Real Sound Synthesis for Interactive Applications

Perry R. Cook

A K Peters
Natick, Massachusetts

Editorial, Sales, and Customer Service Office

A K Peters, Ltd.
63 South Avenue
Natick, MA 01760
www.akpeters.com

Library of Congress Cataloging-in-Publication Data

Cook, Perry R.
Real sound synthesis for interactive applications / Perry R. Cook.
p. cm.
Includes bibliographical references and index.
ISBN 1-56881-168-3
1. Sound–Recording and reproducing–Digital techniques. 2. Computer sound processing. 3. Interactive multimedia. I. Title.

TK7881.4 .C666 2002
621.389'3–dc21
2002021546

Printed in the United States of America

06 05 04 03 02 10 9 8 7 6 5 4 3 2 1

Contents

Introduction

If you're like most people, you haven't thought much about the digital synthesis of sound. If you have thought about sound, or even worked with sound on computers, you still might think that sound is something that you record, or get from a CD, or off the Web, and then manipulate to get the desired result. But there are means and methods to truly "synthesize" sound using physical models, by modeling the actual processes that make sound in nature. You might ask, "Isn't synthesizing sound from physics computationally expensive?" Actually, both old and recent theories in acoustics—combined with algorithmic advances in digital signal processing, and of course, faster processors—now allow many types of sound-producing systems to be physically modeled in real time.

It seems reasonable to ask why all of our requirements could not be satisfied by the status quo in "synthesis" based on prerecorded sounds. Also called *Pulse Code Modulation* (PCM), sampling, or wavetable synthesis, much effort has been spent on making sample-based synthesis more expressive. Using multiple recordings, filters, and various interpolation methods to traverse the space of possible sounds, the state of the art has indeed advanced to the point of absolute realism, at least for single sounds triggered only once. To have truly realistic continuously interactive sound synthesis, however, essentially infinite memory would be required to store all possible samples. The equivalent of a truly exhaustive physical model would have to be computed in order to determine which subset of the samples and parameters would be required to generate the correct output sound. Finally, the requirement that the correct samples be loaded and available to the synthesis engine would tax any foreseeable real-time sound hardware/ software systems.

As an example, imagine a “Virtual Reality Wood Shop” or a “Virtual Kitchen” controlled by *Virtual Reality* (VR) controllers such as sensor gloves and haptic (combined senses of touch, position, and movement) force feedback devices. As sound designers, our required task might be to ensure that the various tools each make realistic sounds when manipulated, such that a user cannot detect (by audition) a difference between the real and virtual worlds. A frightening number of recorded samples would be required to realistically and responsively synthesize a wood saw, hammer, wrenches, etc., or whisking, mixing, frying, cutting, and all of the other everyday devices and interactions in these environments. The act of striking an object, then restriking it while it still resonates (hammering a nail into a large board, for example) causes changes in the sound that could only be predicted by an actual physical model of the system in question.

Even without the stringent requirement of sonic perfection, it is still scientifically and artistically interesting to inspect, measure, and simulate the physics of sound-producing objects. The computer graphics and animated movie communities march bravely ahead with more elaborate and realistic models for human, dinosaur, and other motion, and though the results are still not absolutely real, large industries have been built on using graphics models for movie and commercial production.

A staple of production for movies, stage performance, television, and radio dramas is the (often last minute) addition of artificial and natural sound effects. For movies, “Foley” artists put shoes on their hands and “walk” in boxes of gravel, leaves, cornstarch (for the sound of snow), and other materials in real time, effectively acting the sounds as they watch the scenes over and over. Radio and stage sound effects performers/engineers use tape cartridges or CDs of prerecorded sounds, and physical noisemakers to add sounds in real time. For offline production and real-time dramas, these methods might indeed be the best means to add sounds. However, virtual reality, training simulations, and games are real-time computer-mediated interactive applications. The computer, not a human, is responsible for providing the sounds algorithmically, in response to external (sensors) and internal (simulations) variables. Thus, achieving the best sonic quality and responsiveness for such applications is basically impossible by simply playing back prerecorded sounds.

If we think about our interactions with sound-producing objects in the world, we note that we excite these objects with a variety of other objects, and in a wide variety of ways. Walking, preparing food, working with metal and wood tools, and riding a bicycle all create sounds. Our evolution and experience in the world has trained us to expect certain sonic responses from certain input gestures and parameters. For example, hitting something harder causes the sound not only to increase in power, but to change in sound "quality" as well. Rubbing a flat object on its edge causes different sonic results than scraping it in the center. In beginning to develop a taxonomy of interactive sounds, a list of the sound-producing interactions under our control would be helpful. These would include:

- blowing (voice, whistles, wind instruments, etc.)
- striking, plucking, etc.
- rubbing, scraping, stroking, bowing, etc.

That this is such a short list seems surprising, but it is instructive. To arrive at a more subtle classification of sounds requires looking at sound-producing mechanisms as well as the human gestures that initiate and control sound. Of course, that is exactly what this book is about: Looking at the physics of sound production and how we cause objects to make sound. Some objects exhibit strong ringing resonances (metal, glass, tubes, plates, and musical instruments). Others are characterized as noisy, but have too much structure to be modeled by just simple noise. We will develop techniques and tools for analyzing and synthesizing sounds, looking at a wide variety of methods. In the process, we will learn about *Digital Signal Processing* (DSP), *Fourier analysis*, and the general nature of sounds in the world. The goal is to come up with simple, physical (or physically motivated), and parametric models for sound synthesis. We will define a *parametric model* as one that has a few variable parameters that can be manipulated to change the interaction, object, and sound.

In the process of designing and calibrating a parametric model, we often learn a lot about the system, and we also can achieve significant compression (data reduction) over the raw sound itself. Thus, one clear benefit of parametrized sound analysis/synthesis is the possibility of significantly reducing the memory storage required for a large library of digital sound effects. Even greater benefits, however, include the ability to drive the synthesis in real time from parameters generated by sensors, game simulations, animations, and other interactive applications. The flexibility and expressiveness offered by parametric modeling makes it attractive for music composition and performance, simulation in virtual reality, computer games, interactive arts projects, human-computer interfaces, and for other real-time systems where believable sonic responses to user inputs are necessary or desired.

In this book, environmental background sounds (those produced without our own actions and interactions) could be discussed as well. Such sounds would include rain, wind, animal sounds (at least those animals that are not reacting to our presence), trucks and cars, other industrial sounds, and the sounds of humans (again, those not reacting to us). But these sounds do not respond instantly to our actions, and thus have less stringent requirements for simulation. This is not to say that we do not notice if things aren't right in the background sound, and many of the techniques we'll talk about could be directly applicable to background sounds. However, we're not going to deal directly with background sounds here, instead concentrating our focus on interactive sounds.

Thus motivated by scientific, commercial, and artistic interest, and informed by thinking a little about both physical gestures and sound production mechanisms, we will dive into the structure of the book, with some tips on reading it and using the included CD/CDROM to supplement the reading.

Reading Strategy

Since sounds in the real world are made by physical processes, this book will take a physics-based perspective wherever possible. The book is intended for technical, semi-technical and aspiring technical people who know some physics, some programming, and some math. None of these technical skills are specifically required to understand much of what we'll talk about here.

We will present a variety of techniques, ending in Chapter 16 (Examples, Systems, and Applications) with a discussion of practical applications and scenarios. I recommend that you start by scanning Chapter 16, and then come back and start with Chapter 1. I know the world is divided into two types of readers. One type of reader reads the last chapter of any book first, to decide whether to read the book at all, and to break the tension of waiting to see how things come out. The other type of reader views it as almost immoral to learn the surprise before it is time. But for this book, I really do recommend checking out the last chapter, then reading the whole book. The final chapter will first set the stage for reading the book, and will be even more meaningful on the second reading after having learned about sound synthesis.

At the end of every chapter is a short list of recommended reference readings. These lists by no means represent complete sets of references in the various topic areas. Rather, I have tried to glean out the two or so main references on each topic, in case you want to go search and get a better understanding. Perhaps the most exhaustive and math-intensive reference on physical sound modeling is currently being written by Julius O. Smith. This very large book is not available yet, but in a wonderful move toward open publication, his book is on the web as it is being developed. The topics and techniques of Smith's web book are searchable as well, so I would recommend you check out his web book for extra in-depth reading related to many of the topics in this book.

http://www-ccrma.stanford.edu/~jos/waveguide/

Every chapter in my book also has a list of code examples that illustrate and implement the concepts and techniques that are covered. Many of the examples for the book were generated directly from this code. Some of these are simple stand-alone ANSI C programs that run in a command line and generate a sound file. Others are fairly elaborate C++ projects with real-time synthesis controlled by MIDI and/or Graphical User Interfaces. All of this code, plus much more, is included on the CDROM with this book. I encourage you to check out the code while you're reading. Appendix D lists and describes the CDROM code. Appendix E is a short introductory article on the Synthesis Toolkit in C++, a large collection of classes and applications for real-time sound synthesis.

Every chapter also ends with a list of sound examples on the accompanying CD/CDROM. Whenever you're reading along and see an icon like this

(((#))),

that means that there is a sound example on the CD that pertains to the current topic. The first segment of the CDROM is a CD-compatible audio segment, playable on any standard CD player. Of course, the sound files will also show up as audio track numbers on a computer CDROM drive. I encourage you to listen to the sound examples as you read along. Appendix D lists and describes the sound files on the CD.

Appendices A, B, and C present mathematical material not covered in the chapters. You are encouraged to refer to those appendices when they are mentioned in the text.

Chapters 1 and 2 look at PCM sampling and define some terms and fundamentals of dealing with recorded digital sound. In those chapters we will also look at problems and issues with working with prerecorded sounds, to further motivate why we want to dig deeper into the physics of sound production.

Chapter 3 dives into the heart *of Digital Signal Processing* (DSP), covering digital filters. If this chapter bores or scares you, you could skip it, but digital filter theory forms the basis of our efficient physical modeling algorithms, so

I do recommend reading as much of Chapter 3 as you can. You should at least understand the concept of linearity before moving ahead.

With Chapter 4, our development of physical modeling really gets underway. By analyzing and simulating a very simple physical mass/spring/damper system, we get our first look at oscillation and sine waves. Synthesizing sounds by adding together sinusoidal modes is introduced, as our first real parametric synthesis method.

Once we know that sine waves arise out of physical vibrations, Chapter 5 looks more at the math of sine waves, and introduces the powerful tool of Fourier analysis. We will see that the *spectrum* (energy in a sound broken up by different frequencies) is a powerful perception-related tool for analyzing and understanding sounds. Appendix A has proofs, theorems, and some other thoughts on Fourier analysis.

Chapter 6 looks further at using the Fourier analysis spectrum as a sound analysis tool. The techniques of spectral modeling are introduced, breaking up sounds into important perceptual and physical components.

Chapter 7 looks at breaking up the spectrum into fairly wide bands, extracting information (parameters) from those bands, and using those parameters to modify and "sculpt" sounds.

Chapter 8 deals with a topic that combines pure mathematics, digital filters, and physical models. Linear prediction is developed and applied to various physical systems such as the human voice, hammering a nail, and a guitar body.

Chapter 9 looks at one of the most fundamental physical vibration systems, the plucked/struck string. We develop an increasingly more realistic, yet still computationally efficient, series of models of plucked and bowed string instruments. Appendix B has derivations and proofs related to the plucked string.

Chapter 10 looks at a family of phenomena that make sonic life very lively. Nonlinearity (you'll have to read Chapter 3 and 10 to get what this really means) is investigated and modeled.

Chapter 11 looks at sounds produced by air in tubes and cavities. We develop some simple, but amazingly rich, models of a clarinet and a blown pop bottle. Appendix C has derivations and proofs related to acoustic tubes.

Chapter 12 deals with physical modeling of plates, membranes, bowls, cups, and other high-dimensional geometric structures.

Chapter 13 looks at sonic "particles" as the fundamental unit of sound production, examining wavetables, physical particle-controlled sinusoidal synthesis, and noise/filter-based statistical synthesis of noisy sounds.

Chapter 14 deals with plucking, striking, bowing, and rubbing excitations for physical models. Friction is discussed at length. MIDI and other protocols for controlling physical models are discussed. Finally, some custom-built physical modeling synthesis controllers are shown.

Chapter 15 describes a complete signal processing system for analyzing and synthesizing the sounds of walking.

Chapter 16 looks at applications for parametric sound synthesis, including user-interfaces, data sonification, digital Foley, virtual reality, augmented reality, computer music, interactive art, animation, and gaming. The chapter concludes with thoughts on the future of parametric digital sound synthesis.

Acknowledgements

There are scores of people who had major influences on this book being created in the first place, and in it coming to completion. Certainly I must credit and thank the music DSP researchers at Stanford's Center for Computer Research in Music and Acoustics (CCRMA), specifically Julius O. Smith, John Chowning, Chris Chafe, Xavier Serra, Gary Scavone, and lots of others who were around during the exciting development of DSP-based physical modeling. I am grateful for being able to teach the yearly CCRMA summer DSP courses, where I first began to sketch out the chapters and topics for this book. I also thank the students of those courses for asking great questions. I thank Paul Lansky and Ken Steiglitz for recruiting me to Princeton, and getting me to teach the wonderful undergrad Music/DSP course ("Transforming Reality Through Computers") that they had created. I thank the students of Spring 2001 Princeton COS/MUS325 for their reading and use of the preliminary chapters of this book. I thank the researchers at Interval Research for helping support some of my work, and for giving me a wonderful environment in which to hang out for short periods. Specifically I thank Bob Adams, Bill Verplank, Malcolm Slaney, Geoff Smith, Dan Levitan, and many others. For careful reading and criticisms, I thank Georg Essl, Georgos Tzanetakis, Ge Wang, Tae Hong Park, and Liz Douthitt. Thanks to Georg, Georgos, and James O'Brien for recent collaborations on sound synthesis and analysis. Thanks to Gary Scavone for his energy and persistence in pushing the Synthesis ToolKit forward. Thanks to Steve Lakatos for great collaborations on sound perception. Thanks to Dan Trueman and Curtis Bahn for founding and advancing the sensor-enhanced spherical speaker instrument cult. And thanks to many others for long conversations on the topics of digeridoos, dulcimers, and djembes, on prayer bowls, guitars, and accordions, and so many other aspects of the wonderful world of sound.

Development of this book and some of the algorithms described therein was funded in part by gifts from Intel, Interval Research, and the Arial Foundation, and by grants from the National Science Foundation (Grant # 9984087) and The New Jersey Council on Science and Technology (Grant # 01-2042-007-22).