IMITATIVE AND GENERATIVE ORCHESTRATIONS USING PRE-ANALYSED SOUNDS DATABASES

Grégoire Carpentier, Damien Tardieu, Gérard Assayag, Xavier Rodet, Emmanuel Saint-James

IMITATIVE AND GENERATIVE ORCHESTRATIONS USING PRE-ANALYSED SOUNDS DATABASES

Grégoire Carpentier†, Damien Tardieu†, Gérard Assayag†, Xavier Rodet†, Emmanuel Saint-James‡
†IRCAM-CNRS - STMS
1 place Igor Stravinsky - Paris, France F-75004
{carpenti,dtardieu,assayag,rodet}@ircam.fr
‡LIP6
8 rue du Capitaine Scott - Paris, France F-75015
emmanuel.saint-james@lip6.fr

ABSTRACT

In this paper we will introduce a tool aimed at assisting composers in orchestration tasks. Thanks to this tool, composers can specify a target sound and replicate it with a given, pre-determined orchestra. This target sound, defined as a set of audio and symbolic features, can be constructed either by analysing a pre-recorded sound, or through a compositional process. We will then describe an orchestration procedure that uses large pre-analysed instrumental sound databases to offer composers a set of sound combinations. This procedure relies on a set of features that describe different aspects of the sound. Our purpose is not to build an exhaustive sound description, but rather to design a general framework which can be easily extended by adding new features when needed.

1. INTRODUCTION

In the last few decades contemporary music composers have widely experienced computer-aided composition (CAC) software in the development of their works. Originally, motivation for the design of such tools was to provide composers with the ability to easily manipulate musical symbolic objects, such as notes, chords, melodies, polyphonies... Simultaneously, another main branch in computer music research concentrated its efforts on sound analysis, sound synthesis, and sound processing, leading to a finer comprehension of many aspects of the wide sound phenomenon, and among them, the timbre of musical instruments.

In the meantime, contemporary composers have slowly started - since the beginning of the 1970s - to move away from purely combinatorial aspects of musical structures, and have drawn their attention to the spectral properties of sound. For instance, the French-born Spectral Movement focused on deriving compositional material from sound spectral analysis. Composers such as Gérard Grisey or Tristan Murail developed new music writing techniques relying on a correspondence between the spectral peaks of an analysed sound and the notes played by a set of instrumentalists. This turning point in western orchestral music set up a new aesthetic direction that has been carried on by later composers such as, among others, Lachenmann, Sciarrino, Ferneyhough, Dillon... Simultaneously, the parallel evolution of CAC made these pioneers and their successors dream of a composition tool that could cope with rich timbre information to help them in their orchestration tasks.

Today, the tremendous knowledge inherited from the analysis of instrumental sounds, the breakthrough in timbre research, the accessibility of large sound databases and the computer performance allow for the bridging of the gap between traditional CAC systems and the current potential of sound analysis and manipulation. In this paper we will introduce a new tool for computer-aided orchestration, with which composers can specify a target sound and replicate it with a given, pre-determined orchestra. Such a tool aims at providing composers with an alternative to traditional orchestration techniques, allowing them to explorate hidden directions in the instrumental timbre.

This paper is organized as follows. Section 2 reports on previous work in the field of computer-aided orchestration. Section 3 describes the global architecture of the presented system and the underlying paradigms and assumptions. Next, section 4 introduces the instruments sample databases used in our experiments and proposes a set of features for the sound description. In section 5 the concept of *target sound* is discussed, as well as different mechanisms for its construction. The orchestration procedure itself is detailed in section 6. Finally, section 7 will present the main conclusions and will discuss future work.

2. STATE OF THE ART

As far as we know there are at least three previous works that aim at building orchestration tools.

Rose and Hetrik [13] propose a tool that allows either the analysis of a given orchestration or the proposition of new orchestrations that approach a target sound. The orchestration proposition algorithm, which directly concerns us, uses a method based on Singular Value Decomposition to find the decomposition of a target spec-

trum as a weighted sum of instrument spectra. Spectra are calculated on 4096 points and averaged over time. This method guarantees that the resulting sum is the nearest in the least square sense. Such a method is interesting in terms of calculation cost. However the significance of a comparison on a thousands of points is questionable, in terms of perception but also of results comprehension and exploration. Moreover, it seems difficult with this method to specify that, for example, two sounds cannot be played simultaneously because they are played by the same instrument.

Psenicka [11] addresses this problem by performing the search on instruments, not directly on sounds. In a Lisp-based program called SPORCH (SPectral ORCHestration), the author uses an iterative matching on spectral peaks to find a combination of instruments that best fit the target sound. Each instrument in the database is indexed with a pitch range, a dynamic level range and a collection of the most prominent peaks at various pitches and dynamics. The program finds the peaks of the target and searches for the sound in the database that best matches those peaks. The rating is done by Euclidean distance on the peaks that are closed in frequency. Thus, a peak that does not belong to the target but does belong to the tested sound increases the distance. Then, the peaks of the best fit are subtracted from the target and the program iterates. The use of an iterative algorithm gives good calculation times, but nothing guarantees that the solution is the best. Moreover, it favors proposals with one sound (the first one) very similar to the target, and therefore discards solutions.

The third system is proposed by Hummel [5]. The principle is similar to Psenicka's, except that it works on spectral envelopes rather than on spectral peaks. The program first computes the target's spectral envelope, then iteratively finds the best approximation. Since it does not work on spectral peaks, the perceived pitch(es) of the result can be very different from the target pitch(es). Thus, according to Hummel, it works better for non-pitched sounds like whispered vowels.

3. PROJECT OVERVIEW

3.1. Terminology

A little terminology needs to be introduced before describing our system. We call *feature* a value or a set of values that describe one particular aspect of a sound or a combination of sounds. A feature can be either numeric (scalar,vector, matrix) or textual. A *target* is a set of features to be matched with a combination of sounds. A *candidate* is a sound that can potentially belong to the solution. And finally a *combination* is a set of candidates.

3.2. System overview

In our system (see figure 1) the user specifies the instruments he would like to use (constraints) and the characteristics of the sound to be produced (the target). This

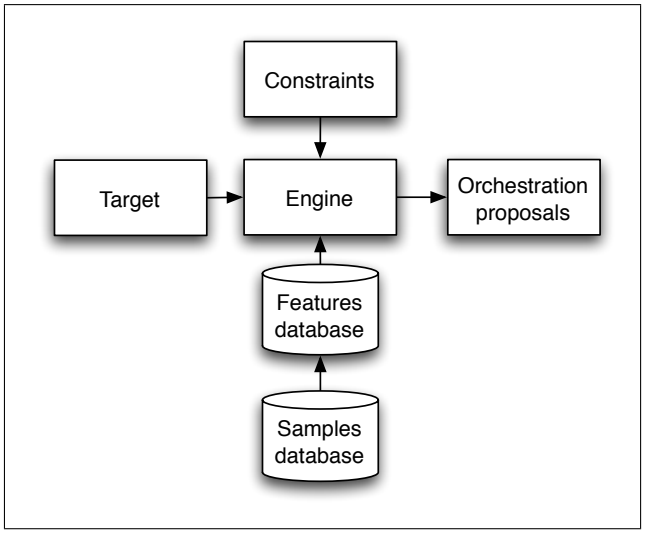

Figure 1. General architecture of our orchestration tool.

target is defined as a set of features (as described in section 4) coming either from a sound analysis or from a compositional process (see section 5). Then, an orchestration engine uses an instrumental knowledge database (features database) created by the analysis and structuring of large sound sample databases, to suggest instruments notes combinations (orchestration proposals) that "realize" the target. More precisely, the procedure searches for a combination whose features best match the target's features.

3.3. Paradigms and assumptions

The system we are presenting is based on two main paradigms. First, the set of parameters used to describe sounds and then to perform a similarity rating must be reasonably small and understandable to a composer. Secondly, we believe that the system must be as interactive as possible, giving the user full control of each step of the orchestration process. Both paradigms are closely related. A high level of interaction implies a good understanding of the sound description, which is a rather difficult task when facing a large number of features.

Within these paradigm definitions, we now need some additional assumptions to separate the wide orchestration problem into smaller ones. First, all the sounds (targets and sounds of the database) are assumed to be sustained, without temporal evolution. Secondly, the target is assumed to contain sinusoidal components, either harmonic or inharmonic. The underlying assumption is that the target does not contain any noise. Concerning the database, we assume its elements to be harmonic, representative of the instruments possibilities, and that the dynamic levels are well calibrated to reflect the real acoustic power of the instruments. Finally, our last assumption concerns room effect. The room's shape and materials, as well as sources and listeners positions may have a strong influence on the resulting sound of a given orchestra. We then neglect room's transfer function and the attenuation due to the

distance between instruments and listeners. All these assumptions seem rather strong and unrealistic, but do not significantly alter the quality of results at this stage of research.

4. DATABASES AND SOUND DESCRIPTION

4.1. Databases

The sound sample database is one of the most important components of our system. It must contain as many different instruments as possible and for each instrument, many different pitches, dynamics and playing styles. Moreover, the levels in the database must be well calibrated.

Up until now, we have experiemented with our system on two separate databases, each one affording specific characteristics. On one hand Ircam's Studio On Line database [2] references a rather small number of instruments, but many playing styles - including contemporary - that are not provided by usual sound databases. On the other hand, Vienna Symphonic Library [7] contains all the instruments of a traditional symphonic orchestra, but with fewer playing styles.

The sounds from these databases are analyzed to extract the features described hereafter and structured in another database (see figure 1).

4.2. Sound description

The orchestration procedure we are introducing relies on a set of features that describe different aspects of the sound. Our aim is less to build an exhaustive description of the sound phenomenon, than to propose a general framework which can be easily extended by adding new features when needed.

Audio features are the purpose of much research and publications, for a review see [10]. They are usually defined by : (i) the source they are calculated on (like waveform or spectrum) ; (ii) a computation method ; (iii) possibly a comparison function, i.e. a method that, given features of two sounds, quantifies the similarity between the sounds. In our context, features also need an addition method, which, given the features of two sounds, returns the features of the combination. Hereafter we will describe the features we are currently using. This set will be extended in the future as we will describe different aspects of the sound in the orchestration process, such as attack transients, roughness, noise or time variations. Currently, we will limit ourselves to the spectral description of sounds.

4.2.1. Pitch and fundamental frequency

Each sound in the database is harmonic, thus has pitch. We obtain this information from the name of the samples, and check this value with a fundamental frequency (F0) estimation using [4]. A different method to find which pitches could explain the target will be described in section 5.3.

4.2.2. Spectral centroid

The spectral centroid is calculated on the perceptually weighted power spectrum as described in [9]. The spectral centroid of a combination correspond to the mean of the components sounds centroids weighted by their loudness.

The spctral centroid is used to describe the general shape of the spectral envelope of the sound. It is well known in the timbre description field since it appears as the second dimension of most instrumental sounds timbre spaces.

4.2.3. Harmonic spectrum

Since the sample database contains only sustained harmonic sounds, calculation of the harmonic spectrum is simple. We analyze the signal with Pm2, a sinusoidal harmonic analysis software [12] that returns index, frequency, amplitude and phase of all the harmonics of the sound. This method is used for the database sounds only, and not for the target, which can be inharmonic.

We assume that adding two sounds spectra means adding the power of their corresponding harmonics. For instance when adding C2- and C3-pitched sound spectra, the power of C3's harmonics will be added to C2's even harmonics (see figure 4 and section 6.3).

4.2.4. Inharmonic spectrum and most important partials

Concerning the target, since it can be inharmonic, we use an inharmonic sinusoidal analysis also provided by Pm2. Then, to reduce the number of partials used in the final comparison, we select what we call the *most important partials*. We consider that a partial is perceptually relevant either if it is resolved by the auditory system or if it is particularly strong.

4.2.5. Time averaging

All the previous features are computed frame by frame, thus we get their evolution over time. Since we assume the sounds to be static, without any temporal evolution, we average features by weighting the value of each frame by its loudness.

5. TARGET DEFINITION

As previously stated, a *target* can be defined without a loss of generality as *a set of features*. These features can be low-level, audio features (any spectral and temporal data) as well as mid-level, compositional features (for instance : a set of pitches, a set of constraints on the resulting orchestrations). With such a distinction, a clear difference is made between two composers' questions : *What should it sound like ?* and *How can I do it with an orchestra ?* The former is addressed by low-level features, the latter by mid-level ones. Note that this approach does not necessarily imply that the composer knows in advance how

a given target should be realized. Indeed mid-level features can be suggested to the composer by post-processing treatments on low-level ones. For instance, in the current version of our system the composer has the ability to find different sets of pitches that best explain the most important partials of the target (see section 5.3). Moreover, our belief is that when not using a target directly issued from a recorded sound, a composer does not necessarily have in mind precise values of audio features, but rather interacts with more complex musical structures (pitches, chords, polyphonies) and should therefore be able to enter the orchestration problem through this medium.

Thus, considering a target as a set of features is a large enough definition to cope with the two main use cases highlighted by the composers involved in this project. In the first scenario the orchestration tool is requested to find a combination of sounds that bear the greatest resemblance with a given, pre-recorded sound. This scenario was called *imitative orchestration*. In the second case (called *generative* - or *experimental* - *orchestration*, the tool has to find a combination whose associated set of features is as close as possible as a user-defined target description. Of course, our target definition makes the former a sub-case of the latter.

Up until now we have used in our system elementary spectral data as low-level features : the most important partials in a given sound, and the first spectral moment, the spectral centroid. As mid-level features we use a set of fundamental frequencies that best match the most important partials (see section 5.3). Note that these frequencies are not given *a priori*, as it is the case for the instrument sounds in the sample database. Indeed, a given instrument sound usually correspond to a unique pitch (multiphonics excepted), referenced by the sound sample name in the database, whereas a target might be made up of several pitches to be determined prior to the orchestration process itself. This is the reason why the target's F0 set is the output of a user-controlled post-processing on the most important partials. During the orchestration process, these fundamental frequencies will be mapped on instruments pitches to discard a large number of candidates (see section 6.1).

5.1. Sound Analysis

Our first approach consisted in trying to replicate with an orchestra a given, pre-recorded sound. This paradigm was chosen to temporarily leave out of scope the problem of defining a target ex nihilo, in other words translating a musical idea into a set of features. Moreover it was a comfortable way to test the accuracy of the adopted sound description. In this approach the features are calculated as described in section 4.

5.2. Importing sound features from OpenMusic and AudioSculpt

In the *imitative orchestration* scenario, skipping the sound analysis step is possible by directly importing a partial set from a SDIF file [14]. Therefore the target's spectral definition can be preliminarily performed in an appropriate sound analysis environment, such as AudioSculpt [3].

Importing spectral data from a simple text file is also possible, letting the composer build a set of partials with a computer-aided composition system. A simple patch was designed in OpenMusic [1] to map the pitches and velocities values of a `chord` object into a set of partials and velocities, and export this data to our orchestration tool. Through this mechanism the target's partial set can not only be seen as the output of a spectral analysis, but also as the result of a calculus, which is the very purpose of OpenMusic. Such a communication with OpenMusic should be seen as the first step towards *generative orchestration* scenarios, where no pre-recorded sound file is available (see upper).

5.3. Extraction of relevant F0s

Assuming the target's low-level features have been defined, we now need to associate with the given partial set, a set of fundamental frequencies which "own" these partials in their harmonics. In other words we are looking for harmonic series that match the most important partial list. As introduced earlier, this step is crucial for discarding a large set of candidates (see section 6.1) in the orchestration process itself.

5.3.1. Multiple F0 extraction

The literature is full of multiple fundamental frequencies extraction algorithms, all using a large set of techniques. Klapuri [6] uses bandwise processing with a spectral smoothness criterion, while Yeh and al. [15] use spectral smoothness and partials synchronicity criterions, and Maher and Beauchamp [8] use a two-way mismatch (TWM) procedure. What makes our problem specific here is firstly that we are not working on the sound signal itself, but on a pre-extracted set of low level features. Among these, only the most important partials (usually from 10 to 15) can be used as input data for an F0-extraction method. Secondly, we are not looking for an exact transcription of *all* the pitches eventually present in the target, but for F0s that somehow "explain" its most important partials. For the above reasons we have implemented our own algorithm, called `melvin`, based on a method similar to the TWM procedure.

Roughly speaking, `melvin` looks for a set of fundamental frequencies whose associated harmonic series best match an input set of partials. A given partial p will be assigned to a fundamental frequency f if and only if it satisfies the following condition :

$$\exists N \in \mathbb{N}, \frac{N}{1+\delta} \leq \frac{p}{f} \leq N(1+\delta) \qquad (1)$$

where δ is the inharmonicity tolerance, whose typical range is $[0.0005; 0.05]$.

Contrary to the TWM procedure, `melvin` does not try every F0 candidate, but rather considers as a potential

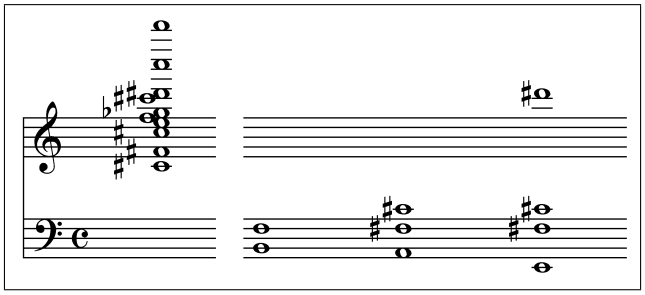

Figure 2. Different set of F0s for a partial set obtained from the analysis of a tam-tam sound. **Left.** The partial set in common music notation. **Right.** 3 F0s sets obtained with different values of f_{min} and δ.

F0 every element in the partial set, plus a set of possible missing (or hidden) fundamentals, preliminarily computed. The missing fundamental discovery is based on the assumption that whenever three consecutive partials of a given frequency f are met in the input partial set, then f might be a potential hidden F0. Mathematically speaking, given three partials p_i, p_j and p_k (with $p_i < p_j < p_k$) in the partial set and $f = (p_k - p_i)/2$, then f is a potential hidden F0 if and only if $\exists N \in \mathbb{N}$,

$$\begin{cases} N/(1+\delta) \leq & p_i/f \leq & N(1+\delta) \\ (N+1)/(1+\delta) \leq & p_j/f \leq & (N+1)(1+\delta) \\ (N+2)/(1+\delta) \leq & p_k/f \leq & (N+2)(1+\delta) \end{cases} \quad (2)$$

5.3.2. User control

Our belief is that in an orchestration context the composer might be interested in finding more than one set of F0s whose harmonics match the target's partial set. With this aim, `melvin` uses two control parameters : the inharmonicity tolerance δ, and a minimum frequency f_{min} under which no F0 can be found. Figure 2 shows different F0s outputs for the same partial set but with different f_{min} and δ values. Of course, the composer has afterwards the possibility to manually add or remove F0s at will.

5.4. Features-controlled synthesis

The sound description paradigm implies a necessary separation between a sound object and its associated set of features. In a *generative orchestration* scenario, the composer might have a strong interest in "listening to" what a target defined as a collection of user-set features "sounds like", before running the orchestration engine itself. With such a functionality, a target can be refined until it sounds satisfying to the composer's ears, preventing therefore surprising results due to a "blind" target definition. In an *imitative orchestration* context, re-synthesizing a sound target from its audio features is also felt to be of paramount importance : through this mechanism, the composer is aware of the how the features "hear" the sound itself.

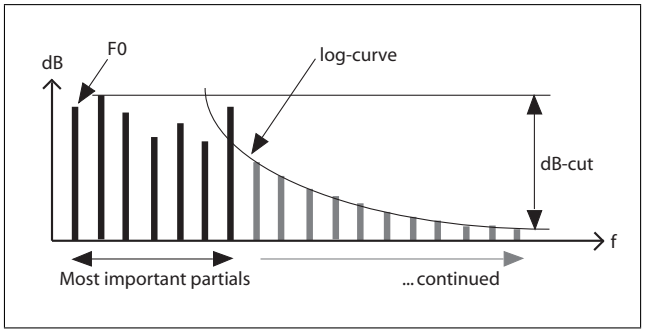

Figure 3. "Reconstruction" of the target's spectrum from its most important partials and F0s features. The log-curve and dB-cut parameters are determined through an optimization process in such a way that the spectral moments of the synthesized sound equal the original target's ones.

With this aim, our system's target definition module includes two basic synthesis features. The first one is a simple additive synthesis of the most important partials, whose resulting signal can be modulated by - if available - the original sound's temporal envelope. This option is particularly useful to estimate how much the sound's temporal evolution influences the timbre perception. In the second one synthesis parameters are first pre-processed in such a way that the spectral moments of the synthesized sound equal the original target's ones. In other words, the harmonic series found by the F0s extractor (see section 5.3) in the target's most important partials are "continued" up to half the sampling rate, on the basis of the underlying F0 (see figure 3). The shape of the spectral envelope is controlled by two parameters, the dB-cut and the log-curve, which are determined through an optimization process. When several harmonic series are considered, we use the same dB-cut and log-curve for all.

Compared to today's most accurate synthesis techniques, the quality of the synthesized sound is quite poor. Remember that, as stated in section 3.3, our system was designed with the idea of a small set of sound features in mind. The more condensed the sound description, the poorest the synthesis. However, note that we are not aiming here at an accurate re-synthesis of a pre-analysed sound. Our intention is rather to show which part of the sound's spectral features are "captured" by a static description.

6. ORCHESTRATION ENGINE

The orchestration engine operates in four consecutive steps. First the candidates are selected, then the combinations are created and selected, their features are computed and finally the combinations are sorted in terms of target fitting.

6.1. Candidates selection

Candidates selection is performed by filtering the elements of the database according to a pitch criterion. Basically, any element whose pitch does not belong to a pre-

calculated set is discarded. This set is built by completing `melvin`'s output. Any pitch in upper harmonic relation with any element of the `melvin`'s output can be added. For instance, assume the `melvin`'s output is {A3, B3}, the extended set would then be {A3, B3, A4, B4, E5, F#5, A5, B5, C#6, D#6 ...}. Ideally, we should try all these possibilities, but the cost would be prohibitive. Thus, we make a selection of the most relevant pitches, either by truncating the extended set or by keeping only the elements that correspond to the strongest partials in the target.

6.2. Combination selection

Once the candidates have been selected, we construct possible combinations. Obviously, a combination cannot contain more notes of one single instrument than it can actually play.

Additionally, each time a candidate is to be added to a combination, the program checks whether this candidate masks or is masked by another sound in the combination. If so, the combination is discarded. This method reduces the number of combinations at the end, thus the computation cost when calculating combinations features. However we still have to run through all combinations and discard them almost one by one.

The algorithm used to find the combinations is almost exhaustive. We chose such a method to be sure that we would not miss any interesting solution, and to obtain a large set of results through which the user can "navigate". Indeed, using techniques such as iterative matching, does not ensure that the solution is the best. For instance if a single sound in the database almost fits the target, it will be systematically chosen first, discarding more complex solutions. Obviously the main drawback of our method is the calculation cost. The number of combinations grows exponentially with the number of sounds in the database. We are currently working on heuristics to reduce this number.

6.3. Combination features calculation

Knowing the candidate's combinations, we can compute their features as described in section 4. The calculation of spectral centroid straightforward.

Concerning the combination's partial set, we only compute the amplitude of the partials whose frequencies match the target *most important partial* frequencies. In order to find the matching partials, we take advantage of the harmonicity of the sounds in the database. Indeed, since we know the pitches of all the sounds in the combination, we can guess the correspondence between the partials of the candidates and target sounds, avoiding expensive testing on their frequencies. For instance, the target whose spectrum is plotted in figure 4 above has its 13th most important partial around 586 Hz (D5 -3 cents). This target is decomposed into two harmonic series, C2 and D2 respectively. When generating a combination of a C2 sound and a D2 sound, we then know that this combination will have

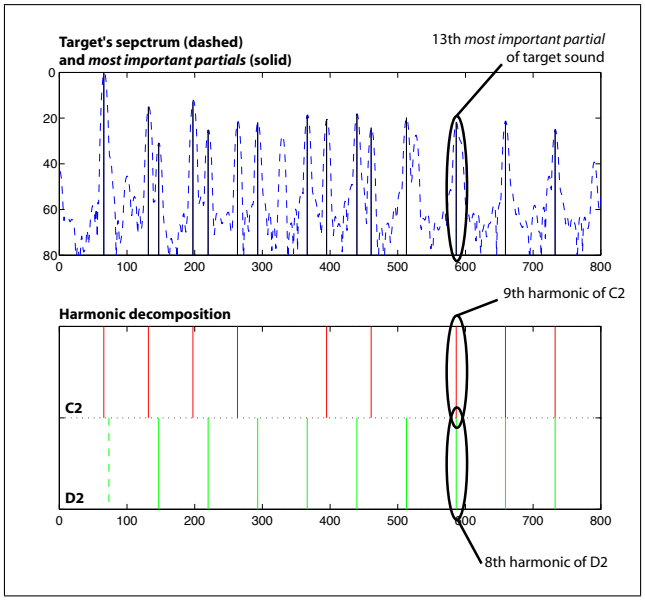

Figure 4. Harmonic decomposition of a target's *most important partials*. The target sound is a mixture of a contrabass playing a C2 and a horn playing a D2.

a partial around 586 Hz. This partial's amplitude is computed by adding the power of the 9th harmonic of the C2 sound and the 8th harmonic of the D2 sound.

Note that the power of the combination's spectrum is only computed at the target's most important partials frequencies. With such a method any combination is described by a partial set of same length as the target's most important partials. Moreover, both partial sets have the same frequencies, and can therefore be compared by considering their respective amplitudes.

6.4. Combination ranking

Now the combinations must be ranked. Finding the rank for each feature is obvious, but determining a global order is more difficult and there is no general solution depending on which aspect of the target is considered more important. This problem is related to the field of preference aggregation. We use two different aggregation methods. One is called the *lexicographic aggregation* and the other is the traditional *weighted Euclidean distance*. *Lexicographic aggregation* assumes that features are sorted in order of decreasing importance. If a combination is better than another one according to the most important feature then it will be globally better. On the other hand, if the combinations are judged equal, the comparison is then performed on the next feature, and so on. This method is interesting since it allows computing only the first feature for all the combinations, selecting the best according to it, and then computing the next feature for the selected subset only. *Weighted Euclidean distance* does not assume the features to be hierarchically ordered, but uses a set of weights expressing their relative importance. Manipulating the weights allows us to give more importance to some particular features.

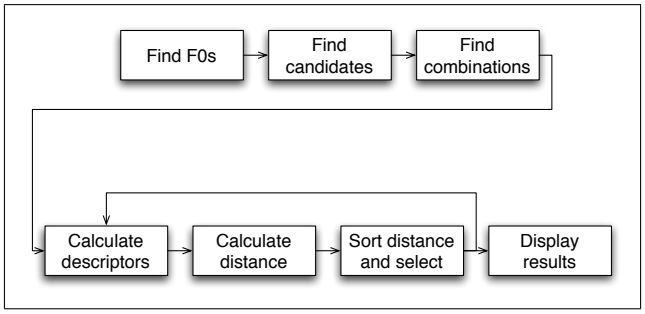

Figure 5. Orchestration procedure flowchart

In the current implementation, the spectral centroid is considered as the first criterion, for which we experimentally define a threshold distance to consider whether two sounds are equal. If a sound is above this threshold it is discarded. If not, the distance is then calculated on partials' amplitudes using uniform weights. Though we cannot guarantee the full accuracy of this method, we still obtain very encouraging results. In the future, we plan to develop interactive methods that allows the exploration of the solution by manipulating the features' order and weights.

7. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a new tool for orchestration purposes. We have introduced the concept of *sound target* and described a framework for the design and the approximation of this target by a combination of instrumental sounds. The proposed orchestration method is based on the decomposition of the target into several fundamental frequencies that enables, in addition to a masking criterion, the selection of the most relevant sounds and combinations.

We then proposed a measure of similarity based on audio features and various methods for distances aggregation. An almost exhaustive method guarantees to find all the best combinations, but with a heavy computation cost. Heuristics on fundamental frequencies and masking effect are the first step towards a solution to this problem, but running the system with a large orchestra is still discouraged due to high computation times. In the future, research will focus on discarding large sets of sounds or combinations within a few tests, by the means of appropriate data structures and heuristics.

The question of sound level calibration in the database was also raised t. We got round this problem by assuming that the sounds were well calibrated and reflect the real acoustic power of the instruments. However, this assumption prevents us from mixing different databases. Future versions of the tool will have to use methods that do not depend on the sample level.

Many experiments with composers were carried out to evaluate the accuracy of our system. Overall, the results are very encouraging, particularly for mid- and high-register targets. In the future the sound description will be extended to better account for the wide range of composers' needs, such as the modelling of noisy and time-varying sounds.

8. ACKNOWLEDGEMENTS

The authors wish to deeply thank the composers Yan Maresz and Joshua Fineberg for their implication and involvement in the project. Their motivation and remarks have contributed to create a rich collaboration and have paved the way to exciting future development and research.

9. REFERENCES

[1] Gerard Assayag, Camilo Rueda, Mikael Laurson, Carlos Agon, and Olivier Delerue. Computer assisted composition at Ircam : From PatchWork to OpenMusic. *COMPUTER MUSIC JOURNAL*, 23(3), 1999.

[2] Guillaume Ballet and Riccardo Borghesi. Studio online 3.0 : An internet "killer application" for remote access to Ircam sounds and processing tools. In *Journées d'Informatique Musicale*, 1999.

[3] Niels Bogaards. Analysis-assisted sound processing with audiosculpt. In *8th International Conference on Digital Audio Effects (DAFX-05)*, pages 269–272, Madrid, Espagne, Septembre 2005.

[4] B. Doval and Xavier Rodet. Fundamental frequency estimation and tracking using maximum likelihood harmonic matching and hmms. In *Proc. IEEE-ICASSP*, pages 221–224, 1993.

[5] Thomas A. Hummel. Simulation of human voice timbre by orchestration of acoustic music instruments. In *Proceedings of International Computer Music Conference 2005*, 2005.

[6] A. P. Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Trans. Speech and Audio Processing*, 11(6) :804–816, 2003.

[7] Vienna Symphonic Library. www.vsl.co.at.

[8] Robert C. Maher and James W. Beauchamp. Fundamental frequency estimation of musical signals using a two-way mismatch procedure. *The Journal of the Acoustical Society of America*, 95(4) :2254–2263, 1994.

[9] Jeremy Marozeau, Alain de Cheveigne, Stephen McAdams, and Suzanne Winsberg. The dependency of timbre on fundamental frequency. *The Journal of the Acoustical Society of America*, 114(5) :2946–2957, 2003.

[10] Geoffroy Peeters. A large set of audio features for sound description (similarity and classification). in the CUIDADO project. Paris, IRCAM, 2004.

[11] David Psenicka. Sporch : An algorithm for orchestration based on spectral analyses of recorded

sounds. In *Proceedings of International Computer Music Conference 2003*, 2003.

[12] Xavier Rodet. Musical sound signal analysis/synthesis : Sinusoidal+residual and elementary waveform models. In IEEE Time-Frequency and Time-Scale Workshop, Coventry, Grande Bretagne., 1997.

[13] François Rose and James Hetrick. Spectral analysis as a ressource for contemporary orchestration technique. In *Proceedings of Conference on Interdisciplinary Musicology 2005*, 2005.

[14] M. Wright, A. Chaudhary, A. Freed, D. Wessel, X. Rodet, D. Virolle, R. Woehrmann, and X. Serra. New applications of the sound description interchange format, 1998.

[15] Chunghsin Yeh, Axel Röbel, and Xavier Rodet. Multiple fundamental frequency estimation of polyphonic music signals. In *2005 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 225–228, Philadelphia, USA, Mars 2005.