Realtime Multiple Pitch Observation using Sparse Non-negative Constraints

Arshia Cont
Ircam, Realtime Applications Team, Paris. and
Center for Research in Computing and the Arts, UCSD, San Diego.
cont@ircam.fr

Abstract

In this paper we introduce a new approach for realtime multiple pitch observation of musical instruments. The proposed algorithm is quite different from others in the literature both in its purpose and approach. It is destined not for continuous multiple f_0 recognition but rather for projection of the ongoing spectrum to learned pitch templates. The decomposition algorithm on the other hand, does not compromise signal processing models for pitches and consists of an algorithm for efficient decomposition of a spectrum using known pitch structures and based on sparse non-negative constraints. After introducing the algorithm along with evaluations, a real-time implementation of the algorithm is provided for free download for the MaxMSP realtime programming environment.

Keywords: Multiple-pitch observation, Non-negative Matrix Factorization, Sparseness constraints, Machine Learning.

1. Introduction

The task of estimating multiple fundamental frequencies of audio and speech signals has attained substantial effort from the research community in the recent years. More interestingly, proposed algorithms in the literature undergo a wide variety of methods spanning from pure signal processing models to machine learning methods. For an excellent overview of different methods for multiple-f_0 estimation, we refer the curious reader to [1].

In this paper, we present an algorithm for realtime observation of multiple pitches of polyphonic instruments. The proposed method uses machine learning techniques in its core for realtime decomposition of ongoing audio using known pitch templates of an instrument. It is thus different from many algorithms in the sense that it does not provide the user with f_0 computations but tells which of many (previously learned) pitch templates are currently active for reconstruction of the ongoing audio. During (one-time) learning, the system browses a library of instrumental sounds and learns spectral structures of the pitches that can be produced by the instrument. In this sense, one can say, the algorithm learns the instrument pitch model that will be used during realtime observation.

The algorithm presented here is based on a modified *Non-negative Matrix Factorization* (NMF) algorithm introduced originally by Lee and Seung [2]. The first musical applications of NMF are reported in [3, 4] for polyphonic music transcription and [5] for source separation. In all approaches, the algorithm learns parts representations of the audio signal which correspond to music events. Despite their significant results, the algorithms are heavy in computation and non-realtime in nature. Recently, Sha and Saul have proposed a real-time pitch determination algorithm using NMF for speech signals [6]. Our approach is quite similar in architecture to Sha and Saul but different by adding sparseness constraints to a regular NMF (and thus changing the algorithm used). Despite their success on speech databases, their algorithm would be far from success for music signals which undergo wider spectral characteristics than speech signals.

The paper is organized as follows. In Section 2 we present the general architecture for training and realtime observations. This general architecture will be detailed in the following sections on learning and multiple-pitch observation with the proposed NMF. Specifically in Section 4 we detail our *sparse NMF* algorithm which is used during realtime observation followed by results and discussions of the algorithm.

2. General Architecture

The proposed method relies on unsupervised learning algorithms that are used for knowledge representation and discovery. During realtime observation, the algorithm tries to reconstruct the ongoing audio using previously learned pitch structures of an instrument, as a linear combination with non-negative weights. This implies an offline learning of pitch structures of all the pitches of an instrument which will be used as templates during learning. We will give details of this learning phase in Section 3.

This architecture is similar to the system proposed in [6] with a crucial difference for music signals. Instead of using a regular NMF algorithm for real-time determination of pitch, we use a modified NMF algorithm with sparseness constraints as outlined casually below and detailed in Section 4. We compare results of the proposed algorithm with that of [6] in section 5.

2.1. Non-negative Matrix Factorization

The learning algorithm used in both learning and realtime observation is based on *Non-negative Matrix Factorization* (NMF). Both algorithms will be presented mathematically in Sections 3 and 4 but like any machine learning algorithm, this choice will bring advantages and limitations which are part of the general philosophy of the proposed method. Non-negative matrix factorization is an unsupervised algorithm for decomposition and learning for multivariate data [2]. Speaking generally, unsupervised learning algorithms such as principal component analysis and vector quantization can be understood as factorizing a data matrix subject to different constraints. NMF in this respect is another factorization algorithm that uses nonnegativity constraint. Nonnegativity in this sense, means that an original matrix V is composed of the desired number of templates (stored as columns in W) which can reconstruct the original by being added linearly with nonnegative weights (stored in H) or $V \approx WH$. The idea behind NMF algorithms is that the signal can be reconstructed using its *parts* (W) through *addition* of the parts with different weights, and thus the parts are not necessarily independent. This fact seems to be consistent with how we hear and transcribe music chords through addition of single pitches or identities we know a priori. However, this implies careful considerations for signal representation and observation as discussed below.

In our formulation of the problem of multiple pitch observation, V would be the ongoing audio representation or the result of the signal processing frontend of the system, W would represent the pitch templates of an instrument, and H would represent nonnegative weights corresponding to presence of pitch templates in V.

2.2. Signal Processing Frontend

The *additive* characteristic of NMF is an essential factor for any kind of representation used for V which, in the case of multiple pitch observation, implies that the spectral representation used for V should demonstrate a harmonic stack of pitch templates added together for a given chord.

The signal processing front end used for this observation is the result of a fixed point analysis of frequency to instantaneous frequency mapping of the ongoing audio spectrum [7]. The short-time Fourier transform (STFT) is an efficient tool for instantaneous frequency (IF) estimation [8]. Given $z(\omega, t)$ as the analytical form of the STFT, the mapping

$$\lambda(\omega, t) = \frac{\partial}{\partial t} arg[z(\omega, t)] \tag{1}$$

can be computed efficiently and in real-time using STFTs [8] and the fixed points of this mapping can be extracted using the following criteria [7, 6] :

$$\lambda(\omega^*, t) = \omega^* \quad and \quad \frac{\partial \lambda}{\partial \omega}|_{\omega=\omega^*} < 1 \tag{2}$$

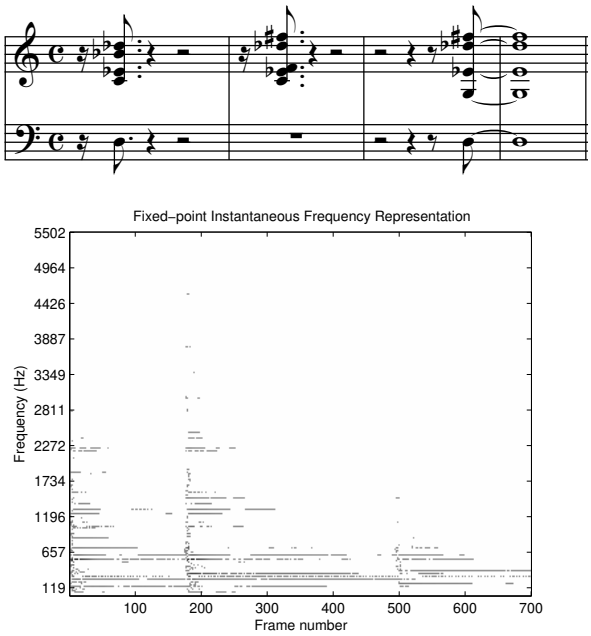


FIG. 1. Fixed Point Instantaneous Frequency Representation of audio (bottom) corresponding to three chords (above) played on a piano

As a result, vector V would be non-negative amplitudes of the fixed-point instantaneous frequency representing harmonic stacks at each analysis frame with the rest of the spectrum zeroed out. Figure 1 shows a snapshot of this representation on three given chords played on a real piano.

2.3. Sparsity of the solution

Despite perceptual advantages of an NMF approach over ICA algorithms for multiple-pitch detection, since pitch templates are not mathematically independent, for a given spectrum (in V) there may exist many possible solutions (H) using templates in W. More specifically for our problem, a given piano chord can be reconstructed by the templates of its original pitches as well as octaves, dominant and other pitches with harmonic relations to the original ones.

To overcome this problem, we use the strong assumption that the correct solution for a given spectrum (in V) uses a minimum of templates in W, or in other words, the solution has the minimum number of non-zero elements in H. This assumption is hard to be proofed for every music instrument and highly depends on the template presentations in W, but is easily imaginable as harmonic structure of a music note can be minimally expressed (in the mean squared sense) using the original note than a combination of its octaves and dominant.

Fortunately, this assumption has been heavily studied in the field of *sparse coding*. The concept of 'sparse coding' refers to a representational scheme where only a few units out of a large population are effectively used to represent typical

data vectors [9]. One of the useful properties of NMF is that it usually produces a *sparse* representation of the data. However this sparseness is more of a side-effect than a goal and one can not control the degree to which the representation is sparse.

3. Learning Pitch Templates

As explained earlier, the system knows the pitch structures of all pitches of an instrument for use during realtime observation. In Section 4 we introduce the decomposition process or how to obtain H and here we show how we learn different pitch templates for an instrument. As a reminder, W contains pitch structures of all pitches of a given instrument. For example, for an acoustic piano, matrix W would contain all 88 pitches as 88 different columns. To this end, training is done using databases of instrumental sounds [10, 11] and an off-line training learns different pitch structures of an instrument by browsing all sounds produced by the given instrument in the database and stores them in matrix W for future use.

For each audio file in the database, training is an iterative NMF algorithm with a symmetric kullback-leibler divergence for reconstruction error as shown in Equation 3, where $\otimes$ is an element by element multiplication. In this off-line training V would be the short-time fixed-point instantaneous frequency spectrum of the whole audio file as described in Section 2.2 and the learning algorithm factorizes V as $V \approx WH$. In order to obtain precise and discriminative templates, we put some constraints on W vectors learned during each NMF iteration. For each sound in the database (or each pitch) we force the algorithm to decompose V into two vectors (W has two columns) where we only learn one vector and have the other fixed as white non-negative noise, where only the first one would be stored for the global W. This criteria helps the algorithm focus more on the harmonic structure of V. Furthermore, we constrain each iteration by an envelope (Env in equation 3). This envelope is constructed from the pitch information of the audio file (usually taken from the name of the file in the database) and emphasizes frequencies around the fundamental with a decreasing envelope towards the end and close to zero for frequencies less than the fundamental. While this assumption does not hold for many instruments (such as violin and piano), it enforces the most important characteristic of the spectrum for pitch classification. This constraint improves common octave and harmonic errors that can be introduced in pitch determination during realtime observation.

$$
\begin{aligned}
H_{a\mu} &\longleftarrow Env \otimes H_{a\mu} \frac{\sum_i W_{ia} V_{i\mu} / (WH)_{i\mu}}{\sum_k W_{ka}} \\
W_{ia} &\longleftarrow Env \otimes W_{ia} \frac{\sum_i H_{a\mu} V_{i\mu} / (WH)_{i\mu}}{\sum_\nu H_{a\nu}}
\end{aligned}
\tag{3}
$$

When the training reaches an acceptable stopping criteria, the harmonic spectra in the local W will be saved in the global W and the algorithm continues to the next audio file in the database until it constructs W for all pitches of the instrument.

4. NMF for Multiple-pitch Observation

As stated in Section 2.3, in order to decompose the spectrum using learned pitch templates, the solution needs to be sparse. In this section, we introduce a modified sparse non-negative decomposition algorithm useful for realtime pitch observation.

Numerous sparseness measures have been proposed and used in the literature. In general, these measures are mappings from $\mathbb{R}^n$ to $\mathbb{R}$ which quantify how much energy of a vector is packed into a few components. As argued in [12], the choice of sparseness measure is not a minor detail but may have far reaching implications on the structure of a solution. Very recently, Hoyer has proposed an NMF with sparseness constraints by projecting results into ℓ_1 and ℓ_2 norm-spaces [13]. Due to real-time considerations and the nature of sparseness in audio signals for pitch determination we propose a modified version of NMF with sparseness constraint of that in [13].

The definition commonly given for sparseness is based on the ℓ_0 norm defined as the number of non-zero elements

$$\|X\|_0 = \frac{\#\{j, x_j \neq 0\}}{N}$$

where N is the dimension of vector X. It is characteristic for the ℓ_0 norm that the magnitude of non-zero elements is ignored. Moreover, this measure is only good for noiseless cases and adding a very small measurement noise makes completely sparse data completely non-sparse. A common way to take the noise into account is to use the ℓ_ϵ norm defined as follows :

$$\|X\|_{0,\epsilon} = \frac{\#\{j, |x_j| \geq \epsilon\}}{N}$$

where parameter ϵ depends on the noise variance. In practice, there is no known way of determining this noise variance which is independent of the variance in x. Another problem of this norm is that it is non-differentiable and thus can not be optimized with gradient methods. A solution is to approximate the ℓ_ϵ norm by $tanh$ function,

$$g(x) = tanh(|ax|^b)$$

where a and b are positive constants. In order to imitate ℓ_ϵ norm, the value of b must be greater than 1.

In addition to the $tanh$ norm, we force an ℓ_2 constraint on the signal. This second constraint is crucial for the normalization of the results and emphasis on significance of factorization during note events in contrary to silent states.

In summary, the sparseness measure proposed is based on the relationship between the ℓ_ϵ norm and the ℓ_2 norm as demonstrated in Equation 4.

$$\text{sparseness(x)} = \frac{\sqrt{N} - \sum tanh(|x_i|^2)/\sqrt{\sum x_i^2}}{\sqrt{N} - 1} \quad (4)$$

For NMF with sparseness constraint, we use gradient descent updates instead of the original NMF multiplicative updates (Equation 3) and project each vector in real-time to be non-negative and have desired ℓ_2 and ℓ_ϵ norms. This projected gradient descent, adapted from [13], is outlined below. Once again this algorithm shows the factorization for H when templates are known.

Given V and W, find the non-negative vector H with a given ℓ_ϵ norm and ℓ_2 norm :

1. Initialize H to random positive matrices
2. Iterate
   (a) Set $H = H - \mu_H W^T(WH - V)$
   (b) Set $s_i = h_i + (\ell_\epsilon - \sum tanh(h_i^2))/N$ and $m_i = \ell_\epsilon/N$
   (c) Set $s = m + \alpha(s - m)$ where $\alpha = \frac{-(s-m)^T m + \sqrt{((s-m)^T m)^2 - \sum(s-m)^2(\sum m^2 - \ell_2^2)}}{\sum(s-m)^2}$
   (d) Set negative components of s to zero and set $H = s$

Algorithm 1. Sparse Non-Negative Matrix Decomposition

Where (a) is a negative gradient descent and (b) through (d) are the projection process on the ℓ_ϵ and ℓ_2 space. In (b) we are projecting the vector to the ℓ_ϵ hyperplane and (c) solves a quadratic equation ensuring that the projection has the desired ℓ_2 norm.

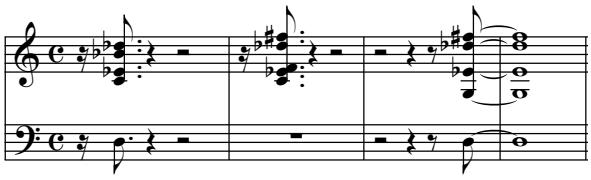
For realtime pitch observation, the ℓ_2 norm is provided by the spectrum energy of the realtime signal and the ℓ_ϵ takes values between 0 and 1, is user-specified and can be controlled dynamically. The higher the ℓ_ϵ, the more sparse is the solution in H. V would be a vector of size n where here we use $n = 512$ for an FFT window of $93msec$ to capture harmonic structure up to about $6KHz$. Equivalently, W would be a matrix of $n \times m$ with m as the number of templates and H would be a vector of size m.

5. Results and Evaluation

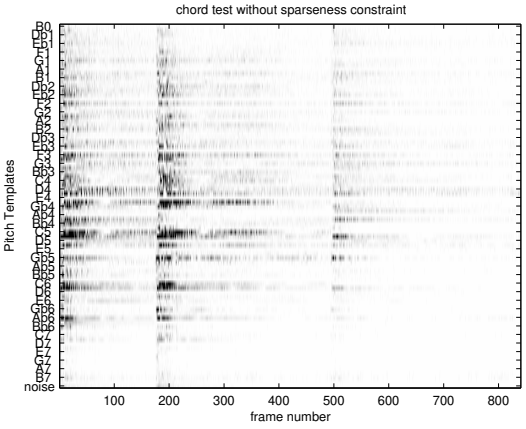
In this section we evaluate sparsity of the solution and pitch observation of the proposed algorithm.

5.1. Sparsity

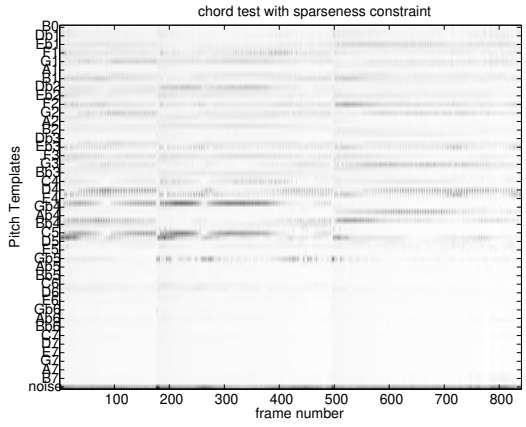
We start demonstrating the results by emphasizing on the sparsity factor of the proposed algorithm. Figure 2 compares two instances of NMF pitch observation on three piano chords represented in Figure 2(a). Results on Figure 2(b) purport to a regular NMF algorithm (learning only H with known W or pitch templates) as proposed in [6] using regular NMF for speech signals and Figure 2(c) corresponds to results obtained using the algorithm presented in the previous section. Both figures show time (or analysis frame number) on the *x-axis* and the contributions of each pitch template (indexed on *y-axis* by their names) are shown in the figure by a colormap representing an interval between 0 and 1. The analysis is done by using a window size of $93msec$, overlap of $12msec$ and fixed ℓ_ϵ of 0.8.



(a) Piano Score being played



(b) Decomposition using regular NMF



(c) Decomposition using sparse NMF

FIG. 2. Comparing sparse and non-sparse non-negative decomposition for pitch observation

An important remark is the presence of an additional *noise* template in W (bottom rows of Figure 2(b) and 2(c)). This template is responsible for absorbing eventual noise, reverberation and non-harmonic structures such as transients that

can not be decomposed using pitch templates of the instrument. As is seen in Figure 2(c), this noise template has important presence (specially during transitions) in the sparse solution and assures *generalization* and robustness of the algorithm.

Comparing Figures 2(b) and 2(c), the sparsity of the second is quite evident. The sparse NMF learns a solution that emphasizes *few* templates where the regular NMF (Figure 2(b)) uses a large number of templates (typically templates of harmonic relationship to the original pitch depicted in the score of Figure 2(a)) for the solution.

Finally, it is worthy to note that the example shown in Figure 2 is the result of playing the score in Figure 2(a) on a (real) Piano different than the one used for learning the pitch templates. Moreover, the sustain pedal of the piano has been pressed down during the whole performance of the score, adding sustained resonance throughout the whole spectrum.

5.2. Evaluating the observation

A close examination of results in Figure 2(c) along with the score in Figure 2(a) reveals that the corresponding templates of the notes in the score are along the most active ones at the appropriate time. However, we prefer evaluating the algorithm and its robustness on a larger corpus of music. For this purpose, we run the algorithm on three classical music pieces which are aligned to their MIDI scores using an external application described in [14]. The audio is taken from the RWC database [15] and the pieces and their specifications are given in Table 1. Pieces 1 and 2 were performed on a Piano and piece 3 is performed on a Harpsichord. Accordingly, pitch templates of appropriate instruments will be used during evaluation. Note that by doing this, we are evaluating the system in a *transcription* framework, eventhough the proposed system does no undertake transcription because of lack of temporal smoothing. In previous state-of-the-art evaluation, authors in [16] construct the reference by aligning the MIDI score to their results using dynamic programming. In our evaluation as mentioned, we use an externally aligned score to the audio.

TAB. 1. Specification of Audio and Midi used for evaluation

#	Piece Name	Duration	Events
1	Mozart's *Piano Sonata in A major,K.331*	9 :55	4268
2	Chopin's *Nocturne no.2 in Eb major, opus 9*	3 :57	1291
3	Bach's *Fugue no.2 in C minor BWV 847*	1 :53	752

For this evaluation, we compare results of the proposed algorithm on the audio with the aligned MIDI score. Specifically, for each note event in the aligned score, we look at the corresponding frames of the sparse NMF observation and check if the corresponding template has high activity and if it is among the top N templates, where N specifies the number of pitches active at the event frame time taken out of the reference MIDI. This way, for each event in the score we can have a precision percentage and the overall mean of these can represent the algorithm's precision (*precision* 1). Moreover, since we do not have any specific temporal model (for note-offs for example) we can consider (subjectively) positive detection during atleast 80% of a note life to be *acceptable* and recalculate the precision (*precision* 2). The results of this evaluation are shown in Table 2 for both the *sparse* algorithm proposed in this paper and the regular (*non-sparse*) NMF proposed in [6].

TAB. 2. Multiple-pitch observation evaluation results

Piece No.	Precision 1		Precision 2	
	Sparse	**Non-sparse**	**Sparse**	**Non-Sparse**
1	78.1%	49.6%	88.0%	68.1%
2	71.0%	32.7%	81.2%	51.2%
3	74.9%	43.1%	87.1%	59.4%

Besides the evident gain of the sparse over non-sparse algorithm, there is a downward shift in the results for the second piece in Table 2. This is mostly due to the fact that for the performance of (Chopin's *Nocturne*), pianists always use the sustain pedal of the piano excessively, thus adding more and more sustained resonance of previous pitches in the spectrum. One reason for providing *precision 2* is that in a multiple-pitch situation (and specially in the presence of the sustain pedal for piano) the duration of each note event becomes intractable or inexact. It should be mentioned that the references used are also erroneous especially with the timing of events and further evaluations need hand-correction of these references to the audio.

Finally, note that the templates were trained on a different piano than the one used for evaluation. The sounds used are professional recordings and ofcourse, in a realtime situation, the recording microphone would be placed in a closer location to the piano. However, the obtained results reveal the somehow surprisingly robust and generalized result of learning.

6. Realtime Implementation

The proposed algorithm has been developed and tested for MaxMSP [1] realtime programming environment and using the FTM library [2] and is available for free download at :

`http://crca.ucsd.edu/arshia/ismir06/`

Figure 3 shows a screenshot of this implementation.

7. Conclusion

In this paper, we proposed a realtime multiple pitch observation algorithm based on sparse non-negative constraints. The algorithm is different from most algorithms in the sense that it knows the pitch templates of the instrument in advance and through an unsupervised learning process as described. Thanks to sparseness constraints it correctly observes

[1] `http://www.cycling74.com/`

[2] `http://www.ircam.fr/ftm/`

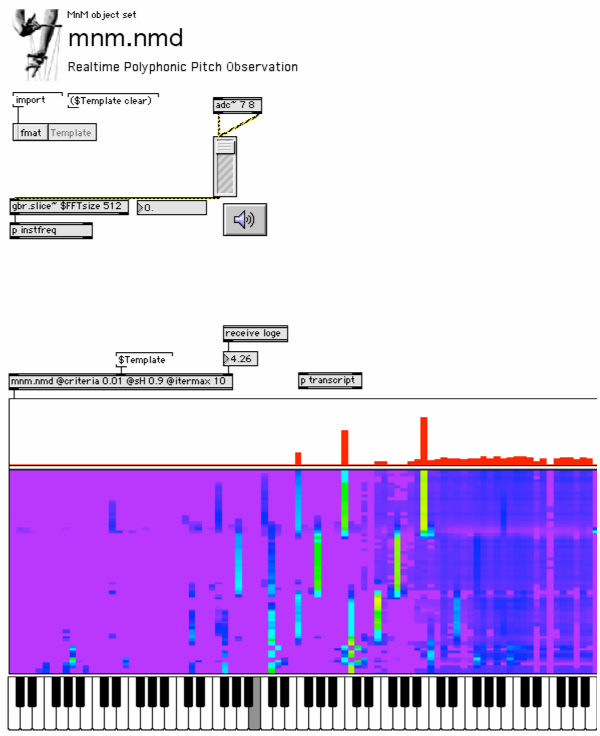


FIG. 3. Sparse Non-negative Multiple-Pitch Observation in action, in MaxMSP environment and using FTM libraries

ongoing pitches. We evaluated the algorithm using (externally) aligned audio to score as reference. The evaluation results in Table 2 are close enough to state-of-the-art multiple pitch algorithms with the huge difference that the proposed algorithm in this paper is destined to work in realtime. We are currently evaluating the algorithm on larger databases and more complicated musical situations.

Perhaps a major drawback of the described model described is the stationary pitch template model rather than a moving spectrum with instrument envelopes. One should note that by this outcome, we gain better generalization and realtime capabilities. Finally, the proposed algorithm can be used along in various applications as a multiple-pitch observation module. One instance of such application is reported in [17] where the proposed algorithm is used in the context of realtime polyphonic score following.

8. Acknowledgments

The realtime implementation of the proposed algorithm would never see the light of day without great help and effort of the following individuals : Rémy Muller for his help with MaxMSP and FTM externals, Olivier Pasquet and Norbert Schnell for their help and availability for MaxMSP patching. Also, great thanks to Chungsin Yeh for providing the aligned database and useful discussions during evaluations.

References

[1] A. de Cheveigné, "Multiple f0 estimation," in *Computational Auditory Scene Analysis : Principles, Algorithms and Applications*, D.-L. Wang and G.J. Brown, Eds. IEEE Press / Wiley, 2006 (in press).

[2] Daniel D. Lee and H. Sebastian Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems 13*, Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, Eds. 2001, pp. 556–562, MIT Press.

[3] P. Smaragdis and J. Brown, "Non-negative matrix factorization for polyphonic music transcription," 2003.

[4] Samer M. Abdallah and Mark D. Plumbley, "Polyphonic transcription by non-negative sparse coding of power spectra.," in *ISMIR*, 2004.

[5] Tuomas Virtanen, "Separation of sound sources by convolutive sparse coding," 2004.

[6] Fei Sha and Lawrence Saul, "Real-time pitch determination of one or more voices by nonnegative matrix factorization," in *Advances in Neural Information Processing Systems 17*, Lawrence K. Saul, Yair Weiss, and Léon Bottou, Eds. MIT Press, Cambridge, MA, 2005.

[7] Hideki Kawahara, H. Katayose, Alain de Cheveigné, and R.D. Patterson, "Fixed point analysis of frequency to instantaneous frequency mapping for accurate estimation of f0 and periodicity," in *Eurospeech*, 1999, vol. 6, pp. 2781–2784.

[8] Toshihiko Abe, Takao Kobayashi, and Satoshi Imai, "Harmonic tracking and pitch extraction based on instantaneous frequency," in *IEEE ICASSP*. 1995, pp. 756–759, Tokyo.

[9] D. J. Field, *Neural Computation*, vol. 6, chapter What is the goal of sensory coding ?, pp. 559–601, 1994.

[10] Lawrence Fritts, "Musical instrument samples from the university of iowa electronic music studios," Webpage : `http://theremin.music.uiowa.edu/`, 1997.

[11] Guillaume Ballet, Riccardo Borghesi, Peter Hoffmann, and Fabien Lévy, "Studio online 3.0 : An internet "killer application" for remote access to ircam sounds and processing tools," in *Journée d'Informatique Musicale (JIM)*, paris, 1999.

[12] Juha Karvanen and Andrzej Cichocki, "Measuring sparseness of noisy signals," in *ICA2003*, 2003.

[13] Patrik O. Hoyer, "Non-negative matrix factorization with sparseness constraints.," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[14] Hagen Kaprykowsky and Xavier Rodet, "Globally optimal short-time dynamic time warping application to score to audio alignment," in *IEEE ICASSP*. May 2006, Toulouse.

[15] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka, "Rwc music database : Popular, classical and jazz music databases.," in *ISMIR*, 2002.

[16] Hirokazu Kameoka, Takuya Nishimoto, and Shigeki Sagayama, "Separation of harmonic structures based on tied gaussian mixture model and information criterion for concurrent sounds," in *IEEE ICASSP*. March 2005, Philadelphia.

[17] Arshia Cont, "Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical hmms," in *IEEE ICASSP*. May 2006, Toulouse.