New Developments in Data-Driven Concatenative Sound Synthesis

Diemo Schwarz

Ircam – Centre Pompidou, Analysis–Synthesis Team, Paris, France
e-mail: schwarz@ircam.fr

Abstract

Concatenative data-driven synthesis methods based on a large database of sounds and a unit selection algorithm are gaining more interest in the computer music world. We briefly describe recent related work and then focus on new developments in our CATERPILLAR synthesis system: the advantages of the addition of a relational SQL database, work on segmentation by alignment, the reformulation and extension of the unit selection algorithm using a constraint satisfaction approach, and new applications for musical and speech synthesis.

1 Introduction

Data-driven concatenative sound synthesis methods are gaining more interest in the computer music community. In speech synthesis, they are widely used in multiple commercial and research systems, resulting in a considerable gain in quality of the synthesized speech.

Data-driven concatenative synthesis methods use a large database of source sounds, segmented into *units*, and a *unit selection* algorithm that finds the units that match best the sound or phrase to be synthesised, called the *target*. The units can be *non-uniform*, i.e. they can comprise a sound snippet, a note, up to a whole musical phrase. The selection is performed according to the descriptors of the units, which are characteristics extracted from the source sounds, or higher level descriptors attributed to them. The selected units are then transformed to fully match the target specification, and concatenated. However, if the database is sufficiently large, the probability is high that a matching unit will be found, so the need to apply transformations is reduced.

Usual sound synthesis methods are based on a model of the sound signal. It is very difficult to build a model that would realistically generate the fine details of the sound. On the contrary, concatenative synthesis, by using actual recordings, preserves entirely the fine details of sound. In the *data-driven approach*, instead of supplying rules constructed by careful thinking as in a *rule based approach*, the rules are induced from the data itself. Findings in other domains, e.g. speech recognition, corroborate the general superiority of data-driven approaches.

2 Related Work

Since the publication of (Schwarz 2000), data-driven concatenative synthesis methods have attracted more interest. Some projects are:

Plunderphonics (Oswald 1999) is John Oswald's artistic project consisting of songs made up from tens of thousands of snippets from a decade of pop songs, selected and assembled by hand.

Soundscapes (Hoskinson and Pai 2001) generates endless but never-repeating soundscapes for installations by splicing segments from a recording. It keeps the "texture" of the original sound file, avoiding discontinuities.

Soundmosaicing (Hazel 2001) constructs an approximation of one sound out of small pieces of other sounds using an unspecified match function without paying attention to concatenation quality.

Musical Mosaicing (Zils and Pachet 2001) performs a kind of automated remix of songs. It is limited to a sound database of pop music and uses only few descriptors, but includes concatenation quality.

La Légende des siècles is a theatre piece performed at the *Comédie Française* 2002, using real-time effects on the voice. One of these transformations uses a data-driven synthesis method inspired by our work: Prerecorded audio is analysed for energy and pitch. The FFT frames are stored in a dictionary and organised into clusters. During the performance, this dictionary is used with an inverse FFT and overlap-add to resynthesize sound with target pitch and energy given by the live audio input.

3 The CATERPILLAR System

The CATERPILLAR software system has been developed to perform data-driven concatenative unit selection sound synthesis. Its components are (figure 1):

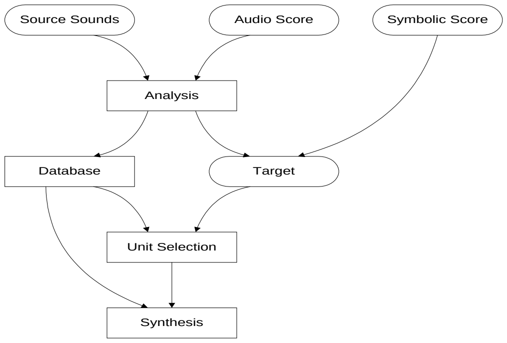

Figure 1: Overall structure of the data-driven CATERPILLAR system, arrows representing flow of data.

Analysis The source sound files are segmented into units (3.1) and analysed to express their characteristics with sound descriptors (3.2).

Database Source and data file references, units and descriptors are stored in a relational database (3.3).

Target The target specification is generated from a symbolic score (expressed in notes or descriptors), or analysed from an audio score (using the same segmentation and analysis methods as for the source sounds).

Selection Units are selected from the database according to given target descriptors and an acoustic distance function (3.4).

Synthesis is done by concatenation of selected units with a short cross-fade, possibly applying transformations.

3.1 Segmentation by Alignment

Before inclusion into the database, the source sounds have to be time-segmented into units. This can be done by blind segmentation or beat segmentation. However, to obtain well-segmented and labeled databases from instrument recordings, when a Midi score is available, a large amount of work has been devoted to segmentation by alignment of the audio signal with the score, described in (Orio and Schwarz 2001). We use a Dynamic Time-Warping algorithm, based on peak structure distance (PSD), which matches the expected harmonic partials with the observed ones. Note that this extends well to polyphonic music. Improvements of this technique for the alignment of multi-instrument music are described in (Soulez, Rodet, and Schwarz 2003), with which a sufficient precision of 23 ms is reached. Information contained in the score, e.g. note numbers or rest, the lyrics sung, or playing instructions, are attached to the units. This symbolic data can later be exploited for unit selection.

On each note segment, a *sub-segmentation* is performed that further subdivides it into an attack and a release *semi-note*. The release semi-note of a unit and the attack semi-note of the following unit together form a transition segment.

3.2 Descriptors

We distinguish three types of descriptors: *Category descriptors* are boolean and express the membership of a unit to a category or class and all its base classes in the hierarchy (e.g. *violin* → *strings* → *instrument* for the sound source hierarchy). *Static descriptors* are a constant value for a unit (e.g. Midi note number), and *dynamic descriptors* are analysis data evolving over the unit (e.g. fundamental frequency).

The descriptors used are given in the following (Rodet and Tisserand 2001). The perceptual salience of some of these descriptors depends on the sound base used.

Signal and Perceptual Descriptors Energy, fundamental frequency, zero crossing rate, loudness, sharpness, timbral width

Spectral Descriptors Spectral centroid, spectral tilt, spectral spread, spectral dissymmetry

Harmonic Descriptors Harmonic energy ratio, harmonic parity, tristimulus, harmonic deviation

Temporal Descriptors Attack and release time, ADSR envelope, center of gravity/antigravity

Source and Score Descriptors Instrument class and subclass, excitation style, Midi pitch, lyrics (text and phonemes), other score information[1]

All of these, except the symbolic source and score descriptors, are expressed as a vector of *characteristic values* that describe the evolution of the descriptor over the unit:

- arithmetic and geometric mean, standard deviation
- minimum, maximum, and range
- slope, giving the rough direction of the descriptor movement, and curvature (from 2$^{\text{nd}}$ order polynomial approximation[2])
- value and curve slope at start and end of the unit (to calculate the concatenation quality)
- the temporal center of gravity/antigravity, giving the location of the most important “elevation” or “depression” in the descriptor curve and the first 4 order temporal moments
- the normalised Fourier spectrum of the descriptor in 5 bands, and the first 4 order moments of the spectrum. This reveals if the descriptor has rapid or slow movement, or if it oscillates.

3.3 Database

The database holds references to the original sound files and to the data files from analysis. It stores the definitions of the descriptors, the units and their data.

As the quality of the synthesis grows with the size of the sound database, an efficient architecture is required. This is provided by using the open-source relational DBMS (database management system) *PostGreSQL*. Although a relational DBMS results in an overhead for data access, the advantages in data handling prevail:

Data Independence Using a relational database, only the logical relations of the data are specified in the *database schema*, not the physical organisation of the data. Access takes place using the declarative query language *SQL*, specifying *what* to do, not *how* to do it. This leads to unprecedented flexibility and openness to change.

[1] Any descriptor can be manually attributed to the units, be it subjective descriptors (say, “glassiness”), or other information not automatically derivable from the sound signal.

[2] We use Legendre polynomials, which have the desirable property that the lower-order polynomials are valid, albeit more coarse, approximations of the curve.

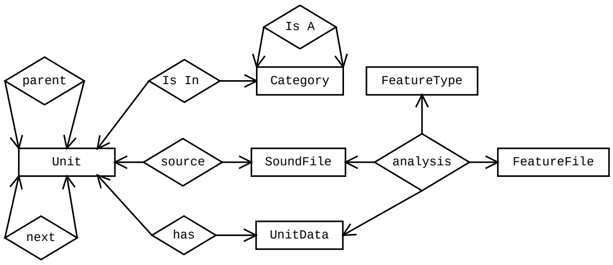

Figure 2: CATERPILLAR database schema showing tables (rectangles) and relationships (lozenges)

Consistency The consistency of the data is assured by the concept of atomic *transactions*, which are either completely executed, or rolled back to the previous state of the database, if an error occurred. This means no intermediate inconsistent state of the database is ever visible. Another safeguard are the automatic consistency checks according to the relations between database tables given in the schema. This is an enormous advantage while developing, because programming errors can't corrupt the database.

Client–Server Architecture Concurrent multi-user access over the network, locking, authentication, and fine-grained control over user's access permission are standard features of DBMS.

The database is clearly separated from the rest of the system by a *database interface*. Therefore, the DBMS used can be replaced by a different system, or other existing sound databases can be accessed, e.g. using the MPEG-7 indexing standard, or the results from the CUIDADO or ECRINS projects (Vinet, Herrera, and Pachet 2002) on sound content description. For low level access from *Matlab*, we wrote *mex*-extensions that send a query to the database and return the result in a matrix. The Sound Description Interchange Format (SDIF) (Wright et al. 1999) is used for well-defined interchange of data with external programs (analysis, segmentation, synthesis).

A simplified database schema in Entity–Relationship notation is given in figure 2. Descriptors are either `categories` or `featuretypes` for analysed dynamic/static descriptors. The system is open to handle all possible descriptors which can be added or changed dynamically. Category membership of a `unit` is expressed as a binary descriptor by the relationship `is in`. Because the database models a containment hierarchy `parent` of units, it is enough to add the highest parent unit (eventually the whole file) to a category. Because we also model an inheritance hierarchy `is a` among categories, this adds all the contained units to the category and all its base categories.

The database can be browsed with the graphical database explorer (figure 3), that allows users to visualize and play the units in the database.

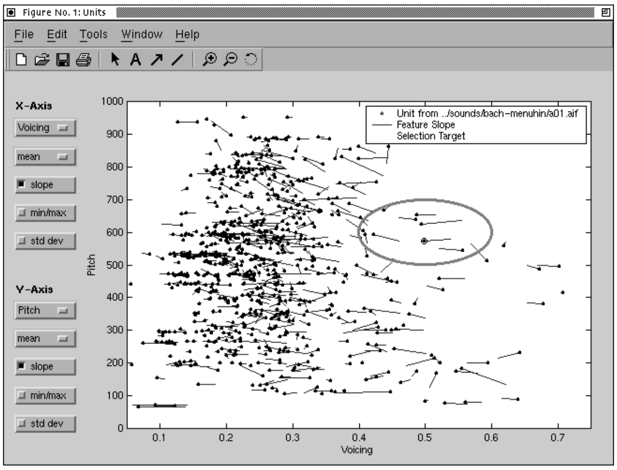

Figure 3: Database explorer feature view: Each point represents a unit, plotted according to two selectable characteristic values of two features. Various characteristic values can be displayed with the units, e.g. min/max, the standard deviation, or the mean slope (the short lines extending from the units). The ellipse serves to interactively select the units for real-time acoustic exploration of the database. The currently played unit within the ellipse is highlighted by a little circle.

3.4 Unit Selection

The classical unit selection algorithm finds the sequence of database units u_i that best matches the given synthesis target units t_τ using three costs: The **basic target cost** expresses the similarity of u_i to t_τ by a weighted sum of p feature costs: $C^t(u_i, t_\tau) = \sum_{k=1}^{p} w_k^t C_k^t(u_i, t_\tau)$

The **extended target cost** includes a context of r units around the target, weighted by w_j, in order to favour the selection of units out of similar contexts in the database and the target: $C^x(u_i, t_\tau) = \sum_{j=-r}^{r} w_j^x \, C^t(u_{i+j}, t_{\tau+j})$

The **concatenation cost** predicts the quality of the concatenation between two units u_i, u_j. It is given by a weighted sum of q feature concatenation cost functions: $C^c(u_i, u_j) = \sum_{k=1}^{q} w_k^c C_k^c(u_i, u_j)$

The concatenation cost depends on the unit type: concatenating an attack unit allows discontinuities in pitch and energy, a sustain unit does not. Consecutive units in the database have a concatenation cost of zero. Thus, if a whole phrase matching the target is present in the database, it will be selected in its entirety. The optimal sequence of units is found by a Viterbi algorithm as the least costly path through the network of database units.

The CATERPILLAR system uses Euclidean distance functions on the descriptor values, normalized by division by the standard deviation, with hand tuned weights.

Unit Selection by Constraint Solving Although the path-search unit selection algorithm shows good results, the algorithm is too rigid and it is hard to integrate other requirements, e.g. replacing just one displeasing unit in a sequence proposed by CATERPILLAR, or forcing all selected units to be different. We reformulated the selection algorithm as a constraint satisfaction problem (CSP) using the *adaptive local search* algorithm described in detail in (Codognet and Diaz 2001; Truchet, Assayag, and Codognet 2001). Satisfaction of a constraint is given by an error function, which allows to use the target and concatenation costs between units to express the constraints. Indeed, one can argue that path-search unit selection is a special case of adaptive local search unit selection.

4 Applications

High Level Instrument Synthesis Because the CATERPILLAR system is aware of the context of the database as well as the target units, it can create naturally sounding transitions. Information attributed to the source sounds can be exploited for unit selection, which allows high-level control of synthesis, where the fine details lacking in the target specification are filled in by the units in the database. As an example, we made a database from pieces for solo violin (J.S. Bach's *Sonata and Partita*, over one hour of music, played by different violinists). The basic unit is the *semi-note*, two of which are grouped to a transition segment (similar to a *diphone* in speech), allowing more transparent concatenation in the more stable middle part of the notes.

Free synthesis from heterogeneous sound databases offers a sound composer efficient control of the result by using perceptually meaningful descriptors. This type of synthesis is interactive and iterative. The CATERPILLAR system supports this by its graphical database browser and the ability to freeze good parts from a synthesis and regenerate others, or to force specific units to appear in the synthesis. Our database contains various recordings of environmental, instrumental, voice, and electronic sounds.

Resynthesis of audio with sounds from the database: A sound or phrase is taken as the audio score, which is resynthesized with the same pitch, amplitude, and timbre characteristics using units from the database.

Artistic Speech Synthesis An interesting new project uses CATERPILLAR to recreate the voice of a defunct eminent personality to render a given text. The goal here is different from fully automatic text-to-speech synthesis: highest speech quality is needed (concerning both sound and expressiveness), manual refinement is allowed. The role of CATERPILLAR is to give the highest possible automatic support for human decisions and synthesis control, and to select a number of well matching units in a very large base according to emotional and expressive descriptors. The adaptations that had to be applied to the CATERPILLAR system were:

- Addition of linguistic descriptors for prosody (word stress, intra-word position), and phonetics (phoneme class), which was easy, because the database is designed to allow extension of descriptors.
- Integration of new distance functions for the new descriptors into the unit selection algorithm
- Exploitation of the phoneme classes added to the database for faster unit search

5 Conclusions

The CATERPILLAR system shows good results in instrument synthesis, and surprising sounds from free synthesis, and will soon be tested with artistic speech synthesis. The concept of high-level synthesis is confirmed by these results, and also, for speech synthesis, by (Prudon and d'Alessandro 2001).

Integration of an SQL database into CATERPILLAR was a quantum leap forward for the ease and security of data handling. Finally, the use of a constraint satisfaction framework allows to combine the requirements for creative synthesis with the proven unit selection algorithm from speech synthesis, keeping maximum flexibility.

References

Codognet, P. and D. Diaz (2001). Yet another local search method for constraint solving. In *AAAI Symposium*, North Falmouth, Massachusetts.

Hazel, S. (2001). Soundmosaic. web page. `http://thalassocracy.org/soundmosaic`.

Hoskinson, R. and D. Pai (2001). Manipulation and resynthesis with natural grains. In *Proceedings of the ICMC*, Havana, Cuba.

Orio, N. and D. Schwarz (2001). Alignment of Monophonic and Polyphonic Music to a Score. In *Proc. ICMC*, Havana, Cuba.

Oswald, J. (1999). Plunderphonics. web page. `http://www.plunderphonics.com`.

Prudon, R. and C. d'Alessandro (2001). A selection/ concatenation TTS synthesis system: Databases developement, system design, comparative evaluation. In *4th Speech Synthesis Workshop*, Pitlochry, Scotland.

Rodet, X. and P. Tisserand (2001). ECRINS: Calcul des descripteurs bas niveaux. Technical report, Ircam.

Schwarz, D. (2000). A System for Data-Driven Concatenative Sound Synthesis. In *Digital Audio Effects (DAFx)*, Verona, Italy.

Soulez, F., X. Rodet, and D. Schwarz (2003). Improving Polyphonic and Poly-Instrumental Music to Score Alignment. In *ISMIR*.

Truchet, C., G. Assayag, and P. Codognet (2001). Visual and adaptive constraint programming in music. In *Proc. ICMC*, Havana, Cuba.

Vinet, H., P. Herrera, and F. Pachet (2002). The Cuidado Project: New Applications Based on Audio and Music Content Description. In *Proc. ICMC*, Gothenburg.

Wright, M. et al. (1999). Audio Applications of the Sound Description Interchange Format Standard. In *AES 107th convention.*

Zils, A. and F. Pachet (2001, December). Musical Mosaicing. In *Digital Audio Effects (DAFx)*, Limerick, Ireland.