2015 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2015, BOSTON, USA

# PIANO MUSIC TRANSCRIPTION WITH FAST CONVOLUTIONAL SPARSE CODING

*Andrea Cogliati, Zhiyao Duan*

University of Rochester
Rochester, NY USA
{andrea.cogliati,zhiyao.duan}@rochester.edu

*Brendt Wohlberg*

Los Alamos National Laboratory
Los Alamos, NM USA
brendt@lanl.gov

## ABSTRACT

Automatic music transcription (AMT) is the process of converting an acoustic musical signal into a symbolic musical representation, such as a MIDI file, which contains the pitches, the onsets and offsets of the notes and, possibly, their dynamics and sources (i.e., instruments). Most existing algorithms for AMT operate in the frequency domain, which introduces the well known time/frequency resolution trade-off of the Short Time Fourier Transform and its variants. In this paper, we propose a time-domain transcription algorithm based on an efficient convolutional sparse coding algorithm in an instrument-specific scenario, i.e., the dictionary is trained and tested on the same piano. The proposed method outperforms a current state-of-the-art AMT method by over 26% in F-measure, achieving a median F-measure of 93.6%, and drastically increases both time and frequency resolutions, especially for the lowest octaves of the piano keyboard.

***Index Terms*—** Automatic Music Transcription, Convolutional Sparse Coding, Shift Invariant, Sparse Representation

## 1. INTRODUCTION

Automatic Music Transcription (AMT) is the process of inferring a symbolic music representation from a music audio file. The output of AMT can be a full musical score or an intermediate representation, such as a MIDI file, which includes note pitches, onsets, offsets and, possibly, dynamics and instruments playing the notes. The complete AMT problem can be divided into several subtasks, not necessarily in this order: multi-pitch estimation, onset/offset detection, loudness estimation, source recognition, note tracking, beat and meter detection, and rhythm identification. Most research thus far has focused on the multi-pitch estimation and onset detection stages [1].

Many music transcription methods attempt to identify pitches in each time frame, and then form notes in a post-processing stage [1]. This approach, however, does not model the temporal evolution of notes. We can call this approach *frame-based* transcription. Most spectrogram decomposition-based approaches fall into this category. Non-negative matrix factorization (NMF) is a method for factorizing a large non-negative matrix into the product of two, low-rank non-negative matrices [2][3]. NMF has been applied to source separation and AMT [4]. An alternate formulation of NMF named Probabilistic Latent Component Analysis (PLCA) was proposed by Smaragdis et al. in 2006 [5]. PLCA is numerically equivalent to NMF but its formulation provides a framework that is easier to generalize and extend [5]. NMF is computationally inexpensive, jointly estimates multiple pitches at the same time and provides a salience of each estimated pitch with its activation weight. NMF can be applied in an unsupervised way to analyze musical signals, where the dictionary elements are learned during the factorization of the audio spectrogram. However, the learned dictionary elements may represent only a part of a note's spectrum, or represent a mixture of multiple notes, and they can also be sensitive to the learning order or factorization rank. Clustering or group sparsity [6] are often employed to improve the correspondence between templates and notes. For AMT, supervised NMF is generally preferred, where a dictionary of templates corresponding to each note is pre-learned, typically from isolated notes. Each template essentially corresponds to the long-time average spectrum of a note, thus ignoring the temporal evolution of the spectral content. To transcribe music played by a different instrument, a dictionary adaptation process can be employed [7]. To obtain note-level transcription results, a post-processing step, such as a median filter or HMM, is required to connect frame-level pitch estimates into notes [8].

Piano notes are characterized by significant temporal evolutions, in both the waveform and the spectral content. In particular, different partials decay at different rates, i.e., higher frequency partials decay faster than lower frequency ones [9][10]. However, only a few methods, all of which operate in the frequency domain, model temporal evolution of notes. A tensor can be used to represent multiple vectors evolving in time, e.g., a dictionary of spectrograms. Non-negative tensor factorization (NTF), an extension of NMF, has been applied to source separation and AMT [11][12][13]. Grindlay and Ellis proposed a generalization to PLCA to account for the temporal evolution of each note [14]. A variant of NMF called Non-negative Factorial Hidden Markov Model (N-FHMM) was introduced to learn multiple spectral templates for each note and a Markov chain describing the temporal evolution between them [15]; Ewert et al. have recently proposed a dynamic programming variation of N-FHMM to reduce its high computational cost [16]. Non-negative Matrix Deconvolution (NMD) as introduced in [17], which concatenates several spectral frames into an entire time-frequency template, is capable of modeling the temporal evolution of non-stationary sounds. In [18], we proposed a two-stage approach for piano transcription which models the temporal evolution of piano notes in the frequency domain. A dictionary of spectrograms of notes was pre-learned from isolated notes, and was then used to decompose each inter-onset interval of the music audio. The performance of this approach is limited by the onset detection accuracy. Regardless of the technique used, all the above-mentioned methods attempt to identify entire notes at once. We call these methods *note-based* transcription, as opposed to frame-based transcription.

Spectrogram factorization methods generally suffer from the time/frequency resolution trade-off introduced by the Short Time Fourier Transform and its variants. The transcription of low-frequency notes often requires a high frequency resolution. But to achieve such a high frequency resolution, the time resolution

978-1-4673-7454-5/15/$31.00 ©2015 IEEE

would be sacrificed. In addition, the phase information is often discarded in spectrogram decomposition methods. This may lead to source number ambiguity and octave ambiguity [19]. The time domain representation, on the other hand, does not have these problems, since it is not subject to the time-frequency resolution tradeoff and also contains the phase information. It is noted that the best performing single-pitch estimation methods work in the time domain [20]. However, for polyphonic transcription, there have been very few methods in the time domain. Plumbley et al. proposed and compared two approaches for sparse decomposition of polyphonic music, one in the time domain and the other in the frequency domain [21]. While they suggest that both approaches can be applied to AMT, to the best of our knowledge no further research was published for the time-domain approach to AMT.

In this paper we present a supervised approach to AMT based on Convolutional Sparse Coding (CSC) of a time domain signal. The proposed method uses an instrument-specific, pre-learned dictionary followed by an efficient convolutional basis pursuit denoising algorithm to find a sparse representation of the audio signal. Finally, note onsets are estimated from the coefficient maps determined in the previous step by peak picking. The advantages of the proposed method are: higher performance with respect to state-of-the-art transcription algorithms, improved temporal resolution, improved resolution at lower-frequencies and reduced octave errors.

In the spirit of reproducible research, the code and dataset used in this paper are available at `http://www.ece.rochester.edu/~acogliat/` under the Code & Dataset Repository section.

## 2. BACKGROUND

Sparse representations have been widely applied to signal and image processing problems. Sparse coding, the inverse problem of sparse representation of a particular signal, has been approached in several ways. One of the most widely used is Basis Pursuit DeNoising (BPDN) [22]:

$$\arg\min_{x} \frac{1}{2}\|D\boldsymbol{x} - \boldsymbol{s}\|_2^2 + \lambda\|\boldsymbol{x}\|_1, \tag{1}$$

where $\boldsymbol{s}$ is a signal to approximate, $D$ is a dictionary matrix, $\boldsymbol{x}$ is the sparse representation, and $\lambda$ is a regularization parameter. Convolutional Sparse Coding (CSC), also called shift-invariant sparse representation, extends the idea of sparse representation by using convolution instead of multiplication. Replacing the multiplication operator with convolution in (1) we obtain Convolutional Basis Pursuit DeNoising (CBPDN) [23]:

$$\arg\min_{\{x_m\}} \frac{1}{2}\left\|\sum_m \boldsymbol{d}_m * \boldsymbol{x}_m - \boldsymbol{s}\right\|_2^2 + \lambda \sum_m \|\boldsymbol{x}_m\|_1, \tag{2}$$

where $\{\boldsymbol{d}_m\}$ is a set of dictionary elements, also called filters, and $\{\boldsymbol{x}_m\}$ is a set of activations, also called coefficient maps.

CSC has been applied in the audio domain to source separation [24], music transcription [21] and audio classification [25]. However, its adoption has been limited by its computational complexity in favor of faster factorization techniques like NMF. Recent research on efficient algorithms for CSC [26] and increased availability of computational power have renewed the interest in CSC for audio applications [27].

The algorithm described in [26] is based on the Alternating Direction Method of Multipliers (ADMM) for convex optimization. The most computationally expensive subproblem handles the convolution operation by transforming to the frequency domain and exploiting the resulting linear system structure to obtain a very efficient solution. Since the overall complexity of the algorithm is dominated by the cost of FFTs, the cost of the entire algorithm is $O(MN \log N)$, where $M$ is the number of atoms in the dictionary and $N$ is the size of the signal.

## 3. RELATION TO PRIOR WORK

The use of CSC on time-domain signals for AMT has been proposed as early as 2005 [21][24], but initial research was dropped in favor of spectrogram-based methods, which have much lower computational cost.

Plumbley et al. [21] proposed a shift-invariant generative model in the time domain based on a sum of scaled and shifted versions of some underlying functions $\boldsymbol{a}_m$, called atoms. Given a discrete audio signal $s[t]$, they select any $I$ consecutive samples into a vector $s_i$ which is approximated as

$$\boldsymbol{s}_i = \sum_{j=1}^{J}\sum_{m=1}^{M} a_{ijm}x_{im} + e_i,\ 1 \le i \le I, \tag{3}$$

where $a_{ijm}$ is a tensor in which the $m$-th slice is a matrix of shifted versions of the original function $\boldsymbol{a}_m$, with $j$ being the time shift, $x_{jm}$ are the activations of atom $m$, and $e_i$ is additive noise. This model is effectively a constrained variant of (2), with the $\ell_1$ penalty term replaced with an upper bound on the $\ell_0$ norm. Plumbley et al. proposed an unsupervised dictionary learning algorithm for this model, in which the dictionary atoms as well as the activation coefficients are learned in a subset selection process to reduce the solution space. They applied this approach to AMT and discovered that the learned templates generally reflected the individual notes present in the piece. They suggest applying this method to AMT in an unsupervised way. However, similarly to all unsupervised methods, this approach suffers from several issues. First, each learned template must be analyzed and labeled according to its pitch (assuming that each element contains a single note and not, for instance, two or more notes). Second, a critical parameter is the number $M$ of dictionary entries, which should be equal to the number of unique notes present in the piece. Finally, some notes in the audio signal can be reconstructed using multiple templates, which might lead to undetected activations. In addition to these issues, the authors only demonstrated the idea using one example piece; no systematic evaluations and comparisons were conducted. It should also be noted that this approach uses short templates (128 ms) that do not capture the temporal evolution of piano notes, and it also segments the audio signal into frames that are analyzed independently.

## 4. PROPOSED METHOD

The proposed method is based on the efficient convolutional sparse coding algorithm presented in [26]. A monaural, polyphonic audio recording of a piano piece $\boldsymbol{s}[t]$ is approximated by a sum of dictionary elements $\boldsymbol{d}_m[t]$ representing each individual note of a piano, convolved with activation vectors $\boldsymbol{x}_m[t]$:

$$\boldsymbol{s}[t] \simeq \sum_m \boldsymbol{d}_m[t] * \boldsymbol{x}_m[t]. \tag{4}$$

A non-zero value at index $t$ of an activation vector $\boldsymbol{x}_m[t]$ represents the activation of note $m$ at sample $t$.

The dictionary elements are pre-learned in a supervised manner by sampling each individual note of a piano at a certain dynamic level, e.g., *mf*, see Fig. 1 for an example.

The activation vectors are estimated from the audio signal by CBPDN as in (2). Note onsets are then derived from the activation vectors by sparse peak picking, i.e., multiple activations of the same note are not allowed inside a sparsity window of 50 ms; in case of multiple activations, the earliest one is chosen as the right one. The resulting peaks are also filtered for magnitude in order to keep only the peaks which are higher than 10% of the highest peak in the entire activation matrix. Fig. 2 shows the piano roll, the waveform, the raw activation vectors and the estimated note onsets for a simple melody of 5 notes (4 unique notes). Note that no non-negativity constraints have been applied, so CBPDN can produce negative values in the activation vector, even though the strongest peaks are generally positive.

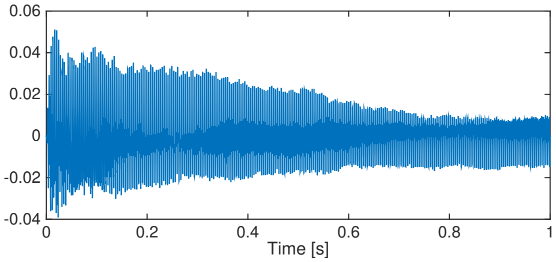
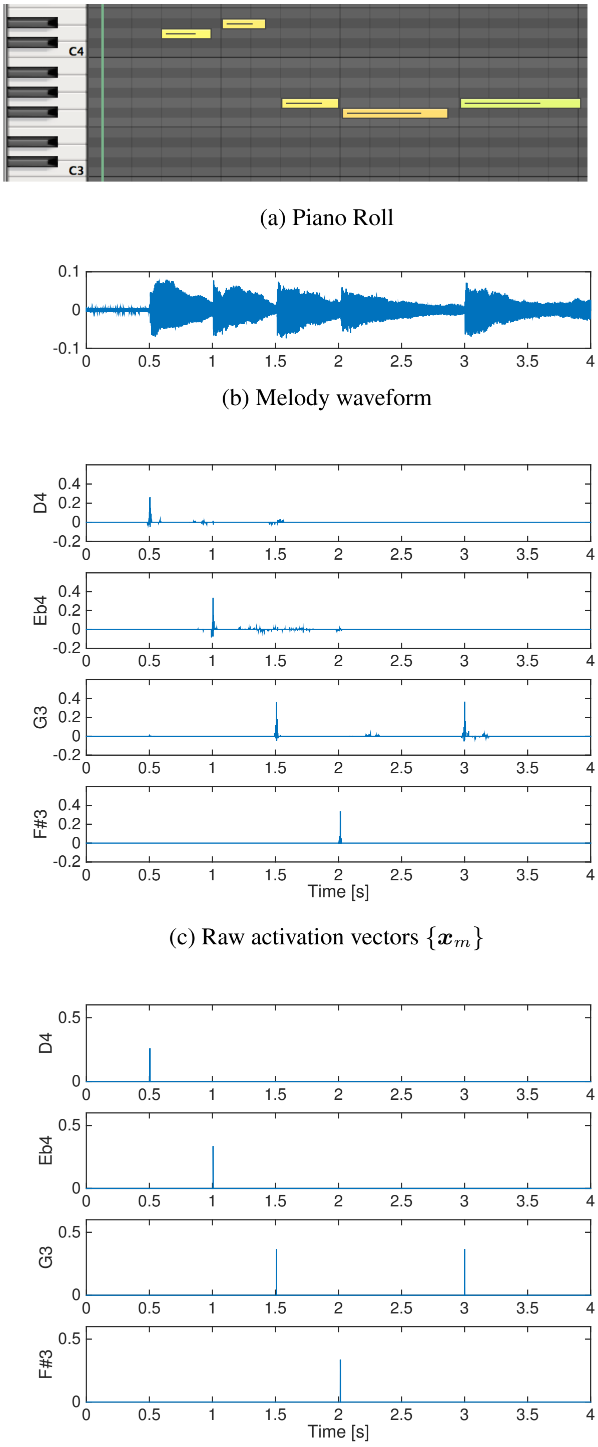


**Fig. 1**. Dictionary element for C4 from Steinway Concert Grand Piano from the Garritan Personal Orchestra played at MIDI velocity 100.



(a) Piano Roll

(b) Melody waveform

(c) Raw activation vectors $\{\boldsymbol{x}_m\}$

(d) Note onsets after sparse peak picking

**Fig. 2**. Piano roll, waveform, raw activation vectors and note onsets for a simple melody of 5 notes, with 4 unique notes.

## 5. EXPERIMENT

In the first experiment, we used the 30 MIDI files of the music pieces in the ENSTDkCl collection of the MAPS dataset [28]. All the pieces have been re-rendered from the MIDI files using a digital audio workstation (Logic Pro 9) with a virtual piano plug-in (Steinway Concert Grand Piano from the Garritan Personal Orchestra); no reverb was used at any stage. The MIDI files represent realistic performances of the pieces and contain a wide range of dynamics; i.e., the MIDI files have been created starting from MIDI files available on the Internet, which have been manually edited to adjust note locations, durations and dynamics to achieve more human sounding performances. The dictionary elements were learned form the same virtual piano but only at a fixed dynamic level (i.e., MIDI velocity of 100). To reduce the computational cost and the memory footprint of the proposed algorithm we downsampled all the audio recordings to 11,025 Hz and transcribed only the initial 30 s of each piece. We compared the proposed method with a state-of-the-art algorithm by Benetos [8], the best performer in MIREX 2013[1], which uses PLCA in the frequency domain. Benetos's method uses a constant-Q transform (CQT) with a spectral resolution of 120 bins/octave and an overlap of temporal atoms of 80% as a time-frequency representation. Each template is essentially the long term average spectrum of each note in log-frequency. Note templates generated from the sampled instruments are also pre-shifted in frequency to account for vibrato, which is not an issue for piano music. We used the author's original implementation submitted to MIREX 2013. The dictionaries for both methods have been generated from the same set of individual notes rendered as before; the individual notes were 1 s long

[1] `http://www.music-ir.org/mirex/wiki/2013:MIREX2013_Results`

| **Parameter** | **Value** |
|---|---|
| Sample rate | 11,025 Hz |
| $\lambda$ | 0.05 |
| Number of iterations | 500 |
| Initial $\rho$ | $100\lambda + 1$ |
| Peak picking threshold | 10% of maximum |
| Peak picking sparsity window | 50 ms |

**Table 1**. Parameters used in the experiment.

and were played at MIDI velocity 100. For the proposed method we used the parameters listed in Table 1. The regularization parameter $\lambda$ has been empirically tuned to make the $\ell_2$-norm and $\ell_1$-norm in (2) of the same order of magnitude.

Fig. 3 shows a comparison of the F-measures for Benetos and the proposed method. The F-measure is calculated at the note level, using the onset only method with a tolerance of ±50 ms. The proposed method shows a dramatic improvement over the frequency domain method, achieving a median F-measure of 93.6% versus 67.5% of Benetos. Since both methods learn dictionaries from the same training notes, the performance difference shows that the time-domain CSC method provides a richer model to recognize notes. The reasons include better time resolution and better frequency resolution at low frequencies, as illustrated in the following experiments. There were only two pieces for which the proposed algorithm achieved an accuracy lower than 80%, and they were characterized by a loud melody played over an accompaniment of very fast, short and soft notes. The notes in the accompaniment are very different from the templates in the dictionary, and the difference in loudness is probably greater than the threshold limit we chose. Using multiple dictionary templates for the same note played at different dynamics might improve the results for these cases. It should be noted that, except for these two pieces, even though the dictionary contains templates of a fixed length and a single dynamic, the proposed algorithm is capable of generalizing to different note lengths and dynamics.

Although the comparison with Benetos's method is fair, in the sense that both algorithms were trained on the same individual sampled notes, it should be noted that Benetos's method is also capable of generalizing to different pianos, as demonstrated by the results in the MIREX competition, in which the transcription tests are performed on pieces played on random instruments.

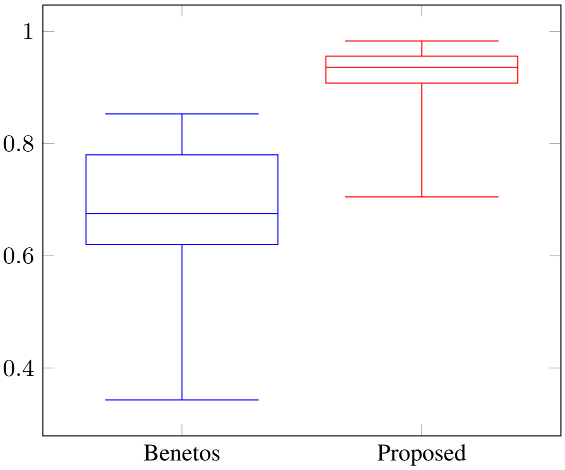


**Fig. 3**. Distribution of the F-measure for the 30 pieces in the ENST-DkCl collection of MAPS. Each box contains 30 data points.

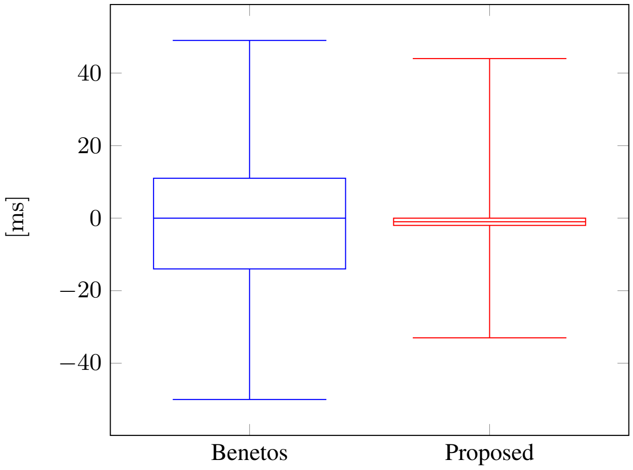


**Fig. 4**. Onset difference from the ground truth for true positves. Positive values indicate late estimated onsets.

Fig. 4 shows the distribution of the onset difference of the estimated notes from the ground truth, calculated for the true positives: a positive value indicates that the estimated onset is later than the ground truth. The proposed method is capable of producing sample-precise activations in most circumstances.

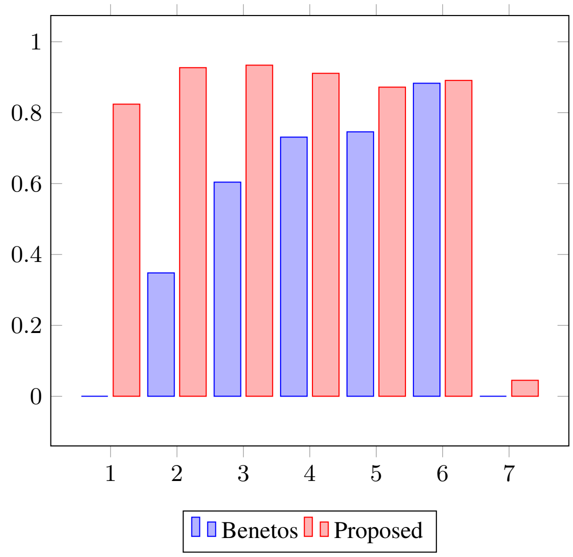


**Fig. 5**. Average F-measure per octave

Fig. 5 compares the average F-measure achieved by the two methods along the different octaves of a piano keyboard (the first octave is from A0 to B1, the second one from C2 to B2 and so on). The distribution of the notes in the ground truth per octave is shown in Table 2. The figure clearly shows that the results of the frequency-domain method are dependent on the fundamental frequencies of the notes; the results are very poor for the first two octaves, and increase

| Octave | Notes | # of notes |
|---|---|---|
| 1 | A0-B1 | 74 |
| 2 | C2-B2 | 497 |
| 3 | C3-B3 | 1,822 |
| 4 | C4-B4 | 2,568 |
| 5 | C5-B5 | 2,035 |
| 6 | C6-B6 | 302 |
| 7 | C7-C8 | 57 |

**Table 2**. Notes in the ground truth per octave.

monotonically for higher octaves. The proposed method shows a more balanced distribution. The poor performance for the last octave might be due to the low sample rate used for the experiment. The sample rate was limited to 11,025 Hz to reduce the computational cost and the memory footprint of the proposed method. The last octave ranges from C7, with a fundamental frequency of 2,093 Hz, to C8, with a fundamental frequency of 4,186 Hz, so only one or two partials will be present in the signal.

To investigate octave errors, Table 3 shows the comparison of the F-measure with the Chroma F-measure, i.e., all F0s are mapped to a single octave before evaluating. There is a slight improvement over the average F-measure in both methods, but the improvement is slightly less pronounced for the proposed method, suggesting a lower incidence of octave errors.

| | Benetos | Proposed |
|---|---|---|
| F-measure | 0.672 | 0.914 |
| Chroma F-measure | 0.691 | 0.930 |
| Difference | 0.019 | 0.016 |

**Table 3**. Average octave errors.

In the second experiment we tested the proposed method in a more realistic scenario using a real acoustic piano. We used an iPhone to record a baby grand piano in a recording studio. We recorded all the 88 individual notes of the piano and tested the algorithm on a simple piano piece, i.e., Bach's Minuet BWV 114. Each note was played for 1 s at a constant dynamic level of mf. The smartphone was placed on the right side of the piano case, roughly 10" above the soundboard. The recording studio has absorbers on the walls to reduce reverb. The ground truth was established by playing the same piece on a MIDI keyboard then manually aligning the note onsets of the recorded MIDI file with the audio recording. For the algorithm we used the same parameters listed in Table 1. The results of the transcription are shown in Table 4. The proposed method achieves nearly perfect results on this simple piece, showing that the algorithm is not limited to synthesized sounds and that different instances of the same note on a real piano are consistent at the signal level.

| | Benetos | Proposed |
|---|---|---|
| Precision | 0.799 | 0.995 |
| Recall | 0.564 | 0.995 |
| F-measure | 0.661 | 0.995 |

**Table 4**. Results on a single piece on a real piano.

## 6. DISCUSSION AND CONCLUSION

In this paper we presented an automatic music transcription algorithm based on convolutional sparse coding of a time-domain musical signal. The proposed algorithm outperforms a state-of-the-art algorithm and shows a sample-precise temporal resolution, increased resolution at lower-frequencies and reduced octave errors. The algorithm shows generalization capabilities to notes of different length and loudness, and an initial experiment shows that it is applicable to real acoustics recordings and not limited to synthesized sounds.

The major limitations of the proposed algorithm are the computational cost and the memory footprint of the sparse coding algorithm. The transcription of 30 s of audio with the parameters shown in Table 1 takes almost 30 minutes on average and uses 2.6 GB of memory for the working variables, when using double precision. The memory footprint grows linearly with the length of the signal and with the number of elements in the dictionary. Another limitation of the algorithm is the poor generalization capabilities of the model, as the time domain representation of the audio is much more specific than, for instance, its linear magnitude spectrogram. Preliminary experiments on audio recorded on a different piano show a dramatic reduction in the performance. Finally, reverb, if not present in the dictionary elements, makes the activation signal less sparse and more difficult to analyze. Thus, future work is needed to adapt the dictionary to a different instrument and to reverberant situations.

## 7. ACKNOWLEDGMENTS

The authors wish to thank Dr. Emmanouil Benetos, Research Fellow at the Centre for Digital Music of Queen Mary University of London, for kindly providing the code of his transcription system to compare the performance.

## 8. REFERENCES

[1] 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.

[2] Daniel D. Lee and H. Sebastian Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–91, 1999.

[3] Daniel D. Lee and H. Sebastian Seung, "Algorithms for non-negative matrix factorization," in *Proc. Advances in Neural Information Processing Systems*, 2001, pp. 556–562.

[4] Paris Smaragdis and Judith C. Brown, "Non-negative matrix factorization for polyphonic music transcription," 2003.

[5] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka, "A probabilistic latent variable model for acoustic modeling," *In Workshop on Advances in Models for Acoustic Processing at NIPS*, 2006.

[6] Ken O'Hanlon and Mark D. Plumbley, "Polyphonic piano transcription using non-negative matrix factorisation with group sparsity," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3112–3116.

[7] Emmanouil Benetos, Roland Badeau, Tillman Weyde, and Gaël Richard, "Template adaptation for improving automatic music transcription," in *Proc. of ISMIR 2014*, 2014, p. 6.

[8] Emmanouil Benetos and Simon Dixon, “A shift-invariant latent variable model for automatic music transcription,” *Computer Music Journal*, vol. 36, no. 4, pp. 81–94, 2012.

[9] Murray Campbell and Clive Greated, *The Musician’s Guide to Acoustics*, Oxford University Press, 1994.

[10] Tian Cheng, Simon Dixon, and Matthias Mauch, “Modelling the decay of piano sounds,” in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, 2015, pp. 594–598.

[11] Tuomas Virtanen, “Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, 2007.

[12] Joonas Nikunen, Tuomas Virtanen, and Miikka Vilermo, “Multichannel audio upmixing based on non-negative tensor factorization representation,” in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 33–36.

[13] Tom Barker and Tuomas Virtanen, “Non-negative tensor factorisation of modulation spectrograms for monaural sound source separation,” in *Proc. Interspeech*, pp. 827–831.

[14] Graham C. Grindlay and Dan 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.

[15] Gautham J. Mysore, Paris Smaragdis, and Bhiksha Raj, “Non-negative hidden markov modeling of audio with application to source separation,” in *Latent Variable Analysis and Signal Separation*, pp. 140–148. Springer, 2010.

[16] Sebastian Ewert, Mark D. Plumbley, and Mark Sandler, “A dynamic programming variant of non-negative matrix deconvolution for the transcription of struck string instruments,” in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, 2015, pp. 569–573.

[17] Paris Smaragdis, “Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs,” in *Independent Component Analysis and Blind Signal Separation*, pp. 494–499. Springer, 2004.

[18] Andrea Cogliati and Zhyiao Duan, “Piano music transcription modeling note temporal evolution,” in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, 2015, pp. 429–433.

[19] Cheng-Te Lee, Yi-Hsuan Yang, and Homer H Chen, “Multipitch estimation of piano music by exemplar-based sparse representation,” *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 608–618, 2012.

[20] Alain De Cheveigné and Hideki Kawahara, “YIN, a fundamental frequency estimator for speech and music,” *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[21] Mark D. Plumbley, Samer A. Abdallah, Thomas Blumensath, and Michael E. Davies, “Sparse representations of polyphonic music,” *Signal Processing*, vol. 86, no. 3, pp. 417–431, 2006.

[22] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[23] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Robert Fergus, “Deconvolutional networks,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2528–2535.

[24] Thomas Blumensath and Mike Davies, “Sparse and shift-invariant representations of music,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 1, pp. 50–57, 2006.

[25] Roger Grosse, Rajat Raina, Helen Kwong, and Andrew Y. Ng, “Shift-invariance sparse coding for audio classification,” *arXiv preprint arXiv:1206.5241*, 2012.

[26] Brendt Wohlberg, “Efficient convolutional sparse coding,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 7173–7177.

[27] Ping-Keng Jao, Yi-Hsuan Yang, and Brendt Wohlberg, “Informed monaural source separation of music based on convolutional sparse coding,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, Apr. 2015, pp. 236–240.

[28] Valentin Emiya, Roland Badeau, and Bertrand David, “Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.