AUTOMATIC INSTRUMENT RECOGNITION IN A POLYPHONIC MIXTURE USING SPARSE REPRESENTATIONS

Pierre Leveau^{1,2}
^{1} GET-ENST (Télécom Paris)
46, rue Dareau
75014 Paris

David Sodoyer^2, **Laurent Daudet**^{2}
^{2} Université Pierre et Marie Curie
Institut Jean Le Rond D'Alembert, LAM team
11, rue de Lourmel
75015 Paris

ABSTRACT

In this paper, a method to address the automatic instrument recognition in polyphonic music is introduced. It is based on the decomposition of the music signal with instrument-specific harmonic atoms, yielding to an approximate object representation of the signal. A post-processing is then applied to exhibit ensemble saliences that give clues about the number of instrument playing and the instrument labels. After a parameter optimization on a development set, the whole algorithm is applied on artificial mixes of solo performances. The identification of the number of instrument reaches 73 % and the identification of the ensemble label without prior knowledge on the number of instruments 17 %.

1 INTRODUCTION

Orchestration is a critical information for the automatic indexing of music. It gives an important clue about the music genres, and is often necessary for the query of sound samples for electronic music composing.

Automatic Instrument Recognition has raised some interest these latest years (see [1] for an overview of the domain). The early studies have addressed the recognition of isolated music notes [2], and solos phrases [3, 4, 5]. For these two contexts, machines now reach the performance of expert musicians. However, mono-instrument music takes a weak part of the overall listened music, that involves natural or artificial mixes of instruments.

To deal with multi-instrument music, several strategies have been adopted. Template-based approaches have first been proposed [6, 7]. Other approaches adapt "bag-of-frames" approaches to polyphony: some aim at identifying the main instrument playing in sonatas by isolating harmonic combs in the signal [8] or using the missing feature theory [9]. Other techniques consist in estimating jointly the instrument sources activated in a probabilistic framework [10], at a heavy computational cost. A few approaches involve blind source-separation as a preprocessing [11], but obviously relies on the performances of the source separation algorithm that are mainly at ease for statistically independent sources, exaggerated hypothesis for music. A recent work [12] presents a representation showing the instrument presence probabilities in the time-pitch plane without note detection. Ensemble classes can also be modeled using standard feature-based representations in addition with a hierarchical taxonomy [13], when the number of instrument combinations is tractable.

In this paper a recent development in the decomposition of music signals is studied for the recognition of music instrument in ensemble music. It relies on principles coming from the sparse approximations domain. To get a useful sparse representation of a signal, two aspects have to be investigated: the building of a signal model (*dictionary design*), and, given a dictionary, the choice of an algorithm and its optimization towards a faster or better approximation. Techniques from the sparse approximation domain have already been used for automatic music transcription in an unsupervised way [14, 15]: the building of the dictionary was done in an data-driven way, prohibiting to analyze the signals in view of prior knowledge of the sources. The introduction of prior knowledge about the sources in dictionaries has been presented in [16]: this knowledge is put in the amplitudes of the note partials. An algorithm decomposing the signal with such dictionaries is presented in Section 2. Section 3 shows how to learn partials amplitudes codebooks. In section 4, a post-processing is introduced to take a decision on the orchestration. The experiments of ensemble recognition are detailed in section 5.

2 DECOMPOSITION ALGORITHM

The signal model and decomposition algorithm have been introduced in [16]. Here the main features of the algorithm are highlighted.

2.1 Signal Model

The signal is decomposed as a linear combination of short pieces of signal h, called *harmonic atoms*:

$$x(t) = \sum_{n=1}^{N} \alpha_n \, h_{s_n,u_n,f_{0_n},A_n,\Phi_n}(t). \tag{1}$$

The set of all the atoms available to decompose the signal is called a *dictionary*.

The parameters of these atoms are the scale s_n, the time localization u_n, the fundamental frequency f_{0_n}, the partial amplitudes $A_n = \{a_{m,n}\}_{m=1:M}$ and thepartial phases $\Phi_n = \{\phi_{m,n}\}_{m=1:M}$. An atom h is itself defined as a linear combination of partials atoms:

$$h_{s,u,f_0,A,\Phi}(t) = \sum_{m=1}^{M} a_m \, e^{j\phi_m} g_{s,u,m.f_0}(t) \qquad (2)$$

where the amplitudes of the M partials are constrained to $\sum_{m=1}^{M} a_m^2 = 1$ and the signal g corresponding to each partial is given by a *Gabor* atom:

$$g_{s,u,f} = w\left(\frac{t-u}{s}\right) e^{2j\pi ft} \qquad (3)$$

with w a time- and frequency-localised window.

In our study, each A vector is linked to an instrument and a pitch (integer Midi Code), and are learned from databases of isolated instrument notes (see section 3).

2.2 Algorithm

Many algorithms exist for the decomposition of a signal on dictionaries of atoms. Among them, the Matching Pursuit algorithm has been chosen for this study. It is known to be relatively fast and to yield to decompositions close to optimal in practical cases. It has been introduced in [17], and proceeds as follows:

1. The correlations between the signal and all the atoms h of the dictionary are computed using inner products $\langle x, h\rangle = \sum_{t=1}^{T} x(t)\, h(t)$.
2. The atom h that has the largest absolute correlation $|\langle x, h\rangle|$ with the signal is selected, then subtracted from the signal with a weighting coefficient[1] $\alpha = \langle x, h\rangle$.
3. Correlations are updated on the residual signal, and the algorithm is iterated to step 2 until the stopping condition is satisfied. This condition can be a target Original-to-Residual energy ratio, or a fixed number of iterations.

With the parameters mentionned in Section 5, the runtime takes about 10 times real-time on a monoprocessor at 3 GHz.

3 LEARNING

3.1 Learning on annotated isolated notes

The vectors of partials amplitudes $\{A_{i,p,k}\}_{k=1...K}$ are learned for each instrument/pitch class $\mathcal{C}_{i,p}$ on isolated notes from three databases: the RWC Musical Instrument Sound Database [19], IRCAM Studio On Line [20] and the University of Iowa Musical Instrument Samples[21]. We select seven instruments producing harmonic notes: bassoon (Bo), oboe (Ob), clarinet (Cl), cello (Co), viola (Va), violin (Vl) and flute (Fl).

For each isolated note signal, the time frame with maximal energy is computed and all the subsequent time frames whose energy lies within a certain threshold of this maximum are selected. This relative threshold is set to a ratio of $1/20$ in the following. The partials amplitudes are computed on each of these training frames by

$$a_m = \frac{|\langle x, g_{s,u,m\times f_0}\rangle|}{\left(\sum_{m'=1}^{M} |\langle x, g_{s,u,m'\times f_0}\rangle|^2\right)^{1/2}} \qquad (4)$$

where f_0 is tuned in order to maximize the SRR on this frame. The vector of amplitudes is then associated to the pitch class p that is related[2] to f_0. The resulting number of vectors per instrument and per pitch class is approximately 300.

The size of the dictionary varies linearly as a function of the number of amplitude vectors. Since the number of vectors is too large to ensure computationally tractable decompositions, we choose to reduce the number of vectors by vector quantization: K amplitude vectors are kept for each class $\mathcal{C}_{i,p}$ using the k-means algorithm with the Euclidean distance. This operation also helps avoiding overfitting by averaging the training data and removing outliers.

3.2 Learning on solo phrases

A weak point of the use of the above-mentionned databases for learning is that the models are learned from isolated notes, that are recorded in almost anechoic conditions. To get a codebook more adapted to realistic recording conditions, A vectors can be learned on real solo phrases from commercial CDs in an adaptive way. Let's consider that we want to learn an adapted codebook for an instrument i, the following steps are performed:

- A codebook is built only with atoms from isolated notes of instrument i with the method described above
- Given this dictionary, The Matching Pursuit algorithm is performed with the following modifications:
  - **at the selection step**: an A vector is computed using Equation 4, by setting f_0 equal the fundamental frequency of the selected atom, then it is stored for further use.
  - **at the subtraction step**: instead of subtracting the selected atom from the signal, the signal is set to 0 on the time range corresponding

[1] Since we want to handle real signals, atoms are practically selected by couple of conjugate atoms, that form a real atom. More details on the computation of the amplitude and the phase of the weight can be found in [18].

[2] The pitch p related to f_0 is the integer the closest to $\log_2(f_0/440)+69$

to the selected atom by multiplying the signal by the function:

$$\omega(t) = 1 - \mathbf{1}_{[u,u+s]} \cos(2\pi(\frac{t-u}{s})) \qquad (5)$$

It prevents the algorithm to extract atoms on the residual of the extraction of a previous atom.

Once this process is achieved, a dictionary of atoms learned on solos phrases can be built. It can be noted that the solo database must be large enough to contain notes covering almost all the pitch range of each instrument. If atoms are missing for a given pitch p, the sub-dictionary from the previous pitch $p-1$ is taken.

4 SCORING

In the previous Section, an algorithm decomposing the music signal into a collection of meaningful objects has been presented. The output of such decomposition can be postprocessed in order to estimate the orchestration of the music signal.

4.1 Decomposition algorithm viewed as a pitch and instrument salience extractor

An atom extraction can be seen as an “pitch-and-instrument” salience extractor, since it correlates both a spectral enveloppe and a harmonic comb with the signal. Given an extracted atom at fundamental frequency f_0, scale s and localization u, we define the f_0-and-instrument salience for instrument i as [3] :

$$S_i = \max_{A \in \mathcal{C}_{i,p}} \{|\langle x, h_{s,u,f_0,A,\Phi_n}\rangle|\} \qquad (6)$$

If an instrument i enveloppe cannot play the pitch p, i.e. $\mathcal{C}_{i,p} = \emptyset$, its salience is set to 0. Although not required for the decomposition, all instrument saliences for every selected atom are kept for the scoring step: they are needed for the ensemble saliences evaluations.

4.2 From Instrument Salience to Ensemble Salience

The scoring algorithm processes the output of the decompositions to have an indication of which instruments are playing. Here, a frame-based scoring is developed: for a given time frame, the score of a given ensemble class depends on which atoms have been extracted and on their f_0-and-instrument salience.

Given a decomposition of a music signal, there can be several atoms per time frame since the music is in general polyphonic. The first step to perform is to select which atoms are present for each time frame, the timeline being sampled at the greatest common divider between the Δu corresponding to each scale. Then, the contribution of each atom a on a given time sample is equal to the value at instant u of the weighting window starting at u_a multiplied by the atom weight. Hence, given a time frame u and an ensemble label e, its ensemble salience is the following [4] :

$$\mathcal{S}_e(u) = \frac{\max_{C_e \in \mathcal{C}_e} \sum_{a \in C_e} \mathcal{S}_{i\,a}(u) w(\frac{u-u_a}{s_a})}{N_e^{\beta}} \qquad (7)$$

where $\mathcal{C}_e$ is the set of all the instrument salience combinations whose time support overlap with u. For example, if two atoms are present at time u, the salience of ensemble *Co&Fl* (Cello and Flute) is the maximum between the sum of the *Fl* salience for the first atom and the *Co* salience for the second one, and sum of the *Co* salience for the first atom and the *Fl* salience for the second one, divided by 2^{β}. An example of book output and corresponding ensemble saliences is displayed on Figure 1.

The β parameter is a sparsity parameter: it balances the weight between the sum of all atom saliences and the number of instrument of the ensemble. Its value has to be optimized on a development set. The use of such a coefficient to compensate the salience by the number of instrument is somewhat related to the Minimum Description Length criterion used in model order estimation, and also with cost functions for sparse approximations. It has also been used in an empirical way by Klapuri for multi-pitch estimation [22].

4.3 Voting

Decisions taken on single time frames does not provide useful information as such. However, one can be interested on decisions taken on the whole music signal, or a segment of it. To get a global decision from local ones, voting techniques must be employed. The technique used in this study is derived from a probabilistic framework. Other techniques, like majority-vote, have been tried but they yield to weaker results. First, the ensemble saliences are mapped to ensemble Pseudo Log-Likelihoods (PLL), then a segment PLL for each ensemble label is computed by adding the PLL of each time frames. The mapping of a ensemble salience $\mathcal{S}_e(u)$ to PLL $\mathcal{L}_e(u)$ is achieved with the following formula:

$$\mathcal{L}_e(u) = (\mathcal{S}_e(u))^{\gamma} \qquad (8)$$

γ weighs the influence of salience amplitudes over the overall score in the segment. Like β in previous Section, the γ coefficient has to be optimised on a development set. The decision over the all segment is obtained by summing all the PLL. It corresponds to an hypothesis of statistical independence between each time frame. This hypothesis is clearly erroneous in music signals (the orchestration does not change at every short time frame), but is commonly taken for fusion of local likelihood.

The whole system is described on figure 2.

[3] Note that the inner product is not depending on the values of Φ if f_0 is high enough since the partials atoms can be considered as orthogonal: $|\langle x, h_{s,u,f_0,A,\Phi_n}\rangle|^2 = \sum_{m=1}^{M} |\langle x, g_{s,u,m.f_0}\rangle|^2$

[4] Using the $L2$ norm $\sqrt{\sum_{a \in C_e} (\mathcal{S}_{i\,a}(u) w(\frac{u-u_a}{s_a}))^2}$ instead of the $L1$ norm $\sum_{a \in C_e} \mathcal{S}_{i\,a}(u) w(\frac{u-u_a}{s_a})$ would be more consistent with the optimality criterion of the decomposition, however it leads to weaker results in the studied applications

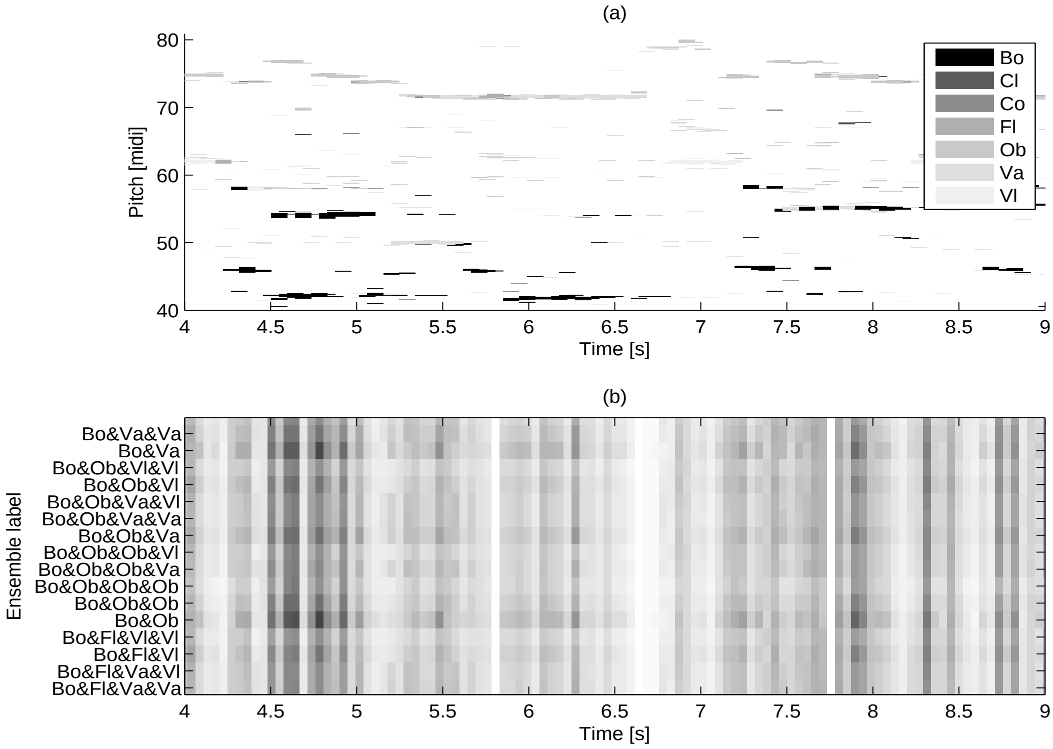

Figure 1. Bassoon (Bo) and Oboe (Ob) duo (synthetic mix): (a) Book representation in the Time-Pitch plane: atoms are represented by rectangles, whose width is the atom scale and height is their amplitudes, (b) Ensemble Saliences for a subset of ensemble labels (high saliences are darker).

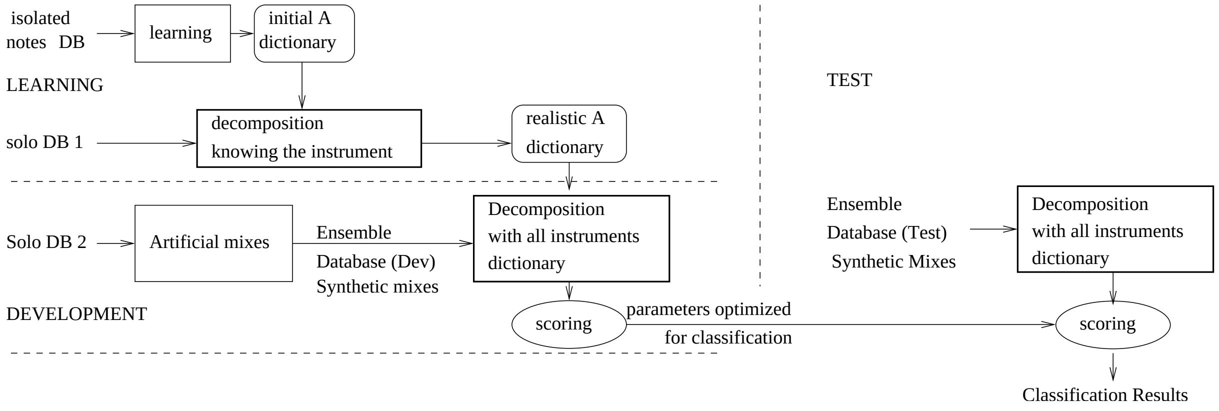

Figure 2. Flow chart of the whole system.

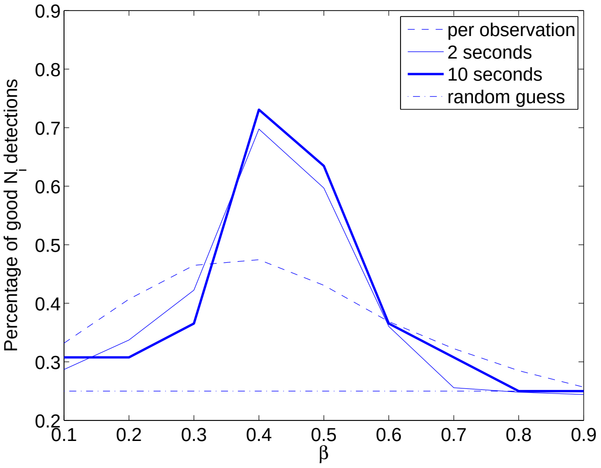

Figure 3. Accuracy of Number of Instrument Detection as a function of β for decision on single times frames, 2 seconds segments and 10 seconds segments.

5 EXPERIMENTS

5.1 Parameters

The parameters used for the decomposition are $s = 46ms$, $\Delta = 23ms$. f_0 is sampled logarithmically with a step of $1/10$ ton. The decompostions are performed until the Signal-To-Residual ratio reaches 20 dB.

5.2 Validation of the algorithm on solo phrases

The algorithm has been tested on solo phrases in [16] on an instrument set (Co, Cl, Fl, Ob, Vl), and the identification results are computed on 2 seconds segments. The overall instrument recognition rate was 68,5 %, with very few training data (3 atoms per pitch and instrument in average). With the dictionary of atoms learned on the set of isolated note, reduced to 16 atoms per pitch and instrument by vector quantization, the recognition rate raises to 75%. Additional improvement is brought by the use of the realistic set of solo phrases: the recognition rate becomes 84%. For this task, the recognition rate is now close to what a SVM-based system with a pairwise classification strategy can perform (84 % with [5]). These experiments now prove that decomposition with instrument-specific atoms are useful for instrument idenfication in realistic playing conditions.

5.3 Optimisation of parameters

The development and test sets are composed of artificial mixes of solo phrases extracted from commercial CDs, from sources different from the one used for atom learning. The mixes are done by summing the monoinstrument signals of instruments Bo, Co, Cl, Fl, Ob, Va and Vl after an energy normalization. For each set, 100 10-seconds samples have been made, 25 for each ensemble cardinal.

The parameters β and γ have to be tuned to maximize the accuracy of the estimation of the number of instruments, which is required to estimated the good instrument label. Optimizing these parameters for instrument label accuracy would overfit the algorithm for the solo recognition, that is the easiest problem. In our experiments on the development set, the best γ parameter has shown to be independant on the decision window: the value $\gamma = 0.8$ give the best results.

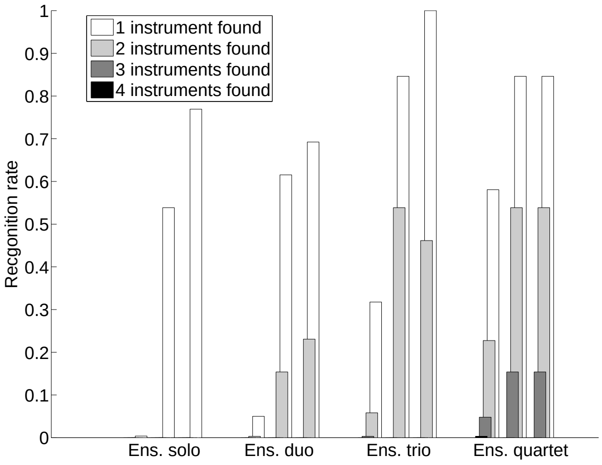

Figure 4. Ensemble recognition results for each subset (solos, duos, trios, quartets). For each ensemble, the three groups of bars depict respectively the results of a random draw, the results of our algorithm with no knowledge on the number of instruments playing, and the results knowing the number of instruments playing.

For these values, the instrument recognition rates for decisions on 10 seconds segments are depicted on Figure 4. It shows that the problem of finding an instrument among the mix is correctly addressed when the number of instrument is known (from 70 % to 100 %, depending on the ensemble type), and a less accurately when it is not known (from 54 % to 84 %). However, as the required number of instrument increases, the method fails at correctly identifying them alltogether. Dealing with ensembles of more than three instruments needs more refined techniques both at decomposition step and post-processing or more prior information, since the problem is more difficult (results for random draw is at less than 1 %).

6 CONCLUSION

In this paper, we developped a novel approach to address the complex problem of finding the instruments playing in ensemble music. The approach consists in getting a knowledge assisted mid-level representation of the signal, then in performing a post-processing using ensemble saliences based on individual instrument saliences derived from representation. The results are encouraging for the estimation of the number of instrument, but weak for the ensemble classification, which is a much more difficult problem without prior information on ensemble labels occurences.

Further work will be dedicated to the improvement of the decomposition step by refining atom parameters to better fit the underlying signal structures, and to group atoms into molecules at the extraction step to catch temporal dependencies. The joint estimation of atom combinations will also be investigated using more elaborated sparse decomposition algorithms. The post-processing will be improved by using melodic line following techniques to disambiguate mixes involving numerous notes.

7 REFERENCES

[1] P.Herrera-Boyer, A. Klapuri, and M. Davy. *Signal processing methods for music transcription*, chapter Automatic Classification of Pitched Musical Instrument Sounds. Springer, 2006.

[2] G. Peeters and X. Rodet. Hierarchical Gaussian Tree with Inertia Ratio Maximization for the Classification of Large Musical Instruments Databases. *Proc. of the 6th Int. Conf. on Digital Audio Effects, London*, 2003.

[3] K.D. Martin. *Sound-Source Recognition: A Theory and Computational Model*. PhD thesis, Massachusetts Institute of Technology, 1999.

[4] A. Eronen and A. Klapuri. Musical instrument recognition using cepstral cofficients and temporal features. 2000.

[5] S. Essid, G. Richard, and B. David. Musical instrument recognition by pairwise classification strategies. *IEEE Trans. on Speech, Audio and Language Processing*, July 2006.

[6] K. Kashino, K. Nakadai, T. Kinoshita, and H. Tanaka. Application of bayesian probability network to music scene analysis. In *Proc. IJCAI Worshop on Computational Auditory Scene Analysis*, 1995.

[7] T. Kinoshita, S. Sakai, and H. Tanaka. Musical sound source identification based on frequency component adaptation. In *Proc. IJCAI Worshop on Computational Auditory Scene Analysis*, pages 18–24, 1999.

[8] J. Eggink and G. J. Brown. Instrument recognition in accompanied sonatas and concertos. In *Proc. of IEEE Int. Conf. on Audio, Speech and Signal Processing (ICASSP)*, 2004.

[9] J. Eggink and G. J. Brown. Application of missing feature theory to the recognition of musical instruments in polyphonic audio. In *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, 2003.

[10] E. Vincent. Musical source separation using time-frequency source priors. *IEEE Trans. on Speech, Audio and Language Processing*, 14(1):91–98, January 2006.

[11] P. Jinachitra. Polyphonic instrument identification using independent subspace analysis. In *Proc. of Int. Conf. on Multimedia and Expo (ICME)*, 2004.

[12] T. Kitahara, M. Goto, K. Komatani, T. Ogata, and G. Okuno. Instrogram: A new musical instrument recognition technique without using onset detection nor f0 estimation. In *Proc. of IEEE Int. Conf. on Audio, Speech and Signal Processing (ICASSP)*, volume 5, pages 229–232, 2006.

[13] S. Essid, G. Richard, and B. David. Instrument recognition in polyphonic music based on automatic taxonomies. *IEEE Trans. on Speech, Audio and Language Processing*, 14(1):68–80, January 2006.

[14] P. Smaragdis and J.C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proc. of IEEE Int. Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 177–180, 2003.

[15] S.A. Abdallah and M.D. Plumbley. Polyphonic music transcription by non-negative sparse coding of power spectra. In *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, 2004.

[16] P. Leveau, E. Vincent, G. Richard, and L. Daudet. Mid-level sparse representations for timbre identification: design of an instrument-specific harmonic dictionary. In *1st Workshop on Learning the Semantics of Audio Signals*, Athens, Greece, dec 2006.

[17] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Trans. on Signal Processing*, 41(12):3397–3415, December 1993.

[18] R. Gribonval. *Approximations non-linéaires pour l'analyse des signaux sonores (in French)*. PhD thesis, UNIVERSITÉ DE PARIS IX DAUPHINE, 1999.

[19] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC Musical Instrument Sound Database. Distributed online at http://staff.aist.go.jp/m.goto/RWC-MDB/.

[20] Iowa database, http://theremin.music.uiowa.edu/mis.html.

[21] Studio online database, http://forumnet.ircam.fr/402.html?l=1.

[22] A. P. Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, 2006.