A DYNAMIC PROGRAMMING VARIANT OF NON-NEGATIVE MATRIX DECONVOLUTION FOR THE TRANSCRIPTION OF STRUCK STRING INSTRUMENTS

Sebastian Ewert *Mark D. Plumbley* *Mark Sandler*

Queen Mary University of London, UK

ABSTRACT

Given a musical audio recording, the goal of music transcription is to determine a score-like representation of the piece underlying the recording. Most current transcription methods employ variants of non-negative matrix factorization (NMF), which often fails to robustly model instruments producing non-stationary sounds. Using entire time-frequency patterns to represent sounds, non-negative matrix deconvolution (NMD) can capture certain types of non-stationary behavior but is only applicable if all sounds have the same length. In this paper, we present a novel method that combines the non-stationarity modeling capabilities available with NMD with the variable note lengths possible with NMF. Identifying frames in NMD patterns with states in a dynamical system, our method iteratively generates sound-object candidates separately for each pitch, which are then combined in a global optimization. We demonstrate the transcription capabilities of our method using piano pieces assuming the availability of single note recordings as training data.

***Index Terms*—** Non-Negative Matrix Deconvolution, Music Transcription, Convolutive Signal Models, Dynamical Systems.

1. INTRODUCTION

Automatic music transcription has a long history in musical signal processing and is often considered a key technology for various tasks in computational musicology and music information retrieval [1–4]. While discriminative methods have been used successfully for transcription, for example using support vector machines [5] or recurrent neural networks [6], most state-of-the-art methods use factorization approaches, with non-negative matrix factorization (NMF) and its variants being the most successful, see for example [7–11]. In general, the underlying idea of NMF-based methods is to model a given time-frequency representation of a recording as a mixture of note- or sound-specific spectral template vectors and to estimate their individual activity over time. While every NMF-variant has its specific advantages and disadvantages, an overarching principle is that spectral properties are decoupled from temporal properties in the model, i.e. neither do the activations provide information about how a note spectrally manifests nor do the templates describe when a note occurs or how it evolves. While this simplifies the model and accelerates the parameter estimation, it also limits the expressivity of the model, in particular if the sound produced by an instrument for a note is non-stationary. More precisely, most approaches employ up to three spectral templates to model a sound or note, which can represent for example the attack, sustain and release segments of a note. However, this leads to several problems. First, often the templates can be used jointly to represent a sound which is often inappropriate, e.g. attack and release templates together. Second, there is no enforced temporal progression of spectral templates. For example, if a key on a piano is hit, we expect a sustain template to occur after the attack after a certain amount of time. Third, even with three templates the full non-stationary behavior of many instruments cannot fully be modeled. Fourth, activations between neighboring frames are not coupled. Such coupling is useful for modeling instruments for which the activation of an attack template has direct implications for the activation of the subsequent sustain templates.

To address the first two problems, a variant of NMF referred to as non-negative factorial hidden Markov model (N-FHMM) was introduced where several sets (or *dictionaries*) of templates are employed in parallel and a Markov process governs the transition between the sets [12, 13]. This way, the model introduces a temporal dependency between spectral templates and can enforce or favor certain transitions between them, e.g. from attack over sustain to release. While effectively solving the first two problems mentioned above, it does not address the last two, and additionally introduces a new one: computational costs. Similar to a regular factorial HMM [14], the size of the state space used in an N-FHMM is exponential in the number of independent Markov chains and dictionary elements used, which can limit its practical use for transcription to only simple cases [15]. Replacing a parameter estimation based on expectation-maximization with a variational formulation can only mitigate this problem to some degree [16]. Also, decoupling a large number of parameters as typically done with either of these two principles can make certain degenerate solutions more likely.

Non-negative matrix deconvolution (NMD) as introduced in [17] is capable of addressing all four problems mentioned above. The idea is to use, instead of individual spectral templates, entire time-frequency patterns, which concatenate several templates over time, as building blocks within the model. The use of patterns does not only enforce a specific temporal order for the templates inside them, but also effectively couples their activations. However, since many instruments allow a fine-grained control over their acoustic properties, an extremely large number of patterns would be required to properly model their sound, which practically limits the application of NMD to percussive instruments, where such control is typically very limited. For example, NMD has been successfully employed to transcribe drum performances [18, 19]. However, since the patterns have a fixed length, NMD has not been used to transcribe struck string instruments such as the piano (or some plucked string instruments including the harpsichord), where notes are typically of variable length.

The idea in this paper is to construct a signal model appropriate for the class of struck string instruments capable of producing sound of variable length. Combining ideas from both N-FHMM and NMD, each sound source (corresponding to individual keys on a piano, for example) is modeled as a dynamical system, which is either in a silent state or in one of several sounding states, where each sounding state is associated with a template in an NMD pattern and activations are coupled across time. Our method proceeds iteratively in

S. Ewert was funded by EPSRC Grant EP/J010375/1.

two steps. First, the state sequence is decoded using dynamic programming (similar to Viterbi decoding), which generates a set of sound-object candidates for each sound source. In a second step, activation parameters for all candidates are updated jointly in a large sparse linear system. By essentially combining ideas from Viterbi training [20] with a subsequent global optimization, our parameter estimation can avoid some degenerate solutions that can result from extreme parameter decoupling.

The paper is organized as follows. Technical details of our method are described in Section 2. We report on some of our experiments in Section 3. Conclusions and prospects on future work are given in Section 4.

2. PROPOSED METHOD

For our method, we assume that a recording for each of K possible sound sources (corresponding to individual keys on a piano or drums in a drumkit) is available as training material. Computing a log-frequency magnitude spectrogram from each recording, we obtain K time-frequency patterns, each consisting of $T>0$ spectral templates. By adding the vector containing only zeros at the beginning of the pattern to model silence for the sound source, we obtain a *pattern dictionary tensor* $\mathcal{D} \in \mathbb{R}_{\geq 0}^{K\times M\times T+1}$, where M is the number of frequency bins. Next, given a log-frequency magnitude spectrogram $V \in \mathbb{R}_{\geq 0}^{M\times N}$ of a recording to be transcribed, we model each entry in V as a mixture of the K sound sources, where each sound source can only be in one of the T sounding states or in the silent state. More precisely, we try to minimize the generalized Kullback-Leibler (KL) divergence $D(V,\widetilde{V}) := \sum_{m,n} d(V(m,n),\widetilde{V}(m,n))$ between the given data and our model, where $d(a,b) := a\cdot\log\left(\frac{a}{b}\right) - a + b$ for $a,b > 0$. The complete model is defined as a sum over K sound models

$$\widetilde{V}(m,n) := \sum_k \widetilde{V}_k(m,n), \tag{1}$$

which are defined as

$$\widetilde{V}_k(m,n) := \mathcal{D}(k,m,\mathcal{S}(k,n))\cdot\mathcal{A}(k,n-\mathcal{S}(k,n)+1), \tag{2}$$

where $\mathcal{S} \in [0:T]^{K\times N}$ is the *state matrix* that encodes in which state each sound source is in each time frame, and $\mathcal{A} \in \mathbb{R}_{\geq 0}^{K\times N}$ is the *activity matrix*, which specifies the intensity (or loudness) of a sound source. For $n \notin [1:N]$, we set $\mathcal{A}(k,n) := \infty$. Note that, the state matrix also controls which activation values are coupled between time frames – a concept similar to NMD. Furthermore, we require for each $k \in [1:K]$ and $n \in [2:N]$ that the transition from state $\mathcal{S}(k,n-1)$ to state $\mathcal{S}(k,n)$ follows a discrete dynamical system, which is illustrated in Fig. 1 and will be described in more detail below. Essentially, the system allows the state to be silent and stay silent, or to transition into the first sounding state. From there, however, it must traverse through at least the next T_M-1 subsequent states, and can only then return to silence, or continue sounding. In this context, T_M is called the *minimum sound length*.

2.1. Estimating the State Matrix

Our objective now is to estimate $\mathcal{S}$ and $\mathcal{A}$. Since our method comprises K independent dynamical systems, each of which is similar to a Markov process as used commonly in probabilistic modeling, one could try to cast this problem as a factorial hidden Markov model. As discussed in detail in [14], even without the temporal coupling between activation parameters that we use in our system, the number of states in the resulting HMM would roughly be T^K such that

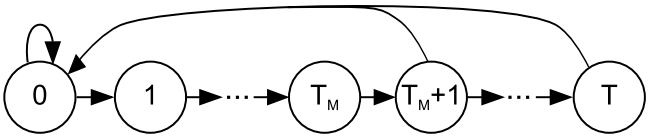


Fig. 1. Dynamical system describing allowed transitions between states for each sound source. State 0 is the silent state while all others are sounding states and are associated with a template in a time-frequency pattern. If the sounding states are entered, at least T_M states have to be traversed.

exact inference would be impossible (a typical setting for our application is T=300 and K=88). A typical approach in such a scenario is to decouple most parameters, which intuitively means to fix most parameters in some way and updating only a few at a time [14,16]. This is typically done either following expectation maximization (EM) or variational Bayes (VB) principles. However, due to the temporal coupling between activations the state space increases even further, such that only a minimal number of parameters can practically be updated jointly, which in some initial experiments often led to useless local minima of $D(V,\widetilde{V})$ and meaningless transcription results.

Therefore, our method approaches the parameter estimation problem from a different angle, resulting in an easy to implement and efficient update process. First, we estimate the best fitting state sequence independently for each of the K sound sources using dynamic programming - this is much simpler than estimating full posterior probabilities as in EM or VB and vaguely resembles the concepts behind Viterbi training [20]. While this simplicity often leads initially to many wrong state estimates, it enables us to create sound-object candidates from the state sequences, whose activations we can optimize jointly across all sound sources. This way, the joint optimization does not only mitigate the problems resulting from the simple first step, it often avoids degenerate local minima typical for updates heavily relying on parameter decoupling.

To describe the first step in more detail, we define for each k a distance matrix $C_k \in \mathbb{R}_{\geq 0}^{T+1\times N}$, which describes the KL-divergence between the $T+1$ templates in the k-th pattern and the N frames in V, as follows:

$$C_k(t,n) := \sum_m d\Big(\mathcal{D}(k,m,t)\cdot\mathcal{A}(k,n-t+1) + \widetilde{V}(m,n) - \widetilde{V}_k(m,n), V(m,n)\Big)$$

for $t \in [0:T]$ and $n \in [1:N]$. Here, $\widetilde{V}(m,n) - \widetilde{V}_k(m,n)$ is the current model excluding the k-th sound source. Using C_k and dynamic programming, we can find the state sequence $\mathcal{S}(k,0),\ldots,\mathcal{S}(k,N)$ minimizing the distance $\sum_n C_k(\mathcal{S}(k,n),n)$ among all sequences valid under the dynamical system shown in Fig. 1. To this end, we recursively define an accumulated distance matrix $D_k \in \mathbb{R}_{\geq 0}^{T+1\times N}$ and a step matrix $E_k \in [0:T]^{T+1\times N}$ as

$$D_k(t,n) := C_k(t,n) + \begin{cases} D_k(t-1,n-1), & t>0 \\ \min\limits_{\tilde{t}\in\Delta}(D_k(\tilde{t},n-1)) & t=0 \end{cases} \tag{3}$$

$$E_k(t,n) := \begin{cases} t-1, & t>0 \\ \operatorname{argmin}_{\tilde{t}\in\Delta}(D_k(\tilde{t},n-1)) & t=0 \end{cases} \tag{4}$$

where $\Delta := \{0, T_M+1, T_M+2, \ldots, T\}$ is the set of states that allow a return to the silence state and $D_k(t,0) := C_k(t,0)$ for

all t. We start by setting $\mathcal{S}(k,N) = \text{argmin}_t D_k(t,N)$ and set $\mathcal{S}(k,n) = E_k(\mathcal{S}(k,n+1),n+1)$ for $n \in [1:N-1]$. Note that the definition in Eqn. 3 only allows state transition that are valid according to our dynamical system. Furthermore, note that we are making a hard decision on the state sequence, which is in contrast to expectation maximization where all states are essentially possible in all time frames and only differ in their probability. Since we will make use of this estimated sequence to update other parameters in our model, this approach is similar to Viterbi training, which is sometimes called *hard-EM* [20]. Finally, note that this process can easily be parallelized over k.

2.2. Estimating the Activations

After updating $\mathcal{S}$, we next need to update the other set of parameters in our model: the activations $\mathcal{A}$. Here, our method has to take into account that one activation value can actually be used across several time frames, which complicates the update process. However, we can exploit two details. First, the coupling is not random but follows an meaningful pattern. Second, by making a hard decision on the state sequences, we do not have to account for a huge posterior probability distribution that describes the likelihood of all sequence combinations, but we have exactly one state for each k in each frame. Combining these aspects, we obtain a straightforward way to update $\mathcal{A}$, following ideas similar to [21].

To this end, we first identify which templates are scaled by the same activation value in Eqn. (2). In particular, assuming that $\mathcal{S}(k,n) = 1$, the model will use the first sounding template in the k-th pattern to represent the n-th time frame by scaling it by $\mathcal{A}(k,n)$. In this case, enforced by the dynamical system, $\mathcal{S}(k,n+1) = 2$, such that the second sounding template will be used in frame $n+1$. However, again $\mathcal{A}(k,n)$ will be used to scale it. This continues, until the state switches back to the silent state, i.e. $\mathcal{S}(k,n+\ell) = 0$ for an $\ell > 0$. This way, we obtain a sound-object starting in frame n and ending in frame $n+\ell-1$, which is linearly scaled in its entirety by the activation $\mathcal{A}(k,n)$. Nothing else will be affected by the value of $\mathcal{A}(k,n)$.

By scanning through $\mathcal{S}(k,1),\ldots,\mathcal{S}(k,N)$, we can easily find all positions where the state sequence enters state 1 as well as the closest, subsequent position where it enters state 0 again. This way, we can create a list of potential sound-object candidates, which we store in a *sound-object candidate list* $L \in \mathbb{N}^{R\times 3}$, where R is the total number of sound-objects identified and each row is of the form $(k,n,n+\ell-1)$, i.e. the three entries encode the sound source index, the beginning frame as well as the end frame. Once we have this list, we can decompose our model into individual sound-objects. To make the next steps more convenient, we store these sound-objects as columns in a matrix $\hat{\mathcal{D}} \in \mathbb{R}^{M\cdot N\times R}$, which is defined as follows

$$\hat{\mathcal{D}}(n\cdot M+m,r) := \mathcal{D}(L(r,1),m,\mathcal{S}(k,n))$$

if $n \in \{L(r,2),\ldots,L(r,3)\}$; we set $\hat{\mathcal{D}}(n\cdot M+m,r) := 0$ otherwise. That means, to create column r, we only keep, from the entire model defined in Eqn. 1, only the templates corresponding to the r-th sound-object, set the rest to zero and stack the columns of the model on top of each other. Similarly, we create a stacked version of the input spectrogram $\hat{V}(n\cdot M+m) := V(m,n)$. Note that while $\hat{\mathcal{D}}$ is potentially R times bigger than the input V, most entries will be zero such that using sparse data structures $\hat{\mathcal{D}}$ will typically be much smaller.

Making the sound-objects accessible like this, the update process for $\mathcal{A}$ is now equivalent to minimizing the generalized Kullback-Leibler divergence $D(\hat{V},\hat{\mathcal{D}}\cdot\hat{\mathcal{A}})$, where $\hat{\mathcal{A}} \in \mathbb{R}^R_{\geq 0}$ are the activations corresponding to the R sound-objects. Since $\hat{V}$, $\hat{\mathcal{D}}$ and $\hat{\mathcal{A}}$ are non-negative matrices, this is a special case of NMF, and we can use one of the available algorithms to solve for $\hat{\mathcal{A}}$. For the sake of simplicity, we follow the method proposed in [22], leading to the following iteratively applied update rule:

$$\hat{\mathcal{A}} \leftarrow \hat{\mathcal{A}} \odot \frac{\hat{\mathcal{D}}^\top \cdot (\frac{\hat{V}}{\hat{\mathcal{D}}\cdot\hat{\mathcal{A}}})}{\hat{\mathcal{D}}^\top \cdot J} \tag{5}$$

where the $\cdot$ operator denotes the usual matrix product, the $\odot$ operator denotes the Hadamard product, $J \in \mathbb{R}^{M\cdot N}$ denotes the vector of ones, and the division is understood point-wise. Once Eqn. 5 has been applied a fixed number of times, the updated entries in $\hat{\mathcal{A}}$ can be copied back into $\mathcal{A}$: $\mathcal{A}(L(r,1),L(r,2)) := \hat{\mathcal{A}}(r)$.

2.3. Practical Considerations

Similar to any reasonably complex model, the update process described in this paper can get stuck in a local minimum of the function to be minimized. Therefore, the parameters should be meaningfully initialized to discourage certain degenerate solutions. In particular, it is desirable to initially create a large number of sound-objects, which has the effect that a large number of activation parameters can be jointly optimized. To this end, we can initialize $\mathcal{S}$ to the silent state, as then each dynamical system responsible for a certain sound source will initially try to explain the energy of other sound sources as well, which leads to additional sound-objects. Furthermore, instead of creating a new list L in each iteration, it can be useful to only amend it with sound-objects newly found in a given iteration. Finally, the activations should not be initialized randomly but with a uniform value A_{min}, which also serves as a threshold to define which sound-objects will finally be used to generate a transcription result, consisting of the list L being plotted in a piano roll representation, see Fig. 2 for an example.

3. EXPERIMENTS

To illustrate the performance of our proposed method, we conducted a series of experiments. In particular, we were interested in how our method would perform in a scenario in which it is reasonable to assume that recordings of individual notes can be provided, for example during a recording studio session or a piano student's exercise session. Since we were not able to find a dataset, which includes both audio recordings of entire piano pieces and single note recordings of a certain length, we decided to create a dataset. In particular, we downloaded 10 MIDI files that are publically available from the University of Minnesota piano-e-competition website[1]. These MIDI files were recorded using a Yamaha Disklavier during an international piano playing competition and therefore closely capture the actual, real-world performance of highly skilled pianists. The pieces were selected to cover a broad range of composers and performers but were otherwise selected randomly, see Table 1 for an overview. To create high quality audio versions from these MIDI files, we employed Native Instruments' Vienna Concert Grand VST plugin, which comprises samples for a Boesendorfer 290 concert grand with an uncompressed size of almost 14 GB. Additionally, we used the plugin to create recordings of single notes for that piano, each 6 seconds long and played in forte (MIDI velocity 100). Overall, while the acoustic data was synthesized, we believe that the choice of the MIDI files in combination with the high quality plugin can be used to obtain some insight into our proposed method.

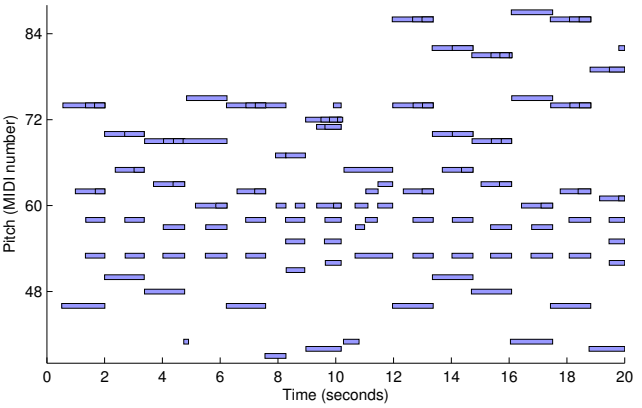
[1] `http://www.piano-e-competition.com`

Composer	Piece	Performer	Pre	Rec	F
Bach	BWV. 851	Colafelice	91%	93%	92%
Beethoven	Op. 10 No. 3	Wang	86%	89%	87%
Chopin	Op. 25 No. 11	Kim	76%	83%	79%
Haydn	HobXVI 52	Mizumoto	86%	87%	86%
Liszt	Polonaise E-Maj	Denisova	90%	92%	91%
Mendelssohn	Op. 54	Sham	94%	84%	89%
Mozart	K284-01	Ozaki	83%	89%	86%
Ravel	Alb. D. Grac.	Teo	86%	87%	86%
Schubert	Op. 142 No. 3	Chon	92%	91%	91%
Stravinsky	Op. 7 No. 4	Lin	84%	96%	89%
Average			**87%**	**89%**	**88%**

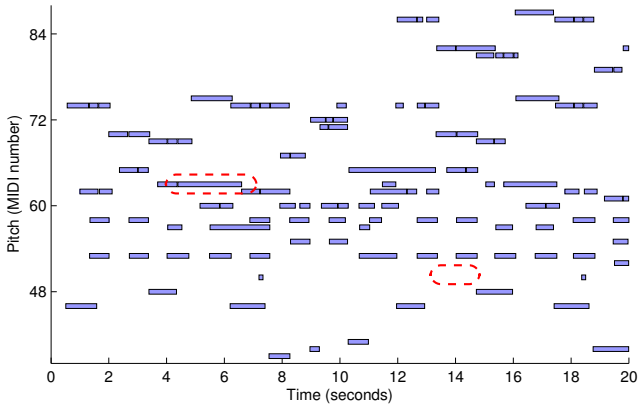
Table 1. Evaluation results: Precision, Recall and F-Measure for detected note onsets for 10 classical piano pieces.

We used the method described above to transcribe the first 30 seconds of each recording and compared it with the underlying MIDI files. Similar to [23, 24], we employed precision (Pre), recall (Rec), and F-measure (F) values as used in the MIREX evaluation campaign for the onset detection task to evaluate how well our method detects note onsets. More precisely, a detected note is considered as correct if there is a note in the corresponding ground truth MIDI file having the same MIDI pitch, with on onset position up to 50ms apart from the detected note. Every ground truth note can only validate up to one detected note. By counting the number of correctly detected notes (TP), incorrectly detected extra notes (FP) and incorrectly missed notes (FN), we can define the precision Pre := TP / (TP + FP), recall Rec := TP / (TP + FN) and the f-measure F := 2 · Pre · Rec / (Pre + Rec). The result for the 10 pieces are given in the last three columns of Table 1.

Some typical errors made by our method are illustrated in Fig. 2, which shows a piano roll representation of the Disklavier MIDI file as well as of the transcription result obtained from the corresponding audio for Schubert's Impromptu Opus 142 No. 3. Comparing Fig. 2a and b, we see that most notes were correctly detected. For our discussion, we highlighted two positions where the transcription failed. In particular, the red marker starting at 4 seconds shows a D♯4 note with an over-estimated length. Note that at the same time an A4 was played, whose length was under-estimated by our method. Here, the underlying reason is that the A4 was played softly, which led to a different energy distribution over the harmonics compared to our single note pattern, which was played in forte and comprises prominent higher partials. This mismatch in the pattern let the parameter estimation to rather prolong the D♯4. A similar problem occurred at the second marker in Fig. 2b. Here, a D3 is played softly as well, and due to the resulting spectral differences compared to our time-frequency pattern, the method does not detect the note event. A similar effect happens around 2 seconds as well. We confirmed these findings experimentally, by replacing the forte time-frequency patterns with patterns played in mezzo forte. In this case, the evaluation measures for the Chopin piece, which are slightly below the others in Table 1, improved considerably as the interpretation by Kim features quite a few softly played notes at the beginning.



(a) Disklavier MIDI



(b) Transcription Result

Fig. 2. Transcription result for Schubert's Impromptu Opus 142 No. 3 performed by Sae-Yoon Chon. The red, dashed markers indicate positions discussed in the text.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a novel method for the transcription of struck string instruments that offer control over the length of the sound they produce, such as the piano. By combining ideas from non-negative factorial hidden Markov models and non-negative matrix deconvolution, our approach models each sound source as dynamical system, which is either in a silent state or in one of a sequence of sounding states, each of which is associated with a time frame in a time-frequency pattern. Our method employs dynamic programming to hard-decode the independent state sequences, which in its simplicity yields results inferior to EM or VB decoding, but at the same time enables a subsequent global optimization over sound-objects, which not only compensates for the simplicity in the first step but helps avoiding local-minima that result from parameter decoupling in EM or VB. Our experiments on classical piano music indicate a high transcription accuracy on the note-onset level of 88% f-measure on average for the proposed method. In the future, we plan to extend some of the ideas behind the parameter estimation process, and to evaluate the method in various real-world and simulated application scenarios [25].

5. REFERENCES

[1] Anssi P. Klapuri and Manuel Davy, Eds., *Signal Processing Methods for Music Transcription*, Springer, New York, 2006.

[2] Samer A. Abdallah and Mark D. Plumbley, "Polyphonic music transcription by non-negative sparse coding of power spectra,"

in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2004, pp. 318–325.

[3] Masataka Goto, "A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication (ISCA Journal)*, vol. 43, no. 4, pp. 311–329, 2004.

[4] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri, "Automatic music transcription: challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[5] Graham E. Poliner and Daniel P.W. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, 2007.

[6] Sebastian Böck and Markus Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, 2012, pp. 121–124.

[7] Benoit Fuentes, Roland Badeau, and Gaël Richard, "Harmonic adaptive latent component analysis of audio and application to music transcription," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 9, pp. 1854–1866, 2013.

[8] Graham Grindlay and Daniel P.W. Ellis, "Transcribing multi-instrument polyphonic music with hierarchical eigeninstruments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1159–1169, 2011.

[9] Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu, "Non-negative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural Computation*, vol. 21, pp. 793–830, 2009.

[10] Nancy Bertin, Roland Badeau, and Emmanuel Vincent, "Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 538–549, 2010.

[11] Emmanouil Benetos, Sebastian Ewert, and Tillman Weyde, "Automatic transcription of pitched and unpitched sounds from polyphonic music," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3107–3111.

[12] Alexey Ozerov, Cédric Févotte, and Maurice Charbit, "Factorial scaled hidden Markov model for polyphonic audio representation and source separation," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2009, pp. 121–124.

[13] Gautham J. Mysore, Paris Smaragdis, and Bhiksha Raj, "Non-negative hidden Markov modeling of audio with application to source separation," in *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, St. Malo, France, 2010, pp. 140–148.

[14] Zoubin Ghahramani and Michael I Jordan, "Factorial hidden markov models," *Machine Learning*, vol. 29, no. 2-3, pp. 245–273, 1997.

[15] Emmanouil Benetos and Simon Dixon, "A temporally-constrained convolutive probabilistic model for pitch detection," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2011, pp. 133–136.

[16] Gautham Mysore and Maneesh Sahani, "Variational inference in non-negative factorial hidden markov models for efficient audio source separation," in *Proceedings of the International Conference on Machine Learning (ICML)*, Edinburgh, Scotland, 2012, pp. 1887–1894.

[17] Paris Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, Grenada, Spain, 2004, pp. 494–499.

[18] Henry Lindsay-Smith, Skot McDonald, and Mark Sandler, "Drumkit transcription via convolutive NMF," in *Proceedings of the International Conference on Digital Audio Effects (DAFX)*, York, UK, 2012.

[19] Mikkel N. Schmidt and Morten Mørup, "Sparse non-negative matrix factor 2-D deconvolution for automatic transcription of polyphonic music," Tech. Rep., Danmarks Tekniske Universitet, 2006.

[20] Valentin I. Spitkovsky, Hiyan Alshawi, Daniel Jurafsky, and Christopher D. Manning, "Viterbi training improves unsupervised dependency parsing," in *Proceedings of the International Conference on Computational Natural Language Learning (CoNLL)*, 2010, pp. 9–17.

[21] Sebastian Ewert, Meinard Müller, and Mark Sandler, "Efficient data adaption for musical source separation methods based on parametric models," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, 2013, pp. 46–50.

[22] Daniel D. Lee and H. Sebastian Seung, "Algorithms for non-negative matrix factorization," in *Proceedings of the Neural Information Processing Systems (NIPS)*, Denver, CO, USA, 2000, pp. 556–562.

[23] Emmanouil Benetos and Simon Dixon, "Polyphonic music transcription using note onset and offset detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 37–40.

[24] Hirokazu Kameoka, Takuya Nishimoto, and Shigeki Sagayama, "A multipitch analyzer based on harmonic temporal structured clustering," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 982–994, 2007.

[25] Matthias Mauch and Sebastian Ewert, "The audio degradation toolbox and its application to robustness evaluation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, Brazil, 2013, pp. 83–88.