Proc. of the 10th Int. Conference on Digital Audio Effects (DAFx-07), Bordeaux, France, September 10-15, 2007

REALTIME MULTIPLE-PITCH AND MULTIPLE-INSTRUMENT RECOGNITION FOR MUSIC SIGNALS USING SPARSE NON-NEGATIVE CONSTRAINTS

Arshia Cont

Ircam-CNRS UMR 9912, IMTR Team,
Paris, France. and
Music Department, UCSD, La Jolla, CA.
cont@ircam.fr

Shlomo Dubnov

Music Department,
University of California in San Diego
La Jolla, CA.
sdubnov@ucsd.edu

David Wessel

Center for New Music and Audio Technologies,
University of California in Berkeley
Berkeley, CA.
wessel@cnmat.berkeley.edu

ABSTRACT

In this paper we introduce a simple and fast method for realtime recognition of multiple pitches produced by multiple musical instruments. Our proposed method is based on two important facts: (1) that timbral information of any instrument is pitch-dependant and (2) that the modulation spectrum of the same pitch seems to result into a persistent representation of the characteristics of the instrumental family. Using these basic facts, we construct a learning algorithm to obtain pitch templates of all possible notes on various instruments and then devise an online algorithm to decompose a realtime audio buffer using the learned templates. The learning and decomposition proposed here are inspired by non-negative matrix factorization methods but differ by introduction of an explicit sparsity control. Our test results show promising recognition rates for a realtime system on real music recordings. We discuss further improvements that can be made over the proposed system.

1. INTRODUCTION

We address two important problems often discussed in the music information retrieval and computer music research communities: estimating multiple fundamental frequencies of music signals and musical instrument recognition. Both topics have received substantial effort from the research community especially in the recent years for polyphonic sounds (as opposed to solo or monophonic audio). Both are also important tasks for many applications including automatic music transcription, music information retrieval and computational auditory scene analysis. Another motivation for this work is the continuing need for live algorithms in computer music where the recognition of musical characteristics of the signal such as pitches and instruments becomes essential.

The multiple-pitch detection literature contains a wide variety of methods spanning from pure signal processing models to machine learning methods for both music and speech signals. For an excellent overview of different methods for multiple-$f0$ estimation, we refer the reader to [1]. The main aim of instrument identification is to determine the number and the names of the instruments present in a given musical excerpt. Whereas musical instrument recognition studies mainly deals with solo musical sounds, the number of those dealing with polyphonic music has been increasing in the recent years. In [2], Kashino et al. develop a template-matching method that compares the observed waveform locally with sum of template waveforms that are phase aligned, scaled, and filtered adaptively. Similarly [2, 3] use feature matching methods where features computed in zones where several notes overlap are modified or discarded before stream validation depending on their type. Other systems directly address the instrument identification without considering note models or pitch detection. In [4], Essid et al. introduce an SVM model with a hierarchical taxonomy of a musical ensemble that can classify possible combinations of instruments played simultaneously. In another approach, Livshin and Rodet [5] use an extensive set of feature descriptors on a large set of pitched instrument sound samples, reducing the feature dimensions with Linear Discriminant Analysis and then classifying the sounds with a KNN method. More recently Kitahara et al. [6] have proposed a method for visualizing the instrument existence probabilities in different frequency regions.

In this paper, we propose a new technique that recognizes multiple pitches along with their instrument origin in polyphonic musical audio signals and in realtime; hence, addressing both problems mentioned earlier. The main difference between our proposed method and the ones discussed above is the fact that our system is geared towards real-time recognition and for realistic musical situations. Our approach is similar to [2] where instrument-based pitch templates are being matched to the ongoing audio but differs significantly by the extensive reliance on sparse machine learning in our approach. Our proposal is inspired by simple observational facts regarding the nature of musical instruments and consists of decomposing an ongoing audio signal using previously learned instrument-dependant pitch templates and sparse non-negative constraints. We discuss the basic idea and general architecture of the algorithm in section 2. The algorithm both in learning and realtime decomposition phases, uses a recently introduced signal representation scheme based on modulation spectrum [7]. The key fact here is that the modulation spectrum of musical instruments seems to be an important discriminating factor among them. We will discuss the modulation spectrum and its pertinence to our problem in

section 3 as the main signal processing front-end for our algorithm. In section 4 we show how instrument templates are learned. This learning is once-for-all and is based on *Non-negative Matrix Factorization* (NMF) algorithm [8]. These learned templates would be imported to the main algorithm for realtime instrument-based pitch detection called sparse non-negative decomposition detailed in section 5. This is followed by some results and discussion on further improvement envisioned for the proposed system. An earlier version of the machine learning algorithm proposed here has appeared previously in [9] by the first author and for a different application. In this paper, we have refined the learning methods and are introducing it in a more elaborate and different context.

2. GENERAL ARCHITECTURE

As mentioned earlier, we attempt to address both the problem of multiple-pitch detection and musical instrument recognition. The motivation behind this mix is the simple fact that for each given musical instrument, the timbral profiles vary along different pitches or notes produced by the instrument. Moreover, the timbral profile of a given pitch on a given instrument varies along different modes of performance for certain instruments (for example playing *ordinario* or *pizzicato* on violin family). Given this fact, we propose learning *templates* for each sound produced in each instrument once and for all, and use these templates during realtime detection.

Another important motivation behind the proposed algorithm is the simple intuition that humans tend to use a reconstructive scheme during detection of multiple pitches or multiple instruments and based on their history of timbral familiarity and music education. That is to say, in music dictation practices, well-trained musicians tend to transcribe music by conscious (or unconscious) addition of familiar pitches produced by musical instruments. The main idea here is that during detection of musical pitches and instruments, there is no direct assumption of *independence* associated with familiar patterns used for reconstruction and we rely more on *reconstruction* using superpositions.

Considering these facts, we can generally formulate our problem by *non-negative* factors. Non-negativity in this case simply means that we do not *subtract* pitch patterns in order to determine the correct combination but rather, we somehow manage to directly point to the correct combination of patterns that reconstruct the target by simple linear superposition. Mathematically speaking, given V as a non-negative representational scheme of the realtime audio signal in $\mathbb{R}_+^N$, we would like to achieve

$$V \approx WH \tag{1}$$

where W is a non-negative $\mathbb{R}_+^{N\times r}$ matrix holding r templates corresponding to objects to be detected and H is a simple non-negative $r \times 1$ vector holding the contribution of each template in W for reconstructing V. During realtime detection, we are already in possession of W and we tend to obtain H indicating the presence of each template in the audio buffer that is arriving online to the system in V. Given this formulation, there are three main issues to be addressed:

1. What is an efficient and pertinent representation for V?
2. How to learn templates in W using this representation?
3. And how to obtain an acceptable result in H in realtime?

We will give a general overview of the three questions above in the following subsections and present algorithmic descriptions in the coming sections.

2.1. Representational Front-end

Any representational front-end chosen for the formulation above, should at least meet two important properties: (1) obviously it should have enough information for discrimination between instruments, and (2) due to the non-negative formulation in equation 1 it should preserve itself when multiple instruments are present at least to a good extent and in our case, observe superposition of different instruments.

Dubnov et al. have shown in [10] that phase coupling is an important characteristic of a sustained portion of sound of individual musical instruments and show results obtained for various instruments observing consistencies in phase coupling templates for at least flute and cello. Furthermore, they note that the statistical properties of a signal due to phase variation can not be easily revealed by standard spectral analysis techniques due to the fact that second-order statistics and the power spectrum are *phase blind*. In their proposal they use a quadratic phase coupling method using higher-order statistics to obtain the phase coupling representation. Using additive sinusoidal analysis, their method is highly sensitive to the fundamental frequencies of the sound itself.

The representational front-end we propose in this paper is inspired by findings in [10] as an indirect but efficient method to represent spectral modulations of the signal, also capable of representing pitch information. We will detail this representational scheme in section 3.

2.2. Sparsity of the solution

Equation 1 simply assumes a linear combination of the previously learned templates with non-negative coefficients for reconstruction of V or learning H. The price to pay for this simplicity is of course solving for the correct results in H where there are many possible combinations of templates that might achieve a given error criterion. This issue becomes even more important if there is no mathematical independence between the basis stored in W as templates. This is a major difficulty with non-negative constraint problem solving. More specifically, for our problem, harmonic relations between pitches of an instrument and among instruments themselves always lead to various approximate solutions for H and leading to the famous *octave errors* and more.

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 verify for every music instrument and highly depends on the template representations 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 [11]. In section 5 we propose a technique to control sparsity in a non-negative constraint problem.

3. MODULATION SPECTRUM

For non-stationary signal classification, features are traditionally extracted from a time-shifted, yet short data window. For instrument classification, these short-term features do not efficiently capture or represent longer term signal variations important for the given task and can barely represent important discriminative characteristics such as spectral envelope or phase coupling for musical instrument recognition. Sukittanon et al. in [7] propose a modulation spectrum representation that not only contains short-term information about the signal, but also provides long-term information representing patterns of time variation on the spectrum itself. In this model, the audio signal is the product of a narrow bandwidth, stationary low-pass modulating random process $m(t)$ and the high-pass carrier, a deterministic function $c(t)$

$$x(t) = m(t) \cdot c(t)$$

For the model to be accurate, $m(t)$ is assumed to be non-negative and its bandwidth does not overlap with that of $c(t)$. The above model has been applied to speech and audio coding [12]. Following the observations in section 2.1 and in [10], we study the feasibility of this representation for our task and hope that $m(t)$ will provide an informative representation for pitched musical instruments.

Modulation Spectrum is based on a two-dimensional representation of the acoustic and modulation frequency or a joint frequency representation. Moreover, it does not require prior estimate of the periodicity of the signal. One possible representation of this form is $P_x(\eta, \omega)$, as a transform in time of a demodulated short-time spectral estimate where ω and η are *acoustic frequency* and *modulation frequency* respectively. To obtain this representation, we first use a spectrogram with an appropriately chosen window length to estimate a joint time-frequency representation of the signal $P_x(t, \omega)$. Second, another transform (Fourier in our case) is applied along the time dimension of the spectrogram to estimate $P_x(\eta, \omega)$. Figure 1 shows the analysis structure undertaken on the audio (top) to obtain the modulation spectrum (down). A more rigorous view of $P_x(\eta, \omega)$ is the convolution in ω and multiplication in η of the correlation function of a Fourier transform of the signal $x(t)$ and the underlying data analysis window $w(t)$, as in equation 2 [7].

$$\begin{aligned} P_x(\eta, \omega) &= \left(W^*(\omega - \frac{\eta}{2})W(\omega + \frac{\eta}{2})\right) \\ &* \; w\left(X^*(\omega - \frac{\eta}{2})X(\omega + \frac{\eta}{2})\right) \end{aligned} \tag{2}$$

Figure 2 shows this representation for one analysis frame of piano and trumpet both playing A_4 ($f_0 \approx 440Hz$). The time-span of this analysis corresponds to the length of the first transform N_1, the length of the second transform N_2 and the sampling frequency f_s which here are 2048, 32, 44100 leading to a span of almost 1.5 seconds. Frequency modulation resolution, similar to frequency and time resolution in Fourier transform, relies on the choice of N_2 and both transforms' overlap size H_1 and H_2.

Interpretation of $P_x(\eta, \omega)$ above is straightforward. The values of $P_x(\eta, \omega)$ lying along $\eta = 0$ is an estimate of the non-stationary information about the signal which, in our case, corresponds mostly to harmonic partials in the spectrum. Values along $\eta > 0$ correspond to the degree of spectral modulation. For example, in figure 2 almost all partials of the trumpet are being modulated whereas for piano (left) modulation decreases for higher partials.

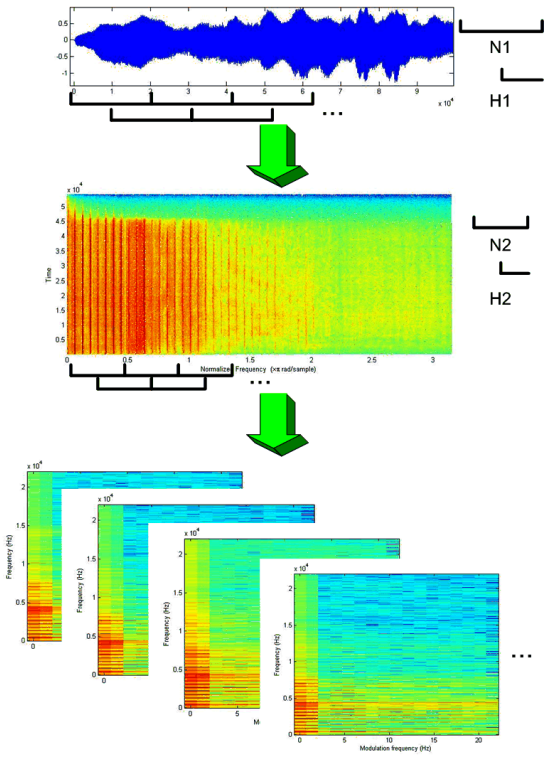


Figure 1: *Analysis structure for obtaining modulation spectrum (bottom) from audio (top).*

The non-negativity of the modulation spectrum representation and its ability to demonstrate phase coupling of instruments as modulation frequencies makes it a perfect candidate for the representation needed for V in our problem definition. Furthermore, Atlas et al. discuss associativity of this representation in [13], leading to superposition of instrument templates when several are present in the spectrum. This is demonstrated in figure 3 where modulation spectrum of flute playing A_6 alone is represented at left and modulation spectrum of a recording of piano playing A_4 and flute playing A_6 at the same time is represented at the right. Intuitively, the figure on the right of figure 3 would be a straight superposition of the left figures in figure 2 and 3 despite their different scaling.

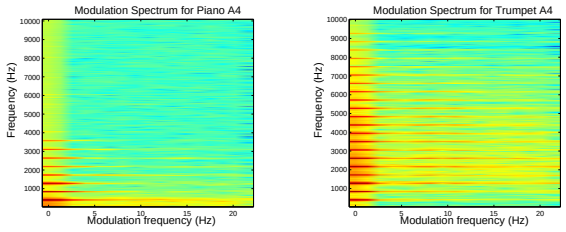


Figure 2: *Modulation Spectrum of Piano (left) and Trumpet (right) playing $A4$, zoomed over $0 - 10KHz$ acoustic frequencies.*

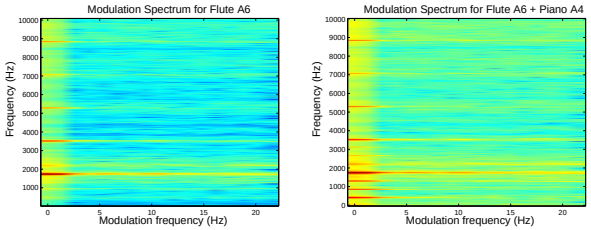


Figure 3: *Modulation Spectrum of Flute playing A_6 (left) and Piano (A_4) and Flute (A_6) playing together (right), zoomed over $0-10KHz$ acoustic frequencies.*

Hence, we adopt this two-dimensional joint frequency representation as the front-end of our system.

4. LEARNING INSTRUMENT-BASED PITCH TEMPLATES

As mentioned in section 2, the proposed system solves for the existence of previously learned instrument-based pitch templates (stored in W in our notation). Here we discuss how these templates are learned and resolve the second question in section 2. As a reminder, W contains modulation structures of all pitches of each given instrument. For example, for an acoustic piano, matrix W would contain all 88 notes as 88 different 2-D representations. To this end, training is done on databases of instrumental sounds [14, 15] using an off-line training algorithm that learns different modulation structures of instruments by browsing all sounds given 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 [8] 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 modulation spectrum of the whole audio file as described in Section 3 and the learning algorithm factorizes V as $V \approx WH$. The subscript a refers to the a^{th} template and other subscripts in equation 3 are vector indexes used during learning. 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 objects (W has two 2-D elements) 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 method helps the algorithm focus more on the harmonic and modulation structure of V. Furthermore, we require modulation frequencies higher that zero ($\eta > 0$) at each iteration by a constant factor ($Emph$ in equation 3). The idea behind this factor is to emphasize non-stationary structure of the signal, important for between instrument discrimination.

$$H_{a\mu} \longleftarrow Emph \otimes H_{a\mu} \frac{\sum_i W_{ia} V_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}}$$
$$W_{ia} \longleftarrow Emph \otimes W_{ia} \frac{\sum_i H_{a\mu} V_{i\mu}/(WH)_{i\mu}}{\sum_\nu H_{a\nu}} \quad (3)$$

When the training reaches an acceptable stopping criteria, the modulation 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 given sounds in the database. Figure 4 shows learned modulation spectrum templates for flute and violin playing A_4. During analysis, the parameters are $N_1 = 2048, N_2 = 32, H_1 = 1024$, and $H_2 = 16$ leading to a time resolution of $\sim 370ms$ and modulation upper bound of around $21Hz$ for a sampling frequency of $44100Hz$. Both templates were trained on audio files in the SOL database [15] and were converged after slightly more than 100 iterations.

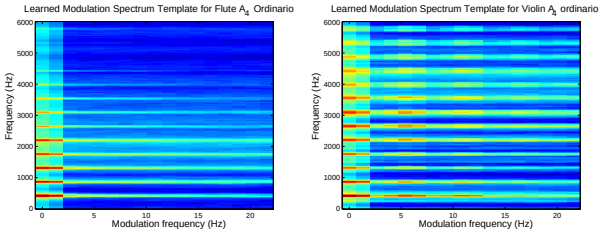


Figure 4: *Learned Modulation Spectrum of Flute A_4 (left) and Violin A_4 (right).*

Note that using this type of representation for templates has an important disadvantage. The modulation spectrum described above provides a large dimension compared to traditional short-term spectral estimations. To compensate for this, we reduce the representation by cutting frequencies above $6KHz$. This choice was adopted by trial-and-error and since most useful partial and modulation information lie below this threshold. Moreover, during learning and decomposition, we consider the 2-D modulation spectrum and templates as *images* that hence, can be reshaped into a 1-D vector and back.

5. SPARSE NON-NEGATIVE DECOMPOSITION

We are now in a position to address the third and last issue brought in section 2. Having V as the modulation spectrum analysis of real-time audio and W as stored instrument-based pitch templates, we would like to obtain H such that $V \approx WH$. As mentioned earlier in section 2.2, in order to decompose the spectrum using learned pitch templates, the solution needs to be sparse. One of the useful properties of the original NMF [8] 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. In this section, we introduce a modified sparse non-negative decomposition algorithm.

Numerous sparseness measures have been proposed and used in the past. 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 [16], the choice of sparseness measure is not a minor detail but has 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 [17]. Due to real-time considerations and the nature of sparseness in audio signals for pitch determination we propose a modified version of Hoyer's method described in [17].

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 a 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 mathematically in Equation 4.

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

Algorithmic realization of this sparsity constraint is a straightforward and cheap iterative procedure that projects the results first to the ℓ_ϵ hyperplane and then solves for the intersection of this projection with the hyperplane possessed by ℓ_2. Figure 5 shows a synthetic signal (left) and its sparse projection using the proposed procedure with $\ell_\epsilon = 0.9$ and ℓ_2 equal to signal's energy.

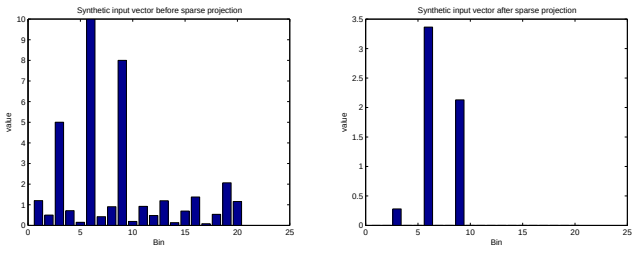


Figure 5: *Synthetic signal before sparse projection (left) and after (right).*

For non-negative sparse decomposition, 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 [17], 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 or to previous value of H in sequence
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

Here, step (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/instrument detection, the ℓ_2 norm is provided by the spectrum energy of the realtime signal (directly calculated from the column in V corresponding to $\eta = 0$) 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 $N_1 \times N_2$ where here we use $N_1 = 2048$ and $N_2 = 32$ and further reduced (along N_1) to capture modulation structures up to about $6KHz$ acoustic frequency in a sampling rate of $44.1KHz$. Equivalently, W would be a matrix of SizeOf$(V) \times m$ with m as the number of templates and H would be a vector of size m.

6. EVALUATION

A clean evaluation of a systems such as the one proposed in this paper bears practical difficulties. It should be clear by now that we are attempting towards a multi-instrument transcription of music signals in form of a *piano roll* presentation. To evaluate such representation one needs an annotated and transcribed music of the same type to an order of milli-seconds. There has been recent attempts in creating such database but for monophonic music or in the best case, polyphonic but mono-instrument sounds (such as piano music). Evaluation procedures that has been undertaken so far in the literature do not seem to be close to ideal either. In systems where the authors aim for multiple-instrument identification, the pitch information is missing [4, 6]. Otherwise, other researchers aimed at manual mixing of single note recordings of different instruments as the basis of their evaluations (for example in [5]).

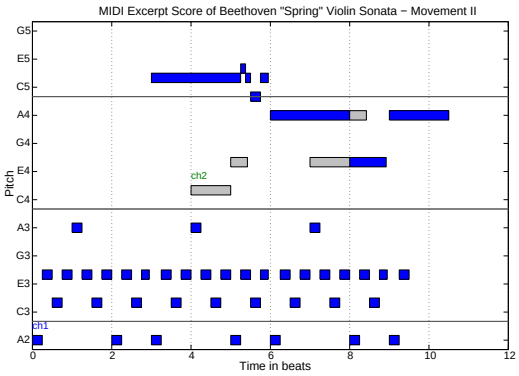
As mentioned in the beginning of this paper, our system is destined towards real-time applications in computer music systems. Therefore, it is vital that the evaluation procedure is done on real music recordings and in real musical situations despite the difficulties of such approach.

In this section we showcase the performance of our system in two manners: (1) A subjective evaluation where we demonstrate the real-time output of the system on short musical examples and compare the results visually with the piano-roll representations of their scores. (2) An objective evaluation where we evaluate the algorithm on mixed music recordings and provide detailed results.

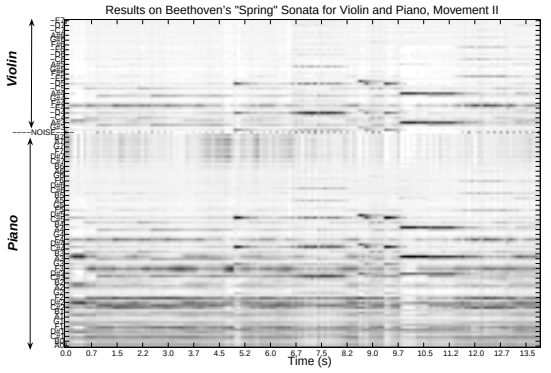
All audio files and results used during evaluation as well as more fine-tuned and detailed images can be viewed on the project's website[1].

6.1. Subjective Evaluation

Figures 6(a) and 6(b) show the performance of the system (bottom) on a real recording of the first phrase of Beethoven's Sonata for Piano and Violin *(The spring)*. A piano roll representation of the MIDI score of the same phrase is represented in figure 6(a) where the Piano section has darker color than the violin part. In the sample result (figure 6(b)), decomposition results are represented as an image where the Piano and Violin results occupy a separate space. The Y-axis represents pitches for each instrument (88 for Piano and 41 for Violin) and are sorted in ascending order to resemble a piano-roll representation.



(a) MIDI Score Representation

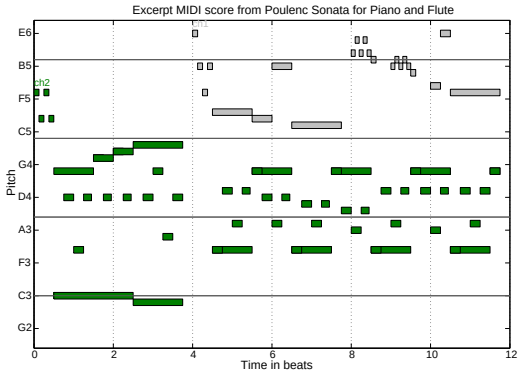


(b) Decomposition Results

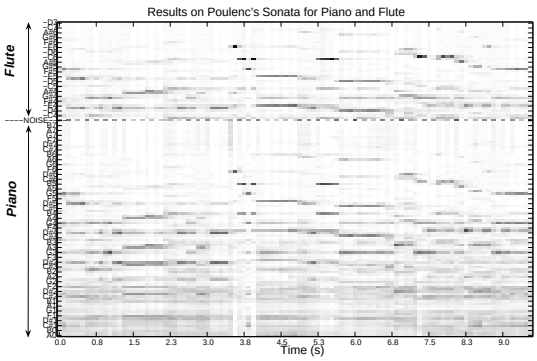
Figure 6: *Sample result (1): Beethoven's* Spring *Violin-Piano Sonata, 2nd movement, starting bars with score (top) and system result (bottom).*

Similarly, figures 6.1 shows the performance of the system (bottom) on a real recording of a few bars from Francis Poulenc's Sonata for Flute and Piano with the score excerpt shown in a piano roll presentation on the top. Here again, the flute section is represented by a lighter color in the piano roll score of figure 7(a).

[1] http://recherche.ircam.fr/equipes/ temps-reel/suivi/Arshia/DAFx07/



(a) MIDI Score Representation



(b) Decomposition Results

Figure 7: *Sample result (2): Francis Poulenc's Sonata for Flute and Piano (excerpts).*

The parameters used for training and real-time decompositions for both examples are as follows: $F_s = 44100Hz, N_1 = 4096, N_2 = 32, H_1 = 256$, and $H_2 = 16$. During real-time analysis, these choices leave us with an analysis time-span of almost 3 seconds and response delay of 93 milli-seconds.

For a subjective evaluation, it suffices to compare the score piano-roll representation with the result images in figures 6(b) and 7(b). For the Piano parts, specially for the Piano and Violin example, the low notes are hard to distinguish, especially with the current scaling of the paper. However, in both figures the main contour of both scores can be easily detected with the eyes. An important remark here is the (weaker) presence of the first instruments (Violin and Flute) in the accompaniment instrument (Piano). Both detailed analysis of the results and the confusions are addressed in the next section.

6.2. Objective Evaluation

Due to the lack of high-resolution annotated and transcribed ensemble recordings, we tend to mix transcribed and annotated monophonic music for different instruments and evaluate the performance of our system on the mixed audio. The advantage of this approach is that first, we will be dealing with real music recordings and two, we can easily calculate precisions for instrument/pitch

detection across instruments since the annotations for each instrument are separate. The disadvantage, of course, is that after-the-fact mixing of two instruments can not demonstrate eventual spectral fusions common in ensemble recordings (which was not the case in our subjective evaluation). For this paper, we focus on two-instrument mixes and address more enhanced evaluations in another publication.

Audio and annotation files used for this evaluation session are taken from the *Score Following Evaluation Task* prepared by the author for MIREX 2006 [18] and also from a previous experiment reported in [9]. Table 1 shows the specification of Audio and (aligned) MIDI files used during the evaluation. The MIDI annotations that come with each audio file, provide aligned score to audio information. Note that although these annotation were created automatically and double checked using a high-resolution analysis software, they are not perfect especially in the case of Piano because of the difficulty in assigning correct note lengths in a polyphonic situation and in the presence of the piano pedal. This issue is quite present for piece number 2 which is usually played with a high utilization of the sustain pedal.

Table 1: Specification of Audio and Midi used for evaluation

#	Piece Name	Time	Events	Instr.
1	Mozart's *Piano Sonata in A major,K.331*	9:55	4268	Piano
2	Chopin's *Nocturne no.2, opus 9*	3:57	1291	Piano
3	Bach's *Violin Sonata 1, Movement 4*	3:40	1622	Violin
4	Bach's *Violin Sonata 2, Movement 4*	5:13	2042	Violin

For this evaluation we created two Piano and Violin mixtures according to their lengths: $1+4$ and $2+3$, and ran the system on both mixtures. Mixing starts at time zero so since the piano recordings are always longer in our case, we are sure that during the length of the violin parts there is always activity in both instruments and we are left with some extra piano-only section in the end. During evaluations, for each note event in the aligned score, we look at the corresponding frames of the analysis 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. Moreover, since we do not have any specific temporal model and also the ending of notes are usually doubtful (especially for Piano) we can consider (subjectively) positive detection during at least 80% of a note life to be *acceptable* and refine the precision. Cross classification can be computed in the same way by switching the piano and violin references between results.

Tables 2 and 3 show confusion matrices out of the above evaluation for each mix (where numbers refer to specifications in table 1). This confusion matrix is to be read as follows: the row elements correspond to the results being evaluated and the column elements correspond to the reference alignment being used for evaluation. For example, element $(1,2)$ refers to the percentage within which the system has decoded violin elements in the Piano results. Therefore, it is natural that this confusion matrix is not symmetrical. On another note, values in the confusion matrices do not add up to 100%. Each row column of the matrix represents the instances in a predicted instrument class while each row represents the instances in an actual class. These measures obviously are not representative of all sorts of errors a transcription system can undergo. For a detailed description of different kinds of errors in a music transcription problem we refer the reader to [19]. In what follows we emphasize the *precision* rate and inner-instrumental confusion thereof through the results. Finally, note that precision rates in Tables 2 and 3 correspond to both (multiple) pitch and instrument classification where the reference for both is obtained from the aligned MIDI scores to audio.

Results in tables 2 and 3 suggest that the precision rate (diagonal values) for the violin parts are significantly higher than the Piano part. This is mainly due to the fact that the Violin sections (files 3 and 4 in table 1) are much louder than the piano audio files and we did not normalize the loudness before mixing to be as natural as possible. Other reasons for the deficiency of the Piano pitch/instrument detection comes from the fact that in both pieces there is an extensive use of sustain pedals which confuses the system when trying to match templates for reconstructing the ongoing modulation structure. Furthermore, lower Piano precision in Table 2 is because the sustain pedal is being used much more in Chopin's *Nocturne* than Mozart's *Sonata*, which is stylistically reasonable.

Confusion Matrices

	Piano	**Violin**
Piano	45%	9.2%
Violin	17.1%	67.5%

Table 2: Mix of $2+3$

	Piano	**Violin**
Piano	52.8%	17.9%
Violin	24.3%	89.3%

Table 3: Mix of $1+4$

Overall, given the nature of the problem, that is simultaneous multiple-pitch and multiple-instrument detection in real-time, the results are satisfactory and not far from other state-of-the-art systems cited earlier in section 1.

7. CONCLUSION AND DISCUSSION

In this paper we presented a technique for detection of multiple-pitches produced by multiple-instruments and in real-time. The core of the proposed system relies on a rather simple machine learning principle based on sparse non-negative constraints. The simplicity behind this algorithm is due to observations on the nature of musical instruments and basic facts regarding musical pitch and timbre structures. After formulating the problem we discussed three main issues regarding the formulation and presented solutions for each.

If the proposed method is to be useful in computer music applications, the precision rates should obviously be higher than the ones in Tables 2 and 3. The work presented in this paper is regarded as a first step towards the complex problem of multiple pitch and instrument recognition in real-time. However, obtained results with a more rigorous evaluation framework as stated before, are close to the state-of-the-arts reported elsewhere. Here we elaborate on the future directions of this project and on how the proposed algorithm can be improved.

The on-line learning algorithm developed in section 5, uses a simple gradient descent update that is projected at each iteration to assure sparsity. From a machine learning perspective, gradient descent updates are not always the best solution and more intelligent optimization techniques such as convex optimization and semi-definite programming would be more suitable. However, for this experiment and due to our real-time constraints, we

adopted the gradient descent approach and will report on more advanced methods in later publications. Also, the sparse constraints introduced are quite powerful in order to avoid inner-instrumental overuse of templates but does not directly address avoiding inner-instrumental confusions. New sparsity measures should be experimented in order to overcome this issue. On another note, we use the amplitude (or absolute value) of the complex modulation spectrum reported in section 3. Later improvements will consider the complex values or in other words, the phase information of the modulation spectrum directly into the decomposition algorithm.

We conducted subjective and objective evaluations of the algorithm but in the limited case of two instruments. A more elaborate evaluation procedure is needed to discover true deficiencies and outcomes of the proposed algorithm. However, for the given task, the evaluation frameworks that has been introduced so far in the literature do not provide sufficient and accurate data for such fine-tuned evaluation. We will be elaborating on this subject to further improve the test-bed that can lead to better frameworks and improved systems.

Finally, as mentioned earlier in the paper, one of the main motivations for this research is to provide real-time tools for the computer musicians and researchers with their growing need for real-time detection tools. The algorithm and application proposed in this paper is currently under development for MaxMSP[2] and Pure Data[3] real-time computer music environments and will soon be available for free download[4].

8. 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., pp. 45–72. IEEE Press / Wiley, 2006.

[2] Kunio Kashino and Hiroshi Murase, "A sound source identification system for ensemble music based on template adaptation and music stream extraction," *Speech Commun.*, vol. 27, no. 3-4, pp. 337–349, 1999.

[3] J. Eggink and G. J. Brown, "A missing feature approach to instrument identification in polyphonic music," in *IEEE ICASSP*. 2003, Hong Kong.

[4] Slim Essid, Gaël Richard, and Bertrand David, "Instrument recognition in polyphonic music based on automatic taxonomies.," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 14, no. 1, pp. 68–80, 2006.

[5] Arie Livshin and Xavier Rodet, "The significance of the non-harmonic "noise" versus the harmonic series for musical instrument recognition," in *International Symposium on Music Information Retrieval (ISMIR)*, 2006.

[6] T. Kitahara, K. Komatani, T. Ogata, H.G. Okuno, and M. Goto, "A missing feature approach to instrument identification in polyphonic music," in *IEEE ICASSP*. 2006, Toulouse.

[7] Somsak Sukittanon, Les E. Atlas, and James W. Pitton, "Modulation-scale analysis for content identification," *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 3023–3035, 2004.

[8] 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.

[9] Arshia Cont, "Realtime multiple pitch observation using sparse non-negative constraints," in *International Symposium on Music Information Retrieval (ISMIR)*. October 2006, Victoria, CA.

[10] S. Dubnov and X. Rodet, "Investigation of phase coupling phenomena in sustained portion of musical instruments sound," *Acoustical Society of America Journal*, vol. 113, pp. 348–359, Jan. 2003.

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

[12] M.S. Vinton and L.E. Atlas, "Scalable and progressive audio codec," *IEEE ICASSP*, vol. 5, pp. 3277–3280, 2001.

[13] Les Atlas and Christiaan Janssen, "Coherent modulation spectral filtering for single-channel music source separation," in *IEEE International Conference in Acoustics and Speech Signal Processing (ICASSP)*, 2005.

[14] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka, "Rwc music database: Popular, classical and jazz music databases.," in *International Symposium on Music Information Retrieval (ISMIR)*, 2002.

[15] 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.

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

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

[18] Arshia Cont, Diemo Schwarz, Norbert Schnell, and Christopher Raphael, "Evaluation of real-time audio-to-score alignment," in *International Symposium on Music Information Retrieval (ISMIR)*. October 2007, Vienna, Austria.

[19] G. Poliner, D.P.W. Ellis, A.F. Ehmann, E. Gomez, S. Streich, and B. Ong, "Melody transcription from music audio: Approaches and evaluation," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 15, no. 4, pp. 1247–1256, 2007.

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

[3] `http://crca.ucsd.edu/~msp/software.html`

[4] For progress, see: `http://recherche.ircam.fr/equipes/temps-reel/suivi/Arshia/DAFx07/`