AUDIOGUIDE: A FRAMEWORK FOR CREATIVE EXPLORATION OF CONCATENATIVE SOUND SYNTHESIS

Benjamin Hackbarth

CRCA
University of California, San Diego
9500 Gilman Drive
La Jolla, CA 92093-0037 USA
hackbarth@ucsd.edu

Norbert Schnell, Diemo Schwarz

IRCAM Centre Pompidou
1, place IgorStravinsky
75004 Paris, France
Norbert.Schnell@ircam.fr
Diemo.Schwarz@ircam.fr

ABSTRACT

This paper outlines the creative and technical considerations behind AudioGuide[1], a program for differed-time concatenative synthesis written in Python. AudioGuide is a framework for experimentation with a flexible concatenative algorithm. Two aspects of this algorithm are discussed in detail. First, strategies for flexible feature mapping are presented which allow the user a degree of expressive freedom in shaping and transforming concatenated outputs. Second, a subtractive spectral algorithm is outlined which enables the selection of simultaneous corpus units. Simultaneous selection permits both vertically stratified units as well as horizontally overlapping units and, more generally, permits a more flexible and comprehensive deployment of corpus resources. These two aspects, used in tandem, create a software framework capable of both sound synthesis and musical writing, enabling the user to move freely between sonic and score outputs.

1. INTRODUCTION

The motivation to develop of AudioGuide was driven primarily by two compositional needs. The first was a desire to create an intuitive tool for generating and controlling gesture in electronic music. When composing electronic sound using samples, forming gestures can become a laborious task in which creativity and musical imagination are limited by the density of detail necessary to prescribe the assemblage of sound segments. AudioGuide is an attempt to create an algorithm where gesture can be specified in a more generalized manner such that compositional attention shifts from the specificity of moments to the relationship of moments over time.

The second compositional impetus was creating a software framework capable of arranging sounds in time such that they are evocative of nuanced, time-varying acoustic morphologies. Of particular interest was having the ability to densely layer sound segments so that they fuse together. This fusion is a form of synthesis and creates an interesting aesthetic tension where, when listening, one hears both the individual sounds' phenomenological identifies as well as a morphological structure which transcends the sum of individual events.

1.1. Target as Gesture-Template

Concatenative synthesis[5] was selected early on because it is well suited for exploring the aforementioned compositional ideas. Similar to other concatenative synthesis applications, AudioGuide is premised upon the idea of using a target soundfile's amplitude profile and spectral features to drive the assemblage of database units. When parameterizing a concatenation, the user selects a target soundfile and a database of soundfiles and the algorithm assembles database units according to the target's features.

Seen from within the context of our application, gesture is the target soundfile's temporal profile populated with the discretely valued spectral measurements. Compared to selecting samples by hand or to constructing breakpoint envelopes to create a morphological profile, using a soundfile as the target is both intuitive and richly-detailed.

However, as is outlined in subsequent sections, AudioGuide includes tools which allow the user to manipulate the target's representation such that resulting concatenations can deviate from a straightforward imitation of the target soundfile. Thus, rather than thinking about unit selection as being predicated upon imitation, the target sound is better thought of as an abstract gesture-template consisting feature contours and their time-dependent correlations.

2. BACKGROUND

AudioGuide was developed during Hackbarth's residency in musical research at IRCAM in 2010. Development started with a comprehensive investigation and evaluation of several programs for realtime and differed-time concatenative synthesis – CataRT[6], concatenative synthesis using Mubu[4], and Orchidée[6]. Each of these projects influenced the features and capabilities of AudioGuide. In particular, AudioGuide's segmentation algorithm is based upon that which is implemented by Schnell and Suarez Cifuentes using Mubu in MAX/MSP. In addition, the idea

[1] http://crca.ucsd.edu/~ben/audioGuide

of segmented metrics based upon the energy-weighted average of time-varying features is identical to that used in Schnell's work.

Orchidée was influential when designing the algorithm for vertical simultaneous unit selection. While AudioGuide's methodology for simultaneous selection is markedly more primitive than Orchidée's, using a soundfile as a time-varying gesture-template complicates the implementation of Orchidée's algorithm.

3. RESEARCH

Unit selection is made according to euclidean distance calculations using continuously valued amplitude and spectral features provided by libXtract[1] as well as monophonic pitch estimates using Yin[2]. Evaluating sound segments with continuously valued features permits AudioGuide to match larger chunks of database content to target segments. Preservation of the morphological profile of corpus units permits the retention of higher order phenomenological identities which do not survive if the database is segmented into window-sized grains (though this possibility is also supported by AudioGuide).

When parameterizing a concatenation, the user may prescribe any number of features with different weights in order to influence each feature's saliency during unit selection. In addition to varying the weights of features, the user may specify different normalization strategies and transformations for each feature which permit more robust compositional control over resulting concatenations.

3.1. Normalization

Before database units are matched to target segments, continuously valued features are normalized in order to facilitate similarity measurements. Normalization may be accomplished through either scaling by the minimum and maximum or through scaling by the median and standard deviation. While AudioGuide includes a variety of strategies for normalizing features, one aspect merits special attention – the decision of whether to normalize the target and database data together or separately.

This choice in normalization affects the *context* of similarity in the resulting concatenation and has marked consequences as to how the database is deployed to follow the target's gesture-data. If target and database data are normalized together, AudioGuide will select database segments that best represent each individual target segment. In general, database units are selected more sparingly in order to optimize moment-to-moment matching. If the target and database data are normalized separately, AudioGuide will select database segments that best represent the *variability* of the target segments taken as a whole. As a consequence, the database is generally deployed more comprehensively as the target's variability is effectively stretched or compressed to better fit the variability of the database. Figure 1 visualizes two concatenations of a synthetic target and a database of piano samples to illustrate the different outcomes of these normalization strategies.

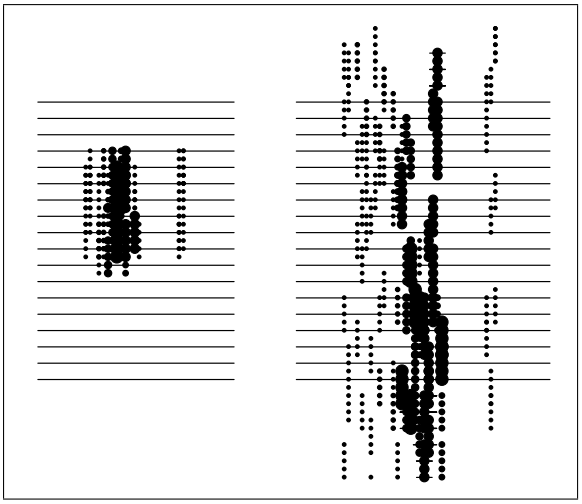

Figure 1. Two concatenations created with AudioGuide using the same target and the same database. On the left: the target and database data are normalized together. On the right: the target and database data are normalized separately. Each concatenation is based upon the target's centroid and spectral power and both contain a total of 96 corpus units. Visualization created with Processing to show piano samples' pitch and amplitude on a semitone grid. (Link to soundfile.)

From a creative standpoint, this distinction is of critical import as it allows the composer to choose between a notion of similarity based upon moments or based upon the relationship of moments over time. When the target and database are normalized together, unit selection can be thought of as imitation. However, when the target and database are normalized separately, unit selection can be thought of as a kind of "sonic transcription" wherein target data is metaphorically *transcribed* onto the variability of the database. Under the sonic transcription model, spectral/morphological influence shifts from a vertical emphasis (matching slices of time) to a horizontal emphasis (matching the relationship of slices over time).

3.2. Feature Mapping

AudioGuide provides a suite of tools for manipulating and distorting the target data's normalization coefficients in order to influence unit selection. In addition, AudioGuide also provides tools for modifying the target's normalized features to affect unit selection. The user may apply transformations which affectively compress, expand and/or invert the target's normalized data. A simple interface provides standard arithmetic operators to the user.

While often decreasing the fidelity of resulting similarity, these normalization and data transformation strategies permit the user to shape and sculpt the target's representation in order to alter the concatenated output. Thus, rather than considering the target soundfile a fixed object for imitation, normalization and transformative tools allow the user to deploy the gestural profile of the target in a multitude of different ways. This encourages the com-

poser to treat the target soundfile as a correlated set of feature contours (a gesture-template) which can be mapped on to the database with differing degrees of likeness and semblance. Therefore a large array of concatenated "variations" can be obtained with a single target and a single database.

3.3. Subtractive Spectral Algorithm

In AudioGuide the target are corpus soundfiles are represented internally by their time-varying mel-amplitude spectra. All of the features which are used for similarity calculation and unit selection – onset detection, loudness and spectral measurements – are calculated off of mel-amplitude representations of both target and corpus segments. While some spectral metrics suffer a loss in precision by being calculated off mel-spectra, perceptually relevant bin spacing permit the implementation of an algorithm for simultaneous unit selection.

Each time a corpus segment is chosen, the segment's time-varying mel-amplitudes are subtracted from the relevant region of the target's mel-amplitudes. This yields a "residual" target spectrum upon which the temporal and spectral features involved in unit selection are recalculated. If onset criteria for selection are still met (usually, if enough energy remains in the residual spectrum), additional units are chosen according to the recalculated features. This strategy for simultaneous selection is a matching pursuit algorithm where the best-matching unit is selected first – usually a database segment with relatively high amplitude – and then subsequent units are chosen to account for spectral energy which is not yet represented – usually relatively softer segments. Figure 2 shows the successive selection of three corpus units to approximate the temporal and spectral characteristics of a single target segment. The target's subtracted mel-spectra are shown in boxes in succession from left to right.

The subtraction-based selection algorithm permits database content to overlap freely both by selecting multiple database segments to begin at the same moment in time or through staggering selected units such that their onsets and offsets overlap in time. This algorithm permits a more comprehensive deployment of database resources especially when corpus units do not match the target's temporal profile with a high degree of precision. Figure 2 presents a typical example where multiple corpus units are chained together to approximate a single target segment.

3.4. Musical Writing

While computer assisted composition was not a consideration during the initial planning phase of AudioGuide, application of the concatenative algorithm to writing acoustic music became increasingly enticing due to AudioGuide's ability to match whole acoustic segments. When creating a MIDI-based score rather than a soundfile output, the user must supply AudioGuide with pitch estimates for each corpus segment along with optional text which

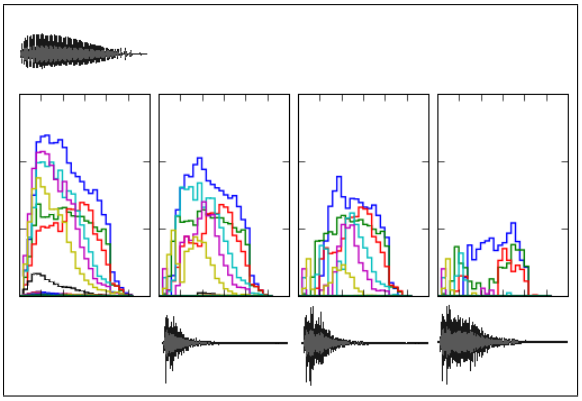

Figure 2. A visualization of the subtractive selection algorithm. A 0.5 second target sound segment is shown on the left both as a waveform and as time-varying mel spectra. The three waveforms to the right show the selection of 1, 2 and 3 corpus units respectively. The target segment's mel-spectra are shown in successive stages of subtraction. Each additional corpus unit is selected to match the temporal and spectral characteristics of each residual spectrum. (Link to soundfile.)

describes performance techniques. Examining Orchidée yielded an initial infrastructure for grouping various sample collections into different families and providing a user interface for including and excluding "instruments."

Simple restrictions are then applied to the concatenative algorithm in order to limit results such that outputted scores are "playable" by an instrumentalist. Most important are time-varying restrictions on the number of segments per second (note speed) and the number of simultaneous segments (polyphonic capability). Also important are tools for limiting the database based to a maximum and minimum change in features for adjacent selected segments. For instance, limiting YIN pitch to within an octave of the last selected segment can help to mitigate problems of unplayable contours.

Depending upon the target sound, quantization routines are likely needed, though currently these are implemented post-concatenation in OpenMusic using the Kant library.

4. FUTURE

The following additional features are planned for future integration.

- Better methods for simultaneous selection. In particular, implementing something similar to Orchidée's more extensively recursive search methodology[3] in order provide an alternative to the current matching-pursuit algorithm.

- Create algorithms for classifying database units according to multidimensional clustering in feature-space. Of particular interest is the ability to augment tools for unit selection to include membership

to sound-categories rather than only measuring N-dimensional distances.

- Develop better tools for deploying the database as a time-varying resource. Currently the database is a fixed collection of possible sound segments, but it seems aesthetically interesting to provide the user with the ability to dynamically include/exclude portions of the database over the course of a concatenation.

4.1. Musical Writing

Continue to refine/augment AudioGuide's tools for computer assisted composition, including:

- Refine and improve the framework for restricting the availability of the corpus in time based upon physica/aesthetic instrumental models. While performative limitations are quite variable from instrument to instrument, high-level models for restricting speed and pitch contour should be applicable to a broad array of western acoustic instruments.

- Create a user interface for specifying hybrid concatenations such that AudioGuide's output can include a mixture of both instrumental parts/scores and sound segments for fixed media playback.

4.2. Video

Another area of interest is augmenting AudioGuide to handle both sound and video data. Preliminary efforts permit the user to assemble video segments based upon purely sonic concatenations. In the future, video could be more comprehensively incorporated into the programatic framework, including:

- Augment AudioGuide's feature analysis to include image feature extraction. Audio and video framerates' would need to be coordinated and segmentation strategies would need to operate jointly.

- Augment unit selection such that, if using a target video, database video segments may be retrieved based upon both sonic and visual features.

- Design a strategy for layering multiple video segments. With sonic concatenations, corpus units can be densely superimposed with a high degree of intelligibility. With video data the situation seems quite different – the superimposition of simultaneous video streams can quickly loose perceptual definition. Strategies need to be developed in order to address the differences in perceptual saturation between sound and vision.

5. REFERENCES

[1] J. Bullock and U. Conservatoire, "Libxtract: A lightweight library for audio feature extraction," in *Proceedings of the International Computer Music Conference*, 2007.

[2] A. De Cheveigne and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, p. 1917, 2002.

[3] E. S.-J. Grégoire Carpentier, Gérard Assayag, "Solving the Musical Orchestration Problem using Multiobjective Constrained Optimization with a Genetic Local Search Approach," *J Heuristics*, vol. 16, no. 5, pp. 681–714, 2010.

[4] N. Schnell, A. Röbel, D. Schwarz, G. Peeters, R. Borghesi *et al.*, "MuBu & Friends-Assembling Tools for Content Based Real-Time Interactive Audio Processing in Max/MSP," in *Proceedings of the International Computer Music Conference (ICMC), Montreal*. Citeseer, 2009.

[5] D. Schwarz, "Concatenative sound synthesis: The early years," *Journal of New Music Research*, vol. 35, no. 1, pp. 3–22, 2006.

[6] D. Schwarz *et al.*, "A system for data-driven concatenative sound synthesis," in *Digital Audio Effects (DAFx)*. Citeseer, 2000, pp. 97–102.