PRINCIPLES AND APPLICATIONS OF INTERACTIVE CORPUS-BASED CONCATENATIVE SYNTHESIS

Diemo Schwarz
Ircam – Centre Pompidou
Paris

Roland Cahen
ENSCI
Paris

Sam Britton
Composer
London

ABSTRACT

Corpus-based concatenative synthesis plays grains from a large corpus of segmented and descriptor-analysed sounds according to proximity to a target position in the descriptor space. This can be seen as a content-based extension to granular synthesis providing direct access to specific sound characteristics. The interactive concatenative sound synthesis system CataRT that realises real-time corpus-based concatenative synthesis is implemented as a collection of *Max/MSP* patches using the FTM library. CataRT allows to explore the corpus interactively or via a written target score, to resynthesise an audio file or live input with the source sounds. We will show musical applications of pieces that explore the new concepts made possible by corpus-based concatenative synthesis.

1. INTRODUCTION

The recent concept of corpus-based concatenative sound synthesis [31] is beginning to find its way in musical composition and performance. It permits to create music by selecting snippets of a large database of pre-recorded sound by navigating through a space where each snippet takes up a place according to its sonic character, such as pitch, loudness, brilliance. This allows to explore a corpus of sounds interactively, or by composing this path, and to create novel harmonic, melodic and timbral structures.

The database of source sounds is segmented into short *units*, and a *unit selection* algorithm finds the sequence of units that match best the sound or phrase to be synthesised, called the *target*. The selection is performed according to the *descriptors* of the units, which are characteristics extracted from the source sounds, or higher level meta-data attributed to them. The selected units are then concatenated and played, after possibly some transformations.

These methods allow various applications, such as high level instrument synthesis, resynthesis of audio, also called *mosaicing*, texture and ambience synthesis, and interactive explorative synthesis in different variants, which is the main application of the CataRT synthesis system.

Explorative real-time synthesis from heterogeneous sound databases allows a sound composer to exploit the richness of detail of recorded sound while retaining efficient control of the acoustic result by using perceptually meaningful descriptors to specify a target in the multi-dimensional descriptor space. If the selection happens in real-time, this allows to browse and explore a corpus of sounds interactively.

Corpus-based concatenative synthesis allows new musical ideas to be experimented by the novel concepts it proposes of re-arranging, interaction with self-recorded sound, composition by navigation, cross-selection and interpolation, and corpus-based orchestration, which are introduced below [33]. These concepts will be expanded by giving concrete musical applications in four musical compositions and performances in section 5.4. The use of these concepts and conlusions that can be drawn will then be discussed in section 7.

Re-arranging (5.4.1) is at the very base of corpus-based concatenative synthesis: units from the corpus are re-arranged by other rules than the temporal order of their original recordings, such as given evolutions of sound characteristics, e.g. pitch and brilliance.

Interaction with self-recorded sound (5.4.2). By constituting a corpus, live- or prerecorded sound of a musician is available for interaction with a musical meaning beyond simple repetition of notes or phrases in delays or loops.

Composition by navigation (5.4.3) through heterogeneous sound databases allows to exploit the richness of detail of recorded sound while retaining efficient control of the acoustic result by using perceptually and musically meaningful descriptors to specify a target in the multi-dimensional descriptor space.

Cross-selection and interpolation (5.4.4). The selection target can be applied from a different corpus, or from live input, thus allowing to extract and apply certain sound characteristics from one corpus to another, and morphing between distinct sound corpora.

Corpus-based orchestration (5.4.5). By descriptor organisation and grouping possibilities of the corpora, a mass of sounds can be exploited while still retaining precise control over the sonic result, in order to insert it into a composition.

The CataRT software system [32] realises corpus-based concatenative synthesis in real-time, inheriting also

from granular synthesis while adding the possibility to play grains having specific acoustic characteristics, thus surpassing its limited selection possibilities, where the only control is position in one single sound file. CataRT is implemented as a collection of patches for *Max/MSP* [1] using the FTM, *Gabor*, and MnM extensions [2] [25, 26, 7]. CataRT is released as free open source software under the GNU general public license (GPL) [3] at http://imtr.ircam.fr.

After an overview of previous and related work in section 2, we present CataRT's underlying model in section 3. The object-oriented software architecture is detailed in section 4, followed by musical applications in section 5, and sections 6 and 7 will give an outlook of future work and a conclusion.

2. PREVIOUS AND RELATED WORK

Corpus-based concatenative sound synthesis methods are attracting more and more interest in the communities of researchers, composers, and musicians.

In the last few years, the number of research or development projects in concatenative synthesis grew rapidly, so that a description of each approach would go largely beyond the range of this article. For an in-depth survey comparing and classifying the many different approaches to concatenative synthesis that exist, the kind reader is referred to [30].

Corpus-based concatenative sound synthesis draws on many fields of research, mainly digital signal processing (analysis, synthesis, and matching), computer science (database technology), statistics and machine learning (classification), music information retrieval and modeling, and real-time interaction. In addition, there are many other topics of research that share methods or objectives, such as speech synthesis, singing voice synthesis, and content-based processing [18].

We could see concatenative synthesis as one of three variants of content-based retrieval, depending on what is queried and how it is used. When just one sound is queried, we are in the realm of descriptor- or similarity-based sound selection. Superposing retrieved sounds to satisfy a certain outcome is the topic of automatic orchestration tools. Finally, sequencing retrieved sound snippets is our topic of corpus-based concatenative synthesis synthesis.

2.1. Early Approaches

In fact, the idea of using a corpus of sounds for composing music dates back to the very beginning when recorded sound became available for manipulation with the invention of the first usable recording devices in the 1940's: the phonograph and, from 1950, the magnetic tape recorder [4, 5].

These historical approaches to musical composition use selection by hand with completely subjective manual analysis, starting with the *Musique Concrète* of the *Groupe de Recherche Musicale* (GRM) of Pierre Schaeffer, tried by Karlheinz Stockhausen in the notorious *Étude des 1000 collants* (study with one thousand pieces) of 1952 [24, 19] and applied rather nicely by John Cage to *Williams Mix* (1953) prescribing a corpus of about 600 recordings in 6 categories.

More recently, John Oswald's *Plunderphonics* pushed manual corpus-based concateative synthesis to its paroxysm in *Plexure*, made up from thousands of snippets from a decade of US Top40 songs [20, 11].

2.2. Caterpillar

Caterpillar, first proposed in [27, 28] and described in detail in [29], performs non real-time data-driven concatenative musical sound synthesis from large heterogeneous sound databases.

Units are segmented by automatic alignment of music with its score for instrument corpora, and by blind segmentation for free and re-synthesis. The descriptors are based on the MPEG-7 low-level descriptor set, plus descriptors derived from the score and the sound class. The low-level descriptors are condensed to unit descriptors by modeling of their temporal evolution over the unit (mean value, slope, spectrum, etc.) The database is implemented using the relational database management system *PostGreSQL* for added reliability and flexibility.

The unit selection algorithm is a Viterbi path-search algorithm, which finds the globally optimal sequence of database units that best match the given target units using two cost functions: The *target cost* expresses the similarity of a target unit to the database units by weighted Euclidean distance, including a context around the target. The *concatenation cost* predicts the quality of the join of two database units by join-point continuity of selected descriptors.

Corpora of violin sounds, environmental noises, and speech have been built and used for a variety of sound examples of high-level synthesis and resynthesis of audio [4].

The derived project *Talkapillar* [6] [5] adapted the *Caterpillar* system for artistic text-to-speech synthesis by adding specialised phonetic and phonologic descriptors.

2.3. Real-Time Corpus-based Synthesis

The many recent approaches to real-time interactive corpus-based concatenative synthesis, summarised in [30] and continuously updated, [6] fall into two large classes, depending on whether the match is descriptor- or spectrum-based: Descriptor-based real-time systems, such as CataRT [32], *MoSievius* [16], the commercial corpus-based

[1] http://www.cycling74.com
[2] http://ftm.ircam.fr/
[3] http://www.fsf.org
[4] http://www.ircam.fr/anasyn/schwarz/
[5] Examples can be heard on http://www.ircam.fr/anasyn/concat
[6] http://imtr.ircam.fr

intelligent sampler *Synful*[7] [17], or the interactive concatenative drum synthesiser *Ringomatic* [2], use a distance between descriptor vectors to select the best matching unit.

Spectrum-based systems, on the other hand, perform lookup of single or short sequences of FFT-frames by a spectral match with an input sound stream. Although they have interesting musical applications, e.g. the *SoundSpotter* [9] system[8] with the *Frank* live algorithm, or the audio-visual performance system *Scrambled Hacks,*[9] descriptor-based systems, seem to be more readily usable for music because the descriptors make sense of the sound database, by pushing the representation higher than the signal level, and thus allowing a compositional approach by writing a target score in terms of sound descriptors.

2.4. Content-Based Processing

Content-based processing is a new paradigm in digital audio processing that is based on symbolic or high-level manipulations of elements of a sound, rather than using signal processing alone [1]. Lindsay et al. [18] propose context-sensitive effects that are more aware of the structure of the sound than current systems by utilising content descriptions such as those enabled by MPEG-7. Jehan [15] works on object-segmentation and perception-based description of audio material and then performs manipulations of the audio in terms of its musical structure. The *Song Sampler* [3] is a system which automatically samples parts of a song, assigns it to the keys of a MIDI-keyboard to be played with by a user.

Along similar lines, the MusEd software [10] permits to browse through a song in 2D according to descriptors pitch and loudness. The segments are derived by an onset-detection algorithm.

2.5. Granular Synthesis

One source of inspiration of the present work is granular synthesis [22], which takes short snippets (*grains*) out of a sound file, at an arbitrary rate. These grains are played back with a possibly changed pitch, envelope, and volume. The position and length of the snippets are controlled interactively, allowing to scan through the soundfile, in any speed.

Granular synthesis is rudimentarily corpus-based, considering that there is no analysis, the unit size is determined arbitrarily, and the selection is limited to choosing the position in one single sound file. However, its concept of exploring a sound interactively, when combined with a pre-analysis of the data and thus enriched by a targeted selection, results in a precise control over the output sound characteristics, as realised in CataRT.

[7] http://www.synful.com
[8] http://www.soundspotter.org/
[9] http://www.popmodernism.org/scrambledhackz

3. MODEL

This section describes the model behind CataRT [32] that realises real-time corpus-based concatenative synthesis in a simple, intuitive, and interactive way. The next section will then dive into the details of the implemented architecture of the software system.

CataRT's model is a multi-dimensional space of descriptors, populated by the sound units. The user controls a target point in a lower-dimensional projection of that space with a selection radius around it, and the selection algorithm selects the units closest to the target or within the radius. The actual triggering of the unit is independent of the selection and can happen at any rate.

The selection is considering closeness in a geometric sense, i.e. on appropriately scaled dimensions: The generic distance measure is a Euclidean distance on the two chosen descriptors, normalised over the corpus, i.e. a Mahalanobis distance, in order to avoid distortions between different distances because of the different ranges of the values.

No concatenation quality is considered, for the moment, and the only transformations applied are a short crossfade to smooth the concatenation and pitch and loudness changes.

The following data-flow diagrams illustrate CataRT's model, boxes stand for data, circles for processes, and lozenges for real-time signals. Figure 1 shows an

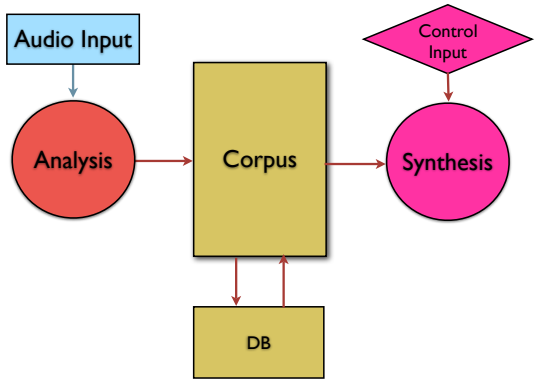

Figure 1. Overview of real-time corpus-based synthesis

overview of real-time corpus-based synthesis with the audio input feeding the corpus, that can be persistently saved and loaded to a database, and synthesis by retrieving data from the corpus.

The analysis part in figure 2 shows the different possibilities to get data into the corpus: either all data (audio, segment markers, raw descriptors) are loaded from pre-analysed files, or the descriptors are analysed in CataRT but the segment markers come from a file (or are arbitrarily chosen), or an additional onset detection stage segments the sound files. At this point, all analysis takes

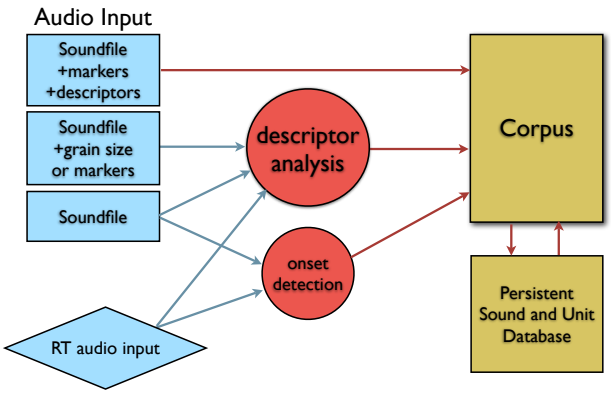

Figure 2. Analysis

place inside CataRT in real-time, which means that we could just as well use real-time audio input that is segmented into units and analysed on the fly, to feed the corpus. The audio could come, for example, from a musician on stage, the last several minutes of whose playing constitutes the corpus from which a laptop improviser selects units, as done in the live recording and interaction application in section 5.4.2.

Last, figure 3 shows the use of the corpus for synthesis by selection by user-controlled nearest neighbour search, with subsequent transformation and concatenation of the selected units.

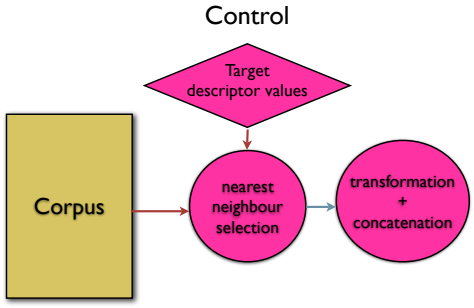

Figure 3. Synthesis

4. ARCHITECTURE

The object-oriented software architecture of the collection of CataRT *Max/MSP* patches is explained in the following, together with more details of the implementation of the model explained in the previous section. A class, in the sense of object-oriented programming and modeling, corresponds here to a *Max/MSP* abstraction (a subpatch that can be instantiated several times), and an abstract program interface is simply a convention of messages that can be sent to a subpatch which is implementing it.

CataRT's implementation makes heavy use of the extension libraries to *Max/MSP* FTM [10] [25] for advanced data structures and object system, *Gabor* [26] for arbitrary rate signal processing, and MnM for matrix processing and statistics [7].

4.1. Module Overview

The main modules of CataRT will be explained in the following paragraphs. Their names and cardinality are:

- one catart.data per corpus
- one catart.import for segmentation and analysis
- any number of catart.data.proxy to access data
- any number of catart.lcd for display and control
- any number of catart.selection per corpus
- any number of catart.synthesis~ per selection or per corpus

4.2. Analysis

The segmentation of the source sound files into units can come from external files or be calculated internally. There is also a mode where imported sound files are taken as a whole, which is appropriate for sets of drum and percussion sounds. Markers generated externally can be loaded from SDIF or ASCII files, or be imported from the marker chunks in the AIFF or WAV soundfiles themselves. Internal segmentation calculation is either by arbitrary grain segmentation, by split according to silence (given a threshold), or by the *yin* algorithm ported to the *Gabor* library [12]. In any case, the markers can be viewed and edited by hand, if necessary.

Descriptor analysis either uses precalculated MPEG-7 low-level descriptors or descriptors calculated in the patch. Details for the 230 imported MPEG-7 signal, perceptual, spectral, and harmonic descriptors can be found in [29], following the definitions from [23, 21].

The descriptors calculated in the patch in batch mode, i.e. faster than real-time, thanks to *Gabor*'s event-based signal frame processing, are the fundamental frequency, aperiodicity, and loudness found by the *yin* algorithm [12], and a number of spectral descriptors from [8]: spectral centroid, sharpness, spectral flatness, high frequency energy, mid frequency energy, high frequency content, first order autocorrelation coefficient (that expresses spectral tilt), and energy.

Note that also descriptors describing the units' segments themselves, such as the unit's unique id, the start and end time, its duration, and the soundfile it came from, are stored. Usually, this data is available directly in the data structures of catart.data, but, to make it available for selection, it is convenient to duplicate this information as descriptors.

[10] http://ftm.ircam.fr/

The time-varying raw descriptors at FFT-frame rate have to be condensed to a fixed number of scalar values to characterise a unit. These *characteristic values* [29] express the general evolution over time of a descriptor with its mean value, variance, slope, curve, min, max, and range, and allow to efficiently query and select units.

4.3. Data

Data is kept in the following FTM data structures: A table contains in its rows the descriptor definitions with the name and the specification where to find this descriptor in an SDIF file (the frame and matrix signatures and matrix indices). The loaded soundfiles are kept in a dictionary indexed by file name, containing metadata, a list of dictionaries for the data files, an FTM event sequence with the segmentation marks, and a vector containing the sound data. The unit descriptor data is kept in one big (N, D) matrix with one column per descriptor and one unit per row. Symbolic descriptors such as label or sound set name are stored as indices into tables containing the symbol strings.

Write access and the actual data storage is situated in catart.data, whereas read access to the data is provided by catart.data.proxy, which references an instance of the former, and which can be duplicated wherever data acess is needed.

For persistent storage of corpora, simple text files keep track of soundfiles, symbols, segments, and unit descriptor data. These can also be generated by Matlab, allowing any user-calculated descriptors to be imported. The Sound Description Interchange Format (SDIF) [11] is used for well-defined interchange of data with external analysis and segmentation programs.

4.4. Selection

Because of the real-time orientation of CataRT, we cannot use the globally optimal path-search style unit selection based on a Viterbi algorithm as in *Caterpillar*, neither do we consider concatenation quality, for the moment. Instead, the selection is based on finding the units closest to the current position x in the descriptor space, in a geometric sense, i.e. on appropriately scaled dimensions: A straightforward way of achieving this is to calculate the square Mahalanobis distance d between x and all units with

$$d = \frac{(x - \mu)^2}{\sigma} \tag{1}$$

where μ is the (N, D) matrix of unit data and σ the standard deviation of each descriptor over the corpus. Either the unit with minimal d is selected, or one randomly chosen from the set of units with $d < r^2$, when a selection radius r is specified, or, third, one from the set of the k closest units to the target.

[11] http://sdif.sourceforge.net

4.5. Synthesis

CataRT's standard synthesis component catart.synthesis~ is based on the *Gabor* library's event-based processing framework: A choosable short fade-in and fade-out is applied to the sound data of a selected unit, which is then pasted into the output delay-line buffer, possibly with a random delay. Other manipulations similar to a granular synthesis engine can be applied: the copied length of the sound data can be arbitrarily changed (de facto falsifying the selection criteria) to achieve granular-style effects or clouds of overlapping grains. Also, changes in pitch by resampling and loudness changes are possible. Note that, because the actual pitch and loudness values of a unit are known in its descriptors, it is possible to specify precise pitch and loudness values that are to be met by the transformation.

One variant, the catart.synthesis.multi module, permits to choose the number of output channels, and accepts amplitude coefficients for each channel for spatialisation. It outputs the scaled grains to a gbr.ola~ module for overlap-add synthesis to a multi-channel output signal.

However, these granular synthesis components are only one possible realisation of the synthesis interface catart.synthesis~. Other components might in the future store the sound data in spectral or additive sinusoidal representations for easier transformation and concatenation.

4.6. User Interface

The user interface for CataRT follows the model–view–controller (MVC) design principle, the model being the data, selection, and synthesis components, and the view and controller being defined by the two abstract program interfaces catart.display and catart.control. For many applications, these two are implemented by one single component, e.g. when the control takes place on the display, e.g. by moving the mouse.

Because displaying and navigating a high-dimensional space is not practical, the descriptor space is reduced to a 2-dimensional projection according to two selectable descriptors. The view (figure 4) plots this projection of the units in the descriptor space plus a 3rd descriptor being expressed on a colour scale. Note that that the display is dynamic, i.e. multiple views can be instantiated that can connect to the same data component, or one view can be switched between several data instances, i.e. corpora.

The implemented display uses *Max/MSP*'s graphic canvas (lcd, see figure 4) to plot a 2-dimensional projection of the units in the descriptor space plus a 3rddescriptor begin expressed on a color scale. Java-controlled display, or an OpenGL display via the *Jitter* [12] graphics library are possible.

In these displays, the mouse serves to move the target point in the descriptor space. Additional control possibilities are MIDI input from fader boxes to set more than two

[12] http://www.cycling74.com/products/jitter

descriptor target values and limit a selectable descriptor range, and advanced input devices for gestural control.

Independent of the current position, several modes for triggering playing of the currently closest unit exist: an obvious but quite interesting mode plays a unit whenever the mouse moves. De-facto, the friction of the mouse provides an appropriate force-feedback, so that this mode is called *bow*. To avoid the strident repetitions of units, the mode *fence* plays a unit whenever a different unit becomes the closest one (named in homage to clattering a stick along a garden fence). The *beat* mode triggers units regularly via a metronome, and the *chain* mode triggers a new unit whenever the previous unit has finished playing. Finally, the *continue* mode plays the unit following the last one in the original recording. There is also a mode *seq* which completely dissociates selection from triggering, which is performed by sending the message *play* to catart.select. This mode is for complete rhythmic control of the output, for instance via a sequencer.

CataRT incorporates a basic loop-sequencer that allows to automate the target descriptor control. Also the evolution of the weight for a descriptor can be sequenced, such that at the desired times, the target descriptor value is enforced, while at others the selection is less dependent on this descriptor.

5. APPLICATIONS

5.1. Explorative Granular Synthesis

The principal application of CataRT is the interactive explorative synthesis from a sound corpus, based on musically meaningful descriptors. Here, granular synthesis is extended by a targeted selection according to the content of the sound base. One could see this as abolishing the temporal dimension of a sound file, and navigating through it based on content alone.

Usually, the units group around several clusters. With corpora with mixed sources, such as train and other environmental noises, voice, and synthetic sounds, interesting overlaps in the descriptor space occur and can be exploited. Figure 4 shows the CataRT main patch with an example of a corpus to explore.

5.2. Audio-Controlled Synthesis

As all the analysis can take place in CataRT itself, it is possible to analyse an incoming audio signal for descriptors and use these to control the synthesis, effectively resynthesising a signal with sounds from the database.

The target descriptor evolution can also be derived from a sound file, by analysing and segmenting it and playing its controlling descriptors from the sequencer.

5.3. Data-Driven Drumbox

A slighly more rigid variation of the CataRT sequencer splits the target descriptor sequence into a fixed number of

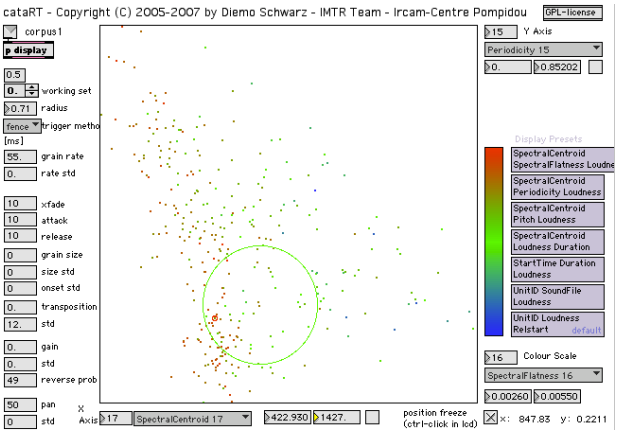

Figure 4. The graphic control interface of CataRT.

beats, where, for each beat, one sound class can be chosen. The selection within each soundclass, however, is gouverned by a freely editable descriptor curve, or real-time control.

5.4. Musical Applications

Of the various musical applications of corpus-based concatenative synthesis that were made since its inception, [13] we have chosen four that, as we reckon, best illustrate the fundamental concepts that are made possible by it, and where the electronic part is exclusively produced by CataRT. Other composers using CataRT in the sound design for the electronic part of their pieces are: Louis Naõn, Hans Tutschku, Matthew Burtner, Sebastien Roux, and Hector Parra.

5.4.1. Re-arranging

The concept of re-arranging the units of recorded sound is fundamental to corpus-based concatenative synthesis so that it is at the base of each of the four applications. It can be seen as abolishing the temporal order—time is just another descriptor amongst many that can serve to make new sense of recorded sound.

CataRT is used as a compositional and orchestration tool in the context of the piece *Junkspace* for banjo and electronics by Sam Britton, performed at Ircam October 14, 2006. The work takes large databases of recorded instrumental improvisations and uses concatenative synthesis to re-sequence and orchestrate these sequences. In this context, the concatenation process acts as a kind of oral catalyst, experimentally re-combining events into harmonic, melodic and timbral structures, simultaneously proposing novel combinations and evolutions of the source material, which might not have otherwise been attempted or acknowledged as viable possibilities.

[13] See [30] for a corpus-based re-reading of electronic music since 1950.

This concept is further pursued by his various musical projects [14] .

5.4.2. Live Recording and Interaction

Two performances by Diemo Schwarz and Sam Britton took place during the Live Algorithms for Music (LAM) conference 2006, [15] the first, to be released on CD, with the performers George Lewis on trombone and Evan Parker on saxophone, improvising with various computer systems. Here, CataRT's live recording capabilities were put to use to re-arrange the incoming live sound from the musician to engage him into an interaction with his own sound. The audio from the musician on stage, was recorded, segmented and analysed, keeping the last several minutes in a corpus from which the system selected units, the target being controlled via a faderbox.

Diemo Schwarz is currently pursuing this approach in a long-term collaboration with the improviser Etienne Brunet on bass clarinet. Please see http://www.myspace.com/theconcatenator for first results and appreciations of this work, and [34].

The second performance by Sam Britton and Diemo Schwarz, *Rien du tout*, draws on compositional models proposed by John Cage and Luc Ferrari. Through a process of re-composition it becomes possible to record environmental sounds and interpret and contextualise them into a musical framework. The performance starts with nothing at all (rien du tout) and by recording and re-composing environmental sound (here the sound of the concert hall and audience), evolves a musical structure by tracing a non-linear path through the increasing corpus of recorded sound and thereby orchestrating a counter-point to our own linear perception of time. The aim is to construct a compositional framework from any given source material that may be interpreted as being musical by virtue of the fact that its parts have been intelligently re-arranged according to specific sonic and temporal criteria.

5.4.3. Composition by Navigation

While navigation is also at the base of all the examples, the *Plumage* project [14] exploits it to the fullest, making it its central metaphor. *Plumage* was developed within the ENIGMES project (Exprimentation de Nouvelles Interfaces Gestuelles Musicales Et Sonores) headed by Roland Cahen: a collaborative experimental educational project at the national superior school of industrial creation ENSCI [16] , bringing together design students with researchers from LIMSI and Ircam. Its subject was "navigable scores" or score-instruments, in which different kinds of users would play the sound or music, cruising through the score.

In *Plumage*, CataRT was connected to and controlled by a 3D representation of the corpus, giving more expressive possibilities, more precision in the visualisation and interaction, and some new paradigms linked to 3D navigation. Yoan Ollivier and Benjamin Wulf imagined and designed this metaphor (*plumage* means the feathers of a bird) based on 3D modeled feather-like objects representing sound grains, allowing to place them in space, link them, apply surface colouring and texturing, rotation, etc., according to the sound descriptors of the grains they represent.

Figure 5. *Plumage*'s 3D space.

The navigation is not a simple movement of a cursor because the score cannot be played by a single instrument. It is rather comparable to an orchestral score, like a collection of sounds, which can be played by a group of instruments according to certain rules.

The score is not fixed but in an open form, rather like Earl Browns approach: "It will never come out the same, but the content will be the same." "Is it more interesting to fill a form or to form a filling?" In *Plumage*, both the composition of the score and the navigation can be set very precisely or not, and the setting of the navigation can become a way to fix a composition, to study a sound corpus or to navigate freely as an improvisation. Concatenative synthesis can be compared to deconstructing a picture into small points and a classification of these points according to the chosen descriptors. Imagine separating all the brush spots of an impressionist painting and reordering them according to hue, luminance, saturation or other descriptors. The resulting work will be an abstract painting. Navigating through such a score will not easily rebuild the original figure and will need special processes such as the one we developed to make this reshaping significant.

5.4.4. Cross-selection and Interpolation

Stefano Gervasoni's piece *Whisper Not* for viola and electronics, created in April 2007 in Monaco, played by Genevieve Strosser, computer music realization by Thomas Goepfer, explores the interaction of the musician with her own sound, segmented into notes and short

[14] http://icarus.nu
[15] http://www.livealgorithms.org
[16] http://www.ensci.com

phrases. Here, CataRT improvises a response to the musician as soon as she makes a pause, recombining her prerecorded sound according to a trajectory through the descriptor space, controlled via a faderbox.

Further on in the piece, the corpus of viola, with playing styles intended to create a resemblance to the human voice, is gradually interpolated with a second corpus of only pizzicato sounds, and then morphed into a third corpus of sounds of dripping water. Here, a new concept of corpus-based cross synthesis, or shorter *cross-selection* is applied: The descriptors of the selected response of CataRT are taken as the target for the parallel third corpus, such that the pizzicatos are gradually replaced by water drops, while retaining their timbral evolution.

5.4.5. Corpus-Based Orchestration

Dai Fujikura's piece *swarming essence* for orchestra and electronics, created in June 2007 with the orchestra of Radio France in Paris, computer music realization by Manuel Poletti, uses 10 different corpora of pre-recorded phrases of 5 instruments (alto flute, bass clarinet, trumpet, violin, cello), segmented into notes. The phrases making up the sound base were composed to match the harmonic content of the orchestral part of the 10 sections of the piece, and to exhibit a large sonic variety by use of different dampers and playing styles.

The composer then explored each corpus graphically, recomposing and manipulating the sound material using CataRT's granular processing capabilities. These trajectories were then transcribed into control envelopes for the concert patch (see figure 6). Each corpus was internally organised into sound sets by instrument, giving precise control of the orchestration of the electronic part by instrument-dependent routing, allowing their separate granularisation and spatialisation.

In this piece, the encounter of the composer with CataRT also inspired him to make the composition of the orchestral part follow sonic effects that were obtained by CataRT, in order to smoothly link both. For instance, the composer thought in terms of the "grain size" of the orchestra's playing.

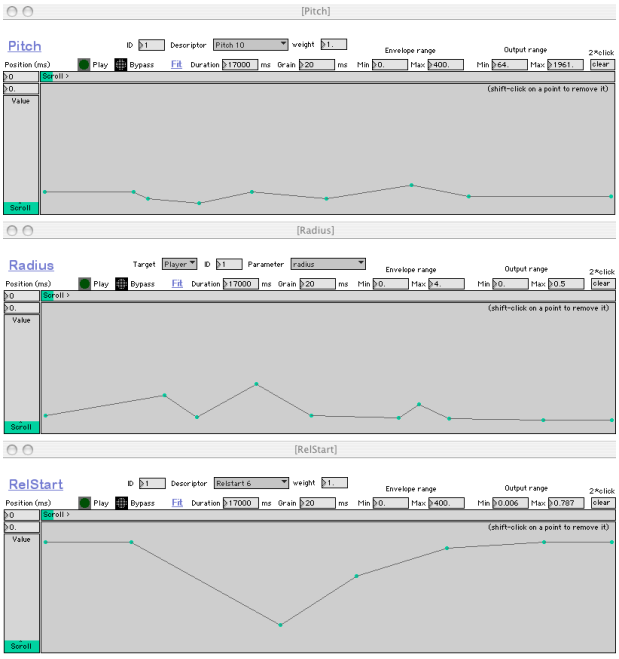

Figure 6. Graphic score for *swarming essence*.

6. FUTURE WORK

Interesting questions of representation, control and interaction are raised by the present work: To improve the efficiency of selection, and thus the scalability to very large sets of data (hundreds of thousands of units), the units in the descriptor space can be indexed by an optimised multi-dimensional k-nearest neighbour index. The algorithm described in [13] constructs a search tree by splitting up the descriptor space along the hyperplane perpendicular to the principal component vector, and thus achieving a maximal separation of units. This is then repeated for each sub-space until only a few units are left in each leaf node of the resulting tree. The k-nearest neighbour search can then, at each step down the tree, eliminate approximately half of the units, by just one distance calculation with the subspace boundary.

The used corpora are in general unevenly distributed over the descriptor space. Many units are concentrated in clusters, whereas large parts of the space are relatively empty. This is first a problem of interaction and visualisation, which should allow zooming into a cluster to navigate through the fine differences within. Clustering, rotation and warping of the descriptor space (by multidimensional scaling or magnifying-glass type transformations) maximises the efficiency of the interaction, leading to greater expressivity. However, the model of navigating through the descriptor space could be refined by a notion of subspaces with links to other subspaces. Note that, within different clusters, possibly different descriptors express best the intra-cluster variation such that each subspace should have its own projection to the most pertinent descriptors therein.

CataRT should take care of concatenation, at least in a limited way, by considering the transition from the previously selected unit to the next one, not finding the globally optimal sequence as in *Caterpillar*. The concatenation cost could be given by descriptor continuity constraints, spectral distance measures, or by a precalculated distance matrix, which would also allow distances to be applied to symbolic descriptors such as phoneme class. The concatenation distance could be derived from an analysis of the corpus:

It should be possible to exploit the data in the corpus to analyse the natural behaviour of an underlying instrument or sound generation process. By modeling the probabilities to go from one cluster of units to the next, we would

favour the typical articulations of the corpus, or, the synthesis left running freely would generate a sequence of units that recreates the texture of the source sounds.

To make more existing sound collections available to CataRT, an interface to the *Caterpillar* database, to the *freesound* repository, and other sound databases is planned. The *freesound* project,[17] is a collaboratively built up online database of samples under licensing terms less restrictive than the standard copyright, as provided by the *Creative Commons*[18] family of licenses. A transparent net access from CataRT to this sound database, with its 170 unit descriptors already calculated, would give us an endless supply of fresh sound material.

7. CONCLUSION

We presented corpus-based concatenative synthesis and its real-time implementation in the CataRT system, permitting a new model of interactive exploration of, and navigation through a sound corpus. The concatenative synthesis approach is a natural extension of granular synthesis, augmented by content-based selection and control, but keeping the richness of the source sounds.

We can see that three of the examples of musical applications in section 5 used the sound of the performers. For all three, the initial idea was to use the live sound to constitute a corpus from which CataRT would then synthesise an electronic accompaniment. In the end, however, Fujikura and Gervasoni chose to prerecord the corpus instead, because of the better predictability of the sonic content of the corpus, in terms of both quality and variety. In Britton and Schwarz's use of live corpus recording, the impredictability of the incoming material was either an integral part of the performance, as in *Rien du tout*, or inevitable, as with the LAM performance, because of the improvised nature of the music.

We see that precise knowledge of the corpus is a great advantage for its efficient exploitation. The 2D display of the descriptor space helps here, but can not convey the higher-dimensional shape and distribution of the space. We are currently exploring ways to represent the corpus by optimising its distribution, while still retaining access by musically and perceptually meaningful descriptors by dimensionality reduction.

CataRT is a tool that allows composers to amass a wealth of sounds, while still retaining precise control about its exploitation. From the great variety of musical results we presented we can conclude that CataRT is a sonically neutral and transparent tool, i.e. the software doesn't come with its typical sound that is imposed on the musician, but instead, the sound depends completely on the sonic base material and the control of selection, at least when the granular processing tools are used judiciously.

CataRT's modular architecture proved its usefulness for its inclusion in concert patches, being able to adapt to the ever changing context of computer music production.

[17] http://iua-freesound.upf.es

[18] http://creativecommons.org

The applications are only beginning to fathom all the possibilities this model allows for interaction modes and visualisation.

8. ACKNOWLEDGEMENTS

The authors would like to thank Alexis Baskind, Julien Bloit, and Greg Beller for their contributions to CataRT, Yoan Olliver, Benjamin Wulf (ENSCI), Christian Jacquemin, and Rami Ajaj (LIMSI) for their involvement in *Plumage*, Manuel Poletti, Dai Fujikura, and Stefano Gervasoni for their venture to use CataRT, and Norbert Schnell, Riccardo Borghesi, Fréderic Bevilacqua, and Rémy Muller from the Real-Time Music Interaction team for their FTM, Gabor, and MnM libraries without whom CataRT couldn't exist that way. CataRT is partially funded by the French National Agency of Research ANR within the RIAM project *Sample Orchestrator*.

We also would like to thank the anonymous reviewers of this article for their precise and constructive comments.

9. REFERENCES

[1] X. Amatriain, J. Bonada, A. Loscos, J. Arcos, and V. Verfaille. Content-based transformations. *Journal of New Music Research*, 32(1):95–114, 2003.

[2] Jean-Julien Aucouturier and Franois Pachet. Ringomatic: A Real-Time Interactive Drummer Using Constraint-Satisfaction and Drum Sound Descriptors. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, pages 412–419, London, UK, 2005.

[3] Jean-Julien Aucouturier, Franois Pachet, and Peter Hanappe. From sound sampling to song sampling. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, pages 1–8, Barcelona, Spain, October 2004.

[4] Marc Battier. *Laboratori*, volume I, pages 404–419. Einaudi, Milan, 2001.

[5] Marc Battier. *Laboratoires*, volume I, Musiques du XXe sicle, pages 558–574. Actes Sud, Cit de la musique, Paris, 2003.

[6] Grégory Beller, Diemo Schwarz, Thomas Hueber, and Xavier Rodet. A hybrid concatenative synthesis system on the intersection of music and speech. In *Journées d'Informatique Musicale (JIM)*, pages 41–45, MSH Paris Nord, St. Denis, France, June 2005.

[7] Frédéric Bevilacqua, Rémy Muller, and Norbert Schnell. MnM: a Max/MSP mapping toolbox. In *New Interfaces for Musical Expression*, pages 85–88, Vancouver, May 2005.

[8] Julien Bloit. Analyse temps réel de la voix pour le contrôle de synthèse audio. Master-2/SAR ATIAM, UPMC (Paris 6), Paris, 2005.

[9] Michael Casey. Acoustic lexemes for organizing internet audio. *Contemporary Music Review*, 24(6):489–508, December 2005.

[10] Graham Coleman. Mused: Navigating the personal sample library. In *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, Denmark, 2007.

[11] Chris Cutler. Plunderphonia. *Musicworks*, 60(Fall):6–19, 1994.

[12] Alain de Cheveigné and Nathalie Henrich. Fundamental Frequency Estimation of Musical Sounds. *Journal of the Acoustical Society of America (JASA)*, 111:2416, 2002.

[13] Wim D'haes, Dirk van Dyck, and Xavier Rodet. PCA-based branch and bound search algorithms for computing K nearest neighbors. *Pattern Recognition Letters*, 24(9–10):1437–1451, 2003.

[14] Christian Jacquemin, Rami Ajaj, Roland Cahen, Yoan Ollivier, and Diemo Schwarz. Plumage: Design d'une interface 3D pour le parcours d'échantillons sonores granularisés. In *IHM*, Paris, France, November 2007.

[15] Tristan Jehan. Event-Synchronous Music Analysis/Synthesis. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx)*, pages 361–366, Naples, Italy, October 2004.

[16] Ari Lazier and Perry Cook. MOSIEVIUS: Feature driven interactive audio mosaicing. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx)*, pages 312–317, London, UK, September 2003.

[17] Eric Lindemann. Music synthesis with reconstructive phrase modeling. *IEEE Signal Processing Magazine*, 24(1):80–91, March 2007.

[18] Adam T. Lindsay, Alan P. Parkes, and Rosemary A. Fitzgerald. Description-driven context-sensitive effects. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx)*, pages 350–353, London, UK, September 2003.

[19] Michael Manion. From Tape Loops to Midi: Karlheinz Stockhausen's Forty Years of Electronic Music. Online article, 1992. http://www.stockhausen.org/tape_loops.html.

[20] John Oswald. Plexure. CD, 1993. http://plunderphonics.com/xhtml/xdiscography.html\#plexure

[21] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification) in the Cuidado project. Technical Report version 1.0, Ircam – Centre Pompidou, Paris, France, April 2004.

[22] Curtis Roads. Introduction to granular synthesis. *Computer Music Journal*, 12(2):11–13, Summer 1988.

[23] Xavier Rodet and Patrice Tisserand. ECRINS: Calcul des descripteurs bas niveaux. Technical report, Ircam – Centre Pompidou, Paris, France, October 2001.

[24] Pierre Schaeffer. *Traité des objets musicaux*. Éditions du Seuil, Paris, France, 1st edition, 1966.

[25] Norbert Schnell, Ricardo Borghesi, Diemo Schwarz, Frederic Bevilacqua, and Remy Müller. FTM—Complex Data Structures for Max. In *Proceedings of the International Computer Music Conference (ICMC)*, Barcelona, Spain, September 2005.

[26] Norbert Schnell and Diemo Schwarz. Gabor, Multi-Representation Real-Time Analysis/Synthesis. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx)*, Madrid, Spain, September 2005.

[27] Diemo Schwarz. A System for Data-Driven Concatenative Sound Synthesis. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx)*, pages 97–102, Verona, Italy, December 2000.

[28] Diemo Schwarz. The CATERPILLAR System for Data-Driven Concatenative Sound Synthesis. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx)*, pages 135–140, London, UK, September 2003.

[29] Diemo Schwarz. *Data-Driven Concatenative Sound Synthesis*. Thèse de doctorat, Université Paris 6 – Pierre et Marie Curie, Paris, 2004.

[30] Diemo Schwarz. Concatenative sound synthesis: The early years. *Journal of New Music Research*, 35(1):3–22, March 2006. Special Issue on Audio Mosaicing.

[31] Diemo Schwarz. Corpus-based concatenative synthesis. *IEEE Signal Processing Magazine*, 24(2):92–104, March 2007. Special Section: Signal Processing for Sound Synthesis.

[32] Diemo Schwarz, Grégory Beller, Bruno Verbrugghe, and Sam Britton. Real-Time Corpus-Based Concatenative Synthesis with CataRT. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx)*, pages 279–282, Montreal, Canada, September 2006.

[33] Diemo Schwarz, Sam Britton, Roland Cahen, and Thomas Goepfer. Musical applications of real-time corpus-based concatenative synthesis. In *Proceedings of the International Computer Music Conference (ICMC)*, Copenhagen, Denmark, August 2007.

[34] Diemo Schwarz and Etienne Brunet. theconcatenator Placard XP edit. Leonardo Music Journal 18 CD track, 2008. To appear.