Journal of New Music Research
2010, Vol. 39, No. 1, pp. 47–61

Predicting Timbre Features of Instrument Sound Combinations: Application to Automatic Orchestration

Grégoire Carpentier[1], Damien Tardieu[1], Jonathan Harvey[2,3], Gérard Assayag[1], and Emmanuel Saint-James[4]

[1]IRCAM, France; [2]Stanford University, USA; [3]Sussex University, UK; [4]LIP6, France

Abstract

In this paper we first introduce a set of functions to predict the timbre features of an instrument sound combination, given the features of the individual components in the mixture. We then compare, for different classes of sound combinations, the estimated values of the timbre features to real measurements and show the accuracy of our predictors. In the second part of the paper, we present original musical applications of feature prediction in the field of computer-aided orchestration. These examples all come from real-life compositional situations, and were all produced with Orchidée, an innovative framework for computer-aided orchestration recently designed and developed at IRCAM, Paris.

1. Introduction

Computer-aided composition systems aim at formalizing compositional intentions through the manipulation of musical objects and their transformation with appropriate algorithmic procedures. They usually provide a large set of primitives and low-level objects from which composers may create according to their needs more elaborated musical data and programs (for a review see for instance Miranda (2001) or Nierhaus (2009)). Over the years, these environments have addressed almost all dimensions of music writing: melody, harmony, rhythm, counterpoint, instrumentation, spatialization, sound synthesis, to name a few. For some reasons however, orchestration has stayed relatively unexplored until recently (Psenicka, 2003; Hummel, 2005; Rose & Hetrick, 2009).

Orchestration comes into play as soon as timbre issues are addressed. Boulez (Barrière, 1985) makes a clear distinction between the two essential aspects of musical timbre, *articulation* and *fusion*. The former refers to fast timbre variations usually performed by soloists or small ensembles, the latter refers to static or slowly varying textures, in which instrument timbres merge into an overall colour. In this paper we essentially focus on fusion. We consider orchestration as the search of vertical instrument combinations to be played at single time slices, during which the orchestral timbre is static.

Even within the scope of this restriction, the problem is still difficult. An orchestra is composed by many instruments, each of them being able to create a large variety of sounds. By combining these instruments together composers may access a virtually infinite set of sonorities. In practice, orchestration involves an outstanding comprehension of the complex relations between symbolic musical variables and the resulting timbre as a sound phenomenon. In fact, this knowledge is so hard to formalize that composition systems have for years stayed away from this essential domain of composition.

A closer look at the complexity of orchestration (Carpentier & Bresson, 2010) shows that it can be analysed along at least two different 'axes of complexity': the combinatorial explosion of possible sound mixtures within the orchestra on one hand, and the multidimensionality of timbre perception (see for instance McAdams, Winsberg, Donnadieu, De Soete, and Krimphoff (1995), Jensen (1999) or Hajda (2007)) on the other hand. A few orchestration tools exist today (see Section 2), but they all circumvent these aspects of complexity. The combinatorial problem is in particular greatly disregarded. In a recent paper (Carpentier, Assayag, & Saint-James, in press) we have proposed an original approach for the discovery of

Correspondence: Grégoire Carpentier, IRCAM—1 pl. Igor Stravinsky, F-75004 Paris, France. E-mail: gregoire.carpentier@gmail.com

DOI: 10.1080/09298210903581566 © 2010 Taylor & Francis

relevant sound combinations that explicitly addresses combinatorial issues and timbre multidimensionality. We have introduced a formal timbre model for single time slices, in which orchestration is viewed as a constrained multiobjective optimization problem. This work led to the development of Orchidée, a generic and extendible framework for automatic orchestration. Starting from an initial target sound, Orchidée searches instrument sound combinations that—when played together—'sound as similar as possible' to the target. In Orchidée, the fitness of orchestration proposals is computed from various sources of information, considering that perceptual similarity judgements rely on several timbre dimensions.

In this paper, we focus on three perceptual dissimilarity functions used in Orchidée. These functions are computed on the basis of three spectral features: the spectral centroid, the spectral spread and the main resolved partials. We show that the features associated with instrument combinations can be reasonably estimated from the features of the individual elements. We also report convincing musical examples discovered with Orchidée by the joint use of these dissimilarity functions.

The paper is organized as follows. In Section 2, we briefly report the few previous orchestration tools and discuss their core limitations. In Section 3, we recall the foundations of Orchidée and explain why this innovative tool differs from its predecessors. In Section 4, we introduce a set of three spectral features for orchestral timbre similarity and show how we can easily predict some timbre properties of any sound mixtures on the basis of their individual elements. Convincing musical examples from real-life compositional situations are reported in Section 5. Last, conclusions and future work are discussed in Section 6.

2. Background

Roughly speaking, computer-aided composition software may be divided into two main categories:

- Systems based on sound synthesis (e.g. SuperCollider (McCartney, 2002) or Csound (Boulanger, 2000)).
- Systems based on the algorithmic manipulation of musical structures (e.g. OpenMusic (Assayag, Rueda, Laurson, Agon, & Delerue, 1999) or PWGL (Laurson, Kuuskankare, & Norilo, 2009)).

The historical distinction between these approaches is slowly disappearing today. Synthesis systems now tend to consider the sound processing chain as a combination of formalized operators, whereas symbolic systems offer the possibility to manipulate and transform sounds within complex symbolic data and networks.

In the meanwhile, recent advances in music signal processing and machine learning now allow one to automatically extract high-level information from music signals. Music Information Retrieval (MIR) is today recognized as an essential research field (Casey et al., 2008) that could lead to a systematic and formalized organization of sound resources. Unfortunately, though the current convergence of signal and symbolic approaches tends to narrow the gap between low-level and high-level music information, specific musical problems are still to be addressed by computer science. Automatic orchestration is one of these.

In this paper, we consider orchestration as the search of vertical instrument sound combinations that best 'imitate' an input sound target. The target timbre is assumed to be static, and the resulting orchestration is valid for a single time slice only.

To our knowledge there have been three previous attempts to computationally address this problem. Rose and Hetrik (2009) have proposed an explorative and educative tool that allows either the analysis of a given orchestration or the proposition of new orchestrations for a given target sound. The orchestration algorithm invokes a Singular Value Decomposition (SVD) method to approximate the target spectrum as a weighted sum of instrument spectra. Regarding the sample set used as instrumental knowledge, timbre information is represented by a time-averaged 4096 FFT-points spectra. The SVD method requires relatively low computational effort and ensures that the resulting sum minimizes the Euclidean distance to the target spectrum. However, the perceptual significance of a thousands-of-points comparison is questionable, and orchestral limitations cannot be handled by this approach: the SVD-based algorithm is indeed unable to take into account that two sounds cannot be played simultaneously if they are played by the same instrument. In order to cope with such orchestral constraints the authors also suggested the CHI procedure, which first computes the set of all feasible combinations, then ranks them on a distance-to-target criterion. However, as it performs an exhaustive search it is intrinsically bounded to small-size problems.

Another method proposed by Psenicka (2003) addresses the orchestration problem with a Lisp-written program called SPORCH (SPectral ORCHestration). In SPORCH the instrumental knowledge is modelled by an instrument database rather than a sound database, ensuring that any orchestration proposal is physically playable by the orchestra (e.g. solutions cannot allocate more instruments than available). The search method relies on an iterative matching algorithm on spectral peaks. The instrument mixture of which main peaks best match the target ones is considered as the best orchestration. 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. SPORCH first extracts the target peaks, then searches the database item that best matches those peaks. The rating is done by Euclidean distance on the peaks that are close in frequency. Thus, a peak that does not belong to the target but does belong to the tested mixture increases the distance. The best fit peaks are then subtracted from the target and the program iterates. The use of an iterative algorithm ensures low computation times, but nothing guarantees the optimality of the solution. Moreover, SPORCH favours proposals with a first sound very similar to the target, and therefore discards many solutions.

The third system is suggested by Hummel (2005). Hummel's principle is similar to Psenicka's, but the algorithm works on spectral envelopes rather than on spectral peaks. The procedure first computes the target 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). This is why Hummel recommends his system for non-pitched sounds like whispered vowels.

All these methods present the significant advantage of requiring relatively low computation times. However, they all rely on the target spectrum decomposition techniques, invoking either SVD or matching-pursuit methods. They first choose the sound to which the spectrum is the closest to the target spectrum, subtract it from the target spectrum and iterate on the residual. Orchestration is therefore implicitly seen as a knapsack filling process in which 'bigger' elements are introduced first. In other words these methods are likely to behave like greedy algorithms, thus may easily get stuck in low-quality local minima when solving complex problems. On the other hand, they fail in considering timbre perception as a complex, multidimensional phenomenon. Indeed, the optimization process is always driven by a unique objective function. Last, these methods offer poor control of symbolic features in orchestration proposals: the search is driven by the optimization of a spectral-based criterion, no matter the values musical variables may take. Consequently, the resulting solutions are generally difficult to exploit in real compositional processes.

3. Orchidée: A generic and extendible framework for computer-aided orchestration

Taking the above considerations into account we have recently designed and implemented Orchidée, a generic and extendible framework for computer-aided orchestration. Compared to its predecessors Orchidée offers many innovative features:

- The combinatorial issues are handled by appropriate optimization techniques based on evolutionary algorithms. Therefore, Orchidée is not restricted to small-size problems and can find instrument combinations for real-life orchestras in reasonable time.
- A constraint language on the symbolic musical variables is used to make orchestration proposals fit in a given compositional context.
- The multidimensionality of timbre perception is considered through the joint use of several perceptual dissimilarity functions that address distinct dimensions of timbre.

A detailed description of Orchidée is clearly out of the scope of this paper. In the remainder of this section we will simply recall some of the foundation principles of Orchidée. Interested readers may refer to previous papers (Carpentier et al., in press; Carpentier & Bresson, 2010).

3.1 Instrumental knowledge

Like all the previous systems presented in Section 2 the instrumental knowledge in Orchidée comes from instrument sound databases. These databases should be large enough (in terms of instruments, pitches, dynamics, playing styles, etc.) to cover the timbre potentials of the orchestra. Note that within the context of automatic orchestration, instrument sound databases can be viewed as a digital instrumentation treatise. As each item in the database comes from a recording session, the overall collection of samples related to a given instrument conveys useful information about its pitch and dynamics ranges as well as its numerous playing techniques.

In Orchidée, timbre information is represented by a set of low-level features extracted from audio samples. These features are correlated to perceptual dimensions and provide objective timbre quality measures along each dimension. As will be discussed in Section 4, each feature is associated with an *estimation function* and a *dissimilarity function*. The former is used to predict the overall feature value of a sound combination given the feature values of its components. The latter reflects the dissimilarity between a given sound combination and the target along the associated perceptual dimension.

Apart from the knowledge database that gathers symbolic and timbre information about instrument capabilities, Orchidée is purely agnostic. That is, we do not look here for a formalization of some 'classic' rules exposed in famous past orchestration treatises (Berlioz, 1855; Rimski-Korsakov, 1912; Koechlin, 1943; Piston, 1955; Casella, 1958; Adler, 1989). Our goal is rather to encourage the discovery of somehow uncommon instrument combinations. Hence, the timbre information contained in the sound features is the only instrumental knowledge accessible to our system and no particular effort is made to encourage more systematic sound combinations.

3.2 Timbre target

Orchidée provides composers with the opportunity of reproducing a given input *target* timbre with an instrument set. We chose to specify this target using a pre-recorded sound. An alternate possibility would be to allow for a verbal description of the target, but such a method would require to solve two difficult problems:

(i) Find a reduced set of words widely accepted to describe a large set of timbres.
(ii) Correlate these words to sound descriptors (i.e. acoustic features).

Timbre verbalization experiments usually give rise to a large set of verbal attributes (Faure, 2000). People often use different words to describe the same stimulus, and sometimes the same word for different stimuli, making things even harder. Interestingly, the problem of verbalization in the context of orchestration has already been studied by Kendall and Carterette (1993a,b). Through three successive experiments, the authors have identified a small set of verbal attributes to describe the timbre of wind duets: 'Power', 'strident', 'plangent' and 'reed'. However, it should be noted that even for a small set of sounds, numerous experiments are necessary to find a reduced set of attributes. Accounting that in our context many kinds of sounds are to be dealt with, such an approach is not possible.

Another possibility would be to assess the correlation between verbal attributes and acoustic features, in order to get the numerical description of the sound from its verbal description. This issue has also been addressed by Faure (2000) and Kendall and Carterette (1993a,b) in a two-step procedure:

(i) Build a timbre space from dissimilarity judgements.
(ii) Find a verbal correlate for each dimension.

Such an approach is also limited. Timbre spaces have only been obtained for small sets of instrumental sounds, and the verbal correlates of the axes differ amongst the studies.

For all the above reasons, the target timbre in Orchidée is specified by a pre-recorded sound, more precisely by a list of acoustic features extracted from the signal. With such an approach, we leave open the possibility of providing the sound features directly from a verbal description based on further research. Note also that in most cases such a sound is not available before the music has been played at least once. We then offer some appropriate synthesis tools (Carpentier & Bresson, 2010) to generate a target sound from symbolic data, e.g. a chord (see Section 5.2. for an example).

3.3 Multiobjective genetic search

Accounting for the multidimensionality of timbre perception and for the unpredictability of the relative importance of each dimension in timbre similarity subjective evaluations (Carpentier, 2008; Tardieu, 2008), we claim that:

(i) several timbre dimensions have to be considered in the computation of timbre dissimilarities,
(ii) the dissimilarity functions cannot be merged into a single scalar value. Indeed, the attack time may be for instance predominant when the target is a percussive sound, whereas the spectral envelope would be the main criterion for noisy non-pitched sounds. In other words, the relative importance of each dimension cannot be known in advance.

In Orchidée, orchestration is formalized as a multiobjective combinatorial problem (Carpentier et al., in press). The strength of the multiobjective approach is to avoid any prior assumption on the relative importance of the timbre dimensions. Indeed, multiobjective problems have a *set* of efficient solutions rather than a *unique* solution. These efficient solutions reflect different tradeoffs between potentially conflicting objectives (for more details see for instance Ehrgott, 2005).

Large-scale combinatorial optimization problems are often hard to solve, especially when objective functions are not monotonic (and ours are definitely not—see Section 4). This category of problems is therefore often addressed with *Metaheuristics* (Talbi, 2009), which are known to return good-quality solutions in reasonable time. The Orchidée optimization engine makes use of Genetic Algorithms (GAs) (Holland, 1975), one of the major classes of metaheuristics. GAs are inspired by species natural evolution. The search process maintains a population of individuals—encoded as chromosomes—from the best of which new individuals are generated by the application of genetic operators. As the evolution goes on, fitter individuals replace less fit ones and the overall population converges towards optimal regions of the search space. An example of genetic encoding for a string quartet is provided in Figure 1. Individuals (i.e. orchestral configurations) are represented by tuple chromosomes, and each element of the tuple is associated with a given instrument. The figure also illustrates how genetic operators are applied: the *uniform crossover* randomly mixes the coordinates of two parent chromosomes to output two offspring, whereas the *1-point mutation* randomly changes a single coordinate.

3.4 Modelling compositional context

Composing music is composing with time: each compositional element does not come on its own, but in close

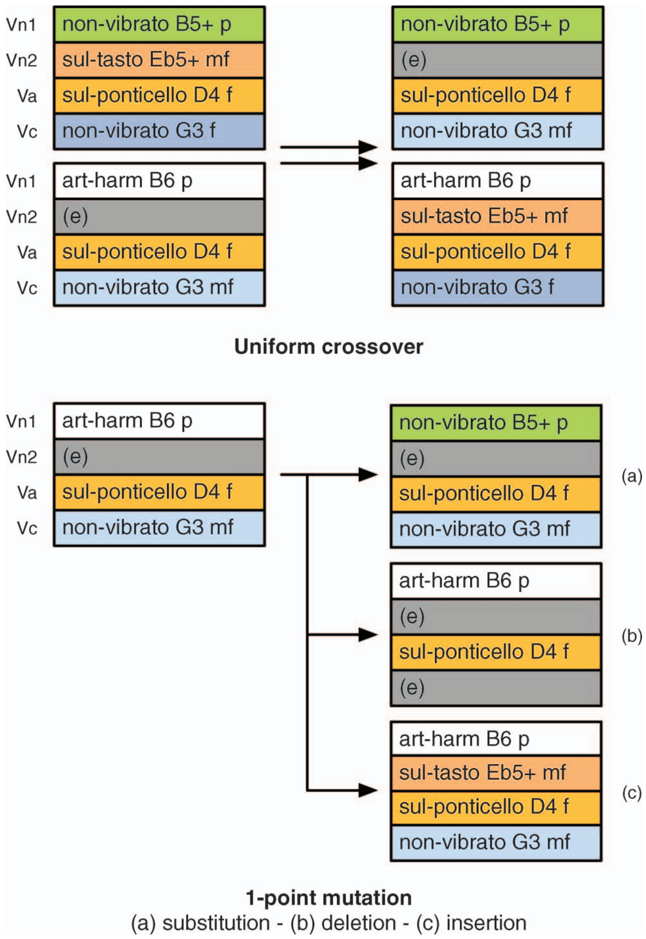

Fig. 1. Integer tuple encoding scheme and genetic operators (example of a string quartet)—(e) denotes the musical rest (i.e. the associated instrument is not used in the mixture). Each slot in the chromosomes corresponds to a well-identified instrument in the orchestra. In uniform crossover, each offspring has in slot *k* the value of slot *k* of one of its parents, randomly chosen.

relation with previous, simultaneous or future other elements. In other words, any musical material is always attached to *a given compositional context*. Consequently, composition tools should help not only in creating new musical material, but also in contextualizing it in a relevant manner.

Consider for example an instrumental texture to be played immediately after a passage where all violins play with a mute. The composer may then require the search algorithm to find solutions with only muted violins, because depending on the tempo instrumentalists may not have enough time to remove the mutes. In a similar manner, composers should be able to specify numerous other requirements, e.g. that orchestration proposals may not require more than ten instruments, involve at most three string-typed instruments, use at least one of each note in a given set, etc.

A natural and expressive paradigm to model such contextual requirements is constraint programming. It has been widely used in past research to address numerous musical problems, e.g. counterpoint (Laurson, 1996), instrumentation (Laurson & Kuuskankare, 2001), harmonic progressions (Pachet & Roy, 2001) or interactive scores (Allombert, Dessainte-Catherine, Larralde, & Assayag, 2008). Musical constraint solvers are today implemented in various computer music environments. OM-Clouds (Truchet, Assayag, & Codognet, 2003) and PWGL-Constraints (Laurson et al., 2009) are such examples.

Orchidée uses global constraints on the music symbolic variables (e.g. notes, dynamics, playing styles, mutes) to formalize this context. In opposition to *local* constraints that are applied to a few variables only, *global* constraints address all variables of a given problem. Orchidée comes with a formal constraint language thanks to which complex constraint networks may be expressed. For instance, the following constraint code is a simple and readable way to restrict orchestration solutions to combinations involving between ten and twelve instruments among which at most six are string-typed, playing all different notes with at least one C2 and all at the same dynamics:

```
[size-min 10]
[size-max 12]
[family at-most 6 strings]
[note all-diff]
[note at-least 1 C2]
[dynamics at-most-diff 1]
```

We will not go into deeper details in this paper. Readers interested in global constraint handling in Orchidée may refer to Carpentier et al. (in press).

4. Sound description

Even if the Orchidée relies on a generic framework that can theoretically consider any number of sound features, estimation functions and dissimilarity functions are hard to define in practice. Indeed, predicting the timbre features of sound combinations is a rather difficult task (Tardieu, 2008). Thus, we currently use three spectral features for which we propose efficient estimation and dissimilarity functions:

(i) The *Spectral Centroid (sc)*, i.e. the mean frequency of the spectrum, which is often correlated to the perceptual brightness (McAdams et al., 1995).

(ii) The *Spectral Spread (ss)*, i.e. the standard deviation of the spectrum, which has been identified as highly correlated to the third dimension of timbre spaces (Peeters, McAdams, & Herrera, 2000).

(iii) The *Main Resolved Partials (MRPs)*, which reflect the harmonic colour of the sound. An auditory model is used to select among prominent spectral peaks the partials resolved by the human auditory system.

All these features are average values on the steady-state of the signal. Temporal features (including the attack time) are not considered here. Note that our purpose is not to build an exhaustive timbre description but rather to design a general framework which can be easily extended by adding new features (Tardieu, 2008) when needed.

4.1 Feature extraction

Figure 2 depicts the way these three features are extracted from the signal. The common step is a short time Fourier transform for each windowed signal frame. From there, a partial tracking procedure is used to select the steady peaks which are then filtered by an ERB (Equivalent Regular Bandwidth) (Glasberg & Moore, 2000) auditory model to compute the MRPs. Simultaneously an inner ear transfer function (Moore, Glasberg, & Baer, 1997) and an ERB model are applied on each FFT frame to extract an instantaneous loudness. The global spectral moments (centroid and spread) are then computed by averaging instantaneous values over time (each frame is weighted by its local loudness).

4.2 Feature estimation functions and dissimilarity functions

We introduce in this paragraph *estimation functions* that predict the features of a sound combination given the features of its components, as well as *dissimilarity functions* that compare the estimated features to the target features. Let $\mathcal{T}$ be the target timbre, s an orchestration proposal, $(sc_i)_i$ the components' centroids, $(ss_i)_i$ the spreads and $(e_i)_i$ the total energies of the individual spectra in s.

Spectral centroid and spectral spread are computed using the energy spectrum. The underlying hypothesis is that the energy spectrum of a sound mixture is the sum of the components' energy spectra. We also assume that all sounds in the mixture blend together in a unique timbre. The centroid and spread of the mixture can then be computed by first finding the mean spectrum of the mixture and then computing the features on this estimated spectrum. This sequence of operations has an analytical solution which is given by Equations 1 and 2. Those formulas can be deduced by using the spectral centroid and spectral spread formulas and the estimation of the mixture energy spectrum by the sum of energy speactra of the instruments.

$$\widehat{sc} = \frac{\sum_i e_i sc_i}{\sum_i e_i}, \tag{1}$$

$$\widehat{ss} = \left(\frac{\sum_i e_i (sc_i^2 + ss_i^2)}{\sum_i e_i} - \widehat{sc}^2 \right)^{1/2}. \tag{2}$$

As far as dissimilarity functions are concerned, we use for both centroid and spread a relative distance to the target feature values $sc_{\mathcal{T}}$ and $ss_{\mathcal{T}}$:

$$D_{\mathcal{T}}^{sc}(s) = \frac{|\widehat{sc} - sc_{\mathcal{T}}|}{sc_{\mathcal{T}}}, \qquad D_{\mathcal{T}}^{ss}(s) = \frac{|\widehat{ss} - ss_{\mathcal{T}}|}{ss_{\mathcal{T}}}. \tag{3}$$

From a psychoacoustic viewpoint this choice of dissimilarity functions might be questionable, however the multicriteria approach used in Orchidée (Carpentier et al., in press) allows us to circumvent this issue. Indeed, Orchidée tries to optimize instrument mixtures by considering perceptual features *separately*, and the only

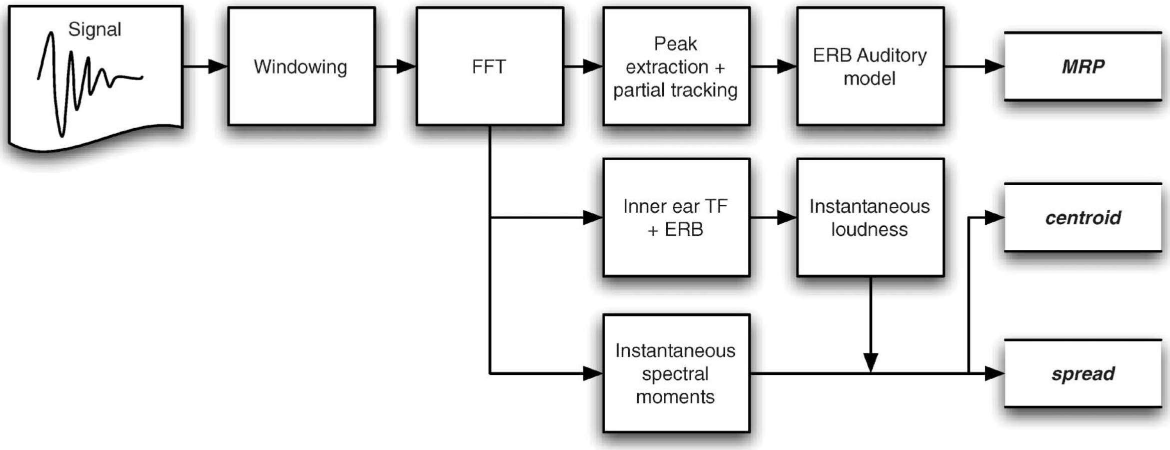

Fig. 2. Extraction of spectral features from audio signals.

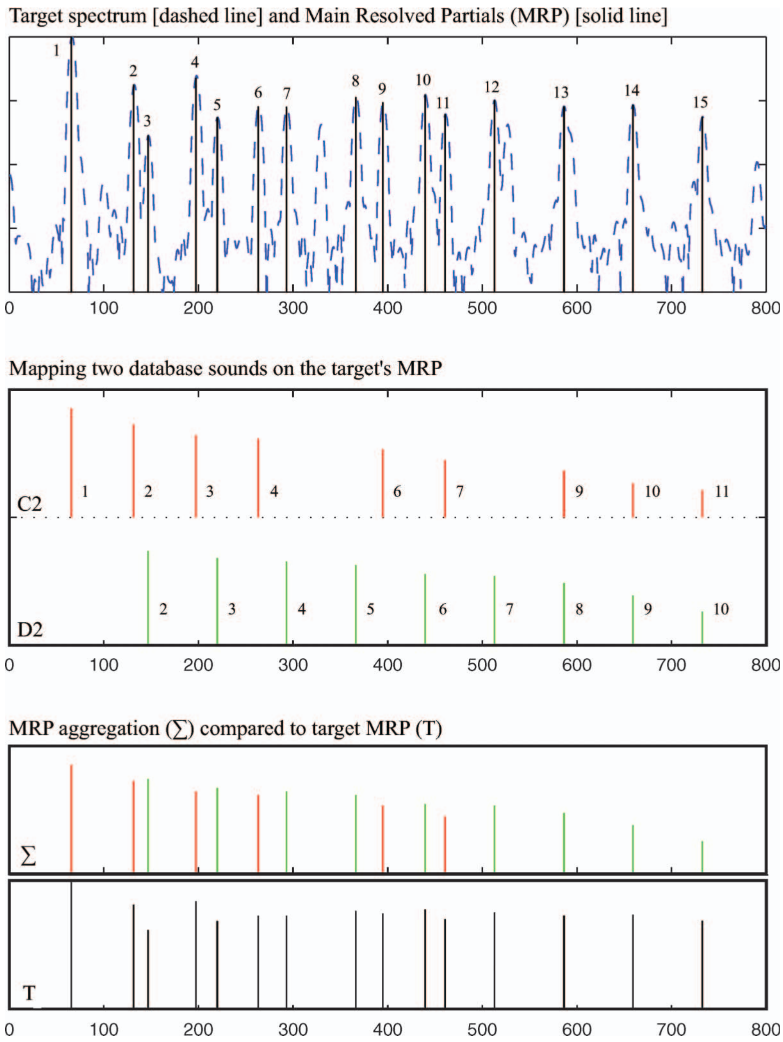

Fig. 3. Extraction and estimation of Main Resolved Partials (MRPs).

Table 1. Dissimilarity values between extracted and computed values of combination features (in parenthesis: standard-deviation of the dissimilarity).

	1 sound	2 sounds	3 sounds	4 sounds
Monophonic mixtures				
centroid	10.4% (9.7%)	10.7% (9.5%)	11.1% (9.2%)	11.2% (9.3%)
spread	8.8% (7.6%)	8.2% (8.5%)	7.7% (8.0%)	7.5% (7.7%)
MRPs	5.0% (5.8%)	5.1% (5.5%)	5.0% (5.1%)	4.7% (5.5%)
Polyphonic mixtures				
centroid	–	11.0% (10.4%)	11.2% (9.9%)	11.2% (10.4%)
spread	–	9.0% (8.8%)	9.5% (9.1%)	9.1% (9.2%)
MRPs	–	5.9% (6.1%)	6.3% (6.9%)	6.6% (7.3%)

requirement is to have orchestral configurations closer to the target on a given objective being better ranked (optimization criteria are interpreted as ‘distances to the target’ only, not pair-wise distances). Additionally, Equation 3 ensures that centroid and spread dissimilarities take values in roughly the same range, which helps the optimization algorithm in managing the potentially conflicting objectives.

Regarding the main partials the process is slightly more complex. For each orchestration problem we first extract the target MRPs according to the process depicted in Figure 2. Then, for each sound in the knowledge database, we identify the partials that match the target MRPs. This operation is a preliminary procedure (before the search process itself) that allows, as we will see, a faster computation of the MRPs of any combination. The matching criterion is a simple threshold frequency ratio δ (usually around 1.5%) on the partials to be matched. For instance if the target has its third MRP at 500 Hz every database sound with a partial between 493 and 507 Hz will get it matched with the third MRP of the target. Let $\{f_n^{\mathcal{T}}\}$ and $\{a_n^{\mathcal{T}}\}$ be respectively the frequencies and amplitudes of the MRPs of $\mathcal{T}$. Let s_i be a sound of the database with associated partials $\{f_p^{\,i}, a_p^{\,i}\}$. The contribution of s_i to the MRPs of $\mathcal{T}$ will therefore be:

$$MRP_{s_i}^{\mathcal{T}}(n) = \begin{cases} a_{p_0}^i & \text{if } \exists\, n_0,\ (1+\delta)^{-1} \le f_{p_0}^i / f_n^{\mathcal{T}} \le 1+\delta, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

To each sound s_i in the knowledge database we then associate a vector of amplitudes $\left\{MRP_{s_i}^{\mathcal{T}}\right\}$ of same length as $\left\{a_n^{\mathcal{T}}\right\}$ which reflects the contribution of s_i to the MRPs of $\mathcal{T}$. Once this preliminary process is completed the computation of the MRPs of any sound combination s is performed in the following way:

$$\widehat{MRP}_s^{\mathcal{T}} = \left\{\max_{i\in I}\left(MRP_{s_i}^{\mathcal{T}}(1)\right), \max_{i\in I}\left(MRP_{s_i}^{\mathcal{T}}(2)\right), \ldots\right\}. \tag{5}$$

Spectra are here computed in decibels, in which case the *max* is a simple and efficient approximation of the mixture spectrum (Roweis, 2000). Note that the preliminary frequency match allows one to deal with partial amplitudes only in the estimation of the MRPs. The computation is therefore extremely efficient.

Figure 3 illustrates the above operations. The upper diagram shows the target spectrum (dashed line) and the corresponding MRPs (solid lines). The middle diagram plots the contributions to the target MRPs of two sounds of the database of pitches C2 and D2. We see that the 4th partial of C2 is matched with the 6th MRP and that both the 9th partial of C2 and the 8th partial of D2 matched with the 13th MRP. Last, the bottom diagram shows the estimation of both contributions.

The dissimilarity function associated with the MRPs is defined in the following way:

$$D_{\mathcal{T}}^{MRP}(s) = 1 - \cos\left(\widehat{MRP}_s^{\mathcal{T}}, MRP_{\mathcal{T}}^{\mathcal{T}}\right). \tag{6}$$

Note that the dissimilarity is minimal when both MRP vectors are proportional. The gain level of the target sound has therefore no effect on the dissimilarity value. Thus, sounds are compared according to the shape of their spectral envelopes. Moreover, the cosine distance implicitly bears more importance to the loudest partials in the MRPs, which seems acceptable from a perceptual viewpoint. Last, $D_{\mathcal{T}}^{MRP}$ lies between 0 and 1, in roughly the same range as $D_{\mathcal{T}}^{SC}$ and $D_{\mathcal{T}}^{SS}$.

4.3 How relevant are optimal solutions?

Our three spectral features and their associated estimation and dissimilarity functions allow us to turn the orchestration problem into a three-objective optimization problem. Before going further however we need to investigate the following issue: how relevant are optimal solutions? Do the optimal solutions of the optimization problem correspond to relevant orchestration proposals for the composer? There are various ways to answer this question.

The first idea is to show that estimation functions are accurate predictors of the sound combination features.

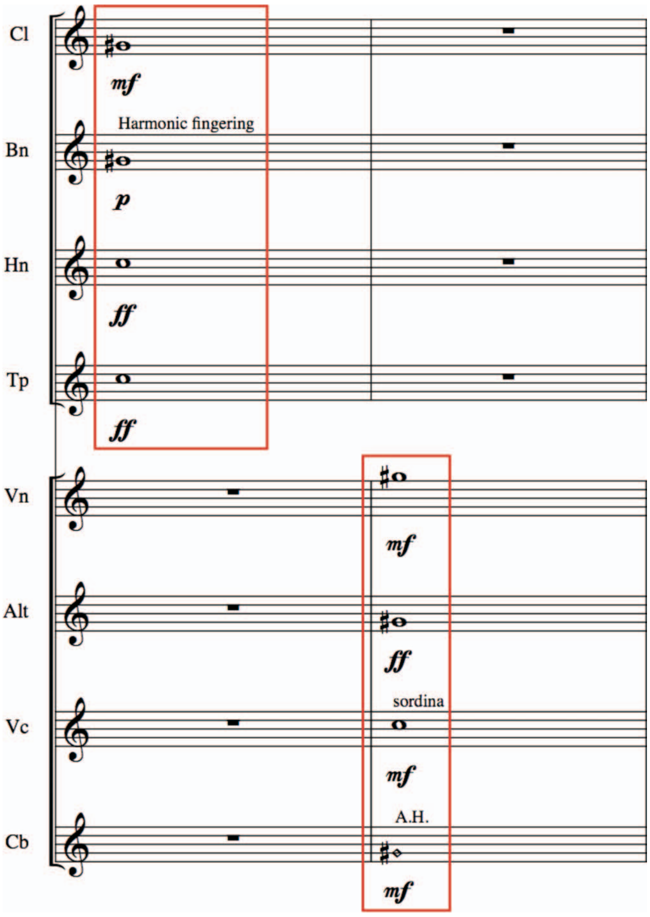

Fig. 4. Two orchestrations of a car horn sound (Alt. is the French abbreviation for viola—score in C).

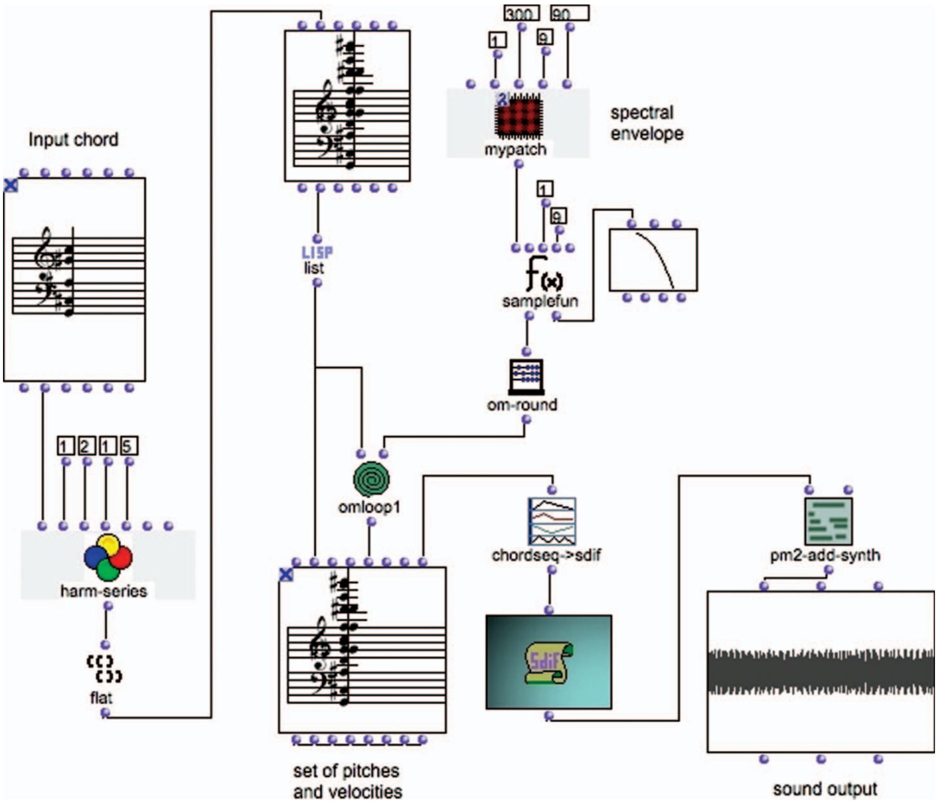

Fig. 5. An OpenMusic patch generating a timbre target from a symbolic chord.

Table 1 reports feature estimation error rates between extracted and computed features for different kinds of mixtures. This experiment was run over 500 instances of each type (monophonic and polyphonic mixtures ranging from one to four sounds). The predicted features were computed from the feature knowledge database with the estimation functions previously introduced. The extracted features were computed by first adding reverberation to each component, then summing the components and applying the extraction process depicted in Figure 2. Reverberation was used to depart from the original samples and simulate mixtures in real auditory conditions.

Considering the difficulty of the prediction task, the error rates are reasonably low and do not depend on the size of the mixtures. It seems that prediction shifts are therefore due to the reverberation rather than the estimation functions themselves. The absence of significant differences in error rates between one-sound mixtures and any other mixture types confirms this hypothesis.

Another way to state the correctness of the problem formulation is to consider the mixtures used in the previous test as orchestration targets and use the information of their components to study the quality of the solutions returned by the search algorithm. Efforts in this direction have been made in Carpentier (2008) and have led to promising results.

5. Musical examples

The Orchidée orchestration prototype has already been used by composers in many real-life situations. We report several significant examples here. All of them—as well as many others—are published on line with convincing sound simulations on the following website: http://recherche.ircam.fr/equipes/repmus/carpentier/

In the remainder of this section we use the scientific pitch notation (middle C is C4).

5.1 Car horns

In this example the target timbre is a field-recorded car horn sound. It is a polyphonic sound in which two pitches can clearly be heard: G#4 and C5. The search space was therefore limited to G#4 and C5 sounds, plus their upper harmonics: G#5, C6, D#6, G6, etc.

Figure 4 reports two orchestration proposals for this horn sound, with two different sets of instruments. Note that the score is in C, and the contrabass harmonic is noted in true pitch.

Surprisingly, the imitation is more convincing with the string section rather than with the wind section. The latter indeed combines woodwind and brass instruments, and it is known that a better perceptual

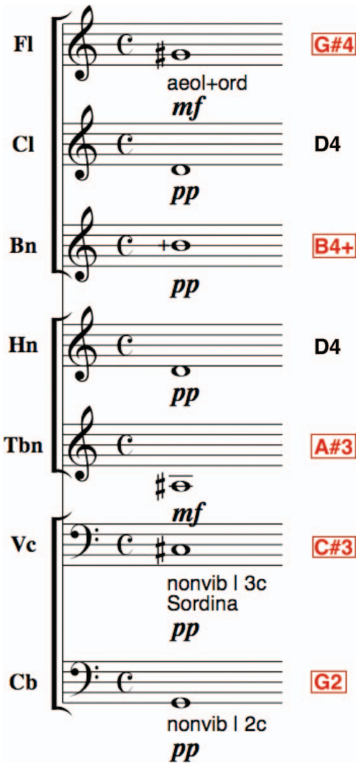

Fig. 6. Generative orchestration: rendering a synthesis mockup with the orchestra (notes in square boxes belong to the initial chord—score in C).

fusion is achieved when all instruments belong to the same family. Moreover, the specific brass timbre at the fortissimo dynamic generates a rougher sound than the original.

5.2 A generative example

This example has been suggested by composer Tristan Murail and illustrates how the orchestration process may be driven by symbolic material in the absence of a pre-recorded target sound. The starting point is here a simple chord (G2, C#3, A#3, G#4, B4 + [1]). Tristan Murail used the OpenMusic (Assayag et al., 1999) composition environment to transform the initial chord into a synthesis sound.

Figure 5 is an OpenMusic patch taking as input argument the symbolic chord (G2, C#3, A#3, G#4, B4+). The chord is first expanded into a harmonic series for each fundamental pitch, and the harmonics intensities are adjusted with a spectral envelope profile. The resulting 'symbolic spectrum' is then mapped on a set of frequency/amplitude pairs and a target sound is generated by simple additive synthesis.

The orchestration of the synthesized sound is reported on Figure 6. It contains all the pitches of the initial chord plus two D4 sounds (third harmonic of G2). The resulting orchestration has the same timbre characteristics as the synthesized target sound and adds the 'richness' of real instruments. This example shows how simple synthesized sounds may be used as 'mock-ups' of complex timbre mixtures for which no pre-recorded sound is available (Carpentier & Bresson, 2010).

5.3 Timbre 'fade in'

In this example suggested by composer Yan Maresz the constraint solver of Orchidée is used to generate a continuous timbre evolution over time. The constraint solving algorithm embedded in Orchidée is a local search procedure (Carpentier et al., in press). It iteratively changes one variable at a time, and keeping trace of all visited configurations gives a continuous sound path from an initial configuration to another.

Figure 7 illustrates this process. Starting from the recording of a trombone played with a bassoon reed, we first looked for a combination of sounds that best imitates the target. We then transformed the output orchestration into a unison of three instruments all playing at the *pp* dynamic thanks to the following constraint set:

c1: [size-min 3]
c2: [size-max 3]
c3: [note at-most-diff 1]
c4: [dynamics at-least 3 pp]

Storing the trace of the Orchidée constraint resolution algorithm gives a continuous timbre motion from the initial orchestration to the final unison. Reversing the whole evolution in Figure 7 we thus obtained a 'timbre fade in'. The first bar is the initial pianissimo C5 unison played by clarinet, bassoon and viola. The contrabass enters at bar 2. The clarinet switches to A#5 + at bar 3, etc. Each bar corresponds to one iteration of the constraint solver at which *only one* variable (i.e. one part) is changed. The timbre gets more and more complex over time and finally reaches the target timbre at bar 12. This example clearly emphasizes how time-varying orchestrations can still be handled by a static timbre model when the evolution is driven by symbolic parameters.

5.4 *Speakings* ostinato

This last, slightly more complex example comes from a collaboration with composer Jonathan Harvey on his lastest piece *Speakings* for orchestra and electronics. In this work Jonathan Harvey focused his writing on the

[1] The + symbol here refers to the quarter-tone notation. B4 + should be understood as a quarter-tone above B4.

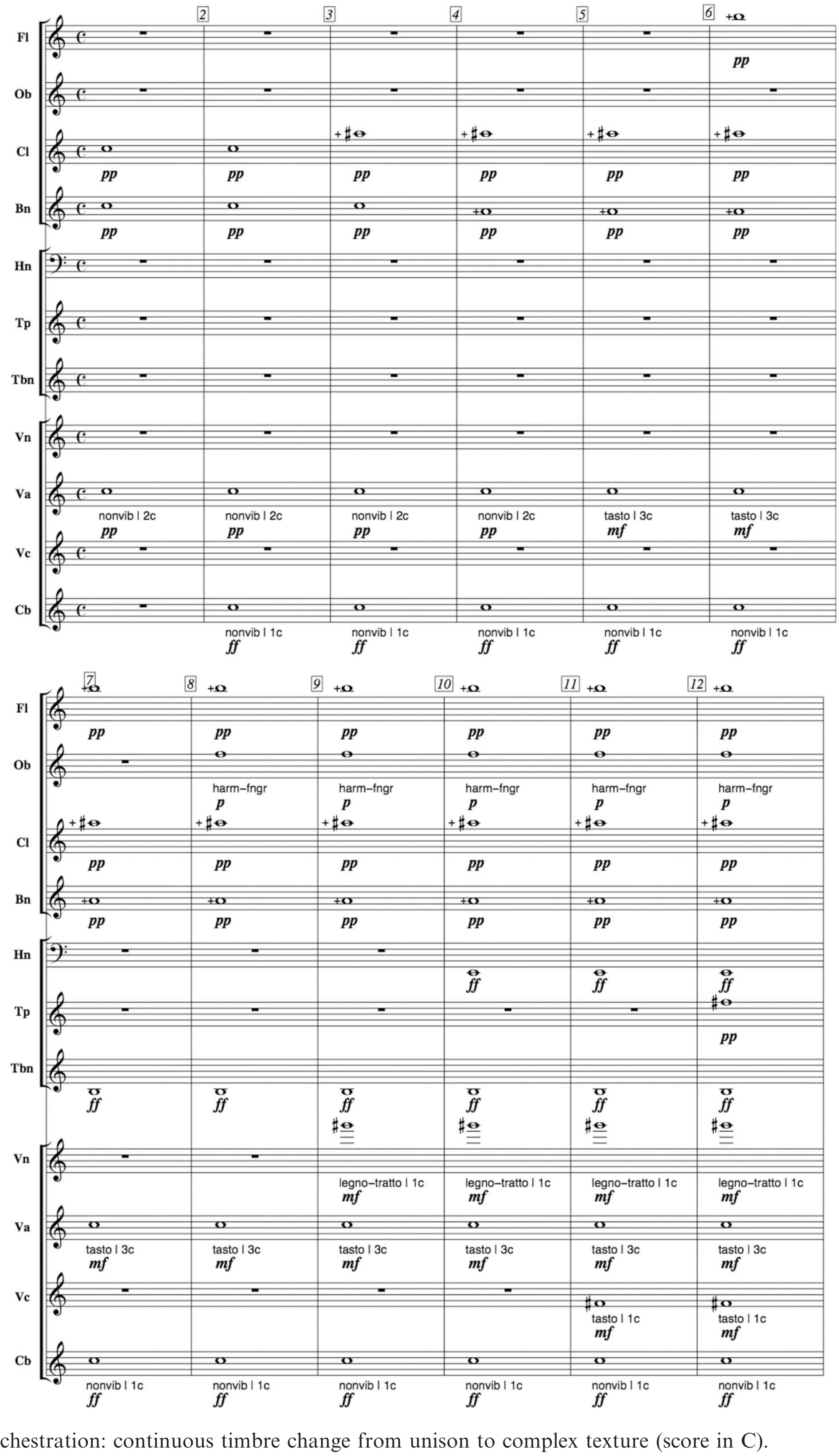

Fig. 7. Evolutive orchestration: continuous timbre change from unison to complex texture (score in C).

imitation of human voice timbres. At the beginning of the third part of *Speakings* a subset of the orchestra plays a tutti ostinato which is used as a harmonic background line for the soloists. These orchestral textures have been largely written with the help of Orchidée.

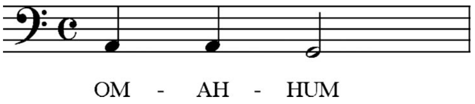

Fig. 8. Mantra used for the automatic generation of *Speakings* ostinato.

The initial material was a simple three-note mantra sung and recorded by the composer. To each note corresponded a given vowel: *Oh*/*Ah*/*Hum* (see Figure 8). The goal was to imitate the sound of the sung mantra with an ensemble of 13 musicians. The composer wanted the orchestra to sing the mantra 22

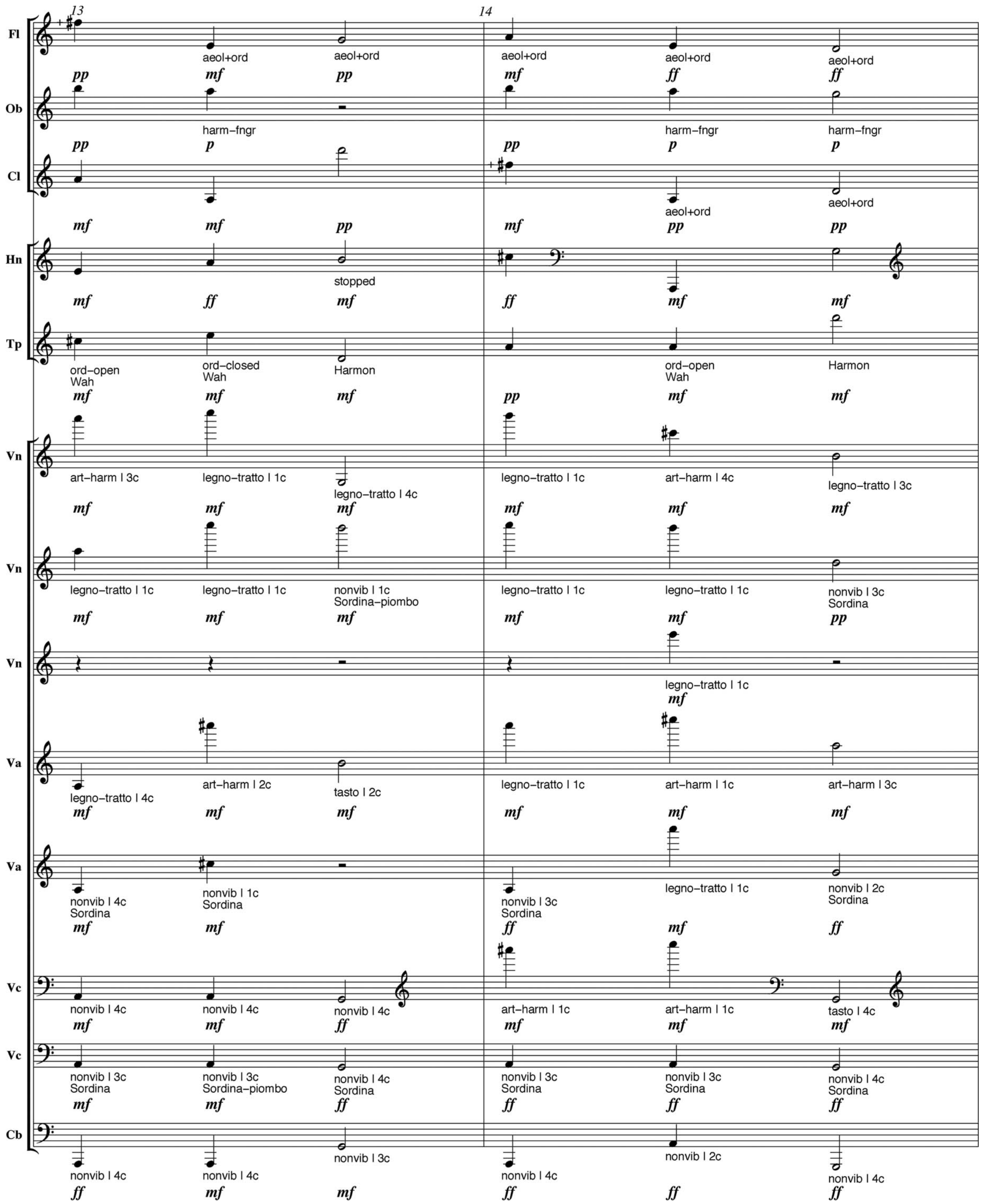

Fig. 9. Automatic orchestration of a mantra ostinato for Jonathan Harvey's piece *Speakings* (bars 11 and 12—score in C).

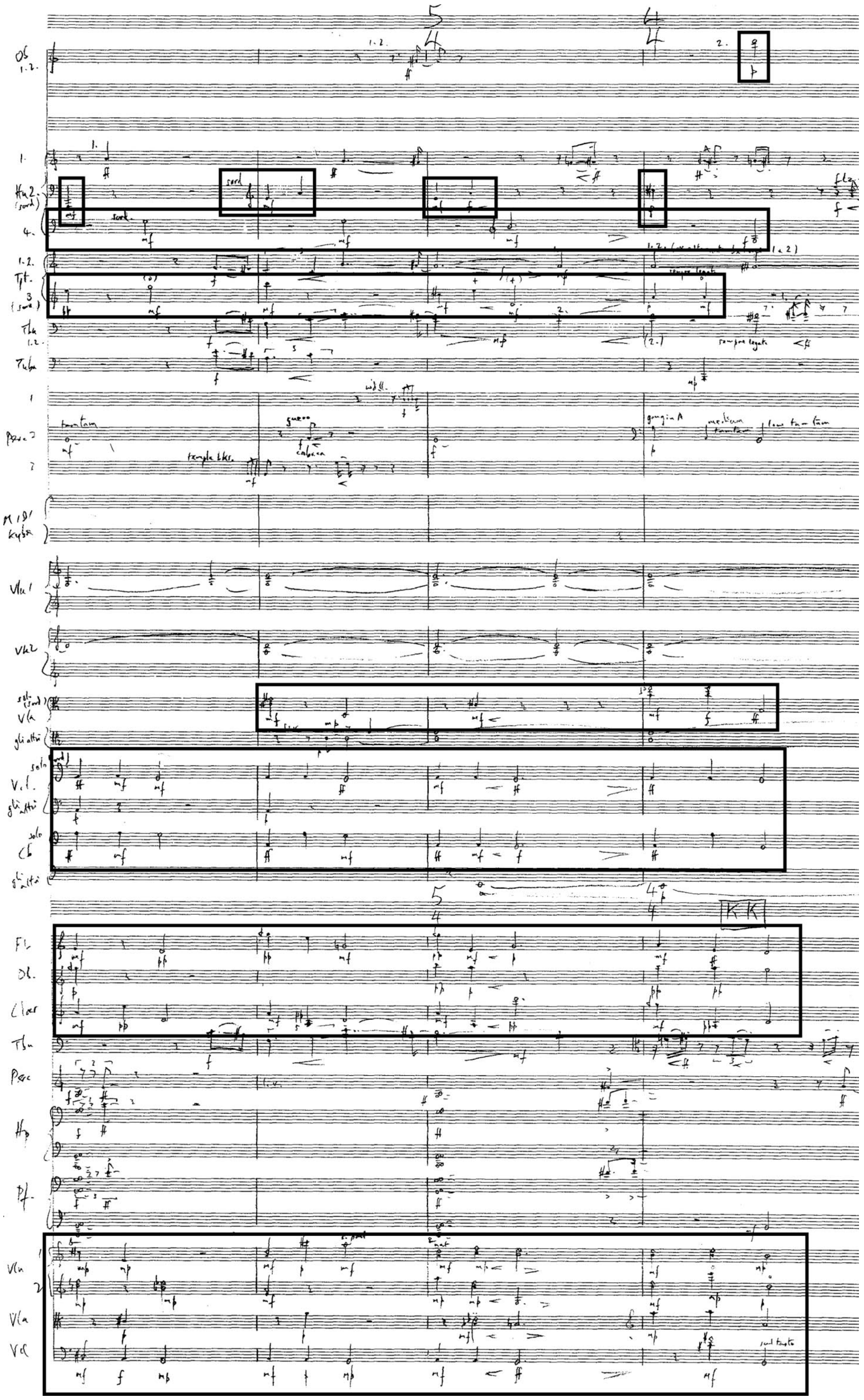

Fig. 10. Bars 11 to 14 of the mantra ostinato in Jonathan Harvey's piece *Speakings* © 2008 by Faber Music Ltd, reproduced by kind permission of the publishers. Parts written with our orchestration tool are enclosed in black boxes—score in C.

times, and wished the resulting timbre to evolve along the ostinato in the following manner: the sound was to become louder and louder over time, brighter and brighter, closer and closer to the target vowel. In addition, the orchestration was to use higher and higher pitches (harmonics of the vowels fundamentals) in harmonically denser and denser[2] chords. This quite complex demand was processed as follows:

(1) As the ostinato was to be played by an ensemble of 13 musicians, at most 13 different pitches were playable simultaneously. For each sung vowel we thus generated 13 sets of 10 different solutions with the following constraint for each set k in $\{1, \ldots ,13\}$: [note at-least-diff k]. This constraint ensures to have at least k different pitches in each solution of set k .
(2) For each sung vowel we used Orchidée to generate 130 solutions of various timbres and chord densities. For each solution we then computed the following features: perceptual loudness, spectral centroid, MRPs, highest pitch, harmonic density (i.e. the number of different pitches in the chord), size of the orchestration. The overall loudness of a solution was computed with the following formula (for more details see Moore (2003)):

$$\hat{L} = \left(\sum_i e_i\right)^{0.3}. \tag{7}$$

(3) For each vowel in the mantra we then used a local search algorithm to find a 'path' linking 22 points in the overall set of 130 solutions, in such a way that all the above features increase over the evolution.

An excerpt of the global solution generated by this procedure is given in Figure 9. Harvey's corresponding final score is given in Figure 10. Parts enclosed in black boxes have been written with Orchidée.

Generally speaking the orchestration suggested by the system has been kept 'as is' in the final score, though a few changes were necessary to cope with voice leading or practical playing issues. For instance the trumpet player who wishes to perfectly follow the score of Figure 9 has to change his mute in the middle of the 13th bar, remove it on the first crotchet of the 14th bar and change it again twice on the following two notes. This is obviously impossible in practice, and the composer chose to keep the same mute for the whole passage. Most other changes deal with specific playing styles resulting in extremely soft sounds (e.g. *col legno tratto* for the strings) that cannot be heard in orchestral music.

Apart from these minor changes it should be noted that most of the dynamics (and their frequent variations) were kept 'as is' by the composer. Jonathan Harvey even left a note to the director at the beginning of the ostinato: '*Great care to respect the dynamics*'. Orchidée was therefore very helpful in finding the finest balance in the instrument intensities.

Speakings was premiered on 19 August 2008 in the Royal Albert Hall, London (BBC Scottish Orchestra, director Ilan Volkov).

6. Conclusions and future work

In this paper we have presented functions that predict the timbre features of an instrument sound combination, given the features of the individual components in the mixture. These functions have been evaluated on a significant collection of test case instances. Results show that timbre features of complex sounds may be accurately predicted with rather simple estimation functions. Embedded in Orchidée, an innovative automatic orchestration tool recently designed and developed at IRCAM, feature estimators, allow one to address various musical situations in which timbre control and exploration are the main issues. Convincing examples from real-life compositional situations confirm the interest and potential of computational timbre estimation for the sake of musical creativity.

Future work will first focus on the addition of new sound features in Orchidée. Recent research in timbre modelling for computer-aided orchestration (Tardieu, 2008) has provided new feature estimation methods that still remain to be experienced. We believe that composers will then be able to address a wider range of musical situations among which the spectral issue will be no more than a particular case. As far as long term research is concerned, we will concentrate on the design of efficient time models for describing and computing time evolving timbres.

Acknowledgements

The authors would like to deeply thank all composers at IRCAM who have strongly contributed to our research. We are particularly grateful to composer Yan Maresz for his daily input into this project, as well as Hugues Vinet for his support. This work has been partly funded by the French National Research Agency (ANR) in the framework of the *Sample Orchestrator* project.

References

Adler, S. (Ed.). (1989). *The Study of Orchestration*. New York: Norton Company.

Allombert, A., Dessainte-Catherine, M., Larralde, J., & Assayag, G. (2008, November). A system of interactive scores based on qualitative and quantitative temporal constraints. In *Proceedings of the 4th International Conference on Digital Arts (ARTECH 2008)*, Porto, Portugal.

Assayag, G., Rueda, C., Laurson, M., Agon, C., & Delerue, O. (1999). Computer-assisted composition at IRCAM: From PatchWork to OpenMusic. *Computer Music Journal*, *23*(2), 59–72.

[2] The density of a chord is here understood as the number of different pitches in it.

Barrière, J.-B. (Ed.). (1985). *Le timbre, métaphore pour la composition*. Paris: Christian Bourgois.

Berlioz, H. (1855). *Traité d'instrumentation et d'orchestration*. Paris: Henri Lemoine.

Boulanger, R. (2000) *The Csound Book*. Cambridge, MA: MIT Press.

Carpentier, G. (2008). *Computational approach of musical orchestration—constrained multiobjective optimization of sound combinations in large instrument sample databases* (PhD thesis). University UPMC-Paris 6, France.

Carpentier, G., Assayag, G., & Saint-James, E. (in press). Solving the musical orchestration problem using multiobjective constrained optimization with a genetic local search approach. *Heuristics*.

Carpentier, G., & Bresson, J. (2010). Interacting with symbolic, sound and feature spaces in *Orchidée*, a computer-aided orchestration environment. *Computer Music Journal*, *34*(1), 10–27.

Casella, A. (1958). *Technique de l'orchestre contemporain*. Paris: Ricordi.

Casey, M., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., & Stanley, M. (2008). Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, *96*(4), 668–696.

Ehrgott, M. (2005). *Multicriteria Optimization* (2nd ed.). Berlin: Springer.

Faure, A. (2000). *Des sons aux mots: Comment parle-t-on du timbre musical?* (PhD thesis). Ecoles des Hautes Etudes en Sciences Sociales, Paris, France.

Glasberg, B.R., & Moore, B.C.J. (2000). A model for prediction of thresholds loudness and partial loudness. *Hearing Research*, *47*, 103–138.

Hajda, J.M. (2007). *The Effect of Dynamic Acoustical Features on Musical Timbre*. Berlin: Springer.

Holland, J.H. (1975). *Adaptation in natural and artificial systems* (PhD thesis). University of Michigan, USA.

Hummel, T.A. (2005, September 5–9). Simulation of human voice timbre by orchestration of acoustic music instruments. In *Proceedings of International Computer Music Conference (ICMC)*, Barcelona, Spain.

Jensen, K. (1999). *Timbre models of musical sounds* (PhD thesis). University of Copenhagen, Denmark.

Kendall, R.A., & Carterette, E.C. (1993a). Verbal attributes of simultaneous wind instrument timbres: I. von Bismarck's adjectives. *Music Perception*, *10*, 445.

Kendall, R.A., & Carterette, E.C. (1993b). Verbal attributes of simultaneous wind instrument timbres: II. Adjectives induced from Piston's orchestration. *Music Perception*, *10*, 469.

Koechlin, C. (1943). *Traité de l'orchestration* (4 vols.). Paris: Max Eschig.

Laurson, M. (1996). *PatchWork: A visual programming language and some musical applications* (PhD thesis). Sibelius Academy, Helsinki, Finland.

Laurson, M., & Kuuskankare, M. (2001, November 26–December 1). A constraint based approach to musical textures and instrumental writing. In *Seventh International Conference on Principles and Practice of Constraint Programming, Musical Constraints Workshop*, Paphos, Cyprus.

Laurson, M., Kuuskankare, M., & Norilo, V. (2009). An overview of PWGL, a visual programming environment for music. *Computer Music Journal*, *33*(1), 19–31.

McAdams, S., Winsberg, S., Donnadieu, S., De Soete, G., & Krimphoff, J. (1995). Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological Research*, (58), 177–192.

McCartney, J. (2002). Rethinking the computer music language: SuperCollider. *Computer Music Journal*, *26*, 61–68.

Miranda, E.R. (2001). *Composing Music with Computers*. Burlington, MA: Focal Press.

Moore, B.C.J. (2003). *An Introduction to the Psychology of Hearing* (5th ed.). New York: Academic Press.

Moore, B.C.J., Glasberg, B.R., & Baer, T. (1997). A model for prediction of thresholds, loudness, and partial loudness. *Journal of Audio Engeneering Society*, *45*, 224–240.

Nierhaus, G. (2009). *Algorithmic Composition—Paradigms of Automated Music Generation*. Berlin: Springer.

Pachet, F., & Roy, P. (2001). Musical harmonization with constraints: A survey. *Constraints*, *6*(1), 7–19.

Peeters, G., McAdams, S., & Herrera, P. (2000, August 27–September 1). Instrument description in the context of MPEG-7. In *Proceedings of International Computer Music Conference*, Berlin, Germany.

Piston, W. (1955). *Orchestration*. New York: Norton Company.

Psenicka, D. (2003, September 29–October 4). Sporch: An algorithm for orchestration based on spectral analyses of recorded sounds. In *Proceedings of International Computer Music Conference (ICMC)*, Singapore.

Rimski-Korsakov, N.A. (1912). *Principles of Orchestration, with Musical Examples Drawn from his Own Works*. New York: Maximilian Steinberg.

Rose, F., & Hetrick, J. (2009). Enhancing orchestration technique via spectrally based linear algebra methods. *Computer Music Journal*, *33*(1), 32–41.

Roweis, S.T. (2000, November 27–30). One microphone source separation. In *Neural Information Processing Systems (NIPS)*, Denver, CO, pp. 793–799.

Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*. New York: Wiley.

Tardieu, D. (2008). *Modèles d'intruments pour l'aide à l'orchestration* (PhD thesis). Université Pierre et Marie Curie—Paris 6, France.

Truchet, C., Assayag, G., & Codognet, P. (2003, August). OMClouds, a heuristic solver for musical constraints. In *MIC2003: The Fifth Metaheuristics International Conference*, Kyoto, Japan.