A HYBRID CONCATENATIVE SYNTHESIS SYSTEM ON THE INTERSECTION OF MUSIC AND SPEECH

Grégory Beller, Diemo Schwarz, Thomas Hueber, Xavier Rodet
Ircam, Institut de Recherche et de Coordination Acoustique/Musique
1, place Igor Stravinsky
75004 Paris, France
{beller,schwarz,hueber,rodet}@ircam.fr

ABSTRACT

In this paper, we describe a concatenative synthesis system which was first designed for a realistic synthesis of melodic phrases. It has since been augmented to become an experimental TTS (Text-to-Speech) synthesizer. Today, it is able to realize hybrid synthesis involving speech segments and musical excerpts coming from any recording imported in its database. The system can also synthesize sentences with different voices, sentences with musical sounds, melodic phrases with speech segments and generate compositional material from specific intonation patterns using a prosodic pattern extractor.

1. INTRODUCTION

Musical concatenative synthesis consists in choosing the most appropriate sequence of sound units from a database and applying various modifications in order to build a desired melodic phrase. In previous work [11, 12] a musical concatenative system named CATERPILLAR has been elaborated. This work has been continued and an experimental TTS system, named TALKAPILLAR, is under construction. One of the aims of this system is to reconstruct the voice of a speaker, for instance a deceased eminent personality. TALKAPILLAR should pronounce texts as if they has been pronounced by the target specific speaker. The main difficulty consists in choosing the best speech segments, called units, to produce intelligible and natural speech with respect to the expressive characteristics of the speaker.

CATERPILLAR and TALKAPILLAR are based on the same architecture and thus allow to create hybrid synthetic phrases with speech units and any other sound units [2].

After a quick overview of related work, this article explains the system and the successive steps of source sound analysis, database, and synthesis. Then it proposes several applications which could lead to new musical experiments.

2. RELATED WORK

2.1. Speech synthesis

Research in music synthesis is influenced by research in speech synthesis to which considerable efforts have been devoted. Concatenative unit selection speech synthesis from large databases, also called corpus based synthesis [4], is now used in many TTS systems for waveform generation [7]. Its introduction resulted in a considerable gain in quality of the synthesized speech over rule-based parametric synthesis systems, in terms of naturalness and intelligibility. Unit selection algorithms attempt to find the most appropriate speech unit in the database (corpus) by using linguistic features computed from the text to synthesize. Selected units can be of any length (non-uniform unit selection), from sub-phonemes to whole phrases, and are not limited to diphones or triphones. Although concatenative sound synthesis is quite similar to concatenative speech synthesis and shares many concepts and methods, both have different goals (see [13] for more details).

2.2. Singing Voice Synthesis

Concatenative singing voice synthesis occupies an intermediate position between speech and sound synthesis, whereas the used methods are most often close to speech synthesis [5]. There is a notable exception [6], where an automatically constituted large unit database is used. See [9] for an up-to-date overview of current research in singing voice synthesis, which is out of the scope of this article.

3. GENERAL OVERVIEW

All the processes involved are presented and summarized in figure 1. They will be explained in more detail in the following sections.

4. SOUND SEGMENTATION

The first step is the segmentation of recorded files, speech or music, in variable length units. For this purpose, sound

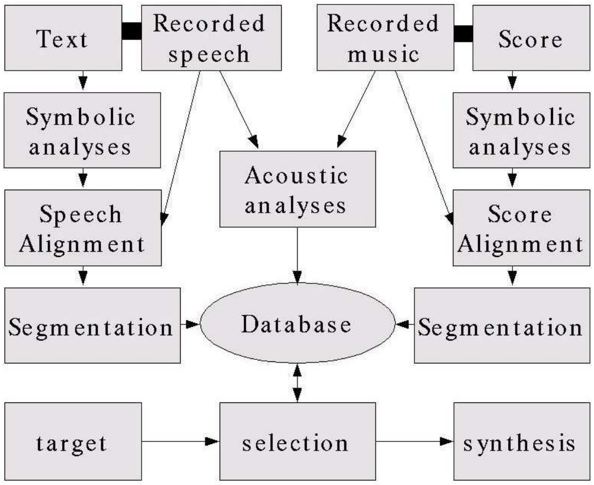

Figure 1. Architecture of the hybrid concatenative synthesis system.

files are first time-aligned with their symbolic representation (phonetic text or score) where unit frontiers are known. Then, units are imported into a POSTGRESQL database.

4.1. Score alignement

Score to audio alignment, or in brief score alignment, connects events in a score to corresponding points on the performance signal time axis [15]. A very similar task is known as score following, this term being reserved for the real-time case such as the one where a computer program is used to accompany a musician. Score alignment can thus use the whole score and the whole audio file if needed to perform the task, while score following specifies a maximum latency between an event in the audio stream and the decision to connect it to an event in the score. By using score information, score alignment permits to perform extensive audio indexing. It allows computing note time-onset, duration, loudness, pitch contour, descriptors and interpretation in general. Automatic score alignment has many applications among which we can mention: Audio segmentation into note samples for data base construction.

4.2. Speech alignement

A slightly different process is used to segment a speech signal. From the phonetic transcription of the sentence to align, a rudimentary synthesized sentence is built with diphones extracted from a small database where diphones frontiers have been placed by hand. Then the MFCC sequences of the two sentences are computed and aligned with a DTW algorithm. This provides the frontiers of phones and diphones in the sentence to align (see figure 2).

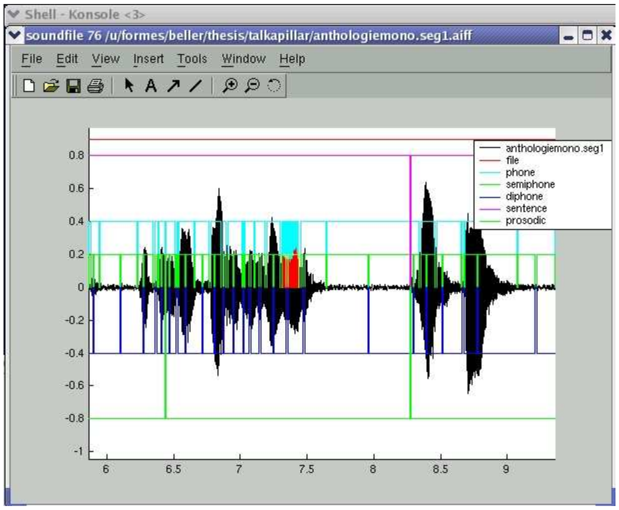

Figure 2. Example of speech segmentation due to alignment.

5. ACOUSTIC ANALYSIS

5.1. Musical 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* take 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 [10]. The perceptual salience of some of these descriptors depends on the sound base they are applied to.

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

All of these, except the symbolic source and score descriptors, are expressed as vectors 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 2nd order polynomial approximation)

- value and curve slope at start and end of the unit (to calculate a concatenation cost)

- 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 normalized 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.

5.2. Prosodic Features: YIN

Some acoustic features are extracted in order to build prosodic units of a specific speaker. Fundamental frequency is calculated with the YIN algorithm [3]. This also gives an energy and an descriptor of the aperiodicity for each frame. These data are averaged for each unit, sorted by phonetic classes and compared all together in order to build a distortion model relative to particular intonation strategies. We also compare the lasts of units, ones compared to the others per phonetic classes. This leads to an acoustic representation of prosodic units independent of the phonetic context.

6. SYMBOLIC ANALYSIS

The symbolic description of musical units is a midi score. The phonetic and syntactic description of texts is provided by the Euler program [1] issued from the European TTS project MBROLA [8]. This module analyzes a text and gives several symbolic representations such as a phonetic transcription, a grammatical analysis, a prediction of prosodic boundaries, etc. These descriptions are joined together in a MLC (Multi Layer Container) file (see figure 3). This file content and other symbolic descriptors corresponding to the relative places of the units are stored into an SDIF file (Sound Description Interchange Format, see [14]), which is the final description of segmental and supra-segmental units.

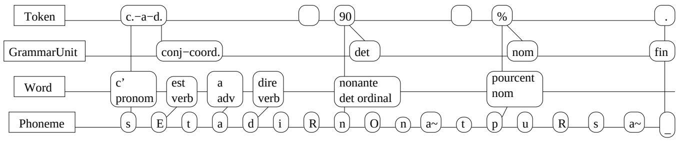

Figure 3. Example of an MLC structure [1] (from http://tcts.fpms.ac.be/synthesis/euler/doc/applicationdeveloper).

7. DATABASE

7.1. Database Management System

Since synthesis quality increases with the size of the sound database, an efficient database architecture is required. A relational DBMS (database management system) is used in this project. Although this results in a performance penalty for data access, the advantages in data handling prevail:

Data Independence
In a relational database, only the logical relations in the database schema are specified, not the physical organization of the data. It is accessed by the declarative query language SQL, specifying what to do, not how to do it, leading to unprecedented flexibility, scalability and openness to change.

Consistency
It is assured by atomic transactions. A transaction is a group of SQL statements 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. Other safeguards are the consistency checks according to the relations between database tables given by the schema, and the automatic re-establishment of referential integrity by cascading deletes (for example, deleting a sound file deletes all the units referring to it, and all their data). This is also an enormous advantage during development, because programming errors cannot 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. In our system, this allows, for instance, to run multiple processes simultaneously in order to increase computation speed.

7.2. Database Interface

The database is clearly separated from the rest of the system by a database interface, dbi, (see figure 4), written in Matlab and procedural SQL. Therefore, the current DBMS can be replaced by another one, or other existing sound databases can be accessed, e.g. using the MPEG-7 indexing standard, or the results from the CUIDADO [16] or ECRINS [10] projects on sound content description. The database can be browsed with a graphical database explorer called dbx that allows users to visualize and play the units in the database. For low-level access, we wrote C-extensions for Matlab that send a query to the database and return the result in a matrix. For convenience, the database interface also centralizes access to external sound and data files. For the latter, SDIF is used for well-defined exchange of data with external programs (analysis, segmentation, synthesis). Finally the free, open source, relational DBMS POSTGRESQL, is used to reliably store thousands of sound and data files, tens of thousands of units, their interrelationship and descriptor data.

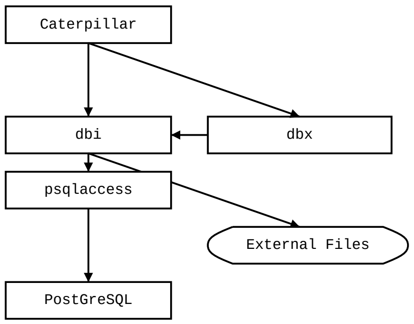

Figure 4. Architecture of the database interface.

8. UNIT SELECTION

8.1. Features Involved

After the importing step, the database is filled with plenty of heterogeneous units: diphones, prosodic groups, dinotes (from the middle of a note to the middle of the next), attacks, sustains. . . Each of them is described by a set of features involving symbolic and acoustic analyses. These features are the basis for comparison between units and unit selection. Adequacy of units (target cost) and concatenation cost are computed as a weighted combination of the features. Minimization of the global cost of a unit selection is done by a Viterbi algorithm

8.2. Unit Selection Algorithm

The classical unit selection algorithm finds the sequence of database units u_i that best match the given synthesis target units t_τ using two cost functions: The *target cost* expresses the similarity of u_i to t_τ including a context of r units around the target. The *concatenation cost* predicts the quality of the concatenation of u_i with a preceding unit u_{i-1}. The optimal sequence of units is found by a Viterbi algorithm finding the best path through the network of database units.

8.3. Prosodic Unit Selection

The first step of TALKAPILLAR is to select supra-segmental units in the database. By a Viterbi algorithm applied on the symbolic representation of prosodic units, we access to the best prosodic sequence able to traduce the natural expressivity of the speaker. Then we add the acoustic parameters of the chosen prosodic units to the segmental units of the target in the aim of select the best sequence of segmental units that fit to a real prosody excerpt from the database.

8.4. Segmental Units Selection

The second step consists in selecting the segmental units according to their symbolic representation and to the acoustic representation derived from the selection of prosodic units coming from the previous step. A similar Viterbi selection algorithm is then applied to find the best sequence of segmental units that match the target string and the prosody selected.

9. SYNTHESIS

9.1. Concatenation

When a correct sequence of units has been selected, it just needs to be concatenated in order to build the desired phrase. This is the last step of the synthesis process and could be aimed at two different goals. Concatenative synthesis has been designed to preserve all sound details so as to improve the quality and the naturalness of the result. In the TALKAPILLAR TTS system, a first strategy is to not transform chosen units. They are concatenated with a slight cross fade at the junction and a simple period alignment to not produce clicks and other artefacts.

9.2. Transformation: PSOLA

Another strategy consists in transforming some of the selected units before concatenating them. For instance, some of the units (the voiced ones) are slightly time-stretched and pitch-transposed to best match the prosody selected beforehand. These transformations are accomplished with a PSOLA analysis-synthesis algorithm [8].

10. APPLICATIONS

10.1. TTS synthesis

The TALKAPILLAR synthesis system is not aimed at a classical TTS use as is the case of the majority of similar TTS systems. It is mainly designed to (re)produce a specific expressivity. It offers an excellent framework for an artistic purpose since the user has access to all the steps of the process, which permits a full control of the result.

10.2. Musical Synthesis

From a Midi score or a target sound file, one can, for instance, generate the most similar musical phrase with the sequence of units extracted from another sound file. This means that the system is able, for example, to synthesize a realistic violin phrase containing the same melody as a recorded voiced line. It simply finds trough its multiple representation of the sound units, the best sequence that matches the given target.

10.3. Hybrid Synthesis

An interesting aspect of the system is that the TTS synthesizer and the musical synthesizer share the same framework. Created on the same software architecture, these two synthesis systems joined together give a powerful tool for composers interested in interaction between music and speech. For instance, voiced parts of a sentence could be replaced by cello's sustained units respecting the prosody of the replaced speech segments. The flexibility of the selection process's parametrization (features involved, cost weights, etc.) sets the user free to create very innovative hybrid synthetic phrases.

10.4. Prosody Extraction

An option of the system permits to extract prosodic units out of the database. Exportation of the acoustic features (f0, energy, flow, etc.) and of the symbolic descriptors (grammatical structure, type of the final accent, etc.) into an SDIF file allows to exploit these supra-segmental groups in other musical software such as OpenMusic, Diphone, or MAX/MSP.

11. CONCLUSION

In this paper, we have described a new concatenative synthesis system able to deal with speech and other sound material. The different steps of the process have been presented so as to provide a global comprehension of the system. The combination of two applications, TTS and musical phrases synthesis, in the same framework augments the capacities for artistic applications. Some examples can be downloaded from the following address: http://recherche.ircam.fr/anasyn/concat

12. REFERENCES

[1] Michel Bagein, Thierry Dutoit, Nawfal Tounsi, Fabrice Malfrère, Alain Ruelle, and Dominique Wynsberghe. Le projet EULER, Vers une synthèse de parole générique et multilingue. *Traitement automatique des langues*, 42(1), 2001.

[2] Grégory Beller. La musicalité de la voix parlée. Maitrise de musique, Université Paris 8, Paris, 2005.

[3] Alain de Cheveigné and Hideki Kawahara. YIN, a Fundamental Frequency Estimator for Speech and Music. *Journal of the Acoustical Society of America (JASA)*, 111:1917–1930, 2002.

[4] Andrew J. Hunt and Alan W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 373–376, Atlanta, GA, May 1996.

[5] M. W. Macon, L. Jensen-Link, J. Oliverio, M. Clements, and E. B. George. Concatenation-Based MIDI-to-Singing Voice Synthesis. In *103^{rd}Meeting of the AES*. New York, 1997.

[6] Yoram Meron. *High Quality Singing Synthesis Using the Selection-based Synthesis Scheme*. PhD thesis, University of Tokyo, 1999.

[7] Romain Prudon and Christophe d'Alessandro. A selection/concatenation TTS synthesis system: Databases developement, system design, comparative evaluation. In *4^{th} Speech Synthesis Workshop*, Pitlochry, Scotland, 2001.

[8] Prudon R., d'Alessandro C., and Boula de Mareüil. Prosody Synthesis by Unit Selection and Transplantation on Diphones. In *IEEE 2002 Workshop on speech synthesis*, Santa Monica, USA, 2002.

[9] Xavier Rodet. Synthesis and processing of the singing voice. In *Proceedings of the 1^{st}IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA)*, Leuven, Belgium, November 2002.

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

[11] 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.

[12] 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.

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

[14] Diemo Schwarz and Matthew Wright. Extensions and Applications of the SDIF Sound Description Interchange Format. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 481–484, Berlin, Germany, August 2000.

[15] Ferréol Soulez, Xavier Rodet, and Diemo Schwarz. Improving Polyphonic and Poly-Instrumental Music to Score Alignment. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, Baltimore, Maryland, USA, October 2003.

[16] Hugues Vinet, Perfecto Herrera, and François Pachet. The Cuidado Project: New Applications Based on Audio and Music Content Description. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 450–453, Gothenburg, Sweden, September 2002.