TOWARD SYNTHESIZED ENVIRONMENTS: A SURVEY OF ANALYSIS AND SYNTHESIS METHODS FOR SOUND DESIGNERS AND COMPOSERS

Ananya Misra, Perry R. Cook[†]

Princeton University
Department of Computer Science ([†]also Music), Princeton, USA
{amisra, prc}@cs.princeton.edu

ABSTRACT

We present an overview of digital audio analysis and synthesis methods for sound design and composition. The sonic landscape available to us contains a multitude of sounds, ranging from artificial to natural, purely musical to purely "real-world." To take full advantage of this diversity, it is helpful to have a comprehensive knowledge of the tools with which we can create and manipulate different types of sounds. We offer a summary of existing techniques organized by their underlying technology and source material, as well as by the kinds of sounds for which they are known to be effective. Our survey aims to support the synthesis of rich sound scenes and environments by facilitating the selection of the most appropriate tools for each component sound. A full toolbox means the whole world need not look like a nail!

1. INTRODUCTION

The sonic landscape available to us contains a multitude of sounds, ranging from artificial to natural, purely musical to purely "real-world." Taking full advantage of this diversity may require mixing and matching different types of sounds in the same piece. Environmental audio or sound scenes, for example, tend to combine many types of foreground and background sound sources: ambient textures, noisy events, pitched events, speech, music, and more. Often, different source types are best suited to different parametric analysis and synthesis paradigms. Parametric synthesis also offers many advantages over mixing raw digital recordings, including improved flexibility, control and compression. Sound designers and composers working with rich sound scenes can therefore benefit from a full knowledge of a variety of analysis and synthesis techniques. To this end, we present a survey of existing digital audio synthesis and analysis methods, including a taxonomy based on which types of sounds they support best.

Related literature includes surveys and taxonomies of digital synthesis techniques [76, 84], computer music [64], sound design for games and film [36], sound and music computing [96], structured audio representations [93] and singing voice synthesis [19]. Many of these touch on common synthesis topics. In [76] (and later [84]), digital synthesis techniques are arranged into four categories: processed (and sampled) recordings, spectral models, physical models and abstract algorithms. [93] discusses analysis and synthesis techniques related to parametric sound representations and describes their domain and range as well as their generality. In [96], topics are divided into sound, music and interaction, with a discussion of analysis and synthesis techniques for each. However, none of these surveys approaches the field from the perspective of creating complex environmental scenes or compositions.

We present a slightly different categorization than [76], considering the technical background and required source material for each method, as well as how it might contribute to a larger sound scene. Of course, no taxonomy is clean or 1-to-1 from method to class. We first discuss abstract synthesis algorithms (section 2), followed by techniques for sound synthesis from "scratch" based on physical or perceptual models (section 3). In our largest section, perhaps most relevant to environmental audio, we examine methods to synthesize sound from existing sound, usually involving analysis as well as synthesis (section 4). Next, we take a brief glance at standalone analysis methods not necessarily designed for synthesis (section 5). We also offer a table linking the described methods to the sound types for which they are known to be effective. Lastly, we present a sound example that combines many synthesis algorithms, highlighting the applicability of multiple techniques to a single coherent piece. We conclude with brief notes to sound designers, developers of interactive applications for entertainment (games, *etc*.), and electro-acoustic composers.

2. ABSTRACT SYNTHESIS ALGORITHMS

Early electronic music with analog synthesizers used oscillator units and modular synthesis to produce sounds [76]. In the digital world, oscillators range from basic sine, triangle, square and other simple wave generators, to more complex and hybrid systems [78, 25]. Because oscillators in some form play a role in many aspects of synthesis, we

may think of them as building blocks of synthesis. We classify them as abstract because they do not necessarily arise from one particular "concrete" paradigm. Abstract synthesis algorithms building on oscillators include amplitude and frequency modulation and waveshaping. Frequency modulation, or the modulation of one oscillator's frequency (or phase) by another's output, was discovered by Chowning to have interesting musical implications [14, 15]. More generally, waveshaping refers to the modification of an existing signal by a non-linear function [93, 33]. Other work on waveshaping includes [52] and [3].

The study of non-linearity and chaos has led to a range of abstract sound synthesis algorithms [85, 26, 55]. An overview of related work and an introduction to the use of circle maps as rich non-linear oscillators are presented in [33, 34]. The idea of "errant sound synthesis", through non-linear oscillators and breakpoint sets, is discussed in [16], where the goal is described as "not the modelling and reproduction of sounds from perceptual or physical acoustical data but the potential of any algorithm, cast into the audio range." Auditory display and sonification, or the mapping of other data to sound, also enable a form of abstract sound synthesis [49, 98]. Software tools for sonification include [8] and [62]. Another abstract approach to sound synthesis is to consider synthesis techniques as structured, abstract entities, facilitating a high-level mathematical understanding of synthesis and transformation methods [32, 24].

3. SYNTHESIS FROM SCRATCH

Synthesis from "scratch" refers to the replication of real-world sounds using physical or perceptual models, without the raw material of existing audio samples. The model may represent a sound's physical source or environment, or the perceptual characteristics desired. A frequent advantage of such model-based synthesis is the option of high-level parametric control over the synthesized sound. Early examples of physical modeling include musical sound synthesis by Hiller and Ruiz [42], and the Karplus-Strong plucked-string algorithm [46, 45]. The plucked-string algorithm originated as an abstract variation of wavetable synthesis, but was later discovered to implement a simplified physical model of a plucked string. This idea of interesting but computationally simple physical models led to digital waveguide synthesis, which can produce a range of musical instrument sounds with expressive control [74, 75, 77, 78, 39]. Extensions to waveguide synthesis include two- and three-dimensional waveguide meshes [91, 92], and banded waveguides to model stiff systems such as struck bars [35].

Other physical models have been used to reproduce acoustic instrument sounds, including reed and bow-string models [73], speech and singing voice synthesis [17, 19], and modal synthesis of percussive sounds [18, 20]. The Synthesis ToolKit [23] provides a library for real-time synthesis of musical instrument sounds using physical models. Many of these have been ported and expanded upon in other systems such as PeRColate (for Max/MSP) [87], ChucK [95], SuperCollider [1], and others. Non-linearity is also important for the physical modeling of some musical instrument sounds [63]. The synthesis of real-world contact and motion sounds such as impact and friction, especially for animations and interactive applications, provides another arena for physical modeling [83, 90, 60, 68].

Perceptual models for synthesis from scratch can include spectral information, such as formants for speech or singing. Formant synthesizers generate the desired formants using second-order resonant filters [19]. While their parameters can be extracted from recorded speech, such an analysis stage is not essential to the algorithm. Formant wave functions (FOFs) are time-domain waveform representations of a formant's impulse response, usually modeled by a sinusoidal oscillator with time-varying amplitude; these are flexibly generated and added to create a voice-like sound [69, 19]. While formant-based synthesis methods produce sound containing the desired formants, the idea of synthesizing audio to match any set of target features is generalized in feature-based synthesis [43]. This method inputs a set of arbitrary acoustic and perceptual feature values and uses a parametric optimizer to generate matching audio via arbitrarily selected synthesis algorithms. It is currently suited to generating abstract sound and audio caricatures, but because the sound produced depends on the features and specific synthesis techniques used, any inherent limitation of the method is still to be discovered.

4. SYNTHESIS FROM EXISTING SOUNDS

A third category of synthesis is the creation of sound from existing sound. As this often involves some form of analysis of the existing sound, we can also broadly refer to it as synthesis-by-analysis. In particular, we look at concatenative techniques, in which existing samples are rearranged in the time-domain, and additive synthesis, which generally takes a more spectral approach.

4.1. Concatenative Techniques

A well known concatenative technique is wavetable synthesis [76, 93], the periodic repetition of a set of time-domain samples, often to create pitched instrument sounds. Wavetable synthesis may also arguably fit into either of the previous categories; an abstraction of it resulted in the plucked-string algorithm [46] (see section 3), while in frequency-domain wavetable synthesis, a specified harmonic spectrum is transformed to yield the time-domain samples [76]. However, the algorithm is concatenative in the general sense of concatenating existing samples in time. Schwarz [70, 71] describes concatenative synthesis as synthesis using a large

database of source sounds segmented into *units*, a *target* sound to be synthesized, and a *unit selection* algorithm that chooses the units best matching the target according to a set of *unit descriptors*. The selected units are transformed as needed and concatenated in the time-domain, possibly with a cross-fade. A range of techniques with varying levels of manual control and automation fall under this umbrella, including methods for speech and singing synthesis [19] and audio mosaicing [51, 70].

Another type of concatenative synthesis is granular synthesis [86], in which sound is created by concatenating usually short "sound grains". This can produce a variety of abstract sounds and textures. FOFs (see section 3) can also be interpreted as granular synthesis, especially when the sinusoidal oscillators are replaced by arbitrary samples [93]. Dictionary-based methods with time-localized waveforms provide an analytical counterpart to granular synthesis, and allow flexible analysis, re-synthesis and transformation of general audio signals [82].

Granular and other forms of concatenative synthesis have also supported the generation of soundscapes and sound textures of arbitrary length. Hoskinson and Pai [44] applied a wavelet-based algorithm to split an existing soundscape into syllable-like segments, which were then selected by similarity and concatenated to produce an ongoing stream. Birchfield *et al.* [9] describe a model for real-time soundscape generation using a database of annotated sound files, dynamically selected and combined to produce a varying soundscape. Fröjd and Horner [40] concatenate longer segments of existing audio, with some overlap, to achieve rapid sound texture synthesis.

4.2. Additive Synthesis

Additive synthesis generally involves spectral analysis, and addition of the signals synthesized based on this analysis. Risset [67] is credited with the first such additive synthesis for music, to analyze and re-synthesize trumpet tones [76]. Other groundwork includes the development of voice coders, or vocoders, originally for audio transmission and compression. The channel vocoder [29] passes input audio through a bank of bandpass filters to compute the energy present in each frequency band. This information is used to synthesize a signal with the corresponding energy in each band, by summing the weighted outputs of a bank of synthesis filters [22]. The phase vocoder [27] uses the Fast Fourier Transform (FFT) to estimate the phase as well as magnitude of each frequency band (or bin), resulting in more convincing reconstruction for general sounds [22]. Both types of vocoders have been used for pitch and time transformations of the source sound, and for cross-synthesis.

As the phase vocoder uses the information in all the (usually numerous) frequency bins to reconstruct the sound via an inverse FFT, it does not achieve significiant compression. McAulay and Quatieri [56, 66] found that speech signals can be modeled well using only a few sinusoids instead of all the frequencies present in the FFT. Spectral modeling synthesis [72] adds filtered noise to this mix, recognizing that the pitched or deterministic elements of a sound are best modeled by sinusoids, while components such as breath noise better suit a stochastic model. Other works introduce transients (brief, bursty events) to achieve a finer decomposition, and also consider transforms besides the FFT [94, 53, 7, 65]. Additive synthesis has typically been used to synthesize speech and instrument sounds, although it has also been appropriated for more general environmental sounds [59]. Tools for additive synthesis include Lemur/Loris [38], the CLAM library[2], AudioSculpt [10] and SPEAR [48].

4.3. Subtractive Synthesis and Other Techniques

Other ways to analyze existing sound for synthesis include subtractive synthesis and Linear Predictive Coding (LPC) [4, 54]. Originally designed for speech coding and synthesis, LPC analyzes a sound into a source-filter model such that a linear combination of the latest samples in a sequence predicts the next sample. The prediction error or any other signal can then be fed into the filter as a source, allowing cross-synthesis and pitch transformations as well as re-synthesis of the original sound. LPC has been explored for musical composition [79, 50] and sound texture synthesis [5, 99].

Sound texture synthesis, or generating an arbitrary quantity of a given source texture, has also attracted other approaches. Dubnov *et al.* [28] employ wavelet-tree learning to synthesize sound textures that are structurally similar to the original, with stochastic variations. Wavelets have also been investigated for modeling and transforming stochastic components of sounds in a parameterized way [58]. Other statistical approaches include modeling short events in the source sound, and synthesizing and mixing these according to an inferred statistical distribution [99]. A survey of sound texture modeling methods is available in [81].

The modeling and detection of transients for sines+transients+noise frameworks also involve analysis for the purpose of synthesis. Levine and Smith [53] detect transients by examining the short-time energy envelopes of both the original signal and the residual (noise) signal. Verma and Meng [94] model transients as the time-domain dual of sinusoidal tracks; they detect transients by comparing the energy in short and long segments of a time-domain signal. Transients can also be found via other onset detection methods, including both time- and frequency-domain analysis [6, 80].

5. ANALYSIS NOT FOR SYNTHESIS

Many sound analysis methods are not necessarily designed with an eye toward synthesis, although the information they yield can lead to synthesis using techniques such as feature-

based (section 3) or concatenative (section 4) synthesis. Widmer *et al.* [96] distinguish between the analysis of music versus sound. Topics to explore for sound include perceptually informed acoustic models, sound source recognition and classification, and content-based search and retrieval. Issues for music concern understanding music at multiple levels and from multiple disciplines. They also mention the growing use of semantic data to understand sound and music. We focus here on content-based methods; although we merge sound and music, we chiefly examine techniques that include a level of signal analysis.

One goal of audio analysis is to represent an audio signal in structurally or perceptually meaningful ways [93]. The separation of a signal into its component sources facilitates such representation. For environmental audio, source separation falls under computational auditory scene analysis [11], or estimating the sources in a composite sound scene to better understand human perception [31, 30] or enable structured representation [57]. Perceptually based and spectral techniques, such as grouping partials by harmonics, modulation, common onset and proximity, are often used to identify independent sources [31, 30, 57]. Source separation also plays a role in the automatic transcription of concurrent musical sounds, in the form of multiple fundamental frequency estimation [47]. Also related are blind source separation techniques, in which the sources are estimated via purely computational rather than perceptually motivated analysis [12, 37]. Besides source separation, analysis for representation also includes other aspects of music transcription [41] and audio compression.

Another set of analysis methods aims to understand and use a collection of sounds on a more global level. These techniques entail extracting and analyzing information from many sounds and using the results for content-based classification, search, recommendation, or other actions involving comparison. Work in this area includes methods to extract audio features and to analyze them over the training set and compute distances for the classification or search. The MARSYAS framework [88] offers tools for performing many of these steps, and has led to foundational work on automatic genre classification [89]. Also related is research on automatic timbre recognition [61]. An overview of content-based music information retrieval, both at the signal-level and the collection-level, is presented in [13].

6. DISCUSSION

We have briefly surveyed methods for audio analysis and synthesis, with a focus on techniques combining both. The organization of methods into categories based on their underlying technology, source information, and goals implies a taxonomy according to both the theory and implementation of these methods. While our taxonomy includes some concrete aspects such as source material and intended usage, it also retains a level of abstraction. This is possible because the theoretical background of a method dictates, to some extent, the types of sounds for which it is effective.

Sound/Goal	**Methods**
Abstract	FM, non-linear oscillators, feature-based synthesis, wavetables, concatenative / granular synthesis
Acoustic instruments	Wavetables, waveguides / physical models, concatenative / granular synthesis, additive synthesis
Contact sounds	Physical models
Cross-synthesis	LPC, vocoders
Pitch / time transformations	LPC, vocoders, additive synthesis, concatenative / granular synthesis
Pitched sounds	Additive synthesis, concatenative / granular synthesis, FM synthesis, oscillators
Singing voice	FM synthesis, formant synthesis, FOFs, concatenative / granular synthesis, additive synthesis
Speech	Formant synthesis, FOFs, concatenative / granular synthesis, vocoders, additive synthesis, LPC
Textures and soundscapes	Concatenative / granular synthesis, LPC, stochastic and wavelet-based methods
Transients	Onset detection, physical models, concatenative / granular synthesis, sines+transients+noise models

Table 1. Taxonomy of analysis/synthesis methods by the types of sounds for which they work well.

For a designer or developer wishing to create specific kinds of sounds, an even more concrete classification of methods may prove useful. Hence, we present a list of different types of sounds or synthesis goals, and analysis / synthesis methods known to work well for each of them (see table 1). The first column (sounds / goals) is inferred informally from the scopes of all the algorithms discussed. This approach allows us to consider a set of sound types that actually map well to specific algorithms. It does not, however, restrict the sounds or methods available to the user. Future exploration of existing and new techniques is bound to yield new mappings between methods and sounds. This paper offers a summary of currently known options and indicates starting points for further investigation.

One perspective on synthesizing environments is that due to the variety of sources they combine, it is most effective to separately model each source with the technique best suited to it. Future work with sound scenes and environments is then likely to explore and apply a mix of meth-

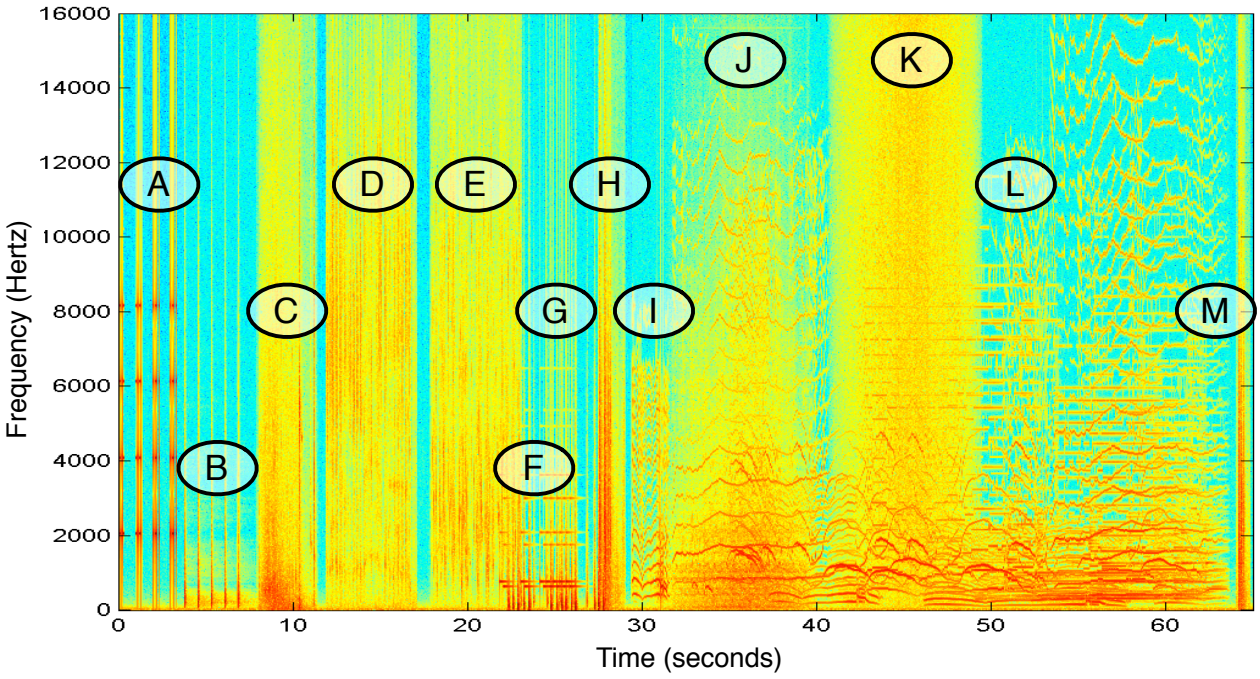

Figure 1. A concluding example (see section 7). This sound scene was created using multiple synthesis algorithms: oscillators (A), FM synthesis (F), parametric synthesis from scratch (B), modal synthesis (G), LPC (C, D, E, H, M), additive synthesis (I, L), concatenative texture synthesis (J), and feature-based synthesis (K). Audio available at `http://soundlab.cs.princeton.edu/listen/synthesis_example/`

ods described here. Another perspective is that a single flexible technique may satisfactorily model the majority of sounds. This presents an exciting avenue to explore, though such a technique may not model each sound in the best possible way. Several options lie between these two extremes: present the entire range of techniques to the machine and let it decide which to use on-the-fly, or select an all-encompassing technique as the default but let the user override it with another method if desired, or other options yet unexplored. Since many of these support the use of multiple algorithms, we believe our overview is relevant to sound designers and the wider computer music community.

7. A CONCLUDING EXAMPLE

To conclude, we present an example sound scene created using multiple synthesis algorithms (see Figure 1). The example combines aspects of "real-world" sound design and more abstract composition. It begins with **(A)** a beeping alarm clock (sinusoidal oscillator synthesis, section 2), followed by **(B)** footsteps on a wooden floor (parametric synthesis from scratch, GaitLab [21], section 3). Next come the sounds of **(C)** a sliding door opening, **(D)** a person brushing his teeth, and **(E)** running water (all re-synthesized using time-frequency LPC [5], section 4.3). **(F)** A doorbell (FM synthesis, section 2) follows, overlapping with **(G)** a door being pounded (modal synthesis excited by exponentially enveloped noise, section 3). The sound of **(H)** a door opening (noise-excited LPC, section 4.3) marks the transition to a less realistic scene. The next section begins with **(I)** a baby crying (sinusoidal additive synthesis [72], section 4.2), leading into a chorus of transformed versions of crying babies (also using additive synthesis). This chorus is overlaid with **(J)** several seconds of playground din (random overlap-add sound texture synthesis [40], section 4.1) followed by **(K)** white noise shaped to match the root-mean-square power of the preceding din (feature-based synthesis [43], section 3). **(L)** Wind-chimes transformed in frequency and time (additive synthesis, section 4.2) are also added to the mix, to create a richer scene. Finally, as this abstract section fades, the sound of **(M)** a door closing (noise-excited LPC, section 4.3) marks the end of the interlude and the sound scene.

This example illustrates the variety of algorithms that may contribute to a single fairly simple piece. Several factors influence the choice of algorithm: the specific sound to be synthesized (see table 1), the available material and tools (existing sound files, physical or perceptual models), the type and range of control desired, and other variables. Abstract algorithms may allow the most freedom of exploration, in that they examine the potential of any structure, algorithm, or data. However, they can also synthesize realistic sounds, such as **(A)** the alarm and **(F)** the doorbell. Synthe-

sis from scratch is especially suitable when there is a model or the means to create one, and often allows high-level parametric control over the synthesized sound. Synthesis-by-analysis is appropriate when given existing sounds to transform, arbitrarily extend, or otherwise re-use. Finally, primarily analytic techniques can also aid synthesis by providing meaningful information. Thus, each set of techniques has a synthesis scope defined by sound type and context.

Available software tools for synthesizing sound include programming languages (ChucK, SuperCollider, Pure Data, Max/MSP and more) as well as specialized software that offer great control over a fixed set of algorithms. All these enable some form of parametric audio synthesis. One advantage of thinking of audio in this way is the data compression it achieves. In our example, physical synthesis from scratch compresses 400:1 (1.1MB of sound files become 2.8kB of synthesis parameters and scripts). Other methods of synthesis together do not achieve as high compression (11.7MB become 3.5MB), partly because our additive synthesis stores the parameters in a verbose text format. One may store the same information in a more space-efficient way by using other formats [97] and performing intelligent compression. Further, a significantly longer example sound may easily be rendered from the same parameter files, suggesting even higher compression potential. Even so, a total of 12.8MB source files become 3.5MB synthesis files to synthesize 5.6MB for our entire 65-second sample.

Another advantage of using parametric sound synthesis algorithms instead of raw sound files is the ability to re-render over and again with minor tweaks or major transformations to one or more components. The timbre, distribution, duration, pitch, and other aspects of each component can be changed in multiple ways according to the synthesis algorithm used. Thus, we may choose to have more toothbrushing and less washing, or change the walker's gait and walking surface. Such variables can also be manipulated interactively and in real-time from external control inputs, aiding sound design for entertainment purposes.

A similar argument applies for composers. In the early days of computer music, compositions most often centered around one method or technique, but now just as the orchestral composer has winds, strings, percussion, *etc*. available in their palette, the electro-acoustic composer has a rich variety of techniques, each with strengths, weaknesses, and characteristics. We hope that our overview provides a working acquaintance with the full set of tools, as well as pointers to more information on specific techniques if needed, thus facilitating the creation of vivid scenes and compositions.

8. REFERENCES

[1] "SuperCollider 3 Plugins," [Software] http://sourceforge.net/projects/sc3-plugins/, retrieved 4 February 2009.

[2] X. Amatriain and P. Arumi, "Developing cross-platform audio and music applications with the CLAM framework," in *Proc. ICMC*, 2005.

[3] D. Arfib, "Digital synthesis of complex spectra by means of multiplication of non-linear distorted sine waves," *J. Audio Eng. Soc.*, vol. 27, no. 10, pp. 757–779, 1979.

[4] B. S. Atal, "Speech analysis and synthesis by linear prediction of the speech wave," *J. Acoust. Soc. Am.*, vol. 47, no. 65(A), 1970.

[5] M. Athineos and D. P. W. Ellis, "Sound texture modeling with linear prediction in both time and frequency domains," in *Proc. IEEE ICASSP*, vol. 5, 2003, pp. 648–651.

[6] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *IEEE T. Speech Audio Proc.*, vol. 13, no. 5, 2005.

[7] J. R. Beltrán and F. Beltrán, "Additive synthesis based on the continuous wavelet transform: A sinusoidal plus transient model," in *Proc. DAFx*, 2003.

[8] O. Ben-Tal, J. Berger, B. Cook, M. Daniels, G. P. Scavone, and P. R. Cook, "SonART: The Sonification Application Research Toolbox," in *Proc. Int. Conf. Audit. Disp.*, 2002.

[9] D. Birchfield, N. Mattar, and H. Sundaram, "Design of a generative model for soundscape creation," in *Proc. ICMC*, 2005.

[10] N. Bogaards, A. Robel, and X. Rodet, "Sound analysis and processing with AudioSculpt 2," in *Proc. ICMC*, 2004.

[11] G. J. Brown and M. Cooke, "Computational auditory scene analysis," *Comput. Speech Lang.*, vol. 8, no. 4, pp. 297–336, 1994.

[12] J.-F. Cardoso, "Blind signal separation: Statistical principles," *Proc. IEEE*, vol. 86, no. 10, pp. 2009–2025, 1998.

[13] M. A. Casey, R. Veltcamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proc. IEEE*, vol. 96, no. 4, pp. 668–696, 2008.

[14] J. Chowning, "The synthesis of complex audio spectra by means of frequency modulation," *J. Audio Eng. Soc.*, vol. 21, no. 7, pp. 526–534, 1973.

[15] ——, "Frequency modulation synthesis of the singing voice," in *Current Directions in Computer Music Research*, M. Mathews and J. Pierce, Eds. Cambridge, MA: The MIT Press, 1989, pp. 57–64.

[16] N. Collins, "Errant sound synthesis," in *Proc. ICMC*, 2008.

[17] P. R. Cook, "SPASM: A real-time vocal tract physical model editor/controller and Singer: the companion software synthesis system," *Comput. Music J.*, vol. 20, no. 3, pp. 38–46, 1992.

[18] ——, "Physically informed sonic modeling (PhISM): Percussive synthesis," in *Proc. ICMC*, 1996.

[19] ——, "Singing voice synthesis: History, current work, and future directions," *Comput. Music J.*, vol. 20, no. 3, pp. 38–46, 1996.

[20] ——, "Physically informed sonic modeling (PhISM): Synthesis of percussive sounds," *Comput. Music J.*, vol. 21, no. 3, pp. 38–49, 1997.

[21] ——, "Modeling BILL'S GAIT: Analysis and parametric synthesis of walking sounds," in *Proc. Audio Eng. Soc. Conf. Virt. Synthet. Ent. Audio*, 2002.

[22] ——, *Real Sound Synthesis for Interactive Applications*. Wellesley, MA: AK Peters, 2002.

[23] P. R. Cook and G. P. Scavone, "The Synthesis ToolKit (STK)," in *Proc. ICMC*, 1999.

[24] R. Dannenberg, "Abstract time warping of compound events and signals," *Comput. Music J.*, vol. 21, no. 3, pp. 61–70, 1997.

[25] J. Dattoro, "Effect design: Part 3 oscillators: Sinusoidal and pseudonoise," *J. Audio Eng. Soc.*, vol. 50, no. 3, pp. 115–146, 2002.

[26] A. DiScipio, "Composition by exploration of nonlinear dynamic systems," in *Proc. ICMC*, no. 324-327, 1990.

[27] M. B. Dolson, "The phase vocoder: A tutorial," *Comput. Music J.*, vol. 10, no. 4, pp. 14–27, 1986.

[28] S. Dubnov, Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, "Synthesizing sound textures through wavelet tree learning," *IEEE Comput. Graph. App.*, vol. 22, no. 4, 2002.

[29] H. Dudley, "The vocoder," *Bell Lab. Records*, 1939.

[30] D. P. W. Ellis, "A computer implementation of psychoacoustic grouping rules," in *Proc. Int. Conf. Pattern Recog.*, 1994.

[31] D. P. W. Ellis and B. L. Vercoe, "A perceptual representation of sound for auditory signal separation," in *Proc. 123rd meeting Acoust. Soc. Am.*, 1992.

[32] G. Essl, "Mathematical structure and sound synthesis," in *Proc. Sound Music Computing Conf.*, 2005.

[33] ——, "Circle maps as simple oscillators for complex behavior: I. Basics," in *Proc. ICMC*, 2006.

[34] ——, "Circle maps as simple oscillators for complex behaviors: II. Experiments," in *Proc. DAFx*, 2006.

[35] G. Essl, S. Serafin, P. R. Cook, and J. O. Smith III, "Theory of banded waveguides," *Comput. Music J.*, vol. 28, no. 1, pp. 37–50, 2004.

[36] A. Farnell, *Designing Sound*. Applied Scientific Press, 2009, retrieved 7 February 2009 from [Online] http://aspress.co.uk/ds/bookInfo.html.

[37] C. Fevotte and S. J. Godsill, "A Bayesian approach for blind separation of sparse sources," *IEEE T. Audio Speech Lang. Proc.*, vol. 14, no. 6, pp. 2174–2188, 2006.

[38] K. Fitz and L. Haken, "Sinusoidal modeling and manipulation using Lemur," *Comput. Music J.*, vol. 20, no. 4, 1996.

[39] F. Fontana and D. Rocchesso, "Physical modeling of membranes for percussion instruments," *Acustica*, vol. 84, no. 3, pp. 529–542, 1998.

[40] M. Fröjd and A. Horner, "Fast sound texture synthesis using overlap-add," in *Proc. ICMC*, 2007.

[41] S. W. Hainsworth, "Techniques for the automated analysis of musical audio," Ph.D. dissertation, University of Cambridge, UK, 2003.

[42] L. Hiller and P. Ruiz, "Synthesizing musical sounds by solving the wave equation for vibrating objects," *J. Audio Eng. Soc.*, vol. 19, pp. 462–472, 1971.

[43] M. Hoffman and P. R. Cook, "Feature-based synthesis: Mapping from acoustic and perceptual features to synthesis parameters," in *Proc. ICMC*, 2006.

[44] R. Hoskinson and D. K. Pai, "Manipulation and resynthesis with natural grains," in *Proc. ICMC*, 2001.

[45] D. Jaffe and J. O. Smith III, "Extensions of the Karplus-Strong plucked-string algorithm," *Comput. Music J.*, vol. 7, no. 2, pp. 56–69, 1983.

[46] K. Karplus and A. Strong, "Digital synthesis of plucked-string and drum timbres," *Comput. Music J.*, vol. 7, no. 2, pp. 43–55, 1983.

[47] A. Klapuri, "Signal processing methods for the automatic transcription of music," Ph.D. dissertation, Tampere University of Technology, Finnland, 2004.

[48] M. Klingbeil, "Software for spectral analysis, editing, and synthesis," in *Proc. ICMC*, 2005.

[49] G. Kramer, Ed., *Auditory Display: Sonification, Audification, and Auditory Interfaces*, ser. Santa Fe Institute Studies in the Sciences of Complexity Proc. Vol. XVIII. Reading, MA: Addison-Wesley, 1994.

[50] P. Lansky, "Compositional applications of linear predictive coding," in *Current Directions in Computer Music Research*, M. Mathews and J. Pierce, Eds. Cambridge, MA: MIT Press, 1989, pp. 5–8.

[51] A. Lazier and P. R. Cook, "MOSIEVIUS: Feature-driven interactive audio mosaicing," in *Proc. DAFx*, 2003.

[52] M. LeBrun, "Digital waveshaping synthesis," *J. Audio Eng. Soc.*, vol. 27, no. 4, pp. 250–266, 1979.

[53] S. N. Levine and J. O. Smith III, "A sines+transients+noise audio representation for data compression and time/pitch scale modifications," in *Audio Eng. Soc. Conv.*, 1998.

[54] J. Makhoul, "Linear prediction: A tutorial review," in *Proc. IEEE*, vol. 63, 1975, pp. 561–580.

[55] J. A. Maurer, "The influence of chaos on computer-generated music," [Online] http://ccrma-www.stanford.edu/~blackrse/chaos.html, 1999, retrieved 27 January 2009.

[56] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE T. Acoust. Speech Sig. Proc.*, vol. 34, no. 4, pp. 744–754, 1986.

[57] K. Melih and R. Gonzalez, "Source segmentation for structured audio," in *IEEE Int. Conf. Multim. Expo (II)*, 2000, pp. 811–814.

[58] N. E. Miner and T. P. Caudell, "A wavelet synthesis technique for creating realistic virtual environment sounds," *Presence*, vol. 11, no. 5, pp. 493–507, 2002.

[59] A. Misra, G. Wang, and P. R. Cook, "Musical tapestries: Re-composing natural sounds," *J. New Music Res.*, vol. 36, no. 4, 2007.

[60] J. F. O'Brien, , P. R. Cook, and G. Essl, "Synthesizing sounds from physically based motion," in *Proc. ACM SIGGRAPH*, 2001, pp. 529–536.

[61] T. H. Park, "Towards automatic musical instrument timbre recognition," Ph.D. dissertation, Princeton University, NJ, 2004.

[62] S. Pauletto and A. Hunt, "A toolkit for interactive sonification," in *Proc. Int. Conf. Audit. Disp.*, 2004.

[63] J. R. Pierce and S. A. Van Duyne, "A passive nonlinear digital filter design which facilitates physics-based sound synthesis of highly nonlinear musical instruments," *J. Acoust. Soc. Am.*, vol. 101, no. 2, pp. 1120–1126, 1997.

[64] S. T. Pope, "A taxonomy of computer music," *Contemp. Music Rev.*, vol. 13, no. 2, pp. 137–145, 1996.

[65] Y. Qi, T. P. Minka, and R. W. Picard, "Bayesian spectrum estimation of unevenly sampled nonstationary data," in *Proc. IEEE ICASSP*, 2002.

[66] T. Quatieri and R. McAulay, "Speech transformations based on a sinusoidal representation," *IEEE T. Acoust. Speech Sig. Proc.*, vol. 34, no. 6, pp. 1449–1464, 1986.

[67] J.-C. Risset, "Computer music experiments, 1964–..." *Comput. Music J.*, vol. 9, pp. 11–18, 1985.

[68] D. Rocchesso, "Physically-based sounding objects, as we develop them today," *J. New Music Res.*, vol. 33, no. 3, pp. 305–313, 2004.

[69] X. Rodet, "Time-domain formant-wave-function synthesis," *Comput. Music J.*, vol. 8, no. 3, pp. 9–14, 1984.

[70] D. Schwarz, "Concatenative sound synthesis: The early years," *J. New Music Res.*, vol. 35, no. 1, pp. 3–22, 2006.

[71] ——, "Corpus-based concatenative synthesis," *IEEE Sig. Proc. Mag.*, pp. 92–104, 2007.

[72] X. Serra, "A system for sound analysis / transformation / synthesis based on a deterministic plus stochastic decomposition," Ph.D. dissertation, Stanford University, 1989.

[73] J. O. Smith III, "Efficient simulation of the reed-bore and bow-string mechanisms," in *Proc. ICMC*, 1986, pp. 275–280.

[74] ——, "Musical applications of digital waveguides," Center for Computer Research in Music and Acoustics, Music Department, Stanford University, Tech. Rep. STAN-M-39, 1987.

[75] ——, "Waveguide filter tutorial," in *Proc. ICMC*, 1987, pp. 9–16.

[76] ——, "Viewpoints on the History of Digital Synthesis (keynote paper)," in *Proc. ICMC*, 1991.

[77] ——, "Waveguide simulation of non-cylindrical acoustic tubes," in *Proc. ICMC*, 1991, pp. 304–307.

[78] J. O. Smith III and P. R. Cook, "The second-order digital waveguide oscillator," in *Proc. ICMC*, 1992.

[79] K. Steiglitz and P. Lansky, "Synthesis of timbral families by warped linear prediction," *Comput. Music J.*, vol. 5, no. 3, pp. 45–49, 1981.

[80] D. Stowell and M. Plumbley, "Adaptive whitening for improved real-time audio onset detection," in *Proc. ICMC*, 2007.

[81] G. Strobl and E. Gerhard, "Sound texture modeling: A survey," in *Proc. Sound Music Computing Conf.*, 2006.

[82] B. L. Sturm, C. Roads, A. McLeran, and J. J. Shynk, "Analysis, visualization, and transformation of audio signals using dictionary-based methods," in *Proc. ICMC*, 2008.

[83] T. Takala and J. Hahn, "Sound rendering," in *Proc. ACM SIGGRAPH*, 1992, pp. 211–220.

[84] T. Tolonen, V. Välimäki, and M. Karjalainen, "Evaluation of modern sound synthesis methods," Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, Espoo, Finland, Tech. Rep. 48, 1998.

[85] B. Truax, "Chaotic non-linear systems and digital synthesis: An exploratory study," in *Proc. ICMC*, 1990, pp. 100–103.

[86] ——, "Composing with real-time granular sound," *Persp. New Music*, vol. 28, no. 2, 1990.

[87] D. Trueman and R. L. DuBois, "PeRColate: A collection of synthesis, signal processing, and image processing objects for Max/MSP," [Software] http://www.music.columbia.edu/PeRColate/, retrieved 4 February 2009.

[88] G. Tzanetakis and P. R. Cook, "MARSYAS: A framework for audio analysis," *Org. Sound*, vol. 4, no. 3, 2000.

[89] ——, "Musical genre classification of audio signals," *IEEE T. Speech Audio Proc.*, vol. 10, no. 5, pp. 293–302, 2002.

[90] K. van den Doel, P. G. Kry, and D. K. Pai, "FOLEYAUTOMATIC: Physically-based sound effects for interactive simulation and animation," in *Proc. ACM SIGGRAPH*, 2001.

[91] S. A. Van Duyne and J. O. Smith III, "The 2-D digital waveguide mesh," in *IEEE Workshop App. Sig. Proc. Audio Acoust.*, 1993.

[92] ——, "The 3D tetrahedral digital waveguide mesh with musical applications," in *Proc. ICMC*, 1996.

[93] B. L. Vercoe, W. G. Gardner, and E. D. Scheirer, "Structured audio: Creation, transmission, and rendering of parametric sound representations," in *Proc. IEEE*, vol. 86, no. 5, 1998, pp. 922–940.

[94] T. S. Verma and T. H. Meng, "An analysis/synthesis tool for transient signals that allows a flexible sines+transients+noise model for audio," in *Proc. IEEE ICASSP*, 1998, pp. 12–15.

[95] G. Wang and P. R. Cook, "ChucK: A concurrent, on-the-fly, audio programming language," in *Proc. ICMC*, 2003, pp. 219–226.

[96] G. Widmer, D. Rocchesso, V. Välimäki, C. Erkut, F. Gouyon, D. Pressnitzer, H. Penttinen, P. Polotti, and G. Volpe, "Sound and music computing: Research trends and some key issues," *J. New Music Res.*, vol. 36, no. 3, pp. 169–184, 2007.

[97] M. Wright, A. Chaudhary, A. Freed, S. Khoury, and D. Wessel, "Audio applications of the Sound Description Interchange Format standard," in *Audio Eng. Soc. Conv.*, 1999.

[98] W. S. Yeo and J. Berger, "Application of image sonification methods to music," in *Proc. ICMC*, 2005.

[99] X. Zhu and L. Wyse, "Sound texture modeling and time-frequency LPC," in *Proc. DAFx*, 2004, pp. 345–349.