# IMPROVING PIANO NOTE TRACKING BY HMM SMOOTHING

*Tian Cheng, Simon Dixon, Matthias Mauch*

Centre for Digital Music, Queen Mary University of London, London, United Kingdom

## ABSTRACT

In this paper we improve piano note tracking using a Hidden Markov Model (HMM). We first transcribe piano music based on a non-negative matrix factorisation (NMF) method. For each note four templates are trained to represent the different stages of piano sounds: silence, attack, decay and release. Then a four-state HMM is employed to track notes on the gains of each pitch. We increase the likelihood of staying in silence for low pitches and set a minimum duration to reduce short false-positive notes. For quickly repeated notes, we allow the note state to transition from decay directly back to attack. The experiments tested on 30 piano pieces from the MAPS dataset shows promising results for both frame-wise and note-wise transcription.

***Index Terms***— piano note tracking, Hidden Markov Model

## 1. INTRODUCTION

Automatic music transcription (AMT) is the process of converting a musical signal into a musical score. The majority of recent systems focus on music transcription at two levels: frame-level and note-level. Frame-level transcription estimates the set of pitches sounding at each time frame (multi-F0 estimation), providing a time-pitch representation, while note-level transcription integrates this information over time (note tracking), giving a set of note events described by pitch, onset and offset time. Recent AMT systems are usually based on non-negative matrix factorisation (NMF) method, which decomposes the spectrogram into spectral bases and the gain matrix (a mid-level transcription) with non-negative constraints [1–7]. Multi-F0 estimation can be achieved by simply thresholding on the gain matrix.

Note tracking systems are usually built on top of multi-F0 estimation algorithms. In these systems, notes are detected in the time-pitch representation by converting pitches found in consecutive frames into notes and removing those that fail to reach a duration threshold (minimum duration pruning) [8]. Ryynänen and Klapuri model note events using a Hidden Markov Model (HMM) based on acoustic features, considering both pitch and onset information [9]. In [10, 11] piano music piece is divided into segments by detected onsets, and pitches are estimated for each segment. Then an HMM is applied on the segment-wise transcription results to track note events. An explicit duration HMM is applied with a convolutive probabilistic framework to model the temporal evolution and duration of the sound states for multi-instrument music [12]. Duan and Temperley build a note-level music transcription system by randomly sampling notes detected by a multi-F0 estimation system [13]. Berg-Kirkpatrick et al. propose a note detection system for piano music, which models the note activation by velocity, temporal envelope and duration [14]. Mauch et al. present an HMM note transcription software for sung or played melody [15]. Recent reviews of AMT systems can be found in [8, 16].

In this paper, we focus on piano note tracking. First we obtain the gain matrix using NMF with fixed templates. For each pitch, four templates are trained to model the silence, attack, decay and release stages of piano sounds. In this way, the percussive onsets, the remaining energy after offsets, as well as any silence between notes can be modelled. After obtaining the gain matrix using the pre-trained templates, we track notes for each pitch using an HMM. The states of the HMM correspond to the stages of piano sounds, which are constrained to occur in a fixed chronological sequence. The most likely state sequence is estimated using the Viterbi algorithm. We increase the likelihood of staying in silence for low pitches and set a minimum duration to reduce short false-positive notes. Fast repeated notes are difficult to detect because the HMM attaches a high cost to starting a new note. To account for notes in quick succession, we modify the model to allow transitions from decay to attack. The results show that the note-wise performance is improved by up to 3 percentage points with an F-measure of around 75% by dealing with this repeated notes, while no obvious difference is observed on frame-wise results.

For note-level transcription of piano music, repeated notes are a typical cause of false negatives. Emiya et al. propose to deal with this problem by considering the note loudness [10], but generally speaking, this problem is rarely addressed in previous work. In this paper, we investigate in detail the temporal dynamics of HMM states and set bidirectional transition possibilities between attack and decay to detect fast repeated notes, arriving at parameter settings which could easily be adopted by other systems.

Tian Cheng is supported by a China Scholarship Council/ Queen Mary Joint PhD Scholarship. Matthias Mauch is funded by a Royal Academy of Engineering Research Fellowship.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

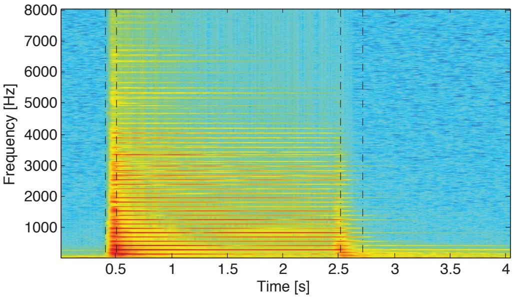

**Fig. 1**. Different stages of a note. The stages divided by the dash lines are silence, attack, decay, release and silence.

## 2. PROPOSED SYSTEM

We first obtain a mid-level transcription representation based on NMF with pre-trained templates, then track notes for each pitch using a four-state HMM.

### 2.1. NMF-based template training and transcription

We provide a transcription to a mid-level representation using NMF. In the NMF framework, the spectrogram $V \in \mathbb{R}_+^{F\times T}$ is factorised into two non-negative matrices:

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

where $W \in \mathbb{R}_+^{F\times R}$ represents the spectral bases, and $H \in \mathbb{R}_+^{R\times T}$ denotes the gain matrix, which is the mid-level representation of transcription results [17]. Firstly we train the templates on isolated piano notes. In order to deal with the time-varying spectrogram, we employ multiple templates to describe the spectrogram of different stages of a piano note. We divide a note into four stages: silence, attack, decay and release. We first initialise the gains according to the onset and offset of the note. Because of the overlap between frames, there are several frames around the onset containing the transient. All these frames are labelled as attack stage. The decay lasts from the onset until the offset. After the keys are released, the strings are still vibrating for a while. We simply set the release to begin from the offset and lasts twice the duration of the attack; and the rest is silence, as shown in Figure 1. We update the spectral bases and gains using NMF with beta-divergence to train templates:

$$W \leftarrow W.\frac{[(WH)^{\cdot(\beta-2)}.V]H^T}{[WH]^{\cdot(\beta-1)}H^T}, \tag{2}$$

$$H \leftarrow H.\frac{W^T[(WH)^{\cdot(\beta-2)}.V]}{W^T[WH]^{\cdot(\beta-1)}}. \tag{3}$$

Then we normalise the sum of each template to be 1.

For the music pieces, we only update the gains using the pre-trained templates, as shown in Figure 2. We found that it was beneficial to initialise the gain matrix with the spectral amplitudes at the corresponding fundamental frequencies.

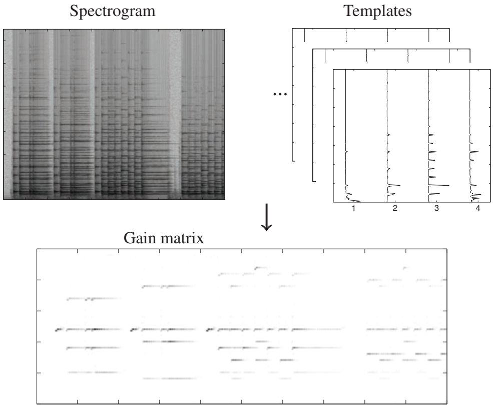

**Fig. 2**. Spectrogram factorisation using pre-trained templates.

### 2.2. HMM-based note tracking

For each pitch, we track notes using an HMM based on the gains obtained above. The states $S_i$ ($i \in \{1, 2, 3, 4\}$) of the HMM correspond to the four templates for each pitch, which represent silence, attack, decay and release, respectively. The normalised gains corresponding to each template also indicate the likelihood of being in each of the states.

The note tracking process is shown in Figure 3. Before tracking each pitch, we apply a median filter to each row of the gain matrix and normalise the gains by dividing by the global maximum. We add a small value to the gains of silent states for noise suppression. This value helps to delete false positives with small energies. When the gains for non-silent states are below the value, then the silent state becomes the most likely state for that frame. When the note gains are much larger than the value, their states are not influenced by the small adjustment to the silent state's gain. For each individual pitch, the gains are normalised by dividing by their sum for each frame, in order to give the observation likelihood of each state.

A left-to-right HMM is used to constrain the transitions between states to follow a fixed chronological sequence, so that each note starts from silence to attack, then to decay and release. After that, the note returns to the silent state again. The transition matrix can be written as follows:

$$T = \begin{bmatrix} T_1 & 1-T_1 & 0 & 0 \\ 0 & T_2 & 1-T_2 & 0 \\ 0 & 0 & T_3 & 1-T_3 \\ 1-T_4 & 0 & 0 & T_4 \end{bmatrix}, \tag{4}$$

where $T_{i,j}$ refers to the transition probability from $S_i$ to $S_j$, $T_{i,j} = P(q_{t+1} = S_j | q_t = S_i)$. The diagonal values of the

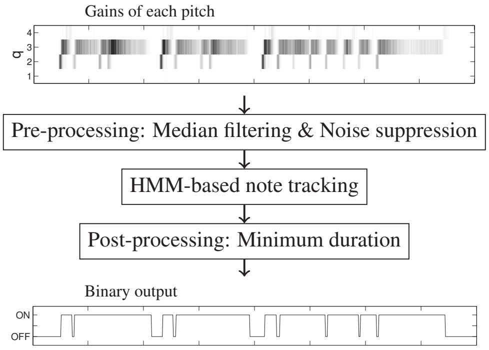
Gains of each pitch
q
4
3
2
1
Pre-processing: Median filtering & Noise suppression
HMM-based note tracking
Post-processing: Minimum duration
Binary output
ON
OFF

**Fig. 3**. Note tracking for each pitch.

transition matrix obtained from the isolated note training data are $T_{diag} = [0.99,\ 0.9,\ 0.99,\ 0.95]$. There are two main differences between the isolated notes and those in real music pieces. The first is that in real music pieces, notes are more likely to stay in the silent state for most of time. The second is the note duration. The isolated notes used in training, for learning the templates, last for about 2 seconds, while most notes in music pieces have much shorter durations. In addition we want a note back to the silence state sooner to get ready for the next note. So we increase the probabilities of remaining in the silent state, and reduce the probabilities of staying in the decay and release states. The diagonal values of the resulting transition matrix are as follows:

$$T_{diag} = \left[\begin{array}{cccc} 0.999 & 0.9 & 0.9 & 0.9 \end{array}\right].$$

All notes start from silence, with the initial state of $[1,\ 0,\ 0,\ 0]$. The most likely state sequence is estimated using the Viterbi algorithm. Because the attack and release stages are not included in the note duration in the training processing, only the decay state is detected as defining the duration of an active note. We set a minimum duration to discard notes which are too short. In the following, we deal with the low frequency noise and repeated notes.

**Low frequency noise**: In piano music, notes in the middle pitch range are more likely to occur than notes at the extremes of the register. It is therefore natural to set pitch-varying transition probabilities from silence to attack. Due to the energy distribution of the spectrogram, which for piano music is always heavily weighted towards low frequencies, the amplitudes of high pitches are relatively small in the gain matrix. This means that a universal transition probability from silence to attack tends to result in many false positives in the low pitch range. To reduce these falsely detected notes, we set pitch-varying probabilities of staying in the silent state to make it harder for the HMM to start a low-pitch note.

$$T_1 = 0.999 + (88 - p)/88 \times 0.001,$$

where $p \in \{1, 2, ..., 88\}$ is the pitch index.

| System | Description |
|---|---|
| Proposed | the system with the transition matrix (4) |
| Proposed-R | the system with the transition matrix (5) |
| Berg | a piano transcription system proposed in [14] |

**Table 1**. Systems tested in the experiments.

**Fast repeated notes**: When a note is repeated quickly, there are few if any frames corresponding to the release and silent states. Especially because the window function blurs the spectrogram, there is no gap between two consecutive notes. Based on the transition matrix in (4), the note states have to cycle through the sequence silence, attack, decay and release. In this case, the second note will not be detected as a new note, but as a continuation of the previous note. To deal with this problem, we simply set a non-zero transition probability to go from decay directly back to attack.

$$T = \left[\begin{array}{cccc} T_1 & 1-T_1 & 0 & 0 \\ 0 & T_2 & 1-T_2 & 0 \\ 0 & (1-T_3)/2 & T_3 & (1-T_3)/2 \\ 1-T_4 & 0 & 0 & T_4 \end{array}\right]. \tag{5}$$

## 3. EXPERIMENTS

The systems are tested on music pieces recorded on a Disklavier piano (ENSTDkCl) from the MAPS database [18]. The transcription experiments are performed on the first 30 seconds of all piano pieces. The templates are trained on the isolated note from the same piano. The spectrogram is computed by the Short-Time Fourier Transform (STFT) with an 4096-sample Hamming window and a hop-size of 441. The Discrete Fourier Transform is performed on each frame with 2-fold zero-padding. The sampling rate is fs = 44100Hz. Beta used in the NMF is 0.5 [4]. We apply a 7-sample median filter to smooth the gain matrix. The value added to the silence state for noise suppression is 0.01. Minimum duration is set to be 60ms. In the experiment we test two systems, denoted by proposed and proposed-R, with different values of T3 (the possibility of staying in decay). We also compare the proposed systems to a state-of-the-art method [14]. Systems in the experiment are listed in the table 3.

**Metrics:** Systems are evaluated by precision ($P$), recall ($R$) and F-measure ($F$), defined as: $P = \frac{N_{tp}}{N_{tp}+N_{fp}}$, $R = \frac{N_{tp}}{N_{tp}+N_{fn}}$, $F = 2 \times \frac{P \times R}{P+R}$, where $N_{tp}$, $N_{fp}$, $N_{fn}$ are the numbers of true positives, false positives and false negatives, respectively. We employ both frame-wise and note-wise evaluation. For each pitch, the detected note is considered note-wise correct if the difference between the detected onset and the ground truth onset is within 50ms.

**Results:** Figure 4 compares the transcription performance of two proposed systems. The proposed-R system provides

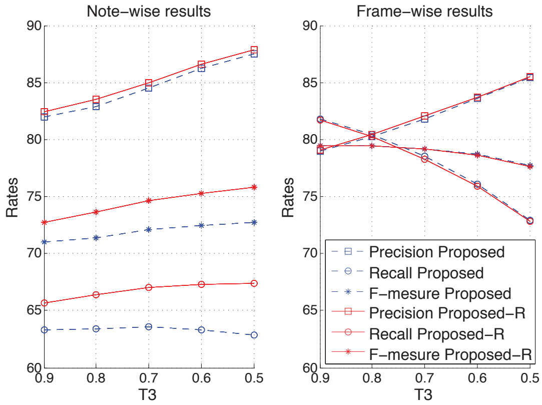

**Fig. 4.** Transcription results.

| System | $P_n$ | $R_n$ | $F_n$ | $P_f$ | $R_f$ | $F_f$ |
|---|---|---|---|---|---|---|
| Proposed | 87.5 | 62.9 | 72.7 | 85.4 | 72.9 | 77.7 |
| Proposed-R | 87.9 | 67.4 | 75.8 | 85.5 | 72.8 | 77.6 |
| Berg [14] | 78.1 | 74.7 | 76.4 | 69.1 | 80.7 | 74.4 |

**Table 2.** A comparison of transcription results.

better note-wise results on precisions (squares), recalls (circles) and F-measures (stars), while the differences of the systems in frame-wise results are subtle. We focus on the note-wise results first. Performance of the proposed-R system constantly increases with decreasing values of T3. Higher recall means more correct notes are found. The corresponding increase in precision means at the same time fewer false positives are detected. There is little difference in precision between the two systems; in both cases the number of false positives is decreased by decreasing the probability of staying in the decay state. With smaller T3, the durations of detected notes are shorter. Then the false positives are more likely to be discarded by minimum duration pruning. The improvement in recall by the proposed system manifests that more true positives are found because we allow transitions from the decay to the attack state. An example of detecting fast repeated notes is shown in Figure 5. We find that more notes are detected when decreasing T3. The F-measure of the proposed-R method increases from 72.7% to 75.8% when T3 decreases from 0.9 to 0.5. The difference between the two systems also increases, from 1.7% to 3.1%, with the decrease of T3. In Figure 4 (right), we find that the F-measure for the frame-wise results decreases with decreasing T3. The results are not influenced by the choice of system. The increasing precision with the decrease of T3 means we found fewer false positives, while the decrease in recall means also fewer true positives were found. On the whole, the F-measure decreases from 79.4% to 77.6% over the range of decreasing T3 values.

In summary, the proposed-R method works well for both note-wise and frame-wise transcription. It helps to detect more true notes for repeated notes and has no obvious impact on frame-wise metrics. The choice of parameter T3 has different effects on the two levels of transcription results, giving an increase in note-wise performance and a decrease in frame-wise performance as T3 is reduced.

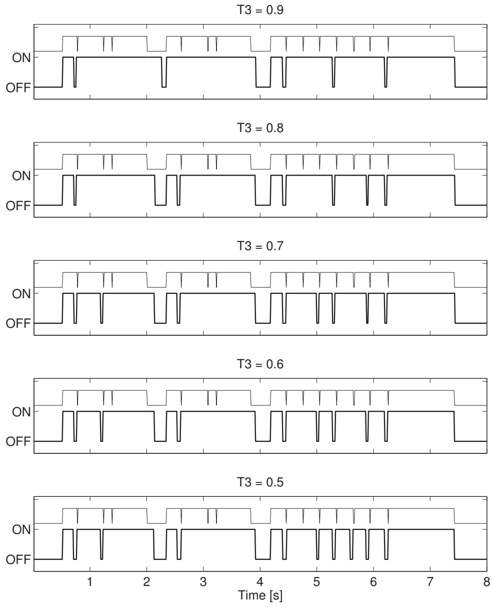

**Fig. 5.** Note tracking results on fast repeated notes with difference T3 for the proposed-R system. In each sub-figure the top line indicates the ground truth notes, while the bottom line shows the detected notes.

By considering both the frame-wise and note-wise results, we choose T3 = 0.5 as the optimal value for the comparison with Berg's method [14]. The results in Tabel 2 show that dealing with the repeated notes (proposed-R) improves note-wise F-measure to 75.8%, which is slightly worse than the system of Berg by 0.6 percentage point, while in frame-wise results the proposed methods outperform the system of Berg by above 3 percentage points on F-measure.

## 4. CONCLUSIONS AND DISCUSSION

In this paper we track piano notes using a pitch-wise HMM on the gains obtained in an NMF framework. To update the gain matrix, templates are trained in advance with four templates per note to represent the different stages of piano sounds: silence, attack, decay and release. Then a four-state HMM is employed to track notes on the gains of each pitch. We increase the likelihood of staying in silence for low pitches and set a minimum duration to reduce short false-positive notes. For quickly repeated notes, we allow the note state to transition from decay back to attack. The experiment shows competitive results on both frame-wise and note-wise transcription. Dealing with fast repeated notes improves the note-wise evaluation by up to 3 percentage points, while no obvious difference is observed on frame-wise results.

The choice for the probability of staying in the decay state is a trade-off between detecting long-duration notes and jumping out from short-duration notes. For this reason, we would like to incorporate onset information or model the duration distribution for note tracking in the future. The transition probabilities from decay to release and to attack are not necessarily the same, but we modelled them as such to avoid introducing another variable. We would expect a further improvement by using individual probabilities for these two state transitions.

## REFERENCES

[1] A. Cont, "Realtime multiple pitch observation using sparse non-negative constraints," in *Proc. 7th International Society on Music Information Retrieval (ISMIR)*, 2006, pp. 206–211.

[2] E. Vincent, N. Bertin, and R. Badeau, "Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 109–112.

[3] N. Bertin, R. Badeau, and E. 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.

[4] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, 2010.

[5] A. Dessein, A. Cont, and G. Lemaitre, "Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2010.

[6] J.J. Carabias-Orti, T. Virtanen, P. Vera-Candeas, N. Ruiz-Reyes, and F.J. Canadas-Quesada, "Musical instrument sound multi-excitation model for non-negative spectrogram factorization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1144–1158, 2011.

[7] K. O'Hanlon and M. Plumbley, "Automatic music transcription using row weighted decompositions," in *Proc. IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP)*, 2013, pp. 16–20.

[8] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[9] M.P. Ryynänen and A. Klapuri, "Polyphonic music transcription using note event modeling," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2005, pp. 319–322.

[10] V. Emiya, R. Badeau, and B. David, "Automatic transcription of piano music based on HMM tracking of jointly-estimated pitches," in *Proc. European Signal Processing Conference (EUSIPCO)*, 2008.

[11] E. Benetos and S. Dixon, "Polyphonic music transcription using note onset and offset detection," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 37–40.

[12] E. Benetos and T. Weyde, "Explicit duration hidden markov models for multiple-instrument polyphonic music transcription," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2013, pp. 269–274.

[13] Z. Duan and D. Temperley, "Note-level music transcription by maximum likelihood sampling," in *Proc. International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 181–186.

[14] T. Berg-Kirkpatrick, J. Andreas, and D. Klein, "Unsupervised transcription of piano music," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 1538–1546.

[15] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Bello, J. Dai, and S. Dixon, "Computer-aided melody note transcription using the Tony software: Accuracy and efficiency," in *Proc. First International Conference on Technologies for Music Notation and Representation (TENOR)*, 2015, p. under review.

[16] P. Grosche, B. Schuller, M. Müller, and G. Rigoll, "Automatic transcription of recorded music," *Acta Acustica united with Acustica*, vol. 98, no. 2, pp. 199–215, 2012.

[17] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2003, pp. 177–180.

[18] V. Emiya, R. Badeau, and B. 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.