REALTIME AUDIO TO SCORE ALIGNMENT FOR POLYPHONIC MUSIC INSTRUMENTS USING SPARSE NON-NEGATIVE CONSTRAINTS AND HIERARCHICAL HMMS

Arshia Cont

Ircam - Realtime Applications Team, 1 Place Igor Stravinsky, Paris 75004. and
Center for Research in Computing and the Arts, UCSD, San Diego, CA.
cont@ircam.fr

ABSTRACT

We present a new method for realtime alignment of audio to score for polyphonic music signals. In this paper, we will be focusing mostly on the multiple-pitch observation algorithm proposed based on realtime Non-negative Matrix Factorization with sparseness constraints and hierarchical hidden Markov models for sequential modeling using particle filtering for decoding. The proposed algorithm has the advantage of having an explicit instrument model for pitch obtained through unsupervised learning as well as access to single note contribution probabilities which construct a complex chord instead of modeling the chord as one event.

1. INTRODUCTION

Traditionally, score following (or realtime audio to score alignment) was introduced for two main reasons : First for musical accompaniment of human performers with a computer and second, for synchronization of live sound processing algorithms for instrumental electroacoustic composition. Today, audio to score alignment has found applications in wider domains such as audio indexing and pedagogical systems.

Among different systems available the only polyphonic score follower systems that we know of are reported in [1] and [2] by Raphael and Dannenberg respectively. Raphael's system in special, has been evaluated and demonstrated for realtime polyphonic performances on the Piano. In both works, the spectral model for a chord is assumed to be fixed and there is no explicit notion of instrument model. In the proposed system, the instrument model is learned explicitly as described in Sections 3.2. Moreover, we have access to individual pitch contribution to a complex chord during the observation process whereas in both systems the model considers a chord as one fixed object when calculating probabilities. For a detailed review of other systems, we refer the curious reader to [3].

In Section 2 we describe the general architecture of the proposed polyphonic score follower. Due to limited space, we focus mainly on the multiple pitch observation system based on non-negative matrix decomposition (NMD) with sparseness constraint in Section 3. The sequential modeling will be briefly described in Section 4. In the end, we provide results and further discussions on the subject.

2. GENERAL ARCHITECTURE

As a basic description, a score following system takes audio chunks in realtime as input and provides an alignment to the music score in the output. In our system, audio input is represented by feature vectors y_k for each realtime frame k. This vector is used to compute the *observation likelihood* $P(y_k|x_k)$ of being on the event x_k in the score. The score is defined as a hidden Markov chain of states representing sequential events as described in Section 4 and providing *transition priors* $P(x_{k-1}|x_k)$. Current *belief* of the system is calculated using the observation likelihood, transition priors and previous belief of the system $P(x_{k-1}|y_{1:k-1})$.

The system proposed in this paper is recursive in nature. This means that the observation (or acoustic) model and the sequential model are tied together and the way they are applied depends on where we are currently in the score through two different decoding schemes. This characteristic is the main feature of our sequential modeling which is based on Hierarchical Hidden Markov Models [4] described in details in Section 4. Generally speaking, it consists of two main hierarchies, the higher modeling the atoms in the score (notes, chords and silences) and the lower modeling time. A *vertical transition* switches from the first hierarchy to the second by the introduction of a new event by the performer and a *strike* or *end* transition sends the system back to the first hierarchy as shown in Figure 1.

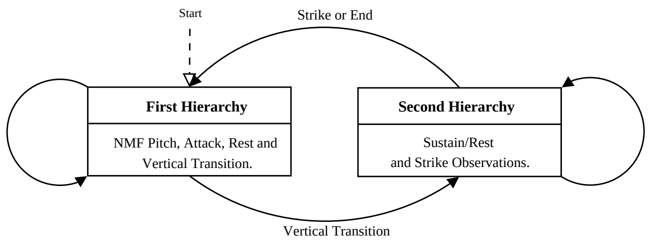

Start
Strike or End
First Hierarchy
NMF Pitch, Attack, Rest and Vertical Transition.
Second Hierarchy
Sustain/Rest and Strike Observations.
Vertical Transition

Fig. 1. General diagram of score follower's realtime message passing with appropriate observations for each hierarchy.

With respect to this structure there are two separate decoding and *belief update* processes for each hierarchy level. If the current position is on the first hierarchy, the decoding is done using *Sequential Importance Sampling with Resampling* (SISR) with a fixed number of particles and using *residual resampling* [5]. If we are in the second hierarchy, the decoding is done using the regular forward (α) variable that contains the transition probabilities and observations of the second level.

The observation process can be regarded as two independent systems : one providing *attack, sustain* and *rest* probabilities (showing the attribute of the ongoing music signal according to the three categories) and another observation process for polyphonic pitch. The *attack, sustain* and *rest* model has been previously discussed along with its training algorithm in [3] and will not be the subject of this paper.

This research was supported by the EU project I-MAESTRO (see http://www.i-maestro.org/).

3. MULTIPLE PITCH OBSERVATION MODEL

Multiple pitch analysis is currently an important topic in music and speech signal processing among others. It is mostly applied for multiple-f_0 recognition and music transcription. However, most of the multiple-f_0 analysis systems available are either computationally very heavy or unadaptable for realtime applications. For a detailed overview of different systems and approaches we refer the reader to [6]. In this section we introduce a novel algorithm for multiple pitch analysis of music signals. The proposed algorithm is based on non-negative matrix factorization (NMF) [7]. NMF was first used to find parts representation of image data. In general terms, given a matrix V, the original NMF algorithm is an unsupervised algorithm that learns desired number of templates stored as columns in W and a decomposition matrix H through multiplicative updates which results in $V \approx WH$. Non-negativity constraint in the algorithm implies additivity of *parts* in the representation in order to construct V.

In our application, W is learned offline and contains spectral templates for all pitches of a given instrument. During live performance W is kept fixed and the algorithm tries to learn a decomposition of the ongoing audio spectrum using the fixed templates (i.e. it only learns H). A similar approach has been recently reported by Sha and Saul [8] where they use regular NMF for both learning and realtime decomposition. Our approach is strictly different in the performance stage with use of Non-negative Matrix Decomposition with sparseness constraints. Later in this section we compare the output of the two and show that the proposed algorithm undergoes a better representation for polyphonic pitch analysis. Sparse non-negative constraint has been previously introduced by Hoyer [9]. Our approach is different from Hoyer's with use of different sparse measures adapted for multiple pitch observation and further refinements for realtime decomposition which is not the case in [9].

We start this section by introducing the signal processing front-end used for the representation of the signal (V). We continue by introducing the training phase of the pitch observation (learning W), demonstrating the sparse non-negative matrix decompotisìon algorithm for realtime performance and how the likelihood probability is calculated from the observation.

3.1. Signal processing front-end

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 [10]. The short-time Fourier transform (STFT) already in use in the observation model is an efficient tool for instantaneous frequency (IF) estimation [11]. Given $z(\omega,t)$ as the analytical form of the STFT, the mapping

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

can be computed efficiently and in real-time using STFTs [11] and the fixed points of this mapping can be extracted using the following criteria [10] : $\lambda(\omega^*,t) = \omega^* \quad and \quad \frac{\partial\lambda}{\partial\omega}|_{\omega=\omega^*} < 1$. As a result, vector V would be non-negative amplitudes of the fixed-point instantaneous frequency. The same representation has been reported in [8].

3.2. Learning pitch templates

Once V is prepared in realtime, it will be decomposed using pre-learned pitch templates stored in W. The idea here is that the pitch observation is *familiar* with all pitches that can be produced by an instrument. Moreover, we are interested in instrument specific pitch models to be learned explicitly by the system. For example, for following a piano score, matrix W would contain all 88 pitches as 88 different columns. Learning pitch templates is done off-line and only once for each instrument using databases of instrumental sounds [12]. For each audio file in the database (corresponding to a known pitch), training is an iterative NMF with symmetric kullback-leibler divergence as error measure. In addition, since each audio spectrum contains additional information such as noise besides a harmonic spectrum, we put two constraints on the learned W. First, we decompose V into two vectors (W has two columns) where we only *learn* one vector and have the other fixed as white noise. This criteria helps the algorithm focus more on the harmonic structure of V. Furthermore, we constrain this harmonic structure in each learning iteration by an envelope. This envelope is constructed from the pitch information of the audio file (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. This constraint improves common octave and harmonic errors.

Equation 1 shows the NMF updates with kullback-leibler divergence for each iteration as described above and adapted from [7]. Here Env is the envelope constraint for each pitch, $\otimes$ is an element by element multiplication and V is the short-time fixed-point instantaneous frequency representation of the audio file described before.

$$\begin{aligned} H_{a\mu} &\longleftarrow Env \otimes H_{a\mu} \frac{\sum_i W_{ia} V_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}} \\ W_{ia} &\longleftarrow Env \otimes W_{ia} \frac{\sum_i H_{a\mu} V_{i\mu}/(WH)_{i\mu}}{\sum_v H_{av}} \end{aligned} \tag{1}$$

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.

3.3. Sparse non-negative Decomposition for realtime application

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. 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.

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. For our application there are two sparseness measures working in parallel : an approximation of ℓ_ε norm by the *tanh* function as $g(x) = tanh(|ax|^b)$, where a and b are positive constants with b greater than 1, and an ℓ_2 norm which is crucial for the normalization of the results. This measure is demonstrated in Equation 2 with N being the dimension of the space.

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

In order to obtain H while V (signal processing representation) and W (pitch templates) are known, we use a gradient descent update, $H = H - \mu_H W^T(WH - V)$, instead of the original NMF multiplicative update and project each vector in realtime to be non-negative and have desired ℓ_2 and ℓ_ε norms. This is achieved by first projecting the new H to the ℓ_ε space and solving a quadratic equation where the projection has the desired ℓ_2 norm for each iteration. Note that this algorithm can be adapted for other applications.

For the multiple pitch observation model, ℓ_2 measure will be provided in realtime to the algorithm from the log-energy of the spectrum. The ℓ_ε constraint represents the sparsity of the output, the bigger the more sparse and vice versa. For this reason, depending on the local activity of future events in the score ℓ_ε can be adjusted in realtime.

In order to understand the importance of sparseness constraints, Figure 2 demonstrates three piano chords (shown as the first three chords in Figure 5(a)) using NMD without sparseness constraints and the main portion of Figure 3 shows the same analysis but with the proposed algorithm. Each value on the *y-axis* corresponds to a specific pitch in the W matrix (in this case Piano) sorted according to their frequency. From the figure, it can be noted that the NMD with sparseness constraint has achieved a more compact representation than the one without. While the activity in the regular NMD spreads over all frequency regions, it can be seen visually that for the sparse representation it is more active in the frequency region that the pitches occur. Regular NMD has a natural tendency to reconstruct a pitch with its octave, dominant and subharmonic whereas NMD with sparseness constraint, due to the imposed constraints tends to enforce main pitches for each chord. Moreover, because of the use of energy as ℓ_2 constraint, we can visually distinguish between three chords and at the onset we can see high activity in the noise template which corresponds to transition states.

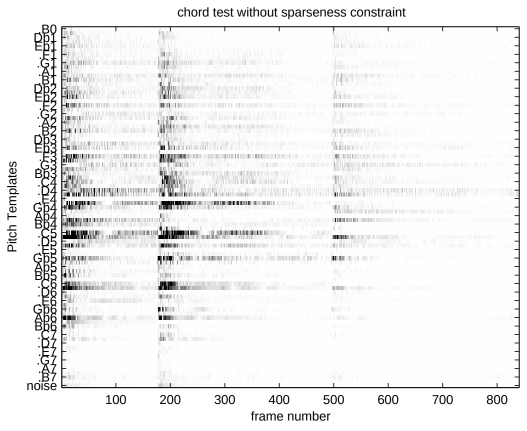


Fig. 2. Chord analysis for the first three chords of Figure 5(a) using regular NMF without sparseness constraints.

3.4. Observation probability

The proposed system above can be used independent of the score follower for multiple pitch analysis of polyphonic instruments with further consideration. For a score following application, we are interested in the probability of each chord in the score played by the musician which will serve as observation likelihoods $P(y_k|x_k)$ during decoding. Moreover, this probability should consider human errors playing the score (for example playing one or more notes off the written pitch in the score). For this purpose, using the given score, we construct Gaussian mixtures over the pitch templates for each chord. For each note in a chord, a Gaussian will be centered on the index of that note in the overall pitch template. Each Gaussian corresponds to a note and moreover, scores possible human errors if adjacent pitches are played. The NMD decomposition at each frame will be filtered by these gaussian mixtures and therefore the normalized energy in each filtered vector correctly represents the probability of each chord at each frame.

Figure 3 represents the steps mentioned above for the NMD results from previous section and for the first three chords in Figure 5(a). The right portion of Figure 3 demonstrates the Gaussian mixtures for the first three chord and the figure on the bottom shows the chord probabilities extracted from filtering the NMD for each frame. Note that the three chords in analysis in this example have at least two common pitches among each other and hence the shown example is not an ideal analysis.

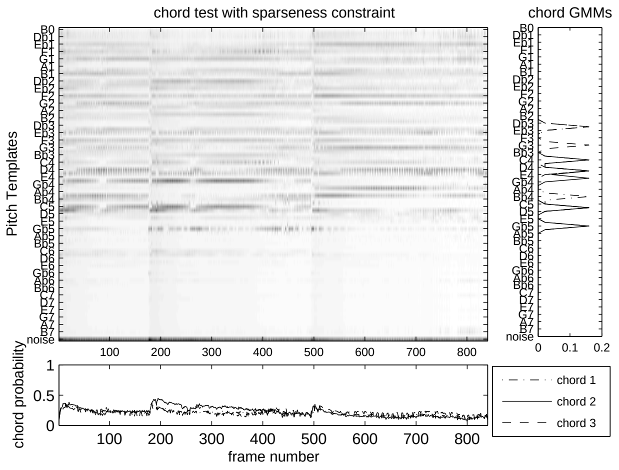


Fig. 3. Chord analysis for the first three chords of Figure 5(a). Figure on the right shows the Gaussian mixtures constructed from the score on the pitch templates. The figure on the bottom shows the probability of chords from the normalized energy of each Gaussian mixture structure multiplied with the raw pitch observation for each frame.

4. SEQUENTIAL MODELING

Once the observations are gathered for an ongoing audio frame, they should be used for alignment to a sequential model of the music score. In our model, observations are calculated for pertinent states and serve as likelihood probabilities in a Bayesian framework. The prior information are transition probabilities of the sequential model of the given score. The sequential model used in our system is inspired by Hierarchical Hidden Markov Models [4] which we will call Score Hidden Markov Model (SHMM) throughout this paper.

The SHMM consists of two different hierarchies. The first level consists of high-level musical states that correspond to atoms in the score, that is notes, chords and rests and the second level contains temporal modeling according to the time-value of each event in the score. Since we are not dealing with music transcription we model each chord as one state. The observations pertinent to the first level are *attacks*, *multiple pitch* and *rest* observations. Moreover, the SHMM model described here allows us to have different activations and entries for each high-level state according to where we are in the sequence. Figure 4 shows the note model used in SHMM. The first-level in the figure corresponds to a note or chord event in the score and its corresponding observations are *attack* and *multiple pitch*. Once this event is detected, SHMM enters automatically into the second hierarchy, the temporal states. For a note/chord state, these states correspond to *sustain* observation. The transition probabilities and the number of states are set using the time-value of the event according to the score and a temporal modeling scheme. The time modeling used here is the same as [3]. There is an additional `end` state in the second hierarchy which automatically sets index to the last first level state observed. Other than passing through the time model, there are *strike* observations whose function is to halt the temporal model if there is a sudden *rest* or *attack*. The rest model is essentially the same but with different observations and only one strike state for attacks.

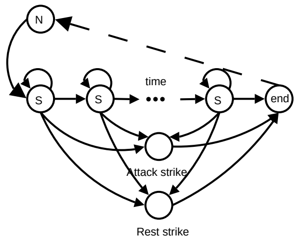
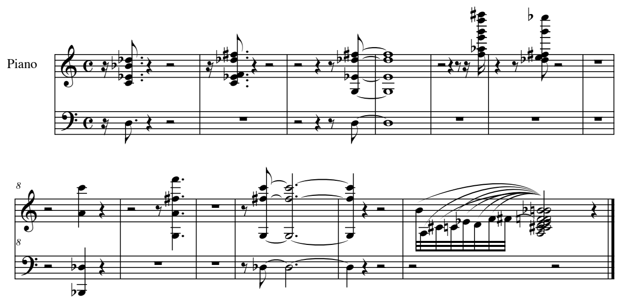


Fig. 4. Score Hidden Markov Model – one note/chord model

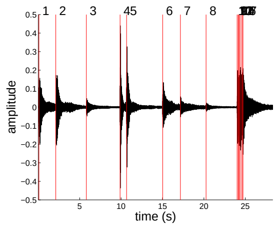
The final SHMM of a given score, would be a chain of note/chord and rest models as described above according to the music score.

5. RESULTS AND DISCUSSION

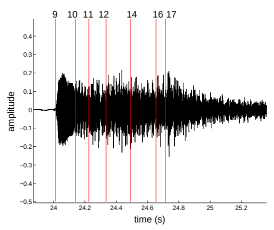
Figure 5 shows results for a simulation of the realtime audio to score alignment with the score shown in Figure 5(a) as an excerpt of a contemporary music piece ("Pluton" by Philippe Manoury) with the alignment results in Figure 5(b) and (c) where Figure 5(c) shows a zoomed view for the last music bar of the score in Figure 5(a). Each index in Figure 5(b) and (c) shows the event (note/chord) number in the order they appear in the score.



(a) Score excerpt of "Pluton" for Piano and live electronics by philippe Manoury performed by Andrew Russo.



(b) Alignment results.



(c) Zoom view of last bar.

Fig. 5. Example of placing a figure with experimental results.

In this paper, we concentrated mostly on the multiple pitch feature of the proposed system. Note that this system is independent of the proposed application and can be used freely in other contexts. Despite learning the templates from a database, the pitch observation has proven to be robust for various performance situations and the observation probabilities maintain the same relative information. Other aspects of this system including particle filtering and results for other instruments were not discussed and will be reported in a separate publication. The realtime version of this system is developed for the Max/MSP realtime graphical programming environment and is currently planned for production of various contemporary music pieces.

6. ACKNOWLEDGMENT

The research was supported in part by the European Commission under Contract IST-026883 I-MAESTRO. The author would like to acknowledge the EC IST FP6 for the partial funding of the I-MAESTRO project (`http://www.i-maestro.net/`).

7. REFERENCES

[1] Christopher Raphael, "A Hybrid Graphical Model for Aligning Polyphonic Audio with Musical Scores," *ISMIR*, 2004.

[2] Roger B. Dannenberg and Ning Hu, "Polyphonic Audio Matching for Score Following and Intelligent Audio Editors," in *Proceedings of the ICMC*, 2003, pp. 27–34.

[3] Arshia Cont, Diemo Schwarz, and Norbert Schnell, "Training ircam's score follower," in *ICASSP*. 2005, Philadelphia.

[4] Shai Fine, Yoram Singer, and Naftali Tishby, "The hierarchical hidden markov model : Analysis and applications," *Machine Learning*, vol. 32, no. 1, pp. 41–62, 1998.

[5] Jun S. Liu and Rong Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[6] 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).

[7] 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.

[8] 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.

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

[10] Hideki Kawahara, H. Katayose, Alain de CheveignÈ, 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.

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

[12] Guillaume Ballet, Riccardo Borghesi, Peter Hoffmann, and Fabien LÈvy, "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.