CORPUS-BASED TRANSCRIPTION AS AN APPROACH TO THE COMPOSITIONAL CONTROL OF TIMBRE

Aaron Einbond

Center for New Music and Audio Technologies (CNMAT)
Department of Music, University of California, Berkeley

Diemo Schwarz, Jean Bresson

Ircam–Centre Pompidou CNRS-STMS
1 Place Igor Stravinsky, 75004 Paris

ABSTRACT

Timbre space is a cognitive model useful to address the problem of structuring timbre in electronic music. The recent concept of corpus-based concatenative sound synthesis is proposed as an approach to timbral control in both real- and deferred-time applications. Using *CataRT* and related tools in the *FTM* and *Gabor* libraries for *Max/MSP* we describe a technique for real-time analysis of a live signal to pilot corpus-based synthesis, along with examples of compositional realizations in works for instruments, electronics, and sound installation. To extend this technique to computer-assisted composition for acoustic instruments, we develop tools using the Sound Description Interchange Format (SDIF) to export sonic descriptors to *OpenMusic* where they may be further manipulated and transcribed into an instrumental score. This presents a flexible technique for the compositional organization of noise-based instrumental sounds.

1. BACKGROUND

The manipulation of timbre as a structural musical element has been a challenge for composers for at least the last century. Pierre Boulez observes that compared to pitch or rhythm, "it is often difficult to find codified theories for dynamics or timbre" [1]. Trevor Wishart proposed that "pitch-free materials" could be organized based on their timbres and that computers would provide an invaluable resource for understanding the "topology" of timbre space [11]. Yet a decade later there is still a predominance of tools for organizing pitch and rhythm as compared to non-pitched materials.

Wishart's observations were informed by research in music perception that suggested timbre could be organized by listeners into a multi-dimensional spatial representation. Wessel and Grey both used multi-dimensional scaling to model listeners' perceptions of timbre into a 2- or 3-dimensional space [10, 6]. Momeni and Wessel have used such low-dimensional models to control computer synthesis based on spatial representations subjectively chosen by the user [7].

We propose an approach to structuring timbre that is based on perceptually-relevant descriptors and controllable in real-time. Using corpus-based concatenative synthesis (CBCS), a target sound is analyzed and matched to sounds in a pre-recorded database. While this technique can be used for more traditional sounds, it is especially effective for organizing non-pitched sounds based on their timbral characteristics. We present the implementation of this technique in *CataRT* and *OpenMusic* (*OM*) and its application in two recent compositions for instruments and electronics.

2. CORPUS-BASED CONCATENATIVE SYNTHESIS

The recent concept of corpus-based concatenative sound synthesis[8] makes it possible to create music by selecting snippets of a large database of pre-recorded sound by navigating through a space where each snippet is placed according to its sonic character in terms of *sound descriptors*, which are characteristics extracted from the source sounds such as pitch, loudness, and brilliance, or higher level metadata attributed to them. This allows one to explore a corpus of sounds interactively or by composing paths in the space, and to create novel harmonic, melodic, and timbral structures while always keeping the richness and nuances of the original sound.

The database of source sounds is segmented into short *units*, and a *unit selection* algorithm finds the sequence of units that best match the sound or phrase to be synthesised, called the *target*. The selected units are then concatenated and played, possibly after some transformations.

2.1. Real-Time Interactive CBCS

CBCS can be advantageously applied interactively using an immediate selection of a target given in real-time as is implemented in the *CataRT* system [9] for *Max/MSP* with the extension libraries *FTM* and *Gabor*,[1] making it possible to navigate through a two- or more-dimensional projection of the descriptor space of a sound corpus in real-time, effectively extending granular synthesis by content-based direct access to specific sound characteristics.

See Figure 1 for an example of *CataRT*'s sound browsing interface, where grains are played according to prox-

[1] http://imtr.ircam.fr/index.php/CataRT, http://ftm.ircam.fr

imity to the mouse- or controller-driven target position in a user-selected 2-descriptor plane.

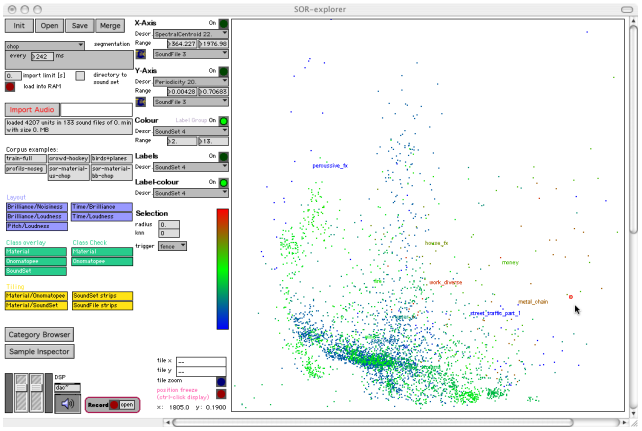

Figure 1. Screenshot of *CataRT*'s 2D navigation interface.

2.1.1. Segmentation and descriptor analysis

The segmentation of the source sound files into units can be imported from external files or calculated internally, either by arbitrary grain segmentation or by splitting according to silence or pitch change. Descriptors are either imported or calculated in the patch. The descriptors currently implemented are the fundamental frequency, periodicity, loudness, and a number of spectral descriptors: spectral centroid, sharpness, flatness, high- and mid-frequency energy, high-frequency content, first-order autocorrelation coefficient (expressing spectral tilt), and energy. For each segment, the mean value of each time-varying descriptor is stored in the corpus. Note that descriptors are also stored describing the unit segments themselves, such as each unit's unique id, its start time and duration, and the soundfile and group from which it originated.

2.1.2. Selection

CataRT's model is a multi-dimensional space of descriptors, populated by the sound units. They are selected by calculating the *target distance* C^t, which is a weighted Euclidean distance function that expresses the match between the target x and a database unit u_i

$$C^t(u_i,x) = \sum_{k=1}^{K} w_k^t C_k^t(u_i,x) \qquad (1)$$

based on the individual squared distance functions C_k^t for descriptor k between target descriptor value $x(k)$ and database descriptor value $u_i(k)$, normalised by the standard deviation of this descriptor over the corpus σ_k:

$$C_k^t(u_i,x) = \left(\frac{x(k)-u_i(k)}{\sigma_k}\right)^2 \qquad (2)$$

A weight w_k^t of zero means that descriptor k is not taken into account for selection.

Either the unit with minimal C^t is selected, or one is randomly chosen from the set of units with $C^t < r^2$ when a selection radius r is specified, or one is chosen from the set of the k closest units to the target.

2.2. Compositional Application

We use *CataRT* as a source for real-time electronic treatment of a live signal, as well as a resource in deferred time for computer-assisted composition. In the real-time case the target for synthesis is the live audio signal. In deferred time the target may be either an audio signal or an abstract trajectory in descriptor space which may be drawn with a mouse or tablet. While the two techniques share similar tools and mechanisms, they yield contrasting results.

3. REAL-TIME PERFORMANCE

3.1. Signal Analysis

To pilot *CataRT* synthesis with a live instrument, the audio signal is analyzed in real time with tools from the *Gabor* library according to the same descriptors and parameters used by *CataRT* for its analysis of pre-recorded corpora. The list of calculated audio descriptor values, ten currently but with more possible in future versions of *CataRT*, are sent to the selection module to output a unit. The relative weights of each of the descriptors used in the selection can be adjusted graphically. At this point the rich possibilities of *CataRT*'s triggering methods are available. We have had particularly attractive results with the *fence* mode, where a new unit is chosen whenever the unit closest to the target changes. The selected unit index is then sent to the synthesis module to output the result taking into account *CataRT*'s granular synthesis parameters. Otherwise the unit indices may be recorded to an SDIF file to be further processed (see below).

The rate at which units are synthesized is affected by the analysis window size, the triggering method, and the segmentation method of the corpus. Units need not be output in a regular rhythm: for example, in the *fence* mode not every new target window triggers selection. The analysis frames may be filtered by rate or descriptor value before being sent to *CataRT*. In particular, loudness and periodicity descriptors may be used to gate signal frames with values below desired thresholds.

3.2. Musical Realizations

3.2.1. Beside Oneself

CataRT has been used by several composers to synthesize fixed electronics, and has been used in improvised performance. However the first work to use it as a tool for pre-composed real-time treatment is Aaron Einbond's *Beside Oneself* for viola and live electronics, written as

part of the Cursus in music composition and technologies at IRCAM. Controlled by a *Max/MSP* patch incorporating *CataRT* and other *Gabor* and *FTM* objects, the computer analyzes sounds from the live violist and synthesizes matching grains from a corpus of close-miked viola samples. The goal is a smooth melding of live and recorded sound, a granular trail that follows the acoustic instrument. *CataRT* is also used indirectly in this work to stimulate resonant filters. Target pitches played by the viola are matched to a corpus containing samples of similar pitch. The synthesized result is then passed through resonance models, avoiding the possibility of feedback from the live signal.

3.2.2. What the Blind See

A similar technique of real-time analysis and synthesis can be used when the microphone is turned toward the public instead of an instrumentalist. In Einbond's interactive sound installation *What the Blind See*, presented as part of the exhibition "Notation: Kalkül und Form in den Künsten" at the Akademie der Künste, Berlin, the sounds of the public as well as the installation's outdoor setting are analyzed as a target for *CataRT* synthesis from a corpus of filtered field recordings of insects and plants.

4. CORPUS-BASED TRANSCRIPTION

Rather than using the synthesized audio output directly, *CataRT*'s analysis and selection algorithms of can be used as a tool for computer-assisted composition. In this case, a corpus of audio files is chosen corresponding to samples of a desired instrumentation. The *CataRT* selection algorithm is called on to match units from this corpus to a given target. Instead of triggering audio synthesis, the descriptors corresponding to the selected units and the times at which they are selected are stored and can be imported into a compositional environment such as *OM* where they can be converted symbolicaly into a notated score. The goal can be for the instrumentalist reading the score to approximate the target in live performance. The target used to pilot this process could be an audio file, analyzed as above, or it could be symbolic: an abstract gesture in descriptor space and time, designed by hand with a controller such as a tablet or mouse. This process is summarized in Figure 2.

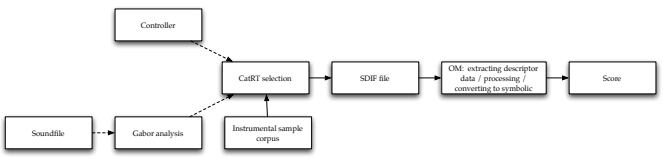

Figure 2. Flowchart for corpus-based transcription.

4.1. Exporting Data with SDIF

The results of the *CataRT* selection algorithm are recorded to an SDIF (Sound Description Interchange Format) file using a specially-created recording module. This file can be read by other programs such as *OM*, or by *Max/MSP* using *FTM* data structures and externals. SDIF is an established standard for the well-defined and extensible interchange of a variety of sound representations and descriptors [4, 12]. It consists of a basic data format framework and an extensible set of standard sound descriptions. This flexible and expandable format suggests a wide range of future applications for exporting audio descriptor data.

The SDIF representation of the selection data is as follows: first, three types of information about the corpus are written to the file header. The list of descriptor names for the data matrix columns are encoded as a custom matrix type definition. The list of sound files with their associated indices, referenced in the data matrices, is stored in a name–value table, as are the symbol lists for textual descriptors such as *SoundSet*, which can be assigned from the sound file folder name.

Then, for each selected unit, a row matrix containing all its descriptor data is written to a frame at the time of the selection since recording started. Both frame and matrix are of the custom extended type defined in the file header.

4.2. Processing Descriptor Data in OpenMusic

OpenMusic (*OM*) provides a library of functions and data structures for the processing of SDIF sound description data, which make it possible to link the results of the corpus-based analysis data to further compositional processing [2]. This integration in the computer-aided composition environment also allows the composer to take advantage of *OM* objects for displaying data lists in music notation.

Once imported, the descriptors of choice can be extracted and displayed along with their SDIF time stamps. The SDIF frames can be filtered by descriptor value: for example to exclude units with loudness below a given threshold. They can also be segmented into multiple streams, which can be useful to group units to be played by different instruments. Finally, `om-quantify` can be used to transcribe the frame time values into traditional rhythmic notation and display the results in a `voice` or `poly` object.

Some descriptors lend themselves to a more intuitive representation as MIDI notes than others: for example pitch or spectral centroid. However other descriptors can be transcribed as a more abstract representation of required information: for example sound file numbers can be assigned arbitrarily to MIDI note numbers, allowing the user to recall the sound file source of each unit. This could then be a useful tool for the user to interpret the results of the *CataRT* selection in a conventional score with verbal performance directions, articulation symbols, or noteheads. This stage of "manual transcription" could become better-automated

with the development of *OM* tools to display extra information on a score page, for example with the `sheet` object [3]. However, a final stage of subjective refinement may always be useful before the score is presented to an interpreter. Figure 3 shows an *OM* patch to transcribe a score with the `poly` object, which can then be exported in ETF format and subjectively-edited in *Finale*.[2]

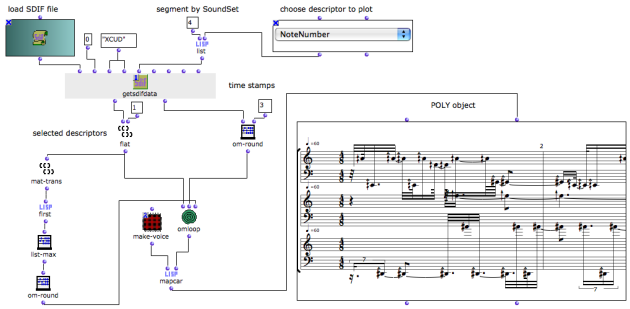

Figure 3. *OM* patch including raw score output.

5. DISCUSSION

5.1. Mapping Paradigms

The applications presented proceed from a direct mapping, where parameters of the target are associated with parameters of synthesis. However other mappings can be proposed both to correct for sources of noise, and to allow novel sources of compositional control. Mathematical transformation of a descriptor of an input signal could be used to normalize it to the range of that descriptor in the corpus. This could compensate, for example, for systematic offsets in loudness or other parameters. A more interesting mapping could be created to "transpose" or "invert" a target before mapping it to a corpus, through an appropriate translation or reflection in descriptor space. Wessel's research suggests that listeners could be sensitive to such a timbral transposition [10]. A further remove could be achieved by mapping one descriptor of the target analysis to a different descriptor in the corpus output. Rather than a transposition, this would be a kind of gestural "analogy."

5.2. Playability Constraints

In the transcription stage of our algorithm there are no constraints based in instrumental playability. Samples of live instruments are selected by the *CataRT* algorithm according to spectral characteristics alone, and then transcribed to notation in *OM*. It would be interesting to incorporate constraints based on speed, register, and playing technique in this process such that the final score would require less manual editing. An example of such constraints on pitch and rhythm are implemented in *OM* by Elvio Cipollone [5]; however a more sophisticated system would be necessary to incorporate extended playing techniques.

[2]ETF may be replaced by the more flexible MusicXML format as the bridge between future versions of *Finale* and *OM*.

6. ACKNOWLEDGEMENTS

This work is partially funded by the French National Agency of Research ANR within the project *SampleOrchestrator*. We thank Alexis Baskind, Eric Daubresse, John MacCallum, David Wessel, and Adrian Freed for their assistance implementing the software and their invaluable feedback on the manuscript.

7. REFERENCES

[1] P. Boulez, *On Music Today*. Cambridge, MA: Harvard University Press, 1971.

[2] J. Bresson and C. Agon, "SDIF sound description data representation and manipulation in computer assisted composition," in *Proc. ICMC*, Miami, USA, 2004.

[3] ——, "Scores, programs and time representation: The sheet object in openmusic," *Computer Music Journal*, vol. 32, no. 4, 2008.

[4] J. J. Burred, C. E. Cella, G. Peeters, A. Röbel, and D. Schwarz, "Using the SDIF sound description interchange format for audio features," in *ISMIR*, 2008.

[5] E. Cipollone, "CAC as maieutics: OM-Virtuoso and Concerto," in *OM Composer's Book*, vol. 2. Paris: Editions Delatour France / Ircam, 2008.

[6] J. M. Grey, "Multidimensional perceptual scaling of musical timbres," *J. Acoust. Soc. Am.*, vol. 61, 1977.

[7] A. Momeni and D. Wessel, "Characterizing and controlling musical material intuitively with geometric models," in *Proc. NIME*, Montreal, Canada, 2003.

[8] D. Schwarz, "Corpus-based concatenative synthesis," *IEEE Sig. Proc. Mag.*, vol. 24, no. 2, Mar. 2007.

[9] D. Schwarz, R. Cahen, and S. Britton, "Principles and applications of interactive corpus-based concatenative synthesis," in *JIM*, GMEA, Albi, France, Mar. 2008.

[10] D. Wessel, "Timbre space as a musical control structure," *Computer Music Journal*, vol. 3, no. 2, 1979.

[11] T. Wishart, *On Sonic Art*. London: Harwood Academic Publishers, 1996.

[12] M. Wright *et al.*, "Audio applications of the sound description interchange format standard," in *Audio Engineering Society 107th Convention*, New York, 1999.