Instrument Recognition in Polyphonic Music Based on Automatic Taxonomies

Slim Essid, Gaël Richard, *Member, IEEE*, and Bertrand David

***Abstract*—We propose a new approach to instrument recognition in the context of real music orchestrations ranging from solos to quartets. The strength of our approach is that it does not require prior musical source separation. Thanks to a hierarchical clustering algorithm exploiting robust probabilistic distances, we obtain a taxonomy of musical ensembles which is used to efficiently classify possible combinations of instruments played simultaneously. Moreover, a wide set of acoustic features is studied including some new proposals. In particular, signal to mask ratios are found to be useful features for audio classification. This study focuses on a single music genre (i.e., jazz) but combines a variety of instruments among which are percussion and singing voice. Using a varied database of sound excerpts from commercial recordings, we show that the segmentation of music with respect to the instruments played can be achieved with an average accuracy of 53%.**

***Index Terms*—Hierarchical taxonomy, instrument recognition, machine learning, pairwise classification, pairwise feature selection, polyphonic music, probabilistic distances, support vector machines.**

I. Introduction

UNDERSTANDING the timbre of musical instruments has been for a long time an important issue for musical acoustics, psychoacoustics, and music cognition specialists [1]–[6]. Not surprisingly, with the recent technology advances and the necessity of describing automatically floods of multimedia content [7], machine recognition of musical instruments has also become an important research direction within the music information retrieval (MIR) community. Computers are expected to perform this task on real-world music with its natural composition, arrangement, and orchestration complexity, and ultimately to separate the note streams of the different instruments played.

Nevertheless, the majority of the studies handled the problem using sound sources consisting in isolated notes [8]–[16]. Fewer works dealt with musical phrase excerpts from solo performance recordings [8], [17]–[25], hence, making a stride forward toward realistic applications.

As for identifying instruments from polyphonic music, involving more than one playing at a time, very few attempts were made with important restrictions regarding the number of instruments to be recognized, the orchestration, or the musical score played. Often in those studies, artificially mixed simple musical elements (such as notes, chords, or melodies) were utilized. Additionally, some proposals related the task of instrument recognition to automatic music transcription or source separation, requiring the different notes to be known prior to recognition [26]–[28]. The success of this task is then intimately connected to the efficiency of the extraction of multiple fundamental frequencies, which is known to be a very difficult problem, especially for octave-related notes.

Using realistic musical recordings, Eggink and Brown proposed a system based on a missing feature approach [29] capable of identifying two instruments playing simultaneously. More recently, the same authors presented a system recognizing a solo instrument in the presence of musical accompaniment after extracting the most prominent fundamental frequencies in the audio signals [30]. It is also worth mentioning a study using independent subspace analysis to identify two instruments in a duo excerpt [31].

In this paper, we introduce a multi-instrument recognition scheme processing real-world music (including percussion and singing voice), that does not require pitch detection or separation steps. Our approach exploits a taxonomy of musical ensembles, that is automatically built, to represent every possible combination of instruments likely to be played simultaneously in relation to a given musical genre. We show that it is possible to recognize many instruments playing concurrently without any prior knowledge other than musical genre.[1] Decisions are taken over short time horizons enabling the system to perform segmentation of the music with respect to the instruments played. We show through experimental work that satisfactory recognition accuracy can be achieved with up to four instruments playing at the same time.

We start by an overview of our system architecture (Section II). We then describe the acoustic features examined, including new proposals, and we detail our approach for selecting the most relevant features (Section III). Subsequently, a brief presentation of various machine learning concepts used in our work is given (Section IV). Finally, we proceed to the experimental validation (Section V) and suggest some conclusions.

II. System Architecture

The primary idea behind the design of our instrument recognition system is to recognize every combination of instruments possibly playing simultaneously. Immediately, one gets puzzled by the extremely high combinatorics involved. If we consider orchestrations from solos to quartets featuring 10 possible

Manuscript received January 31, 2005; revised August 16, 2005. This work was supported by CNRS under the ACI project "Music Discover." The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Judith Brown.

The authors are with LTCI-CNRS, GET-Télécom Paris, 75014 Paris, France (e-mail: Slim.Essid@enst.fr; Gael.Richard@enst.fr; Bertrand.David@enst.fr).

Digital Object Identifier 10.1109/TSA.2005.860351

[1]Note that the genre of a given piece of music can be easily obtained either by exploiting the textual metadata accompanying the audio or by using an automatic musical genre recognition system [32], [33] in the framework of a larger audio indexing system.

instruments, in theory the number of combinations is already $\binom{10}{1} + \binom{10}{2} + \binom{10}{3}) + \binom{10}{4} = 595$.[2] Obviously, a system that tests for such a large number of classes to arrive at a decision could not be amenable to realistic applications. The question is then: how can a system aiming at recognizing possible instrument combinations be viable?

First, the reduction of the system complexity should mainly target the test procedure, i.e., the actual decision stage. In fact, heavy training procedures can be tolerated since they are supposed to be done "once and for all" in laboratories having at their disposal large processing resources, while testing should be kept light enough to be supported by end-users' devices.

Second, although in theory, any combination of instruments is possible, some of these combinations are particularly rare in real music. Of course, choosing a specific music orchestration for a composition is one of the degrees of freedom of a composer. Nevertheless, though in contemporary music (especially in classical and jazz) a large variety of orchestrations are used, it is clear that most trio and quartet compositions use typical orchestrations traditionally related to some musical genre. For example, typical jazz trios are composed of piano or guitar, double bass, and drums, typical quartets involve piano or guitar, double bass, drums, and a wind instrument or a singer... In a vast majority of musical genres, each instrument, or group of instruments, has a typical role related to rhythm, harmony, or melody. Clearly, jazz music pieces involving piano, double bass, and drums are much more probable than pieces involving violin and tenor sax without any other accompaniment, or bassoon and oboe duets. Therefore, such rare combinations could reasonably be eliminated from the set of possible classes (optionally) or included in a "miscellaneous" labeled class.

Even if we consider only the most probable orchestrations, the number of possible combinations is still high. The key idea is to define classes from instruments or groups of instruments (possibly playing simultaneously at certain parts of a musical piece) that can be reduced by building super-classes consisting in unions of classes having similar acoustic features. These super-classes constitute the top level of a hierarchical classification scheme (such as the one depicted in Fig. 3). These super-classes may be divided into classes (final decisions) or other super-classes. The classification is performed hierarchically in the sense that a given test segment is first classified among the top-level super-classes, then it is determined more precisely (when needed) in lower levels. For example, if a test segment involves piano and trumpet, then it is first identified as PnM (where Pn is piano and M is voice or trumpet) and subsequently as PnTr (where Tr is trumpet).

Such a taxonomy is expected to result in good classification performance and possibly to "make sense" so that the maximum number of super-classes can be associated with labels easily understandable by humans. Thus, a "coarse" classification (stopping at the high levels) is still useful.

A block diagram of the proposed system is given in Fig. 1. In the training stage, the system goes through the following steps:

1) *Building a hierarchical taxonomy*:
   a) A large set of candidate features are extracted (Section III-A).
   b) The dimensionality of the feature space is reduced by principal component analysis (PCA) yielding a smaller set of transformed features (Section III-B) to be used for inferring a hierarchical taxonomy.
   c) A hierarchical clustering algorithm (Section IV-A) (exploiting robust probabilistic distances between possible classes) is used to generate the targeted taxonomy.
2) *Learning classifiers based on the taxonomy*:
   a) The original set of candidate features (obtained at step 1a) is processed by a pairwise feature selection algorithm (Section III-B) yielding an optimal subset of features for each possible pair of classes at every node of the taxonomy found at step 1.
   b) Support Vector Machines (SVM) classifiers (Section IV-B) are trained for every node of the taxonomy on a "one versus one" basis using features selected at step 2a.

For testing (gray-filled blocks), only selected features are extracted and used to classify the unknown sounds based on the taxonomy and SVM models obtained at the training stage.

III. Feature Extraction and Selection

A. Feature Extraction

Unlike speech and speaker recognition problems, there exists no consensual set of features such as mel frequency cepstrum coefficients (MFCC) enabling successful instrument recognition. Numerous proposals have been made in various work on audio classification [8], [13], [18], [19], [25], [34], [35] and many have been compiled within the MPEG-7 standardization effort [7] (see [36] and [37] for an overview). Our approach consists in examining a wide selection of potentially useful features to select the most relevant ones thanks to a feature selection algorithm (FSA). We focus on low-level features that can be extracted robustly from polyphonic musical phrases. Moreover, we use the so-called "instantaneous descriptors," i.e., computed locally in sliding overlapping analysis windows (frames) with an overlap of 50%. Three different window sizes are used, standard 32-ms windows for the extraction of most features (used by default) and longer 64-ms and 960-ms windows for specific features when needed. Feature values measured over each long window are then assigned to each 32-ms frame corresponding to the same time segment. To avoid multipitch estimation and attack transient detection, features specifically describing the harmonic structure and attack characteristics of musical notes are not considered. The following temporal, cepstral, spectral, and perceptual features are extracted.

1) Temporal Features: They consist of the following.

- *Autocorrelation Coefficients* (AC) (reported to be useful by Brown [35]) which represent the "signal spectral distribution in the time domain."
- *Zero Crossing Rates*, computed over short windows (ZCR) and long windows (lZCR); they can discriminate periodic

[2] $\binom{q}{p}$ is the binomial coefficient (the number of combinations of p elements among q).

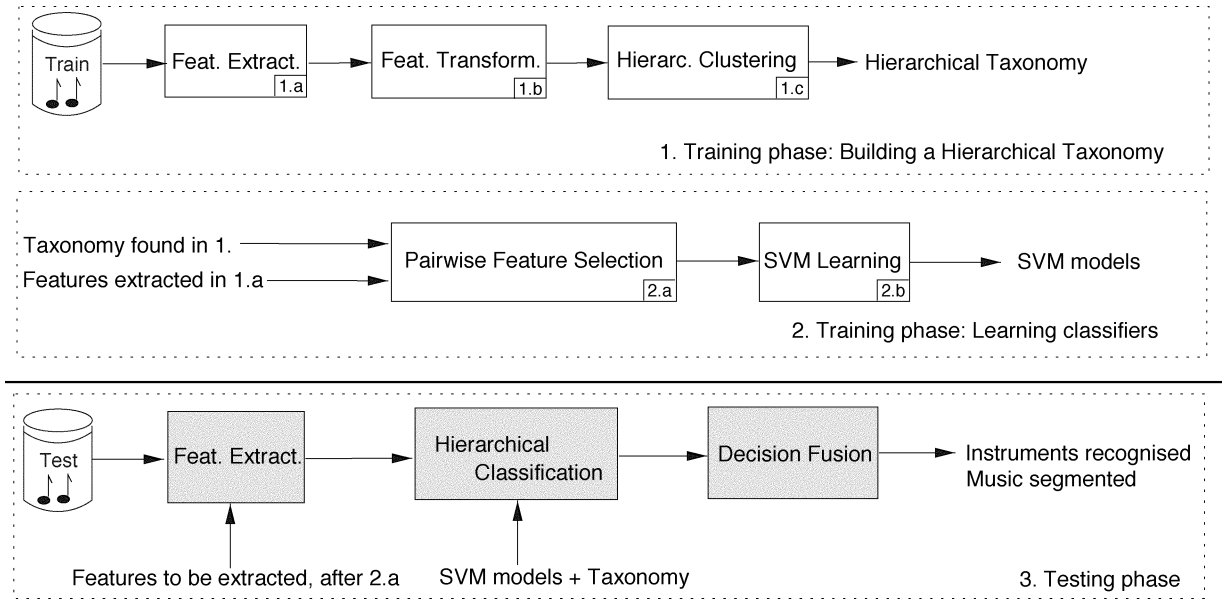

Fig. 1. Block diagram of the hierarchical recognition system. Testing stage blocks are gray-filled.

signals (small ZCR values) from noisy signals (high ZCR values).

- Local *temporal waveform moments*, including the first four statistical moments, respectively denoted by Tc, Tw, Ta, and Tk when measured over short 32-ms windows (this will be referred to as short-term moments) and lTc, lTw, lTa, and lTk when measured over long 960-ms windows (long-term moments). The first and second time derivatives of these features are also taken to follow their variation over successive windows. Also, the same moments are computed from the waveform amplitude envelope over 960-ms windows (eTc, eTw, eTa, and eTk). To obtain the amplitude envelope, we first compute the modulus of the complex envelope of the signal, then filter it with a 10-ms length lowpass filter (which is the decreasing branch of a Hanning window).
- *Amplitude Modulation features* (AM), meant to describe the "tremolo" when measured in the frequency range 4–8 Hz, and the "graininess" or "roughness" of the played notes if the focus is put in the range of 10–40 Hz [13]. A set of six coefficients is extracted as described in Eronen's work [13], namely, AM frequency, AM strength, and AM heuristic strength (for the two frequency ranges). Two coefficients are appended to the previous to cope with the fact that an AM frequency is measured systematically (even when there is no actual modulation in the signal). They are the product of tremolo frequency and tremolo strength, as well as the product of graininess frequency and graininess strength.

2) Cepstral Features: Mel-frequency cepstral coefficients (MFCC) are considered as well as their first and second time derivatives [38]. MFCCs tend to represent the spectral envelope over the first few coefficients.

3) Spectral Features: These consist of the following.

- The first two coefficients (except the constant 1) from an auto-regressive (AR) analysis of the signal, as an alternative description of the spectral envelope (which can be roughly approximated as the frequency response of this AR filter).
- A subset of features obtained from the first four statistical moments, namely the *spectral centroid* (Sc), the *spectral width* (Sw), the *spectral asymmetry* (Sa) defined from the spectral skewness, and the *spectral kurtosis* (Sk) describing the "peakedness/flatness" of the spectrum. These features have proven to be successful for drum loop transcription [39] and for musical instrument recognition [24]. Their first and second time derivatives are also computed in order to provide an insight into spectral shape variation over time.
- A precise description of the spectrum flatness, namely MPEG-7 *Audio Spectrum Flatness* (ASF) (successfully used for instrument recognition [24]) and *Spectral Crest Factors* (SCF) which are processed over a number of frequency bands [7].
- *Spectral slope* (Ss), obtained as the slope of a line segment fit to the magnitude spectrum [37], *spectral decrease* (Sd) describing the "decreasing of the spectral amplitude" [37], *spectral variation* (Sv) representing the variation of the spectrum over time [37], *frequency cutoff* (Fc) (frequency rolloff in some studies [37]) computed as the frequency below which 99% of the total spectrum energy is accounted, and an alternative description of the *spectrum flatness* (So) computed over the whole frequency band [37].
- *Frequency derivative of the constant-Q coefficients* (Si), describing spectral "irregularity" or "smoothness" and reported to be successful by Brown [19].
- *Octave Band Signal Intensities*, to capture in a rough manner the power distribution of the different harmonics of a musical sound without recurring to pitch-detection techniques. Using a filterbank of overlapping octave band filters, the log energy of each subband (OBSI) and also

the logarithm of the energy ratio of each subband to the previous (OBSIR) are measured [25].

4) Perceptual Features: *Relative specific loudness* (Ld) representing "a sort of equalization curve of the sound," *sharpness* (Sh) as a perceptual alternative to the spectral centroid based on specific loudness measures, and *spread* (Sp), being the distance between the largest specific loudness and the total loudness [37] and their variation over time, are extracted.

Additionally, a subset of features new to audio classification is examined, namely, *signal to mask ratios* (SMRs) [40]. The idea behind this is to check whether the masking behavior of different sound sources can be used to classify them. We merely use an MPEG-AAC implementation for the computation of the SMR [41]. The computation procedure is briefly described hereafter.

An estimation of the signal power spectral density is obtained and mapped from the linear frequency domain to a *partition* domain, where a partition provides a resolution of almost 1/3 of a critical band. The spectral data is then convolved by a frequency-dependent *spreading function* yielding a *partitioned energy spectrum*. A measure of the *tonality* of the spectral components is then obtained and used to determine an attenuation factor. This attenuation is applied to the *partitioned energy spectrum* to find the *masking threshold* at a specific partition. Finally, the signal to mask ratios are computed for a number of frequency bands (covering the whole frequency range) as the ratio of the spectral energy to the linear-frequency *masking threshold* at each subband.

B. Feature Selection and Transformation

When examining a large set of redundant features for a classification task, feature selection or transformation techniques are essential both to reduce the complexity of the problem (by reducing its dimensionality) and to retain only the information that is relevant in discriminating the possible classes, hence, yielding a better classification performance. To reach this goal, two alternatives exist: either use an orthogonal transform such as PCA or an FSA. In both cases, a set of d features (possibly transformed) are kept from an original set of D candidates ($D \gg d$ in general).

In PCA, the most relevant information is concentrated in the first few components of the transformed feature vectors which correspond to directions of maximum energy [42]. The transformation is performed as follows. The covariance matrix of all training feature vectors is computed and its singular value decomposition (SVD) processed yielding

$$\mathbf{R_x} = \mathbf{UDV}^t$$

where $\mathbf{R_x}$ is the covariance matrix, $\mathbf{U}$ and $\mathbf{V}$ are, respectively, the left and the right singular vector matrices, and $\mathbf{D}$ is the singular value matrix.[3] The PCA transform matrix is then taken to be $\mathbf{W} = \mathbf{V}^t$ and transformed feature vectors are obtained by truncating the vectors $\mathbf{y}_i = \mathbf{W}\mathbf{x}_i$ to dimension d, where $\mathbf{x}_i$ are the original training feature vectors.

A major inconvenience of this approach is that all features must be extracted at the testing stage before the same transform matrix $\mathbf{W}$ (computed during training) is applied to them. The fact is using PCA can be very useful in various analysis to be performed at the training stage where all features are extracted, yet for testing, computing such a large number of features cannot be tolerated due to the extraction complexity. This is why feature selection techniques are often preferred to transform techniques, since only the subset of selected features (which is much smaller than the original set of candidate features) needs then to be extracted for testing.

An efficient FSA is expected to yield the subset of the most relevant and nonredundant d features. Feature selection has been extensively addressed in the statistical machine learning community [43]–[45]. Several strategies have been proposed to tackle the problem that can be classified into two major categories: the "filter" algorithms which use the initial set of features intrinsically, and the "wrapper" algorithms which relate the FSA to the performance of the classifiers to be used. The latter are more efficient than the former, but more complex. We choose to exploit a simple and intuitive "filter" approach called inertia ratio maximization using feature space projection (IRMFSP) which has proven to be efficient for musical instrument recognition [15], [25]. The algorithm can be summarized as follows.

Let M be the number of classes considered, N_m the number of feature vectors accounting for the training data from class C_m and $N = \sum_{m=1}^{M} N_m$. Let $\mathbf{x}_{n_m}(i)$ be the n_mth feature vector (of dimension i) from class C_m, $\mathbf{m}_m(i)$ and $\mathbf{m}(i)$ be, respectively, the mean of the vectors $(\mathbf{x}_{n_m}(i))_{1\le n_m\le N_m}$ of class C_m and the mean of all training vectors $(\mathbf{x}_{n_m}(i))_{1\le n_m\le N_m;1\le m\le M}$. The algorithm proceeds iteratively, selecting at each step i, a subset $\mathbf{X}(i)$ of i features, which is built by appending an additional feature to the previously selected subset $\mathbf{X}(i-1)$. At each iteration

- the ratio
$$r_i = \frac{\sum_{m=1}^{M}(N_m/N)\|\mathbf{m}_m(i) - \mathbf{m}(i)\|}{\sum_{m=1}^{M}((1/N_m)\sum_{n_m=1}^{N_m}\|\mathbf{x}_{n_m}(i) - \mathbf{m}_m(i)\|)}$$
is maximized yielding a new feature subset $\mathbf{X}(i)$,
- the feature space spanned by all observations is made orthogonal to $\mathbf{X}(i)$.

The algorithm stops when i equals the required number of features (d features).

In our particular approach, we proceed to class pairwise feature selection. A different subset of relevant features is found for each pair of classes in the perspective of "a one versus one" classification scheme. Therefore, the output of our FSA is $\binom{M_n}{2}$ selected subsets $\{\mathcal{E}_{i,j}\}_{1\le i<j\le M_n}$ for the M_n classes considered at the node Nn, where $\mathcal{E}_{i,j}$ is the subset of features which is optimal in discriminating the pair $\{\mathrm{C}_i, \mathrm{C}_j\}$. This has proven to be more efficient than classic M_n-class feature selection [25], [46]. In this paper, we use the PCA in the process of building the taxonomy and prefer pairwise IRMFSP for the classification task.

IV. Theoretical Background on Machine Learning

A. Hierarchical Clustering

Our goal is to obtain a taxonomy of musical ensembles and instruments. In other words, one wishes to group together a number of M classes into a number of M_c clusters C_m, $1 \le$

[3]We assume that the singular values are sorted in descending order in $\mathbf{D}$ so that the top values correspond to the greatest values.

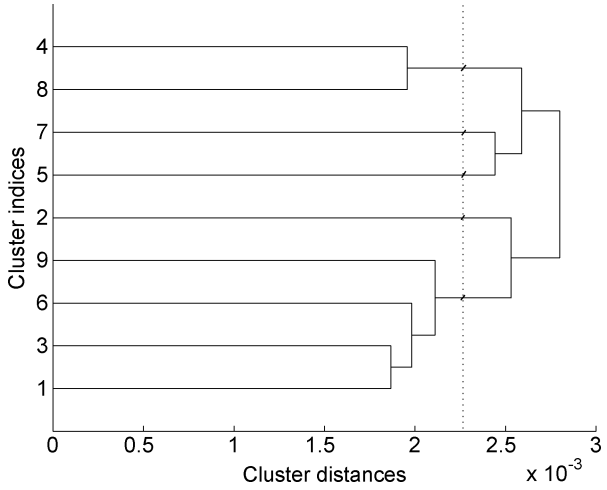

Fig. 2. Example of a dendrogram.

$m \leq M_c$, within L levels of a hierarchical taxonomy to be determined.

To this end, we exploit the family of hierarchical clustering algorithms producing "a hierarchy of nested clusterings" [47]. The agglomerative version of such algorithms starts with as many clusters as original classes ($M_c^1 = M$ at iteration 1), measuring the proximities J_{ij} between all pairs of clusters $\{\mathrm{C}_i, \mathrm{C}_j\}$ and grouping together the closest pairs into new clusters to produce M_c^l new ones at iteration l, until all classes lie in a single cluster (at iteration M).

A convenient way to understand the result of such a procedure is to represent it as a graph (called *dendrogram*) which depicts the relations and proximities between the nested clusters obtained. An example is given in Fig. 2. Clusters linked together into new ones at higher levels are linked with U-shaped lines. Original cluster indices are given along the vertical axis, while the values along the horizontal axis represent the distances between clusters. The distance between two clusters C_p and C_q is measured as the average distance between all pairs of classes in C_p and C_q. For example, the given dendrogram tells us that the original classes C_1 and C_3 are linked together into a new cluster which is linked to the class C_6.

The relevance of the cluster tree obtained can be measured by the *cophenetic correlation coefficient*. This coefficient correlates the distances J_{ij} between any two initial clusters (i.e., original classes) C_i and C_j to the *cophenetic distances* δ_{ij}, i.e., the distances between the two clusters C_p and C_q containing these two classes and linked together at some level of the hierarchy. For example, the cophenetic distance between C_1 and C_6 is the distance between clusters C_6 and C_{13}, where C_{13} is the cluster containing C_1 and C_3. The cophenetic correlation coefficient is defined as

$$c = \frac{\sum_{i<j}(J_{ij} - \bar{J})(\delta_{ij} - \bar{\delta})}{\sqrt{\sum_{i<j}(J_{ij} - \bar{J})^2 \sum_{i<j}(\delta_{ij} - \bar{\delta})^2}} \tag{1}$$

where $\bar{J}$ and $\bar{\delta}$ are, respectively, the means of J_{ij} and δ_{ij}, $1 \leq i < j \leq M$. The closer the cophenetic coefficient is to 1, the more relevantly the cluster tree reflects the structure of the data.

Clustering is then obtained by cutting the dendrogram at a certain level or certain value of the horizontal axis. For example, the vertical dotted line shown in Fig. 2 produces five clusters. Thus, one can obtain the number of desired clusters merely by adjusting the position of this vertical line.

The choice of the "closeness" criterion, i.e., the distance J_{ij}, to be used for clustering is critical. One needs a robust distance which is expected to reduce the effect of noisy features. Also, such a distance needs to be related to the classification performance. A convenient and robust means to measure the "closeness" or separability of data classes is to use probabilistic distance measures, i.e., distances between the probability distributions of the classes [47], [48]. This is an interesting alternative to classic Euclidean distance between feature vectors known to be suboptimal for sound source classification. Many such distances have been defined in various research areas [49]. We choose to consider the Bhattacharryya and divergence distances for our study to obtain two different interpretations. This choice is also guided by the resulting simplification in the computations.

The divergence distance J_D between two probability densities p_1 and p_2 is defined as

$$J_D(p_1, p_2) = \int_{\mathbf{x}} [p_1(\mathbf{x}) - p_2(\mathbf{x})] \log \frac{p_1(\mathbf{x})}{p_2(\mathbf{x})} d\mathbf{x}. \tag{2}$$

The Bhattacharryya distance is defined as

$$J_B(p_1, p_2) = -\log \left(\int_{\mathbf{x}} [p_1(\mathbf{x}) p_2(\mathbf{x})]^{\frac{1}{2}} \, d\mathbf{x} \right). \tag{3}$$

If the class data can be considered as Gaussian, the above distances admit analytical expressions and can be computed according to

$$J_D(p_1, p_2) = \frac{1}{2}(\mu_1 - \mu_2)^T \left(\Sigma_1^{-1} + \Sigma_2^{-1}\right)(\mu_1 - \mu_2) + \frac{1}{2}\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1 - 2I_D\right) \tag{4}$$

and

$$J_B(p_1, p_2) = \frac{1}{8}(\mu_1 - \mu_2)^T \left[\frac{1}{2}(\Sigma_1 + \Sigma_2)\right]^{-1} (\mu_1 - \mu_2) + \frac{1}{2}\log \frac{\left|\frac{1}{2}(\Sigma_1 + \Sigma_2)\right|}{|\Sigma_1|^{\frac{1}{2}}|\Sigma_2|^{\frac{1}{2}}} \tag{5}$$

where (μ_1, Σ_1) and (μ_2, Σ_2) are the mean vectors and the covariance matrices of the multivariate Gaussian densities describing, respectively, class 1 and class 2 in $\mathbb{R}^D$. Nevertheless, it would be highly suboptimal, in our case, to assume that the original class observations follow Gaussian distributions since we deal with data with a nonlinear structure. Moreover, if the class probability densities are not Gaussian, computing such distances is unfortunately a difficult problem since it requires heavy numerical integrations.

In order to alleviate this problem, we follow Zhou's and Chellapa's approach [49] which exploits kernel methods [50]. Their idea is to map the data from the original space to a *transformed* nonlinear space called reproducing kernel Hilbert space (RKHS), where the probability distributions of the data can be assumed to be Gaussian. A robust estimation of the probabilistic distances needed can then be derived using expressions (4) and (5) provided that a proper estimation of the means and covariance matrices in the RKHS can be obtained.

The strength of such an approach is that there is no need for knowing explicitly either the structure of the original probability densities or the nonlinear mapping to be used. In fact, it is shown that all computations can be made using the so-called *kernel trick*. This means that the function ϕ which maps the original D-dimensional feature space to a F-dimensional transformed feature space does not need to be known as long as one knows the *kernel* function k which returns the dot product of the transformed feature vectors, according to

$$k(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n).\phi(\mathbf{x}_m); \quad \mathbf{x}_n, \mathbf{x}_m \in \mathbb{R}^D \text{ and } \phi(\mathbf{x}_m), \phi(\mathbf{x}_n) \in \mathbb{R}^F.$$

In order to obtain expressions of the required distances (4) and (5) in RKHS, Zhou and Chellappa exploit the maximum likelihood estimates of the means and covariances in $\mathbb{R}^F$ based on N given observed feature vectors $\mathbf{x}_n \in \mathbb{R}^D$

$$\mu_i = \frac{1}{N}\sum_{n=1}^{N} \phi(\mathbf{x}_n), \tag{6}$$

$$\Sigma_i = \frac{1}{N}\sum_{n=1}^{N} \left(\phi(\mathbf{x}_n) - \mu_i\right)\left(\phi(\mathbf{x}_n) - \mu_i\right)^T. \tag{7}$$

The main difficulty arise from the fact that the covariance matrix Σ_i needs to be inverted while it is rank-deficient since $F \gg N$. Thus, the authors have obtained a proper invertible approximation of Σ_i and expressions of the distances which can be computed using only the knowledge of the kernel k. The computation procedure of these distances is given in the Appendix.

B. Support Vector Machines

Support vector machines (SVMs) are powerful classifiers arising from structural risk minimization theory [51] that have proven to be efficient for various classification tasks, including speaker identification, text categorization, face recognition, and, recently, musical instrument recognition [23], [24], [52]. These classifiers present the advantage of being *discriminative* by contrast to *generative* approaches (such as Gaussian mixture models) assuming a particular form for the data probability density (often not consistent) and have very interesting generalization properties [53].

SVMs are by essence binary classifiers which aim at finding the hyperplane that separates the features related to each class C_i with the maximum margin. Formally, the algorithm searches for the hyperplane $\mathbf{w}.\mathbf{x} + b = 0$ that separates the training samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ which are assigned labels $y_1, \ldots, y_n$ ($y_i \in \{-1, 1\}$) so that

$$y_i(\mathbf{x}_i.\mathbf{w} + b) - 1 \geq 0, \quad \forall i \tag{8}$$

under the constraint that the distance $2/||\mathbf{w}||$ between the hyperplane and the closest sample is maximal. Vectors for which the equality in (8) holds are called support vectors.

In order to enable nonlinear decision surfaces, SVMs map the D-dimensional input feature space into a higher dimension space where the two classes become linearly separable, using a kernel function. A test vector $\mathbf{x}$ is then classified with respect to the sign of the function

$$f(\mathbf{x}) = \sum_{i=1}^{n_s} \alpha_i y_i k(\mathbf{s}_i, \mathbf{x}) + b$$

where $\mathbf{s}_i$ are the support vectors, α_i are Lagrange multipliers, and n_s is the number of support vectors. Interested readers are referred to Schölkopf's and Smola's book [50] or Burges' tutorial [53] for further details.

SVMs can be used to perform M_n-class classification using either the "one versus one" or "one versus all" strategies. In this paper, a "one versus one" strategy (or class pairwise strategy) is adopted. This means that as many binary classifiers as possible class pairs $\{\mathrm{C}_i, \mathrm{C}_j\}_{1 \leq i < j \leq M_n}$ are trained (i.e., $\binom{M_n}{2}$ classifiers). A given test segment is then classified by every binary classifier, and the decision is generally taken by means of a "majority-vote" procedure applied over all possible pairs. Such a decision strategy presents the drawback that any postprocessing is limited, as no class membership probabilities are obtained, in addition to the fact that when some classes receive the same greatest number of votes, the winning class is indeterminate. In order to remedy these shortcomings, we adopt Platt's approach [54] which derives posterior class probabilities after the SVM. The first step consists in fitting sigmoid models to the posteriors $\mathrm{Prob}(y = 1|f)$ according to

$$\mathrm{Prob}(y = 1|f) = \frac{1}{1 + \exp(Af + B)} \tag{9}$$

where A and B are parameters to be determined. Platt discusses the appropriateness of this model and proposes a model-trust minimization algorithm to determine optimal values of the two parameters.

Once this is done for every pair of classes, one is confronted with the problem of coupling the pairwise decisions so as to get class membership probabilities. This issue was addressed by Hastie and Tibshirani [55] who formalized a method to perform optimal coupling. Assuming the model $\mu_{ij} = p_i/p_i + p_j$, with $p_i = \mathrm{Prob}(\mathrm{C}_i)$, for the probability $r_{ij} = \mathrm{Prob}(\mathrm{C}_i|\mathrm{C}_i \text{ or } \mathrm{C}_j)$ estimated for each pair $\{\mathrm{C}_i, \mathrm{C}_j\}_{1 \leq i < j \leq M_n}$ at a given observation $\mathbf{x}_t$, an estimate of the probability vector $\mathbf{p}(\mathbf{x}_t) = (p_1(\mathbf{x}_t), p_2(\mathbf{x}_t), \ldots, p_{M_n}(\mathbf{x}_t))$ is deduced by means of a gradient approach using the average Kullback–Leibler distance between r_{ij} and μ_{ij} as a closeness criterion. Classification can then be made using the usual maximum *a posteriori* probability (MAP) decision rule [48].

V. Experimental Validation of the Proposed Architecture

We choose to test our system with jazz music ensembles from duets to quartets. This choice is motivated by the diversity found in this music genre which is thought to be representative of a

TABLE I
SOUND DATABASE USED. "TRAIN SOURCES" AND "TEST SOURCES" ARE, RESPECTIVELY, THE NUMBER OF DIFFERENT SOURCES (DIFFERENT MUSIC ALBUMS) USED (0.5 MEANS THAT THE SAME SOURCE WAS USED IN THE TRAIN AND TEST SETS), "TRAIN" AND "TEST" ARE, RESPECTIVELY, THE TOTAL LENGTHS (IN SECONDS) OF THE TRAIN AND TEST SETS. SEE TABLE II FOR THE INSTRUMENT CODES

Ensembles	Train sources	Train	Test sources	Test
BsDr	1	411	4	302
BsDrPn	11	955	1	346
BsDrPnTr	3	310	1	219
BsDrPnTs	1	339	2	90
BsDrPnVf	4	600	1	266
BsDrPnVm	0.5	172	0.5	86
BsDrTr	1	181	1	134
BsDrTs	1	82	1	71
BsEgPn	1	151	1	42
BsPn	6	747	1	666
BsPnVm	1	346	1	32
DrGtPrVm	0.5	138	0.5	69
EgVf	1	628	2	196
GtVf	2	159	1	169
PnTr	0.5	398	0.5	199
PnVf	6	436	1	186
PnVm	1	281	1	166
Piano	15	1108	3	750
DoubleBass	1	134	5	93
Drums	2	236	4	213

large variety of musical compositions. It is believed that the same approach could be easily followed for any other genre (provided that the timbre of the instruments has not been seriously modified by audio engineering effects/equalization). Particularly, we consider ensembles involving any of the following instruments: double bass, drums, piano, percussion, trumpet, tenor sax, electroacoustic guitar, and Spanish guitar. Also, female and male singing voices are considered as possible "instruments."

A. Sound Database

A major difficulty in assembling a sound database for the envisaged architecture is having to manually annotate the musical segments, each with a different combination of instruments. In fact, in a double bass, piano, and drums trio, for example, some segments may involve only piano, only drums, or only double bass and drums. A critical aspect of such annotations is related to the precision with which the human annotators perform the segmentation. Clearly, it is not possible to segment the music at the frame rate (the signal analysis is 32-ms frame based); hence, it is necessary to decide which minimal time horizon should be considered for the segmentation. In order to make a compromise between time precision and annotation tractability, a minimum length of 2 s is imposed to the segments to be annotated, in the sense that a new segment is created if it involves a change in the orchestration that lasts at least 2 s.

Table I sums up the instrument combinations for which sufficient data could be collected (these are the classes to be recognized, see Table II for the instrument codes). A part of the sounds was excerpted from both live and studio commercial recordings (mono-encoded either in PCM or 64 kb/s mp3 formats). Another part was obtained from the RWC jazz music database [56].

There is always a complete separation of training and test data sets (different excerpts are used in each set) and also a complete separation, in most cases, between the sources[4] providing the training data and those providing the test data. Almost 2/3 of the sounds were included in the training set and the remaining 1/3 in the test set whenever this was consistent with the constraint that train and test sources be distinct (when more than one source was available). When only two sources were available, the longest source was used for training and the shortest for testing. Thus, important variability is introduced in the data to test for the generalization ability of the system.

TABLE II
INSTRUMENT CODES

Bs	double bass
Dr	drums
Eg	electro-acoustic guitar
Gt	Spanish guitar
Pn	piano
Pr	percussion
Tr	trumpet
Ts	tenor sax
Vf	female voice
Vm	male voice
V	voice
W	wind instrument
M	V, W or Eg

Note that, given the annotation procedure, one should expect a great number of outliers among different sets. Typically, many segments annotated as double bass, drums, piano, and tenor sax (BsDrPnTs), surely contain many frames of the class double bass, drums, and piano (BsDrPn).

B. Signal Processing

The input signal is down-sampled to a 32-kHz sampling rate. The mean of each signal is estimated (over the total signal duration) and subtracted from it. Its amplitude is normalized with respect to its maximum value. All spectra are computed with a fast Fourier transform after Hamming windowing. Silence frames are detected automatically thanks to a heuristic approach based on power thresholding then discarded from both train and test data sets.

C. Computer-Generated Taxonomy

Since building the taxonomy is an operation that is done "once and for all" at the training stage, one can use all the candidate features and exploit PCA to reduce the dimension of the feature space (see Section III-B). A dimension of 30 was considered as sufficient (94% of the total variance was thus retained).

Computing the probabilistic distances in RKHS (to be used for clustering) requires an Eigen value decomposition (EVD) of $N_i \times N_i$ Gram matrices, where N_i is the number of training feature vectors of class C_i (see Appendix). Such an operation is computationally expensive $(O(N_i^3))$ since N_i is quite large. Hence, the training sets are divided into smaller sets of 1500 observations and the desired distances are obtained by averaging the distances estimated using all these sets. To measure these distances, one needs to choose a

[4]A *source* is a music recording such that different sources constitute different music albums featuring different artists.

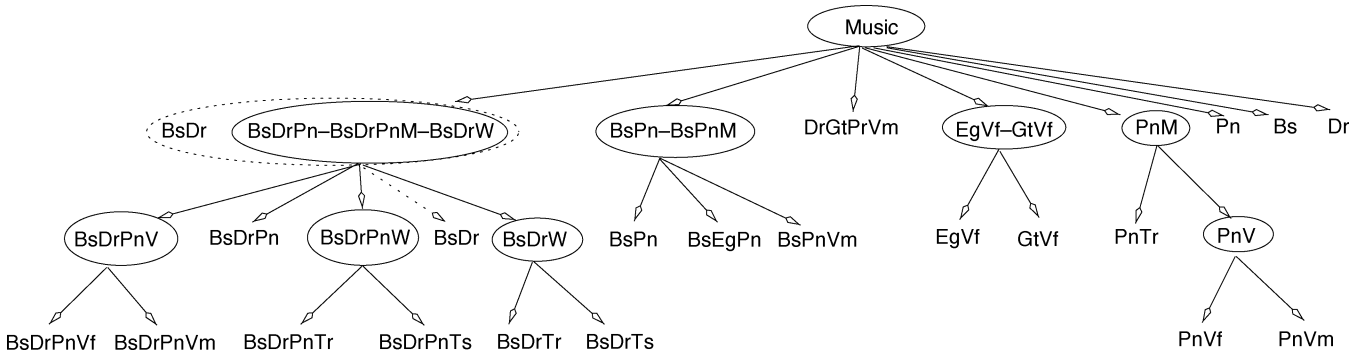

Fig. 3. Taxonomy obtained with hierarchical clustering using probabilistic distances in RKHS.

kernel function. We use the radial basis function (RBF) kernel $k(\mathbf{x},\mathbf{y}) = \exp(-(||\mathbf{x}-\mathbf{y}||^2/2\sigma^2))$, with $\sigma = 0.5$.

As mentioned in Section IV-A, the relevancy of the hierarchical clustering output can be evaluated using the cophenetic correlation coefficient which is expected to be close to 1. In our experiments, it was found that greater cophenetic correlation coefficients could be obtained, i.e., more relevant clustering, if the "solo" classes (piano, drums, and double bass) were not considered in the process of hierarchical clustering of ensembles. Hence, clustering was performed on all classes except solo piano, solo drums, and solo double bass, using both the Bhattacharryya and the divergence distances in RKHS. The value of the cophenetic correlation coefficient obtained with the Bhattacharryya distance is 0.85 against 0.97 with the divergence. Therefore, it can be deduced that efficient hierarchical clustering of the ensembles was achieved using the divergence distance.

We then varied the number of clusters from 4 to 16 with a step of 2 by applying different cuts to the dendrogram. The levels of the hierarchical taxonomy are to be deduced from these alternative clusterings in such a way that the high levels are deduced from "coarse" clustering (low number of clusters) while the low levels are deduced from "finer" clustering (higher number of clusters). The choice of relevant levels is guided by "readability" considerations so that clusters are associated with labels that can be easily formulated by humans. Also, the maximum number of levels in the hierarchy is constrained to three to reduce the system complexity.

Taking these considerations into account, the levels deduced from clustering with 6, 12, and 16 clusters were retained resulting in the taxonomy depicted in Fig. 3 where solos were merely put into three supplementary clusters at the highest level. Preliminary testing showed that better classification of BsDr was achieved if it was associated with the first cluster (BsDrPn-BsDrPnM-BsDrW). This was considered as acceptable since the label of the new cluster (BsDr-BsDrPn-BsDrPnM-BsDrW) became more convenient as it could be easily described as "music involving at least double bass and drums." In fact, all the clusters obtained carry convenient labels that can be formulated intuitively.

D. Features Selected

As mentioned earlier, feature selection is preferred to PCA for classification to reduce the computational load at the test stage. Consequently, only the most relevant features (selected by the FSA) are extracted during testing phase, hence, useless ones (not selected by the FSA) among all the candidates considered at the training phase are not computed.

Pairwise IRMFSP feature selection is performed at each node of the taxonomy yielding subsets of selected features specifically adapted to the context (see Section III-A). Note that, at each node, a different subset of features is selected for each pair of classes. For example, at the node (BsPn-BsPnM), three optimal sets are fetched for the three biclass problems (BsPn)/(BsEgPn), (BsPn)/(BsPnVm), and (BsEgPn)/(BsPnVm). Similarly, ten optimal subsets are selected at the node (BsDr-BsDrPnV-BsDrPn-BsDrPnW-BsDrW) (targeting the ten binary combinations of these classes) and 28 subsets at the highest level. The total number of subsets optimized over all the nodes of the taxonomy is, thus, 47.

Table III lists all the features extracted (described in Section III-A). They are organized in "feature packets." The number of feature coefficients for each feature packet is given in column two. It is worth mentioning that no distinction between "feature" and "feature coefficient" is made. For example, the third MFCC coefficient MC3 is a feature and so is Fc. The total number of candidate features is then 355. Fifty of them are selected using the IRMFSP algorithm for each pair of classes. The more frequently some features are selected the more useful they are. Column three of Table III indicates, among each packet, the features that were the most frequently selected over the 47 pairwise optimized subsets.

The most successful features are SMR coefficients (24 of them were selected on average over the 47 subsets). These features which have not been used in previous work on audio classification turn out to be useful. Though interpreting this result is not very intuitive, it can be deduced that the masking effects of different sound sources seem to be specific enough to enable their discrimination. The other efficient perceptual features are the relative specific loudness, particularly in the high frequency Bark bands and the sharpness.

As far as spectral features are concerned, those deduced from the spectral moments (spectral centroid (Sc), width (Sw), asymmetry (Sa), and kurtosis (Sk)) as well as spectral decrease (Sd) and full-band spectral flatness (So) are found to be more useful than the others.

Both long-term and short-term temporal moments are found to be efficient. Moreover, the variation of the temporal kurtosis

TABLE III
FEATURE PACKETS AND FEATURES MOST FREQUENTLY SELECTED AMONG EACH PACKET. THE FRACTIONS IN PARENTHESES INDICATE THE NUMBER OF CLASS PAIRS (AMONG ALL POSSIBLE) FOR WHICH THE GIVEN FEATURES WERE SELECTED

Feature packet	Size	Most frequently selected among each packet
AC=[A1,...,A49]	49	AC49 (4/47)
Z=[ZCR,lZCR]	2	ZCR (9/47), lZCR (7/47)
Tx=[Tc,Tw,Ta,Tk]+δ+δ^2	12	Tw (24/47), Tk (22/47)
lTx=[lTc,lTw,lTa,lTk]+δ+δ^2	12	lTc (20/47), lTw (27/47), lTk (23/47), δlTk (17/47)
Ex=[eTc,eTw,eTa,eTk]+δ+δ^2	12	eTw (24/47), eTk (23/47), δ^2eTk (14/47)
AM=[AM1,...,AM8]	8	strength of AM in 10-40 Hz (8/47)
MFCC=[MC1,...,MC10]+δ+δ^2	30	MC1 (28/47), MC3 (23/47)
AR=[AR1,AR2]	2	AR1 (15/47), AR2 (14/47)
Sx=[Sc,Sw,Sa,Sk]+δ+δ^2	12	Sc (29/47), Sw (24/47), Sa (28/47), Sk (34/47)
ASF= [A1,...,A23]	23	A22 (13/47)
SCF=[SCF1,...,SCF23]	23	SCF22 (7/47)
[Ss,Sd,Sv,So,Fc]	5	Sd (17/47), So (22/47), Fc (14/47)
Si=[Si1,...,Si21]	21	Si1 (13/47)
OBSI=[O1,...,O8]	8	O3 (8/47), O8 (7/47), O7 (6/47)
OBSIR=[OR1,...,OR7]	7	OR3 (9/47)
Ld=[L1,...,L24]+δ+δ^2	72	L24 (31/47)
[Sh,Sp]+δ+δ^2	6	Sh (30/47), Sp (9/47)
SMR=[S1,...,S51]	51	S38,S51 (31/47), S15,S21 (29/47), S1 (28/47) S19,S29,S41,S43,S46, (27/47)

TABLE IV
CONFUSION MATRIX AT THE FIRST LEVEL (TOP LEVEL)

	C1	C2	C3	C4	C5	C6	C7	C8
C1:BsDr-BsDrPn-BsDrPnM-BsDrW	**91**	1	0	0.	5	2	0.	0.
C2:BsPn-BsPnM	4	**83**	0	0	1	3	0	10
C3:DrGtPrVm	*29*	*3*	***63***	*6*	*0*	*0*	*0*	*0*
C4:EgVf-GtVf	19	2	0	**60**	18	1	0	0
C5:PnM	26	1	2	11	**55**	4	0	0
C6:Pn	0	2	0	0	15	**83**	0	0
C7:Dr	**61**	0	0	0	5	0	34	0
C8:Bs	*0*	*44*	*0*	*0*	*0*	*2*	*0*	***54***

over successive frames is frequently selected to describe the variation of the transients of the audio signal, which is not surprising when dealing with sounds involving percussive instruments.

Finally, only a few cepstral coefficients (or none for some pairs) were selected in the presence of the other features, which confirms that it is possible to circumvent this popular set for sound recognition tasks. The remaining features were selected marginally for specific class pairs where they could improve the separability.

The subsets of selected features for each pair of classes were posted on the web[5] for interested readers to look into it in depth.

E. Classification

Classification is performed using a "one versus one" SVM scheme with the RBF kernel and based on the pairwise selected features (described in V-D). Recognition success is evaluated over a number of decision windows. Each decision window combines elementary decisions taken over T consecutive short analysis windows. The recognition success rate is then, for each class, the percentage of successful decisions over the total number of available decision windows. In our experiment, we use $T = 120$ corresponding to 2-s decisions.

Since short-time decisions can be taken ($\approx$ 2 s), the proposed system can be easily employed for the segmentation of musical ensembles (duos, trios, and quartets). By combining the decisions given over 2-s windows, it is easy to define the segments where each instrument or group of instruments is played.

We present the confusion matrices obtained with our system in Tables IV–VI, respectively, for the first (highest), second, and third (bottom) levels of the hierarchical taxonomy described earlier. The rates presented in parentheses are the ones corresponding to the absolute accuracy (from top to bottom) found by multiplying the recognition accuracy at the current node by the recognition accuracies of the parent nodes which are crossed following the path from the root of the tree to the current node. This path is found by crossing at each level the most probable node.

Some results should be considered as preliminary since we, unfortunately, lacked enough test material for some classes. Consequently, the results for classes for which test data size was less than 200 s are given in italic characters to warn about their statistical validity.[6]

Starting with the first level, the results obtained can be considered as very encouraging given the short decision lengths and the high variability in the recordings. The average accuracy is 65%. For the class C1 (BsDr-BsDrPn-BsDrPnM-BsDrW), 91% accuracy is achieved, while the class C7 (drums) is successfully identified only 34% of the time. The drums were classified as C1 61% of the time. Alternative features should be introduced to improve the discrimination of these two classes. For example, features describing the absence of harmonicity could be efficient in this case since percussive sounds like drums do not present a

[5][Online]. Available: http://www.tsi.enst.fr/%7Eessid/pub/ieee-sa-fsa.html

[6]Work has been undertaken to assemble more data for further statistical validation.

TABLE V
CONFUSION MATRIX AT THE SECOND LEVEL, USING TWO DIFFERENT DECISION STRATEGIES AT THE NODES N1 AND N2. TOP-TO-BOTTOM ABSOLUTE ACCURACY IN PARENTHESES

Node N1	C1.1		C1.2		C1.3		C1.4		C1.5	
	MAP	Heurist	MAP	Heurist	MAP	Heurist	MAP	Heurist	MAP	Heurist
C1.1:BsDrPnV	35 (32)	**46 (42)**	**50**	17	10	32	5	5	0	0
C1.2:BsDrPn	0	1	**100 (91)**	**72 (66)**	0	27	0	0	0	0
C1.3:BsDrPnW	0	0	**92**	50	8 (7)	**50 (46)**	0	0	0	0
C1.4:BsDrW	0	0	0	0	0	0	49 (45)	**79 (72)**	**51**	21
C1.5:BsDr	13	15	8	5	0	1	0	7	**79 (72)**	**72 (66)**

Node N2	C2.1		C2.2		C2.3	
	MAP	Heurist	MAP	Heurist	MAP	Heurist
C2.1:BsPn	**99 (82)**	**94 (78)**	0	5	1	1
C2.2:BsEgPn	***57***	*43*	*33 (27)*	***48 (40)***	*10*	*10*
C2.3:BsPnVm	*0*	*0*	*0*	*0*	***100 (83)***	***100 (83)***

Node N4	C4.1	C4.2
C4.1:EgVf	***100 (60)***	*0*
C4.2:GtVf	***100***	*0 (0)*

Node N5	C5.1	C5.2
C5.1:PnTr	**100 (55)**	0
C5.2:PnV	0	**100 (55)**

TABLE VI
CONFUSION MATRIX AT THE THIRD LEVEL (BOTTOM LEVEL). TOP-TO-BOTTOM ABSOLUTE ACCURACY IN PARENTHESES

Node N1.1	C1.1.1	C1.1.2
C1.1.1 :BsDrPnVf	**87 (37)**	13
C1.1.2 :BsDrPnVm	28	**72 (30)**

Node N1.3	C1.3.1	C1.3.2
C1.3.1 :BsDrPnTr	**100 (46)**	0
C1.3.2 :BsDrPnTs	29	**71 (33)**

Node N1.4	C1.4.1	C1.4.2
C1.4.1 :BsDrTr	***100 (72)***	0
C1.4.2 :BsDrTs	9	***91 (66)***

Node N5.2	C5.2.1	C5.2.2
C5.2.1 :PnVf	***97 (53)***	3
C5.2.2 :PnVm	28	**72 (40)**

strong harmonicity. In general, most classes were mainly confused with C1 except the class C6 (piano). This is an interesting result: it is easy to discriminate the piano played solo and the piano played with accompaniment (83% for the piano versus 91% for C1). The piano was more often confused with the class C5 (PnTr-PnV)- 15% of the time- than with C1.

At the second level, poor results are found at the node N1 when using the traditional MAP decision rule (column labeled MAP). In fact, BsDrPnW is successfully classified only 8% of the time, and BsDrPnV 35% of the time, as they are very frequently confused with BsDrPn, respectively, 92% of the time and 50% of the time. Similarly, BsDrW is confused with BsDr 51% of the time. This is not surprising given the sound database annotation constraints mentioned in Section V-A. In fact, many BsDrPn frames necessarily slipped into BsDrPnV and BsDrPnW training and test data. Also, many BsDrW segments contain BsDr. Fortunately, by exploiting a heuristic to modify the decision rule, one can easily remedy these deficiencies. The fact is that for the pairs BsDr versus BsDrW, BsDrPn versus BsDrPnW, and BsDrPn versus BsDrPnV, the optimal decision surfaces are biased due to the presence of outliers both in the training and test sets. As an alternative to outlier removal techniques [57], which can be inefficient in our context due to the presence of a very high number of outliers, we use a biased decision threshold in this case. Every time a test segment is classified as BsDr using the MAP criterion, if the second most probable class is BsDrW, we review the decision by considering only the output of the BsDr/BsDrW classifier. Then following two actions are taken.

- First, we classify single frames as BsDr only if $\text{Prob}(\text{BsDr}|\text{BsDr or BsDrW}) > 0.8$, instead of using usual Bayes' threshold of 0.5.
- Second, we count the number of frames classified as BsDr within the decision window (120 consecutive frames) and decide for this class only if 2/3 of the frames involved in the decision window carry this label, otherwise the current 2-s segment is classified as BsDrW.

The same is done for the pairs involving BsDrPn, as well as for BsPn versus BsEgPn at the node N2. As a result, on average,

more successful classification is achieved in these contexts as can be seen in columns labeled "Heurist" of Table V.

Finally, successful recognition of four instruments playing concurrently can be achieved as can be seen in Table VI. Since the excerpts used in our experiments translate significantly different recording conditions (both live and studio music was included) and since some of these excerpts were mp3-compressed (which can be considered as imperfect signals corrupted by quantization noise and with bandwidth limitation), we feel confident about the applicability of our approach to other musical genres. The system seems to be able to cope with varying balance in the mix as it is able, for example, to successfully recognize the BsDrPn mixture both over piano solo passages (piano louder than double bass and drums) and over double bass solo passages (double bass louder than piano and drums).

A baseline "flat" (i.e., one level, nonhierarchical) system has been built to assess the consistency of the proposed classification scheme. Let us, however, emphasize that such a baseline system is not generalizable to more realistic situations where many more instruments; hence, a very high number of instrument-combinations are found (see Section II). It is also important to note that our goal is not to prove that hierarchical classification performs better than flat classification,[7] but rather to propose a whole framework enabling to tackle the classification of a potentially very high number of arbitrary instrument mixtures.

20-class IRMFSP feature selection was used for the baseline system yielding 50 selected features. For classification, classic Gaussian mixture models (GMM) [48] were exploited with 16 component densities per class. The classification results found with this system are presented in column two of Table VII against the performance of our proposal (column three). The latter achieves better individual classification performance in most cases and the average accuracy is also higher (+6%). Note that better hierarchical classification results could be obtained at the intermediate and leaf nodes using a more elaborate hierarchical classification strategy than choosing at each level the most probable node. This causes the recognition accuracy to be the product of the recognition accuracy at each node from the top level to the lowest level, and, hence, can be suboptimal since it is then impossible to recover from errors made at the roots. Alternative techniques such as *beam search* can highly improve the final classification performance [60].

TABLE VII
PERFORMANCE OF PROPOSED SYSTEM VERSUS THE REFERENCE SYSTEM

Classes	Baseline	Proposal
Pn	67	83
Bs	*39*	*54*
Dr	73	34
BsDr	48	66
BsDrPn	4	66
BsDrPnTr	6	46
BsDrPnTs	82	33
BsDrPnVf	0	37
BsDrPnVm	91	30
BsDrTr	*94*	*72*
BsDrTs	*23*	66
BsEgPn	*53*	40
BsPn	85	78
BsPnVm	*0*	*83*
DrGtPrVm	*100*	*63*
EgVf	*50*	*60*
GtVf	*8*	*0*
PnTr	78	55
PnVf	*20*	*53*
PnVm	*9*	*40*
Average	47	53

VI. CONCLUSION

We presented a new approach to instrument recognition in the context of polyphonic music where several instruments play concurrently. We showed that recognizing classes consisting of combinations of instruments played simultaneously can be successful using a hierarchical classification scheme and exploiting realistic musical hypotheses related to genre and orchestration.

The hierarchical taxonomy used can be considered efficient since

- it was found automatically thanks to a clustering approach based on robust probabilistic distances;
- it can be interpreted easily by humans in the sense that all nodes carry musically meaningful labels enabling useful intermediate classification.

A major strength of the chosen approach is that it frees one from the burden of performing multipitch estimation or source separation. On the contrary, our system may help addressing these issues as efficient segmentation of the music can be achieved with respect to the instruments played. It may also be used to identify the number of playing sources. This may provide source separation systems with an insight into which pitches to look for.

Additionally, we studied the usefulness of a wide selection of features for such a classification task, including new proposals. An interesting result is that perceptual features, especially signal to mask ratios are efficient candidates.

More successful recognition could be achieved using longer decision windows. It is believed that our proposal is amenable to many useful applications accepting realistic MIR user queries since it can potentially process any musical content regardless of the orchestration (possibly involving drums and singing voice). In particular, our approach could be very efficient in recognizing more coarsely the orchestrations of musical pieces without necessarily being accurate about the variation of the instruments played within the same piece. In fact, decision rules could be adapted very easily to give the right orchestration label for the whole piece as will be discussed in future work.

APPENDIX

Computation of Probabilistic Distances in RKHS

Let N_i be the number of observations for class C_i, let $\Phi_i = [\phi_1, \ldots, \phi_{N_i}]$, with $\phi_n = \phi(\mathbf{x}_n)$, let $\mathbf{s}_i$ be a N_i-length column vector such that $\mathbf{s}_i = (1/N_i)\mathbf{1}$, with $\mathbf{1}$ a vector of ones, let $\mathbf{K}_i = \Phi_i^T \Phi_i$ ($\mathbf{K}_i$ is called a Gram matrix and can be computed using the kernel trick), let $\mathbf{J}_i = (1/\sqrt{N_i})(\mathbf{I}_{N_i} - \mathbf{s}_i \mathbf{1}^T)$, and $\bar{\mathbf{K}}_i =$

[7]This issue has been addressed in previous works on music classification, and the fact that hierarchical systems are more efficient than flat systems tends to be acknowledged [15], [58], [59].

$\mathbf{J}_i^T\mathbf{K}_i\mathbf{J}_i$. The top r_i eigenpairs of the matrix $\bar{\mathbf{K}}_i$ are denoted by $\{(\lambda_{n,i},\mathbf{v}_{n,i})\}_{n=1}^{r_i}$, $\mathbf{V}_{r_i,i}=[\mathbf{v}_{1,i},\ldots,\mathbf{v}_{r_i,i}]$ and $\Lambda_{r_i,i}$ is the diagonal matrix whose diagonal elements are $\{\lambda_{1,i},\ldots,\lambda_{r_i,i}\}$ (r_1 and r_2 are to be chosen and are such that $r_i \ll N_i \ll F$, $r_1 = r_2 = 10$ was found to be a good choice on our data). Let $\mathbf{K}_{ij}=\Phi_i^T\Phi_j$ (can be computed using the kernel trick), $\mathbf{A}_i=\mathbf{J}_i\mathbf{J}_i^T$ and

$$\mathbf{B}_j=\mathbf{J}_j\mathbf{V}_{r_j,j}\Lambda_{r_j,j}^{-1}\mathbf{V}_{r_j,j}^T\mathbf{J}_j^T \tag{10}$$

then the approximation of the divergence distance in RKHS is expressed as

$$\hat{J}_D(p_1,p_2)=\hat{J}_R(p_1\|p_2)+\hat{J}_R(p_2\|p_1) \tag{11}$$

where

$$\begin{aligned}\hat{J}_R(p_1\|p_2)=\frac{1}{2}\Big\{&\hat{\theta}_{121}+\hat{\theta}_{222}-\hat{\theta}_{122}-\hat{\theta}_{221}\\&+tr\left[\Lambda_{r_1,1}\right]-\hat{\eta}_{12}\Big\}\end{aligned} \tag{12}$$

$$\hat{\theta}_{ijk}=\mathbf{s}_i^T\mathbf{K}_{ik}\mathbf{s}_k-\mathbf{s}_i^T\mathbf{K}_{ij}\mathbf{B}_j\mathbf{K}_{jk}\mathbf{s}_k \tag{13}$$

and

$$\hat{\eta}_{ij}=tr[\mathbf{A}_i\mathbf{K}_{ij}\mathbf{B}_j\mathbf{K}_{ji}]. \tag{14}$$

Let $\mathbf{L}_{12}=\Lambda_{r_1}^T\mathbf{J}_1^T\mathbf{K}_{12}\mathbf{J}_2\Lambda_{r_2}$,

$$\mathbf{L}=\begin{bmatrix}0.5\,\Lambda_{r_{1,1}} & 0.5\,\mathbf{L}_{12}\\ 0.5\,\mathbf{L}_{12}^T & 0.5\,\Lambda_{r_2,2}\end{bmatrix} \tag{15}$$

$$\mathbf{P}=\begin{bmatrix}\sqrt{0.5}\,\mathbf{J}_1\mathbf{V}_{r_1,1} & 0\\ 0 & \sqrt{0.5}\,\mathbf{J}_2\mathbf{V}_{r_2,2}\end{bmatrix} \tag{16}$$

and $\check{\mathbf{B}}=\mathbf{P}\mathbf{L}^{-1}\mathbf{P}^T$. The Bhattacharrya distance approximation in RKHS is given by

$$\hat{J}_B(p_1,p_1)=\frac{1}{8}\{\hat{\xi}_{11}+\hat{\xi}_{22}-2\hat{\xi}_{12}\} \tag{17}$$

where

$$\hat{\xi}_{ij}=\mathbf{s}_i^T\mathbf{K}_{ij}\mathbf{s}_j-\mathbf{s}_i^T[\mathbf{K}_{i1}\mathbf{K}_{i2}]\check{\mathbf{B}}\begin{bmatrix}\mathbf{K}_{1j}\\ \mathbf{K}_{2j}\end{bmatrix}\mathbf{s}_j. \tag{18}$$

REFERENCES

[1] K. W. Berger, "Some factors in the recognition of timbre," *J. Acoust. Soc. Amer.*, no. 36, pp. 1888–1891, 1964.

[2] M. Clark, P. Robertson, and D. A. Luce, "A preliminary experiment on [he perceptual basis for musical instrument families," *J. Audio Eng. Soc.*, vol. 12, pp. 199–203, 1964.

[3] R. Plomp, "Timbre as a multidimensional attribute of complex tones," in *Frequency Analysis and Periodicity Detection in Hearing*, R. Plomp and G. Smoorenburg, Eds. Leiden, The Netherlands: Sijthoff, 1970, pp. 197–414.

[4] K. M. Grey, "Multidimensional perceptual scaling of musical timbres," *J. Acoust. Soc. Amer.*, vol. 61, pp. 1270–1277, 1977.

[5] R. A. Kendall, "The role of acoustic signal partitions in listener categorization of musical phrases," *Music Perception*, vol. 4, pp. 185–214, 1986.

[6] S. McAdams, S. Winsberg, S. Donnadieu, G. De Soete, and J. Krimphoff, "Perceptual scaling of synthesized musical timbres: common dimensions, specificities and latent subject classes," *Psychol. Res.*, vol. 58, pp. 177–192, 1995.

[7] *Information Technology—Multimedia Content Description Interface—Part 4: Audio*, Int. Std. ISO/1EC FDIS 15 938-4:200I(E), Jun. 2001.

[8] K. D. Martin, "Sound-Source Recognition: A Theory and Computational Model," Ph.D. dissertation, Mass. Inst. Technol., Jun. 1999.

[9] I. Kaminskyj, "Multi-feature musical instrument sound classifier," in *Proc. Australasian Computer Music Conf.*, Jul. 2000, pp. 53–62. Queensland University of Technology.

[10] I. Fujinaga and K. MacMillan, "Realtime recognition of orchestral instruments," in *Int. Computer Music Conf.*, 2000.

[11] B. Kostek and A. Czyzewski, "Automatic recognition of musical instrument sounds—further developments," in *Proc. 110th AES Convention*, Amsterdam, The Netherlands, May 2001.

[12] G. Agostini, M. Longari, and E. Pollastri, "Musical instrument timbres classification with spectral features," *EURASIP J. Appl. Signal Process.*, vol. 1, no. 11, pp. 5–14, 2003.

[13] A. Eronen, "Automatic Musical Instrument Recognition," M.S. thesis, Tampere Univ. Technol., Tampere, Finland, Apr. 2001.

[14] ——, "Musical instrument recognition using ICA-based transform of features and discriminatively trained HMMs," in *7th Int. Symp. Signal Processing and Its Applications*, Paris, France, Jul. 2003.

[15] G. Peeters, "Automatic classification of large musical instrument databases using hierarchical classifiers with inertia ratio maximization," in *Proc. 115th AES Convention*, New York, Oct. 2003.

[16] A. Krishna and T. Sreenivas, "Music instrument recognition: from isolated notes to solo phrases," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Montreal, QC, Canada, May 2004, pp. 265–268.

[17] S. Dubnov and X. Rodet, "Timbre recognition with combined stationary and temporal features," in *Proc. Int. Computer Music Conf.*, 1998.

[18] J. C. Brown, "Computer identification of musical instruments using pattern recognition with cepstral coefficients as features," *J. Acoust. Soc. Amer.*, vol. 105, pp. 1933–1941, Mar. 1999.

[19] J. C. Brown, O. Houix, and S. McAdams, "Feature dependence in the automatic identification of musical woodwind instruments," *J. Acoust. Soc. Amer.*, vol. 109, pp. 1064–1072, Mar. 2000.

[20] R. Ventura-Miravet, F. Murtagh, and J. Ming, "Pattern recognition of musical instruments using hidden markov models," in *Stockholm Music Acoustics Conf.*, Stockholm, Sweeden, Aug. 2003, pp. 667–670.

[21] A. Livshin and X. Rodet, "Musical instrument identification in continuous recordings," in *Proc. 7th Int. Conf. Digital Audio Effects (DAEX-4)*, Naples, Italy, Oct. 2004, pp. 222–227.

[22] ——, "Instrument recognition beyond separate notes—indexing continuous recordings," in *Proc. Int. Computer Music Conf.*, Miami, FL, Nov. 2004.

[23] S. Essid, G. Richard, and B. David, "Musical instrument recognition on solo performance," in *Eur. Signal Processing Conf. (EUSIPCO)*, Vienna, Austria, Sep. 2004, pp. 1289–1292.

[24] ——, "Efficient musical instrument recognition on solo performance music using basic features," in *Proc. AES 25th Int. Conf.*, London, UK, Jun. 2004, pp. 89–93.

[25] ——, "Musical instrument recognition based on class pairwise feature selection," in *Proc. 5th Int. Conf. Music Information Retrieval (ISMIR)*, Barcelona, Spain, Oct. 2004.

[26] K. Kashino and H. Mursae, "A sound source identification system for ensemble music based on template adaptation and music stream extraction," *Speech Commun.*, vol. 27, pp. 337–349, Sep. 1998.

[27] T. Kinoshita, S. Sakai, and H. Tanaka, "Musical sound source identification based on frequency component adaptation," in *Proc. UCAI Workshop on Computational Auditory Scene Analysis (UCAI-CASA)*, Stockholm, Sweden, Aug. 1999.

[28] B. Kostek, "Musical instrument recognition and duet analysis employing music information retrieval techniques," *Proc. IEEE*, vol. 92, no. 4, pp. 712–729, Apr. 2004.

[29] J. Eggink and G. J. Brown, "A missing feature approach to instrument identification in polyphonic music," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, China, Apr. 2003, pp. 553–556.

[30] ——, "Instrument recognition in accompanied sonatas and concertos," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Montreal, QC, Canada, May 2004, pp. 217–220.

[31] E. Vincent and X. Rodet, "Instrument identification in solo and ensemble music using independent subspace analysis," in *Proc. Int. Conf. Music Information Retrieval*, Barcelona, Spain, Oct. 2004.

[32] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 4, pp. 293–302, Jul. 2002.

[33] J.-J. Aucouturier and F. Pachet, "Representing musical genre: a state of the art," *J. New Music Res.*, vol. 32, 2003.

[34] A. Eronen and M. Slanely, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1997, pp. 1331–1334.

[35] J. C. Brown, "Musical instrument identification using autocorrelation coefficients," in *Int. Symp. Musical Acoustics*, 1998, pp. 291–295.
[36] P. Herrera, G. Peeters, and S. Dubnov, "Automatic classification of musical sounds," *J. New Music Res.*, vol. 32, no. 1, pp. 3–21, 2003.
[37] G. Peeters, "A Large Set of Audio Features for Sound Description (Similarity and Classification) in the Cuidado Project," IRCAM, Tech. Rep., 2004.
[38] L. R. Rabiner, *Fundamentals of Speech Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1993. Prentice Hall Signal Processing Series.
[39] O. Gillet and G. Richard, "Automatic transcription of drum loops," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Montreal, QC, Canada, May 2004, pp. iv-269–iv-272.
[40] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proc. IEEE*, vol. 88, no. 4, pp. 451–512, Apr. 2000.
[41] *MPEG-2 Advanced Audio Coding, AAC*, Int. Standard ISO/IEC 13818-7, Apr. 1997.
[42] M. Partridge and M. Jabri, "Robust principal component analysis," in *Proc. IEEE Signal Processing Soc. Workshop*, Dec. 2000, pp. 289–298.
[43] R. Kohavi and G. John, "Wrappers for feature subset selection," *Artificial Intell. J.*, vol. 97, no. 1–2, pp. 273–324, 1997.
[44] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artif. Intell. J.*, vol. 97, no. 1–2, pp. 245–271, Dec. 1997.
[45] I. Guyon and A. Elisseeff, "An introduction to feature and variable selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.
[46] S. Essid, G. Richard, and B. David, "Musical instrument recognition by pairwise classification strategies," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, Jul. 2006, to be published.
[47] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. New York: Academic, 1998.
[48] R. Duda and P. E. Hart, *Pattern Classification and Science Analysis*. New York: Wiley, 1973.
[49] S. Zhou and R. Chellappa, "From sample similarity to ensemble similarity: probabilistic distance measures in reproducing kernel hilbert space," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[50] B. Sholkopf and A. J. Smola, *Learning With Kernels*. Cambridge, MA: MIT Press, 2002.
[51] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[52] J. Marques and P. J. Moreno, "A Study of Musical Instrument Classification Using Gaussian Mixture Models and Support Vector Machines," Compaq Computer Corporation, Tech. Rep. CRL 99/4, 1999.
[53] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *J. Data Mining Knowl. Disc.*, vol. 2, no. 2, pp. 1–43, 1998.
[54] J. C. Platt, "Probabilistic outputs for support vector machines and comparisions to regularized likelihood methods," in *Advances in Large Margin Classifiers*. Cambridge, MA: MIT Press, 1999.
[55] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1998, vol. 10.
[56] M. Goto, H. Hashigushi, T. Nishimura, and R. Oka, "RWC music database: popular, classical, and jazz music databases," in *Int. Conf. Music Information Retrieval*, Paris, France, Oct. 2002.
[57] J. Dunagan and S. Vempala, "Optimal outlier removal in high-dimensional," in *Proc. 33rd Annu. ACM Symp. Theory of Computing*, Hersonissos, Greece, 2001, pp. 627–636.
[58] C. McKay and I. Fujinaga, "Automatic genre classification using large high-level musical feature sets," in *Proc. 5th Int. Conf. Music Information Retrieval*, Barcelona, Spain, Oct. 2004.
[59] T. Li and M. Ogihara, "Music genre classification with taxonomy," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Philadelphia, PA, Mar. 2005, pp. 197–200.
[60] P. H. Winston, *Artificial Intelligence*. Reading, MA: Addison-Wesley, 1984.

Slim Essid received the electrical engineering degree from the Ecole Nationale d'Ingénieurs de Tunis, Tunisia, in 2001 and the D.E.A (M.Sc.) degree in digital communication systems from the Ecole Nationale Supérieure des Télécommunications (ENST), the Université Pierre et Marie Curie (Paris VI), and the Ecole Supérieure de Physique et de Chimie Industrielle, Paris, France, in 2002. As part of his Master's thesis work, he was involved in a National Telecommunication Research Network (RNRT) project to propose a low bitrate parametric audio coding system for speech and music. He is currently pursuing the Ph.D. degree at the Department of Signal and Image Processing, ENST, Université Pierre et Marie Curie with a thesis on music information retrieval.

Gaël Richard (M'02) received the state engineering degree from the Ecole Nationale Supérieure des Télécommunications (ENST), Paris, France, in 1990, the Ph.D. degree from LIMSI-CNRS, University of Paris-XI, in 1994 in the area of speech synthesis, and the Habilitation à Diriger des Recherches degree from the University of Paris XI in September 2001.

After the completion of the Ph.D. degree, he spent two years at the CAIP Center, Rutgers University, Piscataway, NJ, in the speech processing group of Prof. J. Flanagan, where he explored innovative approaches for speech production. From 1997 and 2001, he successively worked for Matra Nortel Communications and for Philips Consumer Communications. In particular, he was the Project Manager of several large-scale European projects in the field of multimodal verification and speech processing. In 2001, he joined the Department of Signal and Image Processing, ENST, as an Associate Professor in the field of audio and multimedia signals processing. He is coauthor of over 50 papers and inventor in a number of patents, he is also one of the expert of the European commission in the field of man/machine interfaces.

Bertrand David was born in Paris, France, on March 12, 1967. He received the M.Sc. degree from the University of Paris-Sud, in 1991 and the Agrégation, a competitive french examination for the recruitment of teachers, in the field of applied physics, from the Ecole Normale Supérieure (ENS), Cachan, France, and the Ph.D. degree from the University of Paris VI in 1999 in the field of musical acoustics and signal processing.

From 1996 to 2001, he was a Lecturer in a graduate school in electrical engineering, computer science, and communications. He is now an Associate Professor with the Department of Signal and Image Processing, Ecole Nationale Supérieure des Télécommunications (ENST), Paris, France. His research interests include parametric methods for the analysis/synthesis of musical signals and parameter extraction for music description and musical acoustics.