel the "fabric": an audio-haptic interface.* In SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 52–61, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. 35

[Iovino 1997] Francisco Iovino, René Caussé and Richard Dudas. *Recent work around Modalys and Modal Synthesis.* In ICMC: International Computer Music Conference, pages 356–359, Thessaloniki Hellas, Greece, September 1997. 20, 54

[James 2006] Doug L. James, Jernej Barbic and Dinesh K. Pai. *Precomputed acoustic transfer: output-sensitive, accurate sound generation for geometrically complex vibration sources.* ACM Transactions on Graphics, vol. 25, no. 3, pages 987–995, July 2006. 55, 64

[Keller 1998] D. Keller and B. Truax. *Ecologically-based Granular Synthesis.* In Proceedings of the International Computer Music Conference (ICMC), Ann Arbor, USA, oct 1998. 83, 106

[Kling 2004] Garry Kling and Curtis Roads. *Audio Analysis, Visualization, and Transformation with the Matching Pursuit Algorithm.* In Proceedings of the 7th International Conference on Digital Audio Effects (DAFx-04), pages 33–37, Naples, Italy, 2004. 92

[Kobayashi 2003] R. Kobayashi. *Sound Clustering Synthesis Using Spectral Data.* In Proceedings of the International Computer Music Conference (ICMC), Singapore, 2003. 84

[Kronland-Martinet 1987] R. Kronland-Martinet, J. Morlet and A. Grossman. *Analysis of sound patterns through wavelet transforms.* International Journal of Pattern Recognition and Artificial Intelligence, vol. 1, pages 273–301, 1987. 78

[Lagrange 2001] Mathieu Lagrange, Sylvain Marchand, Scrime Labri and Université Bordeaux. *Real-Time Additive Synthesis Of Sound by Taking Advantage Of Psychoacoustics.* In In Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01, pages 6–8, 2001. 25

[Lazier 2003] Ari Lazier and Perry Cook. *MOSIEVIUS: Feature driven interactive audio mosaicing.* London, UK, September 2003. 85

[Lécuyer 2000] Anatole Lécuyer, Sabine Coquillart, Abderrahmane Kheddar, Paul Richard and Philippe Coiffet. *Pseudo-Haptic Feedback: Can Isometric Input Devices Simulate Force Feedback?* In VR '00: Proceedings of the IEEE Virtual Reality 2000 Conference, page 83, Washington, DC, USA, 2000. IEEE Computer Society. 51

[Lopez 1998] Daniel Lopez, Francesc Marti and Eduard Resina. *Vocem: An Application for Real-Time Granular Synthesis.* In Proceedings of the Digital Audio Effects(DAFx), 1998. 85

[Mallat 1993] S. Mallat and Z. Zhang. *Matching pursuits with time-frequency dictionaries.* IEEE Transactions on Signal Processing, vol. 41, no. 12, pages 3397–3415, 1993. 92

[Maxwell 2007] C. B. Maxwell and D. Bindel. *Modal Parameter Tracking for Shape-Changing Geometric Objects.* In DAFx '07: Proceedings of the 10th International Conference on Digital Audio Effects, 2007. 60, 118

[Maxwell 2008] Cynthia Maxwell. *Sound Synthesis from Shape-Changing Geometric Models.* PhD thesis, EECS Department, University of California, Berkeley, Aug 2008. 118

[McAulay 1986] Robert J. McAulay and Thomas F. Quatieri. *Speech Analysis/synthesis Based on a Sinusoidal Representation.* IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 34, no. 4, pages 744–754, 1986. 82

[McNally 1984] G. McNally. *Variable speed replay of digital audio with constant output sampling rate.* In Proceedings of the 76th AES Convention, 1984. 25, 81

[Menzies 2007] D. Menzies. *Physical Audio for Virtual Environments, Phya in Review.* pages 197–202, Montreal, Canada, 2007. Schulich School of Music, McGill University, Schulich School of Music, McGill University. 70

[Misra 2006] A. Misra, P. R. Cook and G. Wang. *Musical Tapestries: Re-composing Natural Sounds.* In Proceedings of International Compupter Music Conference (ICMC '06), New Orleans, USA, 2006. International Computer Music Association. 85

[Moeck 2007] Thomas Moeck, Nicolas Bonneel, Nicolas Tsingos, George Drettakis, Isabelle Viaud-Delmon and David Alloza. *Progressive perceptual audio rendering of complex scenes.* In I3D '07: Proceedings of the 2007 symposium

on Interactive 3D graphics and games, pages 189–196, New York, NY, USA, 2007. ACM. 59

[Moore 1995] B. Moore. Hearing - handbook of perception and cognition. Academic Press Inc., London, 1995. 78

[Mould 2005] David Mould. *Image-guided fracture*. In GI '05: Proceedings of Graphics Interface 2005, pages 219–226, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society. 114

[Moulines 1989] E. Moulines, F. Charpentier and C. Hamon. *A diphone synthesis system based on time-domain prosodic modifications of speech*. In Proceedings of the International Conference of Acoustics, Speech, and Signal Processing, volume 1, pages 238–241, 1989. 25, 81

[Nealen 2006] Andrew Nealen, Matthias Mueller, Richard Keiser, Eddy Boxerman and Mark Carlson. *Physically Based Deformable Models in Computer Graphics*. Computer Graphics Forum, vol. 25, no. 4, pages 809–836, December 2006. 54

[Nesme 2006] Matthieu Nesme, Yohan Payan and François Faure. *Animating Shapes at Arbitrary Resolution with Non-Uniform Stiffness*. In Eurographics Workshop in Virtual Reality Interaction and Physical Simulation (VRIPHYS), Madrid, nov 2006. Eurographics. 63

[Nesme 2009] Matthieu Nesme, Paul G. Kry, Lenka Jeřábková and François Faure. *Preserving Topology and Elasticity for Embedded Deformable Models*. In ACM Transactions on Graphics (Proc. of SIGGRAPH). ACM, August 2009. 70, 135

[O'Brien 1999] James F. O'Brien and Jessica K. Hodgins. *Graphical modeling and animation of brittle fracture*. In SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pages 137–146, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 113, 117

[O'Brien 2001] James F. O'Brien, Perry R. Cook and Georg Essl. *Synthesizing Sounds From Physically Based Motion*. In Proceedings of ACM SIGGRAPH 2001, pages 529–536, August 2001. 20, 55, 56, 57, 64, 71, 104

[O'Brien 2002a] James F. O'Brien, Adam W. Bargteil and Jessica K. Hodgins. *Graphical modeling and animation of ductile fracture*. In Proceedings of ACM SIGGRAPH 2002, pages 291–294. ACM Press, August 2002. 111, 113

[O'Brien 2002b] James F. O'Brien, Chen Shen and Christine M. Gatchalian. *Synthesizing sounds from rigid-body simulations*. In SCA '02: Proceedings of the

2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 175–181, New York, NY, USA, 2002. ACM. 20, 54, 56, 57, 62, 63, 64

[Oda 2005] Ohan Oda and Stephen Chenney. *Fast dynamic fracture of brittle objects.* In SIGGRAPH '05: ACM SIGGRAPH 2005 Posters, page 113, New York, NY, USA, 2005. ACM. 114

[Otaduy 2005] Miguel A. Otaduy, Nitin Jain, Avneesh Sud and Ming C. Lin. *Haptic display of interaction between textured models.* In SIGGRAPH '05: ACM SIGGRAPH 2005 Courses, page 133, New York, NY, USA, 2005. ACM Press. 51

[P. 1997] Hollier M. P. and Voelcker R. *Objective Performance Assessment: Video Quality as an Influence on Audio Perception.* In The 103rd Audio Engineering Society Convention, New York, USA, sept 1997. 16

[Pai 2001] Dinesh Pai, Kees van den Doel, Doug James, Jochen Lang, John E. Lloyd, Joshua L. Richmond and Som H. Yau. *Scanning Physical Interaction Behavior of 3D Objects.* In Proc. SIGGRAPH'01, pages 87 – 96, August 2001. 32, 33, 42, 58, 94

[Pal 1993] N. R. Pal and S. K. Pal. *A review on image segmentation techniques.* Pattern recognition, vol. 26, no. 9, pages 1277–1294, 1993. 79

[Parker 2009] Eric G. Parker and James F. O'Brien. *Real-Time Deformation and Fracture in a Game Environment.* In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pages 156–166, August 2009. 114

[Pauly 2005] Mark Pauly, Richard Keiser, Bart Adams, Philip Dutré, Markus Gross and Leonidas J. Guibas. *Meshless animation of fracturing solids.* ACM Trans. Graph., vol. 24, no. 3, pages 957–964, 2005. 113

[Peretz 2003] Isabelle Peretz and Max Coltheart. *Modularity of music processing.* Nature Neuroscience, vol. 6, no. 7, pages 688–691, July 2003. 114

[Perlin 1985] Ken Perlin. *An image synthesizer.* In Proc. SIGGRAPH '85, pages 287–296, New York, NY, USA, 1985. ACM Press. 37

[Picard 2008] Cécile Picard, Nicolas Tsingos and François Faure. *Audio texture synthesis for complex contact interactions.* In 5th Workshop On Virtual Reality Interaction and Physical Simulation, VRIPHYS 08, Grenoble, France, November 2008. 3, 41

[Picard 2009a] Cécile Picard, François Faure, George Drettakis and Paul G. Kry. *A Robust And Multi-Scale Modal Analysis For Sound Synthesis.* In Proceedings of the International Conference on Digital Audio Effects, Como, Italy, sept 2009. 3, 61, 71, 117, 133

[Picard 2009b] Cécile Picard, Nicolas Tsingos and François Faure. *Retargetting Example Sounds to Interactive Physics-Driven Animations.* In AES 35th International Conference - Audio for Games, London, UK, 2009. 3, 83, 87, 120

[Puckette 2004] M. Puckette. *Low-dimensional parameter mapping using spectral envelopes.* In ICMC '04: Proceedings of International Compupter Music Conference, Miami, USA, 2004. International Computer Music Association. 84

[Raghuvanshi 2006] Nikunj Raghuvanshi and Ming C. Lin. *Interactive Sound Synthesis for Large Scale Environments.* In SI3D'06: Proceedings of the 2006 symposium on Interactive 3D Graphics and Games, pages 101–108. ACM Press, 2006. 20, 22, 54, 56, 59, 63

[Raghuvanshi 2007] Nikunj Raghuvanshi, Christian Lauterbach, Anish Chandak, Dinesh Manocha and Ming C. Lin. *Real-time sound synthesis and propagation for games.* Commun. ACM, vol. 50, no. 7, pages 66–73, 2007. 17

[Rath 2003a] M. Rath and F. Fontana. The sounding object, chapitre High-level models: bouncing, breaking, rolling, crumpling, pouring, pages 173–204. Mondo Estremo, Firenze, Italy, 2003. 113

[Rath 2003b] Matthias Rath. *An Expressive Real-Time Sound Model of Rolling.* In Digital Audio Effects (DAFx) Conference, 2003. 25, 34, 36

[Rath 2005] Matthias Rath and Davide Rocchesso. *Continuous Sonic Feedback from a Rolling Ball.* IEEE MultiMedia, vol. 12, no. 2, pages 60–69, 2005. 36

[Reck 2005] Eduardo Reck and Miranda Adolfo Maia Junior. *Granular Synthesis of Sounds through Markov Chains with Fuzzy Control.* In Proceedings of International Compupter Music Conference (ICMC05), Barcelona, Spain, Sept. 2005. International Computer Music Association. 84

[Reeves 1983] William T. Reeves. *Particle systems—a technique for modeling a class of fuzzy objects.* In SIGGRAPH '83: Proceedings of the 10th annual conference on Computer graphics and interactive techniques, pages 359–375, New York, NY, USA, 1983. ACM. 21

[Roads 1988] Curtis Roads. *Introduction to Granular Synthesis.* Computer Music J., vol. 12, no. 2, 1988. 83

[Roads 1991] Curtis Roads. *Asynchronous granular synthesis.* Representations of musical signals, pages 143–186, 1991. 83

[Salminen 2002] L. I. Salminen, A. I. Tolvanen and M. J. Alava. *Acoustic Emission from Paper Fracture.* Physical Review Letters, vol. 89, no. 18, pages 185503+, October 2002. 113

[Schwarz 2005] Diemo Schwarz. *Current Research in Concatenative Sound Synthesis.* In International Computer Music Conference (ICMC05), Barcelona, Spain, Sept. 2005. International Computer Music Association. 83

[Schwarz 2006] Diemo Schwarz. *Concatenative Sound Synthesis: The Early Years.* Journal of New Music Research, vol. 35, no. 1, pages 3–22, 2006. 83

[Serra 1989] X. Serra. *A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition.* PhD thesis, Stanford University, 1989. 79

[Serra 1997] X. Serra. Musical signal processing, chapitre Musical Sound Modeling with Sinusoids plus Noise, pages 91–122. Swets and Zeitlinger, 1997. 25, 82, 90, 143, 144

[Shewchuk 1998] Jonathan Richard Shewchuk. *A condition guaranteeing the existence of higher-dimensional constrained Delaunay triangulations.* In SCG '98: Proceedings of the fourteenth annual symposium on Computational geometry, pages 76–85, New York, NY, USA, 1998. ACM. 67

[Slater 2000] Mel Slater and Anthony Steed. *A Virtual Presence Counter.* Presence: Teleoper. Virtual Environ., vol. 9, no. 5, pages 413–434, 2000. 16

[Smith 1987] J. Smith and X. Serra. *PARSHL: An analysis/synthesis program for nonharmonic sounds based on a sinusoidal representation.* In Proceedings Internationtal Computer Music Conference, pages 290–297, 1987. 25, 82, 143

[Stoimenov 2007] Boyko L. Stoimenov, Suguru Maruyama, Koshi Adachi and Koji Kato. *The roughness effect on the frequency of frictional sound.* Tribology International, vol. 40, no. 4, pages 659 – 664, 2007. NORDTRIB 2004. 94

[Storms 2000] Russell L. Storms and Michael J. Zyda. *Interactions in Perceived Quality of Auditory-Visual Displays.* Presence: Teleoper. Virtual Environ., vol. 9, no. 6, pages 557–580, 2000. 16

[Strobl 2006] G. Strobl, G. Eckel and D. Rocchesso. *Sound texture modeling : A survey.* In Proc. of the Sound and Music Computing Conference (SMC'06), Marseille, France, 2006. 37

[Takala 1992] Tapio Takala and James Hahn. *Sound Rendering.* ACM Computer Graphics, SIGGRAPH'92 Proceedings, vol. 28, no. 2, July 1992. 20, 48, 92

[Terzopoulos 1988] D. Terzopoulos and K. Fleischer. *Deformable Models.* The Visual Computer, vol. 4, no. 6, pages 306–331, 1988. 113

[Trebien 2009] Fernando Trebien and Manuel M. Oliveira. *Realistic real-time sound re-synthesis and processing for interactive virtual worlds.* The Visual Computer, vol. 25, no. 5–7, pages 469–477, 2009. 59

[Tsingos 2004] Nicolas Tsingos, Emmanuel Gallo and George Drettakis. *Perceptual audio rendering of complex virtual environments.* ACM Trans. Graph., vol. 23, no. 3, pages 249–258, 2004. 18, 25, 59

[Tsingos 2005] Nicolas Tsingos. *Scalable Perceptual Mixing and Filtering of Audio Signals using an Augmented Spectral Representation.* In Proceedings of the International Conference on Digital Audio Effects, September 2005. Madrid, Spain. 59

[Tzanetakis 1999] G. Tzanetakis and P. Cook. *Multifeature audio segmentation for browsing and annotation.* In Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pages 103–106, 1999. 80

[van den Doel 1996] Kees van den Doel and Dinesh. K. Pai. *Synthesis of shape dependent sounds with physical modeling.* In S. P. Frysinger and G. Kramer, editeurs, Proceedings of the International Conference on Auditory Display (ICAD96), Palo Alto, CA, U.S., 1996. International Community for Auditory Display, International Community for Auditory Display. 20, 49, 107

[van den Doel 1999] Cornelis Pieter van den Doel. *Sound synthesis for virtual reality and computer games.* PhD thesis, 1999. Adviser-Pai, Diresh K. 20, 53

[van den Doel 2001] Kees van den Doel, Paul G. Kry and Dinesh. K. Pai. *Foley Automatic: physically-based sound effects for interactive simulation and animation.* In Proc. SIGGRAPH '01, pages 537–544, New York, NY, USA, 2001. ACM Press. 20, 32, 33, 34, 35, 36, 37, 44, 49, 54, 56, 64, 127, 134

[van den Doel 2002] K. van den Doel, D. K. Pai, T. Adam, L. Kortchmar and K. Pichora-Fuller. *Measurements of the perceptual quality of contact sound models.* Kyoto, Japan, 2002. Advanced Telecommunications Research Institute (ATR), Kyoto, Japan. 25, 58

[van den Doel 2003] Kees van den Doel and Dinesh K. Pai. *Modal Synthesis for Vibrating Objects.* Audio Anecdotes: Tools, Tips, and Techniques for Digital Audio, 2003. 34, 35

[van den Doel 2004] Kees van den Doel, Dave Knott and Dinesh K. Pai. *Interactive simulation of complex audiovisual scenes.* Presence: Teleoper. Virtual Environ., vol. 13, no. 1, pages 99–111, 2004. 25, 51, 58, 59

[Wand 2004] M. Wand and Straßer. *Multi-resolution sound rendering.* In In SPBG'04 Symposium on Point-Based Graphics, pages 3–11, Singapore, 2004. 24

[Warren 1984] William H. Jr Warren and Robert Verbrugge. *Auditory Perception of Breaking And Bouncing Events: A Case Study in Ecological Acoustics.* Experimental Psychology. Human perception and performance, vol. 10, no. 5, pages 704–712, 1984. 32, 112

[Wayman 1989] J. Wayman, R. Reinke and D. Wilson. *High quality speech expansion, compression, and noise filtering using the SOLA method of time scale modification.* In Proceedings of the 23d Asilomar Conference on Signals, Systems and Computers, volume 2, pages 714–717, 1989. 25, 81

[Witkin 2001] Andrew Witkin and David Baraff. *Physiclly Based Modeling.* Online SIGGRAPH Course Notes, 2001. 20

[Young 1987] Richard Young. *The Gaussian derivative model for spatial vision: I. Retinal mechanisms.* Spatial Vision, vol. 2, pages 273–293, 1987. 46

[Zehnder 2009] Alan Zehnder. *Lecture Notes on Fracture Mechanics*, 2009. online web resource `http://ecommons.library.cornell.edu/handle/1813/3075`. 109, 112

[Zhang 1999] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer and Mubarak Shah. *Shape from Shading: A Survey.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 8, pages 690–706, 1999. 42

[Zhang 2005] Qiong Zhang, Lu Ye and Zhigeng Pan. *Physically-Based Sound Synthesis on GPUs.* Entertainment Computing - ICEC 2005, vol. 3711, pages 328–333, 2005. 59

[Zheng 2009] Changxi Zheng and Doug L. James. *Harmonic fluids.* ACM Trans. Graph., vol. 28, no. 3, pages 1–12, 2009. 84

Index

Expressive sound synthesis for animation

Abstract: The main objective of this thesis is to provide tools for an expressive and real-time synthesis of sounds resulting from physical interactions of various objects in a 3D virtual environment. Indeed, these sounds, such as collisions sounds or sounds from continuous interaction between surfaces, are difficult to create in a pre-production process since they are highly dynamic and vary drastically depending on the interaction and objects. To achieve this goal, two approaches are proposed; the first one is based on simulation of physical phenomena responsible for sound production, the second one is based on the processing of a recordings database.

According to a physically based point of view, the sound source is modeled as the combination of an excitation and a resonator. We first present an original technique to model the interaction force for continuous contacts, such as rolling. Visual textures of objects in the environment are reused as a discontinuity map to create audible position-dependent variations during continuous contacts. We then propose a method for a robust and flexible modal analysis to formulate the resonator. Besides allowing to handle a large variety of geometries and proposing a multi-resolution of modal parameters, the technique enables us to solve the problems of coherence between physics simulation and sound synthesis that are frequently encountered in animation.
Following a more empirical approach, we propose an innovative method that consists in bridging the gap between direct playback of audio recordings and physically based synthesis by retargetting audio grains extracted from recordings according to the output of a physics engine. In an off-line analysis task, we automatically segment audio recordings into atomic grains and we represent each original recording as a compact series of audio grains. During interactive animations, the grains are triggered individually or in sequence according to parameters reported from the physics engine and/or user-defined procedures.

Finally, we address fracture events which commonly appear in virtual environments, especially in video games. Because of their complexity that makes a purely physical-based model prohibitively expensive and an empirical approach impracticable for the large variety of micro-events, this thesis opens the discussion on a hybrid model and the possible strategies to combine a physically based approach and an empirical approach. The model aims at appropriately rendering the sound corresponding to the fracture and to each specific sounding sample when material breaks into pieces.

Keywords: Virtual reality, sound modeling, physically based modeling, contact modeling, modal analysis, granular synthesis, adaptive simulation.

Synthèse sonore, réaliste et expressive, pour l'animation

Résumé: L'objectif principal de ce travail est de proposer des outils pour une synthèse en temps-réel, réaliste et expressive, des sons résultant d'interactions physiques entre objets dans une scène virtuelle. De fait, ces effets sonores, à l'exemple des bruits de collisions entre solides ou encore d'interactions continues entre surfaces, ne peuvent être prédéfinis et calculés en phase de pré-production. Dans ce cadre, nous proposons deux approches, la première basée sur une modélisation des phénomènes physiques à l'origine de l'émission sonore, la seconde basée sur le traitement d'enregistrements audio.

Selon une approche physique, la source sonore est traitée comme la combinaison d'une excitation et d'un résonateur. Dans un premier temps, nous présentons une technique originale traduisant la force d'interaction entre surfaces dans le cas de contacts continus, tel que le roulement. Cette technique repose sur l'analyse des textures utilisées pour le rendu graphique des surfaces de la scène virtuelle. Dans un second temps, nous proposons une méthode d'analyse modale robuste et flexible traduisant les vibrations sonores du résonateur. Outre la possibilité de traiter une large variété de géométries et d'offrir une multi-résolution des paramètres modaux, la méthode permet de résoudre le problème de cohérence entre simulation physique et synthèse sonore, problème fréquemment rencontré en animation.
Selon une approche empirique, nous proposons une technique de type granulaire, exprimant la synthèse sonore par un agencement cohérent de particules ou grains sonores. La méthode consiste tout d'abord en un prétraitement d'enregistrements destiné à constituer un matériel sonore sous forme compacte. Ce matériel est ensuite manipulé en temps réel pour, d'une part, une resynthèse complète des enregistrements originaux, et d'autre part, une utilisation flexible en fonction des données reportées par le moteur de simulation et/ou de procédures prédéfinies.

Enfin, l'intérêt est porté sur les sons de fracture, au vu de leur utilisation fréquente dans les environnements virtuels, et en particulier les jeux vidéos. Si la complexité du phénomène rend l'emploi d'un modèle purement physique très coûteux, l'utilisation d'enregistrements est également inadaptée pour la grande variété de micro-événements sonores. Le travail de thèse propose ainsi un modèle hybride et des stratégies possibles afin de combiner une approche physique et une approche empirique. Le modèle ainsi conçu vise à reproduire l'événement sonore de la fracture, de son initiation à la création de micro-débris.

Mots-clés: Réalité virtuelle, synthèse sonore, simulation physique, contact, analyse modale, synthèse granulaire, simulation adaptative.