Feature-Based Synthesis: Mapping Acoustic and Perceptual Features onto Synthesis Parameters

Matt Hoffman, Perry R. Cook

Department of Computer Science, Princeton University
{mdhoffma, prc} at cs.princeton.edu

Abstract

Substantial progress has been made recently in finding acoustic features that describe perceptually relevant aspects of sound. This paper presents a general framework for synthesizing audio manifesting arbitrary sets of quantifiable acoustic features of the sort used in music information retrieval and other sound analysis applications. The methods described have broad applications to the synthesis of novel musical timbres, processing by analysis-resynthesis, efficient audio coding, generation of psychoacoustic stimuli, and music information retrieval research.

1 Introduction

Much of the work done in sound analysis focuses on applying machine learning techniques to descriptors consisting of vectors of perceptually motivated features derived from input audio signals. These features are statistics thought to correlate well with the sonic cues that humans use to perceive sound. We have implemented a general framework for synthesizing, given any set of such feature values, frames of audio matching them as closely as possible.

This approach differs from most existing sound synthesis techniques, whose control parameters are generally defined by underlying synthesis algorithms rather than by the desired output. Those output-driven synthesis techniques that do exist mostly either focus on matching the raw frequency spectra of existing sounds or rely on user-defined control points in a predefined synthesizer parameter space. By contrast, our feature-based synthesis system enables sounds characterized by any set of quantifiable features to be synthesized using any synthesis algorithm, so long as there exists a set of synthesis parameters that maps to the desired feature values and can be found reasonably efficiently.

The ability to synthesize sound with arbitrary perceptual characteristics has numerous compelling applications. The problem of mapping real-time musical controller inputs to synthesis parameters can be solved by mapping the inputs directly onto synthesizable perceptual characteristics. Electronic composers can synthesize sounds designed specifically to fit a piece's specific timbral needs. Music information retrieval researchers trying to model specific perceptual characteristics for, e.g., classification purposes can use feature-based synthesis to evaluate how well given feature sets meet their needs. By extracting a set of features from a sound and applying feature-based synthesis to resynthesize a sound matching those features, one can directly observe what information the chosen features do and do not capture. This may suggest new features capable of encoding relevant information. Furthermore, if a sound from a given domain *can* be resynthesized with a high degree of perceptual accuracy from a small set of features then only those few feature values need be stored, resulting in a potentially near-optimal audio coding technique. Finally, if one has a good feature representation of a sound and wishes to create a novel sound that mimics the properties of the original sound, one can simply alter one or more of the features and synthesize a new version that retains only some properties of the original.

Our system frames the problem of feature synthesis in terms of minimizing an arbitrarily defined distance function between the target feature vector and the feature vector describing the synthesized sound over the set of underlying synthesis parameters. The mapping between the feature space and the parameter space can be highly nonlinear, complicating optimization. We present a modular framework that separates the tasks of feature extraction, feature comparison, sound synthesis, and parameter optimization, making it possible to mix and match various techniques in the search for an efficient and accurate solution to the broadly defined problem of synthesizing sounds manifesting arbitrary perceptual features.

2 Related Work

Previous attempts to solve similar problems have tended to fall into one of two categories. The first set of approaches revolves around finding synthesis parameter values that will produce a sound closely matching an existing sound. The second focuses on mapping some set of input parameters onto a set of synthesis parameters in a way that gives users straightforward control over the synthesized output without sacrificing the range of available sounds.

2.1 Matching Synthesis

Much of the work on this problem was done by Andrew Horner and James Beauchamp, whose approaches in some ways closely resemble ours. Horner, Beauchamp, and Haken (93) used genetic algorithms to find the parameters for single modulator, multiple carriers FM that minimize the difference between the resulting signal's spectrum and that of an arbitrary instrumental tone. Horner, Cheung, and Beauchamp (95) also applied similar techniques to other synthesis algorithms, all to good effect. More recently, Horner, Beauchamp, and So (2004) and Wun, Horner, and Ayers (2004) explored other error metrics, including perceptually influenced metrics, and search methods, respectively.

Although the approaches taken by Horner et al. over the years have, like our system, used various algorithms to find synthesis parameters that minimize the error of the audio generated by several synthesis functions with respect to some target sound, several key differences set our approach apart from theirs. Whereas resynthesizing existing sounds accurately has been Horner et al.'s primary interest, this is only part of what we wish to accomplish with feature-based synthesis. Our approach does not require an input sound, and allows for perceptually meaningful and well-defined modifications of input sounds.

2.2 User Input-Synthesis Parameter Mapping

Much of the work on mapping arbitrary control parameters to algorithm-dependent synthesis parameters has focused specifically on facilitating the expressive use of real-time controllers. However, as Lee and Wessel (1992) point out, many if not most reasonably powerful musical synthesis algorithms depend on the settings of a large number of highly nonlinear parameters. Techniques for easily and predictably navigating through this high-dimensional parameter space to desirable sonic results should therefore be of some interest even to offline users of such algorithms, and especially to first-time users.

Lee and Wessel (1992) explored the use of neural networks to map from an input space into a user-defined timbre space. Taking a more straightforward, mathematical approach, Bowler et al. (1990) developed a method for efficiently mapping N control parameters to M synthesizer parameters, where the mapping was again defined in terms of a user-generated timbre space of sorts filled out by interpolating between control points. Although such techniques may be effective for specific instruments and synthesis algorithms, they demand that the user define an explicit mapping from synthesis parameter space to an ersatz timbre space. This must be repeated any time a new synthesis algorithm is used, which limits the generalizability of such techniques.

2.3 Feature-Specific Synthesis

Some work has been done in synthesizing sounds to fit values for specific features. Lidy, Pölzlbauer, and Rauber (2005), for example, developed a technique to synthesize audio from their Rhythm Patterns feature descriptor having very specific beat patterns in various frequency bands, and Lakatos, Cook, and Scavone (2000) have synthesized noises with specific spectral centroids for their studies of human perception. But, although it is frequently possible or even trivial to find a closed-form solution to the problem of synthesizing sounds having specific values for one or two feature values, finding parameters to synthesize a sound exactly fitting many feature values very quickly grows intractable.

3 Motivation

Our approach is substantially more flexible than any of the approaches previously described, which permits its application to quite a wide variety of problems. These can be divided roughly into three categories based on the ways in which the target feature values are obtained and/or modified. We first consider the case where feature values are extracted from an existing sound file and then resynthesized without modification. Next, the case where feature values again are extracted from an existing sound file, but this time are modified before resynthesis. Finally, we consider the case where feature values are specified arbitrarily, without reference to an existing sound file.

3.1 Resynthesis Without Modification

Perceptual Audio Coding. Ideally, one would be able to define and extract a set of features that captured virtually all of the perceptually relevant information present in a frame of audio without any redundancy. If this lofty goal were achieved, then any two frames of audio with nearly identical values for those features would be almost perceptually indistinguishable. Given the ability to extract such a set of features and the ability to synthesize frames of audio conforming to arbitrary values for those features, one would only need to store a relatively tiny amount of data to recreate a perceptually indistinguishable copy of any sound. In fact, that set of feature values would represent the absolute theoretical minimum amount of information that one could use to store a sound. Of course, finding such a set of features is in general a challenge that may never be met, but if certain domain-specific assumptions can be made about one's input data then the problem becomes much more tractable.

Feature Set Evaluation. Of course, in reality it is difficult to design a compact feature set capable of capturing all of the perceptually relevant information in a sound, as is evidenced by the continuing research into improved features for music information retrieval. Although the effectiveness of feature sets used in music information retrieval problems can (and should) be quantitatively evaluated based on their performance in real tasks, such results offer little insight into why a set of features is or is not successful in describing relevant information. As Lidy, Pölzlbauer, and Rauber (2005) observe, one way of qualitatively evaluating the meaningfulness of a feature set is through an analysis-by-synthesis process where one extracts the features in question from multiple sound files from the target domain, synthesizes new sounds matching the extracted features, and compares the original and resynthesized versions. If the resynthesized version of a sound file lacks some quality relevant to the problem at hand, then it is likely that the addition of a feature representing that quality to the feature set will improve performance.

3.2 Resynthesis With Modification

Feature-based resynthesis offers a paradigm for perceptually motivated effects processing of sounds. Given a fairly complete feature representation for a specific domain (i.e. a representation such that two sounds with very similar feature descriptors sound qualitatively very similar within that domain), one can extract a set of feature values from a sound, modify one or more of those values, and synthesize a new sound that should be qualitatively different in the perceptual dimensions that were modified while still evoking the original sound in those that were not. This approach contrasts with many traditional audio effects that are defined in terms of signal processing operations rather than perceptual impact.

Given a less complete feature set, one can create a variety of sounds that sound similar to the original sound but are free to wander away from the original sound in dimensions not codified by the feature set. Depending on the chosen underlying synthesis algorithm, this approach could be used to generate timbres intermediate between that of the original sound and those characteristic of the synthesis algorithm.

3.3 Synthesis of Arbitrary Feature Values

Mapping of Real-Time Controller Input to Synthesis Parameters.

It has been observed many times that finding an appropriate mapping from the user input parameters of a real-time controller to the (usually much more numerous) parameters controlling a synthesis algorithm is often both important to the expressive power of the performance system and extremely challenging (Hunt and Kirk, 2000; Hunt, 1999; Lee and Wessel, 92). Existing techniques for automatically finding such mappings generally require some sort of time-consuming user exploration of the parameter space to construct at least one intermediary layer mapping from the parameters controlled and perceived by the user to the abstract synthesis parameters hidden from the user. Feature-based synthesis conveniently provides an automatic mapping from the high-dimensional synthesis parameter space to the lower-dimensional, more perceptually relevant feature space, to which control parameters can be mapped.

Offline Synthesis of Perceptually Specified Sounds. Finally, feature-based synthesis can be used to synthesize sounds specified exclusively by perceptual qualities. This could be of benefit to electronic composers looking for novel sounds with a sense of what characteristics they would like such sounds to have. Additionally, researchers studying human perception can synthesize sounds with arbitrary feature values for experiments studying the perception of aural stimuli with specific perceptual characteristics.

4 Implementation

Our architecture focuses on four main modular components: feature evaluators, parametric synthesizers, distance metrics, and parameter optimizers. Feature evaluators take a frame of audio as input and output an n-dimensional vector of real-valued features. Parametric synthesizers take an m-dimensional vector of real-valued inputs and output a frame of audio. Distance metrics define some arbitrarily complex function that compares how "similar" two n-dimensional feature vectors are. Finally, parameter optimizers take as input a feature evaluator **F**, a parametric synthesizer **S**, a distance metric **D**, and an n-dimensional feature vector **v** generated by **F** (which **D** can compare to another such feature vector). The parameter optimizer **P** outputs a new m-dimensional synthesis parameter vector **u'**, a new n-dimensional feature vector **v'**, and a frame of audio representing the output of **S** when given **u'**. This frame of audio produces **v'** when given as input to **F**. **v'** represents the feature vector as close to **v** (where distance is defined by **D**) as **P** was able to find in the parameter space of **S**.

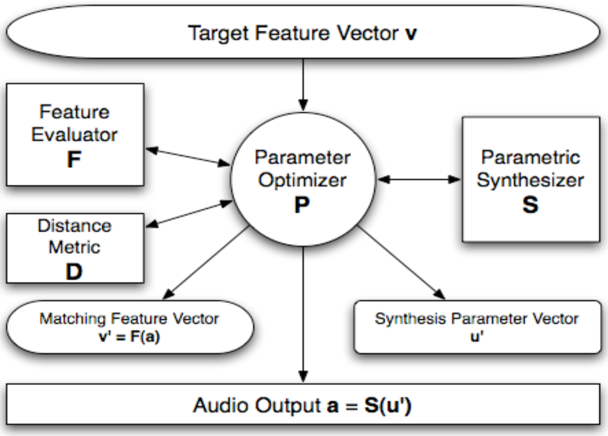

Figure 1. Overview of the architecture of the framework. Given a target feature vector **v**, **P** searches through the parameter space of **S** to find a set of synthesis parameters **u'** that will minimize **D(v, F(S(u')))**.

These four components together make up a complete system for synthesizing frames of audio characterized by arbitrary feature vectors. Any implementation of one of these components is valid, so long as it adheres to the appropriate interface. The design of the framework makes it possible to use any parameter space search algorithm to seek parameters for any synthesis algorithm that produce audio characterized by any set of automatically extractable features and any distance function.

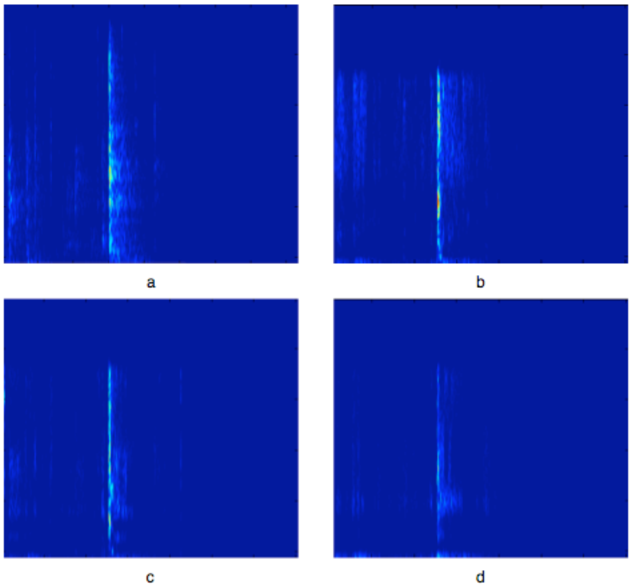

Figure 2. Spectrograms of a door closing resynthesized using noise shaped by 10 MFCCs. 2-a represents the original sound, 2-b is the sound as resynthesized based only on the spectral centroid, 3-b is the sound as resynthesized from the 10%, 20%,…, 100% rolloff points of the spectrum, and 4-b is the sound as resynthesized as though those 10 rolloff points had been 25% lower.

Preliminary results using stationary sinusoids of various frequencies and spectrally shaped noise (with noise shape controlled by Mel-Frequency Cepstral Coefficients as implemented by Slaney (1998)) to match spectral centroids, spectral rolloffs, and multiple pitch histogram data (Tzanetakis, Ermolinskyi, and Cook, 2002) and relatively crude optimizers using a simple Euclidean L2 distance function have been good. Spectrograms resulting from the resynthesis of a door closing are presented in figure 2. Note that in 2-d, we successfully synthesize a modified version of the sound only by modifying feature values.

5 Conclusion and Future Work

We have implemented a framework for synthesizing frames of audio described by arbitrary sets of well-defined feature values. The general techniques our system can be used to a wide variety of ends, from efficient audio coding, to data exploration, to sound design, to composition.

The next stage consists of implementing more feature evaluators, synthesizers, optimizers, and distance metrics, and evaluating the system's performance more rigorously in a variety of domains. Finding ways to improve search times will also be valuable in preparing the framework for use in real-time situations. We also plan to implement an interactive interface to facilitate testing and development.

References

Horner, A., Beauchamp, J., and Haken, L. 1993. Machine tongues XVI: Genetic algorithms and their application to FM matching synthesis. *Computer Music Journal* 17(3), 17-29.

Horner, A., Cheung, N., and Beauchamp, J. 1995. Genetic algorithm optimization of additive synthesis envelope breakpoints and group synthesis parameters. In *Proceedings of the International Computer Music Conference,* pp. 215-222. San Francisco: International Computer Music Association.

Horner, A., Beauchamp, J., and So, R. 2004. A search for best error metrics to predict discrimination of original and spectrally altered musical instrument sounds. In *Proceedings of the International Computer Music Conference,* pp. 9-16. San Francisco: International Computer Music Association.

Wun, S., Horner, A., and Ayers, L. 2004. A comparison between local search and genetic algorithm methods for wavetable matching. In *Proceedings of the International Computer Music Conference,* pp. 386-389. San Francisco: International Computer Music Association.

Lee, M., and Wessel, D. 1992. Connectionist models for real-time control of synthesis and compositional algorithms. In *Proceedings of the International Computer Music Conference,* pp. 277-280. San Francisco: International Computer Music Association.

Bowler, I., Purvis, A., Manning, P., Bailey, N. 1990. On mapping N articulation onto M synthesizer-control parameters. In *Proceedings of the International Computer Music Conference,* pp. 181-184. San Francisco: International Computer Music Association.

Lidy, T., Pölzlbauer, G., Rauber, A. 2005. Sound re-synthesis from rhythm pattern features – audible insight into a music feature extraction process. In *Proceedings of the International Computer Music Conference,* pp. 93-96. San Francisco: International Computer Music Association.

Lakatos, S., Cook, P., Scavone, G. 2000. Selective attention to the parameters of a physically informed sonic model. *Acoustics Research Letters Online*, Acoustical Society of America, March 2000.

Hunt, A. 1999. Radical user interfaces for real-time musical control. Ph.D. dissertation, University of York, York, U.K.

Hunt, A. and Kirk, R. 2000. Mapping strategies for musical performance – trends in gestural control of music. In *Trends in Gestural Control of Music*, M. Wanderley and M. Battier, eds. Paris, France Institut de Recherche et Coordination Acoustique Musique—Centre Pompidou, 2000, pp. 231–258.

Slaney, M. 1998. "Auditory toolbox," Technical Report # 1998-010. Interval Research Corporation, Palo Alto, CA

Tzanetakis, G., Ermolinskyi, A., Cook, P. 2002. Pitch histograms in audio and symbolic music information retrieval. In *Proceedings of ISMIR.*