Daniel Möhlmann

A Parametric Sound Object Model for Sound Texture Synthesis

Dissertation

zur Erlangung des Grades eines
Doktors der Ingenieurwissenschaften
— Dr.-Ing. —

Vorgelegt im Fachbereich 3 (Mathematik und Informatik)
der Universität Bremen
im Juni 2011

Gutachter: Prof. Dr. Otthein Herzog
Universität Bremen

Prof. Dr. Jörn Loviscach
Fachhochschule Bielefeld

Abstract

This thesis deals with the analysis and synthesis of sound textures based on parametric sound objects. An overview is provided about the acoustic and perceptual principles of textural acoustic scenes, and technical challenges for analysis and synthesis are considered. Four essential processing steps for sound texture analysis are identified, and existing sound texture systems are reviewed, using the four-step model as a guideline. A theoretical framework for analysis and synthesis is proposed.

A parametric sound object synthesis (PSOS) model is introduced, which is able to describe individual recorded sounds through a fixed set of parameters. The model, which applies to harmonic and noisy sounds, is an extension of spectral modeling and uses spline curves to approximate spectral envelopes, as well as the evolution of parameters over time. In contrast to standard spectral modeling techniques, this representation uses the concept of objects instead of concatenated frames, and it provides a direct mapping between sounds of different length. Methods for automatic and manual conversion are shown.

An evaluation is presented in which the ability of the model to encode a wide range of different sounds has been examined. Although there are aspects of sounds that the model cannot accurately capture, such as polyphony and certain types of fast modulation, the results indicate that high quality synthesis can be achieved for many different acoustic phenomena, including instruments and animal vocalizations. In contrast to many other forms of sound encoding, the parametric model facilitates various techniques of machine learning and intelligent processing, including sound clustering and principal component analysis. Strengths and weaknesses of the proposed method are reviewed, and possibilities for future development are discussed.

Acknowledgements

First and foremost, I wish to thank my advisor, Prof. Dr. Otthein Herzog, for his support, for his advice and encouragement. Our discussions about music, acoustics, methodology and machine learning have always been inspiring for me, and his constructive criticism has helped to sharpen the focus of this work. I also wish to express my gratitude to the Artificial Intelligence Research Group as a whole, for being helpful and generous in many different ways and giving me the opportunity to travel. In particular, I want to acknowledge the help of Dr. Thomas Wagner, for encouraging me to visit conferences and for his advice regarding a number of machine learning and AI issues.

Furthermore, I want to thank my second supervisor, Prof. Dr. Jörn Loviscach, for his support and for his help with some rather complicated technical problems. I was lucky to profit from his extensive knowledge in audio processing and his inventiveness.

I owe my deepest gratitude to the Klaus Tschira Foundation for its generous support, and to the Graduate School Advances in Digital Media for providing the kind of creative environment and the helpful assistance that is invaluable for a work like this.

Finally, I want to thank my parents for their support, and I want to thank Silke Gennies for being so infinitely encouraging and patient during all of this.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Motivation

In audio processing, music and speech have always been at the center of attention (Athineos & Ellis, 2003). Their relevance for communication and entertainment is immediately obvious. However, in recent years a new discipline has developed that directs its attention to the domain of so-called sound textures, the chaotic multitude of atmospheric weather sounds, traffic noises, singing birds and babbling crowds that make up a great deal of everyone's daily life. This research field has been capturing the interest of an increasing number of people working on audio analysis and synthesis.

Sound texture synthesis tries to mimic the acoustic properties of a textural input recording, but with some variation. While this sounds like a simple task at first, it is really very complex, because it requires intelligent processing algorithms to take the input sound apart and construct something new from it. The problem represents a cut through multiple research fields: signal processing, acoustics, machine learning, data mining and perceptual psychology.

Beginning in the mid 1990's, a number of processing algorithms have been proposed to generate textural sounds, often using recordings of actual sounds as an input, like the methods by Saint-Arnaud (1995), by Saint-Arnaud and Popat (1997) or by Bar-Joseph, El-Yaniv, Lischinski, Werman, and Dubnov (1999). The best of these algorithms are able to continuously synthesize rain, applause or ocean waves with some success, but even for such relatively simple sounds, most methods suffer from poor synthesis quality, continuity errors and noticeable repetition. The existing problems of sound texture algorithms have led to a restriction of the domain, rather than leading to more advanced methods, as will be shown later.

The aim of this thesis is to contribute to a better understanding of the theoretical problems in sound texture research, to set new, ambitious goals for this discipline and to start solving them. As a technical contribution, a parametric model for the synthesis of sound objects has been proposed, implemented and tested, which forms the basis for a flexible and completely parametric synthesis framework.

1.1 Goals and Research Questions

Sound texture processing systems like the ones proposed by Saint-Arnaud (1995) or by Bar-Joseph et al. (1999) are designed to perform synthesis by analysis: given an input recording, a model representation of an acoustic scenery must be formed. This is, in most cases, a statistical representation of the signal's properties, rather than an exact encoding of the signal itself.

Creating a sound texture that is similar to an input, but does not repeat it, is an ill-posed problem: thousands of variations could exist that are rated as being equally good, and while certain defects in any artificially synthesized texture may be identified, there might be no way of knowing whether a better variation exists, i.e., whether the defect can be fixed without producing new flaws in other places[1]. One goal of this thesis is to provide a better understanding of the nature of the problem that needs to be solved, starting with the question of what makes a good sound texture synthesis. Another goal is to de-compose the entire analysis-synthesis task into sub-tasks that can be linked to separate theoretical problems.

Once the individual problems are understood, the ability of existing implementations to solve them can be examined more systematically. To enable such a systematic examination, this thesis aims at conceptualizing a general processing paradigm for sound texture analysis and synthesis. A set of abstract processing modules is proposed, where each module solves one problem separately from the other problems.

Many existing sound texture processing algorithms are based on the repetition of the input material (e.g. Saint-Arnaud and Popat (1997), Hoskinson (2002), Dubnov, Bar-Joseph, El-Yaniv, Lischinski, and Werman (2002), Lu, Wenyin, and Zhang (2004), Parker and Behm (2004), Strobl (2007) and Dubnov, Assayag, and Cont (2007)) and thus lack the ability to create any unique variations of sounds. As a method for a more flexible sound synthesis, a parametric synthesis model is presented with the intention to demonstrate its applicability for sound texture synthesis, and to prove the suitability for parametric models in audio analysis-synthesis frameworks in general.

Apart from the technical contributions in the area of parametric synthesis and sound texture frameworks, a further goal of this work is to contribute to the clarification of sound texture theory. This includes a short survey of existing definitions, as well as a discussion of the various research goals in this diverse field.

1.2 Definitions

Before talking about any algorithms for sound texture analysis and synthesis, it is necessary to define what sound textures are, and what kinds of sound sources should be considered "textural" sounds. The term "sound textures" was proposed by Saint-Arnaud (1995). In order to obtain a precise definition of the term, he asked people to give their opinions on what sounds should be called textures. Saint-Arnaud has listed the results of this discussion, including sounds like rain, traffic noise, applause or heart beat. Some other sounds were excluded from the list, such as a single voice, the program of a radio station or a sine wave. The participants were also asked to name general characteristics or dimensions of textural sounds (volume, density, irritability, complexity etc.), and adjectives that could be assigned to individual textures (periodic, dangerous, rough, natural etc.). As mentioned by Saint-Arnaud, the natural-language terms for auditory phenomena do not normally form orthogonal dimensions and their use differs strongly between individual people.

[1] Saying that there *might* not be an optimal solution, or a method to find it, may seem like a vague statement. The reason is that, at this point, we have not yet defined the model for which a solution has to be found. However, optimality is rarely encountered in problems with a strong perceptual dimension.

The characterization of sound textures by Saint-Arnaud and Popat (1997) is an attempt to distinguish between textural and non-textural sounds, and is built upon the notions of similarity, randomness and constant long-term characteristics. It is perhaps the most concise definition available, and has been quoted by many other authors, e.g., Strobl, Eckel, Rocchesso, and le Grazie (2006), X. Zhu and Wyse (2004), Parker and Behm (2004) or Athineos and Ellis (2003), and the main aspects of that definition are also adopted in this thesis. The different aspects of this characterization will be examined in detail in this section.

1.2.1 Definitions of "Texture"

The term "texture" is encountered in the English language in various constellations and with different meanings. Its use for a certain class of sound phenomena is relatively young and was chosen as an analogy from computer graphics textures, which have their meaning from the latin word "textura" (fabric, woven structure). The original term was primarily used to describe the haptic properties of a surface, in particular the surface of textiles, which can be describes as having an either smooth or rough texture (Strobl, 2007).

In computer graphics, the term was then borrowed as a synonym for the visual properties of a virtual surface, and the haptic qualities of virtual materials, which, given the limitations of screen representation, can be conveyed only indirectly by their visual aspects (Foley, 1995). Today, the term is mostly used in computer graphics to refer to image data that is mapped onto virtual surfaces in order to give them the appearance of grass, wood, concrete or the outer hull of a spaceship. This is an efficient way to simplify the description of complex scenes, because it removes the necessity of modeling individual grass plants on a large grass plain, or individual metal plates, screws and paint works of a spaceship hull.

To avoid confusion, it should be noted here that the term sound texture is sometimes used in the context of musical structure, where it has a different meaning: in music theory, it relates to aspects of the composition and instrumentation (Barrington, Chan, & Lanckriet, 2009).

1.2.2 Definitions of "Sound Texture" by Example

Since the concept of sound textures is not immediately intuitive, many researchers sketch out a rough definition by naming examples for sounds that have a "textural" character, and sounds that do not. In fact, the approach of definition by example is one of the dominant strategies across sound texture publications, and is sometimes the closest thing to a definition that is given[2]. Saint-Arnaud and Popat (1997, p. 293) have used the following examples:

> "Most people will agree that the noise of a fan is a likely 'sound texture.' Some other people would say that a fan is too bland, that it is only a noise. The sound of rain, or of a crowd are perhaps better textures. But few will say that one voice makes a texture."

Furthermore, they list the sounds of copy machines, fish tank bubbling, waterfalls, wind, waves and applause, but explicitly exclude a single voice and a ringing telephone

[2] For an overview of definitions and quotes related to "sound texture", see appendix A.

— the latter apparently because it is a singular sound effect, rather than a continuous texture. Dubnov et al. (2002, p. 38) have given mostly similar examples:

> "Natural and artificial sounds such as rain, waterfall, fire, traffic noises, people babble, machine noises and etc., can be regarded as such textures."

Other sounds added to the list of positive examples include thunderstorms, running water, footsteps, typewriters, air conditioners, sawing, breathing, motors, chirping birds, sea gulls, crickets, humming and traffic [(Misra, Wang, & Cook, 2009) (Di Scipio, 1999) (Athineos & Ellis, 2003)]. The examples by Lu et al. (2004), who list background music, lullabies, game music and screen saver sounds, are less typical. Strobl et al. (2006) lists a number of "activity sounds", like crumpling and clapping, which represent a gentle contradiction to the definitions of other authors, who tend to exclude singular events. Although a definition by example is an intuitive starting point, it is necessary to agree on some more formal definitions for sound textures.

1.2.3 The Principle of Similarity

A central principle that characterizes synthetic sound textures is the principle of similarity: the synthesized texture is generally expected to be a substitute for an original sound, and therefore should be similar to that original. It seems self-understood that making a texture similar to an original is the goal of all sound texture research. Still, this needs to be clarified before starting any discussion about *how* similarity can be achieved. Strobl et al. (2006, p. 64) have named this principle in the form of *comparability*:

> "Repetitions should not be audible and sound textures should be targeted of sounding perceptually 'meaningful' in the sense that the synthesized texture is perceptually comparable to the input texture. In the ideal case, no difference is noticeable so that the generated textures still sound naturally and comprise no artefacts."

Dubnov et al. (2002, p. 38) have mentioned in a similar manner that the textures should "closely resemble the original sound source's sonic impression". By using very general terms, these authors avoid a more precise definition of what similarity means in this context, and how it might be measured. The principle of similarity — as far as it is adopted as a property of sound textures — raises the question how unrealistic, synthetic sounds should be treated. When a sound texture is designed with the goal to create an alien or mysterious sound scape that has never been heard before, it does not strictly resemble anything except maybe an ideal of the texture that the sound engineer has in mind.

1.2.4 The Principle of Randomness

An important property of textural sounds is the randomness of their structures (Saint-Arnaud, 1995). Since most examples for sound textures have a natural or biological origin (e.g., rain or animal sounds), some level or randomness and variation can be expected. Sound textures are often said to be the result of stochastic processes, i.e., processes for which a probability distribution is known, but the precise state of the sound-producing system cannot be predicted. For that reason, sound texture synthesis algorithms include various mechanisms to simulate randomness[3].

It is important to note that randomness is not the absence of all structures. The randomness in natural acoustic scenes is bounded by specific probabilities. While we do not know exactly what sounds a songbird is going to produce, or when, there is still a lot of structure in the way the bird sings, which may be described by statistical measures (Balaban, 1988). Capturing the statistical properties of this randomness is not a trivial task. While both uncorrelated random numbers and perfectly repeating structures are trivial to model, learning actual probability distributions requires a lot more understanding of the signal. The signal has to be observed for a longer time period, until a model can be formed of what parts are subject to random processes, and what high-level statistics can be regarded as stable. Dubnov et al. (2002, p. 38) have included the competing concepts of randomness and coherence as an essential component of their sound texture definition. They define randomness in textures as specifically related to the ordering of elements, which reflects the mechanisms of their algorithm:

> "We can describe sound textures as a set of repeating structural elements (sound grains) subject to some randomness in their time appearance and relative ordering but preserving certain essential temporal coherence and across-scale localization."

A similar characterization of randomness is given by Saint-Arnaud and Popat (1997, p. 294), whose definition is more general than the one given by Dubnov et al. (2002). Randomness appears as an optional property of any sound texture, and the terms structure and randomness are not specifically limited to ordering information:

> "[...] it can have local structure and randomness, but the characteristics of the structure and randomness must remain constant on the large scale."

In the definitions, or rather examples, provided by Saint-Arnaud and Popat (1997, p. 299), no clear line is drawn between local, small scale structures and high-level structures. In particular, they avoid to define any upper bound for the scale of randomness[4]:

> "High level randomness is also acceptable, as long as there are enough occurrences within the attention span to make a good example of the random properties."

Randomness is indeed typical for most sound textures, therefore mechanisms of randomness will naturally be a part of convincing sound texture synthesis algorithms. However, it seems wise to follow the definition of Saint-Arnaud and Popat (1997), which does not rule out sounds with much more regular patterns, as could be expected from machinery or a ticking clock.

[3] For graphical textures, randomness is not always such a core concept. For example, in image tiles of brick walls or wallpapers, a regular, repeating structure may be desired.

[4] There appears to be a mild contradiction in their terminology, since the rest of their definition refers clearly to small scales of "a few seconds" (p. 298).

1.2.5 The Principle of Attention Span

An important element for the definition of sound textures is that a short sample of the texture contains all the necessary information about it. The duration itself varies between different phenomena and ranges from below one second to a minute or more, yet, the important thing to notice is that the duration is always limited. Within the interval of the duration, the elements of the texture can have statistical dependencies. Consider, for example, a recording of a storm with occasional thunder, a particular position t within the recording and a time offset t_Δ: If the recording contains loud thunder at t, a portion at $t+t_\Delta$ is likely to have parts of the same thunder in it, provided that t_Δ is a small time interval. Our experience of thunderstorms might also tell us that it is very unlikely that a second thunder will follow right after the first thunder. As t_Δ increases and we move further away from t, the statistical correspondence gets weaker, and so does our ability to predict what the recording will contain at $t + t_\Delta$, until, beyond some time interval, there seems to be no correspondence between the two points in time — at least none that a human could make sense of. This has been called the *attention span* of a sound texture. The concept was mentioned by Saint-Arnaud and Popat (1997). (Parker & Behm, 2004, p. 317) use a similar definition:

> “Any small sample of a sound texture should sound very much like, but not identical to, any other small sample. The dominant frequency should not change, nor should any rhythm or timbre.”

The exact length of an attention span is not usually defined, but typical values are in the order of magnitude of one second. For sounds with much longer attention spans, the required analysis algorithms and synthesis models tend to become increasingly complex. Almost all examples of sound textures presented so far are sources with short attention spans.

1.2.6 A Wide and a Narrow Definition of Sound Textures

While most publications about sound textures and related algorithms agree on simple sounds that are textural, for example rain, there is less agreement about more complex and structured sounds. There appears to be a tendency to move in either of two main directions: one group tends to limit the term sound texture to more or less simplistic sounds, for which processing algorithms are already available (e.g., Dubnov et al. (2002) or Athineos and Ellis (2003)). The other group regards sound texture research as a field that is still in its infancy, and usually includes sounds with much more high-level patterns (e.g., Lu et al. (2004) or Misra, Cook, and Wang (2006)). In this thesis, I will refer to these different ways to define sound textures as the *narrow* and the *wide* definition.

The narrow definition usually deals with phenomena like rain, applause, water bubbles or crackling fire. Most of these consist of noise and transient sounds only, and have almost no tonal or harmonic components. In general, sounds consisting only of noise are easier to process than harmonic sounds, because they can be cut and re-arranged without disrupting the continuity of harmonic partials. Rain and applause also have another property that makes them easier to process: their attention span is typically well below one second, and few hierarchical or long-range dependencies exist in their structure. Although the preference for noisy sounds does not follow strictly from most narrow definitions of “sound texture”, it is often implied by the choice of examples.

According to the wide definition, only few restrictions are imposed on the types of textural sounds. Instead, aspects like "no information", "short attention span" or "repetition" are used as a guideline. The wide definition generally includes all examples used in the narrow definition, especially rain, crowd noises and applause. But in the wide definition, even speech can be regarded as a textural sound, at least in the case that the listener cannot understand it. To a European or American listener, spoken Chinese could potentially have a textural character, provided that it is spoken in a continuous, long text. A synthesis algorithm could possibly simulate a sequence of pseudo-speech phonemes that would sound convincing[5] — at least as a background sound (Saint-Arnaud & Popat, 1997).

Just like regular, understandable speech, music is typically excluded from any definition of sound texture, mostly because it serves a very different purpose and has a whole industry of specific processing tools dedicated to it. However, it could be argued that some forms of music exist that fall into the wider definition of texture. The average radio pop song — having lyrics, strongly expressed organizational patterns and a gradient of development from start to end — is certainly not textural. But what about forms of Free Jazz, or percussive music used in spiritual ceremonies in various cultures?

In this thesis, a wide definition of sound textures will be used, and the decision of what textural sounds are will be made strictly according to the definitions given above. While "regular" speech and "regular" music are outside this definition, musical and speech-like sounds are not excluded as such. The following is a list of sounds that, in the sense of this thesis, would be acceptable "textural" sounds:

- **Rain:** Continuous rain, either light or heavy, with mostly noisy characteristics and some details of drops hitting surfaces. No individual sounds stand out from the mix. Attention span: approx. 500 ms.

- **Applause:** Continuous sound of clapping hands from a large audience in a concert hall. Mostly noisy characteristics, with some claps standing out from the recording. No shouting or whistling. Attention span: approx. 1 s.

- **Storm:** Continuous, heavy rain without individual drops, occasional rolling thunder in addition to the rain. Attention span: approx. 20-40 s.

- **Highway:** Different cars and trucks driving by in irregular patterns from left to right, with characteristic Doppler effect resulting from their high speed. Occasional silence between cars. Some cars passing in the opposite direction in slightly lower volume. Attention span: approx. 10-20 s.

The list presented here contains some sound scenes that are difficult to process with current sound texture algorithms. In that respect, the list represents the goals of sound texture research, rather than a showcase of its current achievements.

1.3 Applications for Adaptive Sound Textures

The class of sounds that can be called "textural" is an integral part of everyday acoustic environments. Such sounds are also routinely used in movies, where they are referred to

[5] In movie productions, the sound of unrecognizable speech, which large crowds tend to produce, is called *walla*, see `http://www.filmsound.org/terminology/walla.htm` (last visited: December 1, 2010).

as Foley sounds[6], and they are constantly gaining importance in computer games[7]. But as simple — and sometimes even insignificant — as these sounds may appear, they often involve great effort and expensive recording sessions in their creation. They require a lot of storage space, and they are difficult to change, once they have been recorded and mixed. For new media environments and interactive games, these limitations present a big challenge to programmers and sound engineers. It is obvious that the domain of game and film sound would profit greatly from more flexible and adaptive sound effects and sound textures. In the following sections, applications for adaptive sound texture algorithms are shown.

1.3.1 Applications for Video Gaming

In the production of video games, one tendency is to create photo-realistic game environments and immersive worlds that can be explored by the player with many degrees of freedom. A typical representative of that genre is *Grand Theft Auto IV*, published in 2008 by Rockstar Games[8]. Both the storytelling and the aesthetics of such games resemble the patterns long established in motion pictures.

Since the goal is to create the illusion of a realistic world, various techniques are employed to create diversity in the game. Some of that diversity comes from the manual construction of thousands of buildings and terrain elements, while additional complexity can be added by recombining elements, or allowing the world to change at playing time, which may involve buildings being destroyed or structures being moved around. In addition to that, procedural methods have been proposed that can generate content according to some rough specifications by the designers. An example for this is the automatic generation of thousands of trees in a forest, which would be tedious to model by designing and arranging individual trees.

Parish and Müller (2001) have shown a technique for the procedural generation of city maps that are automatically populated with artificial buildings. Using the technique, it is possible to specify large-scale features of a city, while the algorithm fills in the details. The approach is based on an extension of L-systems, which have been used for modeling plants and trees (Prusinkiewicz & Lindenmayer, 1990). More aspects for the generation of virtual cities have been explored by Greuter, Parker, Stewart, and Leach (2003), with a special focus on the construction of complex shaped floors plans for the individual houses.

There is a conflict between the freedom and flexibility provided by procedural content, and the danger of losing control over the game experience. However, while the wrong placement of a bridge in a 3D gaming world may cause trouble to the player, the random-based simulation of acoustic elements of a virtual thunderstorm is less likely to disrupt the gameplay. Instead, it may help to increase the naturalness of the gaming experience and avoid irritations. Players expect the architecture in a game to stay in the same places, but they do not necessarily expect the same sounds to be played when they turn a corner in the virtual city. Of course, this will happen when simple trigger mechanism without randomization are used in a game to play sound.

[6] See `http://filmsound.org/terminology/foley.htm` (last visited: December 2, 2010).

[7] An interview about the sound design for *Medal of Honor* can be found at `http://www.filmsound.org/game-audio/medal_of_honor.htm` (last visited: December 2, 2010). More interviews regarding this topic are collected on the same website.

[8] `http://www.rockstargames.com/IV` (last visted: December 1, 2010)

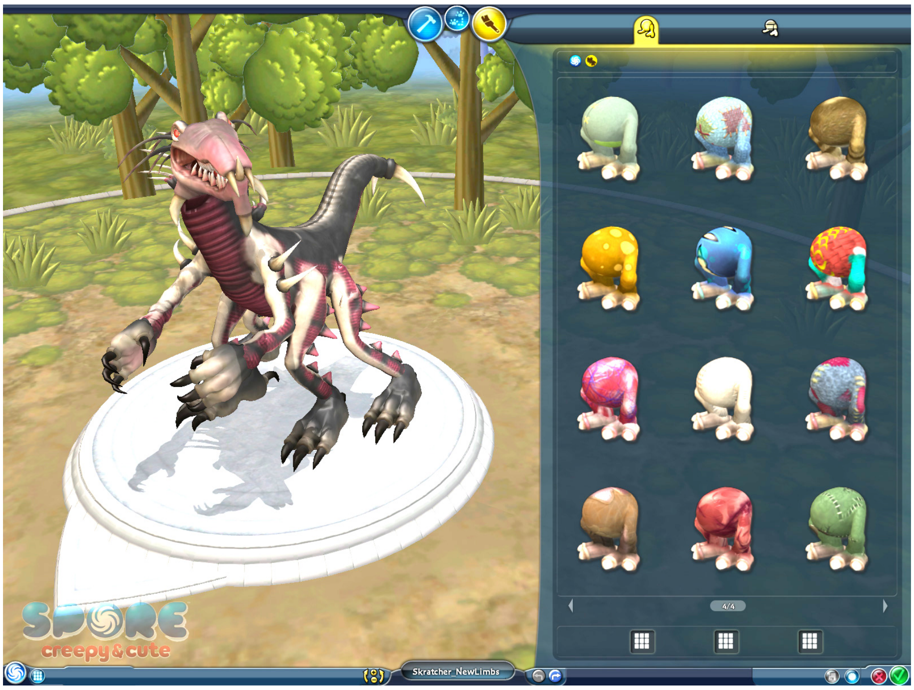


Figure 1.1: Creature design in *Spore*. Motion patterns of the creatures are generated procedurally, based on the distribution of limbs and body mass. Screenshot downloaded from `http://eu.spore.com` (November 29, 2010). © 2009 Electronic Arts Inc.

Today, game designers are already very aware of the necessity to introduce randomness and variability into the acoustic scenery of a computer game (Paul, 2010). While some game studios today rather follow the manual approach, in order to have a maximum of control over their virtual world, other game designers have embraced procedural content generation as a core-concept of their games. For the game *Spore*, published in 2008 by EA Games, procedural content generation was one of the main design concepts. As game designer Will Wright has stated in an interview (Stern, 2008), procedural algorithms drive various aspects of the game, such as the animations of creatures when they move through the game world (see Fig. 1.1). In addition to that, *Spore* also uses procedural elements for its acoustic elements, most notably in the form of procedural music. In the interview, Wright says:

> "It's like your city music is procedural. You can also go in and customize and change and fiddle with it. . . . Depending on what you've put on the creature youve designed, theres a different theme playing. In fact, you're composing the music as youre building the creature."

Randomization and adaptation of sound in a game is most important in those places where a sound is repeated over and over again. For example, great care is taken in many games to introduce some variability into the design of footsteps. In an interview with the magazine Mix (Jackson, 2008), audio designer Mathieu Jeanson talks about the design of footstep sounds in *Assassin's Creed*, published by Ubisoft in 2007, stating that the game contained over 1500 recorded footstep sounds, including variations for different surfaces and motion patterns. Other common game sounds include wind and crowd noises. In film production, the layered, indistinguishable background of voices of a crowd is often referred to as “walla”. It is typically produced by many people who are improvising text[9] (Carlsson, 2010).

As the processing power that is available in gaming consoles is increasing continuously, some of this processing power could be used for advanced sound synthesis in games, which may include anything from physical modeling, particle simulation and reverb calculation to sinusoidal synthesis or sound morphing. Given the enormous effort and money that goes into the recording of hundreds of samples for footsteps and similar sounds, it is reasonable that the adaptive generation of such sounds according to descriptive models will be of increasing importance in the future. For examples, having a compact representation of footsteps, where the aspects of surface roughness, walking speed or character weight can be used as control variables, would be a valuable tool for game sound designers in future games.

1.3.2 Applications for Artistic Expression and Sound Engineering

Playful interaction with sound can also be found in a growing number of multimedia installations and art projects. For example, in the “Magic Carpet” installation by Paradiso, Abler, Hsiao, and Reynolds (1997), sensors on the ground are used to trigger and control musical events. In the installation “Reeds” by Paine (2004), the artist has used an array of sound synthesis algorithms that interact based on mechanisms of chaos in order to create a highly dynamic and engaging sound environment. Countless other examples of (non-musical) sound art can be found, and there are several international conferences dealing with this topic, e.g. the International Conference on Auditory Display (ICAD) or the International Computer Music Conference (ICMC).

Some acoustic installations create virtual spaces within a museum environment, spaces that evoke memories, re-create existing situations or transform them. For such applications, sound can be a central element, e.g. in the mixed reality environment by Hughes, Smith, Stapleton, and Hughes (2004). Using surround sound techniques, the authors aim at creating a more immersive experience that re-creates the ambience of an acoustic environment faithfully, rather than playing back simple sound effects. If the goal is to create an interactive experience, the quality of such an installation could profit from sound synthesis methods that are able to produce truly unique sounds, instead of reacting to a visitor’s presence with pre-recorded material. Provided that such sounds would be of high quality and realism, they could increase the curiosity of people observing the installation. This could help to avoid the common feeling of observers that they have seen the “full repertoire” of a multimedia installation, and that the installation is about to repeat itself.

1.3.3 Applications for Compression

One of the goals for sound texture analysis and synthesis is compression, although the focus is significantly different from regular audio compression methods. While speech and music codecs are designed to transmit or store a perceptually identical version of the input, sound textures are meant to produce a similar, but not identical version. Their purpose is to store an expressive model of an acoustic scenery, with the ability to create countless variations. Since the multitude of possible outputs is not limited by the synthesis model, a compression factor cannot be given. For these reasons, it

[9] People in the crowd are sometimes asked to literally repeat the word “walla”.

would be meaningless to ask whether any particular sound texture analysis-synthesis algorithm is better or worse — compression-wise — than MP3, AAC or any such codec.

Even though the degree of compression cannot be expressed as a factor, it can still be said that the required storage space for a sound texture model would be extremely small, compared to the storage space that would be required to store its synthesized output. What we may call "compression" is really a process of *abstraction*: by substituting the original data with an abstract description of a class of sounds, we are able to produce any number of new instances. Whether an input of 10 megabytes of audio leads to a model that requires 100 kilobytes of storage space, 2 megabytes or even much more than 10 megabytes, is not significant, because it potentially replaces gigabytes of recorded data. This is one of the promises of sound texture research.

Several authors refer to storage space as an important argument for sound textures (e.g., Lu et al. (2004)). However, it seems appropriate to examine this argument in some more detail. Between 1980 and 2009, the storage size of a typical hard disk drive has increased roughly exponentially, with costs dropping from approx. 15 $ (US) per gigabyte in 2000 to 0.07 $ per gigabyte in 2009 (Komorowski, 2009). Saving 50% of storage space on an audio file of several minutes length cannot pass as a significant argument: whatever storage limitation was problematic before the year 2000 should barely matter ten years later. This is especially true for production quality audio in the movie industry, where the cost of individual megabytes or gigabytes can be neglected, compared to the salary of a sound engineer or the cost of other technical production equipment. What does matter is the ability to store audio of potentially unlimited length in a model file of limited size.

1.4 Overview of this Thesis

In this chapter, definitions for the term "sound texture" have been given, and the motivation for research in this area has been stated, with a special focus on computer games and interactive applications.

In Chapter 2, the foundations of sound and acoustics and their relevance for the design of digital sound models will be discussed. The chapter provides an overview on what causes objects to produce certain types of sounds, and how these sounds are transmitted and filtered. In the second part of Chapter 2, the perception of sound by the human auditory system is explained, and an overview is provided on relevant aspects of audio psychology.

Chapter 3 gives an overview on different sound coding models for applications like compression, music synthesis and sound manipulation. The chapter explains how different characteristics of the individual models affect their aptitude for common sound processing problems. In the second half of Chapter 3, the techniques of sinusoidal synthesis and spectral modeling are discussed, because they are essential ingredients of the sound texture model proposed in this thesis.

In Chapter 4, an overview on existing sound texture processing algorithms and related techniques is given. The chapter starts with a side glance on graphical textures, which have inspired much of the sound texture research. After that, existing algorithms for sound texture synthesis are explained, including fully automatic techniques, semi-automatic techniques and frameworks for the manual construction of textures. Chapter 4 concludes with a number of general observations about the state of the art, and provides some arguments in favor of object-based processing.

In Chapter 5, an object-based sound texture description model is introduced, and a workflow concept is described which is suitable for texture analysis on arbitrary input data. The workflow is broken down into four essential steps: *element identification*, *element grouping*, *element variability analysis* and *distribution pattern analysis*.

Chapter 6 describes the implementation of the *parametric sound object synthesis* (PSOS) model, and discusses the implications of the model parameters for its ability to encode various types of sounds. Methods for manual and semi-automatic conversion of recorded sound into parametric objects are shown.

The perceptual quality of the proposed sound element model has been evaluated in an on-line listening experiment. The method and results of the evaluation are shown in Chapter 7. Differences in the ratings of various sound types are discussed and examined with respect to technical properties of the implementation.

Finally, Chapter 8 gives a brief overview on the insights provided by this thesis, and gives an outlook on future research. Apart from the conclusions about the evaluation, some remarks are made about what has been achieved in this work, and how it fits into the greater scheme of sound texture research and general sound modeling.

Chapter 2

Foundations of Sound Production and Perception

The application of sound textures touches the two areas of analysis and synthesis. To form a structured model of any recorded sound or sound mixture, some level of understanding — or at least decomposition — is necessary. In many ways, algorithms that can be applied to this problem have to perform similar tasks as the human hearing system, including frequency measurement, separation and grouping. The research area that integrates these various tasks is *auditory scene analysis (ASA)*. The term was coined by Bregman (1990), who has investigated the complexity of human auditory perception in great detail, and has provided a structured guideline for further research in this area. Bregman has also described the foundations of the so-called *computational auditory scene analysis (CASA)*.

The goal of CASA is an implementation of the essential processing units of human hearing, based on the inputs of only two sound channels. CASA can be seen as an intermediate approach between an imitation of biological processes and mathematical models. In contrast to more specialized sound processing systems, such as speech recognition software, CASA systems are designed to perform sound analysis in largely unconstrained environments (Rosenthal & Okuno, 1998).

Bregman differentiates between two views on the same problem: the *perceptual* view and the *ecological* view. The perceptual view asks what impression is formed in the human auditory system, while the ecological view takes a look at the sources that produce sound. It is tempting to believe that these views should be almost identical, and deal with the same objects, but this is not true: while on the perceptual side, sound is grouped, sorted by relevance, interpreted and linked to other, e.g., visual stimuli, the ecological side is largely unstructured (Bregman, 1990, p. 1).

Consider the sound of a car driving by. To form a model in the "ecological" sense, it would be necessary to integrate all components of the car involved in the production of the sound — or, to be more specific, the thousands of sounds that contribute to the acoustic mix. This would include periodic oscillations from the engine, pressure pulses from the exhaustion pipe, friction noises from the tires, air turbulences from various edges of the car chassis, and countless more phenomena. Yet, on the perceptual side all these things are quickly combined into a much simpler sound stream: the impression of a combined *"vroooooom"* sound, as children imitate it when they play with a toy car.

It seems to occur to humans naturally that a car makes one sound only, not thousands of sounds.

Another example is a loudspeaker: no matter what is played over the speaker — be it a voice, a violin concerto or relaxing nature sounds — from a strictly ecological point of view, the sounds would have to be described as a magnet, pushing a membrane back and forth in the speaker. This is the mechanical truth behind the sound, but almost entirely irrelevant for the perceptual side.

In this chapter, both the ecological and the perceptual side will be discussed in order to provide a better understanding of the domain of acoustics and hearing. These foundations touch a wide range of aspects, including mechanics, biology and psychology.

2.1 Sound Sources and Acoustics

In this section, the mechanical principles of sound production and propagation are introduced, to provide an understanding of the phenomena that digital sound modeling is intended for. They represent the ecological side of acoustics, a side that does not require any knowledge of human hearing physiology or psychology. Sound just "happens", whether anybody is listening or not. A good understanding of the production of sound is useful to conceptualize sound encoding and synthesis — although it is not necessarily required that a computer-implemented synthesis model must have any close resemblance to the actual physical processes.

2.1.1 Sound Wave Propagation

Sound consists of mechanical vibrations, propagated through a medium, such as air or water. The vibrations cause local changes in pressure, which travel away from the source in a radial pattern. In air, the speed of sound is approximately 344 $\frac{m}{s}$, but varies slightly with air pressure and temperature (Luce, 1993). When the frequency of the vibrations is in the range of auditory perception and is sufficiently loud, it can be heard as sound; very low frequency vibrations can sometimes be perceived as tactile stimuli.

The behavior of sound waves is similar to the behavior of light rays in many ways. For example, sound waves are reflected at solid surfaces. Just as for light rays, the angle of incidence equals the angle of reflection (Luce, 1993). Sound that bounces back from a surface is perceived as an echo. Typically, not all sound is reflected, because some of the energy is absorbed by the surface. If a surface is perfectly smooth, the reflections of the sound waves are coherent and all leave the surface at the same angle. If, however, the surface has small irregularities, the reflected sound is scattered into many different directions, thus destroying the coherent wave front. The choice of materials in a room thus determines the acoustic properties just as much as it determines the visual properties. Sound can also be refracted at the border between two media. The angle of refraction then depends on the difference in the speed of sound between one medium and the other (Luce, 1993).

Wave propagation can happen in the form of *transverse* waves or *longitudinal* waves. For a transverse propagation, the displacement happens at a right angle to the direction of propagation. This can be observed in string instruments: as the peak of the wave travels back and forth between the two fixed ends of the string, the string vibrates up and down. The frequency of the traveling wave depends on the string's length, tension and mass (Luce, 1993). In the case of longitudinal waves, the medium is compressed in

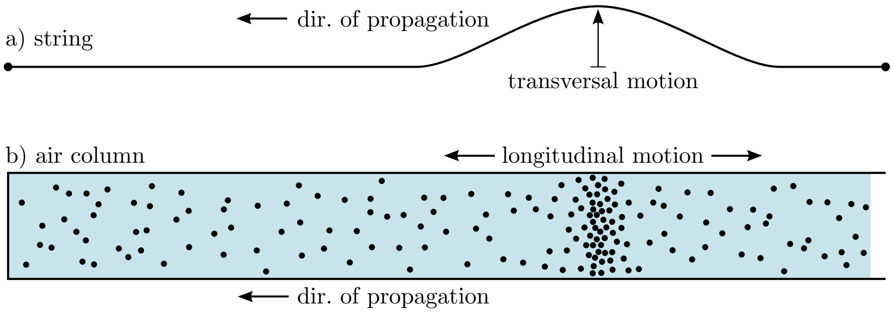


Figure 2.1: Different types of wave propagation: (a) transversal wave propagation, e.g., on a guitar string. (b) longitudinal wave propagation in a vibrating air column.

the direction of the wave propagation. Longitudinal wave propagation can be found in springs and oscillating air columns, like the ones in pipes or flutes (Luce, 1993). The difference of both propagation types is shown in Fig. 2.1.

The pressure waves of multiple sound sources show a linear additive behavior. When two pressure peaks coincide, their amplitudes are added. When the peak of one wave coincides with the valley of another wave, the amplitude is decreased. The sound waves coming from two sources can even cancel each other out completely, so that at one particular location in space, the result is complete silence (Luce, 1993).

On a guitar string, the waves traveling in both directions cause a particular pattern of mutual addition and cancellation, which is called a pattern of *standing waves*. At certain points along the string, peaks of one wave are always met with valleys from an opposite wave, so that the overall displacement of the string at that position is zero. This is called a *node*. A guitar string always has nodes on the two fixed ends. Likewise, in places between two nodes, waveforms have strongly additive behavior, resulting in *antinodes* (Luce, 1993).

2.1.2 Sinusoids and Oscillators

Sound can be treated as a superposition of sinusoids: the theory of *Fourier analysis* states that any function — even a discontinuous function — can be accurately described as a set of superimposed sine and cosine functions, although for discontinuous functions, the number of sinusoids required is infinite (Bracewell, 1989). This decomposition is very relevant to a multitude of natural processes, for which periodic waveforms are often an essential property. For example, the motion of a pendulum can be quite accurately described as a damped sinusoid oscillation. The same is true for oscillating strings of a guitar, or vibrating air columns in a flute. While the Fourier analysis provides an insight about the composition of a sound, *Fourier synthesis* is the reversed process, in which sinusoid functions are combined to re-produce the original signal, or a close approximation of it.

The method used to analyze an incoming signal into its components is the *Fourier transform.* In *digital signal processing (DSP)*, which deals with discrete signals, the *discrete Fourier transform (DFT)*, is particularly relevant, and can be calculated very efficiently using the *fast Fourier transform (FFT)* algorithm (Cooley & Tukey, 1965).

All sinusoids with stationary characteristics can be accurately described by the three parameters *frequency*, *amplitude* and *phase*. The frequency is inversely proportional

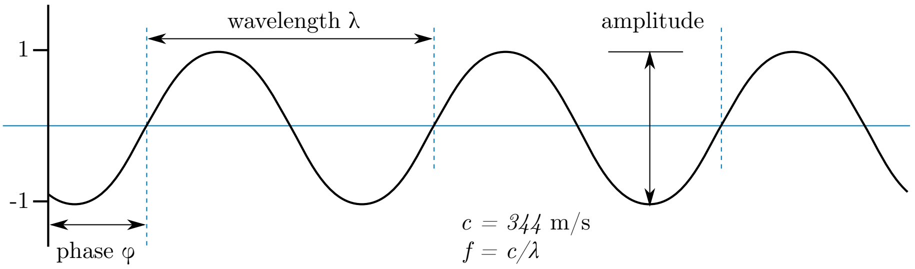


Figure 2.2: A sinusoid is characterized by its frequency, phase and amplitude. The instantaneous phase φ is measured relative to the moment of observation. The frequency f is the sonic speed constant c, divided by the wavelength λ.

to the *wavelength*. Fig. 2.2 illustrates the meaning of the parameters visually. The phase, usually given in the range $-\pi$ to $+\pi$, is the offset of the period with respect to the point of observation. No particular distinction is typically made between sine waves and cosine waves, since the cosine function can be described as a sine function shifted by 90°. The local phase value at any particular point in time is called the *instantaneous phase*. For the purpose of digital processing, the amplitude, which is proportional to the changes in air pressure, is often given on a scale between -1 and $+1$, which corresponds to displacement of a vibrating membrane or string. The *power* of a sinusoid can be computed as the squared amplitude, integrated over time (Luce, 1993).

When the functions of several sinusoids are added, more complex waveforms emerge, which can take arbitrary shapes, including sawtooth, triangle or square waves. The waveform is said to be *periodic* if it has a repeating pattern. The *period size* then gives the amount of time after which the waveform repeats itself. The period size is the lowest common multiple of the wavelengths of all component sinusoids. A device that repeats a given waveform with perfect regularity is called an *oscillator* (Luce, 1993).

A sinusoid may change its frequency or amplitude over time, when energy is lost or when the mechanical properties of the sound-producing system are changed. Although Fourier analysis will always resolve incoming sound into stationary sinusoidal components, the assumption of non-stationary components is often closer to reality. For computer-based implementations, this can be accounted for by replacing the fixed oscillators by flexible oscillators that take time-varying frequency and amplitude values as an input. The rate of change for a time-varying sinusoid typically happens on a much larger time-scale than the individual oscillations. When the frequency or amplitude is changed very rapidly, *modulation* occurs, which is perceived as a change in the sound characteristics (Vercoe, Gardner, & Scheirer, 1998).

Natural oscillators do not normally maintain their oscillation for an infinite time, but lose energy because of damping. For example, the amplitude of a plucked guitar string will decay exponentially after it has been plucked, because no additional energy is introduced into the system. Even though the amplitude decreases, the frequency of the oscillation stays the same (Luce, 1993).

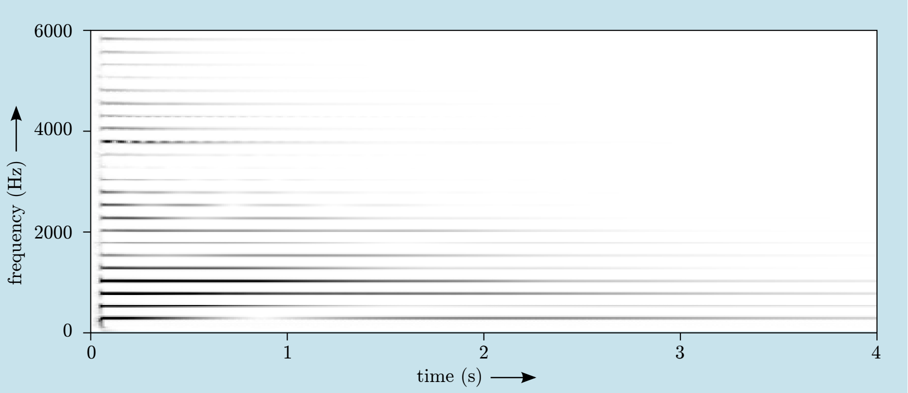


Figure 2.3: Spectrogram of a plucked steel guitar string.

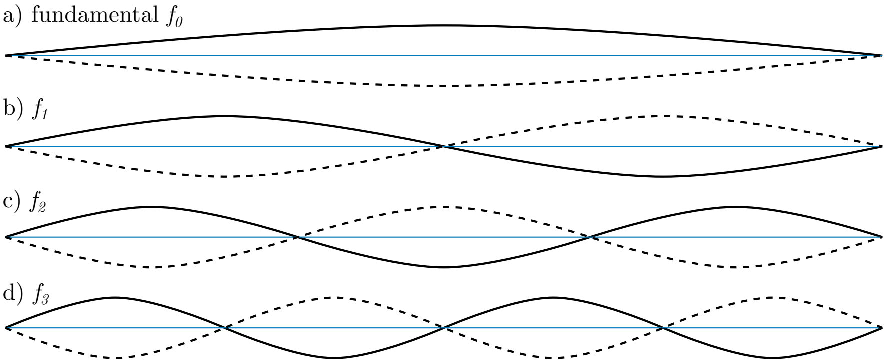


Figure 2.4: Harmonics of a guitar string. (a) the wavelength of the fundamental frequency (f_0) is twice the length of the string. (b) the first harmonic overtone (f_1) has a node in the middle of the string. (c) the second harmonic (f_2) has two nodes. (d) the third harmonic (f_3) has three nodes.

2.1.3 Harmonic Sound Sources

One type of sound that deserves special attention is the *harmonic sound* sound. A harmonic sound source produces a periodic waveform of a particular fundamental frequency f_0. In addition to that, it produces frequencies that are integer multiples of the fundamental, the so-called harmonic overtones or *partials*. The first partial corresponds to an oscillator of the fundamental frequency, the second partial has twice the fundamental frequency. A perfectly harmonic sound has energy at the harmonic frequencies, but no energy at any other frequencies (Luce, 1993).

An example for a harmonic sound is the sound of a plucked guitar string: a string that resonates with a fundamental frequency of 220 Hz will have additional partials at 440 Hz, 660 Hz, 880 Hz, 1100 Hz and so on. The harmonic series is a direct result of the string being fixed on both ends, which forces the vibration to have nodes at the end points, and regularly spaced nodes along the string (Luce, 1993). In a guitar, the energy in the harmonic partials decreases with the frequency, although the exact loudness the partials differs between individual instruments. Fig. 2.3 shows the spectrogram of a plucked guitar string. The groups of harmonic partials are clearly visible, the sound starts with a peak and then slowly fades away. The patterns of standing waves on a string are illustrated in Fig. 2.4.

In a half-open tube, such as an organ pipe, the pressure changes propagate through the tube in longitudinal direction. The excitation happens at the open end, causing it to behave as an antinode. The wave gets reflected at the closed side, which acts as a node. As a consequence, only odd harmonics can arise in a half-open tube. In an open tube, both ends are antinodes. Therefore, although the pattern of nodes and antinodes is reversed, compared to the string example, open tubes can produce both even and odd harmonic overtones. An example for an open tube is a horn (Luce, 1993).

Although the model for harmonic oscillations is accurate enough for most analysis and synthesis problems, objects in the real world show a slightly more complex behavior. The strings of pianos and guitars are not ideal one-dimensional structures, but have some thickness and stiffness, which causes the the frequency of partials to be a bit higher than a perfect integer multiple of the fundamental. As a consequence, the phases of the overtones tend to drift apart, and the shape of the waveform changes gradually. This phenomenon is called *inharmonicity* (Fletcher, 2000). When the influence of inharmonicity is small and the frequencies of partials can be approximately described by the harmonic model, sounds are often called *quasi-harmonic*.

Some instruments produce *mode-locked* signals, which means that all period cycles are identical and the partials always maintain a fixed offset of their phases. For perfect harmonic sounds, in which each partial frequency is an exact integer multiple of the fundamental, this would not normally come as a surprise. However, perfect harmonicity does not occur in the physical world. For example, a violin string cannot normally resonate with perfect harmonics, because it is not infinitely thin and has some stiffness. Yet, when the bow is dragged across the string, the produced sound is clearly mode locked (Fletcher, 2000).

The reason for mode locking in a violin is that the dragging of the bow feeds nonlinear impulses into the mechanical system of the bow and string. The bow “sticks” to the string for a short duration of the period cycle because of the frictional force. Then, the string slips away and the frictional force is reduced immediately. The force and dragging speed applied to the bow have to fall within a certain range in order to allow the mode-locked oscillation to develop. If the force is too low, the inharmonicity takes over and interferes with the impulses from the bow. If the force is too big, the natural oscillation of the string is cut off by the friction and no periodic waveform can develop (Fletcher, 2000). Note that the same violin string will produce inharmonic oscillations when it is plucked, instead of bowed. Nonlinearity is also the reason for mode locking in reed instruments, although the nature of the mechanical system and the origin of the nonlinearity is different (Fletcher, 2000).

2.1.4 Non-Harmonic Vibrations

In bells, metal rods, gongs and other resonating three-dimensional structures, the body itself represents the resonating system. The acoustic properties of such *idiophones* are much more difficult to calculate than those of string instruments, because the system cannot be abstracted by a one-dimensional model. The complex interactions across the resonating surface lead to non-harmonic overtones. Reid (2001) has explained this difficulty in an educational article in “Sound On Sound” with relation to drums. The membrane of a drum is a plane which is (more or less) fixed at the ring on the sides. When set into motion, this membrane will oscillate at non-integer modes, some of which travel radially across the surface. Reid gives the relative frequencies of the first twelve

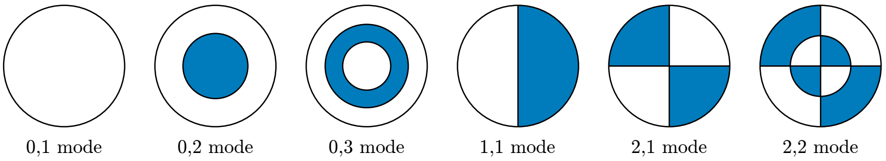


Figure 2.5: The first six vibrational modes of a pitched drum, as given by Reid (2001). The naming of the modes relates to circular vibration and radial vibration, respectively. White areas are raised surfaces, dark areas are lowered.

low-order modes of a Kettle drum as 1.00, 1.59, 2.14, 2.30, 2.65, 2.92, 3.16, 3.50, 3.60, 3.65, 4.06 and 4.15. — which is obviously very different from the familiar ratios 1.00, 2.00, 3.00 etc. that can be expected for string instruments. The first six modes of a pitched drum are shown graphically in Fig. 2.5.

The overtone structure of bells has been the subject of experimentation and scientific evaluation in the past. By shaping the body of the bell, it is possible to tune its fundamental frequency, as well as the presence of vibrational modes. In order to make the sound of bells pleasing to listen to and avoid dissonances, both "minor third" and "major third" bells were developed, depending on the context in which they were to be used (Nigjeh, Trivailo, & McLachlan, 2002). Today, computational models like Finite Element Analysis can be used to optimize the sound of bells (Nigjeh et al., 2002). An overview of different idiophonic instruments, including bells, gongs, marimbas, lithophones and steelpans, is given by Rossing, Yoo, and Morrison (2004).

2.1.5 Subharmonics and Non-Linear Dynamics

In some types of sound-producing systems, two oscillators can be connected that each have a different preferred frequency. Yet, the coupling within the same system may force them to "agree" on a common period, long enough to allow both oscillators to complete their individual cycles. As the period length increases, the frequency is lowered correspondingly, often leading to frequencies that are one half or one third of the frequency of the dominant oscillator. The frequency peaks resulting from this lower octave are called *subharmonics*, and usually carry lower energy than the main harmonics (Fitch, Neubauer, & Herzel, 2002). As shown by Omori, Kojima, Kakani, Slavit, and Blaugrund (1997), subharmonics may occur in the human voice as a pathological effect and are related to perceived roughness.

The interaction of two oscillators can lead to seemingly chaotic behavior, in which the frequency constantly changes between unstable cycles. This phenomenon has been called *deterministic chaos*, to indicate that in fact no randomness is involved: the system still behaves deterministically — although the result is perceptually very similar to a truly noisy process. Deterministic chaos has been shown to exist in infant cries, and also in various mammal vocalizations (Fitch et al., 2002). The role of non-linear phenomena in various animal vocalizations, including barks and screams, has been studied by Tokuda, Riede, Neubauer, Owren, and Herzel (2002). For low-dimensional non-linear systems, the authors have shown a technique for estimating the ratio of non-linear content of a signal by the prediction accuracy of a non-linear model.

Between the normal, harmonic oscillations, the subharmonic oscillation and the deterministic chaos, instant transitions, so-called *bifurcations*, are possible. They can happen when some control parameters of the system, such as the tension of a membrane, exceed a certain threshold, thus pushing the system out of its current, stable state (Fitch et al., 2002).

2.1.6 Pseudo-Periodic Sounds

The term "pseudo-periodic" (sometimes "quasi-periodic") is used to describe sound sources with approximately repeating periods which are subject to some instability or random variation. Pseudo-periodic sounds, according to a definition by Polotti and Evangelista (2000), have narrow noise bands instead of true partials. This model can be used in subtractive synthesis to produce warm sounding instruments like breathy flutes (Reid, 2003). In pseudo-periodic sounds, no two periods are identical, even if the overall impression of pitch is stable. In extreme cases, the perception of noise may dominate the perception of a harmonic tone, as it is often the case with engine noises and machinery: some machines appear to produce a pitched sound, even though no clear sinusoids are detectable in the signal (Strobl, 2007). The term pseudo-periodic has also been used as a generalization of periodic sound types, such as voiced speech, e.g., by Rodet and Depalle (1992), to emphasize that they can be time-varying.

2.1.7 Noise and Turbulence

The term "noise" is used in the domain of acoustics with some ambiguity, and can describe a number of different phenomena. Sometimes it refers to sounds with a hiss-like quality, sometimes to certain irregularities in an otherwise regular sound source, and in other cases to disturbing or loud sounds. Somebody might say that a machine produces noise, when in fact it can be shown to produce perfectly deterministic oscillations. Perceptually, it can be difficult for humans to tell some of these different types of noise apart. Nevertheless, it is helpful at this point to discuss some the physical processes that can cause noise.

For the purpose of this work, the term "noise" will be used with respect to the non-deterministic components of sounds which can be described only by statistical means. For noise, the value of a sample, or a frequency component in a spectrogram, cannot be accurately predicted. However, in many cases a probability distribution can be determined. For *white noise*, the average spectrum is flat over all audible frequencies, i.e., on average all sinusoid components are contained with the same amplitude (Luce, 1993). In white noise, adjacent digital samples are entirely uncorrelated. One method for generating white noise is to use a random number generator with uniform probability distribution.

In order for real-world sounds to be noisy in a mathematical sense, they would have to produce non-deterministic sounds, which means that they would also have to *move* randomly. Since mechanical objects are bound by the principles of cause and effect, their motion cannot really be random. However, it is safe to say that many processes *appear* to be random to a human observer. Noisy sounds are often a combination of a thousands of individual sounds. For example, the sound of wind blowing through the trees is a combined acoustic impression that consists of many simpler sounds, originating from the individual leaves.

Jitter and *shimmer* are two types of degradation of a clean sound that can cause it to sound more noisy (Orlikoff & Kahane, 1991). Jitter refers to variations in fundamental frequency between individual cycles, while shimmer is a rapid variation of the amplitude. Both jitter and shimmer have been linked to the perceptual feature of *roughness*. Algorithms for generating these effects algorithmically in the spectral domain have been proposed by Loscos and Bonada (2004). The term “shimmer” has also been used with a different meaning for the vibration of metal plates, like cymbals, describing the effect that the sound gradually changes and appears to become more chaotic some time after the instrument has been struck (Rossing et al., 2004).

Voiced human speech has a natural amount of jitter, because the pitch is not maintained perfectly between cycle periods. Bregman (1990, p. 540) has noted that synthesized speech without jitter can have an artificial quality.

2.1.8 Transients and Onsets

A special type of noise are the so-called transient sounds (Misra et al., 2006), which can best be described as brief impulses with rapidly changing characteristics. For example, the brief attack noise at the start of a piano note, caused by the hammer hitting the strings, is a transient. The harmonic tone develops a few milliseconds after the noisy impact sound. Other transients include crackling or tapping noises.

Transients can be seen as a special kind of noise phenomena. However, in contrast to noise with more stable characteristics, transients often require specialized algorithms for accurate analysis and synthesis (Verma, Levine, & Meng, 1997). Any analysis that tries to determine the long-term statistical properties of a signal must naturally fail for a transient, since its duration is often much too short to be captured by long analysis windows.

An *onset* is the beginning of a note in a piece of music, for example in the case of the piano note mentioned before. For other instruments and performance styles, the onset may be slower and less well defined in time. A violin, played very softly and slowly, has a very soft onset and no noticeable transient. As a counterpart to the onset, each note has an *offset*, however, the offset of a note is often difficult to determine when it is allowed to decay for a long time.

2.1.9 Resonance and Acoustic Filters

Objects can have preferred *modes of vibration*: when set in motion at a particular frequency, the impulses of the excitation will be amplified, and the object will show *resonance* at this particular frequency. A simple example is a swing that gets pushed at exactly the right moments. Complex shapes will usually have several modes of preferred vibration. They amplify frequencies close to these modes at different amplitudes, while attenuating any other vibrations. The dimensions, mass and shape of an object determine at which frequencies it will resonate. A structure that changes the amplification of frequencies of an excitation force is called an *acoustic filter*, and ranges of frequencies that get amplified by it are called *formants* (Luce, 1993).

Resonance can occur in solid bodies, such as strings, membranes or rigid surfaces, in mechanical systems like springs or pendulums, or in air columns that are set in vibration inside a tube-like structure. The human speech apparatus is such a tube, and its modes of vibration, i.e., its formants, depend upon the length of the tube and the complex shape of the vocal tract, the position of the tongue or the closure of the lips (Wakita, 1973).

In many sound producing objects, a source signal and a filtering structure are integrated within the same mechanical system. This is commonly called the *source-*

filter model (Zölzer et al., 2002). The source-filter model is especially important in the analysis and synthesis of human speech.

2.1.10 Speech as a Special Category of Sound

The human voice is a sound source that is bound by the same acoustic principles as any other source. Still, because of its overwhelming importance for human communication, and because of some particular properties, it useful to take a closer look at human speech production at this point. In sound texture applications, voice-like components are relevant in the case of a sound made by a crowd, and it ca be one of several components in a more complex acoustic environment.

From a mechanical point of view, the vocal tract can be seen as a half-open tube, with the glottis representing the closed end and the open lips representing the open end. In the case of voiced sounds, air from the lungs is forced through the *vocal folds* of the *glottis*, causing them to open periodically. The frequency of voiced speech lies between 75 Hz and 100 Hz for adult men and 150 Hz to 200 Hz for women. The excitation signal produced in the glottis is pulse-like, and therefore contains a rich spectrum of harmonic overtones (Luce, 1993).

Pulses from the glottis travel through the larynx and mouth cavity towards the mouth opening and are reflected to travel back in the opposite direction. For male speakers, the vocal tract has a length of approx. 17 cm, which means that the lowest resonance frequency, i.e., the lowest formant, in the half-open tube can have a wavelength of $4 \cdot 17\,\mathrm{cm} = 68\,\mathrm{cm}$, given a speed of sound of approx. $344\,\frac{\mathrm{m}}{\mathrm{s}}$, which corresponds to approx. 505 Hz[1]. The cross section of the tube varies with the shape of the vocal tract and is further complicated by the nasal cavity. The vocal tract thus acts as a filter for the source signals (Luce, 1993). Fig. 2.6 shows a cross section of the human speech apparatus, according to Carr (1999). In spite of the complexity of real vocal tracts, a much simpler approximation as a tube with segments of different diameters has proven to be useful for computer-based simulations of human speech (Brookes & Loke, 1999).

Bregman (1990) describes two different physical mechanisms by which the amplitude of individual harmonics can be changed. On the one side, the vibrating vocal folds have physical properties that cause harmonic partials to be generated, similar to the vibrating string of a guitar. The relative amplitudes of the partials will be characteristic for any particular glottis, and they remain characteristic even when the fundamental frequency is shifted. The second mechanism is the filtering because of resonances in the mouth and nasal cavities, which can be seen as a separate stage that happens after the sound has been generated.

The above description of the vibrating vocal folds is true in the case of voiced speech. When the glottis is in a relaxed state, air is allowed to pass without vibration, so that no voiced signal is produced. Although the distinction between voiced and unvoiced speech is often sufficient in speech coding applications, the variety of speech utterances and phonemes is much greater in reality. The most widely used classification scheme for speech sounds is the *international phonetic alphabet (IPA)*, which also assigns unique glyphs to each phoneme (International Phonetic Association, 1999).

[1] Note that 505 Hz is not the fundamental frequency of the speech, but the first formant frequency at which oscillations are significantly amplified.

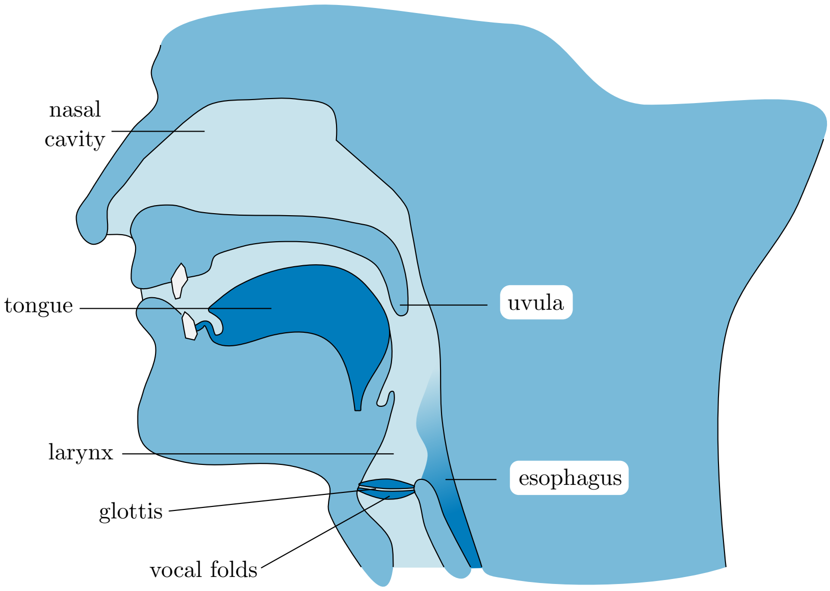


Figure 2.6: Cross-section of the human vocal tract.

Speech phonemes can be distinguished primarily into vowels and consonants. The consonants are further classified according to the way they are produced. The first major distinction among the consonants is whether they are *pulmonic*, i.e., based on airflow from the lungs, or *non-pulmonic*. Pulmonic phonemes are further sorted according to the regions in the mouth where they occur. They can be *bilabial* (produced at the lips), *labiodental* (produced with lips and teeth), *dental* (produced at the teeth), *alveolar* or *postalveolar* (produced in various regions behind the teeth), *retroflex* (produced with the tongue bent backwards and touching the upper roof of the oral cavity), *palatal* (produced with the tongue touching the upper palate of the mouth), *velar* (produced at the back of the tongue), *uvular* (produced with the back of the tongue and the uvula), *pharyngeal* (produced with the root of the tongue against the pharynx) or *glottal* (produced in the vocal folds of the glottis). While this classification relates to the place where sounds are produced, a second sorting criterion for pulmonic consonants is provided by the respective mechanism of sound production. The IPA distinguishes *plosives*, *nasal sounds*, *trills*, *taps* or *flaps*, *fricatives*, *lateral fricatives*, *approximants*, and *lateral approximants* (International Phonetic Association, 1999).

The vowels are arranged in a similar two-dimensional scheme: one dimension indicates how much the mouth is open or closed, the other dimensions specifies whether the resonance occurs at the front, center or the back of the mouth (International Phonetic Association, 1999).

Different languages use different subset of the phonetic alphabet, and some of the phonemes are difficult to produce for non-native speakers. The phonetic alphabet specifies all phonemes in their idealized form, however, many variations to these phonemes can occur in different languages and among different speakers. Still, the standardization of the phonetic alphabet is very helpful for specifying the pronunciation of words in foreign languages, and it allows for a much more detailed sound specification than the letters of the latin alphabet, which can sound very different depending on the context.

Speech sounds have a tendency to produce a smooth spectrogram. Each sound is produced by a certain configuration of the speech apparatus and a certain shape of the mouth cavities. Therefore, instant changes between certain phonemes are not possible. However, the degree of continuity varies for different consonants. While all sounds that have passed through the same sections of the speech apparatus are filtered in the same way (like "a" and "e"), other sounds are produced at the front of the mouth (like "f") and are almost independent with regard to their spectral characteristics (Bregman, 1990, pp. 543ff.).

While speech is described here as a purely acoustic and mechanical phenomenon, this barely explains the importance of speech for human communication. Still, all of the the higher concepts of communication, like syntax, semantics, intonation or emotional expression, still rely on the same mechanical principles for transmission.

2.1.11 Animal Vocalizations

Apart from the human voice, different animal vocalizations have been studied in order to determine their role in animal communication, but also their sound producing mechanisms, which are often entirely different from human speech.

Fletcher (1988) has developed an anatomic and acoustic model of birds' vocalizations, which gives explanations for harmonic oscillations, phase locking effects and formant structures. However, he has pointed out that the model only applies to "screeched", not "whistled" vocalizations. According to Fletcher, whistled vocalizations are the product of airflow phenomena — just as in human whistling — and do not require moving surfaces in the model. Where humans have one set of vocal folds in their larynx, birds have a *syrinx*, which in many cases has two independent membranes sitting in the two bronchi. This allows certain birds two produce two different parallel tones in their songs (Fletcher, 1988). Fee, Shraiman, Pesaran, and Mitra (1998) have analyzed the complex properties of the vocalizations of songbirds, and have shown that patterns of temporal organization emerge beyond the level of neural control, due to the anatomy of their sound-producing organ, the syrinx. A computational model of the syrinx was attempted by Kahrs and Avanzini (2001) with some success. However, there is still some controversy about the exact mechanisms that are at work in bird vocalizations: Tokuda et al. (2002) have pointed out that many classifications schemes used earlier did not take subtle differences in the sound production mechanism into account, and instead relied on — possibly misleading — aspects of spectral similarity.

The nature of other animal calls has been studied by Fitch et al. (2002), with a special emphasis on non-linear phenomena and chaos. The research indicates that — as in bird songs — complex oscillations can occur simply by the anatomy of the vocalization mechanism, and thus do not require detailed neural control. The authors have studied vocalizations of the rhesus macaque, and have shown that these animals use different types of non-linear disturbances in their communication repertoire.

2.2 Human Hearing

The perceptual side of acoustic phenomena deals with human hearing and any higher levels of sound processing and understanding in the brain. Bregman (1990) has noted in this context that the perceptual side mirrors aspects of the ecological side, because the brain has adapted to the sensory inputs coming from the outside. In this section,

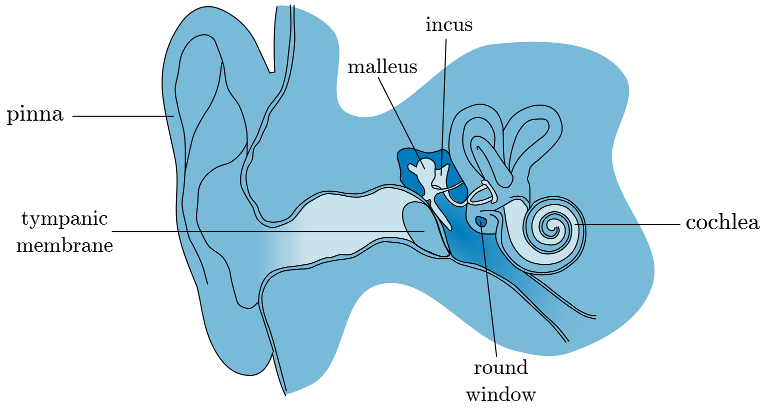


Figure 2.7: Cross-section of the human ear.

first the physiological aspects of hearing will be described. The different strategies used by the brain to decompose and ultimately understand sounds will then be discussed in more detail.

The human auditory system transforms sound at various layers of abstraction, so that the low-level inputs that reach the ear are ultimately transformed into high-level representations with a high degree of semantics and conceptual understanding. Ellis and Rosenthal (1995) remarked that people have “cognitive access” to these high-level representations: they have an understanding of the different sound sources, whether something is music or noise, whether somebody is talking to them and how that relates to them. But while the processes on the low and high level are relatively well understood, there is still little knowledge about the mid-level representations used by the human cognitive system to de-compose and transform the inputs (Ellis & Rosenthal, 1995).

2.2.1 The Physiology of Hearing

The ability of humans to perceive acoustical stimuli is determined in the beginning by the physiology of the outer and inner ear. The most visible outer part of the ear, the ear cup or *pinna*, channels sound pressure waves into the ear canal. Through its complex and irregular shape, the pinna acts as a directional filter, causing sounds from different directions and elevations to have different spectral characteristics (Luce, 1993).

Inside the ear, a conversion takes place from one medium to another at the ear drum. This is necessary because the *cochlea*, the organ containing the hair cells, is filled with liquid, while the acoustic vibration in the ear canal is transmitted through air molecules. Between the two media of different density, an impedance mismatch occurs, which would normally result in a dramatic falloff in amplitude. However, the mechanical structure inside the ear, which includes several tiny bones that are connected to the ear drum and the cochlea, acts as a lever, so that the pressure waves acting on the ear drum are amplified (Luce, 1993). Fig. 2.7 illustrates the ear’s physiology, according to la Cour (1903).

Inside the cochlea, which is shaped like a snail shell, cells on the *basilar membrane* turn the mechanical excitation into electrical signals. The curved shape of the cochlea induces resonances and causes incoming pressure waves to leave characteristic vibratory patterns on the membrane. The so-called *place theory* suggests that all relevant information from a sound is read from the places along the membrane where hair cells are stimulated. In that sense, the cochlea is often said to perform something similar to a Fourier transform. A different theory, called the *temporal theory*, suggests that frequency information is contained in the signal patterns transmitted by individual neurons. The hair cells have neurons that are connected to the *auditory nerve.* Signals coming from the hair cells are transmitted to the brain via *afferent neurons*, but there is also a small number of *efferent neurons* connected in the opposite direction, which transmit signals from the brain to the ear (Luce, 1993).

In the neural pathways from the cochlea to the brain, approx. 32 000 neurons are involved in the transduction of sound impulses to regions of the brain, and countless more are responsible for processing these signals and extracting information. Up to 200 redundant neurons each respond to the same frequency signal, which greatly reduces the response times for detecting certain sounds, even if they are very faint. Tests have also shown that a significant amount of pre-processing occurs in the neural channels of the ear before the signals even reach the brain (Luce, 1993).

Experiments have been conducted to determine to what degree the human ear can distinguish the loudness, or intensity, of two tones. Since intensity is a one-dimensional value, the concept of *just noticeable difference (JND)* can be applied, where ΔI denotes the intensity difference that corresponds to one JND. ΔI is not a fixed constant, but may depend on many factors, most notably on the absolute value of I. Experiments have shown that the ratio $I/\Delta I$ is approximately constant, which is known as Weber's law. However, other experiments have shown that Weber's law is not strictly true in all cases (Luce, 1993).

2.2.2 Loudness Perception

Loudness perception in humans is approximately logarithmic, i.e., a multiplication by a factor corresponds to a perceived increase of loudness by a constant value. Therefore, the logarithmic decibel (dB) scale is usually used when dealing with loudness. The scale of the sound pressure level L_{dB} is relative: it is calculated from the sound's pressure P, divided by the lowest perceivable sound pressure P_0:

$$L_{\text{dB}} = 10 \log_{10} \left(\frac{P}{P_0} \right) \tag{2.1}$$

The dynamic range of human hearing is defined as the ratio between the faintest stimulus that can be perceived and the threshold of pain. Although these values vary between individuals, the average dynamic range is approx. 120 dB. This means that the lowest threshold is only a 1/1 000 000 000 000 fraction of the pain-inducing sound pressure (Moore, 2004).

The perception of loudness depends on several factors besides the amplitude or power of an audio signal. For example, loudness is perceived differently for sounds of different pitch. A 1000 Hz tone with a given amplitude has a particular perceived loudness. At lower or higher pitches, different amplitudes are required to achieve the same impression of loudness. These *equal loudness contours* have been determined at different levels of loudness, and have been found to be approximately the same for most humans with normal hearing (Luce, 1993).

2.2.3 Pitch Perception

The terms *pitch* and *frequency* are not exchangeable: while frequency is an exact physical measure, pitch is the perception of tone height induced by certain stimuli. In theory, a doubling of the frequency should correspond to an increase in perceived pitch by exactly one octave, however, human pitch perception is not linear. Instead, it has aspects of a logarithmic curve. At very high frequencies, above approx. 3000-4000 Hz, pitch perception is significantly compressed, so that much larger frequency increases are required to achieve the same increase in pitch[2]. The ability of humans to detect subtle differences in pitch varies greatly between individuals (Luce, 1993).

To approximate the perceived "melodic" pitch, the *mel* scale has been invented. It behaves almost linear in frequency ranges below 1000 Hz, but shows significantly logarithmic behavior in high frequencies. Slightly different formulas exist for the mel scale. In the most common form, as used by Ganchev, Fakotakis, and Kokkinakis (2005) or Logan (2000), f_{mel} is calculated from the linear frequency f as:

$$f_{\text{mel}} = 1127 \ln\left(1 + \frac{f}{700}\right). \tag{2.2}$$

A study of the perception of simultaneous pitches has been conducted by Marco, McLachlan, and Wilson (2007). The authors have used the term "pitch strength" to describe the certainty with which a pitch is identified by a listener. They have used the term "ambiguous pitch" for "those [stimuli] in which either the number of pitches present or their pitch height is not immediately evident to a large majority of listeners" (Marco et al., 2007, p. 91), which is often true for bell sounds.

The threshold for perceiving sound depends on the frequency and loudness of the source and on the age of the listener, as well as possible damage of the ear, according to the loudness contour mentioned before. Most humans of young age can hear frequencies from approx. 20 Hz to 20 000 Hz (Luce, 1993). There are, however, slightly different numbers to be found in different publications, depending on the definition of "good" hearing and possibly differences in test setups. The ears are more sensitive in the middle frequency range, while near the edges of the audible frequency range sounds need to be very loud in order to be heard. It is worth noting these parameters even when designing a purely automatic detection software, since most electronic equipment for recording, transmitting and storing sound is designed to work with these frequencies and intensities (Luce, 1993).

Some animals have hearing capabilities that are far superior to human ears in many respects, starting with the ability to hear very high or low frequencies. For example, whales, mice and dogs can hear frequencies much higher than 20 000 Hz (ultrasound in human terms), while elephants can hear frequencies below 20 Hz (human infra sound range). Besides conversation, some animals can use sound waves to measure the distance to objects around them. Bats and some whale species are known to use this form of echo localization by sending out short sound waves and receiving the echoes reflected from their environment (Fay & Wilber, 1989). The same is believed to be true for all toothed whales (*odontocetes*) (Ketten, 2000).

[2] This does not mean that an "octave" in the mel scale sounds like a clean sound interval to a human listener. The rules of exact integer multiples still apply. Experiments about the judgment of equal pitch distance are often conducted with non-musical intervals.

The perception of fundamental tones is linked to the structure of harmonic partials (see Subsection 2.1.3). The perception of a fundamental tone is reinforced if matching partials are perceived at the same time. However, the fundamental may also be perceived when only some of the higher partials are present, and not the tone itself. This is called *virtual pitch* (Marco et al., 2007), or the "mystery of the missing fundamental" (Bregman, 1990). The brain is apparently able to infer the fundamental frequency f_0 from the spacing of partials between $3f_0$, $4f_0$, $5f_0$ etc. This is the reason why speech transmitted over a telephone line is still understandable, even if the fundamental frequency is not present in the signal[3] (Bregman, 1990). For some time, it was assumed that the missing fundamental would be "generated" in the ear as a product of nonlinear distortion, which would make it a physical reality, and not something virtual at all. However, it was found later that the fundamental is an illusion induced by higher-level processing in the brain (Luce, 1993). It is difficult to model the human perception of fundamental frequency by measuring the spectrogram peaks, because very subtle changes in the spectrogram can cause an entirely changed perception of fundamental frequency (Luce, 1993).

The combined perception of partials as a single phenomenon with one fundamental frequency appears to be a very strong principle in human hearing. To some degree, it is possible to pay attention to individual harmonic components and hear them as standing out from the harmonic mix, for example by presenting the harmonic in isolation before the whole tone is played. Yet, the fusion principle usually dominates perception and prevents individual partials from being assigned incorrectly when musical tones are played in succession (Bregman, 1990, p. 338).

Beating is an interference phenomenon that occurs between two very similar frequencies. Instead of being resolved into different sources, the tones interact in a regular pattern of mutual cancellation and amplification, which gives the impression of regular pulses, or "beats". The frequency of the beating equals the difference between the two source frequencies. Bregman (1990, p. 504) has pointed out that inharmonic relationships and beating between frequencies are only strongly perceived as conflicting when they occur in the same auditory stream.

2.2.4 Critical Bands

Human hearing does not resolve sound stimuli down to arbitrary details. The theory of *critical bands* states that all sounds within a certain frequency range are combined into one neural channel for further processing, especially in the case of loudness perception. For example, the upper and lower boundaries of a noise band cannot be resolved by the ear more exactly than to the level of critical bands, and two tones of different loudness within the same band will lead to only one combined perception of loudness, even though the presence of two separate tones is clearly detected (Luce, 1993)[4].

Roughly speaking, there are more critical bands in the lower frequency regions than in the high frequency regions. The *Bark* scale, named after physicist Heinrich Barkhausen, identifies 24 critical bands and lists the borders between these bands as determined through listening tests (Zwicker, 1961). Table 2.1 lists the upper limits

[3] A typical telephone line transmits frequencies in the range between approx. 300 Hz and 3300 Hz (Reynolds, Zissman, Quatieri, O'Leary, & Carlson, 1995). Human speech has fundamentals roughly between 75 Hz and 200 Hz.

[4] The theory of critical bands does *not* state that all sounds in a given band sound the same. In particular, the ear can resolve pitched tones at a much higher accuracy than the critical bands.

Bark band	border (Hz)	Bark	border (Hz)	Bark	border (Hz)
1	100	9	1080	17	3700
2	200	10	1270	18	4400
3	300	11	1480	19	5300
4	400	12	1720	20	6400
5	510	13	2000	21	7700
6	630	14	2320	22	9500
7	770	15	2700	23	12000
8	920	16	3150	24	15500

Table 2.1: Upper limits of frequency bands on the Bark scale, given in Hertz.

of Bark bands up to 15 500 Hz. The borders and centers of frequency bands were originally listed in this tabular form. Analytical formulas have been proposed to approximate the conversion of any frequency f in Hz into Bark, although such formulas give slightly different values than the tabular form. The following formula was proposed by Traunmüller (1990):

$$\text{Bark} = \left(26.81 \cdot \frac{f}{1960 + f}\right) - 0.53 \tag{2.3}$$

The Bark scale is closely related to the mel scale, even though the two scales are derived from the different stimuli of pitch intervals and noise bands. For implementations of psychoacoustic models, it appears to make little difference which of the two scales is used.

2.2.5 Timbre Perception

Apart from pitch and loudness, there are many other factors that make up the specific sound of instruments, voices and sounding objects. They all contribute to the perception of *timbre*. Although simple adjectives are often assigned to specific timbres ("bright", "thin", "rough" etc.), timbre is a multidimensional property. Early experiments on timbre were conducted by Helmholtz (1913). Bregman (1990) has compared some earlier definitions of timbre, which mostly use the indirect approach of defining what timbre is not. In particular, timbre is neither sound nor pitch. Most researchers agree that this is not a very satisfactory definition Bregman (1990).

Grey (1975) has investigated the complex topic of timbre perception. He was able to identify physical properties that are closely related to the perception of timbre. There have been other attempts to reduce the complex property of timbre to a much smaller number of dimensions, starting from the hypothesis that, even though countless combinations of sound properties can be conceived, not all of them may lead to a unique timbre perception. The "true" dimensions, hidden behind the physical properties have been called "metametric" dimensions by Bregman. Attempts of discovering such dimensions have been made using *multi-dimensional scaling*.

In *multi-dimensional scaling*, items with different perceptual properties are compared and judged to have a certain difference to each other. Even though the units of this difference measure are not known, it is still legitimate to ask human listeners if some sound A is closer to another sound B or to a sound C. If the judgments of proximity are consistent over a large group of participants, a space with a given number of dimensions can be inferred, and the items can be placed in the feature space (Grey, 1977). A problem is that the proximity judgments are not guaranteed to be free of contradictions. Also, many similar experiments in this field have led to different mappings, and most of them are only valid for a certain type of sound, such as instrument sounds (Bregman, 1990).

It is commonly said that the temporal envelope is an important component of timbre (Grey, 1977). The attack sound of an instrument and the way it decays can be very characteristic: when the temporal envelope of a piano sound is changed to remove the attack and decay, it is much harder for a listener to recognize the instrument. This is certainly true, but does not mean that by reconstructing the temporal envelope alone any good representation of the instrument is obtained. McAdams, Winsberg, Donnadieu, Soete, and Krimphoff (1995) have remarked that the more relevant acoustic information about timbre appears to be contained in the spectral characteristics. Also, the temporal envelope is a quite trivial property of most sounds, and an argument could be made in favor of removing it from the definition of timbre.

In the area of perceptual psychology and music research, the term "timbre" is the subject of ongoing discussion. Grey (1975, p. 1) has noted with respect to this:

> "In the psychoacoustical literature there is also no firm agreement on the meaning of this term with respect to the nature of the various auditory phenomena which should be included in its definition. Even the most quoted definition, approved by the American Standards Association [1960], has given rise to many different interpretations: *Timbre is that attribute of auditory sensation in terms of which a listener can judge that two sounds similarly present and having the same loudness and pitch are dissimilar.*"

Although Grey made these remarks back in 1975, not much has changed, and the "timbre" is frequently accompanied by other terms that are offered as alternatives, including the German word "Klangfarbe" (*tone color*), as used by Helmholtz (1913), or *clangtint*), or simply *color*[5]. Yet, "timbre" seems to be the term used most often when it comes to the description of complex sound properties.

Several challenges arise when writing text about timbre. In order to formalize the scientific debate about sound perception, a common terminology is obviously helpful, if not required. But sound — being a multidimensional concept — eludes traditional vocabulary. Many words exist to describe sound, but their definitions and connotations vary greatly between people, and the terms are almost never orthogonal. A fixed scientific vocabulary of timbre properties would likely suffer from confusion with the casual use of words, unless completely new words were to be invented, which might never be accepted by the community. Also, it is typically very difficult to establish a good correspondence of meanings between different languages.

Terasawa (2009) has proposed the dimensions of *color* and *density* to represent the instantaneous spectral composition and the temporal characteristics, respectively. She motivates her preference of the term "density" over other terms that are sometimes associated with temporal sound characteristics, in particular "texture". She points

[5] I have observed on multiple occasions that, whenever the word timbre was used during conference presentations, it almost always required some form of explanation or further specification.

out that “texture” invokes various connotations that suggest it should not be used for purely temporal features. This seems reasonable, and this thesis, too, uses the word “texture” in a wider context[6].

Most languages, like English, German or French simply do not have enough sound-related words to talk about timbre accurately. As a consequence, there is a danger of over-emphasizing those aspects of timbre for which words do exist. The challenge for any audio researcher is to stay alert with respect to the many relevant dimensions of sound, and to find ways to work with them, even though many of them may not have a name.

2.2.6 Source Localization and Directional Hearing

Humans have directional hearing and are able to locate a sound source in the stereo field with up to two degrees accuracy, depending on a number of factors, such as the loudness, frequency and position of a source relative to the head. The ears can also tell when a sound is coming from behind, above or below. Directional hearing, together with complex processing in the brain, makes it possible to direct the listening attention to certain sounds of interest, even in very noisy environments. It is typically no problem to isolate another persons voice from a multitude of other voices, music and other sounds.

One property that is used by the ear and brain to localize a sound source is the time lag between the sound reaching the left ear and the right ear. This type of localization based on time difference fails for frequencies higher than 2000 Hz, because the shift then gets bigger than one period cycle. Intensity difference is also used: the head attenuates high frequencies, and the resulting difference in loudness is used as a clue for localization. The ear appears to use both principles and switches between them on the transition between low and high frequencies. This is referred to as the *duplex theory*. Around 2000 Hz, neither of both methods can give accurate results. Therefore, source localization is generally poor around this frequency range (Luce, 1993).

2.2.7 Echo Suppression

In most acoustic situations, especially in closed rooms with acoustically reflective surfaces, much echo and reverberation is added to the primary sound waves coming from a source. Since the waves can be reflected from several walls to the same spot, the summed loudness of the reflected waves can easily be larger than the loudness of the direct waves. The result is a very complex mixture of signals that have different time lags. Yet, it appears that the human auditory system has very effective — though little understood — mechanisms to suppress the effect of echo to a high degree (Luce, 1993).

For small rooms, in which the lag between the direct sound and and reflections is less than 20 ms to 30 ms, no echo is perceived at all: the perception of the signal that has traveled the direct path dominates the reflected signal, even if the combined reflections are louder. This is called the *precedence* effect (Luce, 1993). “Real” echoes, where the later impulses are perceived as a distinct, time-shifted version of the original, are only perceived for lags over 500 ms. Reflections that occur on a scale between 30 ms and 500 ms are perceived as *reverberation* or “reverb”. The exact mechanisms by which

[6] It is difficult to find words that are not “contaminated” with a different meaning when it comes to acoustic perception — and “texture” is no exception. In this thesis, the term “sound texture” is used mainly because the analogy to graphical textures is helpful, and because no better alternative comes to mind.

the brain achieves the suppression of echoes is not known. Also, it appears that this suppression works very well in real-world situations, while it is less effective during playback of recorded sounds through loudspeakers (Luce, 1993).

2.2.8 The Perception of Sound Objects

The notion that auditory streams contain individual objects, events or elements can be motivated both from the sound generating processes and from the perception side. Sounds are vibrations of the air, caused by specific events: the collision of rigid bodies or fluids, the releasing of a finger from a guitar string. For a short time the motion of air molecules around the source is dominated by the effect of the released energy, before it is scattered and absorbed. However, from a strictly physical point of view it may be arbitrary to say when any acoustic phenomenon ends, what spatial region it is confined to or how many events are present. A footstep on sand may be looked at as one event, or thousands of events originating from individual sand grains. But even though the question of what constitutes an element may seem philosophical, a practical answer can be found in perception and hearing.

Bregman (1990, p. 221) has asked what causes sound components to be perceived as one single event, and has found that physical laws may be applied "backwards" by the brain in order to identify sound components as having the same origin. Whether any physical laws as such are simulated in the brain is yet unclear, but, according to Bregman, it should be possible for it to "take advantage" of these properties:

> "If a group of components have arisen from the same physical event, they will have relationships between them that are unlikely to have occurred by chance."

Some experiments have been conducted in the past to find out what is necessary to achieve fusion of partials into one harmonic sound. Chowning (1980) has found that partials, when they are added one by one to an acoustic mix, do not fuse well. However, he has found that the fusion can be greatly increased by introducing the same *micromodulation* on all partials..

In music, it is often intended that groups of instruments form a perceptual entity, such as a group of violins playing the same notes. The players have to achieve a high level of synchronization and maintain precisely timed onsets to achieve good fusion. However, it appears that fusion can be achieved easier if a large number of instruments play at the same time, because the blurred onsets of all instruments can be perceived as a combined, smoothed onset (Bregman, 1990, p. 492).

Humans can derive a multitude of physical properties about an object from its sound, such as the dimensions of the object, the material, the shape, or the kind of impact or friction that set the object into vibrating motion. Some researchers have argued that the inference of physical properties from the resulting sound is not only possible, but may be the primary representation used by the brain (Bregman, 1990, pp. 483f.).

Some questions remain, for example if a rise in loudness within a stream causes the perception of a new, distinct object. Many such questions are difficult to answer using laboratory experiments, and the results may be ambiguous (Bregman, 1990, p. 698f.).

2.2.9 Stream Separation and Fusion

Bregman (1990) has described the different stages of auditory processing for human hearing that enable humans to "understand" their acoustic environment, i.e., to direct attention to individual sources, localize those sources in 3D space and extract semantic information from them. A large part of Bregman's auditory scene analysis (ASA) (Bregman, 1990) centers around the principles of streaming, which describes under which circumstances separate sound fuse together perceptually.

The human ability to perform a separation of auditory *streams*, which allows people to understand a single speaker in a room with a babbling crowd, is called the *cocktail party effect* or *cocktail party problem.* While the anatomy of the ear and transmission of signals from the cochlea is well understood, many details about the mid-level auditory processing are still unknown. However, Bregman lists a number of clues that the auditory system apparently uses to achieve sound separation (Bregman, 1990, pp. 529ff.). According to Bregman, the hearing apparatus forms separate streams from the incoming sound mix, based on grouping mechanisms.

Bregman has used the concept of a *perceptual object*, which he uses synonymously with an individual *sound.* Sounds are grouped into *streams*, which Bregman also refers to as "coherent happenings" and "single experienced events"; according to him, "an auditory stream is our perceptual grouping of the parts of the neural spectrogram that go together" (Bregman, 1990, p. 9).

Bregman describes the perception of new sounds against background sounds in terms of an old-against-new heuristic, a concept that was already mentioned by Helmholtz (1913). When a new sound starts, the auditory system assumes that any sound that started earlier will continue to sound with the same characteristics, which can then be subtracted from the new sound. In the case that both sounds start at precisely the same time, it becomes much harder to perceive them as separate sounds. Phase appears to play a role in auditory stream separation. Bregman (1990) has reported that a sudden change in phase can cause a partial to be perceptually segregated from a harmonic sound.

The perceptual grouping of sound is related to visual phenomena from Gestalt psychology. The brain appears to group sound objects that belong together, applying Gestalt laws such as *closure*, *proximity*, *similarity*, *common fate*, *continuity* and *symmetry* (Bregman, 1990).

Loud sounds can mask quiet sounds if they are in the same frequency range. But this masking can also be an illusion: when a quiet sound is interrupted by a louder sound that would be able to mask the quiet sound, the auditory system assumes that the quiet sound continues in the background, even if it is really not there. This is called the *continuity illusion*, and it supports the hypothesis that the brain tries to search for the simplest explanation for any perceptual phenomenon (Bregman, 1990, pp. 345f.). As long as the continuation of the quiet tone behind the louder tone is a possible hypothesis, it will likely be selected from a large pool of other possible hypotheses. In this case, "possible hypothesis" means that the loud, interrupting signal must have sufficient energy in frequencies belonging to the quiet signal (Bregman, 1990, pp. 357ff.).

Bregman has stated that the grouping of sounds in the brain may be weak or ambiguous if the similarity of sounds in different streams is small. He has mentioned a mechanism of competition that "votes" on a sound to belong to either one group or the other, however, the allocation of a sound to a stream may not be strictly exclusive (Bregman, 1990, pp. 170f.).

Two principles have been described so far: fusion and decomposition. The question remains, as Bregman has pointed out, which of the two is actually used by the human auditory system. Does it fuse all sounds by default, i.e., does it assume that everything is one coherent sound, unless there is specific evidence against that? Or does it decompose all sounds by default, maintaining hundreds of separate streams, unless certain auditory clues suggest that they should be grouped? In this regard, Bregman has argued in favor of fusion as the “default” mode (Bregman, 1990, pp. 333f.).

When two tones, an ascending tone and a descending tone, cross frequency trajectories, it is apparently hard for a listener to perceive either of them as a continuous trajectory. Instead, the grouping principles will give the impression that the descending tone turns around in the middle and rises again, because frequencies within the same frequency regions are grouped more easily (Bregman, 1990, pp. 418f.). On the other hand, there is evidence that the brain can predict pitch trajectories to a certain extent, similar to the so-called *visual momentum*, in which the brain expects objects to keep moving in the same direction because of inertia. However, Bregman has argued that there is little support for a theory of acoustic momentum, since the human speech apparatus or other sound-producing objects in nature do not have any significant acoustic inertial properties (Bregman, 1990, p. 442).

Given an example of several interfering sources, such as voices at the cocktail party mentioned above, an interesting question arises about cause and effect in stream segregation. On the one hand, it seems straightforward to say that physical properties of sound — like pitch, timbre and loudness — lead to the formation of auditory streams. On the other hand, the physical properties of the streams are not available — at least not in any easily accessible form — before the streams have been identified. Bregman has proposed to solve this “apparent contradiction” by differentiating between the real physical properties of sound, and the supposed physical properties inferred by the brain, which are only available after a significant amount of processing has taken place (Bregman, 1990, pp. 530f.).

2.2.10 Phase Perception

Harmonic sounds are composed of harmonic partials, possibly inharmonic overtones and additional noisy phenomena, such as air turbulence or non-linear interaction. All components of the sound add up to a characteristic pattern in the time domain, which is repeated in each period cycle with some stability. The pattern may form spikes, ramps or more complex structures, depending on how the different sinusoids overlap and what their relative phase offset is.

For many applications of sound analysis and synthesis, the relative alignment of phases for a sound is ignored, and only information about the powers of partials is kept. It appears that the phase can be ignored without sacrificing much relevant information — as long as the wrong alignment of phases does not lead to any unwanted cancellation artifacts because of interference.

There is some controversy on the subject of phase perception. For a long time, researchers believed that phase information is not perceived by the human hearing system at all. For example, Ohm (1843) concluded from popular experiments that humans cannot perceive phase. A similar observation was later made by Helmholtz, who noted in 1896 that the perception of musical timbre does not depend on the phases in the partials (Helmholtz, 1913).

However, more recent findings have indicated that differences in phase can lead to audible differences in timbre. Audible effects of phase change were shown by Licklider (1957) and Andersen and Jensen (2001). It appears that the exact shape of a waveform, i.e., the relative alignment of the partials' phases, influences the perception for many sounds, however, the results vary between sound types[7].

Paliwal and Alsteris (2003) have investigated the effect of phase information on the intelligibility of speech, and have found that, under certain circumstances, phase information can be more relevant than power information, especially for very large STFT windows. However, it is difficult to draw general conclusions from their experimental setup[8].

2.2.11 Speech Perception

Although speech sounds can be accurately described by the same physical properties as any other sounds, there is indication that the human auditory system has a high sensitivity for speech understanding and speaker identification. As noted by Luce (1993), the ability to understand speech is remarkable if one considers how different the utterances of individual words can be, even by the same speaker. It is also largely unaffected by distortion and clipping phenomena (Luce, 1993).

The sequence in which speech phonemes are perceived is an important prerequisite for speech understanding. An example for this are the brief pauses before a consonant when the lips of a speaker are closed and the airflow is interrupted. The perceptual effect on the listener is not that of a pause, but is correctly interpreted as pressure building up. Experiments have shown that this type of contextual knowledge influences the perception of a consonant that follows the interruption: if the speech is continued after the interruption with a different voice, the perception of closing lips does not occur and a different phoneme is perceived afterwards (Bregman, 1990, p. 532).

Although speech is composed of very different sounds alternating in rapid succession, the auditory stream does not fall apart, but is quite easily perceived as one continuous source. Noisy consonants and voiced sounds can fuse into a syllable stream, with no apparent acoustic segregation — even though the difference between tones and noise is known to be a strong clue for segregation (Bregman, 1990, pp. 105f.). The ability to identify different sounds at high temporal resolutions appears to be particularly high for speech. One reason for this, as noted by Bregman, could be the brief transition between speech sounds that occurs as the mouth moves into a new position. It appears to provide sufficient clues for the auditory system to strengthen the impression of streaming. Another factor for streaming in speech is the spatial continuity, i.e., the property of a voice to remain at roughly the same location (Bregman, 1990, pp. 549ff.).

Speech can also be subject to the continuity illusion mentioned before: it has been shown in experiments that, when a short phoneme is masked by a loud noise or a cough, the brain tends to fills the gap with a phoneme that fits into the sentence.

[7] It has been suggested that the phase alignment is more important for low-pitched sounds than for high-pitched sounds. A discussion on this topic can be found in (Andersen & Jensen, 2001).

[8] Paliwal et al. have only tested consonants in their speech experiments, even though the question of phase alignment seems much more relevant for pitched vowels. They also randomized the phases of extremely long windows (over 1 second), thus destroying the entire temporal structure of their speech recordings.

Obviously, the restoration of the phoneme — whether correct or incorrect — requires knowledge about the language, and is more than just an interpolation of harmonic partials (Bregman, 1990, p. 376).

2.2.12 Schemas

Looking at the processing steps that have been described here, in which the sound enters the ear, stimulates nerves in the cochlea and is grouped or segregated, it appears that sound perception is a bottom-up process. Yet, research indicates that significant top-down processing takes place at the same time, especially in the domain of speech perception, which differs significantly from simple low-level perception. This is not surprising because speech has such an overwhelming importance in human communication.

It appears that the human brain integrates knowledge about certain recurring phenomena into so-called *schemas*, which relate to vowels, words or even grammatical constructions (Bregman, 1990, pp. 665f.). The presence of a schema allows the brain to capture a concept as a whole, similar to the instant visual recognition of a word in contrast to its deciphering letter by letter.

2.2.13 Semantics and World Models

The highest level of auditory perception — the *semantic* level — has been left out so far in this chapter, mostly because there are no practical solutions to the problems associated with it. Semantics describe the meanings associated with a single symbol or a set of symbols, often in the context of communication (Ernst, 2002). In the auditory domain, we can relate semantics to the meaning that can be extracted from a spoken sentence. For example, the sentence "come over here!" can be analyzed acoustically, then syntactically, ultimately arriving at the conclusion that someone wants somebody else to move towards the direction of the speaker. While written sentences are limited in expression to the characters that are available[9], the spoken or shouted version can additionally convey a sense of emergency, or it may be received as a friendly invitation. Most importantly, a listener can usually decide from the context whether the sentence was directed at him or someone else, thus influencing decisions or actions. This is called the *pragmatic* dimension of communication (Ernst, 2002).

The semantic level is not only relevant for language-based auditory content. Various acoustic events can have meaning associated with. This includes signals like doorbells, sirens and ringtones, but can also be applied to sounds that are not actively used to communicate. There are different views in the literature on semantics and semiotics about what can be a semantic concept, and whether any phenomena of the acoustic environment should be included (Ernst, 2002). Still, it is clear that sounds in acoustic environments can have various types of *meaning* associated with them, and that knowing about those meanings presents an enormous advantage for the analysis of such sounds. A human listener knows that the sound of coins being inserted into a slot of a vending machine will likely be followed by a rumbling noise (or maybe by an angry kick), or that it takes an object a certain amount of time to change its location, or

[9] The unchangeable nature of letters exists mostly in typewriters and digital processing. However, in calligraphy and cartoon typography, the design of letters and punctuation is often used as a stylistic device to enrich the symbolic level with mood or volume clues.

that the radio will broadcast a news segment at every full hour. All this knowledge is part of a world model which every human listener posseses, and it greatly facilitates the discovery of aspects like causality or meaning.

2.3 Temporal Patterns and Structure

The past sections have been mostly about individual sounds and single acoustic events, their segregation and fusion. The concept of a perceptual sound object has been introduced as something that is linked to a physical event (see Subsection 2.2.8). Also, some of the basic principles of perceptual grouping of sequential events into longer streams have been discussed (see Subsection 2.2.9). But in addition to the individual events, it is also interesting to look at their principles of temporal organization and structure. Bregman (1990) has pointed out that perceptual streaming can happen on different scales, and the presence of scales can be motivated both from the ecological side and from the perceptual side of sound: it seems that there are different scales on which sound develops, and different scales on which sound is perceived.

Some of the principles that are important for low-level segregation have been discussed before. But a separation into distinct layers is difficult to perform conceptually. For example, it is difficult to say how many layers and scales are actually contained in a recording of a forest with singing birds, or of waves rolling onto the shore.

2.3.1 Patterns of Placement

Looking at one specific time scale, a model has to be defined that can describe the patterns of occurrence of individual sound objects. A simple model for a pattern can be a sequence that is repeated over and over again, as it is the case for a basic drum pattern in music. Another form of fixed pattern is an algorithmic pattern, in which the placement of new items is determined according to procedural rules (e.g., “double the duration of the pauses at each step, reset the duration of the pause to back to 1 second if 60 seconds are exceeded.”). If the elements of a sound texture model are blocks that contain silent portions as well, the patterns may be based on sequential concatenation of blocks, thus avoiding the question of timing and placement. However, such a block-based pattern model, being strictly sequential, would not allow any overlapping of sound events.

When looking at the different possible scales on which the structure of sound may be observed, a decision has to be made what kinds of scales are relevant to the question, which, in the case of this work, is the task of sound texture analysis and synthesis. In particular, it is necessary to exclude very large scales, because, according to the principle of attention span mentioned in Subsection 1.2.5, they should contribute no additional or relevant information to a texture model. Apart from that, the computational overhead for discovering statistical correspondence rises significantly when larger and larger time-scales are included in the search space, which is also why in sound texture analysis algorithms (e.g., in Dubnov et al. (2002)), the search space is typically strongly restricted to local patterns.

2.3.2 The Choice of Appropriate Scales

There can be no doubt that different time scales of acoustic phenomena exist in the physical world. Consider a thunderstorm: At the very basic level, the sound of individual raindrops hitting the ground can be seen as a “physical event”. This happens on a scale of milliseconds. From the combination of millions of rain drops, other patterns emerge, such as gradual changes in the intensity of the rain. The scale would be in the range of seconds. Of course, many additional levels could be layered on top of each other, each linked to other phenomena. The rain could stop and then start again a few minutes later, as the next raincloud approaches. The whole thunderstorm may last between several minutes and one hour, with changing characteristics that may be very meaningful to a meteorologist. In a bottom-up perspective, time scales in the physical world could be linked to chains of causality — given that the physical processes can be sufficiently well understood.

If a phenomenon is to be modeled using different time-scales, the question arises how many scales should be used, what the smallest and largest scale should be, and whether they should be allowed to overlap. One approach would be to use many scales and to guarantee that all time ranges are covered well, possibly by defining an exponential factor of increase (e.g., 500 ms, 1 s, 2 s, 4 s, ...). Another approach would be to let higher layers define statistics of lower layers. In that case, each unit of the higher layer must be large enough so that it can contain useful statistics (e.g., 100 ms, 1 s, 10 s, 100 s, ...).

Complex sounds are often modeled as two-layer phenomena: the lower layer contains atoms, blocks or objects, the higher layer describes their ordering [e.g., (Saint-Arnaud & Popat, 1997) (Hoskinson, 2002) (Lu et al., 2004)]. This is, of course, a great simplification of the actual complexity of the world: in reality, it would be very difficult to determine that a given nature sound has *exactly* two disjoint layers of time scales, neither would it have exactly three or four disjoint time-scales, unless it is the result of some very well understood process that is executed with mechanical precision. No matter what kinds of fixed scales we would invent, a given sound would likely have patterns that fall between those scales, and even if the scales were adapted for every individual texture, their upper and lower duration would most likely be fuzzy.

Although the choice of using two layers is arbitrary and most likely sub-optimal for many phenomena, it still has some justification, because it is the most simplistic model that still enables the description of distribution patterns on top of the basic element layer. The rationale behind using two layers is that the extra modeling flexibility gained by a third, fourth or fifth layer may not be worth the effort in many cases.

2.3.3 Patterns of Element Variation

When it comes to patterns of organization, the question is not only *where* the sounds should be placed in the output stream, but also *what variation* of a sound should be used. This, again, could be subject to unknown rules and causalities, many of which would be difficult to discover. A simple approach could be to add some randomization to every sound element, i.e., to alter certain characteristics within acceptable ranges in order to make the sounds seem less repetitive. The problem of how such changes of characteristics are best achieved is one of the main contribution of this thesis, and will be discussed in Chapter 5.

Not all sounds sources are well approximated by using a random variation of elements. While waves rolling onto the shore appear to a human listener as separate events without any correlation — the sound of one wave will not give us any clue of what the next wave will sound like — other sounds are much more structured. For example, bird songs contain not just rhythmical regularity, but also regularity in the use of tone variations, rising or falling tone sequences, or even complex melodies (Balaban, 1988).

The designer of a sound texture algorithm has to decide beforehand what types of patterns in the variations of sound elements should be considered during the analysis and synthesis. An algorithm that is mainly intended for uncorrelated random events, such as rain drops or waves on the shore, could produce good results based on purely random event variations. On the other hand, an algorithm for the processing of bird songs would have to use more advanced learning algorithms in order to capture the deterministic portions of the element variation.

2.3.4 Stochastic Processes

The differentiation between deterministic and random processes is important, because the two require different methods of processing. Complex deterministic patterns occur often in combination with living beings (e.g., in bird songs) or in machinery (e.g., the sounds made by a computer during startup), but are less prominent in inanimate nature scenes. This does not mean that there is no regularity in non-living nature: drops dripping from a stalactite in a cave can be very regular, and the radio signals from pulsars are known to be among the most accurately timed signals in the universe (Hewish, 1970). However, as shown in Section 1.2, the domain of sound textures takes a special interest in irregular sounds, and so the question has to be answered what methods are best suited to analyze the nature of this irregularity, and how to create signals with such properties.

The mathematical discipline which deals with processes of chance or probability is *stochastics*. The description of sound sources as being "stochastic" is common in audio processing literature, where it often refers to audible noise (Serra, 1989), but the concept itself also relates to other forms of randomness. It should be noted that the perception of randomness usually depends on the knowledge of the observer: a roll with a pair of dice — although often used as an example for a chance outcome — is bound to physical causality just like any other mechanical system. An observer who knew all relevant properties of the dice and the environment could simply calculate the outcome of the roll. But since it is impossible — at least for a human observer — to make such predictions, the result appears as a truly random process.

What is typically relevant is not the causality behind a process, but the resulting distribution of outcomes across a long period of observation, i.e., the statistics of the process. In the case of a well-manufactured, fair die, all six sides should have an equal probability of coming out on top, which corresponds to a *uniform distribution*. Another form of probability distribution that is often encountered is the *normal*, or *Gaussian distribution*. The density of a Gaussian distribution is given by:

$$\mathrm{f}(t) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} , \tag{2.4}$$

where μ is the expected value of t and σ is the standard deviation of t. The graph of a Gaussian distribution is bell-shaped and has its maximum peak at μ. The sum of all probabilities under the curve is 1. The parameter σ determined to what degree the curve is concentrated or spread out. The normal distribution is a natural property of many processes: if a variable depends upon many independent variables, each of which is normal distributed, their cumulated effect results in a normal distribution of probabilities for the value (Bronstein, Semendjajew, Musiol, & Mühlig, 2001).

Both uniform distributions and Gaussian distributions are standard models of mathematics, and they can be used easily to generate random variables with desired distributions. Amongst the many types of possible probability distributions, these two are typically most relevant for natural random processes, simply because they are entirely void of any domain-specific assumption. Another, more adaptive method for modeling probability distributions is a histogram, which partitions the value range of a variable into so-called *bins* and lists the probability or frequency with which the variable occurs in each individual bin (Wasserman, 2006).

2.3.5 Musical Structure

Among the various possible sources of sound in everyday life, music has some very particular qualities, both in its high degree of structure and in its psychological effects. By most practical definitions of sound texture, music is not a textural phenomenon (see Subsection 1.2.2). Yet, even though no attempt will be made in this work to synthesize and process music, it is still worth taking a look at the structure of music, which combines many effects discussed in this chapter.

The organization of music is essentially two-dimensional: the horizontal dimension organizes timing information and sequential order, while the vertical dimension corresponds to the frequency axis. In that respect, musical notation resembles a spectrogram (Bregman, 1990, pp. 456ff.). unlike sounds found in nature, the music spectrogram shows a high degree of regularity and structure in both dimensions. Along the temporal axis, sound events do not occur according to probability distributions, but according to the organizational patterns of beat, meter and rhythm.

The temporal organization of music into structured patterns with pauses and accentuations defines the music's *rhythm*. As a piece of music is played, the rhythm may be subject to small alterations or speed changes. Although rhythm mostly relates to temporal organization, the loudness and the degree of accentuation of individual components is also considered as part of it (London, 2002).

The rhythm of the music usually follows a continuous, regular concept, called the *meter*. London (2002, p. 531) has defined meter as "a stable, recurring pattern of temporal expectations, with peaks in the listeners expectations coordinated with significant events in the temporally unfolding musical surface". The peaks of expectations may coincide with actual note onsets, but this is not necessarily the case. The number of actual musical events may be larger or smaller than the number of peaks of expectation. While listening to music, people derive the concept of the underlying meter from the rhythm. The meter integrates aspects of the rhythm that are considered the most stable, and it can take a short while before a fixed pattern of expectations can be formed. There is some indication that, for notes sustained over long periods, listeners insert peaks into the meter that never occur in the music, because they are hinted at by other, audible patterns of regularity (London, 2002).

There is a fundamental "speed limit" for the perception of meter: humans apparently cannot recognize rhythmic details at a time-scale smaller than 100 ms. This limit applies both to onset intervals and to differences in length between notes. Perhaps more surprisingly, there is also an upper limit for metric perception: large scales beyond 5 s to 6 s are not perceived to contain metric either. As the tempo of a music performance is varied significantly, the perception of meter can change, even though everything except the time scale is kept exactly the same (London, 2002).

There appear to be several thresholds of time scales that are relevant for the perception of meter and beat. For note inter-onset intervals below 250 ms a perturbation in the beat as small as 6 ms can be detected. Another threshold appears to be around 600 ms to 700 ms, a time scale that has been found to match most people's *natural pace*[10] (London, 2002).

The way in which humans follow the structure of meter is related to another concept, *entrainment*, which describes the tendency of a listener to adjust motor coordination to a beat or meter, especially if the loudness peaks are clearly accentuated. This is most obvious in dancing, but may also happen on a subconscious level. It has been found that neurons in the supplementary motor area of the brain respond to auditory beat stimuli. The effect can be observed when people spontaneously tap along to music with their fingers or feet (Grahn & Brett, 2007).

Such observations about meter represent fundamental properties of music perception, mostly independent of any particular style of music. But above the fundamental level, other levels exist, related to harmonies, verses and choruses. Just as in meter perception, expectation is an important principle. Listeners are familiar with certain sequences of chord progressions and cadences, which are used by composers to create tension and relief, or to create a sequence of tones that leads to an end note. In contrast to the low-level auditory perception mechanisms, these stylistic elements are specific for individual musical styles and cultures. The ability to enjoy stylistic patterns like that is also subject to learning (Bregman, 1990, p. 496).

Given all the high-level patterns found in music, it seems that the principles of auditory scene analysis are of little importance: it seems far fetched that complex classical music or Jazz should be strongly based on the basic principles of auditory grouping or segregation. But even if many more factors, including emotional and intellectual aspects, play their part in the perception of music, the underlying principles are still in place. Bregman (1990, p. 457) writes:

> "To the extent that [the organizations in music] are [based on general principles of auditory organization], the underlying principles transcend any musical style. If this is true, different musical styles simply give the underlying principles the opportunity to operate to greater or lesser degrees, but cannot change them."

An interesting property of music is that it is frequently designed to fool the human auditory system. For example, several instruments may be perceived as one sound source that produces a harmonic chord, given that their onsets match. In the case of so-called *virtual polyphony*, a composer creates a sequential stream of notes that, when played in rapid succession, gives the impression of disjoint streams (Bregman, 1990, pp. 457ff.).

An important concept in music composition is *counterpoint*, a principle of combining different and independent musical voices into a harmonic structure. While, on the one

[10] To determine a person's natural pace, a common experiment is to ask people to tap regular intervals at a speed that they feel most comfortable with, and which they consider neither too slow nor too fast.

hand, the mix of voices should sound pleasant and well integrated, another goal of counterpoint is to allow the listener to perceive individual voices clearly. This can be seen as a problem of auditory stream separation, and it involves placing notes in very distinct frequency ranges, possibly combined with spatial separation of singers or instruments in the performance (Bregman, 1990, pp. 494ff.). There is an ambivalent relationship between fusion and separation in music. Bregman gives the example of a performance with two instruments, where the notion of two separate streams is just as valid as the notion of one combined stream. Although it it clearly possible for a listener to distinguish between the two sound streams of the instruments, Bregman remarks that there would be no point in letting them perform together, if their performance did not lead to any combined perception (Bregman, 1990, p. 204).

Chapter 3

Sound Synthesis and Manipulation

A model is a conceptual description, typically applied to a problem in order to understand the processes that lead to the observed data. For example, the one-dimensional model of a guitar string in motion provides an insight about the patterns of standing waves that can be expected, and it makes predictions about the tone frequency, based on the control parameters *string length* and *string tension*. This is helpful for somebody who wants to learn about the mechanical properties of guitars, and possibly for somebody who wants to simulate the behavior of guitars. However, this model says nothing about whether the sound is pleasant to the ear, how similar it is to a piano sound or how it can be most efficiently transmitted over a digital communication channel. There are numerous ways to model and encode sound, each designed to fit a special field of application, such as music synthesis, speech transmission or the creation sound effects. In this chapter, several standard techniques for audio coding and sound modeling will be discussed with respect to their suitability for sound texture synthesis. The spectral model, which covers most of the basic concepts this thesis is built on, is described in more detail, with an emphasis on sinusoidal modeling and envelope coding techniques. This chapter will also discuss the process of converting sound into the model and vice versa, a process which differs significantly between the representation models.

3.1 Model Requirements for Different Applications

There is no single optimal sound encoding model, because the requirements for sound representation vary greatly between different applications. Every model imposes a certain conceptual view on a sound, which emphasized aspects that are relevant for the task at hand, while ignoring other aspects as insignificant details. The purpose behind using a model can be a question of knowledge gain, audio quality, compression ratio, flexibility or efficiency of implementation. Often the choice of a model represents a trade-off between several of such aspects (Painter & Spanias, 2002).

An important question is whether a sound representation model is suitable for synthesis only, or if a method exists to convert recorded sounds into the model representation. While for some models the conversion is a straightforward encoding which can be precisely specified, other models are based on less robust parameter estimation algorithms or even parameter search algorithms, and are not necessarily guaranteed to produce the same outputs on multiple runs.

3.1.1 Audio Models Dealing With Compression and Transmission

Sound compression is one of the primary concerns of sound research and is used in mobile communication and digital music distribution. A source signal is encoded into a format requiring significantly less storage space than the uncompressed digital signal, where the compression ratio represents a trade-off between signal fidelity and resulting file size. The original can be obtained from the compressed stream using a matching decoder. The combination of an encoder and decoder specification is called a *codec*.

A sound codec for transmission has to fulfill a number of requirements. The encoding must work reliably and produce valid encoded streams regardless of the input. It is also very desirable that the encoding can be performed in linear time for continuous inputs, so that streams of arbitrary length can be processed. For applications like speech communication, realtime performance is required.

The decoded output of the transmission should sound very similar to the original input, although the degree of quality varies strongly between different methods. Sometimes the output is required to be identical to the input. Such *lossless* codecs are often used to store audio in archives with high quality, but cannot normally achieve high compression rates (Painter & Spanias, 2002).

Music codecs, such as MPEG-1 Audio Layer III (MP3) (Brandenburg, 1999) or Ogg Vorbis (Xiph.org Foundation, 2010), can achieve higher compression rates by allowing some tolerance in the fidelity of the encoded audio and leaving out details from the sound spectrum that have been determined to be perceptually irrelevant. These *lossy* codecs often use a psycho-acoustic model to specify how sensitive the human ear is to certain changes in the sound (Levine, 1998). MP3 and Vorbis, although designed primarily for music, produce good encoding results for a wide variety of input sounds, including speech or noise. However, when they are forced to operate at very low bit rates, the sound quality can be significantly degraded, especially if the input signal does not match the assumptions of the psycho-acoustic model (Brandenburg & Bosi, 1997).

Some codecs are highly specialized for speech transmission and can achieve extremely low bit rates for voice signals: the human voice has fewer degrees of freedom than arbitrary music recordings, and knowledge about the specific characteristics of human speech can be used to reduce the transmitted data to a minimum (Atal & Hanauer, 1971). However, when music is transmitted using a speech codec, the assumptions of the model are violated and the result is usually a very poor sound quality (Tancerel, Ragot, Ruoppila, & Lefebvre, 2002).

Compression models are normally not suited for manipulation, especially when they include entropy coding techniques to remove redundancy from the data stream. A compressed representation, as it is used by MP3 and Vorbis, does not provide any useful access to synthesis parameters and is not intended to be a creative tool. In fact, changing a value in the compressed stream would very likely lead to a corrupt file. Haus and Vercellesi (2005) have shown techniques for applying effects to MP3 files without having to perform a complete de-compression, but they do not suggest that MP3 is particularly well suited for manipulation. On the contrary: they argue that at least the (lossless) stream unpacking and Huffman coding steps have to be reversed complete, before any effects can be applied (Haus & Vercellesi, 2005).

3.1.2 Audio Models Dealing With Music Instrument Synthesis

The requirements for creative and musical synthesis applications are very different from those for compression and transmission. In particular, the model representation of a sound must have control parameters that make it possible for a human operator to change the characteristics of the sound predictably without breaking the model. A typical example for this is the adjustment of a pitch parameter in a music synthesizer: the fundamental frequency is a flexible synthesis parameter for the model of an instrument tone. Loudness, decay characteristics or reverb are some other model parameters that may be controlled using keys, dials or other interface elements. Realtime capability is often required for instrument synthesis, especially when it is used in a live performance.

Instrument synthesis models can be highly specific for different instruments. The model for a piano sound could be based on very different algorithms than the model for a pan flute or the model for a percussion instrument (Vercoe et al., 1998). For that reason, it is not normally possible to encode arbitrary input sounds into a model representation, because some creative engineering is normally required to invent an appropriate model. However, once a good model exists for an instrument, it can offer flexible controls, very efficient transmission of the control parameters and high audio quality.

3.1.3 Audio Models Dealing With Sound Morphing and Cross-Synthesis

A morph of two sounds — in analogy to the morphing of images in graphics — is a form of acoustic interpolation, where the properties of two sound are mixed to create new sounds (Caetano & Rodet, 2009). In that respect, it is a simple form of parametric synthesis, using only one control parameter instead of a multi-dimensional parameter space. “Sound morphing” can have one of two meanings: In the one type of morphing, stationary characteristics of one sound are gradually changed over time into the characteristics of a second sound. An example for this is a steady flute sound that is changed into an oboe sound. In the other type of morphing, two sound clips of finite lengths are morphed to create another clip (Slaney, Covell, & Lassiter, 1996). For example, a dog bark and the spoken word “hello” could be morphed to create a hybrid sound that is 60 % human and 40 % dog.

It is not always possible to say what constitutes a good morph between two sounds. Obviously, the start and end state should be identical to the two input sounds. But the requirements for the in-between states are more difficult to define. If sound A is from an organ pipe of length 2 m, and sound B is from an organ pipe of length 4 m, it may be a reasonable assumption that a 50 % mixed sound should correspond to an organ pipe of length 3 m. On the other hand, an interpolation according to musical pitch perception would suggest a length of $2\mathrm{m}\sqrt{2} \approx 2.83\,\mathrm{m}$ [1].

Some morphs that can be technically implemented may not even have an equivalent in the physical world. This is often the case for sounds created by different physical phenomena, such as string instruments and wind instruments: it may be possible to find a method of interpolation that is smooth and pleasing to the human ear, but one

[1] Pitch perception is approximately logarithmic: a multiplication of the frequency by factor 2 corresponds to an increase by one octave in terms of musical pitch. Half an octave (or 6 semitones) therefore corresponds to a frequency factor of $\sqrt{2}$.

cannot picture the instrument that could produce it. In that respect, it cannot be said if the chosen interpolation method is "correct" or not.

A trivial morph would be a cross fade or simple mixing of the source sounds, which may take place either in the time domain or in the spectral domain. Just as in image morphing, the results are rarely satisfying, and half way through the interpolation two distinct sounds can usually be heard — one of them slowly fading, the other one appearing (Slaney et al., 1996). This form of interpolation is unsuitable for convincing perceptual morphs and does not qualify as morphing in the strict sense. For any more advances types of morphing, a correspondence of features between the two sounds has to be established: the interpolation should smoothly shift the fundamental frequency, change formants and filter characteristics, and interpolate perceptual aspects like noisiness or roughness of the sounds. This is easiest when the two sounds are represented in the same parameter space (Slaney et al., 1996). However, transforming sounds into such a parameter space is non-trivial, because their parametric dimensions have to be extracted from the sampled representations.

A sound model which is appropriate for morphing must satisfy two main criteria: it must make the hidden, perceptually relevant parameters of the sounds accessible for the interpolation mechanism, and it must offer methods for converting sampled sound into the model parameter space and back again. It is desirable that the conversion into the model space is reversible without any perceived loss in audio quality; when morphing between two sounds A and B, the parametric representations $\overline{A}$ and $\overline{B}$ are expected to actually resemble A and B, or otherwise the whole purpose of the algorithm would be questionable. However, some minor degradation in the conversion to the model domain and back may be acceptable. For example, in the domain of computer graphics, Blanz and Vetter (1999) have demonstrated a technique for automatically adapting a parametric 3D head model to the photograph of a person. While their technique produces astonishing results, it has some limitations regarding details such as lighting conditions or hair. But in this case, the flexibility introduced by the model clearly outweighs the missing detail.

3.1.4 Audio Models Dealing With Parametric Manipulation

In audio applications like virtual instrument performance or interactive games, it can be desirable to change individual aspects of a sound, or to make a property of the sound accessible for live manipulation. Such aspects may either be controlled beforehand during synthesis, or changed afterwards in the form a post-processing effect or filter. Some things are easily implemented as post-processing effects. For example, adjusting the volume can be achieved by a multiplication of the audio output stream by a constant factor. However, other parameters are not as easily accessible. Just as for the morphing problem, the sound must be transformed into a different model representation that provides access to the relevant control dimensions (Vercoe et al., 1998).

In a truly parametric model for sound synthesis, the sound is generated according to the current state of its control parameters. Each parameter has a defined range and can often be varied continuously between its minimum and maximum value. So, in order to change a parameter of an existing sound, the sound first has to be transformed into the model representation, and the current state of the desired control parameter has to be extracted. The parameter can then be set to a new value and the changed sound can be re-synthesized.

The main difficulty is that a perceptual aspect of a sound does not always correspond to a single technical property. In a recording studio, somebody might have a request like "make this voice more *masculine*", or "make the sound of this electric guitar more *aggressive*." The problem is then not only to find out what people associate with these words, but also to identify the synthesis aspect related to them. The other part of the problem is that even if the technical properties can be accurately identified, it is not trivial in general to change them independently from all other aspects of a sound (Vercoe et al., 1998).

3.2 An Overview of Sound Coding Concepts

Vercoe et al. (1998) have used the term *structured media representation* to define audio encoding models. They have noted that any model represents certain assumptions about the nature of sound and is connected to a parameter space. It follows that "sounds that fall outside the assumptions" of the model cannot be represented well by it. In a low dimensional model each dimension carries relatively more meaning and makes a high-level aspect of a sound accessible for manipulation[2].

In this section, a number of well-known sound coding techniques will be presented, with a special focus on how they may be used to synthesize components of a textural sound. "Pure" compression algorithms, such as MP3, will not be discussed in any detail here, because they lack many important properties of truly structured coding models, i.e., models which provide access to synthesis parameters. Such perceptual audio coding methods are focused on leaving out detail from frames of spectral data, without any intention of obtaining a model that is understandable or useful for manipulation (Painter & Spanias, 2002).

It will be of interest for this work in how far the models really provide access to relevant synthesis parameters — those that would allow it to give an instrument a different timbre or change the pattern of a bird tweet. Therefore, it is important to note that the term "parametric" is used in the literature with different meanings. In a loose sense, almost all audio coding models can be called parametric, as soon as they apply any kind of processing to the original inputs to abstract from the original data. In the more strict sense, the term "parametric", according to Vercoe et al. (1998, p. 923), is often used for models in which "the dimensions of variation of a sound can be described using a simple equation, generally continuously varying in the parameters". In this work, the strict interpretation will be used, as it provides a more useful distinction with respect to the requirements for the encoding of textural sounds.

There is a distinction between *lossless* coding and *lossy* coding, or *lossy synthesis*, respectively. Lossless models remove entropic or information-theoretic redundancy (Vercoe et al., 1998), thus shrinking the file size. The corresponding decoder is then able to re-create the original data exactly as it was. In lossy coding algorithms, some details are left out from the audio input to achieve yet smaller representations, preferably such that will not be noticed by a human listener (Vercoe et al., 1998). Most analysis-synthesis algorithms presented in this chapter are lossy, especially those with a strong focus on parametric manipulation.

[2] Vercoe et al. (1998) even use the term *semantic meaning*, but use quotes to indicate that they do not necessarily refer to the literal meaning of "semantics".

As pointed out by Vercoe et al. (1998), some sound representation models are designed for “descriptive analysis”, i.e., they are designed to provide insights about a sound and are not invertible to be used as synthesis models.

3.2.1 Symbolic Representation Models

The *Musical Instrument Digital Interface* format (MIDI), which encodes musical data as a list of note control events, can be seen as an extreme form of structured audio (Vercoe et al., 1998). However, MIDI is just a synthesizer control mechanism, rather than a synthesis model. While timing information and high-level structure can be defined in a MIDI file, all information about timbre is usually missing.

Being a protocol for the transmission of musical control signals, the MIDI standard does not allow the encoding of natural signals, such as spoken conversation, sound effects or environmental sounds. The General MIDI standard does define a short list of sound effects that are played when certain keys are pressed (“guitar fret noise”, “breath noise”, “seashore”, “bird tweet”, “telephone ring”, “helicopter”, “applause” and “gunshot”) (Kaiser, 2009, p. 705), however, it is up to the synthesizer manufacturer what is actually heard in these effects. For the task of encoding arbitrary natural sound environments it is therefore quite unusable.

3.2.2 Sampled Sound

Perhaps the simplest model to encode sound on a computer is the sample-based model, in which sound is represented as a sequence of instantaneous amplitudes, corresponding to the air pressure of a loudspeaker membrane. The basic *pulse code modulation* (PCM) model provides no abstraction from the input source, does not remove redundancy and does not specifically facilitate any transformations. The PCM format is the usual way in which sound enters or leaves a computer, it does not impose any conceptual level of sound processing. Vercoe et al. (1998) list the PCM model as a non-structured representation. However, some synthesis and analysis models are built around the concept of sampled sound. They are based on recorded sounds from an external source, stored in the computer’s memory. Transformations of the sound are achieved primarily by concatenating samples in different order.

3.2.3 Wavetable Synthesis

Wavetable synthesis is a common method for synthesizing instrumental sounds, and is widely used in hard- and software synthesizers. Sampled sound for an instrument is stored in memory, along with marks defining loops in the audio. When a note is held for a long time, segments of the audio can be looped for an arbitrary amount of time, until the note ends, e.g., because the corresponding key on a synthesizer keyboard is released. The looped segment has to be seamless at the borders, so that it loops without artifacts. It also has to be long enough to capture subtle variations between different periods, since loops of a single period usually give artificial and “lifeless” sounds (Vercoe et al., 1998). The time-varying amplitude of the sound may be changed from the original recording to respond to the interactive events sent to the synthesizer. For example, a piano note may be forced to sound for an extended period of time, whereas the note of a real piano would have faded much earlier (Vercoe et al., 1998).

The wavetable model is very specific to music synthesizers and mainly deals with the problem of interactively changing the length of pre-defined instrumental sounds, based on user interaction or note sheets. It adds a layer of complexity to basic sampled sound by explicitly defining start, middle and end parts of a sound, but does not usually deal with the problem of automatic conversion from example sounds into the model. It is also not applicable as a general-purpose mode, since it is not meant to handle anything else than instrument-like sounds. As said before, “effect” instruments, such as “applause” or “seashore”, are quite common in synthesizers[3], but are not meant to be controlled during playback.

3.2.4 Frequency Modulation (FM) Synthesis

Frequency modulation (FM) synthesis is a sound synthesis technique in which a carrier oscillator wave is modulated by a second waveform. It is often encountered in the first generation of consumer sound cards and its sound is easily recognized. The modulation creates symmetric sidebands with an offset to the fundamental frequency of the carrier waveform and can be used to create sounds that resemble bells and similar objects. The parameters of the modulation are easily accessible and can be used for interactive controls, but apart from that, the FM model is too limited to serve as a general-purpose sound model. Also, there is no straightforward method for converting an input sound into an FM sound (Vercoe et al., 1998).

3.2.5 Subtractive Synthesis and Source-Filter Models

In the subtractive synthesis model, sound is conceptually separated into a source signal and a filter, thus mimicking the physical sound production mechanism of many natural sound sources. The corresponds to the source-filter model mentioned in Subsection 2.1.9. Since the model is based on subtraction, the source signal has to be “harmonically rich” (Vercoe et al., 1998).

The subtractive synthesis model has been used extensively in music synthesizers. The source signal is typically a triangle, sawtooth, square or pulse wave. Voltage-controlled filters have been used to change the characteristics of the sound, other controllers are used to modify the amplitude of the sound, often according to the *attack-decay-sustain-release model (ADSR)* (Massie, 2002). Although the controls of the subtractive synthesis models in classic synthesizers have been very popular and allow for musically expressive performances, the parameter space is limited and does not allow for arbitrary sounds to be modeled. Sounds from such synthesizers often have a distinctive sound that is easily identified by experts (Vercoe et al., 1998).

Speech Coding with the Source-Filter Model

The human voice is a popular example for a source-filter model: the pulses generated at the back of the vocal tract are filtered by the mouth and cavities of the speech apparatus. It is possible to build speech coding algorithms that make use of this model assumption. Speech has a number of characteristic formants that need to be encoded accurately, so that the speech is intelligible and the individual voice characteristics of different

[3] The General MIDI specification defines eight sound effects as part of its 128 intrument repertoire: “guitar fret noise”, “breath noise”, “seashore”, “bird tweet”, “telephone ring”, “helicopter”, “applause” and “gun shot” (MIDI Manufacturers Association, 2010).

speakers can be identified. An early system for speech synthesis was described by Rabiner (1967). Wakita (1973) has shown a technique for calculating the configuration of a human vocal tract from a recorded waveform, and translating the shape description into filter coefficients. Speech coding with the source-filter model is a type of low-bitrate coding or even ultra-low-bitrate coding, depending on the implementation. It is possible to achieve bit rates as low as 2 kb/s (Vercoe et al., 1998).

3.2.6 Granular Synthesis

A family of synthesis methods is centered around the notion of acoustic *grains*, short fragments of sound that are used as atomic building blocks from which a wide variety of sounds can be constructed. For that reason they have been called "acoustic quanta" by Roads (1988). In the standard granular model, very short windowed fragments of sound each cover a small patch on the time-frequency plane, typically at a resolution in the range of milliseconds. By combining thousands of grains, complex sounds can evolve (Roads, 1988). The granular model therefore is a two-layered concept that depends both on the grains themselves and the combination of grains into complex sounds.

Vercoe et al. (1998, p. 928) have pointed out that the granular method is "highly abstract" and best suited for "noisy or textural sounds". Although there are numerous applications of the model for artistic and musical purposes, it is not possible to directly transform an input sound to a corresponding granular representation. The number of control parameters for the selection of and placement of grains is huge, which makes automatic processing difficult (De Poli, 1983).

3.2.7 Physical Modeling

Instead of using one general synthesis model for all sounds, it is of course possible to invent models that are highly specific for an instrument or a sound-producinging object. In *physical modeling*, the physical process generating the sound is simulated by calculating the instantaneous forces acting on vibrating air columns, plucked strings or rigid bodies over time. The control parameters for the simulation can directly correspond to the natural interaction with the real instrument, such as the force and direction in which a bow is dragged across a bowed string instrument (Sinclair, Scavone, & Wanderley, 2009) or the airflow at the mouthpiece of a saxophone (Vercoe et al., 1998).

The simulation usually involves the solution of differential equations for a number of virtual points placed on the on the instrument (Vercoe et al., 1998). The degree of realism of a physical model can vary greatly. Whether a flute is modeled as an abstract pipe — defined only by its radius and diameter — or as a complex shaped wooden body, is subject to the decisions made by the sound engineer. The addition of subtle physical details — such as turbulence or friction — requires that these phenomena are well understood, which is not always the case. However, convincing sound properties for an instrument can often be achieved in spite of extreme simplification. The more detailed the simulation is, the more processing power is necessary to use it in a realtime environment.

Although real instruments are 3-dimensional structures, they can often be approximated as a one-dimensional structure, a so-called *waveguide*, along which pressure waves travel back and forth. One-dimensional waveguides can be used as quite accurate representations of guitar strings and similar mechanical structures. The waveguide model has been used as a model for songbird vocalizations by Kahrs and Avanzini (2001). Two-dimensional waveguides have been used to model the behavior of drums, gongs and similar instruments (Vercoe et al., 1998). Mullen, Howard, and Murphy (2004) have used two-dimensional waveguides to synthesize human speech, claiming that a two-dimensional model adds perceived realism, compared to the one-dimensional model.

3.2.8 Feature-Based Synthesis and Genetic Algorithms

In some of the previous examples, e.g., in the source-filter model, the parameters of a synthesizer are derived directly from an input sound. But as stated earlier in this chapter, there is not always a straightforward method to “guess” the correct parameters from the inputs. There are several reasons why this can be difficult:

1. The relation between perceptual sound properties and synthesis parameters is not understood well enough.
2. The relationship between synthesis parameters and perceptual sound properties is highly non-linear, thus making a straightforward parameter calculation impossible.
3. The number of parameters in the model and their type are not fixed, but have to be chosen adaptively.

The third case is particularly interesting: a synthesis model does not even need to have a known number of dimensions, and does not need to be parametric. For example, a sound can be defined by a set of instructions in a programming language, or by wiring together modules and connections in a modular synthesizer. Of course, the notion of “model” then becomes very abstract. For such models, the ideal synthesizer configuration cannot usually be calculated directly, and in most cases there is no compact combination of parameters that can produce an output which is identical to the target. The challenge in that case is to find a set of synthesis parameters, so that the output resembles the target according to some pre-defined perceptual features. This technique is sometimes called *feature-based synthesis*. Parameter search algorithms, such as *genetic algorithms (GA)* have been used for feature-based synthesis applications with some success (Hoffman & Cook, 2007).

In computer science, a genetic algorithm is a method of searching a large parameter space of possible configurations for good solutions, using metaphors from Darwin’s Theory of Evolution, in particular the concepts of *selection*, *crossover*, and *mutation* (Horner, Beauchamp, & Haken, 1993). Any set of system parameters corresponds to the genetic code of a particular individual. At each round of the algorithm, a population of individuals with different genetic codes is generated. According to the principle of “survival of the fittest”, each individual’s fitness to solve the given problem is evaluated, and only the best individuals are selected. Using the crossover principle, new offspring are then added to the population by combining the genes of surviving individuals. Some random mutations are finally added, so that new variations of genes can be explored. Given enough iterations, the fitness of the population will gradually increase, and ultimately some individuals can be found that provide good solutions for the given problem. While classical hill-climbing algorithms are likely to be trapped in local optima, genetic algorithms have mechanisms for escaping the local optima, because of their mutation mechanisms (Horner et al., 1993).

The concepts of GA can be applied to a wide range of problem classes. For typical implementations, the processes of the genetic crossover and mutation are strictly separated from the evaluation functions, and do not need to know anything about the semantics or data types of genes. The genetic code is often represented in binary form, so that genes can simply be switched on and off, rather than dealing with real-valued parameters. Therefore, coming up with a useful genetic encoding is one of the main challenges when working with GA (Horner et al., 1993).

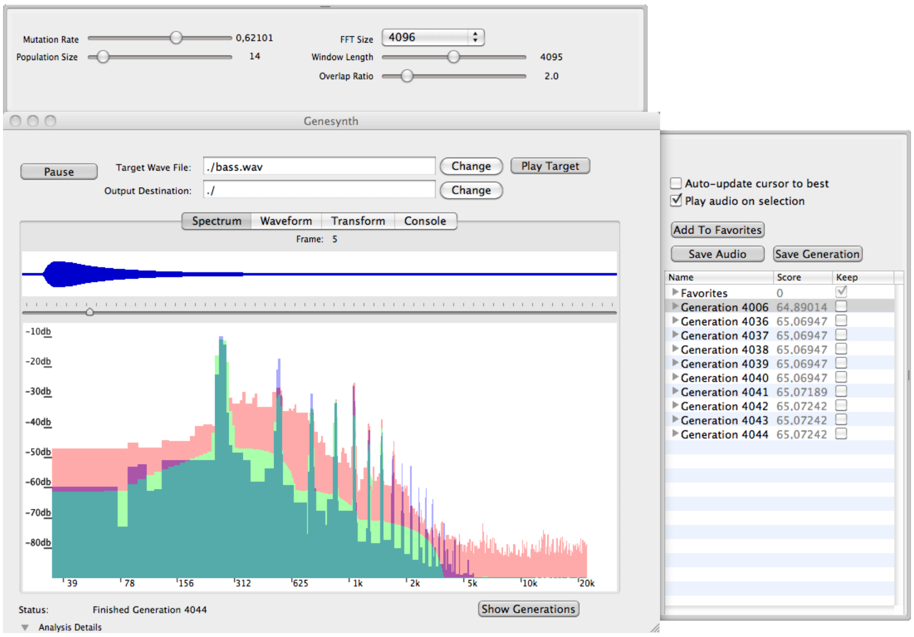
Horner et al. (1993) have used GA to find the parameters of an FM synthesizer (see 3.2.4) with multiple parallel carrier frequencies. According to the authors, the algorithm, when initialized with a random population, is able to find good matches for input sounds, although the results are never guaranteed to be optimal.

Chinen and Osaka (2007) have introduced the GeneSynth framework (Fig. 3.1), which uses a genetic algorithm to find synthesis parameters of a noise-band synthesis model. Sounds are composed of noisy sinusoids that can vary over time in center frequency, bandwidth and amplitude. So-called *PlaceGenes* are used to model the changes of parameters over time. The authors propose a hierarchical *chromosome* structure to describe the configuration of noise bands, in which dependent noise bands can be attached to a parent noise band, e.g., to model harmonic partials. Since the structure of the chromosome differs significantly from standard GA principles and can vary in length, the authors propose specialized methods for crossing and mutating individuals. They state that the solutions found by their algorithm are often "far from optimal", but that the intermediate solutions during the exploration of the search space may be interesting for creative applications. A nice feature of GeneSynth is that it assigns fictional first and last names to the individuals to illustrate their "family relationship".

The FeatSynth framework by Hoffman and Cook (2007) provides routines for feature-based synthesis. For a short frame of audio, it can search for a set of synthesis parameters matching a given set of perceptual features. For longer files, many frames are combined. Since the genetic algorithm may find different combinatory solutions for similar, consecutive frames, the authors point out that it is useful to constrain the search to find parameters which provide good continuity. The authors have called the approximation of example sounds by synthesized replacements "non-phonorealistic" synthesis.

3.2.9 Wavelets and Fractal Additive Synthesis

Wavelet decomposition of signals has become popular, because it solves some of the problems of the FFT. It uses basis functions that are compact in time and convolute the source signal with a small number of wave cycles (therefore "wavelets"). A multiresolution analysis of the input is achieved by scaling and translating copies of the "mother" wavelet. In the common case of *dyadic* wavelets, the windows sizes of the wavelets are powers of two (Dubnov et al., 2002). Wavelet representations have been used in image compression models to reduce the coding accuracy of perceptually less relevant details (Mallat, 1989), but they also offer interesting possibilities for the analysis and coding of periodic sound signals.



Mutation Rate 0,62101
Population Size 14
FFT Size 4096
Window Length 4095
Overlap Ratio 2.0

Target Wave File: ./bass.wav
Output Destination: ./

Frame: 5

Status: Finished Generation 4044

□ Auto-update cursor to best
☑ Play audio on selection

Name	Score	Keep
Favorites	0	☑
Generation 4006	64,89014	□
Generation 4036	65,06947	□
Generation 4037	65,06947	□
Generation 4038	65,06947	□
Generation 4039	65,06947	□
Generation 4040	65,06947	□
Generation 4041	65,07189	□
Generation 4042	65,07242	□
Generation 4043	65,07242	□
Generation 4044	65,07242	□

Figure 3.1: The GeneSynth software in action: the main window compares the synthesized spectrum to the original spectrum of the input sound. The best individuals of each generation are kept and displayed in the right window.

Instead of using wavelet sizes that are powers of two, it can be useful to adapt the wavelets to the fundamental frequency of a sound. Natural sound sources are rarely perfectly periodic, but can be better described as pseudo-periodic signals in the general case. Each period will typically resemble other periods directly before and after it, and most analysis methods are built to capture this deterministic trend of the signal. However, it has been noted that some of the most interesting characteristics of sound sources are contained in the fluctuations and deviations from this main trend in individual periods (Evangelista, 1993). The *pitch-synchronous wavelet transform (PSWT)* can be used to lock the scale of the wavelet decomposition exactly to the fundamental frequency of a pseudo-peroidic signal. The pitch synchronous analysis can be used to capture the local fluctuations in a signal and encode the harmonic part and the fluctuations separately. Stochastic models for the fluctuations can be used to achieve a compression of the signal (Evangelista, 1993).

Polotti, Menzer, and Evangelista (2002) have presented a synthesis model called *fractal additive synthesis (FAS)*, which is built on a *harmonic band wavelet transform (HBWT)*, instead of the PSWT model. While PSWT processes periods of a signal in the time domain, HBWT works in the spectral domain and is combined with a decomposition into harmonic bands, thereby overcoming the inability of the standard dyadic wavelet decomposition to model harmonic bands of a sound (Polotti et al., 2002).

3.2.10 PSOLA

For synthesizing speech with a high degree of realism, the *pitch synchronous overlap-add (PSOLA)* method has been proposed (Valbret, Moulines, & Tubach, 1992), which is also referred to as *time-domain pitch-synchronous overlap-add (TD-PSOLA)*. The technique uses a database of pitched speech fragments, or speech wavelets, which can be dynamically added together to produce speech (Evangelista, 1993). The fragments themselves can be scaled and placed at different intervals as required for the intonation. Since the fragments are not just sinusoids but complex waveforms, they already contain the spectral characteristics of the phonemes they belong to. PSOLA requires a large database of waveform fragments, which has to be specifically set up for a particular synthesis task. This is useful for speech synthesis, but not easily applicable to general sound coding tasks.

The PSOLA technique can lead to artifacts resulting from the scaling of the fragments and non-matching phases in subsequent fragments (Oudeyer, 2003). The speech synthesizer MBROLA by Dutoit, Pagel, Pierret, Bataille, and Van Der Vrecken (2002) uses a variation of the technique, in which the pitches and phases are normalized for all waveform fragments, so that artifacts are minimized.

3.2.11 The Phase Vocoder Model and its Applications

The *vocoder*, a contraction of the term “voice coder”, is a concept for sound transformation that allows for a number of musically interesting effects. The so-called *channel vocoder* was first described by Dolson (1986).

A vocoder splits the input signal into time-varying frequency and amplitude information for a large number of bands. For harmonic sounds, the different partials will then end up in different bands, so that they can be manipulated individually, provided that the number of bands is high enough.

A so-called *phase vocoder* can be implemented either as a filterbank of bandpass filters, or can be based on the FFT (Zölzer et al., 2002, pp. 315ff.). For the filterbank implementation, the input signal is fed into a structure of parallel filters. Each filter uses so-called *heterodyning*[4] and low-pass filtering in order to act as a band-pass filter for the specific center frequency of the band. Both the FFT and the filterbank method produce phase and amplitude information and enable the calculation of the exact frequency in a channel by unwrapping the phase between subsequent frames and deriving the frequency from the phase increase. The phase vocoder is an extension of the earlier *channel vocoder*, which only measures the time-varying amplitudes in the channels (Dolson, 1986). Since the FFT-based implementation is more efficient than the filterbank implementation, but mathematically equivalent, it is usually preferred. There is always a trade-off between frequency and time resolution of the filters: a good separation of filter bands can only be implemented at the cost of slower time response (Dolson, 1986).

In the phase vocoder, effects can be applied to the converted representation, before it is converted back into a time signal, typically using the overlap-add method. If no effects or transformations are applied to the intermediate representation, the original input is perfectly reconstructed in many vocoder implementations. The vocoder model makes some assumptions about the signals that are to be processed, and although the analysis steps can always be inverted to obtain the original input from the intermediate representation, the quality of the results for manipulated sounds depend strongly on the nature of the input signal (Dolson, 1986).

The first use of vocoder techniques was the detailed study of partials of various instrument tones (Grey, 1975). In addition, a number transformation effects have been

[4] In heterodyning, a signal is shifted into a different frequency band, using frequency modulation.

made possible through the vocoder, including relatively simple manipulations like pitch transposition and time scaling.

The pitch of a sound can be changed by speeding up or slowing down its playback. Since frequency and speed are closely coupled, it is somewhat difficult to change one without the other. However, manipulating pitch independently from temporal changes is precisely what the vocoder is good for. In the case of the FFT-based implementation, it suffices to change the spacing of the time frames between the analysis and the synthesis. However, the phase information must be changed accordingly to reflect the correct number of periods between the time-stretched frames (Dolson, 1986).

Changing the pitch is simply the counterpart of time stretching. It can be implemented most easily by first speeding up or slowing down the recording, i.e., by resampling it, and then correcting the timing using the method described above (Dolson, 1986). Another way of implementing a pitch change would be to use sinusoidal oscillators for the re-synthesis and operating them at changed frequencies. Again, it must be made sure that the phases align correctly between the frames.

As discussed in Section 2.1.10, the formant structure of sound, especially of speech, is highly relevant for perception. However, when transposing the pitch of an input signal by changing the speed of the playback, the formant structure is changed together with the partials. This effect is typically disturbing for speech signals, since the changed formants alter the characteristics of the vowels and decrease the intelligibility of the speech (Dolson, 1986). A solution is to separate the formant information from the signal and re-applying it to the output later. Using linear prediction in combination with the phase vocoder, the spectral shape can either be manipulated or perfectly re-constructed without error, so that the dimensions of time, frequency and filtering characteristics can be changed individually (Moorer, 1978).

A vocoder can be used to perform various types of mixing of two sounds, such as using the phase information of the first sound in combination with the amplitudes from the second sound. Other effects include masking of one sound with the second sound, or adding the phases of both sounds. However, not all of these operations are musically meaningful and some of them may not produce valid time-frequency data (Zölzer et al., 2002).

By forcing the phase information to be zero at each frame, natural frequency changes are wiped out and the frequencies are fixed to the center frequencies of the FFT bins. A so-called *robotization* of the human voice can be obtained with this technique. An inverse effect to robotization, called *whisperization* can be obtained if the phases are randomized, but the magnitudes are preserved. Harmonic components will then become noisy components, given that the frames are chosen to be very short, so that the phases are unpredictable (Zölzer et al., 2002).

As described by Zölzer et al. (2002), the magnitudes of a sound can be changed individually for each frequency band in the vocoder processing chain. It is possible to cut off low magnitudes below some threshold to obtain a noise-filtered signal.

3.3 Additive Synthesis

Most harmonic and quasi-harmonic sounds can be accurately described by a set of slowly time-varying sinusoids (Serra, 1989). For examples, most musical instruments have a clear structure of harmonic partials that can be almost perfectly approximated by controlling a bank of few sinusoidal oscillators. The representation of sound in the additive model enables a range of transformations, such as time stretching and pitch transformation, and it is widely used in the scientific analysis of instrument sounds (Vercoe et al., 1998). The synthesis of sounds by sinusoidal oscillators can be relatively expensive computationally. For a sound with p oscillators, it requires the evaluation of p sine functions at each sample. Using an equation by McAulay and Quatieri (1986) (presented here in a slightly different notation), we can write:

$$S(t) = \sum_{i=0}^{p-1} a_i \sin(t2\pi f_i + \varphi_i) \quad , \tag{3.1}$$

where $S(t)$ is the sampled value of the combined sinusoids at time t, a_i is the amplitude of sinusoid i, f_i is its frequency, and φ_i is its phase offset. In the form given here, stationary sinusoids are assumed, which is sufficient for describing the stationary characteristics of an instrument sound without decay. In practice, the parameters a, f and φ may be time-varying. It is common to sample their values at regular intervals, store a triplet of time, frequency and amplitude, and to interpolate them for synthesis. Another way of synthesizing sinusoids with changing parameters is the overlap-add method (Zölzer et al., 2002, pp. 242 ff.): a, f and φ are treated as stationary parameters for the duration of a frame. Frames are synthesized separately and then added together using overlapping triangular windows. One problem of the overlap-add method is that the phases between frames are not aligned and phase cancellation may occur (Serra, 1989).

3.3.1 Peak Tracking

The time-varying parameters for the oscillators can be obtained by tracking the dominant peaks in the frequency spectrum over time, so that coherent tracks of sinusoids can be formed. Most implementations are based on the *McAulay-Quatieri (MQ)* algorithm (McAulay & Quatieri, 1986).

The MQ algorithm was originally developed for speech processing and is based on the idea that intelligible speech can be transmitted when the dominant spectral peaks of the sound source are encoded for short time frames at fixed intervals. The authors have pointed out that the problem of identifying the sinusoids in a possibly noisy signal is difficult to solve analytically, and therefore a "pragmatic approach" has to be taken, in which the input is assumed to be clearly pitched and stable within the analysis window (McAulay & Quatieri, 1986).

Sinusoidal signals in the time domain form compact peaks in the frequency domain. Therefore, to obtain information about the sinusoids, the power spectrum of a sound has to be examined and searched for dominant peaks. McAulay and Quatieri (1986) have used a 512 point STFT to compute a *periodogram*. The signal is windowed using a Hamming window, where the size of the window is adapted continuously to be roughly $2\frac{1}{2}$ times the estimated pitch period. The locations of peaks are extracted from the periodogram by looking at points where the slope changes from positive to negative (McAulay & Quatieri, 1986).

At each frame during the analysis, a number of sinusoidal components are active. Each sinusoid is allowed to change gradually in frequency and amplitude, and its parameters at each frame are stored in a list. When the analysis progresses to the next frame, each trajectory is extended with the peak that provides the best continuation: as long as there is no sudden jump in the frequency trajectory, a sinusoid can be assumed

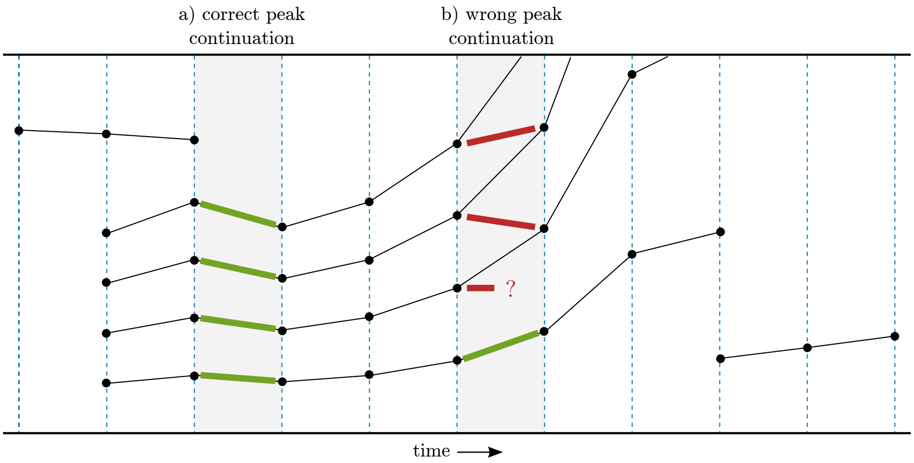


Figure 3.2: Illustration of the tracking procedure. Existing tracks can either be continued or killed. New tracks are created if peaks cannot be matched with an existing track. In the case of (a), peaks are continued correctly, while (b) shows a case of wrong peak continuation.

to originate from a continuous source. New tracks are "born" when new sinusoidal components are detected that do not match any of the currently tracked components. Correspondingly, tracks are "killed" when they have faded and can no longer be detected. McAulay and Quatieri (1986) have proposed the following procedure:

1. If no matching peak was found to continue a trajectory, it is declared "dead" and faded out to zero amplitude at the last observed frequency.

2. For all tracks, matching peaks are searched. If more than one trajectory claims a certain peak as the peak of optimal continuation, the conflict is resolved by assigning the peak to the best matching trajectory and forcing other trajectories to pick a different peak or be killed.

3. If new peaks remain that were not connected to an existing track, a new track is born.

In the MQ algorithm, the absolute frequency difference between the track and a candidate peak is used as a matching score. An absolute threshold value Δ_f is defined, beyond which a match will be rejected (McAulay & Quatieri, 1986). Fig. 3.2 illustrates the tracking procedure and common problems that may occur in the peak continuation.

3.3.2 Phase Continuation and Phase Unwrapping

To synthesize the sound from the spectral encoding model, the sinusoid tracks are generated one frame at a time. As mentioned above, the overlap-add method provides a simple mechanism of blending the frames with stationary characteristics together, but may cause interference and peak cancellation artifacts (Serra, 1989).

Using the more advanced technique of *phase unwrapping*, amplitude, frequency and phase of each peak can be interpolated, so that they line up exactly with the values of the next frame and artifacts are avoided (Serra, 1989, pp. 261ff.). The amplitude is simply interpolated by a linear function. Phase and frequency are not independent and have to be calculated together, because the frequency is the derivative of the phase and thus strictly increases over time (McAulay & Quatieri, 1986). Given that the instantaneous frequencies and phases are known at each frame and a linear increase or decrease in frequency is assumed, the task is to find an interpolation function that produces exactly the observed phases at the frames and has a frequency slope close to the predicted slope. This technique of obtaining a steadily increasing phase from the instantaneous phase estimates is known as *phase unwrapping* (McAulay & Quatieri, 1986).

3.3.3 Limitations of Additive Synthesis

Although conventional additive synthesis can be used to analyze and synthesize arbitrary input sounds, it has a number of drawbacks, resulting from inaccuracies in the analysis, but also from limitations of the sinusoidal model itself (Serra, 1989).

The MQ algorithm (McAulay & Quatieri, 1986) was designed specifically for speech, and was tested primarily using monophonic speech signals. For arbitrary signals, the assumptions made during the tracking are often too specific. A large number of sinusoids can be required to approximate complex sounds, and the frequencies of sinusoids may be spaced much closer together than what the MQ algorithm assumes. The frequency resolution of an FFT transform is limited, based on the window size and sampling rate. For a signal sampled at 44 100 Hz and a window size of 1024 samples the resulting distance between two frequency bins is ~43 Hz, which is not nearly detailed enough for most tracking purposes.

A common method to increase the frequency accuracy without increasing the number of samples in the transform is *zero padding*: the actual windowed samples are placed in the center of a larger FFT window, and the rest of the signal is filled with zeros (Serra, 1989). While this can help to measure individual frequencies more accurately, the distance at which two similar peaks can be resolved is still limited by the number of samples in the original window.

Even when the peaks in a spectrogram are resolved well enough, some problems remain. Some sound sources have significant slopes in their frequencies, so that the stability of sinusoidal signals implied by the choice of the Δ_f threshold is not a useful assumption (Bartkowiak & Żernicki, 2007). Furthermore, signals may “cross” in the spectrogram, when one source has a falling fundamental frequency and the other has a stable or rising frequency. The conventional tracking procedure will likely lead to false continuation in this case (Bartkowiak & Żernicki, 2007).

Another problem is that absolute frequency thresholds that work well for low frequencies can be too restrictive for higher frequencies. For example, while a fixed threshold Δ_f of 5 Hz would tolerate a change of the fundamental frequency from 130 Hz to 134 Hz, the 10th partial of the same fundamental would increase by from 1300 Hz to 1340 Hz within the same frame, which would not be accepted as valid continuation of a tracked peak. The result is often a fragmentation of the tracking data in the high-frequency regions, which can be easily observed with any test audio material that contains highly non-stationary signals.

The sinusoidal analysis-synthesis technique is not well suited for noisy sounds, because the synthesis of noise by using sinusoidal oscillator is extremely inefficient and can lead to audible artifacts. Also, very short impact noised and attacks cannot be modeled well, because the spacing of the analysis frames limited the temporal resolu-

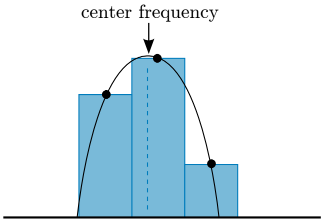


Figure 3.3: Illustration of parabolic interpolation: a parabolic curve model is fit to three FFT bins assumed to belong to the peak. The peak of the parabola is the estimated center frequency.

tion of the model. To address some of these problems, particularly the coding of noise, a more comprehensive model was introduced, called *spectral modeling*.

3.4 Spectral Modeling

Spectral modeling, or *spectral modeling synthesis (SMS)*, is an extension of additive synthesis models. Serra (1989) has coined the term "deterministic plus stochastic decomposition" for the analysis component of spectral modeling. The spectral model uses noise to encode portions of the audio that are not well captured by a pure sinusoidal model, and is motivated from several fields of audio processing, including speech coding and experimental music.

3.4.1 Improved Sinusoid Tracking

Serra (1989) has suggested a number of refinements to increase the robustness of sinusoid tracking, starting with a more accurate estimation of the peak frequencies. In addition to the standard zero padding technique, he has used a parabolic curve model that is fitted to the three main FFT bins belonging to a peak. The relative amplitude of the peaks depends on the exact location of the center frequency of the peak. Therefore, the location of the maximum of the parabolic curve will give a more accurate estimate for the true center frequency.

Serra has proposed a logarithmic dB scale for the peak energy, because it better approximates human loudness perception (Serra, 1989, p. 12). An equal-loudness curve was suggested to compensate for the different loudness perception of components at different frequencies. For complex sound sources with noisy components, it is not always trivial to decide whether a value in an FFT bin should be regarded as a peak or not. In the model proposed by Serra, both the absolute energy of a peak candidate and the energy in the *valley* next to the peak are taken into consideration. The concept is illustrated in Fig. 3.3.

3.4.2 Harmonic Tracking and Fundamental Frequency Estimation

Since sinusoids are typically parts of a harmonic series, it can be useful to track complete harmonic groups together, as dependent partials of a fundamental frequency. This is a more complex problem than finding individual peaks, because it contains the additional question what fundamental best explains the set of observed peaks. However, the awareness of harmonics can make the tracking more accurate, since information from multiple partials can be integrated into a more exact estimate of the fundamental frequency.

Partials that belong to the same source will generally show similar behavior, i.e., they will have almost the same onset and offset, and their frequency trajectories are linked to the fundamental. Incorporating this knowledge about the harmonic structure of partials can help to group them into harmonic sounds. In turn, the multiple observations of partials in different frequency bands can help to make the detection more robust against wrong estimation of partials in individual frequency bands. However, the combination of partials into combined harmonic events is difficult to perform without prior knowledge about the correct frequencies (Bregman, 1998).

In contrast to general peak tracking, the tracking of the fundamental frequency and its harmonic partials contains the additional problem of identifying which of several peaks is the fundamental, which peaks are dependent harmonic partials, and what peaks result from interfering sounds or noise. The lowest sinusoid of a harmonic series is not always the loudest, and may even be missing (see Subsection 2.2.3). Also, when the structure of partials is not clearly visible in the power spectrum, estimation methods based on the distances of partial to each other can fail. This uncertainty can cause "octave confusion", i.e., the selection of a higher or lower octave instead of the correct one. Automatic tracking procedures are also often inaccurate when sinusoids are rapidly modulated or sloped, which is a general problem of frequency measurement for non-stationary sounds (Roebel, 2006).

Serra (1989) has named several relevant strategies for the estimation of a fundamental frequency, including a strategy of picking the three highest peaks and then searching for a frequency that is a fundamental for all three peaks. He has also used a concept of so-called *guides* to lock partials to a fundamental frequency.

Klapuri (2006) has proposed an F_0 *salience spectrum*, obtained by summing up the energy in harmonic peaks for each fundamental candidate. Yeh, Roebel, and Chang (2008) have estimated multiple pitches by subtracting identified fundamentals iteratively. They have considered a spectral smoothness criterion in the selection of the best matching fundamental. Poliner and Ellis (2005) have proposed a machine learning approach to F_0-estimation, based on support vector machine (SVM) classification. A fundamental frequency estimation method based on scaled frequency spectra and SVM classification was proposed by Möhlmann and Herzog (2010b).

3.4.3 Encoding of the Stochastic Component

After the encoding of sinusoidal signal components, some noisy components remain that are not well described by the sinusoidal model. These are referred to as the *residual*, since they are typically obtained as the part that is left over when the detected sinusoids are subtracted from the original input. The residual, which is typically assumed to be the result of a stochastic process, can quite accurately be modeled as filtered noise. Examples for this are breathy noises in flutes or friction sounds (Serra, 1989).

Although even noise can in theory be modeled as a combination of many sinusoids, this is typically not desirable for manipulation purposes, because the multitude of noise fragments would require much storage space without providing any useful insight into the structure of the sound. Also, many sound transformations require that noise is treated as a separate phenomenon. For a time-stretched signal, the noise component can be synthesized with the appropriate new length, and will not contain artifacts that would otherwise appear when sinusoids are stretched out over a longer period of time.

The residual is typically encoded with less accuracy than the harmonic components. If it is generated by a noise source anyway, there is no reason to store the exact amplitudes of noise peaks. Instead, it is more important to capture the statistical aspects of the signal, i.e., how strong the noise is in different frequency bands, or how it changes over time. Serra (1989) has discussed different encoding models for the spectral envelope of a noise residual, including spline models and *linear predictive coding (LPC)*, and he has favored the spline model over the LPC model for being more more flexible.

The model favored by Serra (1989) uses a linear spline with 50 equally spaced breakpoints to approximate the spectral envelope[5]. The value of the breakpoint in each segment is the maximum peak in the segment to ensure that the envelope really encloses the peak spectrum.

A different model, which also includes both harmonics and noise, is the *harmonic plus noise synthesis (HNS)* model introduced by Laroche, Stylianou, Moulines, and Paris (1993). It is not based on fixed frame intervals, but uses pitch-synchronous offsets, similar to the PSOLA algorithm mentioned in Subsection 3.2.10. The authors point out that — unlike the MQ algorithm (see Subsection 3.3.1) — the HNS model does not require any correction of the phase at the concatenation borders, because pitch-synchronous processing guarantees the alignment of phases. The residual component can be obtained in the time domain by subtracting the estimated sinusoidal components from the original inputs. The characteristics of the noise are extracted by computing LP coefficients of the residual. During synthesis, a time-varying lattice filter is applied to a Gaussian noise source to synthesize the noisy components.

3.4.4 Effects and Transformations with the Spectral Model

When morphing two spectral envelopes, the main problem is to establish a correspondence for the morph. Ezzat, Meyers, Glass, and Poggio (2005) have proposed a matching algorithm called “audio flow”, which establishes a correspondence between DFT spectra. Caetano and Rodet (2009) have performed spectral morphing using *line spectral pairs (LSP)*, a different representation of LPC coefficients. They have addressed the problem that a linear interpolation of the LPC coefficients does not lead to a linear interpolation of the perceptual aspects of the sound. To solve this problem, they have proposed a genetic algorithm for finding non-linear interpolation curves, thus obtaining a more linear interpolation behavior of the perceptual features.

3.4.5 Processing of Transients

In addition to sinusoids and noise, transients are sometimes considered as a separate phenomenon that requires a separate processing step, as it is not represented well by either sine waves or noise models. Transients are impulses that occur when a mechanical system suddenly changes its state, as it is the case for a guitar string that gets plucked, or a drum that gets hit, or a gun that gets fired.

In the *sinusoids+transients+noise* model by Verma et al. (1997), also called *transient modeling synthesis (TMS)*, transients are detected in the signal after the deter-

[5] The amount of 50 breakpoints appears to be chosen rather intuitively. No claim is made that it is optimal in any way, but it seemed to be reasonable for the examples presented by Serra (1989).

ministic sinusoidal components have been removed, using regular tracking methods from spectral modeling synthesis. The transients are then subtracted from the rest of the signal, leaving only slowly time-varying noise. Verma and Meng (1998) use the DCT domain to detect transients, as transients cause the DCT signal to have a periodic structure. The main DCT components can also be used as descriptive parameters for the transient, which avoids storing the transients as sample buffers. They have presented an extension of the standard tracking model using a matching pursuit method for the detection of sinusoids.

3.4.6 Limitations of Spectral Modeling

Although the concept of *sinusoids plus noise* is very powerful and versatile, there are many sounds that are not well captured by it, especially inharmonic or quasi-harmonic sounds. Consider, for example, a dog bark: the sound is pitched, but at the same time highly modulated and perturbed, resulting in almost noisy characteristics. It is important to notice that the sound is neither noisy nor harmonic: it is something in between. But what effect does this have on the modeling problem? What happens if the dog bark is fed into a spectral modeling framework? There are two possible outcomes, depending on some internal threshold parameters. On the one hand, it may assume the presence of “true” noise and ignore the pitched characteristics completely. On the other hand, it may replace the sound by a multitude of unrelated sinusoids. The second alternative would likely be worse, because the sinusoids would interfere with each other in unpredictable ways. Also, having hundreds of fragmented sinusoids would be bad for storage, transmission, time-warping and most other applications of spectral modeling.

3.5 Spectral Envelope Coding

The spectral envelope is a curve that encloses the shape of the spectral peaks across the frequency spectrum, separated from any information about the signal content under the curve. The envelope is characteristic for different sounds and represents the filtering and resonance characteristics of the object producing the sound. In human in speech perception, it is the envelope that mainly discriminates different vowels, and to some extent also consonants (Pols, Kamp, & Plomp, 1969). The envelope also enables listeners to recognize individual speakers, tell the acoustic difference between a piano and a guitar, or attribute terms like “soft” or “bright” to a tone.

In many sound coding applications, such as low-bitrate speech transmission, the shape of the spectral envelope is encoded separately from the source components of a sound, especially from the pitch information (Atal & Hanauer, 1971). There are different reasons for the separation, the main reason being that it provides a better insight into the process that produced the sound (Vercoe et al., 1998). The concept that a source signal is first produced and then filtered (the source-filter model) is not artificially imposed, but relates to actual physical processes. For example, in the human speech apparatus a source signal, consisting of pitched impulses, is produced in the glottis and then filtered by the shape of the vocal tract and tongue (see Subsection 2.1.10). However, the concept of a source-filter model is not always physically accurate: there is not always a source component that “sits inside” a filtering component. For example, a metal rod resonating in the open air does not have a separate filtering mechanism, and a complex sound like howling wind cannot be accurately described as single source at all. Still, from a technical point of view all these sounds have a spectral envelope which can be measured and encoded.

Encoding the envelope information separately from the source can bring a number of advantages. In some applications, only one of the two components is needed for further processing: a tool for tuning a guitar needs only pitch information, while most speech recognition systems process only the spectral shape. The separation is also needed for many lossy compression algorithms. Knowing how the human hearing system processes each of these components — and with what accuracy — quantization can be applied to model each component individually with the required detail.

The effect of different spectral encoding methods for the application of speaker identification has been studied by Reynolds (1994), including Mel-frequency cepstral coefficients (MFCC), linear prediction cepstral coefficients (LPCC) and perceptual linear prediction cepstral coefficients (PLPC). He has found that all of the evaluated techniques can perform equally well when the number of coefficients is chosen carefully. Some problems occur when too many coefficients are used, because some of the coefficients then capture details of "spurious spectral events" or noise, thus degrading the overall performance. Reynolds (1994) has used the windowed short-time spectrum as the input directly, with no separation of the peak information from the filter characteristics. A much greater effect on the recognition performance was attributed to the use of channel compensation techniques, which can compensate for long-term stationary noise or filter effects in the source inputs. It should be noted that speaker identification has different objectives than sound object synthesis. In particular, it is not meant to be an invertible synthesis method, and perceptual aspects of the encoding were not discussed by Reynolds (1994).

3.5.1 Linear Predictive Coding

The shape of the spectral envelope, i.e., the filter characteristics for a speech model, can be implicitly coded through the calculation of coefficients of a linear prediction model. *Linear prediction coding (LPC)* has been used in speech coding because it is easy to implement and can adapt more immediately to a rapidly changing signal than frequency-domain methods like FFT windows, which limit the time-resolution of the analysis (Atal & Hanauer, 1971).

Linear prediction is based on the assumption that the next sample in a stream of audio samples can be predicted to some degree by looking at the p previous samples, as long as the source contains something other than white noise, by using a linear finite impulse response (FIR) filter model (Zölzer et al., 2002). Let $\hat{x}(n)$ be the prediction of an input signal $x(n)$. At every sample n, the predicted value is calculated by a weighted combination the past p samples of $x(n)$, using the coefficients a_k:

$$\hat{x}(n) = \sum_{k=1}^{p} a_k x(n-k) \tag{3.2}$$

If the coefficients a_k are chosen appropriately, and enough coefficients are used, the resulting sequence $\hat{x}(n)$ will be very similar to $x(n)$, and the error signal $e(n)$, obtained by subtracting $\hat{x}(n)$ from $x(n)$, will be very small and contain approximately white noise only. The corresponding z-transform of the prediction filter is given as

$$P(z) = \sum_{k=1}^{p} a_k z^{-k} \,, \tag{3.3}$$

and the inverse filter $A(z)$, i.e., the filter that, given the original input, produces only the noise signal, is given as

$$A(z) = 1 - P(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} \,. \tag{3.4}$$

The inverse filter $A(z)$ can be used to construct an all-pole infinite impulse response (IIR) filter $H(z)$ of the form

$$H(z) = \frac{1}{A(z)} \,. \tag{3.5}$$

Different methods exist to calculate the coefficients a_k from the input signal the, including a method based on autocorrelation and improved algorithms by N. Levinson and J. Durbin, which are sometimes simply called “the Levinson-Durbin algorithm” (Makhoul, 1975).

Not all sounds fit to the LPC coding model equally well as speech. As Serra has pointed out, the LPC model is a good choice for sounds with a clear formant structure, but can lead to synthesized sounds that are “quite different” from the original input in other cases (Serra, 1989, p. 43). Even in cases where LPC is effective, the meaning of the coefficients in LP models is not intuitive, since the coefficients cannot be mapped directly to frequency bands or shapes of the spectrum. For a sound designer, it would be quite impossible to obtain the intended result for a spectral envelope by changing individual LPC coefficients manually. Serra (1989, pp. 129f.) has compared the LPC coding method to his use of linear splines for approximating a spectral envelope. Although he has found the LPC coefficients to be quite suitable for compression, he criticizes that they are difficult to use in a flexible synthesis context, and that they are very sensitive to numerical errors: small changes to the coefficients may even lead to an unstable filter. Likewise, the interpolating between sets of LPC coefficients, which would be required for the synthesis of sounds with mixed spectral characteristics, does not give satisfactory results, as shown by Paliwal (1995).

A different, but equivalent representation, called *line spectral frequencies (LSF)* can be used to obtain coefficients that are less sensitive to quantization and show a more stable behavior (Kabal & Ramachandran, 1986). Fig. 3.4 shows the interpolation behavior of an 28-order LPC model in the LSF form for two artificial spectra.

3.5.2 Mel-Frequency Cepstral Coefficients

The shape of the spectrum can also be approximated using coefficients of a *discrete cosine transform (DCT)* (N. Ahmed, Natarajan, & Rao, 1974). In that case, the spectrum’s structure of peaks and valleys is approximated by superimposed cosine functions. Since this is essentially a frequency analysis of a frequency analysis — or an inversion of the spectrum — the term “cepstrum” has been coined for this representation. The first

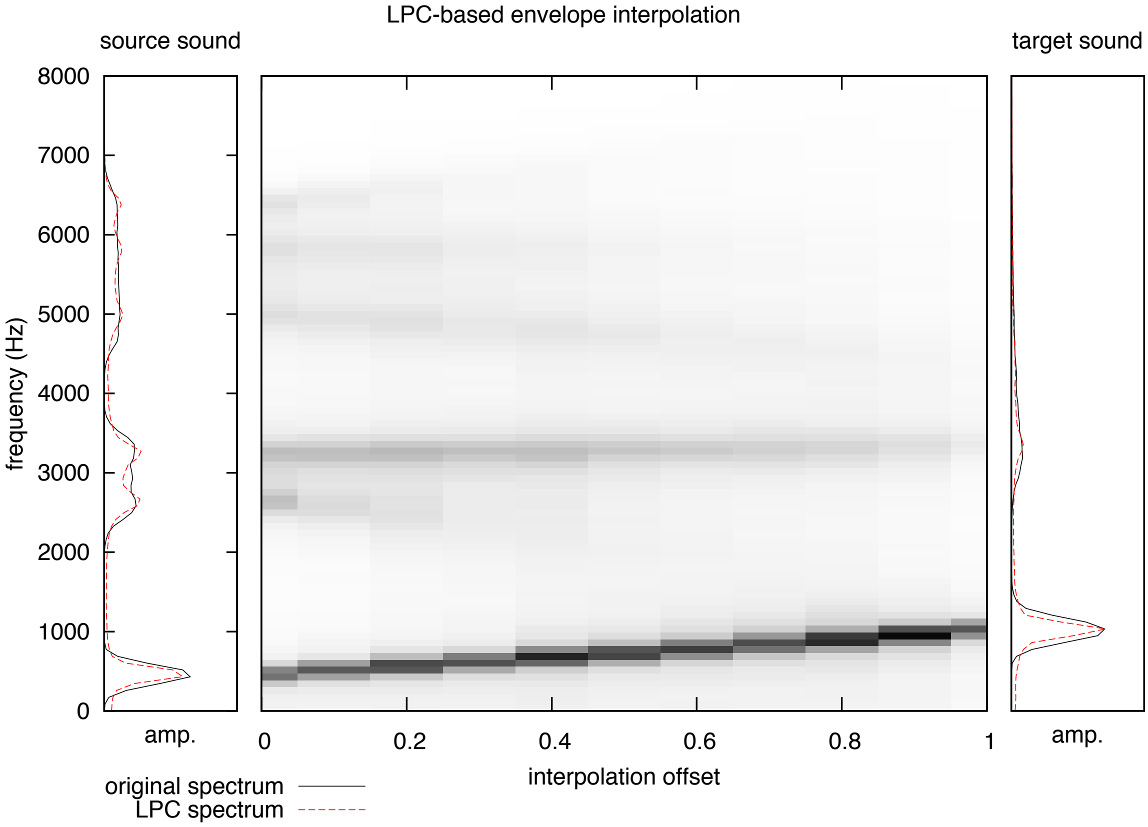


Figure 3.4: Interpolation behavior of the LPC model (lpc order 28): The envelope of the source sound (left) is changed into the target sound (right) through linear interpolation of line spectral pairs (LSP) coefficients. Narrow-band formants are approximated reasonably well, but the all-pole characteristic of the filter does not provide a good model for continuous and flat regions. The center frequencies of formants get shifted smoothly during the interpolation.

coefficient relates to the average energy, the second encodes a low-frequency gradient and further coefficients encode higher-frequency structures in the spectral shape.

Although a direct DCT of the spectrum could be used as a representation, most implementations use a different method. Human auditory perception is highly non-linear. From the lower threshold of perception at about 20 Hz to the upper limit at about 20 000 Hz, the ear is much more sensitive to spectral information in the lower half of the range (see chapter 2.2). To account for this property of the human ear, and to avoid modeling information that is perceptually irrelevant, the spectrum is not encoded directly, but first warped into the mel scale (see 2.2.3) before the DCT conversion. Also, the logarithm of the peak amplitudes is used, because it corresponds more closely to human loudness perception. The resulting set of coefficients are the so-called *mel-scale cepstral coefficients (MFCC)* (Logan, 2000).

The MFCC are based on two principles: a non-linear frequency scale and a DCT transform. Logan (2000) has critically evaluated both properties with respect to speech and music analysis. She has found that using the Mel-scale for this purpose is “at least not harmful” (Logan, 2000, p. 8), but points out that it has never been proven to be an optimal scale in any way. Still, the use of a DCT transform — as opposed to using the energy in triangular bands directly — can be motivated by evaluating typical spectra from speech and music recordings. Logan has performed a *principle component analysis (PCA)* of many example spectra, and found that the principle components are in fact similar to cosine functions of increasing frequency (Logan, 2000).

Mel-Frequency cepstral coefficients are used mainly in speech recognition or music information retrieval (MIR) applications. Typically, only a small number of coefficients are used, and thus much information is discarded that is insignificant for a detection or classification task. Although the encoding of MFCC can be reversed to obtain the original spectral shape, MFCC are rarely used for synthesis applications. Synthesis would require the use of sufficiently high resolution, but implementations like the one by Slaney (1993) use only a small set of features. Also, manipulating individual coefficients is not very practical, since each coefficient changes aspects of the whole spectrum in the shape of a non-linear cosine function.

In Fig. 3.5, the interpolation behavior of the MFCC model for two artificial spectra is shown. The filterbank implementation by Slaney (1993) was used to calculate the features.

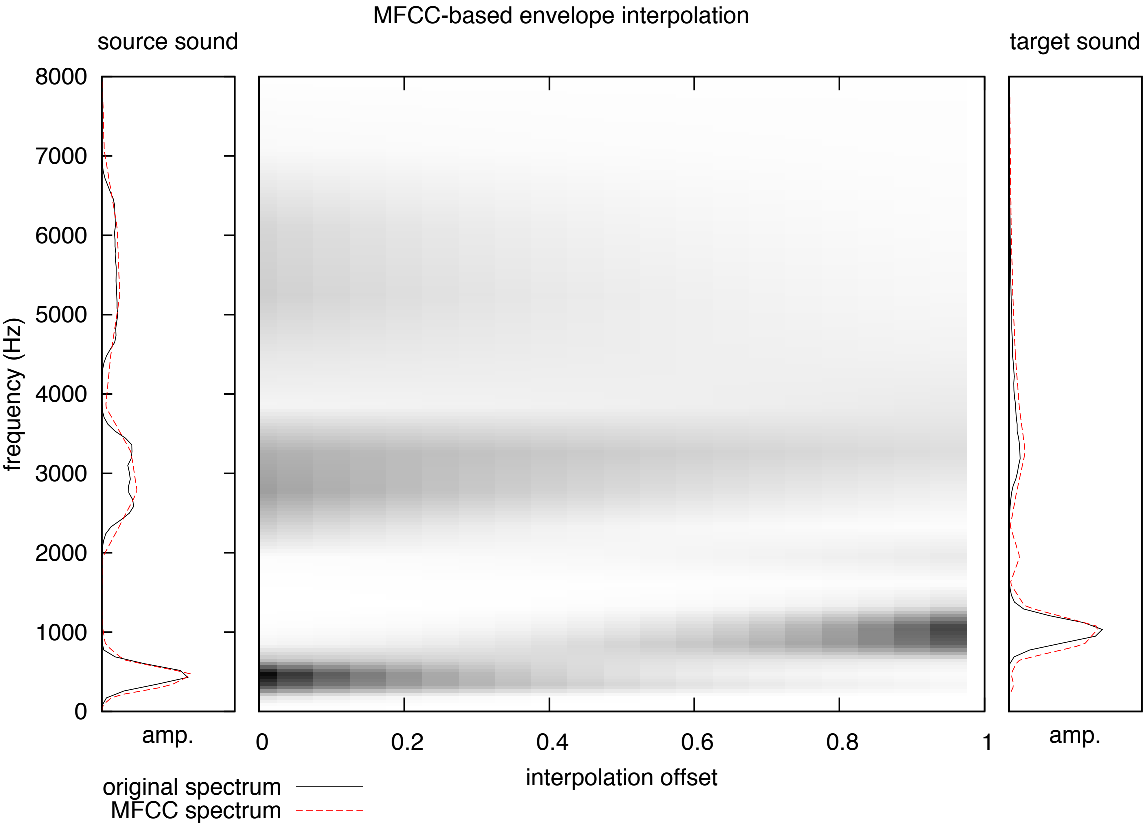


Figure 3.5: Interpolation behavior of the MFCC model: The envelope of the source sound (left) is changed into the target sound (right) through linear interpolation of the MFCC parameters. Source and target sounds are approximated reasonably well. The spectrum is continuous and valid across the whole range. The strong formants on the low-frequency region almost vanish half way through the interpolation.

3.5.3 Spline-Based Models

The shape of a spectral envelope can be approximated using line segments or smooth spline curves. Basic concepts for spline approximation have been described by Phillips (1968), Cox (1971) and de Boor (1978). Strawn (1980) has described curve approximation methods to save storage space in the spectral representation of sound. However, he has used the splines to model changes in amplitude over time, not spectral shapes. He has pointed out that the goal of curve approximation is to make the result indistinguishable from the original. He has favored the use of linear line segments over smooth cubic splines, mainly because linear segments are conceptually easier to handle and allow local manipulations to the envelope.

The curve or set of line segments is fit to the shape of the spectrum, typically by minimizing the squared error between the spline and the actual amplitude of the FFT bins (Strawn, 1980). The number of segments and the position of the so-called breakpoints between segments can be specified in advance or can be found adaptively by a fitting algorithm. The envelope representation based on linear splines was also favored by Serra (1989).

3.6 Sound Source Separation

Most algorithms for sinusoidal and spectral modeling are not well suited for processing mixed recordings of sound, because the various issues of peak tracking increase with the number of parallel sound streams. *Sound source separation*, sometimes called *source segregation* (Bregman, 1990), deals with the problem of un-mixing a recorded audio signal into separate channels, either in the auditory system of humans and animals, or in technical systems. The problem can be approached either from the ecological side or from the perceptual side (see Section 2). Seven different strategies employed in source separation are described below.

3.6.1 Linear Sound Segmentation

A relatively simple form of sound separation is the linear segmentation, or partitioning, of a longer recording into sequential blocks. The blocks can then be processed further by subjecting them to an encoding, by re-arranging them or by applying various effects. The use of linear segmentation is a strong simplification of the general sound source separation problem, and can be successful only if two criteria are fulfilled:

1. The resulting blocks must be useful for further processing.
2. The linear segmentation should not cut through continuing structures between blocks.

Regarding the first point, the usefulness of a block depends highly upon the processing task. For example, for a tasks of parametric modeling, it is desirable that the blocks after segmentation can be encoded well with the model. This often means that one block must contain only one element of limited complexity.

Most segmentation algorithms in the domain of sound textures relate to the second requirement and try to make cuts that do as little damage as possible, rather than trying to measure the quality of the obtained blocks. A common strategy is to cut

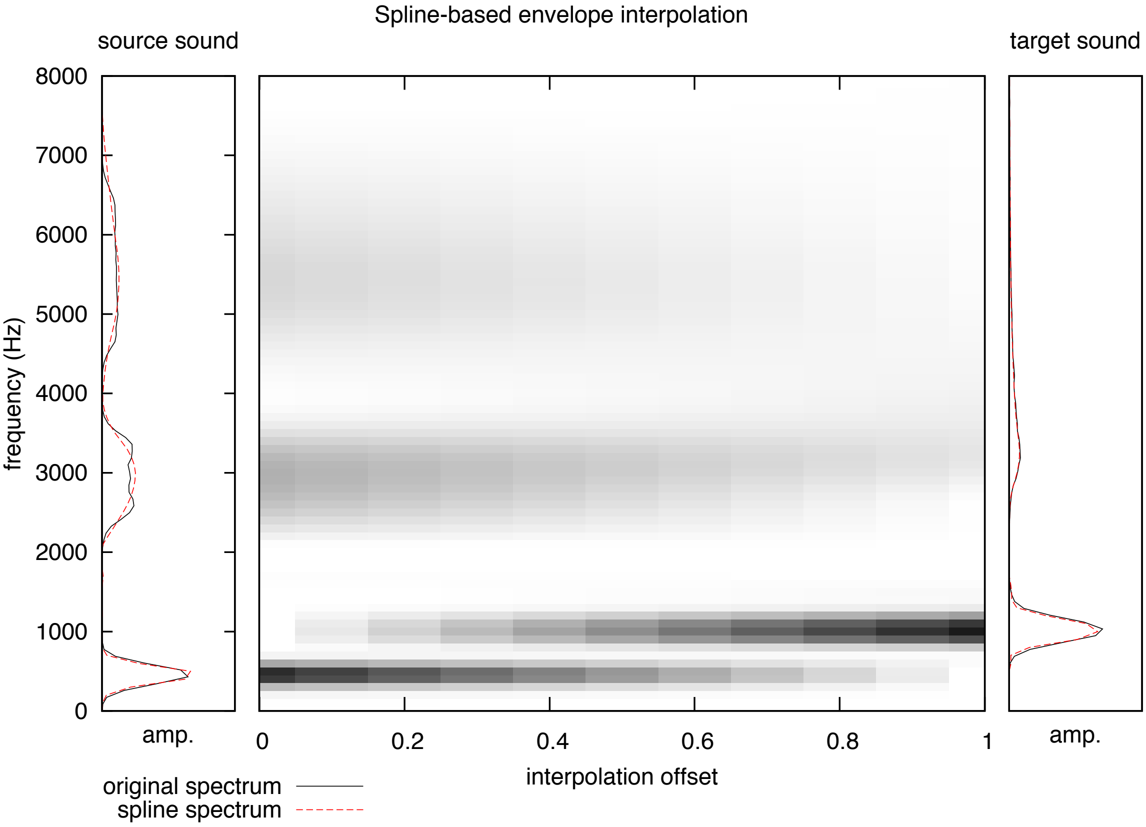


Figure 3.6: Interpolation behavior of the spline model (order 3): The envelope of the source sound (left) is changed into the target sound (right) through linear interpolation of the spline coefficients. Source and target sounds are approximated reasonably well, with some visible smoothing effects in the higher frequency ranges. The spectrum in continuous and valid across the whole range. Formants do not change their center frequencies during the interpolation, but rather appear and disappear. Half way through the interpolation, both low-frequency formants are reduced in amplitude, but present.

the audio at points where the characteristics change very strongly, a principle which is sometimes called *novelty detection*. This approach is used by Lu et al. (2004). A contrary approach by (Hoskinson, 2002) tries to separate the audio at the points of least change, arguing that — at least for speech signals — these points are likely to be breaks between syllables.

A quite different approach to separation is the search for an optimal partitioning scheme, based on a global scoring function to be maximized (Möhlmann & Herzog, 2010a). The scoring function is based on the "usefulness" criterion mentioned above. The core assumption of the strategy is that the signal can be approximated by a sequence short sections that can each by described by a simple sound block model. Each block is described through a set of parameters for the fundamental frequencies and spectral characteristics. An error value can be calculated between the model approximation of a block and its original content. The error e is low when the model fits the data well, which is most easily achieved for small blocks. To prevent blocks from becoming too short, the length l of a block is integrated into the scoring function $\mathrm{S}(e, l)$, so that an acceptable trade-off can be found:

$$\mathrm{S}(e, l) = \frac{1}{ae} \cdot l^b \mid b > 1, a > 0 \; , \tag{3.6}$$

where a and b control the weights of the error measure and length factor, respectively (Möhlmann & Herzog, 2010a).

Linear segmentation is only applicable for recordings that do not have overlapping sources, because otherwise cuts are likely to affect sounds in multiple frequency bands. For sound sources that are playing simultaneously, to useful separation can be achieved by making a simple cut somewhere along the timeline. Instead, more elaborate separations or de-correlations in time-frequency space are usually required.

3.6.2 Separation Based on Multiple Microphones

Similar to humans' ability to recognize simultaneous sound sources at a cocktail party through binaural hearing (see Subsection 2.2.9), it is possible to use two or more microphones to achieve the spatial separation of sources. Based on the relative positions of the sound sources to the microphones, the source signals enter each microphone with a different amplitude, and each microphone receives a linear mix of the signals. This can be expressed in terms of an $n \times m$ mixing matrix that contains a mapping from n sources to m sensors. Using *independent component analysis (ICA)* (Hyvärinen & Oja, 2000), the coefficients of the matrix can be computed with some success, starting from the assumption that the signals of the sources are statistically independent. The matrix can then be inverted to obtain the individual source signals from the mix. When no information is given about the placement of sources or microphones, the problem becomes a *blind source separation (BSS)* problem (Hyvärinen & Oja, 2000).

The signal path assumed by this method is greatly simplified. For real-world recordings, the source signals cannot be assumed to be a perfect linear mix, because the same sound wave reaches two microphones with a small time lag and slightly different filter characteristics. The results also degrade in the presence of noise and reverb, or when the signals are not statistically independent (Asano, Ikeda, Ogawa, Asoh, & Kitawaki, 2003).

The fact that the sound waves of a source reach each microphone with a different time delay can be used to fine-tune an array of microphones to one source. In so-called *beamforming* systems, the delays in the microphones are adjusted, so that the incoming sound waves from one source are perfectly aligned. In the mixed signal obtained from the microphone array, the signal from the respective source will be amplified. Signals from other sources will not be aligned and have a tendency to cancel each other out. The technique can be used in industry applications to measure noise originating from specific parts of a vehicle or airplane, where it is often not possible or practical to place a microphone directly at the source. Beamforming requires a known placement of sources and microphones and therefore cannot be used in arbitrary recording situations (Van Veen & Buckley, 1988).

3.6.3 Separation of Sources with Known Characteristics

Jang and Lee (2003) have proposed a sound separation method that learns basis functions of an ICA to decompose a mixed or noisy signal. They have reported some success in separating a male voice with roughly known source characteristics from a pink noise background.

While the acoustic properties of individual sources, such as instruments, voices or noise signals, can be learnt or modeled, it is not always practical to anticipate the characteristics of all sources in advance. For pure speech processing systems, it may be advantageous to integrate domain knowledge, but for the source separation of music recordings or even nature recordings there would be no point in trying to model all possible components. Therefore, algorithms that rely heavily on domain knowledge will not be discussed here in detail.

3.6.4 Blind Source Separation

Blind source separation (BSS) deals with the problem of separating a number of audio signals in a mixed recording, with very little or no prior knowledge about the nature of the signals or the mixing process. The number of signals may also be unknown. The problem is underdetermined: some restriction about the possible solutions has to be imposed, otherwise any separation (including the trivial case of keeping the original signal) would be correct. Even though no prior knowledge about the sources is available, some general assumptions can still be made, for example that the properties of each source maintain some level of continuity and that the signals of each source are *sparse*. Sparsity of a signal in the spectral domain means that one sound only affects a relatively small area of the spectrum, and most time-frequency coefficients are zero. If the observations are simple and compact, they are more likely to be true than if they are widely spread out and seemingly random. The performance of such a separation algorithm can be determined either by comparing its output to a known ground truth, or by letting humans rate the results of a separation subjectively (Virtanen, 2006, p. 10).

Music signals differ significantly from speech or signals recorded in nature. Instruments playing in the same music piece often have tones that match harmonically, therefore the energy in partial frequencies must be attributed to more than one source. Also, beats and harmonic onsets are temporally aligned, and mixed sources are thus not statistically independent as in the sparsity assumption (Virtanen, 2006).

In the *Independent Component Analysis (ICA)*, the goal is to identify signals that are statistically independent from each other, and thus can be assumed to originate from different sources (Virtanen, 2006, pp. 19f.). When the number of sensors is bigger than the number of sources, as would be the case with an array of microphones, near-perfect separation is possible, as described in 3.6.2. The problem becomes more difficult if the number of sensors is too small, or only one channel of the recording is available.

Independent subspace analysis (ISA) is an extension of independent component analysis where the subspaces are not required to be linearly independent. ISA methods can be used when the number of sensors is smaller than the number of sources to separate, as is the case for mono recordings. Dubnov (2002) has applied the ISA principle by first using ICA with a large number of components and then clustering similar components back into subspaces based on a similarity measure called *higher order statistical distortion (HOSD).*

3.6.5 Non-Negative Matrix Factorization

It is reasonable to assume that every source contributes a positive amount of power to the observed spectrogram, even though interference between sources can lead to lower amplitudes when sources are added. Based on this assumption, the mixing matrix is often assumed to be non-negative. An algorithm for *non-negative matrix factorization (NMF)* has been discussed by Virtanen (2006, pp. 26f.), in which the mixing matrix and the estimated contributions of the source signals are updated iteratively, until the values converge. Knowledge about the sources can be incorporated, such as known characteristics of instruments and their harmonic partials.

Virtanen has proposed an iterative approach to separate signals. The model is based on an observation matrix $[X]_{k,t}$, which signifies the spectrogram with time frames $t = 0...T-1$ and frequency bin $k = 0...K-1$. The spectrogram is modeled as a linear combination of the spectra of arbitrary basis functions $b_j, j = 0...J-1$, J being the number of source signals, which has to be known beforehand. At time each frame, each of the signals has a gain $g_{j,t}$. The source signals and their gains must be estimated, which is done using a *gradient descent* algorithm. Virtanen has proposed a cost function to be minimized, which is composed of a reconstruction error term, a temporal continuity term, and a sparseness term (Virtanen, 2006, pp. 34ff.). The temporal continuity term discourages strong changes from one frame to the next in any given source signal, therefore directing the algorithm towards the physically more probable solution. The author reports much better results for the examined variations of NMF methods than for the examined ISA method, but states that the sparseness term in the cost function did not have any positive effect on the results.

The NMF algorithm described above assumes static spectral characteristics and time-varying gains for the components. While this works for stationary source signals, it is mostly impractical for complex real-world input signals. One method of resolving this problem is to extend the model to allow for time-varying changes in the parameters of the individual components. The simple multiplication of the basis function with the gain factor then becomes a convolution with a component's spectrogram. Separation of components can be obtained by finding repeating instances of the components, such as drum beats. While the contributing signals in the background change, the repeating component can be de-correlated. The method can work for harmonic and noisy signals, although the correlation is typically less exact for noisy sounds (Virtanen, 2006, pp. 50ff.).

3.6.6 Sound Separation with Sinusoids and Spectral Modeling

In the spectral model, harmonic sounds are treated as a sum of time-varying sinusoids. Every sinusoid can change its frequency and amplitude over time, where time is typically resolved in discrete frames. Sound sources in the spectral model are described as the sum of an arbitrary number of sinusoids plus residual noise (Serra & Smith III, 1990). The spectral model has been explained in Section 3.4.

Sinusoid tracking has been used by McAulay and Quatieri (1986) to process speech signals. In theory, partials can be extracted from a mixed signal using tracking algorithms and then grouped back into harmonic sound sources. However, as Virtanen (2006, pp. 68f.) has pointed out, grouping sinusoids after the extraction can be difficult for complex mixed recordings with overlapping frequency tracks. The continuity of individual sinusoids is not maintained well in that case, and different kinds of fragmentation occur in the different frequency bands.

It can be advantageous to enforce harmonic relationships between sinusoid tracks already during the extraction (see Subsection 3.4.2). To account for inharmonicity (see 2.1.3), some tolerance can be allowed in the spacing of partials. Goto (2000) has used a Bayesian approach to model the placement of partials as a probability density function. A method for decomposing a signal of mixed harmonic sounds into *harmonic atoms* has been proposed by Gribonval and Bacry (2003), in which the atoms are extensions of so-called *Gabor atoms*. Using a greedy matching pursuit algorithm, the signal is reconstructed from a dictionary of such atoms.

The estimation of frequencies, amplitudes and phases can be refined iteratively. Without knowing the exact phases and amplitudes of partial tones, it is still possible to estimate the fundamental frequencies present in the signal. Starting from this approximate frequency estimate, a more accurate estimate of the phases and amplitudes can be obtained (Virtanen, 2006, p. 74).

Some problems are caused when harmonically related tones have a partial of the same frequency. The energy could then be attributed to only one sound, causing the partial to be too loud in one sound and missing in the other. A *spectral smoothness* criterion can be used to guess the correct amplitude of the partials, assuming that the energy of each partial can be roughly estimated from the powers of the surrounding partials (Klapuri, 2003). Virtanen (2006, pp. 78ff.) has demonstrated different strategies for the estimation of partial powers, including MFCC-based smoothing and linear combinations.

3.6.7 Biologically Motivated Perception Models

Besides the use of matrix factorization or spectral modeling, there are approaches to solve the problems auditory scene analysis by using approaches that correspond more directly to human hearing and perception, i.e., by computational auditory scene analysis (CASA, see Chapter 2). A computational model for biologically inspired pitch detection has been proposed by Meddis and O'Mard (1998). The implementation includes a simplified model of hair cells in the cochlea, and is, as the authors remark, in some ways counterintuitive to a more technical, straightforward solution of the problem. In particular, biologically inspired models would not use a Fourier transform, because it has no direct correspondence in human or animal physiology.

Chapter 4

Sound Texture Analysis and Synthesis Algorithms

In this chapter, existing methods for sound texture synthesis, and also sound texture analysis, will be discussed. These include methods specifically designed to process sound textures, but also other related algorithms, which either solve similar problems under a different label, or have a research focus other than naturalistic sound texture synthesis. Related methods from graphics processing are also discussed, since they share many similarities with corresponding algorithms in the audio domain. While many of the algorithms reviewed are synthesis-by-analysis algorithms, some are designed as tools to aid in the process of manual texture creation.

At the start of this chapter, the formal requirements for high-quality sound texture synthesis are examined in some detail. They will serve as a guideline for the assessment of the various methods. The wide definition of "sound texture", as described in Section 1.2, will be used in this chapter as a reference point, i.e., we are interested in sounds with inherent randomness, stationary long-term characteristics and attention spans ranging from less than one second to about one minute. The definition includes sounds such as rain, applause, traffic noise or thunderstorms, but not music or speech recordings.

4.1 Requirements for Sound Texture Synthesis

The general goal of this thesis, the goal to create "good sounding" textures, needs some clarification. A specific list of requirements is necessary to be able to judge whether any method is successful in what it does. The word "method" is used here to describe a collection of algorithms or manual steps that cover the whole signal processing chain, including aspects of the analysis, modeling, storage and synthesis. The method considered should facilitate the creation of new textures through as much automation as possible. Most current sound texture processing systems require manual steps somewhere in their analysis part, e.g., the setting of threshold parameters. This can be acceptable, especially if there is hope to replace the manual step by an automatic step in future implementations. It is also desirable that the respective methods can be applied to a wide range of input signals, preferably all types of signals listed in Section 1.2.

For any particular component of the system, very different implementations and design decisions could be made. In order to be able to assess the usefulness of any particular component, five requirements for the synthesized textures will be used: *similarity*, *continuity*, *variability*, *compression* and *controllability*.

4.1.1 Requirement I: Similarity

Perhaps the most basic requirement for any synthesized sound texture is that it must be *similar* to the input. This appears to be trivial, because without similarity, the output would not even have to depend on the input: the algorithm would be allowed to always output the same texture, regardless of what was heard in the input example. But even though similarity is clearly required, it is difficult to formalize the exact nature of similarity. In particular, it is important to note that *similarity* is not *identity*. According to the most common definitions, the synthesized sound texture should not be identical to the input. In fact, it cannot be identical if the input has a limited length and the output is continuous and not limited. The problems of finding reliable measures of similarity and quality for sound textures have also been mentioned by Parker and Behm (2004).

Similarity does not refer to values on the signal level, but refers to a higher level of abstraction. The output should match the input with respect to its statistical properties, not its sampled waveform. This is a question of perceptual similarity, rather than numerical distance. While many details of the synthetic texture may be different, overall aspects like the spectral distribution, the roughness or smoothness, the frequency of occurrence of certain elements, should stay the same. This work adopts the definition used by De Bonet (1997) for graphical textures: a texture is good if it “appears as though it was generated by the same underlying stochastic process as was the original texture” (De Bonet, 1997, p. 479). At the heart of this definition of similarity is the subjective judgment by a human observer, either hypothetical or as part of an actual evaluation.

A number of specific properties are implied in the concept of similarity. For example, the synthesized texture may not contain disturbing artifacts not present in the original recording. The audio quality should not be degraded or changed in amplification. No important elements should be missing. Their frequency of occurrence should be roughly the same as in the original input. Regular patterns of order and rhythm should be respected, as far as they are present[1].

Different types of similarity and distance measures have been proposed in the past. Grey (1975) has defined a perceptual timbre space using *multi-dimensional scaling (MDS)*. Using subjective measures between pairs of instrument sounds, a timbre space with an arbitrary number of dimensions can be constructed. However, some distortion can occur when a higher-dimensional model is forced into a two- or three-dimensional space, and there is not always a meaningful interpretation of the axes. Also, the method by Grey does not easily allow synthesis from the timbre space. Atal (1974) found that MFCCs in combination with a Euclidean distance measure give good results for speaker identification, compared to LP coefficients and several other features. The value of MFCCs was confirmed by Terasawa, Slaney, and Berger (2005) for arbitrary sounds. The authors have pointed out that linearity and orthogonality are desirable for a perceptual timbre space. Isolated perceptual dimensions, such as brightness or

[1] This might be ignored in a basic implementation, since many definitions of sound textures specifically exclude signals with regular patterns, because they typically require specialized models and analysis methods.

pitch, are known to be correlated with certain spectral properties, but do not define complete timbre spaces (Terasawa et al., 2005).

4.1.2 Requirement II: Continuity

Synthesized sound textures should have the same smooth continuity as the original input sounds. If the input texture contains howling wind without any audible breaks in between, the output should be equally smooth. On the other hand, if the input consists of isolated impulses with little continuity, the output should reflect that accordingly. The continuity criterion is thus a special case of the similarity criterion, and much in the same way, it could be satisfied simply by repeating the original input without any changes. However, a problem arises at the edges of the input recording: being of limited length, it will end abruptly at some point. But the mere fact that the recording ends cannot be taken as evidence that the texture's structure has a border. Instead, the texture is believed to continue into yet "unexplored territory".

In computer graphics, the concept of *tileable* textures is often used. A photo of some textural surface is altered, so that the left and right borders align seamlessly, as do the top and bottom borders. It is then possible to put many tiles next to each other in a grid, thus obtaining a large, seamless surface of a brick wall or grass plain. In audio, this corresponds to a seamless loop, which can usually be created by choosing the loop point at a position where the transition is not noticeable, and by using cross-fading techniques to conceal the cut.

Still, the seamless loop alone is not enough to solve the texture problem. The identical repetition of an acoustic scene over and over is likely to alienate listeners, unless the recording is very long and does not contain any easily recognizable elements. A synthesis based only on repetition will not satisfy any ambitious formulation of the sound texture problem. What is really required is *variability* of the elements, and with them, a flexible handling of the continuation between these elements.

4.1.3 Requirement III: Variability

As stated above, the identical repetition of the input waveform is not considered a correct solution for the sound texture synthesis problem — although it would clearly be one of the easiest. Identical repetition is not normally a property of natural signals: the surf on the beach sounds slightly different for each wave rolling in, and although birds have a limited repertoire of songs, no two chirps are exactly the same. This does not necessarily mean that every element must be unique, but it means that the number of different elements must be at least very large, so that it will be impossible for a human listener to identify two duplicates.

While the signal should sound similar to the input, its re-occuring elements vary, and although only a limited number of variations is observed for each element, it is natural to assume that beside these few examples an unlimited number of variations exists. The true repertoire of the process generating the sound is much larger than the small group of items that can be observed in a short recording. Ideally, the synthesized texture should reflect this variety. We would like the algorithm to "understand" the general principle behind the sound, and we would like it to "surprise" the listener with new instances.

Clearly, this is the most difficult aspect about the sound texture problem – and it has been essentially avoided in most implementations, as this chapter will show. Inferring a combined sound model from a group of examples with little a priori knowledge about the data is a difficult research problem. To some extent, variability can be achieved without creating unique variations of the elements themselves: when their order is randomized and a sufficient number of different elements is used, repetitions may not be noticeable, especially for very short transient events like raindrops or individual clapping hands in applause. However, the repetition of very recognizeable elements, such as spoken syllables or cries, may be much more noticeable.

To some extent, variability of the elements can be achieved by applying transformations to existing elements. A number of sound texture algorithms, such as those built into TAPESTREA (Misra et al., 2006), use this method. For subtle stretching or pitch changing effects, the transformed sound will still be similar to the original, however, if more extreme transformations are applied, the result may loose its realistic quality. Methods are required for creating variability during synthesis, rather than applying it as a post-processing effect later in the processing chain.

4.1.4 Requirement IV: Compression

It follows from the variability requirement that a synthesized texture signal contains a huge number of different variations for the elements. In fact, the number should not be limited at all in the ideal case[2]. The representation of the model in computer memory, on the other hand, is limited, and is created from a fixed length input recording. We expect, according to the definition of sound textures, that all relevant phenomena in the signal can be observed and understood from a short recording, that no higher-order concepts exist beyond the signal's attention span. This implies that a closed description of the texture exists: the size of the model depends on the input only, it does not have to increase in size for longer outputs.

This is closely related to the notion of compression. A model that can be stored or transmitted in compact form, but can produce a huge variety of sounds, can obviously be of great value for many applications. The acoustic properties of complex weather phenomena might be stored in models requiring only several kilobytes of storage space. Compared to the space required to store a complete thunderstorm or several minutes of rain, wind or other noises, this is certainly attractive[3].

4.1.5 Requirement V: Controllability

All of the requirements mentioned above could be met more or less by using a very long pre-recorded audio file of the texture, and playing it in a loop. The loop would be very similar to the original texture (because it would mostly *be* the original texture), it would be seamless (given that the looping at the end is done right), it could contain enough material so that repetition would rarely be noticed, and it could be compressed using any standard compression codec, like MP3. The one requirement that cannot normally

[2] Even in cases where the parameter space is continuous and infinite variations are possible conceptually, digital processing and quantization lead to a theoretical maximum of different parameter combinations that can be encoded. However, this theoretical limit is far beyond any practically consideration

[3] As already mentioned in Subsection 1.3.3, the issue of absolute storage space is becoming gradually less important as the available storage space increases. Still, the promise to synthesize an *unlimited* amount of sounds from a model of *limited* size still holds.

be met by loops is *controllability*: playing back pre-recorded material seriously limits the ability to react to inputs.

Although some effects and control mechanisms can be applied to a loop, like volume control, surround field placement, reverberating, filtering and mixing, the goal of flexible sound textures for computer games is a structural change of sound. When it starts to rain in a simulated acoustic environment, we expect that the size of the rain drops changes, and the intensity with which they hit the ground. Simply "playing with the volume knob" will likely not give the intended results.

4.2 Texture Algorithms in Computer Graphics

The term *texture* is commonly used in computer graphics, and the use of the word in audio processing is borrowed from this domain (see 1.2.1). But the two domains not only refer to the same metaphor, they share a set of common algorithms, and many principles from graphics processing can be transferred to the audio domain and vice versa. In this section, a number of common texture synthesis algorithms from the graphics domain are described, and their applicability to audio is examined. This overview is focused on algorithms that create new textures from existing input images, covering both analysis and synthesis, as this is our primary concern for the sound textures, as well.

4.2.1 Similarities Between Sound Textures and Graphical Textures

The similarity between sound and graphics becomes especially obvious in the form of a sound's spectrogram image, which can be seen both as a graphical object and as a representation for a sound recording. In fact, it can be converted back into sound — with some loss of quality — by inverting the Fourier transforms. Many image operations performed on a spectrogram image have an analogy in the audio domain:

- Adjusting the brightness or intensity of the spectrogram corresponds to changes in volume. Erasing portions of the spectrum mutes corresponding portions in the audio.
- Visual noise in the spectrogram leads to noise in the audio.
- Copy and paste operations on the spectrogram image can be used to cut a piece of audio and insert it at another position in time, possibly transposed in frequency space.

The same basic concepts can often be applied to either of the two domain. For example, Bar-Joseph, El-Yaniv, Lischinski, and Werman (2002) have used the same basic concept of wavelet-tree modeling to both graphical textures and sound textures (Bar-Joseph et al., 1999). The similarity between graphic textures and sound textures can also been found in the definitions used by the authors. Wei and Levoy (2000, p. 479) have used a description of graphical textures that has a striking resemblance to the definitions used for sound textures shown earlier:

> "Given a texture sample, synthesize a new texture that, when perceived by a human observer, appears to be generated by the same underlying stochastic process."

While the analogies are significant, it should not be concluded that an algorithm that was designed for one domain will do anything useful in the other: sound and graphics are still very different modalities. For example, Parker and Behm (2004, p. 317) have come to the conclusion that a *pixel* in graphics represents a larger portion of an image than a *sample* does in relation to a sound, and have pointed out that sound texture processing requires specialized solutions:

> "Again and again, a careful examination of the issues shows that audio data is just as complicated as image data, and in some cases more so. There is no reason to believe that audio texture generation will be faster or easier to implement than the methods currently used for images, or that those methods will adapt precisely to the audio domain."

One obvious difference between images and sound is that images are two-dimensional objects, while a sound recording is a linear, i.e., a one-dimensional phenomenon[4]. Sound can be visualized as an image (by computing a spectrogram), but it is important to remember that this is merely a change of representation, in which the one-dimensional data is folded into a different shape. Each point on the time-frequency plane of a spectrogram corresponds to a range of samples in the linear recording, and each change to one "pixel" of a spectrogram brings an effect on the temporal signal with it. As a consequence, "cutting" through a spectrogram vertically will bring changes to all frequency bands and is not a local operation. In a photograph of a patch of green grass, the x-axis and y-axis can be switched, and the result will still look like a patch of grass. Likewise, because both axes are of the same "kind", an image can be rotated by an arbitrary angle, and will still be an image. On the other hand, switching the axes of a spectrogram will turn it into something quite different and possibly meaningless.

Another important difference between sound and graphics is that sound is additive: the pressure waves of two sources are mixed into a combined pressure waveform. Any moment in a recording may contain the sum of an arbitrary number of sources, rather than the waveform of a single source. In graphics, however, objects in the foreground usually mask objects in the background completely. A common, although idealized assumption in image processing is that each pixel in an image belongs to exactly one object, and that objects can thus be segmented from other objects by finding their outline. Of course, this is only true as long as semi-transparent objects or reflections can be ignored, objects are not blurred and the quality of the image is sufficiently clear. Texture algorithms have been applied to video, as well, e.g., by Bar-Joseph et al. (2002), in which case they have to operate in a three-dimensional domain.

There are many other properties of sound that have no direct correspondence in graphics. For example, harmonics and partials are a central principle of organization in sound, but do not match any mechanical principle in the visual domain. Also, sound has a natural direction of causal development along the temporal axis, while the assumption of a direction in image space is arbitrary. In conclusion, it can be said that the differences between sound and graphics are quite fundamental, and that there is no trivial mapping of algorithmic solution from one domain to the other. However, looking at the respective other domain can inspire interesting solutions, as will be shown in several examples in this chapter.

[4] In contrast to a sound recording, the original sound pressure waves propagating through three-dimensional space are of course not linear. However, sound is represented as linear data streams on CDs, tapes and other media, which therefore seems to be an appropriate representation.

4.2.2 Procedural Graphical Textures

In procedural texturing methods, patterns are produced through algorithmic processes and random distributions. A classic example are synthetic marble or wood textures that can be created using so-called Perlin noise functions (Perlin, 1985). For Perlin noise, several noise functions are added, decreasing the noise scale and amplitude in each step. The result is a naturalistic and continuous, yet very detailed noise function in arbitrary dimensional spaces. The noise values can then be mapped to RGB color values, gradient ramps, transparency, reflectivity or arbitrary other properties of a surface.

Instead of directly looking up values in a noise function, noisy or pseudo-random patterns can be understood as the result of a growth process. For example, simulation systems for the growth of virtual plants, so-called Lindenmayer-systems (or "L-systems"), have been described by Prusinkiewicz and Lindenmayer (1990). A reaction diffusion model for texture generation was introduced by Turk (1991), where patterns grow according to biologically motivated procedural rules. The author has demonstrated the capability of the algorithm to produce leopard spots, zebra stripes and related patterns.

The generative rules for the texture are typically given by a programmer, and there appears to be no straightforward way to obtain a procedural texture from a complex input image. In that respect, procedural methods cannot easily be applied to analysis-synthesis problems, still they are mentioned here for their historic importance in computer graphics.

4.2.3 Random Field Methods and Pattern Theory

A 2-dimensioal image texture is a grid of colored cells. Depending on the nature of the pattern, the pixel colors may be purely deterministic (e.g., alternating black and white pixels), or purely random (white noise). In the more common case, however, patterns have mixed aspects of randomness and determinism. Every pixel has a conditional probability to have a certain color, depending on the colors of pixels in its surrounding. If a pixel's color depends strictly on its immediate predecessors (in x and y direction), the probability distribution can be treated as a Markov random field.

The application of random fields to the texture synthesis problem was first introduced by Cross and Jain (1983). In a random field, the value of cell (i.e., the color of a pixel) depends locally on the values of adjacent cells. For example, the pixel may have a certain conditional probability of being blue if most pixels around it are blue as well. The probabilities in the random field can be derived by analyzing the pixel neighborhoods in an input image. Simple random field methods can reproduce typical patterns of the input texture faithfully, but they often fail to capture larger regular structures, because they only respect immediate neighborhood relations (Cross & Jain, 1983). The synthesis of pixels based on neighborhood relations can be seen as a problem from *pattern theory* (S. Zhu, Wu, & Mumford, 1998).

4.2.4 Hierarchical Model, Pyramid Transforms and Wavelets

Heeger and Bergen (1995) have introduced a technique for texture analysis and synthesis based on a steerable pyramid transform. The approach is based on the notion that a texture appears similar to another texture if the filter responses of directed filters are distributed in statistically similar ways. The authors have proposed steerable pyramid filters to de-compose the original texture. For the synthesis of the output texture, noise is generated the histogram of which is then matched to correspond with the filter histogram of the input. According to the pyramid principle, the procedure is repeated at different scales.

De Bonet (1997) has developed a slightly different technique to find interchangeable regions in a texture image at multiple scales. The procedure uses a Laplacian pyramid decomposition of the input image. It begins at a coarse, down-sampled resolution and adds higher frequency detail at each step. A component may be copied from the original image into the new image if its parent grid cell has similar perceptual features. The algorithm works for textures that have limited local patterns, but cannot capture large-scale structures.

Another algorithm by Wei and Levoy (2000) also works on different scales, but uses Markov random fields to select the best matching pixel. The authors have proposed *tree-structured vector quantization (TSVQ)* for a fast lookup of the closest matching neighbor.

Bar-Joseph et al. (2002) have used a wavelet tree decomposition of textures to capture details at different scales. The algorithm is technically similar to the methods by Heeger and Bergen (1995) and De Bonet (1997), but was also used by the authors to synthesize 3-dimensional graphical structures (i.e. movies). A method for mixing textural aspects of several input textures was also demonstrated. The application of the method to the audio domain by the same authors is discussed later in this chapter.

Han, Risser, Ramamoorthi, and Grinspun (2008) have described a method for synthesizing huge image maps with details on different (and possibly infinite) scales, using so-called *exemplar graphs* as an input. The root node in the graph contains the coarsest level of detail, other nodes contain images of magnified details, which in turn have child-nodes of yet higher resolved details. The authors even allow loops in the graph structure, so that infinitely recursive textures can be described. Strategies for selecting details and color values at the appropriate graph hierarchy and resolving contradictions are included in the algorithm.

4.2.5 Tile and Patch-Based Texture Synthesis

Efros and Freeman (2001) have described a *quilting* technique to create large textures from patches taken from an input texture. Patches are stitched together with a small overlap, and the degree of correspondence in the overlapping region determines what candidate patch is inserted. In a final step, the borders between the patches are changed from straight borders to borders of arbitrary shapes, finding a minimum cost path with respect to the resulting error.

Cohen, Shade, Hiller, and Deussen (2003) have used so-called *Wang tiles* to fill a plane using a small set of tiles. In contrast to a strictly repetitive single tile, wang tiles may only be combined in special ways. The borders of the tiles are color-coded, and the colors of borders must line up correctly in all four directions. The set of tiles must be designed so that the whole image plane can be filled, i.e., that at least one tile exists at every position that has the required border colors. Typically, the Wang tile set contains more than one matching tile in each case, so that randomization is introduced and the filled plane does not end up with a periodic texture. The same basic principle can be applied to sound. The two-dimensional concept of the image plane would have to be reduced to a linear sequence, because phenomena along the vertical frequency axis cannot always be treated independently. A related algorithm from the audio domain is shown later in this chapter, along with the drawbacks of this synthesis method.

4.2.6 Limits of the Sound-Graphics Analogy

Although the domain of graphical textures can serve as a source of inspiration for sound textures, and although some of the concepts can be — and have been — almost directly applied to the acoustic domain, there are limits to this analogy. From a technical point of view, graphical textures are a two-dimensional medium, while sound is only one-dimensional[5]. This, at first glance, seems like it should make matters simpler, but the one-dimensional structure of samples in an audio buffer is deceiving: sound is still a phenomenon with very complex inner dependencies.

Parker and Behm (2004) have remarked that generating sound textures is far from trivial, compared to visual textures, because individual samples carry no frequency information, while color information is directly available from single pixels in an image. Also, the concepts of a local neighborhood in images and in sound are quite different in scale: while in an image, identifiable structures can be formed by groups of a few pixels, sound requires long sequences of correctly arranged samples to produce a single fragment of identifiable acoustic timbre. For low frequencies with long wavelengths, thousands of samples may contribute to a local perception of one low-pitched tone.

One of the most important differences between sound and graphics is that sound is an oscillatory phenomenon, where perception is not induced by the absolute values of samples, but by the frequencies of superimposed waveforms. These waveforms are often in harmonic relationships to each other, and any disturbance within the harmonic structure, or any discontinuity in phase, will immediately cause audible errors.

4.3 Manually Constructed Sound Textures

In the domain of audio processing, some algorithms have been proposed to produce synthetic textures automatically from input examples, much like graphical textures. But before investigating those methods that work automatically, this section will introduce concepts for sound texture synthesis that require some degree of human interaction. Using such manual methods, it is typically easier to obtain high-quality results, however, the outcome depends significantly on the skills of sound engineers to transform the inputs into an attractive texture. Therefore, an estimation of the quality of the respective method is not easy to make.

4.3.1 Iterated Nonlinear Functions

Di Scipio (1999) has investigated the use of iterated nonlinear functions in phase space to produce “sound textures reminiscent of rains, thunderstorms and more articulated phenomena of acoustic turbulence” (Di Scipio, 1999, p. 109). Although most sound texture algorithms work in the frequency domain, the author points out that the processes of “burning materials, rocky sea shores, certain kinds of insects, etc.” are much better modeled in the time domain. The method is based on *functional iteration synthesis*

[5] As stated in Subsection 4.2.1, even when visualized as a 2-d spectrogram, sound does not have two actual, independent dimensions.

(FIS) and uses a set of parameters to control the synthesis. Di Scipio explains that the periodic properties of generated sounds are mostly encoded in the iteration itself, rather than in the parameters of the function. The use of nonlinear functions has a serious drawback: no method is available to select the functions and their parameters to match a desired output sound. Instead, manual fine-tuning and experimentation is required to obtain useful results (Di Scipio, 1999). The nonlinear behavior of the iteration makes it difficult to control and puts the research into the field of experimental sound design.

4.3.2 TAPESTREA

Automatic analysis and synthesis methods for sound textures available today suffer from a number of common problems. A fully automatic approach would have to face to problems of auditory scene analysis described in Section 2.2, including problems to recognize what constitutes an element in a given recording, to separate elements from the background, and to extract structural combination patterns from the observed input. Therefore, instead of dealing with all these hard problems, it may be appropriate to design complex sound textures with manual interaction. The TAPESTREA software[6] offers a toolbox for audio engineers to construct complex sound textures from simpler building blocks (Misra et al., 2006). Single harmonic sounds and brief transient events can be marked in the audio spectrogram and extracted semi-automatically as re-usable *templates*.

TAPESTREA can produce new sound textures of arbitrary length by placing the templates into a new audio stream at randomized positions. The frequency of occurrence can be specified for each template. Additionally, some simple effects can be applied to the templates to make them appear less repetitive.

The separation of harmonic sounds from the background is performed by tracking sinusoids. Some parameters, such as the peak-to-noise ratio or the number of expected sinusoids, can be adjusted. While the technique built into TAPESTREA works reasonably well for strong harmonic sounds against a low-noise background, the tracking is less successful for very noisy signals. During tracking, all sinusoids are treated as independent tracks, not as partials belonging to a common fundamental. The lack of grouping can cause unrealistic dissonances between the partials, as discussed in Subsection 3.4.2.

The extracted templates in TAPESTREA are stored as sampled sound or sinusoidal data according to the spectral modeling paradigm. No abstraction or parametric model is applied to the data. Therefore, only simple transformations of the templates are possible, for which no parameter space is necessary. Time-stretching and frequency-warping are possible, but TAPESTREA cannot morph two templates, or change a curved frequency trajectory, or make a judgment about the similarity of two templates. Still, the partly manual approach taken by the developers seems reasonable, as it avoids the auditory scene analysis problems of a fully automatic system. Some concepts found in TAPESTREA, such as the concept of placement patterns, are also part of the sound texture framework proposed in Chapter 5 of this work.

[6] TAPESTREA, v0.1.0.6, `http://taps.cs.princeton.edu` (last visited: December 1, 2010)

4.4 Automatic Synthesis-by-Analysis Methods

In this section, some methods will be examined that are designed to automatically produce textural sounds from input examples. Although they are essentially automatic algorithms, many still require the specification of parameters for the analysis or synthesis stage, in particular threshold values or window sizes that adapt the algorithm to inputs of various characteristics.

4.4.1 Grains and Blocks

Saint-Arnaud and Popat (1997) have suggested a conceptual model of sound textures that consists of two levels, with *atomic* elements on the lower level and structural information at the higher level. They specifically include sounds like large crowds, rain and “fishtank bubbles” (Saint-Arnaud & Popat, 1997, p. 293). The strict two-level separation is made primarily for practical reasons, and the authors point out that for other textures a different grouping concept may be more appropriate, and that the border between the two levels is not strictly fixed.

The authors use energy in time-frequency channels as their atoms, with eight frequency bands at each time frame. They point out that in the extreme case of just one frequency channel, the algorithm will work on individual PCM samples as atoms, which would turn the spectrogram-based method into a sample-based method.

The second level of the sound texture describes likely combinations of atoms: the neighborhood of certain atoms is assumed to be characteristic for the signal. For each atom, its own value and the values of its neighbors are combined into a feature vector and mapped into a d-dimensional space. The space describes a *probability mass function (PMF)* for the signal. Where the space is densely populated, many similar neighborhood relations between atoms can be found.

The combination of atoms back into sounds is done by concatenating atoms in rows and columns, starting with the lowest frequency band. According to a sampling mask, neighbors of the new atom are considered, and a coordinate in the PMF space is obtained from their values. A good match for the new atom can then be looked up in the according neighborhood in feature space (Saint-Arnaud & Popat, 1997).

Although the algorithm can perform reasonably well on rain and applause sounds, the authors name some limitations, mostly the algorithm’s inability to maintain long-term structures and to respect regular patterns [pp. 302 ff](Saint-Arnaud & Popat, 1997). The coarse approximation of the frequency space with just eight frequency bands is also mentioned as a reason for degradation in perceived quality. Although harmonic sounds are introduced as an important atomic type for sound textures, all given examples are inharmonic, and the poor frequency resolution of the algorithm should make the faithful rendering of harmonic sounds quite impossible, including voices, animal vocalizations and humming engines.

Hoskinson (2002) has used the concept of *natural grains* to synthesize sound textures from input examples. A grain is any portion of audio that is not further subdivided, and is typically a much smaller object than the “block” concept used by Saint-Arnaud and Popat (1997), although the distinction is not always made. The main focus of the work is on smooth transitions between grains to obtain a pleasant texture and avoid border artifacts found in some other algorithms. To achieve this, a transition map is computed for the set of available grains. The segmentation of audio into grains is purely temporal and does not allow for independent treatment of phenomena in different frequency bands.

For the segmentation of audio into grains, the audio is split into small frames and wavelet coefficients are computed. The coefficients of subsequent frames are compared, and the audio is split between frames that have a small difference according to a threshold value. This means that the variability of coefficients is greater within grains than between grains, and ensures that the transition from one grain to the next occurs in a less noticeable frame border. No grain can be smaller than 40 ms.

During re-synthesis, the choice of the next grain depends on the last grain, according to a Markov chain principle. The most likely grain is the one with the greatest transition smoothness. Some randomization is added to the selection process, and the insertion of recently used grains is discouraged (i.e., reduced in probability), so that the synthesized audio does not repeat the same portions over and over. The main drawback of the method is that the actual statistical property of the sound is not reflected during synthesis. Just because two grains line up well at their borders does not mean that there is any evidence they belong together in a sequence.

Dubnov et al. (2007) have created automata on-the-fly from input audio streams. In their framework, called “Audio Oracle”, new sound can be synthesized by finding a matching suffix sequence in the automaton and traversing to on of several possible target states. The system is designed to be modular, so that different representations of frame-based audio features and similarity measures can be integrated. The authors do not mention any specific advantages over existing block-based concatenation methods, such as the one by Saint-Arnaud and Popat (1997) or by Hoskinson (2002).

4.4.2 Constrained and Unconstrained Synthesis

The algorithm by Lu et al. (2004) also uses the concept of smaller *building patterns* or *subclips* to generate longer streams, which the authors call “unconstrained synthesis”. They also give examples for the restoration of lost audio segments, which they refer to as “constrained synthesis”. The basic patterns are obtained by grouping sequences of frames that are similar, based on their MFCCs. For the synthesis, a similarity measure is computed between subclips. The similarity is 1.0 if the clips line up perfectly, which is typically only true for the original sequence. The algorithm is then forced to add randomization, by forbidding the selection of subclips within a certain range. The differences between this algorithm and the algorithm by Hoskinson (2002) consist mainly in the different grouping principles: instead of maximizing the inter-frame difference within grains, Lu et al. (2004) choose homogeneous grains with little variation in the MFCCs. Again, the original sequential structure is not kept, and homogeneity is chosen as main criterion for synthetic sequences. The authors state that their method can only be applied to sounds with “simple structures” [p. 175] and will fail to reproduce music or other complex signals. They also mention a potential problem of poor continuity for pitched sounds, because the MFCC-based similarity measure does not respond well to smaller changes in frequency.

Another variation of the same principles was proposed by Parker and Behm (2004). Their concept of *tiling and stitching* is borrowed from methods in computer graphics, mostly from Efros and Freeman (2001), but in spite of the different metaphor, the details of the algorithm are very similar to the algorithm by Lu et al. (2004). Audio is cut into blocks, the similarity of blocks is considered during synthesis. Repetition of the original sequence is discouraged through penalty values, according to a least-recently-used (LRU) principle. Additionally, borders between blocks are smoothed. The optimal block sizes depend on the type of input signal. The authors propose an automatic selection of block sizes, based on amplitude peaks, but admit that this experimental method is not as reliable as manual block size selection.

Strobl et al. (2006) provide an overview of sound texture publications, in which they compare definitions, domain restrictions and algorithmic details from various authors. In her diploma thesis, Strobl (2007) has aimed at improving existing sound texture methods, in particular Hoskinson (2002) and Lu et al. (2004), with a strong focus on parameter selection. She claims that subjective judgment of audio quality and artifacts is necessarily a part of this process. Her comparison between the two methods shows an advantage for the method by Lu et al. (2004), for producing segments that more closely resemble segments as perceived by a human listener. Strobl et al. (2006) has listed examples of sound textures that worked very well, including a nature recording with singing birds and a very homogenous background. For most cases, she has recommended increasing the size of coherent blocks, so that continuity errors and artifacts are minimized. However, she does not address the problem that — even if artifacts could be reduced — this could make larger blocks much more recognizable and probably lead to a disturbing repetition of familiar blocks.

4.4.3 Sound Textures Based on Linear Prediction Coding

Athineos and Ellis (2003) have proposed a combination of time-domain and frequency-domain methods to encode sound textures, which they have called *cascade time-frequency linear predictive (CTFLP)* analysis. This is an extension of simpler models that only use linear prediction filtering to re-create the spectral aspects of a sound, but neglect the temporal micro-structure of the signal. The algorithm uses a two-stage process, in which the broad spectral aspects of a frame are encoded in the form of LP filter poles. The temporal structure is then encoded by a second set of filter coefficients in the frequency domain. In their tests, the authors have used 40+10 coefficients combination for the two stages. For sounds with a dense temporal structure, they report an improved perceptual quality over a model that uses all 50 coefficients to encode only the spectral shape, however, they restrict their application domain to noisy, stochastic sounds only.

To evaluate the quality of the encoding, Athineos and Ellis (2003) propose a similarity measure that compares the energy in the matching time/frequency bin between the original sound and the re-synthesized version. As the authors point out themselves [p. 3], this is not necessarily the same as perceptual similarity. To evaluate that, psycho-acoustic effects like masking would have to be taken into account. The CTFLP technique is not directly applicable to the sound texture analysis-synthesis problem, because it simply encodes sounds and does not introduce any variation into the synthesis. However, the authors express their hope that the proposed model can provide a parameter space that can be the basis for flexible synthesis, using a statistical model of the filter parameters.

X. Zhu and Wyse (2004) have used a very similar approach of encoding time-frequency linear prediction parameters , which they call *time and frequency domain linear prediction coding (TFLPC)*. They separate foreground events from the background by isolating audio frames around local peaks. The coefficients obtained from frames are mapped into a feature space and clustered to reduce the amount of data in the model. During synthesis, the frequency of occurrence of the events is used to create a Poisson distribution of new events. The background is modeled in the form of time-domain LPCs. Since no audio clips of their generated textures are available for listening, it is difficult to say anything about the success of the method. However, it appears that such an algorithm would not be able to capture harmonic events well, since the encoding of spectral characteristics by 40 LPC coefficients is not high enough to resolve harmonic structures. Also, the choice of a Poisson distribution is arbitrary, and no analysis of the actual distribution is performed. X. Zhu and Wyse (2004) report some problems in the isolation of transient events in cases where the events are very dense.

4.4.4 Wavelet-Based Algorithms

Bar-Joseph et al. (1999) have described a method for synthesizing sound textures based on wavelet tree learning, which uses a similar technique as their algorithm for image textures by Bar-Joseph et al. (2002). Nodes are copied into the newly created wavelet tree based on the values of ancestor nodes, so that the structure of branches resembles branches in the original tree. As an additional requirement, the sequential values along the time axis are taken into consideration. The wavelet tree is composed of Daubechies wavelets. The authors claim that their method has advantages over other methods, mostly because it is built upon a simple mathematical principle across different scales and does not require separate treatment of events and structural information.

There are some pitfalls in this wavelet-based approach. As pointed out by Hoskinson (2002), almost all examples of sounds synthesized with this method contain severe continuity errors. The specified similarity tolerance value for suffix trees has a strong influence on the result: if the tolerance is too big, the synthetic audio becomes random and has discontinuities. If the tolerance is too small, the input is repeated identically. Another problem is that the search for candidate wavelets is very slow and the search space grows quadratically with the length of the audio. Of course, the search can be limited, as proposed by the authors, but this takes away most of the essential concept.

The promise of the wavelet method is that the wavelets are useful atoms, and that a combination of such atoms will sound like a valid variation of the original input. The hierarchy in the wavelet tree should allow for phenomena in different frequency bands to be treated independently. But there lies another problem: the orthogonal wavelet decomposition used by Bar-Joseph et al. (1999) is not narrow-band, as it would be the case for a Fourier transform. Wavelets in the lower branches of the wavelet tree are instead wide-band components and are therefore almost never useful atoms for sound objects that are located in particular frequencies[7].

4.4.5 Feature-Based Synthesis

Hoffman and Cook (2007) have taken a radically different approach to the analysis-synthesis problem. Their FeatSynth system does not re-use data from the original input sound to create a new sound. Instead the new sound is synthesized from scratch, using a purely parametric model. This has many advantages: the parametric model can be very compact, synthesis parameters are relatively easy to control and artifacts

[7] For dyadic wavelets and a signal sampled at 44100 Hz, all frequencies between 689 Hz and 1378 Hz would end up in one frequency band, which is problematic for most harmonic sounds.

from cutting and re-arranging, common in other methods, can be largely avoided. The task is to find the parameters that produce an acceptable texture, i.e., one that has similar perceptual properties as the input texture.

The proposed system is modular, so that the synthesis model, the perceptual feature extraction and the difference metric can be changed individually. But as elegant as the method is, the problem of finding the synthesis parameters to match a desired output sound is a difficult search problem and requires an iterative loop of synthesis, analysis and adaptation of the parameters. The mapping from synthesis parameters to perceived perceptual aspects is not known for complex, nonlinear combinations of features, and the authors state that future work should concentrate on "finding explicit models of the relationships between synthesis parameters and features to reduce our dependence on expensive iterative optimization" [p. 185]. They propose a genetic search algorithm to search for an optimal solution, given the parametric synthesis model, perceptual features and distance metrics.

While the current implementation of the FeatSynth software[8] is able to approximate some simple sounds, it typically fails to approximate more complex sounds. One reason may be that the currently implemented parametric synthesis model does not offer the necessary complexity for sounds with a noticeable temporal structure[9]. But even in cases where it does, the genetic search can suffer from the usual problems of genetic algorithms to converge to a global optimum (see Subsection 3.2.8).

4.5 Comparison of the Methods

Looking at the different approaches for sound texture synthesis discussed here, some differences become apparent. Tab. 4.1 presents an overview of the sound texture analysis and synthesis algorithms presented in this chapter, and lists the intended scope, the model concepts, the type of analysis (automatic or manual), and the main drawbacks that can be identified for each method.

4.5.1 Differences in the Goals

The working definition of "sound textures" differs significantly between authors (see Section 1.2), which makes it difficult to compare them side by side. A comparison of the outputs of all algorithms based on the same input would be helpful, but would have to be done with audio material that all of the authors would agree on as valid "textural" sounds. "Rain" and "crowd noises" are among the most widely used examples, so they would be a candidate for a direct comparison, provided that implementations of each algorithm could be obtained. It should also be noted that not all authors use the term "sound textures", but occasionally "audio textures" (Lu et al., 2004) or less specific terminology, like "scenes" (Misra et al., 2006).

[8] FeatSynth, v0.1.0, `http://featsynth.cs.princeton.edu` (last visited: December 1, 2010)

[9] FeatSynth is built to be modular, however, only very few modules have been implemented, so far.

ALGORITHM / SOFTWARE	INTENDED SCOPE	MODEL	ANALYSIS	DRAWBACKS
Saint-Arnaud and Popat (1997)	large crowds, rain, fishtank bubbles	concatenation of "time-frequency atoms", sampling from density functions of a feature space	AUTOMATIC	does not preserve long-term structures well, frequency resolution too low for harmonic phenomena
Di Scipio (1999)	"rains, thunderstorms and more articulated phenomena of acoustic turbulence" [p. 109]	iterated nonlinear functions	NO (only synthesis)	experiemental method, very difficult to obtain desired results
Bar-Joseph et al. (1999)	rain, waterfall, fire, traffic noises, people babble, machine noises	nodes from a multi-resolution wavelet tree are copied to an output tree	AUTOMATIC	audible repetitions, quadratic complexity of the algorithm
Hoskinson (2002)	wind, animal cries, traffic noises [p. 24]	concatenation of wavelet-grains, Markov process	AUTOMATIC	true statistics of grain transitions not captured, no separation of frequency bands
Athineos and Ellis (2003)	crackling fire, running water, applause	encoding of spectral frames with LP coefficients, encoding of temporal aspects with a second set of coefficients	AUTOMATIC	only direct encoding of sound without variation, no treatment of harmonic sounds
Lu et al. (2004)	background music, lullabies, game music, screen saver sounds, horse neighing, rooster crowing, thunder, explosion, raining, stream, ripple, simple music clips	concatenation of blocks of different lengths, based on transition smoothness	AUTOMATIC	true statistics of grain transitions not captured, not suitable for harmonic sounds
Parker and Behm (2004)	ocean, wind, engines, rain, cheering crowd	concatenation of blocks of different lengths, based on transition smoothness, additional smoothing	AUTOMATIC (block size has to be specified)	true statistics of grain transitions not captured, requires fine-tuning of parameters

X. Zhu and Wyse (2004)	crowd sounds, traffic, wind, rain, machines, typing, footsteps, sawing, breathing, ocean waves, motors, chirping birds	foreground events generated from clusters in the LPC domain, placement according to Poisson distribution, background encoded with LPC	AUTOMATIC	true event distribution not analyzed, not suitable for harmonic sounds, some problems in the event isolation
Misra et al. (2006) (TAPESTREA)	complex acoustic scenes (e.g., fireworks)	sampled sound templates with explicit placement patterns, separate model for background residue	MANUAL (manual selection of elements, assisted sinusoid tracking)	no learning of sound element patterns, limited possibilities for creating element variations
Dubnov et al. (2007) (Audio Oracle)	not specified, bird song used as example	automaton used to define valid transitions between blocks	AUTOMATIC (degree of innovation has to be specified)	feature representation not specified, advantages over other block-based methods not specified
Hoffman and Cook (2007) (FeatSynth)	"non-phonorealistic sounds"	modular synthesis, configuration obtained using a genetic algorithm	AUTOMATIC	expressivity of implemented modules is limited, genetic algorithm not guaranteed to converge

Table 4.1: Comparison of the sound texture synthesis and analysis algorithms discussed in this chapter.

One particular thing can be observed in most of the analysis-synthesis algorithms discussed in this chapter: they mostly ignore harmonic signals (see Tab. 4.1), either by excluding them from the definition of sound textures, or by using examples that are heavily biased towards noisy sounds. Most authors imply that environmental sounds, being neither speech nor music, are noisy. While this is certainly true in many cases, e.g., for ocean waves, wind or rain, there can be a number of harmonic sources in textural sound, in particular sounds from man-made machinery and non-speech vocalizations from animals and humans. Harmonic sounds require processing techniques that preserve their grouping of harmonic partials, their phase continuity and their stability over time. They have more complex dependencies in the time-frequency plane than most noisy sound sources. Therefore, algorithms that are based only upon strictly local distribution of energy will often perform poorly for harmonic sounds.

4.5.2 Wide and Narrow Definitions of "Sound Texture"

Since the term sound texture is not used very consistently in the audio processing community, an argument could be made that its restriction to only fine-grained structures is just as legitimate as any more wide definition. The term would then relate to something for which processing algorithms are already available. That definition is too narrow, and it is being used for the wrong reasons.

When studying available publications about sound textures, it appears that many authors start from a fairly general concept, which is defined by statistical properties or the attention-span principle (see Subsection 1.2.5). Lu et al. (2004) even include lullabies and game music. Di Scipio (1999) includes thunderstorms, yet, both do not test their algorithms with such signals: when it turns out that the particular algorithm is unable to satisfy important aspects of any wider definition, it appears that the authors fall back to the narrow definition as a convenient way to fix the problem. This has led to harmonic sounds being dropped from the definition, without giving good reasons. A definition according to which rain is a texture, but wind chimes are not, seems artificial and of little use. The narrow definition is proposed from the side of the technical implementation, rather than from the view of perceptual psychology, or from the side of physical phenomena.

Most authors limit the term to fine-grained structures, in particular those that have an attention span in the 100 ms range. Accordingly, the existing algorithms discussed here mostly fail to capture characteristics of anything outside the narrow definition (see Subsection 1.2.6), especially those based on block-based processing or wavelet trees.

The manual approach implemented in TAPESTREA does not have these limitations, but no attempt has been made yet by the authors to construct textures automatically from the inputs. Some other approaches, like genetic algorithms, could potentially produce more complex textures without the limitations of block-based processing. However, the genetic algorithm by Hoffman and Cook (2007) has only been used for low-complexity sounds so far, and the authors have not stated what kind of feature space would be necessary in order to enable them to produce complex textures. The problem of guessing the right parameters from the inputs is also the main problem for some of the more experimental methods, like the iterated nonlinear functions (see Subsection 4.3.1) or procedural algorithms (see Subsection 4.2.2).

4.5.3 Object and Element Concepts

Looking back at the various methods examined in this chapter, the likely reasons can be found why most methods fail to capture the properties of the input audio at large scales. Local statistics of a signal are often reproduced quite well, while the structures on higher levels of the organizational hierarchy of the textures are lost. The simple answer to the question why the proposed algorithms cannot capture the large-scale structures is: because they do not try. This is especially true for the algorithms by Lu et al. (2004) and by Hoskinson (2002), which are based on purely sequential concatenation and have no sense of hierarchy. The wavelet-based method — although introducing some new problems — overcomes these limitations to some degree, but constructs the sound texture from wavelet components that have no direct connection to anything in the real acoustic world[10].

Going back to the wavelet-based synthesis by Bar-Joseph et al. (1999), an argument could be made that is employs a weak, implicit model of objects: the wavelets themselves. In contrast to that, the object-oriented approach is very explicit and visible in TAPESTREA. Objects are inserted according to placement patterns, they can vary and they can overlap with other objects. They can even have names assigned to them, making it easy for the sound engineer to organize the acoustic scenery in an understandable way. The object paradigm appears to be very helpful for analyzing and synthesizing sound textures with complex hierarchies and clearly audible structures, and appears to be a promising approach to extend sound textures beyond the domain of primitive granular phenomena.

4.5.4 Difference Approaches for Assessing Output Quality

The perceptual similarity of a sound texture to its original input is a central requirement for a useful algorithm, and it is also one of the most difficult concepts to define. Across various publications, it appears that there is little agreement on a similarity measure. Similarity is expected to emerge from certain properties of the algorithm: If the original building blocks are used for the construction, the result must be similar to the input. But is this true?

Similarity in auditory perception is a non-trivial concept, which involves not only the right amount of energy in frequency bands, but can relate to just about any detail of the signal, from transients and phase alignment to harmonic relations, noise, echos, continuity and high-level statistics. The development of algorithms therefore will have to be accompanied and refined by research in perceptual psychology, much more than it has been done in the past. X. Zhu and Wyse (2004) explain that “it is not easy to quantitatively measure the dissimilarity between the generated sound and the sample audio principally due to the statistical variation in the model” [p. 348]. Athineos and Ellis (2003) state:“A difficulty in devising an optimal solution, however, is the poorly defined criterion of perceptual quality: No single error analysis window adequately captures the perceptually salient properties of the resynthesis [...]” [p. 651].

The statistical properties of the temporal composition are among the most important features of any sound texture. They are reflected in the behavior of most existing

[10] Bar-Joseph et al. (1999) claim that “a principled mathematical approach to granular analysis and resynthesis is possible”, and that their approach is “the first approach to sound texture analysis/resynthesis that does not assume an implicit sound model” (p. 47). Whether this is really an advantage seems uncertain, given the insufficiencies of the synthesized output.

sound texture algorithms to control the placement of textural elements in some way, mostly through probabilistic methods of sequential ordering [(Bar-Joseph et al., 1999), (X. Zhu & Wyse, 2004), (Lu et al., 2004), (Saint-Arnaud & Popat, 1997)]. But not every algorithm derives the probabilities from actual statistics of the input signal: in the algorithm by Hoskinson (2002), the smoothness of continuity drives the combination of the texture, even though smoothness may not be a feature of the input sound.

4.5.5 Implicit and Explicit Aspects of Texture Models

This chapter has given an overview of existing algorithms for sound texture processing that are all based on different conceptual models of sound. This is most evident in the types of "building blocks" which are used. While the TAPESTREA system is mostly object-based, the automatic methods reviewed here tend to use grains and blocks. The other respect in which these methods differ is the strategy according to which the elements are arranged. In TAPESTREA, users of the system can specify how many objects per time unit should be used. In the grain and block-based approaches, the algorithm is based on local concatenation, and thus has no concept of the frequency of occurrence of particular objects. Yet, a frequency of occurrence emerges as a result from the local concatenation.

Leaving aside for a moment the obvious difference that TAPESTREA is a manual tool, the two approaches described above differ primarily because the grains and blocks are *implicit* models, while TAPESTREA uses an *explicit* model. In the explicit model, the sounds carry labels and the statistics of the texture are stored and presented separately, in a form that is quite easy to read for the user (an equivalent of "insert 100 randomly placed bird-tweet sounds per minute"). The explicit model corresponds roughly to a top-down approach. The block-based approach cannot explicitly store the concept of a bird tweet, since it does not assign labels, however, it can have blocks that are bird tweet sounds, or fragments of them. Looking at the random-based concatenation method of the block-based approach, it is also difficult to recognize how many elements of a particular kind will be inserted per minute. Instead, the statistical distribution is merely implied by the statistics of the source signal.

In theory, both approaches could produce almost identical outputs. But in explicit models, it is much easier to see, and possibly easier to influence, how elements will be placed. Explicit models are also less likely to produce unstable outputs or to be trapped in loops, as it can easily happen in the case of local concatenation. The main disadvantage of explicit models is that they are difficult to create automatically from input recordings, because this would involve the assignment of labels and the discovery of rules.

4.6 Chapter Conclusions

One important question is whether a fine-tuning of parameters, the consideration of a larger search space or a higher resolution can solve all or most problems currently observed in sound texture analysis-synthesis algorithms. Some authors, including Strobl (2007, p. 62), are confident that a near perfect result can be obtained without making fundamental changes:

> "I still consider ocean waves and wind signals as sound textures. However, using the parameters proposed in this thesis, the algorithms do not capture the properties of these signals. I am sure that using very large analysis parameters might succeed but this was not tested in the context of my work."

This statement might turn out to be too optimistic. Although some improvements can surely be reached by increasing the range of searched parameters, many of the limitations seem to be founded in the algorithms themselves, rather than in the parameters they are run with. For example, if identical repetition is to be avoided, searching for a matching segment to repeat undermines that goal. If independent phenomena are present in different frequency bands, making a vertical cut through the spectrogram is likely to cause some damage. These problems will not go away completely by increasing the amount of input data or making the search for matching parts more exhaustive. In the best case, the pool of available input audio would be so big that a good match can be found for almost any state of the synthesis: if the amount of available types of blocks approaches infinity, the probability to find a matching block will approach 100 %. But for any real implementation, which has to produce useful results with only minutes of input audio, this condition can never be satisfied, and limitations in processing power often impose strict limits on the range of parameters that can be searched.

Instead of artificially restricting the scope of the definition, it seems more appropriate to openly address the challenges that will have to be solved in future sound texture applications. After all, the research goal of synthesizing much more complex sceneries does not go away by excluding it from the terminology. The wide definition of sound textures is used as the basis in this thesis (see Subsection 1.2.6). Building upon the existing algorithms discussed here, some techniques will be demonstrated in the following chapters that could overcome many of the existing limitations, such as the inability to produce truly unique variations of sound objects, or the inability to identify the temporal patterns of the input. At the heart of the proposed techniques is object-based sound processing.

As described already in Subsection 2.2.8, the object-based view on sound — as opposed to the block-based view — appears to be particularly rewarding: sound objects may be assigned to sound sources located in 3D space, they may overlap with other objects, and they can be treated as instances of an abstract class of sounds. This matches very well with the ecological view (see Chapter 2) of an acoustic environment, which always deals with the sources around the listener.

Chapter 5

Proposed Sound Texture Description Model and Workflow

In this chapter, a workflow model for sound texture processing is proposed. Starting from the requirements that were stated in Section 4.1, the problem is broken down into a chain of technical aspects that need to be addressed, and a general concept of a synthesis-by-analysis framework is derived.

5.1 Technical Aspects of the Sound Texture Problem

There are very different approaches to the sound texture problem. What is described as an essential concept for some methods may be almost irrelevant for others. These differences are particularly large between methods with a more high-level object paradigm (e.g., Misra et al. (2006)) on the one side and a low-level sample/wavelet paradigm (e.g., Saint-Arnaud and Popat (1997), Hoskinson (2002), Lu et al. (2004)) on the other. Still, some technical aspects can be identified that have to be addressed regardless of the paradigm. For some aspects, approaches for future implementations are briefly discussed in this section.

5.1.1 The Abstract Workflow of Sound Texture Processing

To shed some light on the different concepts of sound texture processing, and in order to propose a good solution for improved implementations, it is useful to provide an abstract workflow of processing steps, a description that can be used as a reference for any sound texture processing algorithm. All sound texture processing frameworks discussed so far can be broken down into the following steps:

1. Element identification
2. Element grouping
3. Element variability analysis
4. Distribution pattern learning

Although the sequence of these exact steps is certainly not obvious in all cases, an argument can be made that each step is necessary. Without having identified the elements, it is impossible to compare any two of them. Without comparing elements, it is impossible to make any judgment about their similarity and whether they are instances produced by the same sound source. Without knowing which elements originated from which source, it is impossible to say what distribution patterns are characteristic for the source. The four steps will now be explained in more detail.

5.1.2 Step 1: Element Identification

A representation of a sound texture is a statistical model of auditory phenomena, the variations with which they occur and their patterns of distribution. This very general description is based only on the concept that sound textures have something that repeats, a thing which we may want to call *elements*, *parts*, *blocks*, *components*, *fragments* or *objects*. The choice of words can indicate something about the scale or layer of abstraction; while the use of the word "fragment" indicates something very tiny, possibly in the range of a few digital samples, the word "block" indicates a slightly longer structure. The word "element" will be used in the following paragraphs to refer to the general class of all these concepts, regardless of length or conceptual nature.

The identification of elements, i.e., the precise analysis of where they start and end, and what energy in which frequency bands belongs to them, is the first important part of the processing chain, and is also one of the most difficult. Sound texture analysis requires a blind identification of elements, in which the nature of the elements is not known beforehand (see Subsection 3.6.4). Therefore, it is not possible to use simple matching algorithms to search for elements. Instead, more elaborate principles, both from signal analysis and perceptual psychology (see Subsection 2.2.9), have to be used to determine the most likely *onsets* and *offsets* (the start and end points) of each element.

Problems arise from the fact that recordings of textural sounds often contain a mix of many sources, overlapping in time and frequency. In that case, it is not sufficient to determine the onsets and offsets of elements, it is also necessary to "untangle" them from the other elements and background noises. As discussed in Section 3.6, sound separation for overlapping sources is difficult to achieve, and no methods of sufficient quality are yet available for the separation of arbitrary sources. For the processing of naturalistic sound textures, this is quite problematic, because further processing steps depend heavily upon the identification of separate streams and elements. As shown in Section 4.4, most algorithms do not attempt to achieve element identification or separation based on perceptual principles, but decompose the inputs according to much more simplistic principles. For texture types in which the elements do not overlap strongly, useful results could be expected using linear sound segmentation (see Subsection 3.6.1). The result of successful element separation would be a collection of elements whose properties and precise locations in the original audio recording are known.

The identification of elements can be followed by a mapping into a parameter space, so that it can be more easily compared to other elements. Estimating parameters could also be part of an iterative element identification procedure, which improves both the parameters and the identification of an element in the form of an expectation-maximization (EM) strategy (Dempster, Laird, Rubin, et al., 1977). However, this is not currently done in any of the sound texture processing algorithms discussed in this thesis. The ability to compare elements in a feature space serves an important purpose, because it provides the basis for the task of variability analysis.

5.1.3 Step 2: Element Grouping

The correct placement of elements to use in a synthesized sound texture, and also the variability of these elements, can only be known by examining the input recording. Statistical analysis can then show if individual elements should be placed in short succession or far apart, if they appear according to regular patterns or random distributions, if they are independent from other elements or correlate with other patterns. But all of these analyses need groups of elements: it is in the nature of statistics that no patterns can be derived from a single occurrence of anything.

Technically, the grouping of elements is a problem of clustering, i.e., a problem of *unsupervised* learning. The number of clusters in a sound texture is not known beforehand: given a recording from a nature scene, the number of distinct sound producing sources or sound types is often far from obvious. To an observer with an intuitive knowledge about sounds, the clustering will be successful if all sounds belonging to the same sound type end up in the same cluster, and different sounds end up in different clusters. However, as discussed in Subsection 4.1.1, the notions of similarity, distance and identity are very difficult to formalize with respect to acoustic phenomena.

For some domains, external knowledge about classes is available, such as the fixed set of phonemes used in linguistics to model speech sequences (International Phonetic Association, 1999). Apart from such expert categorization, sounds can sometimes be assigned to classes based on the different sources that produce them, assuming that these sources are known. However, with respect to the sound texture problem, the definition of classes is not necessarily a problem of semantic correctness, but of acoustic appropriateness: a set of classes is appropriate if it enables the modeling of structures that sound right. Also, an algorithm to create and learn sound textures for arbitrary scenes should not depend on external domain knowledge, as this would limit the possibilities of application drastically.

In a parametric sound model with real-valued parameters, a common distance measure is the Euclidean distance between points in parameter space. However, the axes of the space have different units and scales. Some relate to frequency in Hertz, others to amplitude or length. Additionally, frequency perception is non-linear: the perceptual difference between 100 Hz and 200 Hz is much greater than the perceptual difference between 10 100 Hz and 10 200 Hz (see Subsection 2.2.3). Without knowledge about human hearing, it is impossible to define perceptual distance measures. Parameters have to be weighted and combined — possibly with non-linear factors — into a coherent distance measure. Methods for defining distances between sounds and timbres have already been discussed in Subsection 4.1.1.

Provided that useful distance measures are identified, unsupervised clustering is a procedure for which many standard algorithms are available, such as k-means clustering (MacQueen et al., 1967), which belongs to the group of *expectation maximization* (EM) algorithms (Dempster et al., 1977). Such clustering can iteratively partition objects into k clusters, such that all objects within the same cluster are very similar, while objects in different clusters are dissimilar.

Since the number of clusters k is typically not known when processing arbitrary recordings of acoustic environments, a standard method would be to try increasing numbers of k and stopping at a configuration that provides little errors, but keeps the number of clusters small as well. K-means clustering assumes that clusters are roughly convex shaped, a requirement that is not always fulfilled by natural sound sources.

Other methods of clustering include density-based clustering (Ester, Kriegel, Sander, & Xu, 1996) and hierarchical clustering (Hastie, Tibshirani, Friedman, & Franklin, 2005). Such methods can solve some of the problems of k-means clustering, but at the cost of introducing additional complexity and requiring yet more parameters to be set in advance. Of the algorithms discussed in Section 4.4, only the algorithm by Saint-Arnaud and Popat (1997) performs a grouping of elements.

5.1.4 Step 3: Element Variability Analysis

Each group of sound elements that was formed in the grouping step now contains different expressions of a class of sounds. Such instances could be phonemes, isolated from a speech recording, sounds of doorbells or the sounds of individual cars passing by. These instances may be very similar, or quite different: While the same doorbell will produce an almost identical sound again and again, different cars passing by may have very different acoustic characteristics, depending on their different weight, engines and other properties. For textures that have a strong variability of instances — like the cars passing by — it would be disturbing to repeat the same sounds over and over. Variation is necessary. It can be achieved to some degree by selecting instances from the original input at random. As shown in Section 4.4, this is the principle that current analysis-synthesis algorithms for sound textures use most. The instances in these algorithms are sometimes wavelets, sometimes grains or blocks.

But re-using instances from the original input has a serious disadvantage: it implies that the observed examples are the only existing variations of that type of sound. It also implies that the time of observation was just long enough to see every possible instance once, but just short enough so that they do not repeat. This, obviously, would be a suspicious coincidence. When observing a number of instances, and finding that they are all different to each other, the more intuitive assumption is that what has been observed is a subset of a far greater variety; if the observed instances were really the only ones that occur in the texture, it could be expected that they are observed not only once, but multiple times. There may be sounds that occur only once in the input recording. In that case, any assumption about variation is arbitrary. For lack of statistical evidence, the designer of a sound texture algorithm may wish to exclude such single events from the texture, or otherwise may assume that they always appear exactly once within the observed time span.

No matter how long the time of observation, i.e., the length of the input recording, is, it will likely only contain a small subset of all possible variations of a sound that can occur. However, the subset has statistical properties that can be used to learn — with some uncertainty — what the prototypical essence of each cluster is, which properties vary among the instances, and in what range. To discover the important properties that define the nature of a cluster, the representation of sound as a buffer of digital samples is not appropriate, because the relevant hidden variables, such as the fundamental frequency, are not immediately accessible. Sound clustering requires a specialized acoustic parameter space for analysis and synthesis, which is introduced later in this thesis.

As mentioned above, variability analysis is not included in any of the existing sound texture analysis-synthesis algorithms. Instead, they repeat elements from the input recording. Still, acceptable synthesis results can sometimes be reached with this methods, provided that enough different instances are available. The listener will then hear repeating instances from the input without noticing it. That way, the problem cannot be solved, but it can sometimes be concealed.

In the next processing step, a combined model of parameter distributions is learned for each cluster. It is assumed that the instances in a cluster contain a representative distribution of the parameters of a sound class, from which a model of common and less common representatives of the class can be formed. The analysis should also consider that many parameters can be correlated for a given sound class.

One method of analyzing typical parameter distributions and correlations between parameters is *principal component analysis (PCA)*, which finds the Eigenvectors in a covariance matrix of the input data (Schölkopf, Smola, & Müller, 1997). The rationale behind this is that in spite of the relatively high-dimensional parametric model, most sound types have a low number of actual, hidden control parameters. For example, the sounds of piano keys from one particular piano do not vary in arbitrary dimensions. Instead, they can be defined primarily along two dimensions: the key pitch and the force with which it was struck. Since force correlates with volume, the hidden force parameter will influence the amplitudes of all partials, and therefore all coefficients of the filter envelope. But the force of hitting a piano key also has an influence on some other, more subtle characteristics. Such correlations are reflected in the covariance matrix, which serves as a basis for the PCA.

Discovering that model parameters are correlated brings a number of advantages: (1) it provides a convenient way of compressing high-dimensional information to fewer parameters, (2) it allows for the direct manipulation of hidden parameters, thus simplifying the control of acoustic properties, and (3) it prevents the configuration of unnatural parameter combinations.

If regular PCA is used, the principal components can only be linearly correlated with coefficients. This is good enough for a wide range of acoustic principles. For example, the correlation of loudness with a set of amplitude parameters is a physically plausible assumption. However, linear combinations are less useful for quadratic, logarithmic, or periodic correlations with hidden control parameters. Some types of sounds may also have several distinct sub-types, with no valid intermediate states. An example for this would be an industrial machine that only produces very specific sounds, or a bird call that consists sometimes of three, sometimes of four vocalizations[1]. For such a non-linear or even non-continuous case, a linear dependency of parameters would be wrong, thus leading to a synthetic output that deviates from the characteristics of the input.

There are methods of subspace analysis which are able to detect non-linear subspaces in high-dimensional data. One of these methods is *kernel PCA*, which combines a non-linear warping of the feature space with regular PCA (Schölkopf et al., 1997). Other methods include *independent component analysis (ICA)*[2] (Jutten & Herault, 1991), *independent subspace analysis (ISA)* (Póczos & Lőrincz, 2005) and the analysis of slow varying features in motion patterns (Wiskott & Sejnowski, 2002). An approach for implementing PCA-based dimensionality reduction for sound elements is shown in Appendix C.

[1] A mismatch like that could be an argument to put these sounds into different classes.

[2] Originally called INCA.

5.1.5 Step 4: Distribution Pattern Learning

Another aspect which is relevant for each group of sound elements is the pattern according to which the sounds occur in the texture. Such patterns can have aspects of regularity or randomness, they may be sufficiently described by density models or may require the precise description of sequential rules or grammars. The true nature of these patterns cannot normally be discovered with certainty, because nothing is known about the causality. Nevertheless, by observing a sufficiently large sample of sequential items, rules can be extracted with some statistical degree of certainty.

From a technical point of view, discovering rules is a search problem, in which an algorithm can only find rules that are contained in its search space. For example, an algorithm cannot discover rules about musical rhythm and meter if it is not designed to look for them, and it will likely draw false conclusions for a pattern that continually accelerates, unless acceleration is a feature that is explicitly handled. A practical way to deal with this challenge is to limit the search for the intended application: algorithms designed to process nature sounds may not need to respect musical patterns. Just like in the variability analysis, singular events cause problems for the analysis of patterns: obviously, not much can be learned about the frequency of occurrence of an event that shows up only once. Excluding such outliers can be a valid way of dealing with them.

The task of distribution pattern learning does not depend on the results of the variability analysis if it is assumed that variation and placement are uncorrelated[3]. In this case, both tasks can be processed in an arbitrary order.

A pattern may be a regular, repeating sequence of sound objects, as used in a drum pattern, it may be an algorithmic rule or may be describing a random distribution. A comprehensive implementation of a sound texture processing chain could include all of these mechanisms and select appropriate strategies from the analysis of the input data: if the source signal is found to have musical rhythm, a module for repeating patterns could be used; if no regular structure is found, a random-based mechanism could be selected.

Some sound sources produce sound sequentially and without any overlap. This is true for a speech recording with a single speaker, in which one phoneme is produced at a time. A similar case would be a flute, which can also produce just one note. Pauses may be allowed in this model, since they can be represented as a special type of event. Markov chains can be used to model such sequential phenomena. The central assumption is that the next element in the chain depends only on a limited number of predecessors. For a 1st-order chain, this would be just the immediate predecessor. For a 3rd-order Markov chain, the next element is chosen based on three predecessors (Rabiner, 1989). Anything further in the past does not influence the probability of what the next element will be. This matches well with the concept of attention span, discussed in Subsection 1.2.5. Of course, many sound sources in the real world do have structure and causality that spans much longer temporal dimensions, but the assumptions made for the sound texture case allow for cutting away much of the complexity and limit the search space to dependencies within the attention span (See Subsection 1.2.5).

[3] There could be correlations between sequential placement and model parameters, for example in cases where animals produce sequences of tones with rising pitches. However, such structures are not taken into account in the proposed framework.

Many sound sources are not sequential, especially if the observed sound instances originate from independent sources. For example, in a sound texture of rain there is no physical justification for a sequential model: one rain drop is not sequentially linked to another rain drop that goes before it, and their sounds of hitting the ground may very well overlap. In that case, a parameter that expresses the expected number of instances per second or per minute is more useful. Ideally, the choice whether to use a sequential or non-sequential model should be made automatically.

5.2 A Hierarchical Description Model for Sound Textures

In Section 4.5.3, some advantages of object-based sound textures have been listed. Starting from these assumptions, the goal is to conceptualize a texture model that works with flexible objects, synthesizing unique instances of them as the texture is generated. Although a framework for synthesizing textures from this model has not been implemented in the course of this dissertation, the description model is presented here as a frame of reference for the sound element model that belongs to it. The hierarchical components of the description model will now be presented in detail, starting with the main class: `Texture`.

A `Texture` consists of one or more tracks (of the class `Track`), which allows it to store uncorrelated phenomena separated from each other. During synthesis, the outputs from each track can simply be added together. While, in theory, a single distribution pattern could handle the placement of uncorrelated phenomena as well, having the additional concept of tracks simplifies the distribution patterns and gives better control over the synthesis. While simple textures — like rain — will most likely have only one track, complex sceneries benefit from using multiple tracks. In a "forest" texture, the voices of different birds could get their own tracks, while atmospheric sounds could be stored in other tracks. This has the additional benefit that tracks can be switched off, changed in volume or placed at different locations in the surround or stereo field. Fig. 5.1 shows a UML component diagram for sound textures.

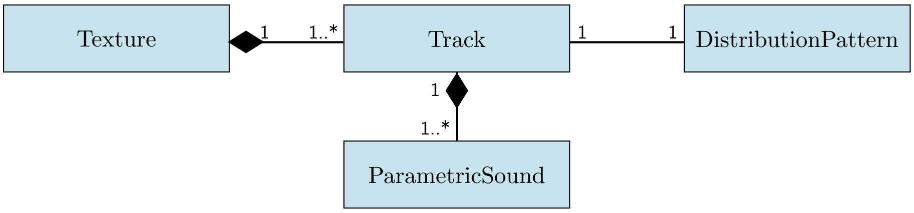


Figure 5.1: UML component diagram of the sound texture model: A texture can have several tracks. A distribution pattern is associated with each track. A track can have one or several parametric sound objects associated with it.

The *parametric sound objects* (represented in the class `ParametricSound`), which are in the primary focus of this thesis, are the fundamental components of the flexible sound texture model. Being at the bottom of the hierarchy, they represent the components of the model that produce sound output. They receive instructions from the upper layers, and do not control any other objects themselves.

In addition to a description of the "atomic" objects, the texture model requires rules to arrange many objects in sequences and characteristic *distribution patterns*. These

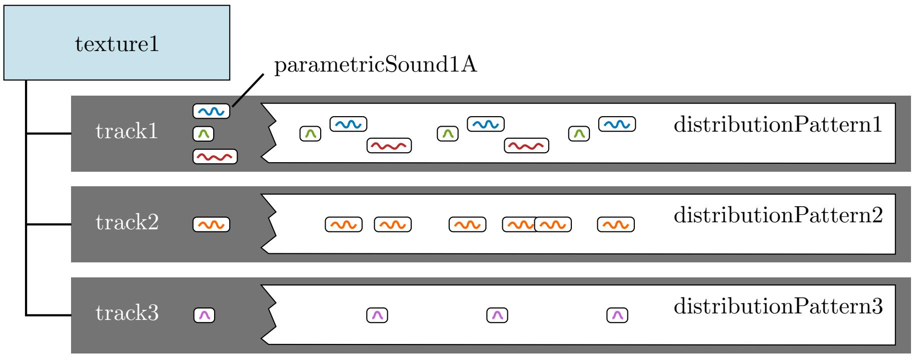


Figure 5.2: Example structure of a sound texture with three simultaneous tracks. Each track stores its own sound objects and its own distribution pattern.

patterns (of the class `DistributionPattern`) cannot be properties of the individual sound objects themselves, because a pattern may describe rules of sequences between different objects. For example, a sound texture of a snoring person could consist of two main sound objects, the snoring sound for breathing in, and another sound for breathing out. The corresponding distribution pattern would need to reference both these sounds and specify that they should be generated in an alternating pattern. Simple distribution patterns may consist of only one type of sound. Different sound objects are only parts of the same pattern if their placement has to be correlated. Note that the term *distribution pattern* is used here as a placeholder for a wide range of possible implementations (see Subsection 5.1.5). Phenomena within the same distribution pattern are assumed to be connected by some causality or underlying stochastic process.

Since a distribution pattern is independent from other patterns, a simple implementation of the `DistributionPattern` module does not need to process any inputs, except for some control signals that specify what time range of the texture should be produced. The outputs of the module are control signals sent to the sound objects to trigger the synthesis of new audio samples into an output buffer. But a more complex implementation could include higher-level control inputs to the tracks. For example, a distribution pattern for road traffic could offer a “time of day” controller, which would have an effect on the frequency of cars going by. Fig. 5.2 contains an example of the resulting data structure.

Chapter 6

Implementation of the Sound Element Analysis and Synthesis

At the heart of the new sound texture model described in Chapter 5 is the parametric sound object model that is the main subject of this thesis. The implementation of a parametric sound object for tonal sounds, which uses spline curves to encode the trajectory of a fundamental frequency and coefficients to model a time-frequency envelope, is described in detail in this chapter, where also advantages and disadvantages of the model are discussed.

6.1 Overview of the Parametric Sound Model

As already explained, the proposed sound texture framework uses parametric sound objects to encode the basic elements of a sound texture. The proposed parametric model, which will be referred to as the *parametric sound object synthesis (PSOS)* model, can be seen as an extension of the spectral modeling paradigm (See Section 3.4). The main difference is that the temporal dimension is modeled in a coherent set of parameters, while the standard spectral model represents signals as a concatenation of frames. The proposed model is object-based: the set of parameters encodes properties of one single sound and is independent of the sound's length.

In the original spectral model, all sinusoids are encoded as independent trajectories, stored as a sequence of frequency-phase-amplitude triplets (see Section 3.3). In the proposed model, the frequency information is decoupled from the filter characteristics of the signal. The time-varying fundamental frequency is encoded as a smooth B-spline. The time-varying loudness of the partials is encoded as a spectral envelope plane. Separating the fundamental frequency from the filter characteristics of the signal provides a great amount of flexibility for further signal transformations. The same concept is also used in speech codecs to compress the transmitted audio signal more effectively, or in the vocoder, where it facilitates a number of sound manipulation effects (see Subsection 3.2.11).

The separation of the fundamental frequency from filter characteristics is not only practical, it is also closer to the physical reality of sound sources in the real world. The physical properties that damp or amplify portions of the frequency spectrum exist regardless of the current fundamental frequency being produced. In addition to the harmonic signal content, the noisy residual of the signal is encoded by another spectral envelope plane. During synthesis, the envelope is used to shape a white noise signal.

In the following section, the B-spline model will be explained. The next sections deal with the encoding of the fundamental frequency, the encoding of the harmonic spectral envelope and the noise spectral envelope, each of which makes use of the B-spline model.

6.2 B-Spline Coding

In the PSOS model, spline curves are used in several places to encode time-varying parameters, such as the fundamental frequency or changes in the spectral envelope over time. Splines provide the smoothness associated with most natural sound sources and can be defined by few coefficients, which makes them especially useful (Möhlmann, Herzog, & Wagner, 2009).

Two common forms of such splines are basis splines (*B-splines*) — which are used in this work — and *Beziér splines*. They differ mostly in their choice of basis functions. The basis functions are polynomials, chosen in such a way that they add up to 1.0 at each offset. They control the influence of each point along the offset of the spline. Given a local offset u, a set of n basis functions $\mathbf{B}_i(u)$ $, 0 \leq i < n$ and a set of associated control points $\mathbf{P}_i$, the local spline value $\mathbf{S}(u)$ of the spline can be calculated as the weighted sum of the control points (de Boor, 1978):

$$\mathbf{S}(u) = \sum_{i=0}^{n-1} \mathbf{P}_i \mathbf{B}_i(u) \ , u \in [0, 1] \tag{6.1}$$

In 2D graphics, the control points (or *de Boor* points) $\mathbf{P}_i$ are 2-dimensional coordinates, but they can also be scalar values or arbitrary vectors, depending on what is to be interpolated. Within the proposed model, control points are scalar values, related to either frequency or amplitude. A spline usually begins and ends at a control point, but does not necessarily pass directly through its other control points.

Long splines can be formed by concatenating several splines and by making sure that the spline is smooth at the points of concatenation. The smoothness depends on the degree k of the spline. Linear splines ($k = 2$) are continuous between spline segments, but not smooth. Quadratic splines ($k = 3$) are continuous and smooth, cubic splines ($k = 4$) have additional continuous derivatives. The order k of a spline corresponds to $p + 1$, where p is the highest polynomial order of the basis functions[1].

The influence of a control point is largest at the offset where the associated basis function has a maximum. At positions further away from the maximum, the influence of the control point gradually decreases. It is desirable to use basis functions that are non-zero only in a limited range of the spline, so that the influence of a control point is only local. B-splines achieve this by dividing the spline into sections using so-called *breakpoints*. Each basis function has non-zero values within a span of k segments (see Fig. 6.1).

The placement of breakpoints is defined as a sequence of *knots*, which is a non-decreasing vector of offset positions (de Boor, 1978). When the distances between knots are all equal, the B-spline is *uniform*. In that case, the basis functions are

[1] This may seem confusing, but is a convention used quite consistently in the literature on splines.

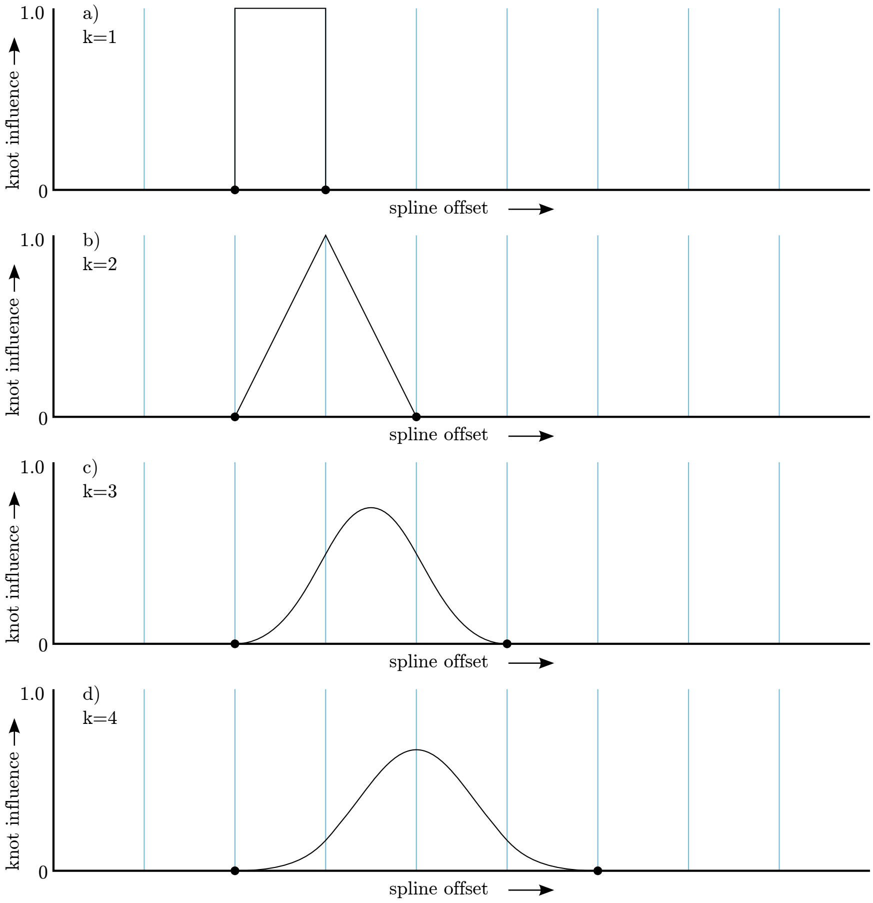


Figure 6.1: For splines of order $k = 1$, basis functions are constant and defined only in one segment (a). For $k = 2$, basis functions are linear and span two segments (b). For $k = 3$, basis functions are smooth and span three segments (c). For $k = 4$, basis functions span four segments.

identical, shifted copied of each other. For splines of order $k \geq 2$, the number of knots has to be larger than the number of breakpoints, because in that case the basis functions span more than one segment. It is typically avoided to define knots outside the interval of the spline. Instead, the additional knots at both ends, required by higher-dimensional splines, can all be set to the same offset. A B-spline of order $k = 3$ with four equally spaced breakpoints, defined over an interval [0...1], would have the knot vector $\{0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1\}$. The basis functions at the start and end are then squashed[2] to fit into the defined range (see Fig. 6.2).

The i-th basis function $\mathbf{B}_{i,k}(t)$ is non-zero between the knots t_i and t_{i+k}. For a B-spline of order k, the set of basis functions $\mathbf{B}_{i,k}(t)$ can be calculated using a recursive

[2] This means that, strictly speaking, the B-spline is not entirely uniform any more.

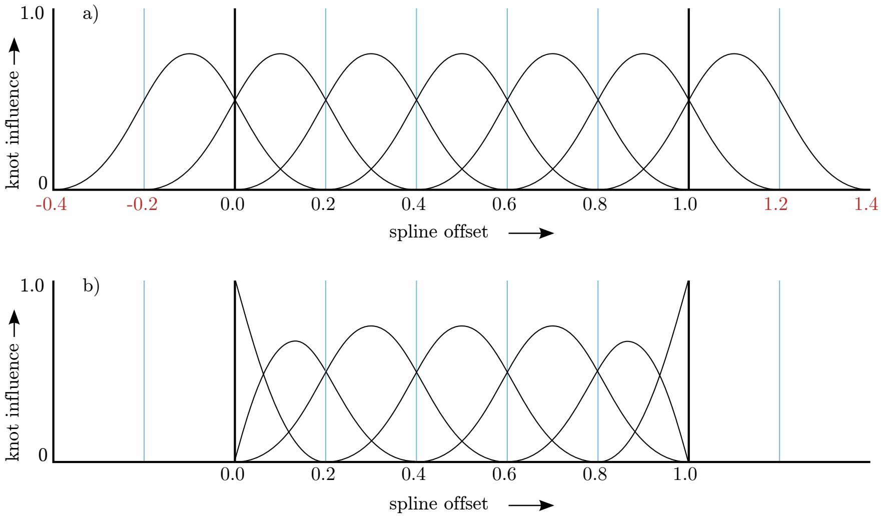


Figure 6.2: (a): at the start and end point of a strictly uniform spline, the knots are outside the range covered by the spline, and basis functions do not add up to 1.0 within the spline interval. (b): By making the first and last knots repetitions of the first and last last breakpoint, all basis functions remain within the spline range (b).

formula (de Boor, 1978, p. 131). The basis functions of order 1 are simply segment-wise constant values:

$$\mathbf{B}_{i,1}(t) := \begin{cases} 1 \quad \text{if} \quad t_i \leq x < t_{i+1} \\ \\ 0 \qquad \text{otherwise} \end{cases} \tag{6.2}$$

For higher values of k, $B_{i,k}(t)$ can be calculated as:

$$\mathbf{B}_{i,k}(t) := \frac{t - t_i}{t_{i+k} - t_i}\mathbf{B}_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}}\mathbf{B}_{i+1,k-1}(t). \tag{6.3}$$

The influence of a control point is strongest at a particular offset, where its corresponding basis function has a maximum. This location depends on the relative spacing of knots in the spline. For truly uniform splines, the maxima are located either exactly at the breakpoints (for orders 2, 4, 6, ...) or exactly half way between breakpoints (orders 3, 5, ...). However, for a non-uniform spacing of knots, this is not true. Inserting a breakpoint at a particular offset does not give direct access to the curve value at that offset, only near it.

B-Splines have some advantages over Beziér splines. In Beziér splines, all basis functions are non-zero over the whole range of the spline, so that changes to one control point have an effect on all parts of the curve. The concept of breakpoints in B-splines overcomes this problem. The polynomials for the basis functions do not have to extend over the whole range of the spline, but can be defined only for the sections in the neighborhood of the corresponding control point. Fig. 6.3 illustrates the difference. The limited, local influence of individual control points speeds up possible automatic curve-fitting algorithms and also simplifies any manual interactions with the curve (de Boor, 1978).

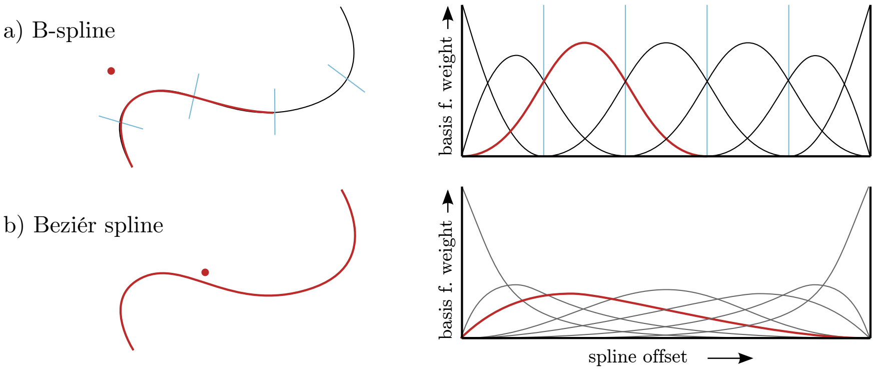


Figure 6.3: Comparison of the B-spline model (a) and the Beziér spline model (b). For the Beziér spline, the weight of the marked control point is non-zero over the whole range of the spline, while in the B-spline model, the influence is local.

6.2.1 Curve-Fitting

Automatic curve fitting algorithms are able to find the optimal parameters of a curve model, i.e., a set of spline coefficients, to match a set of data points. The goal is to find the curve for which the distance between the curve and the data is minimized in the least-squares sense. Looking at the formula for B-splines, it becomes clear that the coefficients all have a linear influence on each point on the spline (see Eq. 6.1). This implies that an efficient and near-optimal solution can be found using standard least-squares fitting techniques. Furthermore, because of the unique properties of B-splines, the influence of most coefficients on most data points will be zero. This leads to a sparse matrix representation and allows for fitting algorithms to speed up the computation (Green & Pierre, 2002).

In the implementation for this thesis, the `gsl_multifit_linear` function of the GNU Scientific Library[3] is used to compute the spline coefficients, given the pre-calculated basis functions and the data points. The function computes a least-squares fit of the spline coefficients to the observed data and is based on the *singular value decomposition* (SVD) algorithm by Golub and Reinsch (Golub & Reinsch, 1970). it is guaranteed to converge to the optimal solution within the accuracy of floating point arithmetics. Because the model itself remains fixed for all sounds, the necessary basis functions can be calculated once when the software is initialized. The result of the fitting is a set of coefficients, i.e., control point values, which represent the optimal smooth spline through the data points.

6.3 Encoding of the Fundamental Frequency Trajectory

Sounds in the proposed parametric model have exactly one fundamental frequency at any time t. In the case of pure noise, the value of the fundamental frequency is ignored and the harmonic part of the signal is not synthesized. All sinusoids that belong to the harmonic content of the signal are assumed to be integer multiples of the fundamental frequency. This greatly simplifies the representation of the signal and eliminates the

[3] `http://www.gnu.org/software/gsl/manual` (last visited: December 1, 2010)

need to encode redundant information for all partials. Although this assumption does not hold strictly in all cases, a wide variety of sound sources can be modeled with high accuracy, including human and animal vocalizations, as well as string and wind instruments. However, for sounds with strong inharmonicity, non-harmonic partial structures or modulation, this assumption is not accurate and will lead to audible differences between the input and the output.

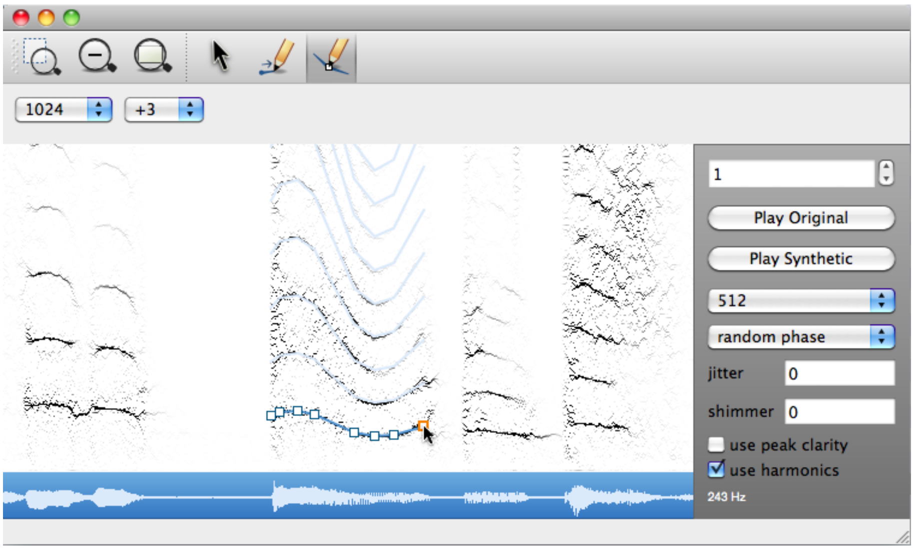
Most sound sources in nature have fundamental frequencies that evolve relatively smoothly over time. This smooth continuity is one of the main clues used by the brain to group sounds and to fuse auditory perceptions into coherent entities (see Subsection 2.2.9). This still allows for fluctuations and strong slopes in the signal, as long as they can be sufficiently well described by curve models. For the proposed PSOS model, sounds can be treated as coherent objects if they respect this continuity. A strong, sudden jump in frequency thus requires the separation of a sound into smaller sounds.

A uniform B-spline model is used to encode the time-varying fundamental frequency. The n spline coefficients c_i, $0 \leq i < n$ are values in the frequency domain. The knots in the uniform spline model are spaced at regular intervals, so that the change of frequency is resolved at the same level of detail over the whole length of the sound. The actual number n of coefficients and the spline order k should be fixed in advance for all sounds, because a direct mapping is needed between any two sounds: if the temporal resolutions of two sounds do not match, they cannot be processed as coordinates within the same parameter space. While fewer coefficients lead to smaller storage sizes and performance gains, more coefficients improve the accuracy of the model.

6.3.1 Conversion from Paths to Spline Coefficients

Purely automatic tracking methods are likely to produce errors for certain sounds. Although the heuristic of following strong peaks of energy over a sequence of frames can yield very good results (see Subsection 3.4.2), it is problematic to rely on it when the correct identification of a fundamental is crucial. In the presence of strong chirps or noise, the automatic estimation of the fundamental can deviate from the actual value, causing higher partials to be seriously misaligned. In the case of octave confusion, the error is much bigger, since the measured fundamental will be half or double the correct frequency. Even few outliers in an otherwise correct sequence of frames will cause the curve fit to be off the correct fundamental trajectory. Until improved automatic tracking methods become available, manual tracking tends to give more robust results.

Fig. 6.4 shows the interface of the implemented sound object editor software. On top of a spectrogram, a path can be drawn using a set of drawing tools familiar from graphics applications. The resulting sequence of partials is always displayed on top of the fundamental trajectory, so that the alignment of the annotation and the actual spectrogram can be easily assessed. The interface also allows the user to move the control points on one of the higher partials directly, which gives better control in many cases. Another method that was not investigated in this work is semi-automatic tracking, in which a human operator draws a rough trajectory, which is then refined by an automatic fitting method.



1024
+3
1
512
random phase
jitter 0
shimmer 0
□ use peak clarity
☑ use harmonics
243 Hz

Figure 6.4: Screenshot of the implemented tracking and conversion software. Control points are inserted and manipulated using a drawing tool. The resulting series of harmonic partials is superimposed over the actual spectrogram in the background as a visual reference.

6.4 The Spectral Envelope

In the proposed PSOS model, two spectral envelopes are stored for each sound. The *harmonic spectral envelope* describes the varying loudness of the sinusoid partials in time and frequency. The *noise spectral envelope* describes the filter characteristics of a noise source and is used to encode the residual of otherwise harmonic sources. Both envelopes use the same envelope model, which stores data in a two-dimensional matrix of coefficients.

6.4.1 Coefficients of the Time-Frequency Envelope Model

The harmonic content of a sound is synthesized by convolving a source signal with a time-varying filter envelope. The envelope is a two-dimensional plane across time and frequency. Its height at any coordinate is the amplitude with which the source signal is multiplied. The source signal is generated by a large set of sinusoidal oscillators, the frequencies of which are controlled by the time-varying fundamental spline. The amplitudes of the partials are obtained from the envelope directly during the synthesis of the sinusoid signal.

Similar to the encoding of the fundamental trajectory, B-spline models are used to model the filter envelope as well. Two different sets of basis functions are necessary: one for the temporal offset axis and one for the frequency axis. Let the temporal basis functions be called $\mathbf{bT}_j(u)$, where u is a time offset and j ,$0 \leq j < n$ is the index of the current basis function, n denoting the number of basis functions in the temporal domain. Correspondingly, let the frequency basis functions be called $\mathbf{bF}_k(f)$, where f is a frequency and k ,$0 \leq k < m$ is the index of the current basis function, m denoting the number of basis functions in the frequency domain. The resolution of the envelope plane in time and frequency is determined by the sequences of breakpoints along both axes. In the temporal direction, breakpoints are spaced uniformly, just as in the fundamental frequency spline. Across the frequency axis, they are spaced

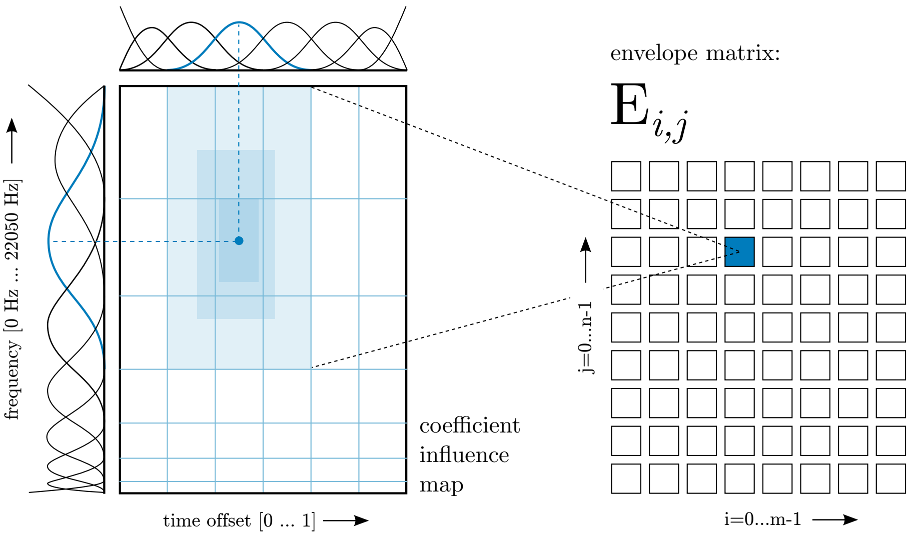


Figure 6.5: Correspondence between the matrix E and the spectral envelope model, with spline order k=3, uniform breakpoint spacing along the time axis and adaptive breakpoints along the frequency axis (blue grid lines). The matrix has m columns and n rows.

non-uniformly, to account for the different sensitivity of human hearing in different frequency bands. The breakpoints along the two axes define a grid that determines the resolution of the filter envelope.

Fig. 6.5 illustrates the correspondence of the coefficient matrix $\mathbf{E}_{j,k}$ and the spline-based envelope model. To evaluate the envelope $\mathbf{e}(u, f)$ at time offset u and frequency f, spline functions are first evaluated in time, then in frequency:

$$\mathbf{e}(u, f) = \sum_{k=0}^{n-1} \left(\sum_{j=0}^{m-1} \mathbf{E}_{j,k}\, \mathbf{bT}_j(u) \right) \mathbf{bF}_k(f) \tag{6.4}$$

It is important to note that the coefficients do not exactly correspond to intersections in the breakpoint grid, except for the case of linear spline models. Instead, the local influence of a coefficient is spread across k breakpoint sections, where k is the order of the spline. The local maximum of the basis function is not normally located at a breakpoint, and there are more basis functions than breakpoints for $k > 2$. Nevertheless, the placement of breakpoints is directly linked to the local resolution of a spline: adding breakpoints to a particular range of the spline means that each coefficient controls a smaller region, and thus that region can be captured more exactly.

The same envelope model is used twice in the parametric sound object synthesis model, for the harmonic content and for the residual noise, respectively. Different methods have to be used in both cases to extract data points from the input signal, which are then taken as inputs to the spline fitting algorithm.

6.4.2 Computation of the Harmonic Envelope

The parameters of the harmonic envelope are calculated from the peak amplitudes of the harmonic content. FFT analysis could be used to measure the amplitude of peaks,

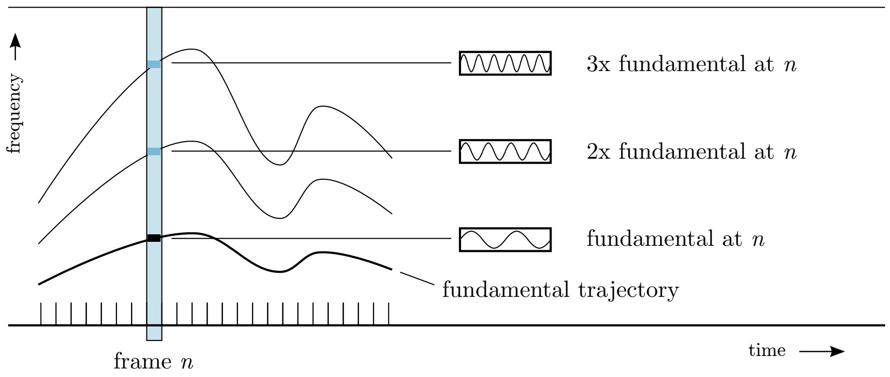


Figure 6.6: Computation of the partial peak amplitudes: at each frame n, the known fundamental frequency is used to convolve the source signal with windowed complex sinusoids with the exact frequency of the partials.

however, since the energy of peaks in an FFT transform is usually spread out over several FFT bins, estimating the precise amplitude of a particular frequency is difficult. To avoid such problems, a more exact method of calculating the partial amplitudes is used here, utilizing the previously determined fundamental frequency in each frame.

The amplitude of each partial is measured by convolving a complex sinusoid of the known partial frequency with the windowed signal. A Hann window is used for the convolution, with a window size between 256 samples and 2048 samples. For a sampling rate of 44 100 Hz (CD quality), a window size of 512 samples is a good choice in most cases, except for very low-pitched signals. The appropriate choice of the window size should be made with respect to the nature of the signal. Assuming that the frequency of a partial has been annotated or otherwise measured exactly, its amplitude and phase can be determined with high accuracy as the magnitude of the complex correlation (see Fig. 6.6). The effect of windowing has to be compensated for. In the case of Hann windows, the compensation factor is exactly 2.0 [4]. The procedure is repeated for all partials up to a specified cutoff frequency, in this case 22 050 Hz, or until a maximum of 100 partials is reached. This number is only relevant for fundamentals below 220 Hz, because all higher fundamentals are limited by the cutoff frequency, rather than the partial count limit.

The precise peak amplitude information is used at the same time to subtract the harmonic signal from the input signal to obtain the residual, which is analyzed later to create a noise envelope component. To achieve this, a copy is first created from the input signal. In each processing window, sinusoids with the measured partial amplitudes and phases are subtracted from this buffer. Since the windows are very short, stationary sinusoids can be assumed. A triangular window function is used in the subtraction to interpolate between overlapping windows. The resulting residual is very precise, because it is based on the removal of exactly specified sinusoids, rather than wiping out whole FFT bins.

The next processing step is the calculation of the magnitude envelope model from the list of peak amplitudes. This can be done by applying the spline-fitting procedure

[4] The Hann window uses the area below a cosine curve for multiplication. Since the area below the curve is the same as the area above the curve, the window reduces the average amplitude by a factor of 0.5.

to the peak data. However, this direct conversion contains a number of pitfalls. For high-pitched sounds, the number of peaks in the spectrum is much too low, and no information is available for the large frequency ranges between them: although no energy is observed between harmonic peaks, this does not mean that the harmonic envelope model should assume zero-energy amplitudes, since this assumption would almost certainly be wrong with respect to natural filter structures.

Even if it was decided that the values between peaks are unimportant, the fitting algorithm would still need enough peak information for each basis spline to obtain a determined solution: in order to fit a model to a list of data points, the number of points has to be bigger than the number of model parameters. With respect to a spline model of order k and polynomial order $k-1$, the minimal number of data points (or peaks) required to obtain a definite solution is equal to the number n of basis functions: the first breakpoint section contains a polynomial curve of order $k-1$, and therefore needs k data points to be determined exactly. For the next section of the spline, one coefficient drops out of the equation, because its basis function is constant zero, so one additional data point is required, and so on[5].

However, with just n peaks, n being the number of basis functions, the optimal polynomial fit would likely be extremely irregular. It would pass exactly through the points, but may take arbitrary extreme values between points. It is therefore wise to insert at least k points into every breakpoint section, so that the polynomial in that range is well behaved. To create additional "virtual" peaks in the empty space between peaks, the question has to be asked what value the envelope *would have* had if there *had been* a peak[6]. This hypothetical question can be answered most reasonably by interpolation.

Fig. 6.7 illustrates the algorithm of interpolating virtual peaks from the list of originally observed peaks. Since a piecewise linear interpolation would create edges in the spectrum, a cosine interpolation is used. k virtual peaks are created for every breakpoint section of the frequency spline, resulting in more peaks to be inserted in the low-frequency regions. This procedure ensures that a valid envelope can always be computed, regardless of the frequency of the original sound. Now that a sufficient number of virtual peak amplitudes is available for every breakpoint section along the frequency axis of the model, the B-spline fitting procedure is executed to obtain the control point coefficients.

The procedure of computing virtual peaks and calculating spline envelopes is repeated for each frame of the sound, leading to an intermediate representation of the time-frequency envelope as a sequence of frame envelopes. Each frequency spline uses the same n coefficients, and the change of each coefficient can be tracked over time. Once more, a spline model is used to encode the evolution of each frequency envelope coefficient over time, thus making it independent from the length and temporal resolution of the input sound. This leads to the two-dimensional envelope model described in Eq. 6.4.

[5] The `gsl_multifit_linear` function indeed requires at least n data points for a model with n basis functions, but technically allows for all points to be placed in one single breakpoint section, which leaves large parts of the spline under-determined.

[6] The term "envelope" describes exactly this: it refers to an outer hull of the spectrum that is touched by the loudest peaks.

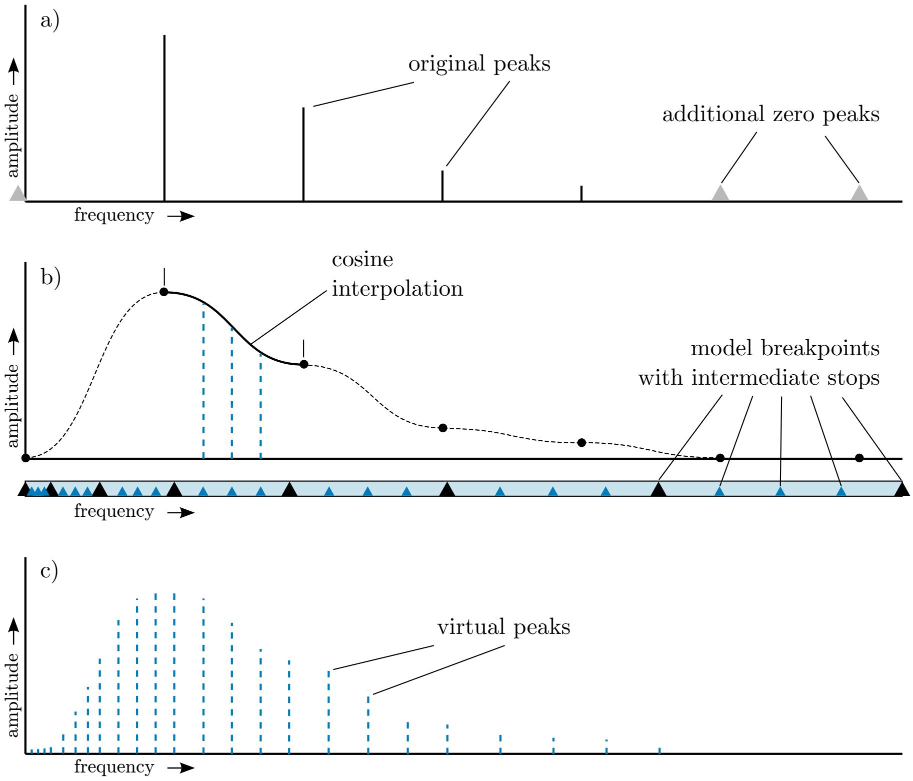


Figure 6.7: Conversion of observed harmonic peaks into more densely spaced virtual peaks. (a) The original list of peaks is completed by inserting necessary zero peaks up to the cutoff frequency. (b) Four virtual peaks are inserted into every breakpoint section of the frequency spline model. (c) The resulting virtual peak structure.

6.4.3 Computation of the Noise Envelope

In the standard spectral model (see Section 3.4), noise components are added to the harmonic content of a signal. Noise, according to that concept, is anything left after the harmonic partials have been subtracted from the input signal. Therefore, the noise component is often called the *residual*. However, noise may be a quite dominant part of the signal, and some sounds do not contain any harmonic components at all. Noisy, stochastic components are needed to model a wide variety of non-harmonic details, such as breathy sounds in speech, transients in instruments or noisy phenomena in natural environments. The proposed parametric model uses the same time-varying spectral envelope model to approximate the noise content as it uses for the harmonics. The main difference is in the source signal, which consists of a white noise source, instead of sinusoidal oscillators.

The resolution of the noise spectral envelope in time and frequency is set to be the same as for the harmonic spectral envelope. Although this is not strictly required, it is more practical for the implementation and helps to keep the harmonic signal and the noise residual well aligned: if a sudden change in the harmonic signal is not matched by an equally sudden change in the residual, there is some danger that the two components do not fuse well and are perceived as originating from separate sources. This problem has also been mentioned by Serra (1989, p. 96) with respect to applications of spectral modeling:

> “The characterization of a single sound by two different [representations] may cause problems. When different transformations are applied to each representation it is easy to create a sound in which the two components, deterministic and stochastic, do not fuse into a single entity. This may be desirable for some musical applications, but in general it is avoided, and requires some practical experimentation with the actual representations.”

The residual signal, obtained by subtracting the harmonic components, is processed frame-wise, just like the harmonic signal. At each frame, an FFT is performed, using a window size of 512 samples (approx. 12 ms). This is just enough to measure frequencies in the lower Bark bands. Larger window sizes do not add any useful detail to the noise, and the spline-based approximation would not encode it anyway. The magnitude of the FFT output is converted into a peak data structure and is subject to the envelope conversion method described in the previous subsection.

While the harmonic peak signal is very sparse and requires the concept of virtual peaks, the residual envelope can be calculated directly from the bins of an FFT. In contrast to the harmonic envelope, which encodes magnitudes, the noise envelope encodes power, i.e., the squared magnitude of bins. The magnitudes themselves vary strongly in a noise signal, and their height depends on the specific statistical properties of the source noise, as well as the shape of any window function that is applied before the transform. Averaging magnitudes over a range of bins, as it is done during the calculation of the spline coefficients, would lead to a signal that has the power associated with the average magnitude, which is an under-estimation of the actual average power[7]. Since the energy of a noise band is more relevant for perception than the amplitude of spectral bins, the noise envelope is based entirely on the power spectrum. The DFT of a signal $x[n]$ with N samples is defined as

$$\mathrm{X}[k] = \sum_{n=0}^{N-1} x(n)\mathrm{e}^{-j2\pi nk/n}\ , k = 0, 1, ...N-1\ , \tag{6.5}$$

where k is a frequency bin and X is a complex-valued array (Zölzer et al., 2002). Square brackets are used here to indicate that n and k only take integer values. The real and imaginary components of $X[k]$ are referred to as $Re[k]$ and $Im[k]$. The real-valued magnitude spectrum $MagX[k]$ gives the amplitude of sinusoid components, regardless of their phase:

$$MagX[k] = \sqrt{Im[k]^2 + Re[k]^2} \tag{6.6}$$

The power of a component can be computed from the square of its magnitude. Since the spectrum of the Fourier transform has two mirrored sides, the energy has to be multiplied by the factor 2, with the exception of the bins $k = 0$ and $k = N/2$:

$$P[k] = \begin{cases} 2N \cdot MagX[k]^2, \text{if } 0 < k < N/2 \\ N \cdot MagX[k]^2, else \end{cases} \tag{6.7}$$

[7] The sum of squares is not the same as the square of sums. Therefore, to preserve the correct energy, the *root-mean-square* (RMS) of amplitudes is sometimes computed, which is the square root of the mean of the sum of squared values.

Since the window size of the noise frames is $N = 512$, 256 power peaks are obtained from the FFT. There power peaks, and the frequencies associated with them, are used as the input of the curve-fitting algorithm[8].

6.5 Re-Synthesis

For the re-synthesis of sounds, no samples from the original sound sources are used, as the Parametric Sound Object Synthesis creates sounds entirely from the stored parameters. As such, the model is independent of the sampling rate of the source and of the original sound's length. During synthesis, a new sampling rate can be specified for the output. A sampling rate of 44 100 Hz (CD quality) is assumed as the standard synthesis sampling rate throughout this chapter. The parameters for the synthesis can either be obtained directly from an input sound, or may represent morphed or otherwise parametrically altered versions of input sounds.

6.5.1 Re-Synthesis of the Harmonic Component

During synthesis, the spline model provides values of a time-varying fundamental at an arbitrarily detailed resolution, making it possible to model gradual frequency changes per audio sample, not just per analysis frame. In fact, the concept of frames can be discarded entirely for the storage and synthesis of the parametric sounds. Since all values can be obtained from the spline model at any desired resolution, the synthesized sounds may be of a different — and possibly higher — sampling rate than the original input.

Since the spline model is based on a set of polynomial functions, the instantaneous frequency and phase values of the fundamental can be determined analytically and are independent from the sampling rate. The time offset $u \in [0, 1]$ is the local time within the sound object. Given the sound's global start time t_S and end time t_E, the offset u at global time t is calculated as:

$$u(t) = \frac{t - t_\mathrm{S}}{t_\mathrm{E} - t_\mathrm{S}} \tag{6.8}$$

Let $\mathbf{Q}_i\ , 0 \leq i < n$ be a set of frequency coefficients, and $\mathbf{B}_i(u)\ , 0 \leq i < n$ be the corresponding set of basis functions for the relative time offset u. According to the standard B-spline formula, the fundamental frequency function $\mathbf{F}(u)$ is then given as:

$$\mathbf{F}(u) = \sum_{i=0}^{n-1} \mathbf{Q}_i \mathbf{B}_i(u)\ , u \in [0, 1] \tag{6.9}$$

The phase increment $\Delta\varphi$ of the fundamental frequency oscillator, relative to the start time t_S of the sound, can be calculated from the integral of the frequency spline function at offset u:

$$\Delta\varphi(u) = 2\pi \int_0^u \mathbf{F}(u_1)\, \mathrm{d}u_1 \tag{6.10}$$

[8] In the algorithm suggested by Serra (1989), a magnitude envelope is computed by connecting the loudest peaks. This is a quite rough approximation of the actual amplitudes, as it ignores the valleys between peaks. During synthesis, Serra obtains the amplitudes of sinusoids directly from the smooth envelope and randomizes their phases. This seems to work good enough, although, in reality, the magnitudes would not be smooth.

The harmonic signal $\mathbf{S}_\mathrm{H}(u)$ is the sum of P oscillators, P being the maximum number of partials used for the synthesis. The instantaneous phases of the oscillators are linked to the time-varying phase of the fundamental, i,e., the first partial ($p = 1$). The partial frequency of the oscillators is an integer multiple of the fundamental, and is simply computed as $p \cdot \mathbf{F}(u)$. It follows that the phase increment for each oscillator is the phase increment $\Delta\varphi$ of the fundamental, multiplied by p. The harmonic signal $\mathbf{S}_\mathrm{H}(u)$ is obtained by summing up the oscillator values for all partials:

$$\mathbf{S}_\mathrm{H}(u) = \sum_{p=1}^{P} p \cdot \sin(\Delta\varphi(u) + \varphi_{0,p}) \ , \tag{6.11}$$

where $\varphi_{0,p}$ is the phase offset of each partial at the beginning of the sound. The most accurate — though computationally expensive — synthesis of the additive harmonic model is a sample-by-sample synthesis with individual oscillators for each partial. Before the synthesis begins, the start phases of the oscillators are initialized using either random phases or zeroes and stored in a `phases[]` array. The instantaneous fundamental phase is calculated incrementally, using a phase unwrapping technique. The algorithm uses an outer loop for the samples and an inner loop for the partials. The function `getFundamental(double offset)` returns the local value of the fundamental by evaluating the B-spline function. Likewise, `getEnvelopeAt(double offset, double frequency)` gives the local amplitude of the harmonic spectral envelope. The combined output is written into a `synthBuffer[]` array. The algorithm proceeds as follows:

```
double amplitude;              // local amplitude
double phase;                  // local phase
double offset;                 // relative time offset [0 ... 1]
double frequency;              // instant. fundamental freq.
double pFrequency;             // partial freq.

double fundamentalPhase = 0;   // unwrapped fund. phase

for(int n = 0; n < length; n++) {
    offset = (double)n/length;
    frequency = getFundamentalAt(offset);

    for(int p = 0; p < maxpartials; p++) {
         pFrequency = (p+1)*frequency;
         phase = fundamentalPhase*(p+1)+phases[p];
         amplitude = getEnvelopeAt(offset, pFrequency);
         synthBuffer[n] +=  amplitude*sin(phase);
    }

    fundamentalPhase += frequency*2.0*M_PI/samplerate;
    if(fundamentalPhase > 2.0*M_PI) {
        fundamentalPhase = fmod(fundamentalPhase, 2.0*M_PI);
    }
}
```

The number of partials (`maxpartials`) should usually be big enough to create partials up to the cutoff frequency, which is the Nyquist frequency of the input signal, typically 22 050 Hz. For a low-frequency sound of 80 Hz, this would require approx. 275 oscillators. However, low-frequency sounds tend to have less energy in high-frequency regions, so `maxpartials` can be kept much smaller, if performance is an issue. Ignoring partials beyond the 100th partial should be acceptable for most applications[9]. The frequencies of high partials can easily exceed the cutoff frequency of the model, which is set to 22 050 Hz (the Nyquist frequency of CD-quality audio). This problem is handled in the `getEnvelopeAt(double offset, double frequency)` function, which automatically returns zero for frequency values above the cutoff frequency. In high frequency regions approaching the cutoff frequency, natural envelopes usually slope off to near zero, so that no aliasing occurs when partials cross the cutoff frequency.

The performance of the synthesis procedure could possibly be increased by using a windowed overlap-add method and an inverted Fourier transform. However, a windowed method for harmonic partial synthesis is less accurate for strongly sloped fundamental frequencies, because the assumption of stationarity within the window does not hold in that case.

The evaluation of a `sin()` function for every sample is a costly operation, especially when it has to be executed for every partial in the synthesis. If stationary frequencies are assumed within short windows, a more efficient implementation can be realized in the form of a *quadrature oscillator*, where the instantaneous sine and cosine values are calculated recursively from the preceding values. This corresponds to a multiplication with a rotation matrix, and therefore does not require the repeated calculation of `sin()` or `cos()` functions (Turner, 2003).

6.5.2 Re-Synthesis of the Noise Component

The noise component, or residual, is synthesized separately from the harmonic signal components. The synthesis is based on short frames of filtered noise, which are combined using an overlap-add technique. It uses overlapping windows of 512 samples (approx. 12 ms), where the overlap is 50 % of the window size (256 samples). The number of frames N is calculated from the target length of the sound, which is identical to the length of the harmonic component.

A white noise buffer is created, long enough to contain the complete sound. The buffer is filled with random numbers between -1.0 and +1.0. It is important that overlapping windows process samples from this common buffer, rather than creating their own random noise independently. The mean of two independent random variables is biased towards the overall mean of the random distribution, i.e., towards zero. Therefore, if noise from two independent sources was added together in the overlapping portion, the amplitude of the noise would be decreased between window centers.

Two FFT transforms are initialized: a real-to-complex transform to convert the real-valued noise into the complex Fourier domain, and a complex-to-real backwards transform, which exactly reverses the first transform. The `fftw_plan_dft_r2c_1d(...)` and `fftw_plan_dft_c2r_1d(...)` functions of the GNU Scientific Library (GSL)[10] are used in this implementation.

The synthesis algorithm loops through all N frames of the sound and copies the corresponding 512 samples of the noise source into a local buffer. The forward FFT is then computed for the window. The filtering is performed in the FFT domain by multiplying each complex FFT bin with an amplitude factor, which is read from the

[9] This is more than enough to re-create a complete formant structure for a male speaker with a low voice frequency, e.g., 100 Hz.

[10] `http://www.gnu.org/software/gsl/manual` (last visited: December 1, 2010)

previously stored noise envelope, using the `getEnvelopeAt(double offset, double frequency)` function. The offset is the relative position of the currently processed frame, relative to the sound length. The frequency is the center frequency of the current bin, given as `index` · `samplingrate`/`fftsize`.

After the multiplication of the FFT values with the envelope function, the transform is reversed, so that a filtered noise buffer is obtained. The buffers of the individual frames are copied into an output buffer, using triangular overlapping window functions.

Finally, the results of the harmonic synthesis and the residual synthesis are added and copied into the final synthesis buffer. The resulting synthesis buffer may contain individual samples that exceed the value range of sampled sounds, and thus would cause clipping in the playback. This cannot easily be avoided during synthesis: although the value range can be bounded for each individual harmonic and noise component, the final value of a sample is subject to some randomness, and the added effect of all sinusoids and noise may lead to local values exceeding the value range in the positive or negative direction. However, for sounds that do not make use of the full value range, clipping is rarely encountered. In cases where clipping becomes a problem, the buffer can be normalized, or a non-linear compression function can be used to force values to stay within the intended range, at the cost of some distortion in loud portions.

6.6 Choice of the Model Resolution

The proposed parametric model can be configured to use different resolutions, both in time and frequency, by specifying the spline parameters and adapting the placement of frequency bands. A minimal configuration, using only linear spline segments, no temporal subdivision and no frequency subdivisions, has 11 parameters[11]. However, a more typical configuration with quadratic splines, 20 temporal subdivisions and frequency bands matching the Bark scale uses as many as 1156 coefficients, of which $2 \cdot 567 = 1134$ are used to encode the details of the two spectral envelopes.

The intention behind sound modeling is often compression. For the domain of transmitting or storing sound objects, models should have as few parameters as possible to be useful — especially compared to well-established compression algorithms like MP3 or low-bitrate speech codecs. Parametric models, as the one presented in this thesis, can be used for extreme compression and very-low-bitrate coding, provided that the process of encoding and decoding can be fully automated and efficiently implemented. A model of parametric sound objects, when used as a codec, would have a number of limitations, because it could not encode modulated, inharmonic or polyphonic sounds. Still, for specialized applications, the parametric model could be useful as a compression mechanism, and could achieve compression rates of 1:100 or more, when some loss of fidelity is acceptable.

As already described, for complex sounds, i.e., sounds with strongly varying frequency trajectories and many significant overtones, naturalistic modeling can require a high number of parameters: a model with 25 frequency breakpoints, 40 time breakpoints and quadratic polynomial degrees will already have more than one thousand coefficients. A fixed, general number of parameters cannot be given, because the perceptual fidelity depends upon the characteristics of the sound. If the model is configured

[11] 1 parameter for the length, 2 for the end points of the linear fundamental frequency spline, 4 for the edges of the spectral pane for the harmonic content and 4 for the edges of the spectral plane for the noise content.

to use a thousand coefficients or more, subtle details like the vibrato of a violin sound can be captured in great detail. In such extreme cases, the number of coefficients can be bigger than the number of samples used in the original representation of the sound, indicating that the model is not very useful for high-fidelity compression, unless further coding methods were used to reduce the bitrate[12]. And yet, even in cases where the model size is comparatively big, the parametric model is still useful, because it provides access to the relevant synthesis parameters of the sound and turns it into something flexible.

Using a large number of coefficients does not necessarily interfere with the goals of parametric modeling, still, keeping their number low is desirable. Not only does using thousands of coefficients slow down the processing, it also decreases the relative importance of each coefficient. A central principle of parametric modeling is that each parameter controls a significant, and possible intuitive aspect of the sound. This is not the case for a coefficient which controls a tiny detail in a local region of a spectral envelope. In addition to this, there is a point at which increasing the number of coefficients cannot improve the fidelity of modeling: no matter how much the detail of the time and frequency resolution is increased, the PSOS model can never accurately represent polyphonic sounds or certain types of noisy partials. Increasing the detail of the model also tends to cause problems in the analysis step: when the spacing between time frames becomes smaller than the resolution provided through the respective analysis window size, the resulting fit of the B-spline-plane can get very irregular. Some of these problems could possibly be solved, but that would require much more elaborate tracking methods than those examined in this thesis.

In general, it can be said that the resolution of the model should be as low as possible, and as high as necessary. The same basic problem can be observed in graphical vector drawings, where a similar trade-off has to be made between naturalism on the one hand, and storage requirements and practical drawing limitations on the other.

6.6.1 Uniform vs. Adaptive Temporal Resolution

Some sounds suffer from a lack of precision in their start transients when they are represented in the PSOS model. Examples for this are gunshot and plucked guitar strings. In the proposed analysis-synthesis setup, the temporal resolution of the model is spread evenly across the sound, and sharp transients tend to get smoothed out. When a high temporal resolution is used, the degradation is not very noticeable, but increasing the resolution of the whole model just to capture the start transient accurately is often impractical.

Fig. 6.8 shows four strategies of dealing with the transient problem when few breakpoints are used. 6.8 (a) shows the unchanged modeling approach, in which the transient is seriously degraded. In (b), the model itself is the same, but the quiet part before the transient is not included in the model, which allows the envelope breakpoint to be exactly aligned with the maximum transient peak. This is only an improvised solution, and although a sharp transient impression can be achieved with it, cutting off the first half of the transient is not always acceptable. In (c), a different modeling approach is shown with adaptive breakpoint placement. However, such a model introduces a major problem: the mapping between any two sounds would no longer be

[12] For example, the required storage space of the model coefficients could be reduced by applying different degrees of quantization to different coefficients, and to use redundancy coding.

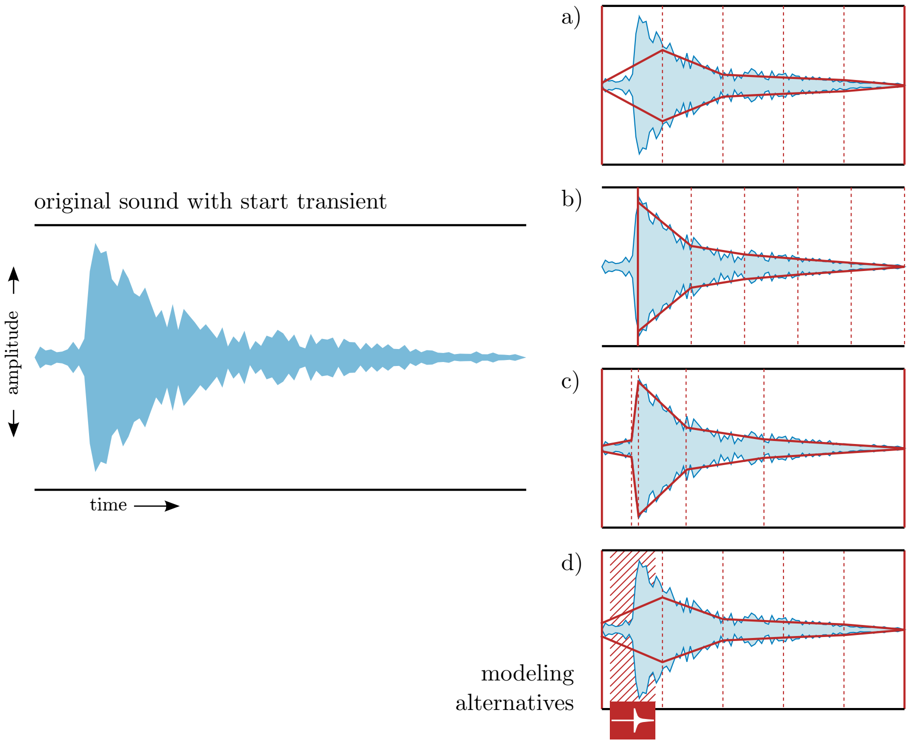


Figure 6.8: Different methods for improved transient modeling: (a) standard distribution of breakpoints without improvement. (b) Alignment of the first breakpoint with the transient peak. (c) Adaptive placement of breakpoints. (d) Standard breakpoints with added transient information.

valid. Fig. 6.8 (d) shows a possible solution to these problems: the standard uniform model could be enriched with a special set of transient parameters. Many types of sounds could profit from this extension, which could also reduce the required resolution in the non-transient parts of the sound. Methods for detecting and encoding transient information have already been discussed in Subsection 3.4.5. They could be adapted to the proposed parametric model.

6.6.2 Automatic and Manual Conversion Methods

To transform recorded sounds into the model parameter space, different strategies can be used, ranging from fully automatic methods to methods requiring a high degree of manual interaction. The proposed model is based on the concept of harmonic partials evolving over time. For a fully automatic conversion approach, tracking (see Subsection 3.3.1) is therefore a major aspect of the conversion process. Before the actual conversion starts, a decision has to be made — either automatically of through user interaction — what portion of audio should be treated as a coherent sound, and whether the sound has harmonic content at all. Provided that a portion of a recording has been marked as containing a harmonic sound, the tracking proceeds by estimating the most likely fundamental frequency in each frame of the sound, which, should produce a slowly-varying trajectory for the fundamental frequency.

There are several ways in which this simple tracking mechanism can go wrong. For portions of audio in which no clear harmonic structure exists, and for brief pauses within a sound, a strategy has to be implemented for interpolating or extrapolating the fundamental. Such inharmonic portions often occur at the start of a sound, e.g., during the attack sound of a piano tone. Even the decision whether harmonic content is present or not can be difficult. The typical problems that can occur in automatic methods for harmonic tracking have been discussed in Subsection 3.4.2. Therefore, although fully automatic processing is clearly desirable for future implementations, manual interaction is still required in the current implementation to achieve optimal results.

6.7 Possible Extensions and Additional Parameters

Some characteristics of a sound can be altered after the sound has been produced at the source in the form of post-processing. For example, digital filters applied to a recording are applied *after* the sound has been produced. Such post-processing effects include a wide range of mixing and mastering techniques to add sounds, apply non-linear amplitude scaling ("compressor") to signals, or change the intensity of certain frequency bands ("equalizer") (Zölzer et al., 2002). Echos and reverberation effects are also typically separated from the sound production mechanism. Although the current implementation does not consider reverb or digital effects, this section will provide an overview of techniques that would have to be implemented in order to achieve such processing.

6.7.1 Processing Echo and Reverb

In acoustics, a difference is often made between clearly distinct sound reflections, called "echo", and very dense reflections, called "reverb" (see Subsection 2.2.7). Echo and Reverb can cause a number of problems: a naïve modeling algorithm, which is unaware of echo, will observe multiple time-shifted versions of the sound, overlapping in time, as if they were uncorrelated and had been produced by separate sound sources. This is not only physically incorrect, but also introduces serious problems at subsequent analysis and synthesis steps: reverb introduces frequency components into the mix that interfere with the tracking and analysis of the original components. Therefore, it would be desirable to separate the reverb and echo from the original signal content, and re-apply it to the output signal later, in case that should be required.

Methods for de-reverberation and deconvolution can be used to estimate and remove the effect of reverberation to some degree. Most of these methods have been developed in order to improve the quality of speech transmission systems. Yegnanarayana and Murthy (2002) have described a method for speech de-reverberation that takes differences in the *signal-to-reverberant component ratio (SRR)* into account. However, the authors point out that, using their method, removing reverb can only be achieved at the cost of some distortion. A method for blind echo cancellation in speech signals has been described by Torkkola (1997), in which parameters of a recursive filter are estimated. However, the author mentions that for long filters, the results of the estimation can get quite inaccurate and may introduce noise.

6.7.2 Processing Modulation, Vibrato, Distortion, Jitter and Shimmer

Besides echo and reverb, several other effect can be applied to sound, with different consequences for the analysis and synthesis. If the analysis is unaware of the presence of these effects, the result is often a false estimation of the true acoustic properties, or a representation of simple phenomena by overly complex combinations. Vercoe et al. (1998) have pointed out the advantages of separating effects from the sounds that are to be encoded.

When a carrier frequency f_c is amplitude-modulated by another low-frequency oscillator of frequency f_x, the result is a tremolo-like effect, in which the changes in amplitude are perceived as such. However, if the modulator frequency is a frequency in the audible range, the result of the modulation is a change in timbre. Three tones are perceived: the carrier frequency f_c, the difference frequency $f_c - f_x$ and the sum $f_c + f_x$ (Zölzer et al., 2002). Although the cause is just a simple multiplication, the effect would prevent the proposed model from being effective: since it is only aware of a single fundamental frequency, it would miss the frequency components of the modulation, causing a false estimation of the residual energy or other negative effects. A method for measuring modulation parameters has been described by Roebel (2006), which is able to approximate the parameters of a non-stationary sinusoid in the presence of strong slopes and reduce the estimation bias compared to other methods.

Vibrato is a much slower modulation of the fundamental frequency, often encountered in music. Rossignol, Depalle, Soumagne, Rodet, and Collette (1999) have proposed and compared several methods for detecting vibrato and its parameters in audio recordings, but have not described methods for removing the vibrato effect in detail.

When distortion is present in a signal, sinusoids can be clipped or otherwise warped in a non-linear way. The resulting sound is still periodic, but the waveform is changed, causing additional harmonics to be generated that were not present in the original signal (Zölzer et al., 2002). Although a distorted signal could still be processed with the harmonic model, distortion causes a simple signal to be represented by an overly complicated model, which makes it difficult to access the original sound parameters.

Two other effects, jitter and shimmer, are period-by-period variations of the frequency and amplitude, respectively (see Subsection 2.1.7). They are often caused by various irregularities and random influences in sound producing objects. Integrating them into a synthesis model can contribute to the realism of some sound types, including the synthesis of the human voice. The fluctuations introduced by jitter and shimmer are typically too rapid to be accurately tracked by the sinusoidal analysis. During the analysis of a sinusoid containing jitter, a regular frequency tracking technique would detect a wide-band signal instead of a rapidly changing narrow-band signal, again causing problems in the subsequent processing steps.

6.7.3 Processing Spatial Information an Binaural Clues

In its current form, the PSOS model only processes monaural recordings. Since many types of sound textures would profit from stereo or surround sound synthesis, parameters could be integrated into the model to encode the placement of sounds within the stereo field or even in virtual 3D space.

In the simplest case, stereo information could be added to the model in the form of a panning parameter, which controls the relative loudness of the left and right speakers. For surround speaker setups, the placement usually has to be encoded as a two-dimensional or three-dimensional position. From this placement information, the proper loudness and filtering coefficients for an array of speakers, or a set of headphones, can be calculated. To simulate the properties of directional hearing more accurately, the lag of arrival time of the sound between the left and right ear can be calculated, as well. For the sound texture model, the exact method used for the rendering of the sound is not important, and the model does not need to know how many speakers will be used to play the sound.

The problems of adding spatial information are once again on the analysis side: while it is trivial to encode the information that a sound is coming from a certain angle, it can be difficult to obtain this information from a recording automatically. The loudness difference between a left and right channel can be measured easily — provided that the level of background noise and interfering sounds is low — but when the tracks have been recorded by spatially separated microphones, the time lag between channels requires further methods of integrating the two components back into one sound.

Even if a proper analysis of spatial information is difficult, or if only monaural recordings are available, “fake” stereo effects can still be used to simulate a spatial distribution of sounds. For example, rain drops of a rain texture could be synthesized at random stereo panning offsets, with no guarantee that this is physically accurate.

6.7.4 Respecting Phase Alignment

While the exact alignment of the phases between different partials does not matter for many types of sounds, it has a noticeable effect on some others. Although no in-depth evaluation of this phenomenon has been attempted in this thesis, it appears that the ear is more sensitive to phase for speech sounds than for non-human sounds, such as instrument sounds. Experiments by Helmholtz (1913) had originally claimed that the ear was “phase-deaf”, but his experiments only involved mechanically-produced artificial sounds. More thorough experiments with better equipment reliably indicate that the phase alignment changes the perception of timbre, especially for pulsed sounds. However, little is known about how the human brain processes this information, or if it helps in the process of understanding speech (Andersen & Jensen, 2004).

For the synthesis of voice sounds, the random phase initialization can be replaced with a synchronized phase, thus producing a crude approximation of the glottis pulse. In some experiments related to the model proposed here, this improved the perceived sound quality and realism significantly — again, especially for low-frequency voices — but had almost no effect for other sounds. The advantage or disadvantage of such synchronization was not investigated in more detail, because it is apparently very speech-specific.

A concept for *relative phase delay* (or *normalized relative phase delay*) was introduced by Di Federico (1998). The model is built on the notion that the shape of the waveform is mostly constant over many periods of a sound, i.e., the phases of partials develop in sync with the phase of the fundamental. Instead of encoding the absolute phases of partials at a given time frame i, their relative delay with respect to the fundamental partial can be encoded as a small time offset. The phase delay is defined as the time offset between the next maximum of a partial sinusoid and the center of the analysis frame at which the phase was measured:

$$\tau_{i,k} = \frac{\theta_{i,k}}{\omega_{i,k}} \quad , \tag{6.12}$$

where k is the k-th sinusoid of the partial series, i is the frame number, θ is the instantaneous phase at the frame center and ω is the angular frequency of the partial. The relative phase delay $\Delta\tau_{i,k}$ can then be defined as the difference between a partial's phase delay and the phase delay of the fundamental frequency partial:

$$\Delta\tau_{i,k} = \tau_{i,k} - \tau_{i,1} \quad . \tag{6.13}$$

For any given partial, this formula may give a delay that is longer than one period size of the partial. Therefore, the normalized relative phase delay $\tilde{\Delta}\tau_{i,k}$ applies a modulo function to the relative phase delay, so that the delay for any partial lies within a range between 0 and 2π of the period (Di Federico, 1998).

6.8 Limitations of the Sound Element Model

Even with the integration of various additional parameters, conceptual problems remain that make it difficult to achieve high-quality coding results for some types of sounds. A parametric model with a high degree of abstraction, like the one presented in this thesis, does not behave like an all-purpose codec. Many advantages of the model are only possible at the cost of certain drawbacks. Some of the limitations introduced by the chosen modeling paradigm are described below.

6.8.1 Limitations in Accuracy and Detail

When a complex object is represented parametrically, some loss of detail often occurs: it is impossible to represent a high-dimensional signal in a much lower-dimensional parameter space, unless the original object contains inherent redundancy. However, most sounds contain a lot of perceptual redundancy, and so, although much detail is lost, the results can still be very acceptable.

The resolution of the model can be increased for sounds that require more detail, by increasing the resolution of the time-frequency grid. While this solves some problems of limited model accuracy, it comes at the cost of making the model overly specific, and may easily lead to model storage requirements that are much larger than the original sound (see Section 6.6). Similar problems are encountered in graphics, when an image has to be approximated by a vector drawing. Forcing the algorithm to convert every single pixel of the source image into a vector square is guaranteed to preserve all details — but it is also a mockery of the principles of vector drawings. Likewise, for most remaining problems of the proposed sound model, increasing the resolution is not a useful solution. When the resolution is increased up to a level where all rapid changes and quick modulations can be captured, the concept of parametric modeling degenerates to a complicated representation of a spectrogram and tends to be much less useful for any subsequent processing and machine learning tasks.

Even if the model resolution would be increased by arbitrary amounts, some limitations of the model would persist, as they are a direct consequence of the modeling paradigm. In particular, polyphonic sounds cannot be generated with the model, no matter how high the resolution is set.

Some degradation of accuracy is usually acceptable: inaccuracies in the spline-based approximation of the fundamental frequency or the filter envelope are often too small to be noticed, and if the fundamental trajectory of a synthetic sound is a little smoother than the behavior of the fundamental in the original sound, few people will actually notice this, especially if they do not know the original. The gained flexibility usually outweighs such problems. For some applications, an artificial quality of the synthesized sounds may even be a stylistic choice, as is often the case for vector drawings.

6.8.2 Sounds Outside the Scope

The PSOS model is based on assumptions about the physical nature of harmonic sounds. These assumptions — especially the assumption that harmonic partials are integer multiples of a fundamental — are general enough to be true for a wide range of sounds, but there are sounds that can not be accurately represented within the model. The choice of parameters in the PSOS model introduces some limitation concerning the range of sounds that can be modeled, like sounds with strong inharmonicity, modulations or non-harmonic partials.

As was explained earlier, inharmonic sounds, such as produced by bells, metal rods and related bodies, do not produce partials at integer multiples of a fundamental (see Subsection 2.1.3). A bell sound, when represented in the PSOS model, would be reduced to its most dominant harmonic group of partials, thus cutting away other sinusoids, and it would not sound much like a bell.

A similar problem occurs with sounds that contain subharmonics (see Subsection 2.1.5), since they have not only one fundamental, but produce additional frequencies lower than the fundamental frequency. These sinusoids, which would not be captured by the harmonic model, would be added to the residual energy, thus leading to disturbing noise content in lower frequency bands. Similar effects would be observed for sounds with strong modulation. This applies to many types of animal vocalizations and bird calls, especially trills.

Polyphonic sounds cannot be processed by the model either, as it is designed to describe exactly one acoustic phenomenon parametrically. This concerns musical chords and acoustic scenes with several sound sources, even in cases where they fuse into one perceptual stream for a human listener.

Pseudo-harmonic sounds would loose much of their perceived roughness if they were forced into the PSOS model. Noisy partials would be approximated by accurate partials and added noise. While this does not do justice to the actual phenomena in the source sound, the result may still be acceptable.

6.8.3 Problems with Discontinuities in the Sound

The proposed model assumes that the processed sounds are smooth in nature. They can have time-varying pitches, but clearly localized changes in the signal are not considered. When such significant points do occur in certain signals, this can have negative effects on the quality of the encoding, and even worse effects on any subsequent mixing or morphing operations.

Consider two signals that each have a sharp step increase in frequency, but at different time offsets. The uniform B-spline model will only shift the coefficients at

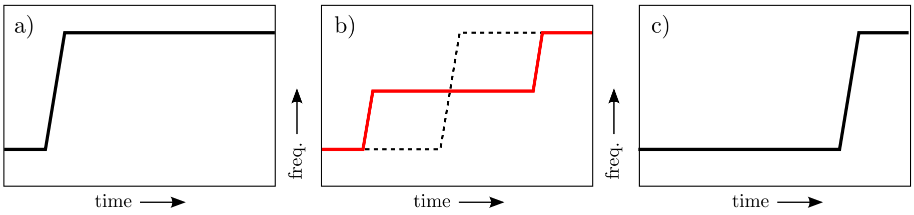


Figure 6.9: Example of two signals (a) and (c) lacking good correspondence. Instead of shifting the location of the step, the intermediate signal (b) has aspects of both steps (red line). The dotted line marks the intuitively expected result.

breakpoints up or down, but cannot shift the breakpoints in time. This restriction is imposed on the model in order to guarantee deterministic results and secure that each coefficient in one parametric sound will always have a matching counterpart in another sound — at the cost that the model cannot adapt its parameter set individually to different sounds. Therefore, a blend between two sounds that have a significant step in different time offsets will not shift the time offset, but create a superposition of two weak steps (see Fig. 6.9).

Problems like this are sometimes solved by enforcing correspondence, i.e., by marking the significant points and warping the timelines of the two sounds to align them. An example for such a warping of the timeline can be found in the `dilate` function of the Loris[13] software (Fitz, Haken, Lefvert, & O'Donnell, 2002). Ezzat et al. (2005) have used a concept called *audio flow* to define a mapping between two sounds, however, this was done between spectral envelopes, not frequency trajectories.

[13] `http://www.cerlsoundgroup.org/Loris` (last visited: December 1, 2010)

Chapter 7

Evaluation

The quality of sound objects, and their fitness for a use in realistic sound textures, was evaluated in an online listening test, using a large number of human listeners to judge the quality of individual sounds. In this chapter, some general problems in the design of listening tests and inter-subjective evaluations are discussed, including methods to avoid bias. The test setup of the online evaluation, which follows the so-called MUSHRA concept, is then explained. Results from the evaluation are presented in detail, and strengths and weaknesses of the proposed model are discussed with respect to the ratings obtained for different types of sounds.

7.1 Inter-Subjective Evaluation Methods

As already mentioned in Subsection 4.1.1, there is no useful technical measure through which the quality of a synthetic sound could be assessed. The modeled sound necessarily leaves out details of the original sound. Therefore, the important question is not whether anything is missing, but what effect this degradation has on a listener. The usual way to test these effects is to present examples of processed sounds to many test listeners and let them rate the quality, the similarity or any other relevant aspect. Although each of these judgments represent only one subjective preference, a stable inter-subjective judgment can be formed if the number of participants is large enough.

Two criteria that any useful test has to satisfy are *validity* and *reliability.* Validity refers to the question if "an instrument measures what it is intended to measure" (Svensson, 2000), i.e., if the method of the test is actually suited to answer a particular question. For example, in an evaluation to determine the most musically talented pianist in the world, a method that would simply compare the figures of CD sales would — or should — be criticized as being invalid[1]. In cases where a true answer to the evaluation question is known to exist, a valid test will produce results that strongly agree with the truth.

The other important aspect, reliability, is concerned with the question how exact the results are, and in how far they can be reproduced. A test is only reliable if it produces the same outcome within acceptable tolerance ranges, regardless of small

[1] Authors of an algorithm often ask subjects for specific aspects of the result, typically those aspects that were in the focus of the design of the algorithm. Sometimes general conclusions about quality are drawn, or implied, even though they were never evaluated, thus reducing validity. Lu et al. (2004) have asked participants to rate the "smoothness" and "variety" of synthesized sound textures, which, exactly speaking, is not the same as an assessment of overall quality.

measurement errors or differences between subjects or test conditions. In general, the reliability of an evaluation increases with the number of measurements taken: while individual measurements are subject to randomness, their mean (or median) value is likely to converge to a stable value.

As in any experiment about perception, an evaluation method for sound elements has to be chosen carefully in order to avoid *bias*. Bias occurs when, through some aspect of the test design, the participants' judgment gets influenced to produce a tendency in one direction. Possible sources of bias are non-neutral questions on the experimenter's side, an interest of the participants to produce a certain favorable outcome, familiarity with some of the evaluated samples or the ordering of evaluated samples. To avoid any influence of the presentation order, e. g., a tendency to become more strict or more lenient as the test session progresses, many test setups use randomized orders that are changed for each group of participants, or even for each individual participant (Day & Altman, 2000).

Before trusting that the combined judgment of many test subjects will lead to a useful estimate, the question should be asked if individual people are able to discriminate at all between the samples presented to them. If so, they should consistently give very similar ratings in multiple test runs. If, however, a subject's ratings are very inconsistent, the reliability decreases. In market research, *consistent preference discrimination testing* is used to test a subject's ability to discriminate between different samples. This can be done by presenting the same pairs of samples multiple times, anonymized and in randomized order, possibly without telling the subject. To increase the reliability of the test, an expert group can be selected out of all test subjects, containing only those individuals with the best discrimination ability (Buchanan, Givon, & Goldman, 1987).

7.1.1 Open versus Blind Experiments

If a preference has to be expressed for one of two alternatives, it is often desirable to conceal the identity of the samples for the test participants, because knowing which sample is which may influence their judgment. Making the participants *blind* of any labels is usually the preferred method of testing, because it excludes many sources of bias. However, blind methods are of little value when the identity of the test samples is immediately obvious to the test participants. For example, in audio tests, comparing high-quality uncompressed sounds to strongly degraded sounds may counteract the mechanisms of the blind test setup to some degree.

In a *double-blind* experiment, the identity of samples is not even known to the experimenters who interact with participants. If the experimenters know which sample is currently shown, there is some danger that they will — consciously or unconsciously — change their style of presenting, their wording or body language, in favor of the desirable outcome (Day & Altman, 2000). Double-blind test setups are easily implemented in the case of fully automated, computer-based tests, where the direct interaction between the participants and the experimenter during the test is usually not necessary. Blinding can also be used during the evaluation of results, especially if subjective judgment is involved.

Blind and double-blind experiments are traditionally of great importance for the development and testing of new medical drugs, where the effectiveness of a substance has to be proven and compared to the placebo effect. In that case, each participant usually receives only one item — i.e., a drug or placebo — and the allocation of participants to items is randomized (Day & Altman, 2000).

7.1.2 Interviews and Expert Judgment

A category of evaluation methods that is sometimes overlooked is the category of qualitative methods, which offers the possibility to record participants' reactions and judgments without the use of scoring systems and scales. Qualitative methods can involve interviews and questionnaires, but may also include diaries or video recordings of participants' behavior (Shaw, 1999). In an interview situation, there is some danger that the interviewer will influence the participant by his or her selection of questions, body language or remarks made during the interview, thus tainting the outcome of the evaluation. Nevertheless, interviews can play an important part early in the design process, as an inspiration to direct the research, rather than a validation of results (Bogner, 2005, pp. 7ff.).

The audio examples of parametrized sound objects explored in this thesis, however, can be quite easily evaluated by quantitative methods; as long as the question is one about preference, similarity or realism, the tests can be designed with scoring or ranking mechanisms. Again, the results of the quantitative evaluation may lead to more complex, qualitative questions, like "how should the algorithm be changed?" or "what properties of the sounds are most disturbing?".

7.1.3 Scalar Values Versus Binary Choices

There are different methods available for letting test participants assign a value to an item under examination. The granularity of scales differs from simple binary choices to continuous value ranges. There are also different methods of labeling the scale.

A type of scale that is sometimes used is the *visual analogue scale (VAS)*, which is a line that allows the participant to set a mark at an arbitrary position with a pen (Svensson, 2000). Only the far ends of the scale carry labels, such as "very low quality" and "perfect quality". During the evaluation, the positions of marks are measured in terms of millimeters. A similar form of scale is the *graphic rating scale (GRS)*, which is also continuous, but has additional labels placed along the line, so that each label covers an interval of the same size. The two extreme ends of the scale do not normally carry labels (Svensson, 2000).

Instead of continuous scales, *verbal descriptor scales (VDS)* can be used, in which the scale is converted into a set of choices that can be checked by the participant. The number of discrete choices given is sometimes expressed as VDS-n, where VDS-5 describes an evaluation with five different choices offered. According to Svensson (2000), the use of discrete choices can lead to more inter-subject agreement and emphintra-scale stability, without sacrificing important detail of the scale. On the other hand, enforcing strong agreement is not always necessary in an evaluation, especially if the experiment is meant to reveal trends and differences, rather than absolute ratings.

When subjects are asked to rate items on a scale of numbers, they will apply different criteria and will likely use different concepts of distance or absolute value. If a test participant rates one item with the value "4" and another with the value "5", that implies that one item is better than the other, however, there is no meaningful way to calculate *how much* better. Such problems in the evaluation design cannot be completely avoided for questions dealing with subjective ratings. As stated by Svensson

(2001), it is important — especially if the number of discrete choices is small — to treat the numbers strictly as ordinal numbers, not as marks with a known quantitive value. It follows that adding or subtracting of such ordinal numbers does not yield any meaningful result, and mean scores or standard deviations are not meaningful either. However, the use of median values, minimum and maximum ratings is always appropriate, because they do not interpret the data in any way (Svensson, 2001).

The theory of interval scales and ordinal scales has been described by Stevens (1946, p. 679), along with a list of allowed operations for each case. Stevens advises to treat mean scores in ordinal scales with care, but admits that, although incorrect in a strictly mathematical sense, they may still serve a purpose:

> "In the strictest propriety the ordinary statistics involving means and standard deviations ought not to be used with these scales, for these statistics imply a knowledge of something more than the relative rank-order of data. On the other hand, for this 'illegal' statisticizing there can be invoked a kind of pragmatic sanction: In numerous instances it leads to fruitful results. While the outlawing of this procedure would probably serve no good purpose, it is proper to point out that means and standard deviations computed on an ordinal scale are in error to the extent that the successive intervals on the scale are unequal in size. When only the rank-order of data is known, we should proceed cautiously with our statistics, and especially with the conclusions we draw from them."

It is useful, perhaps even necessary, to provide examples of samples on both ends of the scale, especially for domains in which the phenomenon under evaluation is not intuitive for the test participants. But while examples for perfection are typically easy to find, it is much less obvious what the worst possible item should be. Scales of quality tend to fixed at the upper end, but loose at the lower end. To construct a frame of reference, so-called *anchors* can be introduced into the evaluation, samples that have some known, bad quality. Even if the anchors can never represent the "worst possible" item, they can still increase the amount of inter-subject agreement on the lower end of the scale.

In evaluations where it is important to utilize the full range of the scale, the ratings of individual participants can be scaled or normalized. Scaling can help to emphasize differences between items.

7.2 Evaluation Methods of Audio Coding Quality

The insights provided above can be used to design adequate evaluation methods for the audio domain. Since the conversion of sound recordings to parametric sounds is performed with the intention to preserve the important aspects of the input sound, it becomes clear that the evaluation has to involve a comparison of the two, in either direct or indirect form. Such forms of evaluations are sometimes used to compare codecs for music or speech compression, and although the proposed model is not actually a codec — as it is lacking an automatic encoding component — similar principles of similarity and perceived quality apply.

7.2.1 Quality, Similarity or Realism?

There are many different aspects of sound that may be assessed in an evaluation, therefore, finding the right question is not a trivial task: two questions that appear to ask for the same thing may actually be asking for quite orthogonal properties. Even the assumption that the most exact encoding of an original is also the most favored encoding, may very well be wrong. For example, it has been observed that young music listeners have grown so accustomed to the artifacts in MP3 files that they increasingly prefer music with such artifacts (M. Ahmed & Burgess, 2009).

A form of question that is easy to answer and does not require any interpretation is the question whether two sounds A and B sound *identical.* Only the belief of perfect identity should lead to the answer "yes", in any other case the answer would be "no". A test setup for this question usually involves the presentation of pairs of sounds, where sounds are sometimes identical (e. g., in 50 % of the cases), but different in the other cases. Participants have to decide between the options "same" or "different" in the form of a two-alternative forced-choice experiment (Luce, 1993). If they are able to discriminate the two, their performance will be well above chance level, which serves as the baseline for the test. A variation of this setup is the double-blind-triple-stimulus-with-hidden-reference method, where first an original sound A is played, and then two sounds are played in random order, one of which is the original sound A, and one the different sound B (Yang, Kyriakakis, & Kuo, 2005, p. 82). If participants are unable to detect the difference between an original recording and its processed form, i.e., if they only perform at chance level in the test, there is typically no need for improvement, and no need to ask any more detailed questions.

Just noticeable difference (JND) is a measure for describing how much two observations can differ so that a certain percentage of people will not be able to detect the difference (Luce, 1993). The result of such an evaluation can be plotted as a curve, with the difference between two items at the x axis and the probability of correct detection at the y axis. The probability is typically the percentage of test subjects that were able to detect changes for a particular difference, and should approach 100 % for large differences. To obtain useful probability estimates, a sufficiently high number of trials — in the order of magnitude of 100 — is necessary (Luce, 1993). The concept of JND requires that a difference measure exists as a one-dimensional property, which can be controlled directly and continuously. Therefore, typical tests for JND refer to differences in loudness or frequency only, rather than multi-dimensional differences in timbre.

If a test for perceived identity reveals that almost all participants can hear a difference, little useful knowledge is gained, because nothing was asked about the amount of difference or the practical implications. Other questions have to be asked in order to quantify the perception of listeners. For example, an experimenter may just ask for the *quality* (i.e., the "goodness") of a sound. This can be done either in the form of a relative comparison, providing the original recording as a reference, or in the form of an absolute rating for just a single sound.

The question for quality is simple in its form, but not always easy to answer. Rating quality requires a test participant to integrate various aspects and consider their importance. Given only the task to rate quality, some participants may be asking for more detailed instructions. Instead of quality, a more specific dimension of a sound can be rated, such as *realism*, *smoothness* or *clarity.* However, this pre-selection of criteria is already imposing a certain view on the evaluation and prevents participants from coming up with a more intuitive overall rating[2]. Also, specific attributes may not apply to all sounds.

Realism is a particularly problematic criterion for people to rate. For example, the degree of realism of an abstract synthesizer pad is completely arbitrary, and the participants' idea of a realistic gunshot may be quite far away from actual reality, as it was likely shaped by the sound design of action movies.

Instead of using only one dimension, it is sometimes useful to let the subjects rate several dimensions. The main problem with this approach is that it dramatically increases the time required to take the test. In addition to that, it may be frustrating for participants to work through a large number of seemingly similar questions.

As an alternative to "quality", which is a mostly technical term, the experimenter can also ask "how much do you like sound A?". This question targets the emotional impact of a sound more directly, and thus may be easier to answer than the question for quality; regardless of someone else's criteria, a participant can decide how much he or she likes a sound. The problem with likability is that it only applies to sounds that are nice in the first place. For sounds of pain, sickness and horror, other questions would have to be found.

7.2.2 MUSHRA Tests

A popular test for comparing the quality of different audio coding methods is the *MUSHRA* test (Multiple Stimuli with Hidden Reference and Anchor) (International Telecommunications Union, 2003). Several variations are presented to the subject in parallel, with the possibility to listen to each one multiple times. The quality of each clip is marked on a continuous scale which carries additional labels ("excellent", "good", "fair", "poor" and "bad"). In addition to the samples under examination, an uncompressed version is added as a hidden reference, so that the reliability of the judgments can be assessed. Additional degraded versions of the signal are usually added as low-quality "anchors", so that the quality ratings for other signals can be compared against those standard stimuli. A typical anchor is a 3.5 kHz low-pass filtered version of the original (Vincent, Jafari, & Plumbley, 2006). The MUSHRA test setup is best suited for evaluations in which several versions of the same signal are to be compared, and has been used for the comparison of compression codecs in the past (Stoll & Kozamernik, 2000).

7.3 Sound Element Test Method

An evaluation of sound elements was conducted with human listeners in order to reveal strengths and weaknesses of the proposed model. It was implemented as an online MUSHRA test, with a focus on overall sound quality. A total number of 22 sounds from different domains was included in the test, and each was provided in the original and in an encoded form.

[2] Lu et al. (2004) have asked subjects to rate synthesized sound textures according to *smoothness* and *variation*. For each of the two values, a score of 1 ("bad"), 2 ("acceptable") or 3 ("satisfying") could be given. They have reported average values of 2.55 for "smoothness" and 2.36 for "variety", but neither of these values relates specifically to how much people *liked* the textures. Also, the did not provide any reference sounds or anchors to increase inter-subject agreement.

7.3.1 Corpus for the Evaluation

A corpus of test sounds was assembled for the evaluation. The sounds were collected from freely available sources, most of them are available with a Creative Commons license. The sounds were chosen to represent a wide variety of sound types relevant for different applications, including speech synthesis, music and video gaming.

All sounds in the corpus are monophonic sounds, containing one source only and no disturbing background. The recordings contain minimal or no reverb and do not contain any other effects. A list of the 15 clips used in the evaluation of directly converted sounds is presented in Tab. 7.1. A list of the 7 clips used in the evaluation of parametrically morphed sounds is presented in Tab. 7.2.

sound	**source clip(s)**	**presented variations**	**description**
barking dog	Source clip from the recording “BigDogBarking_02.wav” © by the user “mich3d” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 512, noise only).	Barking of a large dog. Low pitch, mostly noisy with some turbulent harmonic components.
buzzing bee	Source clip from the recording “20100424.bee.wav” © by the user “dobroide” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 512, harmonics+noise).	Buzzing sound of a bee. Mostly harmonic, some variation in pitch.
creak	Source clip from the recording “Creak_3.wav” © by the user “HerbertBoland” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 512, harmonics+noise).	A creak of a wooden door.
flute	Source clip from the recording “little_E_samplefile.mp3” © by the user “kerri” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 1024, harmonics+noise).	A breathy flute tone with some vibrato.
footstep	Source clip from the recording “heels_wind.aif” © by the user “tigersound” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 256, noise only).	A single footstep of a shoe with heels on solid ground. Very subtle background wind noise.
guitar	Source clip from the recording “parker_piezo_a.wav” © by the user “sleep” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 1024, harmonics+noise).	A single plucked guitar string.
gunshot	Source clip from the recording “RemingtonGunshot.wav” © by the user “fastson” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 256, noise only).	A gunshot from a Remington gun, recorded outdoors.
hawk	Source clip from the recording “red-tailed_hawk.mp3” © Macaulay Library, http://macaulaylibrary.org, (clip id = 4177), recorded by Robert C. Stein	Original, lowpass filtered, parametric (analysis window size: 512, harmonics+noise).	Scream of a red-tailed hawk. High-pitched harmonics, noisy modulation, some echo.
large-splash	Source clip from the recording “Water_Splash_Objects_falling .aif” © by the user “Dynamicell” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 512, noise only).	Splash of a large object falling in water. Low-frequency main splash, followed by many higher pitched smaller drops.

sound	source clip(s)	presented variations	description
mooing cow	Source clip from the recording “TwoCows.wav” © by the user “acclivity” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 1024, harmonics+noise).	*Moo*-sound of a cow.
pony	Source clip from the recording “Neigh2.flac” © by the user “acclivity” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 512, harmonics+noise).	Neighing sound of an upset pony. Unsteady pitch fluctuation, inharmonic frequencies, growling components.
rain	Source clip from the recording “Rainfall.ogg” © by the user “abinadimeza” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 256, noise only).	Steady, light rain with some audible drops in the foreground.
singing cuckoo	Source clip from the recording “Cuckoo1.flac” © by the user “acclivity” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 512, harmonics+noise).	Characteristic *coo-coo* sound of a cuckoo. Two distinct sounds, some noisiness, subtle echo.
warbler	Source clip from the recording “CetisWarbler.flac” © by the user “acclivity” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 256, harmonics+noise).	Very short tweeting sound of a bird (cetti’s warbler). High fundamental frequency, rapid frequency modulation.
waterdrop	Source clip from the recording “waterdrop24.wav” © by the user “junggle” of freesound.org.	Original, lowpass filtered, parametric (analysis window size: 512, harmonics+noise).	Single sound of a drop falling into water. Mostly harmonic, very brief transient.

Table 7.1: Test samples in the evaluation for directly parametrized sounds. The comparison is always between the original clip, a low-pass filtered variation of the same clip, and a parametrized version of the same clip.

sound	source clip(s)	presented variations	description
gravel	Four source clips (A, B, C, D) from the recording “gravel_walking.wav” © by the user “tigersound” of freesound.org.	Three presented clips: A (original), B (lowpass filtered), morph created from C and D (parametric).	Different variations of a short footstep on gravel. Slightly different lengths, different degree of granularity.
piano	Four source clips (A, B, C, D) from the recordings “Grandmither_s_Piano_18_.wav”, “Grandmither_s_Piano_19_.wav”, “Grandmither_s_Piano_20_.wav” and “Grandmither_s_Piano_21_.wav” © by the user “Techsetsu” of freesound.org.	Three presented clips: A (original), B (lowpass filtered), morph created from C and D (parametric).	Different variations of a single note played on a slightly de-tuned piano, with noticeable inharmonicity of the higher partial tones.

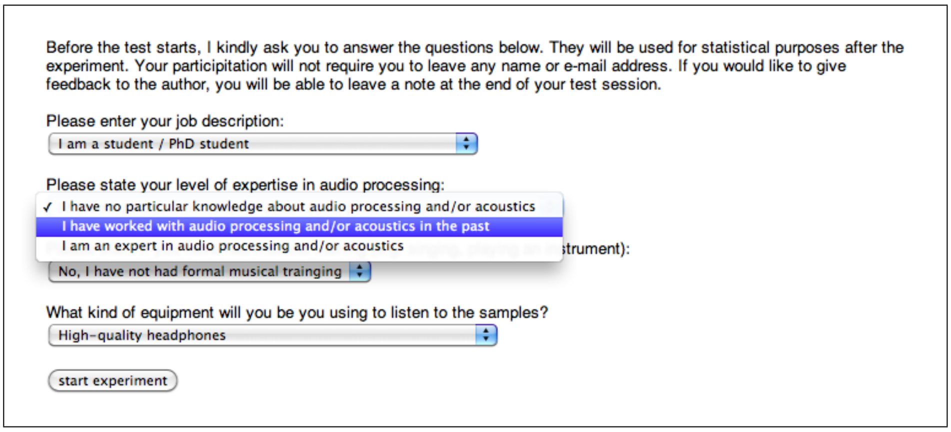
sound	source clip(s)	presented variations	description
rooster	Four source clips (A, B, C, D) from the recordings “Rooster_chicken_calls_2.wav” © by the user “AGFX”, “20070812.rooster.wav” © by the user “dobroide”, “Rooster_Crows.wav” © by the user “promete” and “Rooster1.wav” by the user “acclivity” of freesound.org.	Three presented clips: A (original), B (lowpass filtered), morph created from C and D (parametric).	Different variations of a crowing rooster (four different animals). Sequence of staccato-like tones. Different lengths, some modulation and subharmonics.
singing	Four source clips (A, B, C, D) from the recording “Katy_Sings_Melisma_2.wav”, served by the freesound project and produced by www.digifishmusic.com	Three presented clips: A (original), B (lowpass filtered), morph created from C and D (parametric).	Different variations of a woman singing a steady tone.
song-thrush	Four source clips (A, B, C, D) from the recording “SpringSongThrush.mp3” © by the user “acclivity” of freesound.org.	Three presented clips: A (original), B (lowpass filtered), morph created from C and D (parametric).	Short tweet of a bird (song thrush). Rapid modulation of the frequency, with an interruption between two parts of the vocalization.
thunder	Four source clips (A, B, C, D) from the recording “thunderstorm2.flac” © by the user “Erdie” of freesound.org.	Three presented clips: A (original), B (lowpass filtered), morph created from C and D (parametric).	Different variations of thunder from a longer recording of a thunderstorm. Loud foreground thunder with light background rain.
traffic	Four source clips (A, B, C, D) from the recordings “PassingMotorCycle02.wav”, “PassingMotorCycles02.wav”, “PassingMotorCycle01.wav’ and “PassingMotorCycle03.wav” © by the user “Pingel” of freesound.org.	Three presented clips: A (original), B (lowpass filtered), morph created from C and D (parametric).	Different variations of a motor cycle passing by. Conversion of the synthetic sample was done using a noise-only model.

Table 7.2: Test samples in the evaluation for morphed parametrized sounds. The comparison is always between one original clip, one low-pass filtered variation of a different clip, and a parametrized version, created from two additional different clips.

7.3.2 Evaluation Setup

The evaluation was conducted in the form of an online survey, in order to reach as many participants as possible. The MUSHRA method was used to compare synthetic sound clips to original clips and anchor clips, i.e., clips that are known to have a low quality. The low-quality clips were obtained by low-pass filtering original recordings at 3.5 kHz.

Participants were told that the evaluation was part of a dissertation about “sound textures”. They were asked to provide some information for statistical purposes. This was done using an HTML form that let users select alternatives from a drop-down list (see Fig. 7.1). All participants were asked to select the best matching job description, their level of expertise in audio processing, their years of music education (if any) and



Before the test starts, I kindly ask you to answer the questions below. They will be used for statistical purposes after the experiment. Your participation will not require you to leave any name or e-mail address. If you would like to give feedback to the author, you will be able to leave a note at the end of your test session.

Please enter your job description:
I am a student / PhD student

Please state your level of expertise in audio processing:
✓ I have no particular knowledge about audio processing and/or acoustics
I have worked with audio processing and/or acoustics in the past
I am an expert in audio processing and/or acoustics

strument):
No, I have not had formal musical trainging

What kind of equipment will you be you using to listen to the samples?
High-quality headphones

Figure 7.1: Screenshot from the online evaluation: Participants were asked to enter their profession, level of expertise, their years of music training (if any) and their choice of listening equipment.

the kind of equipment they would use for the evaluation. Before the actual evaluation started, the following text (including the emphasis) was shown to the participants:

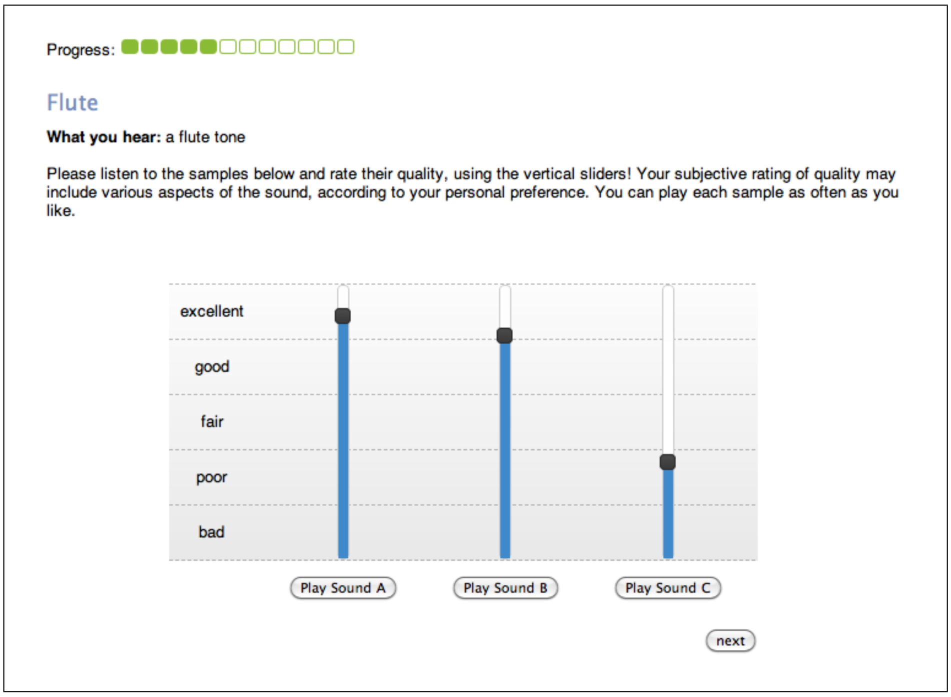
> "You are now going to hear 12 short sounds. Each one is presented in three variations. You will be asked to listen to the samples and rate their *quality*, using vertical sliders. Your subjective rating of quality may include various aspects of the sound, according to your personal preference.
> All samples are provided with the same sampling rate. You can play each sample as often as you like."

Each clip was then presented on a separate page and had to be rated before the participant proceeded to the next clip (see Fig. 7.2). The short name of the sample, as well as a brief description of the sound, were printed on each page. The description was provided so that the subjects could get an understanding of what a sound was supposed to sound like. However, these descriptions were kept very brief (e.g. "a flute tone" or "a single shot from a Remington gun"). The following instruction was repeated on each page:

> "Please listen to the samples below and rate their quality, using the vertical sliders! Your subjective rating of quality may include various aspects of the sound, according to your personal preference.
> You can play each sample as often as you like."

As recommended in the MUSHRA setup, a continuous scale was used, with a range from 1 to 100 points, but the resolution of the sliders was not visible for the participants. The scale was visually divided into five sections that carried the labels "bad", "poor", "fair", "good" and "excellent", to provide a rough guideline for the meaning of slider settings. When a new test sound was presented, all sliders were set to a resting position at "50". A "play" button was placed directly below each of the three sliders, with which the sounds could be played instantly and repeatedly. The technical functionality and browser compatibility was tested beforehand with a small group of test subjects.

Each participant rated up to twelve of the twenty-two clips, which were randomly selected from the larger set of 22 evaluation sounds, and presented in random order.



Progress:

Flute

What you hear: a flute tone

Please listen to the samples below and rate their quality, using the vertical sliders! Your subjective rating of quality may include various aspects of the sound, according to your personal preference. You can play each sample as often as you like.

Figure 7.2: Screenshot from the online evaluation: For each clip, sliders could be manipulated to indicate the subjective rating of quality. The progress of the survey was indicated by a progress bar at the top.

They rated less than twelve clips only when they quit the procedure before the last sound.

7.4 Survey Results

The online survey of the sounds reveals significant differences between types of sounds: while for some clips, the difference between originals and synthetic versions could be detected reliably by the subjects, there is a huge difference in other clips. The results of the evaluation are presented here in terms of median values and quartiles, rather than mean values and standard deviations.

From 169 survey data-sets, 45 were excluded from the evaluation. This was done when they had only rated less than five clips, when they had left the sliders untouched (e.g., “50”, “50”, “50”) or moved several sliders to obviously unreasonable settings (e.g., selecting “1” for the original sound). Participants were also excluded when they specifically reported technical problems in the comment field. 124 data-sets were kept, most of them containing twelve rated clips. Each clip was rated by 63 to 68 participants.

Fig. 7.3 shows the statistical properties of the participants in the online evaluation. 50.0 % of the participants were students or PhD students, 24.2 % were researchers or instructors, 16.1 % were professionals from either audio, media, or gaming-related fields. Regarding their level of expertise in audio and acoustics, 36.3 % of the participants stated that they had no particular knowledge about audio processing and/or acoustics, 44.4 % stated that they had worked with audio processing and/or acoustics in the past, and 19.4 % of participants stated that they were experts in the field. Answering to the question about music education, 43.5 % of participants stated that they have had

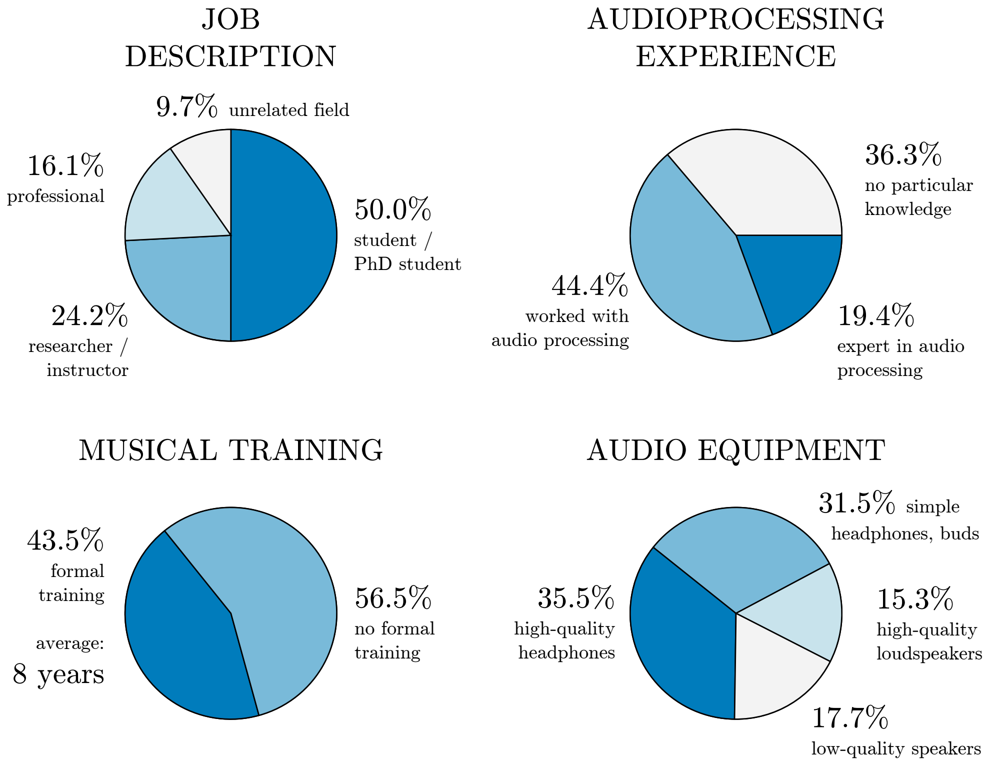


Figure 7.3: Statistical properties of the group of participants, obtained from the online questionnaire.

formal training in music, such as playing an instrument, with an average of approx. eight years of training. 35.5 % of participants stated that they used "high-quality headphones" to listen to the clips, 31.5 % used "simple headphones or in-ear phones", 15.3 % used "high-quality loudspeakers" and 17.7 % used "low-quality speakers", such as built-in laptop speakers.

On average, participants did not use the full range of points (1. . .100), and they tended not to give the maximum score to the originals. The highest rated original samples were the "rain" sample, which scored a median rating of 84 points and the "large splash" sample (80 points). The original sample of the mooing cow received only a median score of 50 points, possibly because of the slightly reverberant environment in which it was recorded, and the "songthrush" and "footstep" samples received only 55 points each. All other samples had various ratings in between. Participants used different styles of scoring. Most made subtle distinctions between samples, while a few tended to use the extreme ends of the scale, giving a score of "1" to the sound they liked the least, "100" for the best sound and "50" to the sound they felt was in the middle.

The differences between originals and anchor sounds, i.e., low-pass filtered sounds, varied greatly between different clips. As could be expected, the difference was greatest for clips that had very strong high-frequency components above the 3.5 kHz filtering frequency of the anchor sounds. The difference was very extreme for the median ratings of the "rain" sample (29 points vs. 84 points), and also for the "large splash" sample (38 points vs. 80 points) — the same clips that also received the highest ratings for the originals. Two explanations for this correlation are possible: either a high quality of the original sample leads to a clearer perception of the degradation, or a strongly degraded anchor boosts the perceived quality of the original in direct comparison. It is likely that both effects contribute to the ratings.

For the "singing cuckoo" clip, the median rating of the anchor was almost identical to the original (61 points vs. 60 points), even though individual participants occasionally rated the two alternatives quite differently. The outcome is understandable, because the original clip has barely any energy above the 3.5 kHz threshold, and thus the two clips are in fact almost identical.

In the graphs presented in this chapter, the results are shown in the form of a box-and-whisker plot, where the box indicates the two middle quartiles of the given scores, i.e., 50 % of the participants have given a score within the range of the box. The median value of the scores is indicated as a line within the box. For symmetric distributions, the median is close to the center of the box, while a placement off the center indicates a skewed data distribution. The lowest and highest extremes of the data distribution are indicated by two lines ("whiskers") extending from the box in both directions (McGill, Tukey, & Larsen, 1978). Since the data in the sound object evaluation represents subjective judgments, some extreme ratings can be expected, even when the mean or median values converge to stable and reliable values.

7.4.1 Results for Directly Converted Clips

The best synthetic sound, according to the survey, was the "flute" sample, which scored a median of 68 out of 100 points in its quality rating. This shows that the harmonic component and the noise component fuse well to a convincing breathy sound. The temporal resolution of the model (30 breakpoints) was good enough in this case to capture the vibrato of the flute. Interestingly, the original flute was rated marginally lower (with a median of 68 points as well, but with lower scores above the median value).

The brief tweet of the "warbler" sample, which contains extreme harmonic chirps, was rated quite high as well (57 points), with only little difference from its original sample (63 points). The corresponding anchor sample was rated much lower (43 points).

The synthetic "guitar" sample is among the best rated synthetic clips, too (60 points vs. 68 points for the original). The stable harmonic structure of the resonating guitar string presents no particular challenge to the model, which even captures the slow change in timbre from start to end quite well. The only audible difference between the original and the synthetic version is the sharpness of the plucking noise right at the start of the sample.

The "hawk" sample received acceptable ratings (50 points vs. 64 points for the original) — which is remarkable, since the approximation of the hawk's scream through harmonics and noise is physically quite incorrect. The spectrograms are shown in Fig. B.2. Screaming noises are very difficult to analyze in detail, because they are prominent examples of nonlinear dynamics and "deterministic chaos", rather than true stochastic noise (see Subsection 2.1.5). The model approximation through only harmonics and stochastic noise still seems to work well enough. Likewise, a subtle smearing effect of the sharp changes in vibratory mode, which are present in the original, did not lead to a significant decrease in the score for the synthetic sample. The low-pass filtered anchor sample was rated much lower in this case, likely because most of the high-frequency content of the scream is missing.

Dog barks, just like screams, contain aspects of nonlinear dynamics. But again, the approximation of the “dog bark” original through just filtered stochastic noise proved to be good enough to produce an acceptable rating of the synthetic clip.

For the “gunshot” sample, some degradation was identified by the test subjects, although it only led to a moderately lower rating for the synthetic clip. The synthetic gunshot lacks some sharpness at the starting transient, and also some subtle grainy structure of the original, which may originate from echos bouncing back from various surfaces in the distance after the shot is fired. The synthetic model instead introduces a mild flanging effect.

The “cuckoo” sample did not present any particular challenges to the synthetic model, as the cuckoo’s voice can be well approximated by harmonic components and noise. Some minor degradation of the synthetic sound happens in the gap between the higher and the lower tone, where the spline model interpolates the pitches and causes a subtle gliding pitch to be created, while in the original sample, only the damped echo of the first tone is heard in the gap. The spectrograms are shown in Fig. B.7. The low-pass filtered anchor sound of the “cuckoo” sample was rated higher than the synthetic version, almost identical to the original. This is due to the fact that the original does not have any significant frequencies above 3.5 kHz, and therefore hardly any loss of fidelity is encountered.

The original recording of the “footstep” sample has a high amount of rumbling, low-frequency noise mixed into the footstep sound, which is degraded in the synthetic version by the spline-smoothing of the spectral envelope. As in other sounds that have an impact-like quality, the starting transient of the step is slightly blurred. However, the synthetic “footstep” sample still received acceptable scores from the evaluation participants. The anchor sound was rated lower than the synthetic version.

The “rain” sample received the lowest rating of the directly converted sounds, compared to its original (23 points vs. 84 points). The reason for this is most likely the lack of granular detail in the synthetic version, as details below the sound object level cannot be accurately rendered by the spline-based model. Where the recording contains dozens or even hundreds of so-called micro-transients, the model can only capture gradual changes in the overall characteristics. The model assumption that one set of parameters corresponds to one acoustic event is violated in this case. It is interesting that the anchor clip — which contains granular events — was rated almost as bad as the synthetic version (median: 29 points). However, most of the characteristic detail of the rain sample is contained in precisely the high frequency bands that were filtered out in the anchor sound, thus leaving only noisy mush in the anchor sound.

The “large splash” sample of a heavy object being dropped into water was also rated extremely low, and lower than the anchor sound. The clip suffers from the same problems as the “rain” clip, because the original “large splash” sample also contains a lot of detail from drops falling onto the water surface after the splash. Additionally, the original contains some subtle turbulence sounds, which act like many tiny, overlapping quasi-harmonic sounds and are therefore impossible to capture accurately as just one fundamental harmonic tone. Other than in the “rain” sample, much of the detail is contained in the lower frequencies, so that the low-pass filtered anchor sound does not suffer as much from the degradation as in the “rain” sample.

For the “creaking door” sample, participants were able to identify the difference between the original and the synthetic version clearly, as well (27 points vs. 72 points). However, the reasons for the loss of quality are more subtle in this case than for the

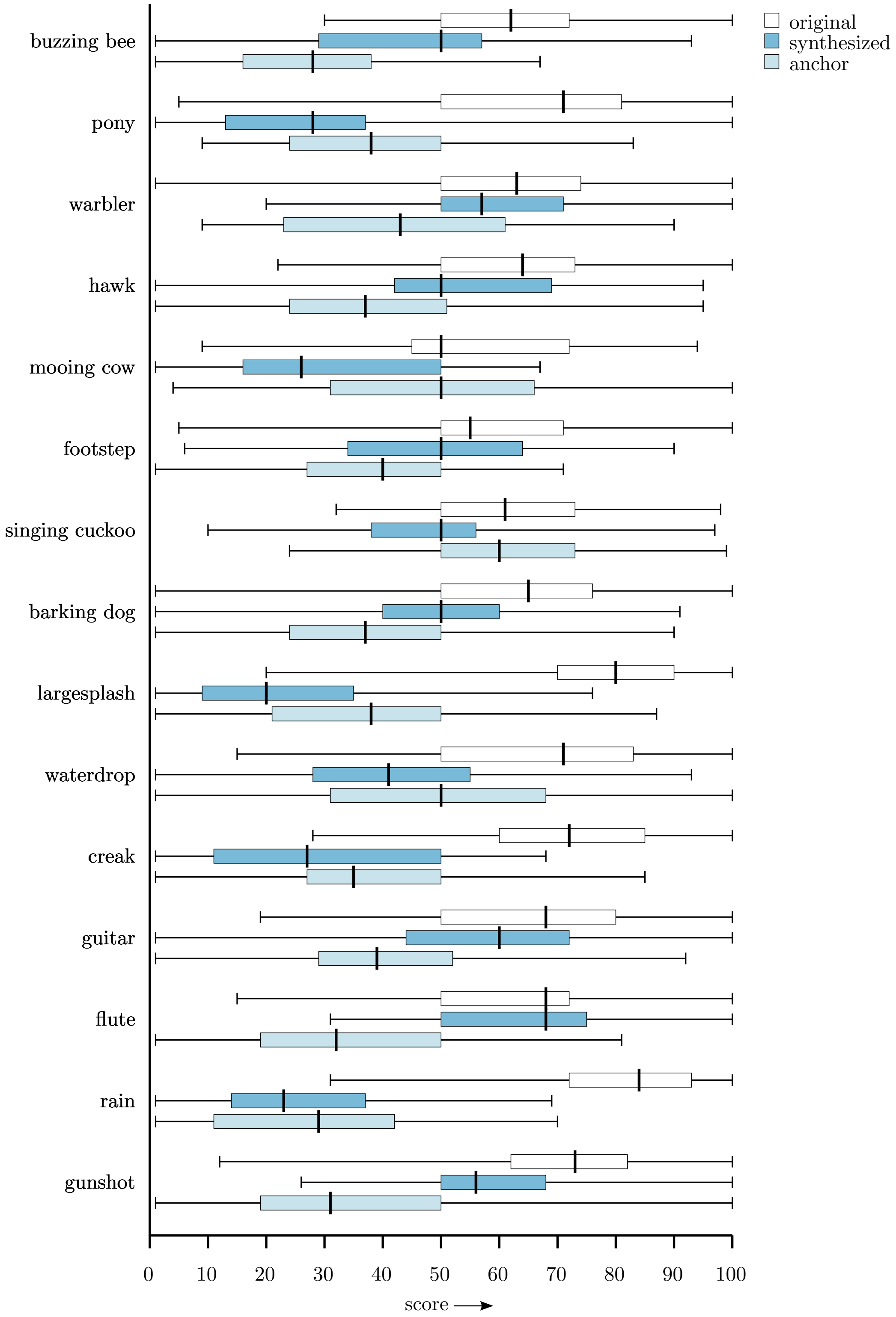


Figure 7.4: Box-and-whisker plot for the sounds that were converted directly from one original. The boxes indicate the ranges between the lower quartile and upper quartile, with the medians indicated in between. The "whiskers" indicate the lowest and highest scores given.

"rain" or "large splash" samples. The creak consists of a starting sound, a long, high-pitched middle part, and an end part, in which the pitch drops again. The middle part is quite accurately modeled through the harmonic components, although the original sample has some instabilities in pitch, which are too fine-grained to be picked up by the analysis. The problematic portions are the start and end parts, which contain pulsed spikes of sound in the original[3]. The pulses are not rendered well by the synthetic sound model. A more subtle effect of the original is that the vibrations go through different mode changes at the start and end, depending on the speed with which the door is moved.

The "pony" clip also revealed some limitations of the model. While the hoarse, raspy timbre of the neighing is captured quite well by the harmonics and noise components, an increasingly large modulation of the sound towards the end is not modeled with enough detail, thus changing the nature of the sound significantly. At the end of the original "pony" recording, the pony makes a characteristic *'ppppprrrrrr'* sound. The temporal structure of this pulsed sound component is approximated through residual noise, but that does not lead to a convincing rendition.

Other directly converted synthetic sounds received scores about half the score of their respective original, including the "cow", "water drop" and "buzzing bee" clips. The synthetic "mooing cow" clip suffers from the missing glottal pulses, due to the lack of phase alignment of the partials, with the consequence that the *'moo'* is not perceived as a properly voiced animal sound. This is similar to the lack of pulses in the "creaking door" clip. The synthetic clip also has a slightly disturbing noise component near the start of the sample, where some energy, which should have been assigned to a rapidly rising harmonic fundamental, was assigned to the residual.

Test participants were also able to identify a subtle difference between the original "water drop" clip and its synthetic counterpart. The quick rise of the tonal component in the brief *'blip'* noise was captured quite accurately by the model. However, the original sound contains two sharp impulses — one from the impact of the main drop and one higher-pitched from the impact of a secondary drop — which are apparently too sudden to be picked up by the smooth spline model. As a result, the transient are slightly blurred in time, and some energy around the transients is falsely assigned to the noise component.

The "buzzing bee" sample suffered a little from a lack of temporal detail. Although the timbre of the buzzing seems to be approximated very well, there is some fast fluctuation in the original sound that gets smoothed out by the spline approximation.

7.4.2 Results for Parametrically Blended Clips

In addition to the directly converted sounds, seven clips were tested in which the synthetic samples were synthesized from a mixture of two parameter sets. The clips in this part of the evaluation reveal some problems that can occur when morphs and changed versions are created from the inputs, but they also show that such morphs work well for a variety of sounds, especially when they are relatively short. In the evaluation, one original sample is always compared against a different synthetic sound of the same kind, created from two different samples, and against a low-pass filtered

[3] The effect is the same as in bowed string instruments: as the hinge of a wooden door is turned, the friction causes the surfaces to stick together. The tension is released periodically, causing the surface to snap back and cause a pulsed sound.

anchor sound, created from yet another clip. This was done so that the synthetic clip would not stick out from the group as the only different clip. However, this setup tends to make it more difficult for test participants to judge the differences between clips, as well as their quality. Also, some differences in the scores may be attributed to the fact that — by coincidence — some sounds just sound better than others. This should be kept in mind, although great care was taken to select sounds of similar acoustic quality for the test sessions.

The "song thrush" clip of a bird tweet, which contained a short sequence of tones, was processed very well by the model; the median score of the synthetic sample is slightly higher than the median score of the original (59 points vs. 55 points). The synthetic clip is a morph of two very similar instances of the whistling song thrush. Both contain mostly purely harmonic whistling, with some rapid changes in frequency, and some atmospheric background residual, which is disjoint from the bird's tweet. Because of the high similarity between the input clips, there is a direct correspondence between features of the two instances, and therefore a morph works well[4].

The crowing in the "rooster" clip was rated worst of all clips (12 points for the synthetic version vs. 76 points for the original), and even worse than the directly converted "rain" and "large splash" clips. A number of factors likely contribute to this low score. Firstly, the sound of a crowing rooster is a problematic case for the model, even in the case of direct conversion. The rooster's call is a staccato of several cries with short interruptions, which complicates its modeling by a continuous spline. Even the use of a spline with 30 breakpoints is barely enough to allow for exact tracking. The spectrograms are shown in Fig. B.21. In that sense the input sound is a violation of the model assumption, which strictly refers to one sound only. There are also some elements of chaos, nonlinearities and sub-harmonics in the original rooster call, which are difficult to analyze and approximate using only harmonics and stochastic noise (see Subsection 2.1.5). Additional complications arise from the blending of two rooster calls: since the calls of two different animals differ strongly in length and in their sequence of "syllables", their correspondence is most likely invalid. The half-way blend between the sounds therefore contains parts that appear smeared, as they contain influences of non-matching syllables or pauses.

The synthetic "traffic" sample, containing the sound of a passing motor cycle, received a low rating as well (29 points vs. 73 points for the original). In contrast to the original sample, and also to the anchor sample, the blended synthetic sound does not contain a clear harmonic component. The low-frequency buzzing of the motorcycle engine, which was present in the raw recording, was very difficult to capture by the harmonic tracking mechanism, and therefore the conversion was performed using a noise-only setting. The synthetic sample therefore contains a quite convincing *'whoosh'* sound of the vehicle passing by, in which the characteristics of two input sounds are nicely blended, but the lack of an actual motor sound is problematic.

The synthetic piano sample was rated relatively low, compared to its original (31 points vs. 68 points). The strings of the originally recorded piano are slightly out of tune, resulting in two different sets of harmonic partials. Since only one of the groups can be encoded as the fundamental frequency, the energy contained in the non-

[4] The four samples used from the song thrush recording (one original, two for the morph, one for the anchor sound) had slightly different loudnesses. The synthetic sound turned out to be louder than the two other sounds. Some participants noted that they found it hard to assess the importance of this change in loudness.

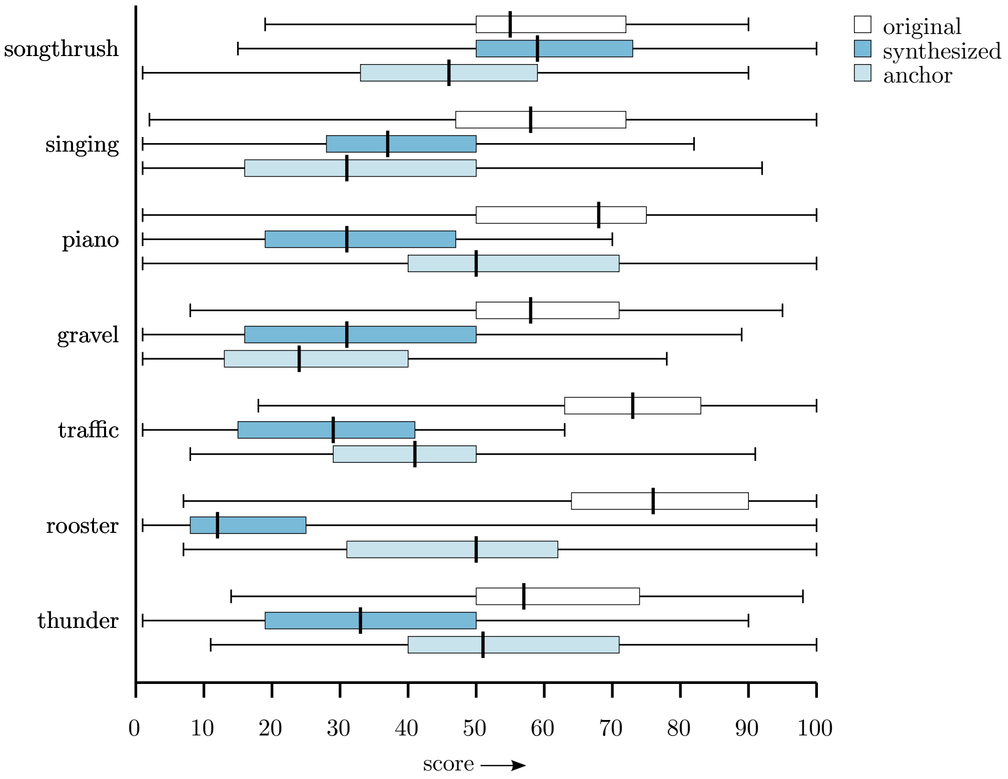


Figure 7.5: Box-and-whisker plot for hybrid sounds, created from two originals. The boxes indicate the ranges between the lower quartile and upper quartile, with the medians indicated in between. The "whiskers" indicate the lowest and highest scores given.

matching partials is missed by the harmonic analysis, and is instead encoded as residual noise. As a consequence, the de-tuned character of the piano is gone in the synthetic tone, and wrong noisy components are added to the mix. A lower rating is therefore understandable.

The "thunder" sample suffers from similar problems as the crowing rooster: since the blended sound is created from two quite different thunder recordings with non-matching temporal structure, the synthetic sound is strongly degraded (33 points vs. 57 points for the original). This is reflected in the low rating for the synthetic clip. As could be expected, the anchor sound receives a good rating in this case, because most of the structure of the thunder sound is contained in the low-frequency region, which is not degraded by the low-pass filtering. Besides the problem of correspondence in hybrid sounds, thunder also has some acoustic properties that are challenging for the model. Thunder sounds can have several strong transients and crackling sounds, originating both from the primary sound of the electric discharge, and from the reflection of echos in the landscape. The temporal resolution of the splines limits the ability to capture these transients, which can lead to flanging artifacts in the output.

The two remaining clips, "gravel" and "singing", received better ratings than their respective anchor sounds, although participants still found it easy to distinguish them from un-altered recordings. The four clips used to produce the test samples in the "gravel" test each contain one footstep on a gravel surface. Although the footsteps contain some internal temporal structure, they are all similar enough to avoid the problem of correspondence. The main problem in the “gravel” clip comes from the grainy micro-transient structure of the recordings, which gets smoothed out in the synthetic sound, causing a subtle flanging effect.

7.4.3 Comments Made by the Participants

The online evaluation contained a comment field that allowed participants to enter any additional observations if the wanted to. Of the people who chose to enter something into the comment field, some entered greetings or humorous comments about the samples, some were curious about details of the experimental setup, e.g., whether the order of presentation was random[5]. A few reported various forms of technical problems during the test. Depending on the nature and severity of the problems mentioned, some of these sessions were excluded from the results. A few participants used the comment field to try to hypothesize about the reasons why the model performs poorly in some cases — even though they had no insight into the model or the algorithms used for the synthesis.

Some people reported in the comments that they found it hard to distinguish different samples in some cases, even though there were no truly identical samples used in the whole evaluation. However, this is not surprising: the better the conversion works, the more difficult it is generally to distinguish between originals and synthetic versions. Some participants mentioned specifically that they had trouble comparing very short sounds, such as footsteps. While there was in fact little difference between the “footstep” samples, other forms of presentation could make it easier to perceive differences. For example, in future evaluations, short sounds could be presented in groups, rather than as isolated sounds.

Some people mentioned that they found it difficult to decide what sounds are supposed to sound like, especially for exotic sounds like the screaming hawk. This is a well known property of the MUSHRA test setup, which uses the concept of the hidden reference. For the morphed sounds, presenting the original reference, as some people suggested, is not even possible — simply because there is no single original. One participant pointed out that each of the three versions of the “gunshot” sample could be the “correct” one, depending on the type of gun, or the distance between the gun and the microphone. For example, the synthetic sample has a softened transient, which would be perfectly correct for a gun recorded from a long distance. However, the same sound would be bad for a sound recoded from a short distance.

Several people commented that they had difficulties to find appropriate criteria for their assessment of quality, especially when all the sounds were clearly different. Nevertheless, they managed to find scores that reflected their intuitive judgment.

7.4.4 Effects of Using Low- or High-Quality Equipment

All participants were asked to select the type of equipment they would use for the evaluation, so that results from sessions conducted with low-quality equipment (“Low-quality speakers (e.g., laptop speakers, monitor speakers)”) could be examined separately from the other entries. Although the group of participants who used low-quality equipment was much smaller than the other group, some statistical observations can be made.

[5] Yes, it was.

	originals	**synthetic**	**anchor**
mid-high quality	65.4	40.4	41.4
low quality	63.4	44.3	42.1
all	65.1	41.1	41.5

Table 7.3: Influence of the equipment quality on the overall rating preference of participants (mean scores of all clips).

Regarding the mean scores of all clips for original samples, synthetic samples and anchor samples, no strong difference was observed between the two groups. As could be expected, there was a weak global trend to rate synthetic sounds and anchor sounds higher when the equipment had low quality (see Tab. 7.3).

However, there was a clear difference between the equipment-based ratings for some individual sounds. For example, subjects with low-quality equipment rated the anchor sounds of the "buzzing bee", "guitar" and "hawk" clips much better than subjects in the other group, likely because the negative effects of low-pass filtering were less audible in comparison. For the "large splash" sample, participants with low-quality equipment gave approximately the same score to the synthetic version and the anchor sound, while those with higher-quality headphones found the synthetic version to sound much worse.

7.5 Compression Performance

With relation to compression methods, a common task is to determine the relationship between a transmission data rate and the resulting audio quality. *Rate-distortion* theory offers a method to determine the theoretical lower bound of the data rate necessary to encode a source signal with a given distortion. A measure for distortion between source and target signals needs to be provided. For images, a common measure is the squared error between the pixel color values of the source and target images. For sound, the squared error between samples can be used. For every type of source signal, a characteristic rate-distortion curve can be computed, which describes the mutual relationship between the two: when large distortions are allowed, the data rate is low, but when the distortion is zero, the required data rate has a maximum, which is linked to the entropy of the signal. Although the theory does not define how an optimal encoding can be developed, it provides a way to assess the effectiveness of a given encoder (Cover & Thomas, 1991). Xu and Yang (2006) have used a rate-distortion measure to optimize the codebook of an MP3 encoder.

To obtain a rate-distortion curve for the parametric model, the first question would be how many bits are required to encode sound without any distortion. However, since the low-dimensional model parameter space cannot cover all possible variations of sounds (see Section 6.8), no general answer can be given to this question. Some sounds, like single piano tones, can be approximated almost perfectly even when the model is configured to use only a low temporal and frequency resolution. For other sounds, even increasing the data rate tenfold may not give acceptable results. Rate-distortion is well suited for measuring the effects of quantization, a technique that is not addressed yet by the parametric encoding. After all, the limitations of the model originate from the choice of parameters, even when no quantization is applied.

To get an understanding of the relation between audio quality and storage requirement, it would be necessary to conduct other listening tests with the parametric model. Since the resolution of the model was kept constant in the evaluation, no insights can be derived in how far the perceived quality varies with different model configurations. Based on the evaluation data, only static observations can be made. The "flute" sound, which was encoded using 1706 parameters[6], was rated equally good as the original flute sound, which is composed of 54 386 samples. This corresponds to a compression rate of approx. 1:32, which is a pessimistic estimate of the compression, without any quantization or redundancy coding. It is reasonable to assume that the perceived quality would go down when the resolution of the model is lowered. However, the dependency would likely be nonlinear. For example, a certain temporal resolution is required to capture the vibrato of the flute. As long as the sampling resolution is sufficiently high, the degradation is not very noticeable. However, once the resolution drops below the minimum number of points required to resolve the vibrato, a drastic reduction in perceived quality can be expected. More experiments would be necessary to investigate the relationship between sound quality and model size, conducted with different sound types.

[6] 1 coefficient for the length, 31 coefficients for the fundamental trajectory, $2 \cdot 837$ coefficients for the two envelopes.

Chapter 8

Conclusion

8.1 Contributions of this Work

This thesis has spanned a large range of topics, ranging from basic acoustic phenomena, human physiology and psychology to newly developed sound analysis and synthesis algorithms. All of these aspects are essential to the fascinating and multi-faceted research field of sound textures. As the past chapters have shown, this field is not just about algorithms for “making noise” — it is also about the structural understanding of sound sources in our everyday environment.

In this chapter, the main contributions of this thesis will be discussed, including theoretical, conceptual and algorithmic aspects. The inherent conflict between model simplicity and model generality will be examined in detail. Finally, future directions will be named, both for sound texture research in general and for parametric modeling techniques.

8.1.1 Advances in Parametric Modeling

The parametric sound object synthesis (PSOS) model introduced in this thesis is based on the concepts of spectral modeling, but adds all the advantages of a fixed parameter space. While basic spectral modeling typically decomposes sound into a collection of hundreds of line segments, the PSOS model offers one set of parameters with known semantics. Mixing different sounds becomes possible simply by interpolating parameters in parameter space.

The new model also removes much of the redundancy contained in the basic spectral model: instead of encoding the trajectories of harmonic partials separately, only the fundamental frequency is encoded, and the structure of partials follows automatically. Not only is this a more compact representation, it also keeps the partials locked to each other and prevents them from causing artifacts when they drift apart. Of course, the idea to use the fundamental to drive a system of harmonic overtones is not new, but the extended concept of modeling the time-varying properties of harmonics and noise residual for whole sound objects is a change which offers a whole new range of possibilities.

The concept of the parametric sound object is what sets this work apart from spectral modeling and traditional frame-based codecs. Long sounds are obtained not by a concatenation of hundreds of frames, but simply by changing the length parameter of the parametric model. While changing any of the stored time-frequency-phase triplets of a regular spectral model will just disrupt the continuity of the sound, manipulating parameters in the parametric model results in a meaningful change of overall acoustic properties.

Perhaps the biggest advantage of the parametric model is the ability to apply learning mechanisms to sounds, such as clustering or principal component analysis (see Subsection 5.1.4). Such an analysis can reveal important correlations between various aspects of pitch, timing and spectral characteristics. But, since the parametric model is also a synthesis model, sets of parameters can also be converted back into realistic sounds: the mean coordinate of a cluster of sounds therefore is not just a feature vector for classification, but corresponds to an actual sound that can be played back.

8.1.2 Contributions to Sound Analysis and Sound Understanding

Even though this work has sound synthesis as its main research goal, many aspects of parametric modeling are clearly relevant for a range of analysis applications. The decomposition of complex sounds into partials and noise components reveals a structure that is not accessible otherwise, a structure that is in many cases surprisingly simple. The conversion of sounds into parametric structures, as implemented in the sound object editor software (see Subsection 6.3.1), can be a useful tool in acoustic research, offering not only de-composition, but ultimately understanding. Several further tools could be developed from this, including tools to “sculpt” sounds visually or to examine precisely the acoustic properties of individual partials or transients.

Parametric models could also be implemented into sound recognition algorithms or database retrieval tools. The parameter space of sound objects provides a very convenient distance measure, which is essential for judging the similarity or identity in any retrieval task. In a database of sound effects, parametric representations of each sound could be compared efficiently, because the parameter vectors are very small data structures and all have the same size, regardless of the length of a sound.

8.1.3 New Goals for Sound Texture Research

In the research field of “sound textures”, some of the biggest breakthroughs have yet to happen. The algorithms presented in the past years are limited solutions for particular classes of sounds, and the concepts of how the research should proceed have been vague so far. One of the main goals of this thesis has been to structure the existing research conceptually, to point out existing weaknesses of algorithms, and to set new, ambitious goals.

There have been various attempts of definitions for sound textures in the past, each with its own list of phenomena that should be included or excluded (see Appendix A). While restricting the domain to a manageable range of sound types is certainly helpful, the goal must be to include further classes of sounds and make the algorithms more powerful, not to make the definitions more restrictive. The main problems of existing algorithms have been described in Chapter 4. They include various problems of discontinuity, poor awareness of the long-term structure, an inability to process overlapping sounds and a tendency for repetition. This thesis has aimed at providing theoretical insights about the origins of these problems. As discussed in Section 4.6, fundamental limitations exist in some modeling concepts that cannot be overcome by using faster computers or differently configured analysis parameters.

As shown in the overview in Section 1.2, something that is missing in most publications about sound textures so far is a thorough examination of requirements. This is also the reason why it is difficult to assess in how far proposed systems have succeeded or failed. In Section 4.1, five requirements for sound texture analysis and synthesis have been named: *similarity*, *continuity*, *variability*, *compression* and *controllability*. From these requirements, a list of technical and conceptual problems has been developed in Section 5.1.

During the research described in this thesis, no existing set of algorithms was found that can reliably de-compose audio recordings into their components, learn about the components' similarity and their inherent rules of sequence and variation, infer spatial placement of sources or properly process reverberant environments. But as these tools become available, the processing concept of sound textures described earlier will provide the basis for modeling complex natural environments, including those with long attention spans, those with harmonic components and those with many overlapping sources.

8.2 Observations About Sound Modeling

In the context of this thesis, a number of insights were gained that have not been mentioned yet, as they are not related directly to my own implementation or to other researchers' work. At this point, some more general thoughts on sound modeling are provided. The tension between simplicity and comprehensiveness of acoustic models in general is examined, the special case of continuous, non-object sounds is briefly discussed, and the possibility of separating different influences on the spectral characteristics is considered.

8.2.1 Model Generality Versus Model Simplicity

There are be thousands of aspects in any complex mechanical system that contribute to its sound characteristics. Yet, the ambitious goal of sound modeling research is to identify general coding models that only need a small number of synthesis parameters, while still producing highly natural sounds. Herein lies a conflict that will not go away: increasing the expressivity of the model through new parameters will inevitably increase its complexity, while keeping the model simple comes at the cost of reduced realism.

It is tempting to add new features to the model, extensions that fix the particular insufficiencies of individual synthetic sounds: for better piano sounds, an inharmonicity parameters could be added. For some bird species, two independent fundamentals could be used to model the two independent membranes in their bronchi (see Subsection 2.1.11). And for electric guitars, why not extend the model with a distortion parameter?

Although there are reasons to add some of these extensions, there are also good reasons against it. Each new parameter adds complexity to the model and makes it less universal, while the current model is restricted to aspects that all sounds share to a greater or lesser degree: harmonic components and noise. Many of the possible extensions would be relevant only to a small subset of sounds, and would have to be "switched off" for other sounds. The whole advantage of having a parameter space with a fixed number of dimensions would be lost. It is therefore useful to draw a line between universal parametric modeling and highly specialized techniques, such as physical modeling. There is a justification for both approaches, and a decision for either of them should be made with respect to the application domain.

Looking at the specific limitations of the parametric model to capture certain types of acoustic phenomena, it might appear as if a single model could barely accommodate a wide range of sounds. However, the aim of designing versatile parametric models has lots of potential, as the evaluation in Chapter 7 has shown: the parametric sound object synthesis (PSOS) model is able to encode such different acoustic phenomena as bird-tweets, gunshots and instrument sounds, even though it is based on an extreme simplification of the actual sound mechanics. This is possible because the goal is not to create perfectly accurate models of sound, but models that are *good enough* to be accepted as naturalistic sounds. The example of the "screaming hawk" clip in the evaluation illustrates this: although the model entirely lacks the ability to express the characteristic non-linearities of the scream, the much simpler approximation by stochastic noise is widely accepted as realistic by human listeners.

8.2.2 Continuous Sound Sources

The concept of sound objects is at the heart of this thesis, and has been motivated both from a technical point of view and from the side of sound ecology (see Chapter 2). For most sound textures, it is easy to name the objects of which they are composed: rain consists of drops, traffic noise consists of cars driving by, applause consists of hand claps. If a sound object is defined as something that originates from a single source and has a clearly defined start and end, then this concept is broad enough to include almost any acoustic phenomenon.

Still, there are some steady, continuous phenomena that are at odds with the object paradigm. Consider the example of howling wind, or the sound of machinery running constantly in the background. Forcing these sounds into a concept of objects would be highly counter-intuitive, and would probably lead to disturbing effects in the synthesis. Even if separate "machinery" sounds were to be blended smoothly, it would still be difficult to obtain an effect of true continuity or gradual change.

The solution could be found in a variation of the parametric model that is specifically designed to play a single sound indefinitely, without a concept of start and end. In this model, many of the parametric concepts could be kept from the PSOS model, including the representation of the envelope and the separation of harmonic and noise components. Splines could be used to interpolate between keyframes, rather than start and end points. The TAPESTREA software uses the concept of a background "din" to achieve a similar effect, and uses the algorithm by Dubnov et al. (2002) to synthesize it, in addition to the foreground objects.

8.2.3 Conceptual Separation of Excitation Signal and Resonance Characteristics

As mentioned in Subsection 2.1.10 in the context of speech, two different phenomena can be responsible for the shape of a spectral envelope: the mechanics of the vibration of the source signal and the filter resonances of the surrounding structure. In spectral modeling and in the PSOS model, no such distinction is made. A single representation of the spectral shape encodes the combined effects of both phenomena, thus treating the mechanics of the sound object like a black box.

This brings up the question whether an extended model, which would be aware of the two influences, would bring any benefit for either processing efficiency or synthesis quality. In principle, any successful de-coupling of separate phenomena gives more control over the synthesis, especially when the resulting sets of parameters relate directly to physical properties of an object. Also, different degrees of quantization and temporal accuracy could be applied to different phenomena.

The downside of this approach is once more the added complexity of the model, and the difficulty to extract the separation from an input sound automatically. It would also have to be considered that many sound sources do not fit into the model assumption of separated source and filter structures at all. Physical accuracy is not the goal in spectral modeling and related techniques. In fact, the particular strength of these techniques may be that they are physically inaccurate black boxes.

8.3 Future Work

The parametric sound object model presented in this thesis is a starting point for further developments in sound textures, but also in parametric audio coding in general. In this section, some aspects for future research in sound texture analysis and synthesis will be listed.

8.3.1 Advanced Analysis and Automatic Conversion

There is clearly a need for better audio analysis algorithms. Although this goal is certainly not unique to sound texture research, it is a requirement without which further development in this field is held back. Improved analysis means a better understanding of the sounds and their properties, but also, quite literally, the *taking apart* of sound. Unless an algorithm exists that is able to take a sound scenery apart into useful elements, any algorithms further down the processing chain have nothing useful to process. Manual separation — or assembly — of elements can bridge some of these gaps, but it introduces serious limitations in quantity and quality of the separated elements. Better sound separation algorithms for arbitrary mixes of sound are therefore needed.

Vibrato, reverb, modulation and polyphony have all been mentioned as possible extensions to the parametric sound synthesis model. For some of these parameters, algorithms exist to extract them from the input recording. For example, the characteristics of echo and reverb can be measured, and even reversed, using methods of autocorrelation analysis and deconvolution (Torkkola, 1997). Other methods exist to estimate vibrato (Rossignol et al., 1999) and modulation (Roebel, 2006). It would have to be investigated whether any of these techniques is robust enough to integrate them into a sound texture processing framework. The availability of proper analysis methods could be the guiding principle for the addition of a feature to the parametric model: a synthesis feature should only be added if a robust method exists to obtain the relevant parameters from the input automatically.

8.3.2 Adding Effects and Spatial Information to the Model

While the sound element model in its current form is able to approximate the overall spectral and temporal characteristics of an acoustic event, many natural sounds have added details and modifications that define their characteristics. This includes the previously mentioned effects of jitter, shimmer (see Subsection 2.1.7) and modulation, but also distortion. Distortion is rarely encountered in nature, but often present in recoded and digitally processed material. When distortion is applied to a signal, the linear excitation of the amplitude is translated into a non-linear excitation, causing peaks to be clipped or squeezed. Although the mechanism is very simple, it results in strongly altered signal characteristics and changes the amplitudes of the partial peak significantly (Zölzer et al., 2002).

The PSOS model does not deal with the spatial placement of sources and does not add any reverb to the sound. Since the spatial configuration of sources and the reverberation of the environment are not properties of the sound sources themselves, they would have to be handled by a different layer of the processing chain, such as the `Track` layer (see Section 5.2). While parametric representations for spatial placement and reverb are easily defined and their addition to the synthesis procedure is straightforward, the biggest challenge is once more found on the analysis side. An elegant way of processing effects would be to first recognize that a particular effect is present, then reversing the effect and modeling the “clean” signal, in order to re-apply the effect on the synthesis side.

8.3.3 Improving the Model for Granular Sounds

The evaluation of parametrically encoded sound objects has revealed a problem of the model to process “granular” sounds, i.e., sounds with micro transients, crackle and temporal detail (see Section 7.4). This is a direct consequence of the modeling paradigm, which is based on gradually changing characteristics. If, for example, several seconds of rain are treated as just one sound, the individual rain drops are lost. In this case, the solution may be quite simple: the modeling has to be applied to a different scale, turning the rain drops into separate sound objects and capturing the combination of many drops in the form of a statistical distribution pattern.

A more complicated modeling problem can be observed in the “footsteps on gravel” example. There is a strong argument for treating each footstep as an object: the footsteps are arguably the most dominant perceptual entity. On the higher level, distribution patterns could model the sequential regularity of footsteps, depending on the walking speed or step patterns (e.g., walking vs. skipping). This leaves the problem of the grainy structure within the footsteps. Increasing the temporal resolution until every grain is properly modeled is most likely the wrong way. Not only would this increase the storage space and remove every advantage of parametric modeling, it would also incorrectly imply that the occurrence of a grain g at time t is in any way significant for a type of sound. Instead, it seems much more useful to keep just the information that the sound is “grainy”, and store only very few parameters about the nature of the grains. Different methods would have to be tested in order to find useful parametric dimensions for *graininess*.

Whether or not the addition of graininess to the model would be useful is difficult to say. Before attempting an implementation, the question should be answered whether the possible quality improvement of some specific sounds is worth the added complexity, and also whether the estimation of the corresponding parameters could be achieved automatically.

8.3.4 Improving the Model for Start Transients

As mentioned in Subsection 6.6.1, sounds with sharp attacks could benefit from an explicit transient model, which would allow for a better representation of strong and sudden onsets. A parametric representation that integrates well into the overall spline-based parametric model would yet have to be found, but improved transients could likely be implemented using only a handful of additional parameters, as described in Subsection 6.6.1. At the same time, the overall temporal resolution could be drastically lowered, because, apart from transients, sounds like gunshots or plucked guitar strings behave very smoothly.

However, while there may be some justification for specifically modeling the start transient of sounds, this does not apply to arbitrary transients within a sound, or to sounds with several strong transients. New parameters for arbitrary transients cannot just be added at runtime, and sounds with two or more transients would require different, incompatible models. In fact, even the extension for one transient may turn out to be overly specific, as many sounds do not have a start transient. Further experiments are necessary to determine whether the addition of transient parameters is beneficial, and what types of parameters are required to model transient shapes.

8.3.5 Modeling of Phase Information

If relative phase alignment were to become a part of a parametric model — which could be useful, as described in Subsection 2.2.10 — the question would arise what kinds of parameters would be useful for encoding this information. A simple list, containing the phase offsets of all partials, would not be practical in this context, mainly because sounds have different numbers of partials, but also because such an encoding would be excessive and possibly redundant. A better parametric model of phase should be linked to the mechanical principle behind the observed phases and describe patterns of regularity, instead of simply listing all observations. The relationship between relative phases of partials and the mechanical properties of the sound-producing system has not been investigated in this thesis.

8.4 Final Thoughts

At the beginning of the research for this thesis, I did not consider an object-based model to be a good representation for sound textures. Starting from the observation that the block- and grain-based methods had problematic limitations, I did not want to introduce yet another type of "block". The idea in my mind was that of a much more fluent, more abstract and more mathematical model: an algorithm that would take the input sound data and transform it into a statistical entity, some sort of feature space that could express conditional probabilities across multiple dimensions. The acoustic properties would be completely obscure when looking at the model, but would emerge automatically upon synthesis. Some aspect of this is contained in the wavelet-based approach by Bar-Joseph et al. (1999) (see Subsection 4.4.4), and in the FeatSynth framework by Hoffman and Cook (2007) (see Subsection 4.4.5). But as intriguing as this highly abstract concept is, it is based on the — maybe irrational — hope that statistical analysis can indirectly and automatically solve all the known problems —

sound separation, object identification, pattern discovery — that are so difficult to solve even when they are targeted directly.

The research on sound texture analysis and synthesis integrates some of the most difficult, but also most interesting problems of acoustics and signal analysis. Given that the group of textural sounds, of rain, traffic noise or bird chirps, is often neglected in audio processing, it seems surprising that there should be any unique challenges hidden in this topic. If audio compression algorithms are built into every telephone and every flat-screen television, what significant challenges could possibly be left? The big challenge, as it turns out, lies not in *transmitting*, but in *understanding*.

The success of transmitting something can me measured quite easily by comparing the inputs to the outputs. If they are the same, or if they sound *as if* they were the same, the transmission can be considered successful. This is precisely where sound texture synthesis is different: if the outputs sound identical to the inputs, the processing has failed! In order to work well as a natural acoustic scenery, a synthesized sound texture has to sound similar, but always different. The first insight is that there has to be some random influence to the process. This notion has inspired synthesis techniques based on the random concatenation of grains, blocks or sub-clips (see Subsections 4.4.1 and 4.4.2). But randomization is only a small ingredient of what needs to be achieved. If we compare the synthesis of a sound texture to the creation of visual art, we would now have an artist who has learned to make new images from collages of other paintings, but still has not learned to draw or paint in any way. This is where completely new challenges in audio processing will be found.

For decades, there have been two different worlds in sound processing: sampled sound for natural sounds and parametric models for electronic music and experimental sounds. Creating convincing, natural sounds from entirely artificial, mathematical models has traditionally been very difficult. This is beginning to change, as physical models are already being used to synthesize a variety of instruments, allowing for fine-grained control over detailed aspects of playing styles. Parametric sound models could one day replace sampled sound in many synthesis applications, allowing for a more storage-space efficient, more elegant and more versatile representation. This thesis has shown that this way is viable, not only for instruments, but also for a wide range of natural phenomena. It will require some improvements for the synthesis quality of parametric sounds to reach the natural quality of recorded sounds. But given more time and combined research effort, parametric synthesis holds great promises for future applications.

References

Ahmed, M., & Burgess, K. (2009). Young music fans deaf to iPod's limitations. *The Times, London, March 5th.*

Ahmed, N., Natarajan, T., & Rao, K. (1974). Discrete cosine transfom. *Computers, IEEE Transactions on*, *100*(1), 90–93.

Andersen, T., & Jensen, K. (2001). Phase modeling of instrument sounds based on psycho acoustic experiments. In *Proceedings of the Workshop on Current Research Directions in Computer Music* (pp. 174–186). Barcelona, Spain.

Andersen, T., & Jensen, K. (2004). Importance and representation of phase in the sinusoidal model. *Journal of the Audio Engineering Society*, *52*(11), 1157–1169.

Asano, F., Ikeda, S., Ogawa, M., Asoh, H., & Kitawaki, N. (2003). Combined approach of array processing and independent component analysis for blind separation of acoustic signals. *IEEE Transactions on Speech and Audio Processing*, *11*(3), 204–215.

Atal, B. (1974). Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *The Journal of the Acoustical Society of America*, *55*, 1304-1312.

Atal, B., & Hanauer, S. (1971). Speech analysis and synthesis by linear prediction of the speech wave. *Journal of the Acoustical Society of America*, *50*(2), 637–655.

Athineos, M., & Ellis, D. (2003). Sound texture modelling with linear prediction in both time and frequency domains. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-03)* (Vol. 5, pp. 648–651). Hong Kong.

Balaban, E. (1988). Bird song syntax: learned intraspecific variation is meaningful. *Proceedings of the National Academy of Sciences of the United States of America*, *85*(10), 3657-3660.

Bar-Joseph, Z., El-Yaniv, R., Lischinski, D., & Werman, M. (2002). Texture mixing and texture movie synthesis using statistical learning. *Transactions on Visualization and Computer Graphics*, *7*(2), 120–135.

Bar-Joseph, Z., El-Yaniv, R., Lischinski, D., Werman, M., & Dubnov, S. (1999). Granular synthesis of sound textures using statistical learning. In *Proceedings of the International Computer Music Conference* (pp. 178–181). Beijing, China.

Barrington, L., Chan, A., & Lanckriet, G. (2009). Dynamic texture models of music. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 1589–1592). Taipei, Taiwan.

Bartkowiak, M., & Żernicki, T. (2007). Improved partial tracking technique for sinusoidal modeling of speech and audio. *Poznan University of Technology Academic Journals: Electrical Engineering.*

Blanz, V., & Vetter, T. (1999). A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on computer graphics and interactive techniques* (pp. 187–194). Los Angeles, CA, USA.

Bogner, A. (2005). *Das Experteninterview: Theorie, Methode, Anwendung.* Wiesbaden, Germany: VS Verlag.

Bracewell, R. (1989). The Fourier Transform. *Scientific American*, *260*(6), 86–95.

Brandenburg, K. (1999). MP3 and AAC explained. In *Proceedings of the AES 17th International Conference on High-Quality Audio Coding.* Florence, Italy.

Brandenburg, K., & Bosi, M. (1997). Overview of MPEG audio: Current and future standards for low-bit-rate audio coding. *The Journal of the Audio Engineering Society*, *45*(1), 4–21.

Bregman, A. (1990). *Auditory Scene Analysis: The Perceptual Organization of Sound.* Cambridge, Massachusetts: MIT Press.

Bregman, A. (1998). Psychological data and computational ASA. In D. Rosenthal & H. Okuno (Eds.), *Computational Auditory Scene Analysis* (pp. 1–11). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Bronstein, I., Semendjajew, K., Musiol, G., & Mühlig, H. (2001). *Taschenbuch der Mathematik* (fifth ed.). Thun und Frankfurt am Main: Verlag Harry Deutsch.

Brookes, D., & Loke, H. (1999). Modelling energy flow in the vocal tract with applications to glottal closure and opening detection. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'99)* (Vol. 1, pp. 213–216). Phoenix, Arizona.

Buchanan, B., Givon, M., & Goldman, A. (1987). Measurement of discrimination ability in taste tests: An empirical investigation. *Journal of Marketing Research*, *24*(2), 154–163.

Caetano, M., & Rodet, X. (2009). Evolutionary Spectral Envelope Morphing by Spectral Shape Descriptors. In *Proceedings of the International Computer Music Conference (ICMC'09)* (pp. 171–174). Montreal, Quebec.

Carlsson, S. E. (2010). *FilmSound.org: Walla.* on-line article available at `http://www.filmsound.org/terminology/walla.htm` (last visited: December 2, 2010).

Carr, P. (1999). *English Phonetics and Phonology: An Introduction.* Hoboken, NJ: Wiley-Blackwell.

Chinen, M., & Osaka, N. (2007). Genesynth: Noise band-based genetic algorithm analysis/synthesis framework. In *Proceedings of the International Computer Music Conference (ICMC 2007).* Copenhagen, Denmark.

Chowning, J. (1980). Computer synthesis of the singing voice. *Sound Generation in Winds, Strings, Computers*, *29*, 4–13.

Cohen, M., Shade, J., Hiller, S., & Deussen, O. (2003). Wang tiles for image and texture generation. *ACM Transactions on Graphics*, *22*(3), 287–294.

Cooley, J., & Tukey, J. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, *19*(90), 297–301.

Cover, T., & Thomas, J. (1991). Rate Distortion Theory. In *Elements of Information Theory* (2nd ed., pp. 301–346). Hoboken, New Jersey: Wiley & Sons, Inc.

Cox, M. (1971). An algorithm for approximating convex functions by means by first degree splines. *The Computer Journal*, *14*(3), 272.

Cross, G., & Jain, A. (1983). Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*(1), 25–39.

Day, S., & Altman, D. (2000). Blinding in clinical trials and other studies. *British Medical Journal (BMJ)*, *321*(7259), 504.

de Boor, C. (1978). *A Practical Guide to Splines.* New York: Springer.

De Bonet, J. (1997). Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (p. 368). Los Angeles, CA, USA.

Dempster, A., Laird, N., Rubin, D., et al. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*(1), 1–38.

De Poli, G. (1983). A tutorial on digital sound synthesis techniques. *Computer Music Journal*, *7*(4), 8–26.

Di Scipio, A. (1999). Synthesis of environmental sound textures by iterated nonlinear functions. In *Proceedings of the 2nd COST G-6 Workshop on Digital Audio Effects (DAFx-99)* (pp. 109–117). Trondheim.

Di Federico, R. (1998). Waveform preserving time stretching and pitch shifting for sinusoidal models of sound. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx-98).* Barcelona, Spain.

Dolson, M. (1986). The phase vocoder: A tutorial. *Computer Music Journal*, *10*(4), 14–27.

Dubnov, S. (2002). Extracting sound objects by independent subspace analysis (paper #252). In *Proceedings of 22nd International Conference of Audio Engineering Society* (Vol. 17). Espoo, Finland.

Dubnov, S., Assayag, G., & Cont, A. (2007). Audio oracle: A new algorithm for fast learning of audio structures. In *Proceedings of International Computer Music Conference (ICMC'07)* (pp. 224–228). Copenhagen, Denmark.

Dubnov, S., Bar-Joseph, Z., El-Yaniv, R., Lischinski, D., & Werman, M. (2002). Synthesizing sound textures through wavelet tree learning. *IEEE Computer Graphics and Applications*, *22*(4), 38–48.

Dutoit, T., Pagel, V., Pierret, N., Bataille, F., & Van Der Vrecken, O. (2002). The MBROLA project: Towards a set of high quality speech synthesizers free of use for non commercial purposes. In *Proceedings of the Fourth International Conference on Spoken Language (ICSLP 96)* (Vol. 3, pp. 1393–1396). Philadelphia, PA, USA.

Efros, A., & Freeman, W. (2001). Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2001)* (pp. 341–346). Los Angeles, CA, USA.

Ellis, D., & Rosenthal, D. (1995). Mid-level representations for computational auditory scene analysis. In *Proceedings of the Computational Auditory Scene Analysis Workshop; 1995 International Joint Conference on Artificial intelligence.* Montreal, Canada.

Ernst, P. (2002). *Pragmalinguistik: Grundlagen, Anwendungen, Probleme.* Berlin, Germany: Walter De Gruyter Inc.

Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining* (Vol. 96, pp. 226–231). Portland, Oregon, USA.

Evangelista, G. (1993, 12). Pitch synchronous wavelet representations of speech and music signals. *IEEE Transactions on Signal Processing, Special Issue on Wavelets and Signal Processing*, *41*(12), pp. 3313–3330.

Ezzat, T., Meyers, E., Glass, J., & Poggio, T. (2005). Morphing Spectral Envelopes Using Audio Flow. In *Proceedings of the 9th European Conference on Speech Communication and Technology (INTERSPEECH 2005).* Lisboa, Portugal.

Fay, R., & Wilber, L. (1989). Hearing in vertebrates: a psychophysics databook. *The Journal of the Acoustical Society of America*, *86*, 2044.

Fee, M., Shraiman, B., Pesaran, B., & Mitra, P. (1998). The role of nonlinear dynamics of the syrinx in the vocalizations of a songbird. *Nature*, *395*(6697), 67–71.

Fitch, W., Neubauer, J., & Herzel, H. (2002). Calls out of chaos: the adaptive significance of nonlinear phenomena in mammalian vocal production. *Animal Behaviour*, *63*(3), 407–418.

Fitz, K., Haken, L., Lefvert, S., & O'Donnell, M. (2002). Sound Morphing using Loris and the Reassigned Bandwdith-Enhanced Additive Sound Model: Practice and Applications. In *Proceedings of the International Computer Music Conference (ICMC'02).*

Fletcher, N. (1988). Bird song – a quantitative acoustic model. *Journal of Theoretical Biology*, *135*(4), 455–481.

Fletcher, N. (2000). Inharmonicity, nonlinearity, and music. *The Physicist*, *37*(5), 171–175.

Foley, J. (1995). *Computer Graphics: Principles and Practice.* Addison-Wesley Professional.

Ganchev, T., Fakotakis, N., & Kokkinakis, G. (2005, 10). Comparative evaluation of various MFCC implementations on the speaker verification task. In *Speech and Computer, 10th International Conference (SPECOM 2005)* (Vol. 1, pp. 191–194). Patras, Greece.

Golub, G., & Reinsch, C. (1970). Singular value decomposition and least squares solutions. *Numerische Mathematik*, *14*(5), 403–420.

Golub, G., & Van Loan, C. (1996). *Matrix Computations.* Johns Hopkins University Press.

Goto, M. (2000). A robust predominant-F0 estimation method for real-time detection of melody and bass lines in CD recordings. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-99)* (Vol. 2).

Grahn, J., & Brett, M. (2007). Rhythm and beat perception in motor areas of the brain. *Journal of Cognitive Neuroscience*, *19*(5), 893–906.

Green, R., & Pierre, J. (2002). Preliminary results in B-spline enhanced source-tracking. In *Proceedings of the Ninth IEEE Workshop on Statistical Signal and Array Processing* (pp. 260–263).

Greuter, S., Parker, J., Stewart, N., & Leach, G. (2003). Undiscovered worlds – towards a framework for real-time procedural world generation. In *Fifth International Digital Arts and Culture Conference.* Melbourne, Australia.

Grey, J. (1975). *An exploration of musical timbre.* Unpublished doctoral dissertation, Center for Computer Research in Music and Acoustics, Department of Music, Stanford University.

Grey, J. (1977). Multidimensional perceptual scaling of musical timbres. *Journal of the Acoustical Society of America*, *61*(5), 1270–1277.

Gribonval, R., & Bacry, E. (2003). Harmonic decomposition of audio signals with matching pursuit. *IEEE Transactions on Signal Processing*, *51*(1), 101–111.

Han, C., Risser, E., Ramamoorthi, R., & Grinspun, E. (2008). Multiscale texture synthesis. *ACM Transactions on Graphics*, *27*(3), 51.

Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83–85.

Haus, G., & Vercellesi, G. (2005). State of the art and new results in direct manipulation of mpeg audio codes. *Proceedings of Sound and Music Computing*.

Heeger, D., & Bergen, J. (1995). Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (p. 238).

Helmholtz, H. L. von. (1913). *Die Lehre von den Tonempfindungen: Als Physiologische Grundlage für die Theorie der Musik* (6th ed.). Braunschweig, Germany: Vieweg.

Hewish, A. (1970). Pulsars. *Annual Review of Astronomy and Astrophysics*, *8*, 265.

Hoffman, M., & Cook, P. (2007). The featsynth framework for feature-based synthesis: Design and applications. In *International computer music conference* (Vol. 2, pp. 184–187).

Horner, A., Beauchamp, J., & Haken, L. (1993). Machine tongues XVI: Genetic algorithms and their application to FM matching synthesis. *Computer Music Journal*, *17*(4), 17–29.

Hoskinson, R. (2002). *Manipulation and Resynthesis of Environmental Sounds with Natural Wavelet Grains.* Unpublished doctoral dissertation, The University of British Columbia.

Hughes, C., Smith, E., Stapleton, C., & Hughes, D. (2004). Augmenting museum experiences with mixed reality. In *Proceedings of the 2004 International Conference on Knowledge Sharing and Collaborative Engineering (KSCE)* (pp. 22–24).

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks*, *13*(4-5), 411–430.

International Phonetic Association. (1999). *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet.* Cambridge, UK: Cambridge University Press.

International Telecommunications Union. (2003, 1). *Method for the subjective assessment of intermediate quality levels of coding systems (ITU-R BS.1534-1).*

Jackson, B. (2008, March). Assassin's Creed: 21st-century game audio for the world of the crusades (interview with Mathieu Jeanson). *Mix*. (online article, available at `http://mixonline.com/game_audio/features/audio_assassins_creed` , last visited: December 16, 2010)

Jang, G., & Lee, T. (2003). A maximum likelihood approach to single-channel source separation. *The Journal of Machine Learning Research*, *4*, 1365–1392.

Jutten, C., & Herault, J. (1991). Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture. *Signal Processing*, *24*(1), 1–10.

Kabal, P., & Ramachandran, R. (1986, 12). The Computation of line spectral frequencies using Chebyshev polynomials. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *34*(6), 1419–1426.

Kahrs, M., & Avanzini, F. (2001). Computer synthesis of bird songs and calls. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx-01)* (pp. 23–27). Limerick, Ireland.

Kaiser, C. (2009). *Homerecording.* mitp-Verlag. Available from `http://books.google.de/books?id=3FdMmEw8cWAC`

Ketten, D. (2000). Cetacean ears. *Springer Handbook of Auditory Research*, *12*, 43–108.

Klapuri, A. (2003). Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Transactions on Speech and Audio Processing*, *11*(6), 804–816.

Klapuri, A. (2006). Multiple fundamental frequency estimation by summing harmonic amplitudes. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)* (pp. 216–221). Victoria, Canada.

Komorowski, M. (2009). *A History of Storage Cost.* on-line article, available at `http://www.mkomo.com/cost-per-gigabyte` (last visited: December 2, 2010).

la Cour, P. (1903). *Tidens Naturlære.* Copenhagen: Gyldendalske Boghandels Forlag.

Laroche, J., Stylianou, Y., Moulines, E., & Paris, T. (1993). HNS: Speech modification based on a harmonic+noise model. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-93)* (Vol. 2, pp. 550–553). Minneapolis, Minnesota.

Levine, S. (1998). *Audio representations for data compression and compressed domain processing.* Unpublished doctoral dissertation, Stanford University, Stanford, CA.

Licklider, J. (1957). Effects of changes in the phase pattern upon the sound of a 16-harmonic tone. *The Journal of the Acoustical Society of America*, *29*, 780.

Logan, B. (2000). Mel frequency cepstral coefficients for music modeling. In *International Symposium on Music Information Retrieval (MUSIC IR 2000)* (Vol. 28). Plymouth, Massachusetts.

London, J. (2002). Cognitive constraints on metric systems: some observations and hypotheses. *Music Perception*, *19*(4), 529–550.

Loscos, A., & Bonada, J. (2004). Emulating rough and growl voice in spectral domain. In *Proceedings of the 7th International Conference on Digital Audio Effects (DAFX-04)* (pp. 49–52). Naples, Italy.

Lu, L., Wenyin, L., & Zhang, H. (2004). Audio textures: Theory and applications. *IEEE Transactions on Speech and Audio Processing*, *12*(2), 156–167.

Luce, R. (1993). *Sound and Hearing: A Conceptual Introduction.* Hillsdale, New Jersey: Lawrence Erlbaum Associates.

MacQueen, J., et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, p. 14). California, USA.

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE*, *63*(4), 561–580.

Mallat, S. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*(7), 674–693.

Marco, D., McLachlan, N., & Wilson, S. (2007). The perception of simultaneous pitches in ambiguous sounds. *Proceedings of ICoMCS December*, 91–94.

Massie, D. (2002). Wavetable sampling synthesis. *Applications of Digital Signal Processing to Audio and Acoustics*, *437*, 311–341.

McAdams, S., Winsberg, S., Donnadieu, S., Soete, G., & Krimphoff, J. (1995). Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological Research*, *58*(3), 177–192.

McAulay, R., & Quatieri, T. (1986). Speech Analysis/Synthesis Based on a Sinusoidal Representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *34*(4), 744–754.

McGill, R., Tukey, J., & Larsen, W. (1978). Variations of box plots. *American Statistician*, *32*(1), 12–16.

Meddis, R., & O'Mard, L. (1998). Psychophysically faithful methods for extracting pitch. In D. Rosenthal & H. Okuno (Eds.), *Computational Auditory Scene Analysis* (pp. 43–58). Mahway, New Jersey: Lawrence Erlbaum Associates.

MIDI Manufacturers Association. (2010). General MIDI Level 1 Sound Set [Computer software manual]. (available online at `http://www.midi.org/techspecs/gm1sound.php` , last visited: December 6, 2010)

Misra, A., Cook, P. R., & Wang, G. (2006). A new paradign for sound design. In *Proceedings of the International Conference on Digital Audio Effects (DAFx-06)* (pp. 319–324). Montreal, Quebec, Canada.

Misra, A., Wang, G., & Cook, P. (2009). TAPESTREA: a new way to design sound. *Proceedings of the seventeen ACM international conference on Multimedia*, 933–936.

Möhlmann, D., & Herzog, O. (2010a, 5). High-level sound coding with parametric blocks (paper no. 8096). In *Proceedings of the 128th Convention of the Audio Engineering Society.* London, UK.

Möhlmann, D., & Herzog, O. (2010b, 8). Multiple fundamental frequency estimation using machine learning and frequency-scaled feature vectors (paper no. 160). In S. Demorest, S. Morrison, & P. Campbell (Eds.), *Proceedings of the 11th International Conference on Music Perception and Cognition (ICMPC).* Seattle, Washington, USA.

Möhlmann, D., Herzog, O., & Wagner, T. (2009, 7). Music analysis and spectral modeling based on cubic B-Splines. In G. Scavone, V. Verfaille, & A. da Silva (Eds.), *Proceedings of the international computer music conference* (pp. 327–330). Montreal, Canada: The Schulich School of Music, McGill University, Montreal, QC, Canada.

Moore, B. (2004). *An Introduction to the Psychology of Hearing* (5th ed.). London, UK: Elsevier Academic Press.

Moorer, J. (1978). The use of the phase vocoder in computer music applications. *Journal of the Audio Engineering Society*, *26*(1/2), 42–45.

Mullen, J., Howard, D., & Murphy, D. (2004). Acoustical simulations of the human vocal tract using the 1D and 2D digital waveguide software model. In *Proceedings of the 7th International Conference on Digital Audio Effects (DAFX-04)* (pp. 311–314). Naples, Italy.

Nigjeh, B., Trivailo, P., & McLachlan, N. (2002, 11). Application of modal analysis to musical bell design. *Acoustics 2002 – Innovation in Acoustics and Vibration, Annual Conference of the Australian Acoustical Society.*

Ohm, G. (1843). Über die Definition des Tones, nebst daran geknüpfter Theorie der Sirene und ähnlicher tonbildender Vorrichtungen. *Annalen der Physik und Chemie*, *59*, 513–565.

Omori, K., Kojima, H., Kakani, R., Slavit, D., & Blaugrund, S. (1997). Acoustic characteristics of rough voice: subharmonics. *Journal of Voice*, *11*(1), 40–47.

Orlikoff, R., & Kahane, J. (1991). Influence of mean sound pressure level on jitter and shimmer measures. *Journal of Voice*, *5*(2), 113–119.

Oudeyer, P. (2003). The production and recognition of emotions in speech: features and algorithms. *International Journal of Human-Computer Studies*, *59*(1-2), 157–183.

Paine, G. (2004). Reeds: A responsive sound installation. In *Proceedings of the 2004 International Conference of Auditory Display (ICAD).* Sydney, Australia.

Painter, T., & Spanias, A. (2002). Perceptual coding of digital audio. *Proceedings of the IEEE*, *88*(4), 451–515.

Paliwal, K. (1995). Interpolation properties of linear prediction parametric representations. In *Proceedings of the Fourth European Conference on Speech Communication and Technology (EUROSPEECH)* (pp. 1029–1032). Madrid, Spain.

Paliwal, K., & Alsteris, L. (2003). Usefulness of phase spectrum in human speech perception. In *Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH-2003)* (pp. 2117–2120). Geneva, Switzerland.

Paradiso, J., Abler, C., Hsiao, K., & Reynolds, M. (1997). The magic carpet: physical sensing for immersive environments. In *CHI'97 Extended Abstracts on Human Factors in Computing Systems: Looking to the Future* (pp. 277–278). Ft. Lauderdale, FL.

Parish, Y., & Müller, P. (2001). Procedural modeling of cities. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2001)* (pp. 301–308). Los Angeles, CA.

Parker, J. R., & Behm, B. (2004). Creating Audio Textures by Samples: Tiling and Stretching. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-04)* (Vol. 4, pp. 317–320). Montreal, Quebec.

Paul, L. (2010). *GameSoundCon 2010: Procedural Sound Design.* presentation available at `http://www.videogameaudio.com` (last visited: December 2, 2010).

Perlin, K. (1985). An image synthesizer. *ACM SIGGRAPH Computer Graphics*, *19*(3), 287–296.

Phillips, G. (1968). Algorithms for piecewise straight line approximations. *The Computer Journal*, *11*(2), 211–212.

Póczos, B., & Lőrincz, A. (2005). Independent subspace analysis using k-nearest neighborhood distances. *Artificial Neural Networks: Formal Models and Their Applications*, 163–168.

Poliner, G., & Ellis, D. (2005). A classification approach to melody transcription. In *Proceedings of the 2005 International Conference on Music Information Retrieval* (pp. 161–166). London, UK.

Polotti, P., & Evangelista, G. (2000). Harmonic-band wavelet coefficient modeling for pseudo-periodic sound processing. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFx-00)* (pp. 103–108). Verona, Italy.

Polotti, P., Menzer, F., & Evangelista, G. (2002). Inharmonic sound spectral modeling by means of fractal additive synthesis. In *Proceedings of the 10th International Conference on Digital Audio Effects (DAFx-02)* (pp. 127–132).

Pols, L., Kamp, L. Van der, & Plomp, R. (1969). Perceptual and physical space of vowel sounds. *Journal of the Acoustical Society of America*, *46*, 458–467.

Prusinkiewicz, P., & Lindenmayer, A. (1990). *The Algorithmic Beauty of Plants (The Virtual Laboratory).* Berlin, Germany: Springer.

Rabiner, L. (1967). *Speech synthesis by rule: An acoustic domain approach.* Unpublished doctoral dissertation, Massachusetts Institute of Technology, Cambridge, MA.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Reid, G. (2001). *Synthesizing Percussion.* article from Sound On Sound magazine, available at `http://www.soundonsound.com/sos/nov01/articles/synthsecrets1101.asp` (last visited: December 2, 2010).

Reid, G. (2003). *Synthesizing Pan Pipes.* article from Sound On Sound magazine, available at `http://www.soundonsound.com/sos/aug03/articles/synthsecrets.htm` (last visited: December 2, 2010).

Reynolds, D. (1994). Experimental evaluation of features for robust speaker identification. *IEEE Transactions on Speech and Audio Processing*, *2*(4), 639–643.

Reynolds, D., Zissman, M., Quatieri, T., O'Leary, G., & Carlson, B. (1995). The effects of telephone transmission degradations on speaker recognition performance. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-95)* (pp. 329–332). Detroit, Michigan.

Roads, C. (1988). Introduction to granular synthesis. *Computer Music Journal*, *12*(2), 11–13.

Rodet, X., & Depalle, P. (1992, 10). Spectral envelopes and inverse FFT synthesis (paper no. 3393). In *Proceedings of the 93rd Convention of the Audio Engineering Society.*

Roebel, A. (2006). Estimation of partial parameters for non stationary sinusoids. In *Proceedings of the 2006 International Computer Music Conference (ICMC'06)* (pp. 167–170). New Orleans, Louisiana.

Rosenthal, D., & Okuno, H. (Eds.). (1998). *Computational Auditory Scene Analysis.* Mahway, New Jersey: Lawrence Erlbaum Associates.

Rossignol, S., Depalle, P., Soumagne, J., Rodet, X., & Collette, J. (1999). Vibrato: detection, estimation, extraction, modification. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx-99)* (Vol. 99). Trondheim, Norway.

Rossing, T., Yoo, J., & Morrison, A. (2004). Acoustics of percussion instruments: an update. *Acoustical Science and Technology*, *25*(6), 406–412.

Saint-Arnaud, N. (1995). *Classification of Sound Textures.* Unpublished master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Saint-Arnaud, N., & Popat, K. (1997). Analysis and synthesis of sound textures. In D. Rosenthal & H. Okuno (Eds.), *Readings in Computational Auditory Scene Analysis: Proceedings of the IJCAI-95 Workshop* (pp. 293–308).

Schölkopf, B., Smola, A., & Müller, K. (1997, 10). Kernel principal component analysis. In *Artificial Neural Networks, 7th International Conference (ICANN'97)* (pp. 583–588). Lausanne, Switzerland: Springer.

Serra, X. (1989). *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition.* Unpublished doctoral dissertation, CCRMA, Department of Music, Stanford University.

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

Shaw, I. (1999). *Qualitative Evaluation.* London, UK: Sage Publications Ltd.

Sinclair, S., Scavone, G., & Wanderley, M. (2009, 7). Audio-haptic interaction with the digital waveguide bowed string. In G. Scavone, V. Verfaille, & A. da Silva (Eds.), *Proceedings of the International Computer Music Conference* (pp. 327–330). Montreal, Canada: The Schulich School of Music, McGill University.

Slaney, M. (1993). Auditory Toolbox. *Apple Computer Company: Apple Technical Report No. 45*.

Slaney, M., Covell, M., & Lassiter, B. (1996). Automatic audio morphing. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-96)* (pp. 1001–1004). Atlanta, Georgia.

Stern, Z. (2008). *An Interview with Spore's infectious designers.* Mac|Life, February 2008, on-line article available at `http://www.mkomo.com/cost-per-gigabyte` (last visited: December 2, 2010).

Stevens, S. (1946). On the theory of scales of measurement. *Science*, *103*(2684), 677–680.

Stoll, G., & Kozamernik, F. (2000, June). *EBU Listening Tests on Internet Audio Codecs (Tech 3296)* (Tech. Rep.). Grand-Saconnex, Geneva, Switzerland: European Broadcasting Union.

Strawn, J. (1980). Approximation and syntactic analysis of amplitude and frequency functions for digital sound synthesis. *Computer Music Journal*, *4*(3), 3–24.

Strobl, G. (2007). *Parametric Sound Texture Generator.* Unpublished master's thesis, Universität für Musik und Darstellende Kunst, Graz, Austria.

Strobl, G., Eckel, G., Rocchesso, D., & le Grazie, S. (2006). Sound texture modeling: A survey. In *Proceedings of the 2006 Sound and Music Computing (SMC) International Conference* (pp. 61–65). Marseille, France.

Svensson, E. (2000). Comparison of the quality of assessments using continuous and discrete ordinal rating scales. *Biometrical Journal*, *42*(4), 417–434.

Svensson, E. (2001). Guidelines to statistical evaluation of data from rating scales and questionnaires. *Journal of Rehabilitation Medicine*, *33*(1), 47–48.

Tancerel, L., Ragot, S., Ruoppila, V., & Lefebvre, R. (2002). Combined speech and audio coding by discrimination. In *Proceedings of the IEEE Workshop on Speech Coding* (pp. 154–156). Delavan, WI.

Terasawa, H. (2009). *A Hybrid Model for Timbre Perception: Quantitative Representations of Sound Color and Density.* Unpublished doctoral dissertation, Stanford University, Stanford, California.

Terasawa, H., Slaney, M., & Berger, J. (2005). Perceptual distance in timbre space. In *Proceedings of the International Conference on Auditory Display (ICAD'05)* (pp. 61–68). Limerick, Ireland.

Tokuda, I., Riede, T., Neubauer, J., Owren, M., & Herzel, H. (2002). Nonlinear analysis of irregular animal vocalizations. *The Journal of the Acoustical Society of America*, *111*, 2908–2919.

Torkkola, K. (1997). Blind deconvolution, information maximization and recursive filters. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-97)* (Vol. 4, pp. 3301–3304). Munich, Germany.

Traunmüller, H. (1990). Analytical expressions for the tonotopic sensory scale. *The Journal of the Acoustical Society of America*, *88*, 97–100.

Turk, G. (1991). Generating textures on arbitrary surfaces using reaction-diffusion. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH-91)* (pp. 289–298). Las Vegas, Nevada.

Turner, C. (2003). Recursive discrete-time sinusoidal oscillators. *Signal Processing Magazine, IEEE*, *20*(3), 103–111.

Valbret, H., Moulines, E., & Tubach, J. (1992). Voice transformation using PSOLA technique. *Speech Communication*, *11*(2-3), 175–187.

Van Veen, B., & Buckley, K. (1988). Beamforming: A versatile approach to spatial filtering. *IEEE ASSP Magazine*, *5*(2), 4–24.

Vercoe, B. L., Gardner, W. G., & Scheirer, E. D. (1998). Structured audio: Creation, transmission, and renderig of parametric sound representations. *Proceedings of the IEEE*, *86*(5), 922–940.

Verma, T., Levine, S., & Meng, T. (1997). Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals. In *Proceedings of the International Computer Music Conference (ICMC'97)* (pp. 164–167). Thessaloniki, Greece.

Verma, T., & Meng, T. (1998). An analysis/synthesis tool for transient signals that allows a flexible sines+transients+noise model for audio. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-98)* (Vol. 6, pp. 3573–3576). Seattle, WA.

Vincent, E., Jafari, M., & Plumbley, M. (2006). Preliminary guidelines for subjective evaluation of audio source separation algorithms. *Department of Electronic Engineering, Queen Mary, Univeristy of London.*

Virtanen, T. (2006). *Sound source separation in monaural music signals.* Unpublished doctoral dissertation, Tampere University of Technology, Tampere, Finland.

Wakita, H. (1973). Direct estimation of the vocal tract shape by inverse filtering of acoustic speech waveforms. *IEEE Transactions on Audio and Electroacoustics*, *21*(5).

Wasserman, L. (2006). *All of Nonparametric Statistics.* New York, NY: Springer.

Wei, L., & Levoy, M. (2000). Fast texture synthesis using tree-structured vector quantization. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2000)* (pp. 479–488). New Orleans, Louisiana.

Wiskott, L., & Sejnowski, T. (2002). Slow feature analysis: unsupervised learning of invariances. *Neural Computation*, *14*(4), 715–770.

Xiph.org Foundation. (2010, 2). Vorbis I Specification [Computer software manual]. (available online at `http://www.xiph.org/vorbis/doc/Vorbis_I_spec.pdf` , last visited: December 6, 2010)

Xu, J., & Yang, E. (2006). Rate-distortion Optimization for MP3 Audio Coding with Complete Decoder Compatibility. In *Proceedings of the IEEE 7th Workshop on Multimedia Signal Processing* (pp. 1–4).

Yang, D., Kyriakakis, C., & Kuo, C. (2005). *High-Fidelity Multichannel Audio Coding.* New York, NY: Hindawi Publishing Corporation.

Yegnanarayana, B., & Murthy, P. (2002). Enhancement of reverberant speech using LP residual signal. *IEEE Transactions on Speech and Audio Processing*, *8*(3), 267–281.

Yeh, C., Roebel, A., & Chang, W. (2008). Multiple-F0 estimation for MIREX 2008. *The 4th Music Information Retrieval Evaluation eXchange (MIREX08).* (available online at `http://mediatheque.ircam.fr/articles/textes/Yeh08b`, last visited: December 15, 2010)

Zhu, S., Wu, Y., & Mumford, D. (1998). Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, *27*(2), 107–126.

Zhu, X., & Wyse, L. (2004). Sound texture modeling and time-frequency LPC. In *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx'04)* (Vol. 4, pp. 345–349). Naples, Italy.

Zölzer, U., Amatriain, X., Arfib, D., Bonada, J., De Poli, G., Dutilleux, P., et al. (2002). *DAFX: Digital Audio Effects* (U. Zölzer, Ed.). West Sussex, England:

John Wiley & Sons.

Zwicker, E. (1961). Subdivision of the audible frequency range into critical bands (Frequenzgruppen). *Journal of the Acoustical Society of America*, *33*, 248.

Appendix A

Sound Texture Definitions

Over the past years, the term “sound texture” has been defined and characterized in several ways by different authors. Some of them have pointed out the difficulty of finding a solid and universal definition. The table below provides an overview of relevant statements that have been made in this respect.

publication	definition(s)
(Saint-Arnaud & Popat, 1997)	“Defining *sound texture* is no easy task. Most people will agree that the noise of a fan is a likely ’sound texture.’ Some other people would say that a fan is too bland, that it is only a noise. The sound of rain, or of a crowd are perhaps better textures. But few will say that one voice makes a texture.” (p. 293) “The first constraint we put on our definition of a sound textures is that it should exhibit similar characteristics over time; that is, a two-second snippet of a texture should not differ significantly from another two-second snippet.” (p. 293) “[...] it can have local structure and randomness, but the characteristics of the structure and randomness must remain constant on the large scale.” (p. 294) “A sound texture is characterized by its sustain.” (p. 294) “We call attention span the maximum time between events before they become distinct. A few seconds is a reasonable value for the attention span. We therefore put a second time constraint on sound textures: high-level characteristics must be exposed or exemplified (in the case of stochastic distributions) within the attention span of a few seconds.” (p. 298) “High level randomness is also acceptable, as long as there are enough occurrences within the attention span to make a good example of the random properties.” (p. 299) “Speech or music can provide new information at any time, and their ‘potential information content’ is shown here as a continuously increasing function of time. Textures, on the other hand, have constant long term characteristics, which translates into a flattening of the potential information increase. Noise (in the auditory cognitive sense) has somewhat less information than textures.” (p. 294)

(Di Scipio, 1999)	“[...] emergent properties in the output sound signal result into acoustic turbulences and other textural sound phenomena” (p. 109) “This paper discusses the use of iterated nonlinear functions in the modelling of the perceptual attributes in complex auditory images. Based on the chaotic dynamics in such algorithms, it is possible to create textural and environmental sound effects of a peculiar kind, hardly obtained with other methods.” (p. 109) “This research opens to new experiments in electroacoustic music and the creation of synthetic, but credible, auditory scenes in multimedia applications and virtual reality.” (p. 109)
(Bar-Joseph et al., 1999)	“Granular synthesis is one of the most appealing models for sound texture synthesis” (p. 178) “Testing the same idea without checking the predecessors (only checking the ancestors), can still produce good results if the original input is completely ‘textural’, i.e. a sound that contains noise and percussion components only.” (p. 181)
(Dubnov et al., 2002)	“Natural and artificial sounds such as rain, waterfall, fire, traffic noises, people babble, machine noises and etc., can be regarded as such textures.” (p. 38) “We can describe sound textures as a set of repeating structural elements (sound grains) subject to some randomness in their time appearance and relative ordering but preserving certain essential temporal coherence and across-scale localization.” (p. 38) “[...] we can assume that the sound signals are approximately stationary at some scale. (p. 38) “[...] treating the input sound texture as a sample of a stochastic process, [...]” (p. 38) “[the proposed technique] results in new sound textures that closely resemble the original sound sources sonic impression without exactly repeating it.” (p. 38)
(Athineos & Ellis, 2003)	“sound textures [...] are distinct from speech and music” (p. 648) “Although a rigorous definition is elusive, [...]” (p. 648) “[sound] textures should have an indeterminate extent (duration) with consistent properties (at some level), and be readily identifiable from a small sample.” (p. 648) “Many of the sounds we have collected as textures are noisy (i.e. without strong, stable periodic components) and rough (i.e. amplitude modulated in the 20-200 Hz range)” (p. 648)
(Parker & Behm, 2004)	“A sound texture can be described as having a somewhat random character, but a recognizable quality. Any small sample of a sound texture should sound very much like, but not identical to, any other small sample. The dominant frequency should not change, nor should any rhythm or timbre.” (p. 317)

(Lu et al., 2004)	“[. . .] we can only store the short audio clip, and then generate a long audio stream of any length in the user end.” (p. 156) “Such sounds are relatively monotonic, simple in structure, and characterize repeated yet possibly variable sound patterns.” (p. 156) “[a sound texture] exhibits repeated or similar patterns, just like image textures and video textures.” (p. 156)
(X. Zhu & Wyse, 2004)	“The common character of this class of sounds is that they have a background din and a foreground transient sequence.” (p. 345) “Sound textures are sounds for which there exists a window length such that the statistics of the features measured within the window are stable with different window positions. That is, they are static at long enough time scales.” (p. 345) “Using this definition, at some window length any signal is a texture, so the concept is of vale only if the texture window is short enough to provide practical efficiencies for representation.” (p. 345) “The texture window length is signal-dependent, but typically on the order of 1 second. If the window needs to be longer in order to produce stable statistics when time shifted, then the sound would be unlikely to be perceived as a static texture.” (p. 345)
(Misra et al., 2006)	“A sound texture can be described as a sound with structural elements that repeat over time, but with some randomness.” (p. 319)
(Strobl et al., 2006)	“Sound textures are an important class of sounds in interactive applications, video games, virtual reality and webbased applications, movie sound effects, or in extensive tracks of art installations.” (p. 61) “Like in image processing there is no universally valid definition of a sound texture.” (p. 61) “In the context of this paper we would like to adhere to the initial definition from Saint-Arnaud et al. [. . .]” (p. 61) “We specially want to emphasize that repetitions should not be audible and sound textures should be targeted of sounding perceptually meaningful, in the sense that the synthesized texture is perceptually comparable to the example clip.” (p. 64) “In the ideal case, no difference should be noticeable, i.e. the generated sounds still sound natural and contain no artefacts.” (p. 64)

Table A.1: Definitions of the term “sound texture”, collected from publications between the years 1997 and 2006. Less formal statements about sound textures are also included.

Appendix B

Spectrograms of Sounds in the Evaluation

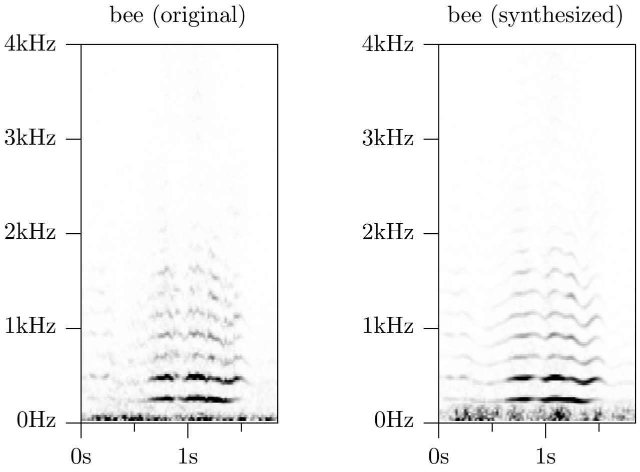


Figure B.1: Spectrograms of the original and synthetic "bee" sound. The synthetic clip is converted directly from the original.

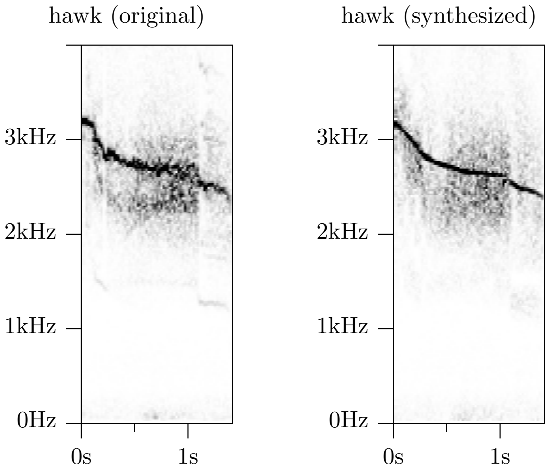


Figure B.2: Spectrograms of the original and synthetic "hawk" sound. The synthetic clip is converted directly from the original.

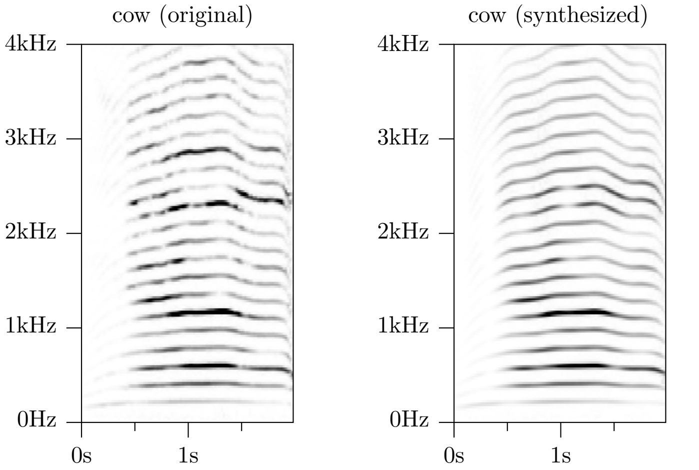


Figure B.3: Spectrograms of the original and synthetic "cow" sound. The synthetic clip is converted directly from the original.

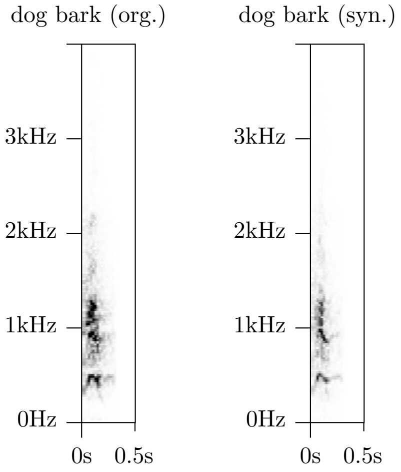


Figure B.4: Spectrograms of the original and synthetic "dog bark" sound. The synthetic clip is converted directly from the original.

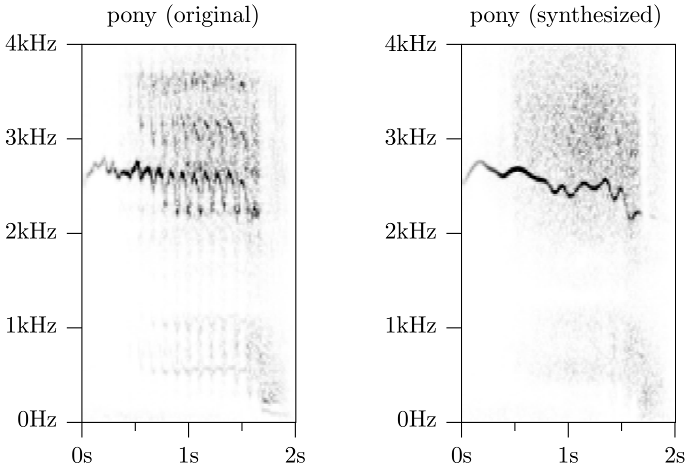


Figure B.5: Spectrograms of the original and synthetic "pony" sound. The synthetic clip is converted directly from the original.

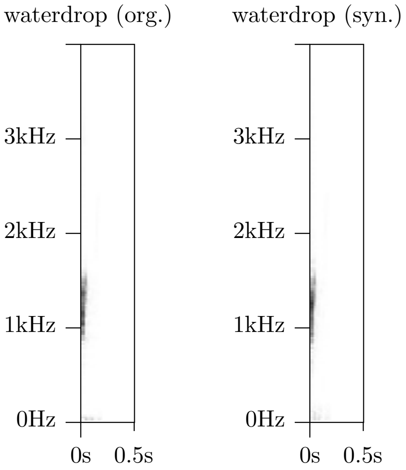


Figure B.6: Spectrograms of the original and synthetic "waterdrop" sound. The synthetic clip is converted directly from the original.

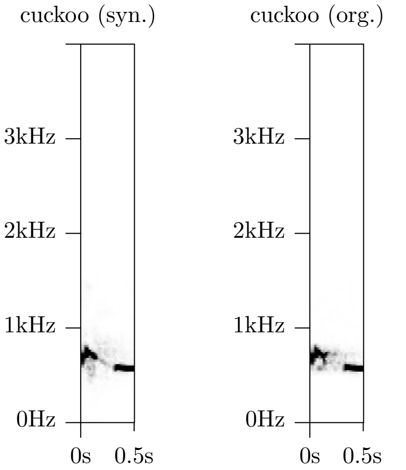


Figure B.7: Spectrograms of the original and synthetic "cuckoo" sound. The synthetic clip is converted directly from the original.

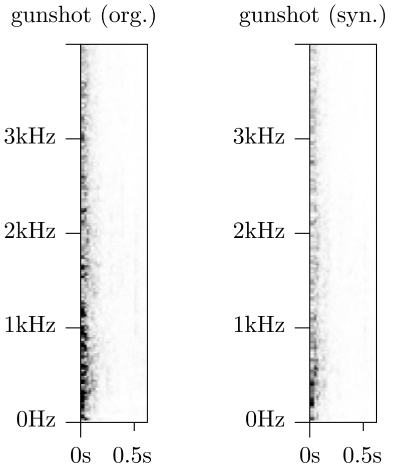


Figure B.8: Spectrograms of the original and synthetic "gunshot" sound. The synthetic clip is converted directly from the original.

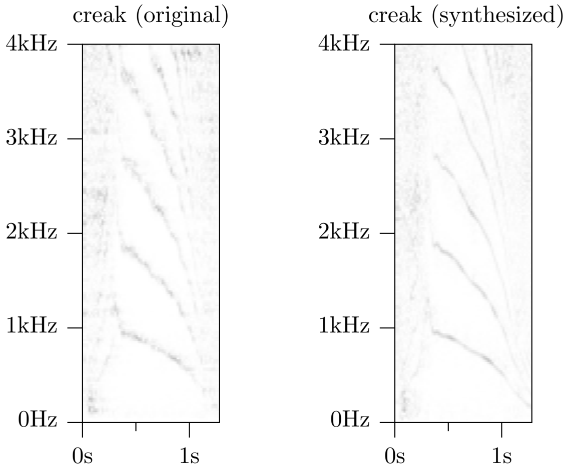


Figure B.9: Spectrograms of the original and synthetic "creak" sound. The synthetic clip is converted directly from the original.

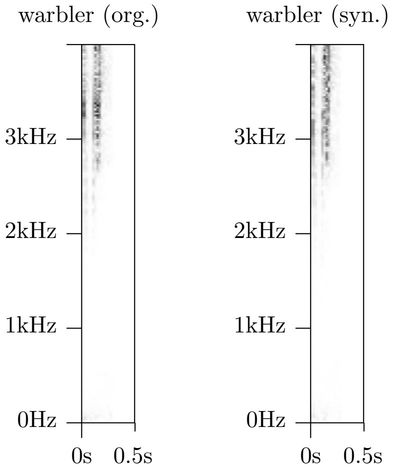


Figure B.10: Spectrograms of the original and synthetic "warbler" sound. The synthetic clip is converted directly from the original.

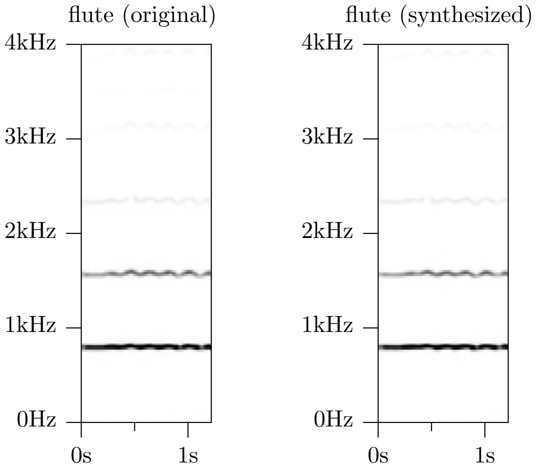


Figure B.11: Spectrograms of the original and synthetic "flute" sound. The synthetic clip is converted directly from the original.

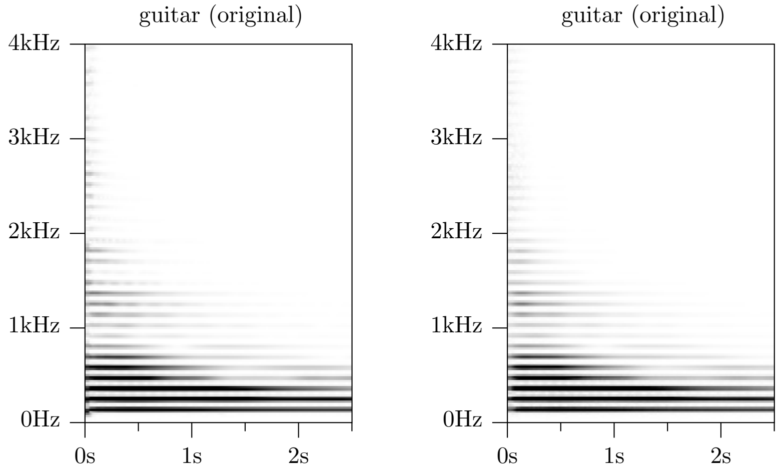


Figure B.12: Spectrograms of the original and synthetic "guitar" sound. The synthetic clip is morphed from two different originals.

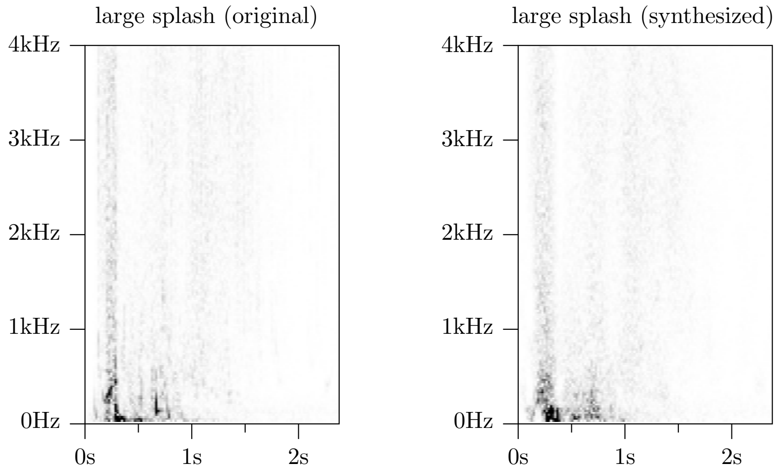


Figure B.13: Spectrograms of the original and synthetic "large splash" sound. The synthetic clip is morphed from two different originals.

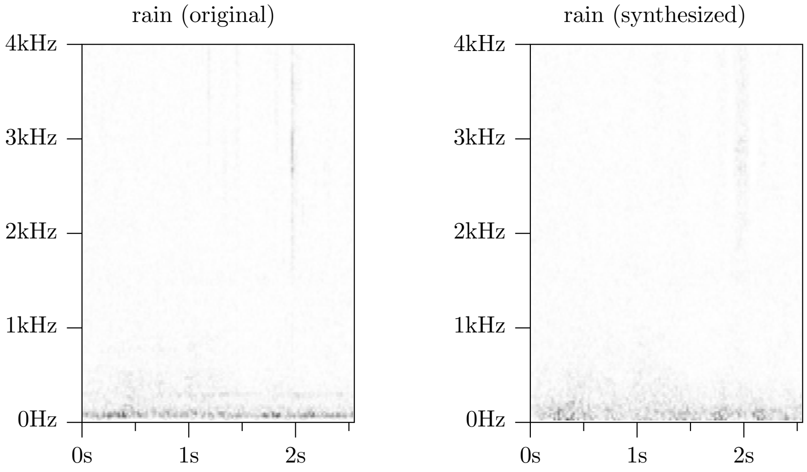


Figure B.14: Spectrograms of the original and synthetic "rain" sound. The synthetic clip is morphed from two different originals.

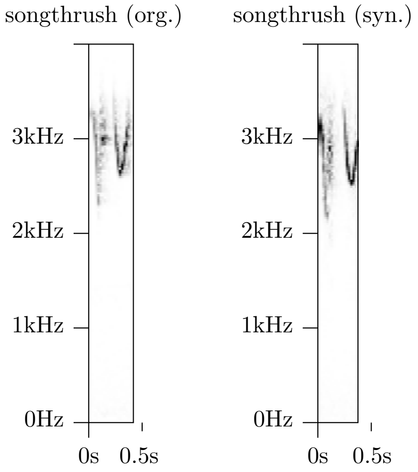


Figure B.15: Spectrograms of the original and synthetic "songthrush" sound. The synthetic clip is morphed from two different originals.

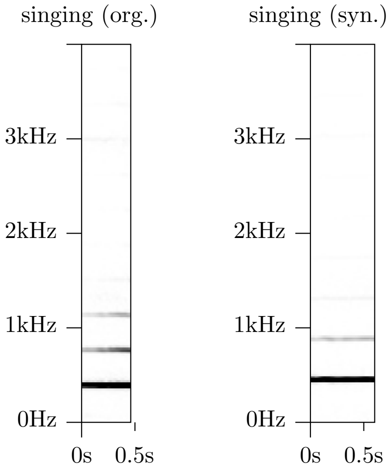


Figure B.16: Spectrograms of the original and synthetic "singing" sound. The synthetic clip is morphed from two different originals.

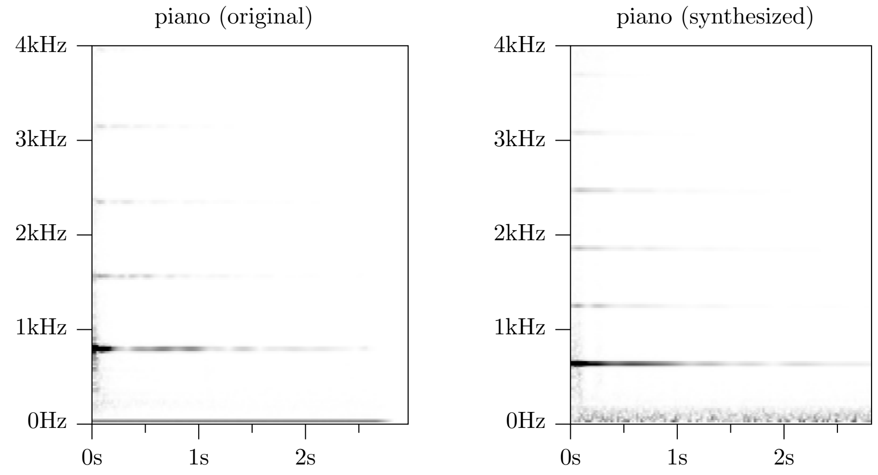


Figure B.17: Spectrograms of the original and synthetic "piano" sound. The synthetic clip is morphed from two different originals.

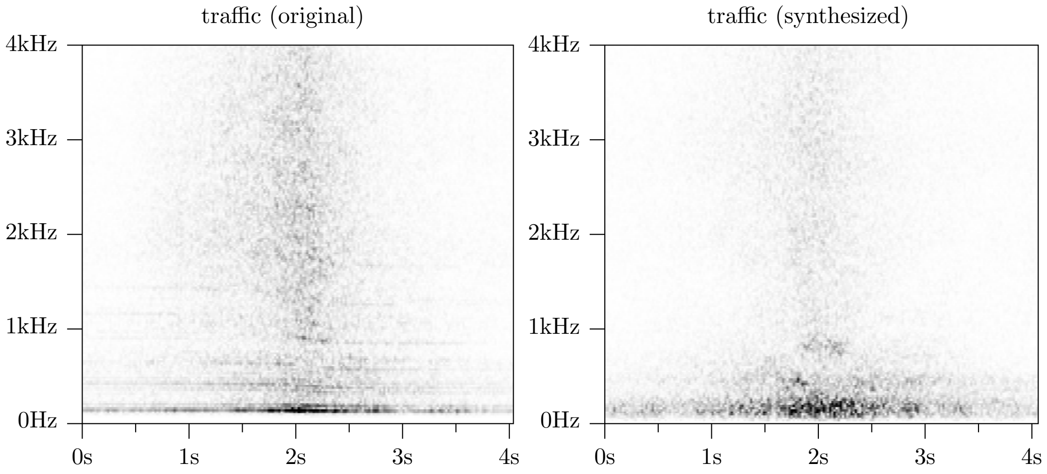


Figure B.18: Spectrograms of the original and synthetic "traffic" sound. The synthetic clip is morphed from two different originals.

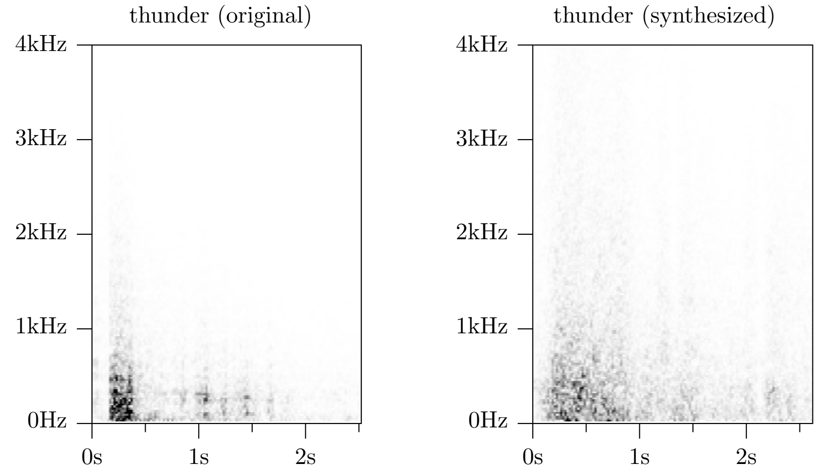


Figure B.19: Spectrograms of the original and synthetic "thunder" sound. The synthetic clip is morphed from two different originals.

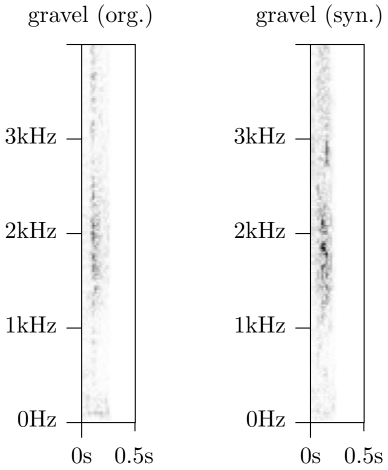


Figure B.20: Spectrograms of the original and synthetic “gravel” sound. The synthetic clip is morphed from two different originals.

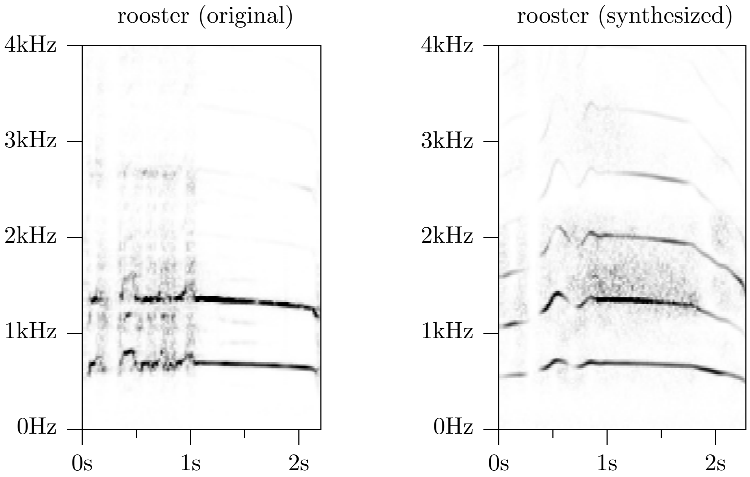


Figure B.21: Spectrograms of the original and synthetic “rooster” sound. The synthetic clip is morphed from two different originals.

Appendix C

PCA-Based Dimensionality Reduction for Sound Elements

To conduct preliminary tests of the principal components approach, a dimensionality reduction method based on simple PCA was implemented. For each sound instance, all parameters of the model are unwrapped and aligned into an n-dimensional vector. For every dimension, the mean value is computed and subtracted from the values, so that each parameter is centered around the origin. An $n \times n$ covariance matrix is then computed as a preparation for the PCA. Using the `gsl_eigen_symmv` function of the GNU Scientific Library[1], Eigenvectors and Eigenvalues of the covariance matrix are calculated. The library uses the QR reduction method (Golub & Van Loan, 1996) for the computation. The results are guaranteed to be precise down to an ϵ value, which is limited only by the machine precision. The decomposition gives the axes of the PCA space, but does not contain any information about what range of values in this space is used. To determine the upper and lower bounds that are acceptable for each dimension, all instances of the sound are transformed into the new PCA space by matrix multiplication. For every dimension in PCA space, the lowest and highest occurring values are noted. The procedure is illustrated in Fig. C.1.

For generating novel sounds from the PCA space, a value is selected from the allowed value range in each dimension, and a uniform probability distribution is assumed within the range, for matters of simplicity. The selection of these low-dimensional control parameters can be random-based (for an application that generates textures), or can be set manually by a human operator using a graphical user interface, if experimentation with the sound is the goal. In this graphical user interface, sliders can be used to control the principal components. The first sliders contain the most relevant, i.e., the principal components of the feature space, so that by using only three sliders, much of the feature space can be covered. The configuration of sliders yields a vector in PCA space, which can be transformed back into the model parameter space by using an inverse matrix multiplication. The set of model parameters can be used to synthesize an output sound. Preliminary results indicate that this method produces sounds that are typical for the given class of input sounds, but are different from each individual input sound. However, no formal evaluation of these results was attempted so far.

[1] `http://www.gnu.org/software/gsl/manual` (last visited: December 1, 2010)

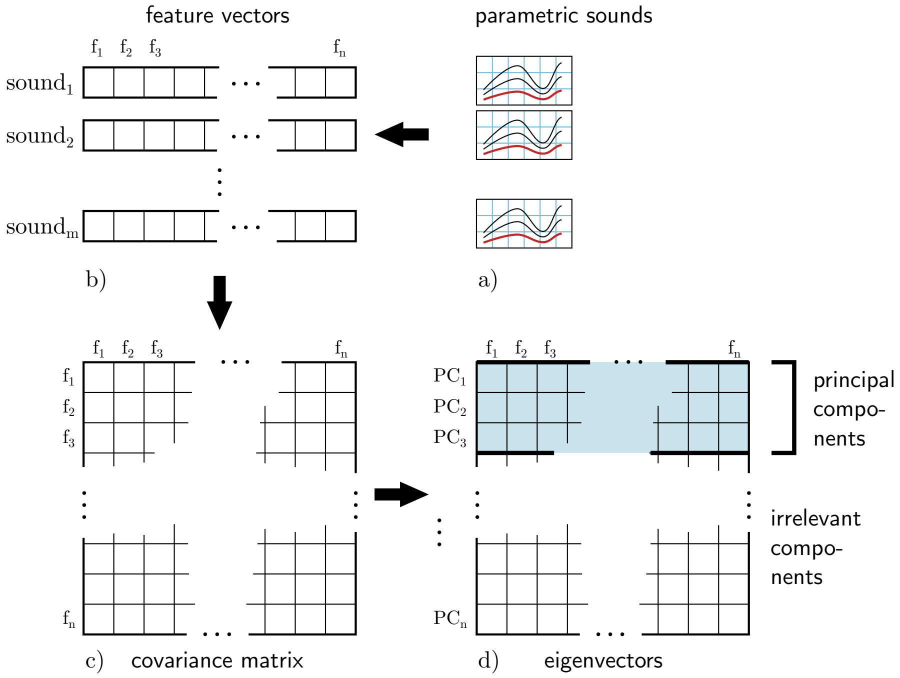


Figure C.1: Illustration of the principal component analysis (PCA): parametrized sounds (a) are arranged in feature vectors (b). The mean of each feature is subtracted to center the data around the origin, and a covariance matrix is computed (c). The first eigenvectors are the principal components of the sounds (d). Eigenvalues are discarded.

Index