AN ATTACK/DECAY MODEL FOR PIANO TRANSCRIPTION

Tian Cheng, Matthias Mauch, Emmanouil Benetos and Simon Dixon
Centre for Digital Music, Queen Mary University of London
{t.cheng, m.mauch, emmanouil.benetos, s.e.dixon}@qmul.ac.uk

ABSTRACT

We demonstrate that piano transcription performance for a known piano can be improved by explicitly modelling piano acoustical features. The proposed method is based on non-negative matrix factorisation, with the following three refinements: (1) introduction of attack and harmonic decay components; (2) use of a spike-shaped note activation that is shared by these components; (3) modelling the harmonic decay with an exponential function. Transcription is performed in a supervised way, with the training and test datasets produced by the same piano. First we train parameters for the attack and decay components on isolated notes, then update only the note activations for transcription. Experiments show that the proposed model achieves 82% on note-wise and 79% on frame-wise F-measures on the 'ENSTDkCl' subset of the MAPS database, outperforming the current published state of the art.

1. INTRODUCTION

Automatic music transcription (AMT) converts a musical recording into a symbolic representation, i.e. a set of note events, each consisting of pitch, onset time and duration. Non-negative matrix factorisation (NMF) is commonly used in the AMT area for over a decade since [1]. It factorises a spectrogram (or other time-frequency representation, e.g. Constant-Q transform) of a music signal into non-negative spectral bases and corresponding activations. With constraints such as sparsity [2], temporal continuity [3] and harmonicity [4], NMF provides a meaningful mid-level representation (the activation matrix) for transcription. A basic NMF is performed column by column, so NMF-based transcription systems usually provide frame-wise representations with note transcription as a post-processing step [5].

One direction of AMT is to focus on instrument-specific music, in order to make use of more information from instrumental physics and acoustics [5]. For piano sounds, several acoustics-associated features, such as inharmonicity, time-varying timbre and decaying energy, are examined for their utilities in transcription. Rigaud *et al.* show that an explicit inharmonicity model leads to improvement in piano transcription [6], while a note-dependent inharmonicity parameter is needed for initialisation. Modelling time-varying timbre not only provides a better reconstruction of the spectrogram, but also improves note tracking results by imposing constraints between note stages (attack, sustain and decay) [7, 8]. For decaying energy, Chen *et al.*'s preliminary work uses an exponential model for energy evolution of notes [9]. Berg-Kirkpatrick *et al.* represent the energy evolution of a piano note by a trained envelope [10]. Cogliati and Duan use a sum of two decaying exponentials to approximate decays of piano partials [11]. Ewert *et al.* represent both time-varying timbre and temporal evolution of piano notes by time-frequency patches [12]. Temporal evolution modelling allows a note event to be represented by a single amplitude parameter for its whole duration, enabling the development of note-level systems with promising transcription results [9, 10, 12].

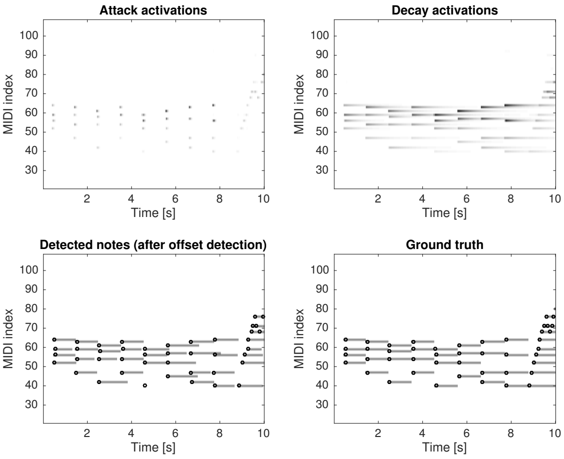


Figure 1: An example of output from the proposed model.

The proposed method is also motivated by piano acoustics. Based on our previous studies on piano decay, we know that exponential decay explains the major energy evolution for each partial in spite of various decay patterns [13]. Here, we further simplify the decay stage using an exponential decay function and a harmonic template per pitch. We separately represent the attack stage for the percussive onset of piano sounds. These two stages are coupled by shared note activations. A supervised NMF framework is used to estimate note activations, and hence activations of the attack and decay stages (see Figure 1). We detect note onsets by peak-picking on attack activations, then offsets for each pitch individually. Experiments show that the proposed method significantly improves supervised piano transcription, and compares favourably to other state-of-the-art techniques.

The proposed method is explained in Section 2. The transcription and comparison experiments are described in Section 3. Conclusions and discussions are drawn in Section 4.

2. METHOD

In this section we first introduce the attack and decay model for piano sounds. Parameters are estimated using a sparse NMF. Then we explain onset and offset detection methods, respectively.

2.1 A model of attack and decay

A piano sound is produced by a hammer hitting the string(s) of a key. It starts with a large energy, then decays till the end of the note. At the attack stage, the strike of the hammer produces a percussive sound. It evolves quickly to an almost harmonic pitched sound, and then immediately enters the decay stage. Considering the different spectral and temporal features, we reconstruct these two phases individually. The attack sound is generated by:

$$V^a_{ft} = \sum_{k=1}^{K} W^a_{fk} H^a_{kt}, \tag{1}$$

where $\mathbf{V^a}$ is the reconstructed spectrogram of the attack phase, as shown in Figure 2(d), and $\mathbf{W^a}$ is the percussive template (Figure 2(e)). $f \in [1, F]$ is the frequency bin, $t \in [1, T]$ indicates the time frame, and $k \in [1, K]$ is the pitch index. Attack activations $\mathbf{H^a}$ (Figure 2(c)) are formulated by the convolution as follows:

$$H^a_{kt} = \sum_{\tau=t-T_t}^{t+T_t} H_{k\tau} P(t-\tau), \tag{2}$$

where $\mathbf{H}$ are spike-shaped note activations, shown in Figure 2(b). $\mathbf{P}$ is the transient pattern, and its typical shape is shown in Figure 5. The range of the transient pattern is determined by the overlap in the spectrogram, with T_t equal to the ratio of the window size and frame hop size.

For the decay part we assume that piano notes decay approximately exponentially [13, 14]. The harmonic decay is generated by

$$V^d_{ft} = \sum_{k=1}^{K} W^d_{fk} H^d_{kt}, \tag{3}$$

where $\mathbf{V^d}$ is the reconstructed spectrogram of the decay phase (Figure 2(g)), and $\mathbf{W^d}$ is the harmonic template (Figure 2(h)). Decay activations $\mathbf{H^d}$ in Figure 2(f) are generated by convolving activations with an exponentially decaying function:

$$H^d_{kt} = \sum_{\tau=1}^{t} H_{k\tau} e^{-(t-\tau)\alpha_k}, \tag{4}$$

where α_k are decay factors, and e^{α_k} indicates the decay rate per frame for pitch k. Offsets are not modelled; instead it is assumed that the energy of a note decays forever. Then the complete model is formulated as follows:

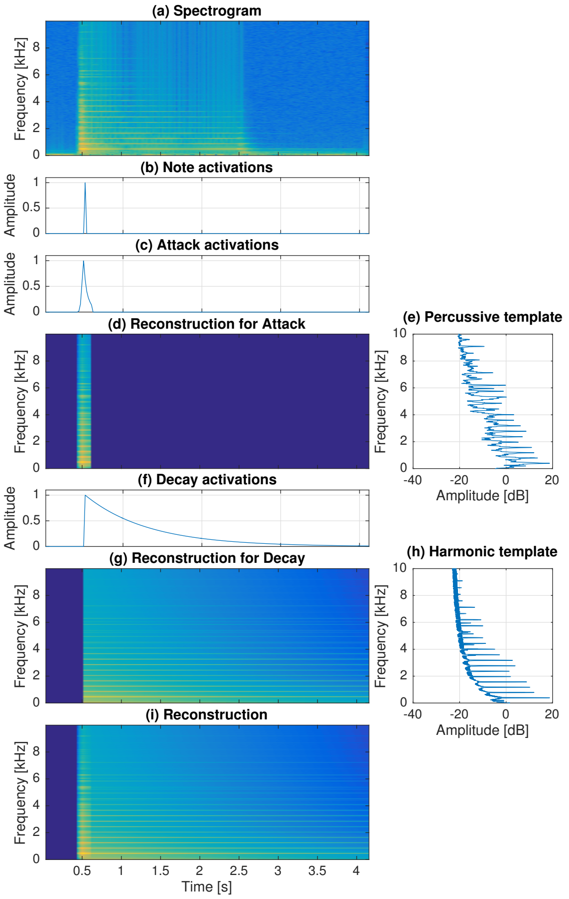


Figure 2: An illustration of the proposed model (note D3 with the MIDI index of 50).

$$\begin{aligned} V_{ft} &= V^a_{ft} + V^d_{ft} \\ &= \sum_{k=1}^{K} W^a_{fk} \sum_{\tau=t-T_t}^{t+T_t} H_{k\tau} P(t-\tau) \\ &+ \sum_{k=1}^{K} W^d_{fk} \sum_{\tau=1}^{t} H_{k\tau} e^{-(t-\tau)\alpha_k}, \end{aligned} \tag{5}$$

where $\mathbf{V}$ is the reconstruction of the whole note, as shown in Figure 2(i).

Parameters $\theta \in \{\mathbf{W^a}, \mathbf{W^d}, \mathbf{H}, \mathbf{P}, \alpha\}$ are estimated by minimising the difference between the spectrogram $\mathbf{X}$ and the reconstruction $\mathbf{V}$ by multiplicative update rules [15]. The derivative of the cost function D with respect to θ is written as a difference of two non-negative functions:

$$\nabla_\theta D(\theta) = \nabla^+_\theta D(\theta) - \nabla^-_\theta D(\theta). \tag{6}$$

The multiplicative algorithm is given by

$$\theta \leftarrow \theta . \nabla^-_\theta D(\theta) ./ \nabla^+_\theta D(\theta). \tag{7}$$

We employ the β-divergence as the cost function. The full update equations are provided online. [1]

[1] https://code.soundsoftware.ac.uk/projects/decay-model-for-piano-transcription.

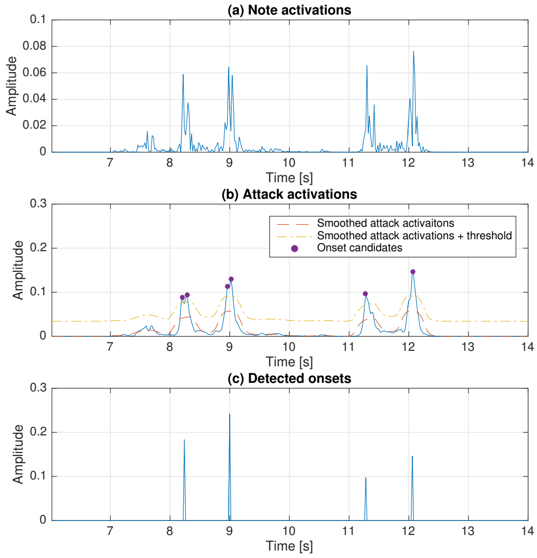


Figure 3: Example of onset detection showing how activations are processed.

2.2 Sparsity

To ensure spike-shaped note activations, we simply impose sparsity on activations $\mathbf{H}$ using element-wise exponentiation after each iteration:

$$\mathbf{H} = \mathbf{H}^{\gamma}, \tag{8}$$

where γ is the sparsity factor, usually larger than 1. The larger the factor is, the sparser the activations are.

A preliminary test confirmed that the number of peaks in activations decreases as the degree of sparsity increases. We also apply an annealing sparsity factor [16], which means a continuously changing factor. In this paper, we set γ to increase from 1 to $\gamma_a \in [1.01, 1.05]$ gradually within the iterations.

2.3 Onset detection

Different playing styles and overlapping between notes may cause a mismatch between the observed attack energy and the trained transient pattern. This results in multiple peaks around onsets in the activations. Figures 3(a) and (b) show note activations and attack activations of pitch G2 in a music excerpt, respectively. Attack activations indicate the actual transient patterns of notes obtained by the proposed model. Therefore, we detect onsets from attack activations by peak-picking. First, we compute smoothed attack activations for each pitch, using a moving average filter with a window of 20 bins. Only peaks which exceed smoothed attack activations by a threshold will be detected as onset candidates, as shown in Figure 3(b). The threshold is adapted to each piece with the parameter δ:

$$Thre = \delta \max_{k,t} H^{a}_{k,t}. \tag{9}$$

We test various $\delta \in \{-21\text{dB}, -22\text{dB}, \ldots, -40\text{dB}\}$ in this paper.

We find that there are still double peaks around onsets after thresholding. In order to deal with this problem, we simply merge pairs of peaks which are too close to each other. We set the minimal interval between two successive notes of the same pitch to be 0.1 second. If the interval between two peaks is smaller than the minimal interval, we generate a new peak. The index of the new peak is a weighted average of the indices of the two peaks, while its amplitude is the sum of that of the two peaks. Figure 3(c) shows detected onsets after merging double peaks. We apply the above process again to get rid of triple peaks.

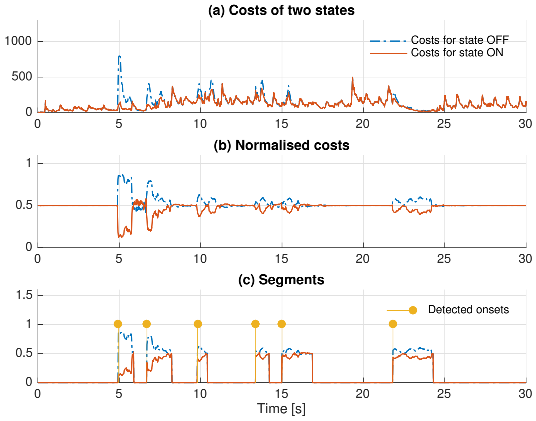


Figure 4: Costs and segments for pitch F3 (MIDI index 53).

2.4 Offset detection

We adapt the method of [12] to detect the offsets by dynamic programming. For each pitch, there are two states $s \in \{0, 1\}$, denoting state ‘off’ and ‘on’ respectively. The costs are defined below:

$$C_k(s,t) = \begin{cases} \sum_{f=1}^{F} D_{KL}(X_{ft}, V_{ft} - V^{k}_{ft}), & s = 0 \\ \sum_{f=1}^{F} D_{KL}(X_{ft}, V_{ft}), & s = 1 \end{cases} \tag{10}$$

where V^k is the reconstruction of pitch k, and $V - V^k$ is the reconstruction excluding pitch k. $D_{KL}(a, b)$ denotes the KL-divergence between a and b. Then we normalise the costs per pitch to sum to 1 in all frames: $\widetilde{C_k}(s,t) = C_k(s,t)/\sum_{\tilde{s}} C_k(\tilde{s},t)$. Figures 4 (a) and (b) show the costs and normalised costs for pitch F3 in a music piece, respectively.

We can find the optimal state sequence by applying dynamic programming on the normalised costs. To do this, we need an accumulated cost matrix and a step matrix to store the smallest accumulated costs and previous states. The accumulated cost matrix D_k is recursively defined as

$$D_k(s,t) = \begin{cases} \min_{\tilde{s}\in\{0,1\}}(D_k(\tilde{s}, t-1) + \widetilde{C_k}(s,t)w(\tilde{s}, s)), & t > 1 \\ \widetilde{C_k}(s,t), & t = 1 \end{cases} \tag{11}$$

where w is the weight matrix, which favours self-transitions, in order to obtain a smoother sequence. In this paper, the weights are $[0.5, 0.55; 0.55, 0.5]$. The step matrix E is defined as follows:

$$E_k(s,t) = \arg\min_{\tilde{s}\in\{0,1\}}(D_k(\tilde{s}, t-1) + \widetilde{C_k}(s,t)w(\tilde{s}, s)), t > 1 \tag{12}$$

The states are given by

$$S_k(t) = \begin{cases} \arg\min_{\tilde{s}\in\{0,1\}} D_k(\tilde{s}, t), & t = T \\ E_k(S_k(t+1), t+1), & t \in [1, T-1] \end{cases} \tag{13}$$

We find that when the activation of the pitch is 0 or very small, the costs of two states are the same or very close, and no state transition occurs. In these parts, the pitch state is off, while dynamic programming can not jump out from the previous state. In order to deal with this problem we need to exclude these parts before applying dynamic programming. Figure 4(c) shows the segmentation by detected onsets and the costs. Each segment starts at a detected onset and ends when the difference of the smoothed normalised costs is less than a set threshold. We track the states of the pitch for each segment individually.

3. EXPERIMENTS

In the experiments we first analyse the proposed model's performance on music pieces produced by a real piano from the MAPS database [17]. Then we compare to three state-of-the-art transcription methods on this dataset and two other synthetic datasets.

To compute the spectrogram, frames are segmented by a 4096-sample Hamming window with a hop-size of 882.[2] A discrete Fourier Transform is performed on each frame with 2-fold zero-padding. Sample frequency f_s is 44100Hz. To lessen the influence of beats in the decay stage [13], we smooth the spectrogram with a median filter covering 100ms. During parameter estimation, we use the KL-divergence ($\beta = 1$) as the cost function. The proposed model is iterated for 50 times in all experiments to achieve convergence.

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. In addition, we use the accuracy in [18] to indicate the overall accuracy: $A = \frac{N_{tp}}{N_{tp}+N_{fp}+N_{fn}}$. We employ both frame-wise and note-wise evaluation [19], denoted by subscript 'f' and 'on', respectively.

3.1 Transcription experiment

The main transcription experiment is performed on the 'ENSTDkCL' subset of the MAPS database [17]. The piano sounds of this subset are recorded on a Disklavier piano. We train percussive and harmonic templates, decay rates and the transient pattern on the isolated notes produced by the same piano. The transcription experiment is run on the music pieces using the first 30s of each piece.[3]

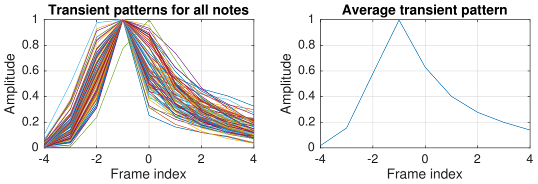


Figure 5: Transient patterns.

Table 1: Note tracking results with different fixed sparsity factors (above) and annealing sparsity factors (below).

γ	P_{on}	R_{on}	F_{on}	A_{on}	δ(dB)
1.00	**88.52**	77.70	**82.24**	**70.54**	-29
1.01	87.70	**78.18**	82.23	70.53	-30
1.02	87.67	77.36	81.80	69.87	-30
1.03	87.22	77.31	81.62	69.66	-31
1.04	86.95	76.84	81.26	69.17	-32
1.05	86.38	75.99	80.51	68.14	-33
1 → 1.01	87.77	**78.08**	82.18	70.49	-30
1 → 1.02	**88.49**	77.79	**82.36**	**70.73**	-30
1 → 1.03	88.22	77.78	82.27	70.60	-31
1 → 1.04	87.86	77.66	82.09	70.35	-32
1 → 1.05	86.83	77.83	81.76	69.84	-34

3.1.1 The training stage

The training stage includes two rounds. In the first round, we first fix note activations ($\mathbf{H}$) for each isolated note according to the ground truth, then update all other parameters ($\mathbf{W^a}$, $\mathbf{W^d}$, $\mathbf{P}$ and α). The transient patterns are normalised to maximum of 1 after each iteration. In theory, the transient patterns follow a certain shape and could be shared by all pitches. So we use the average of the trained transient patterns to reduce the number of parameters and to avoid potential overfitting. The trained transient patterns and the average transient pattern are shown in Figure 5. In the second round, we fix the note activations ($\mathbf{H}$) and the transient pattern ($\mathbf{P}$), then update all other parameters ($\mathbf{W^a}$, $\mathbf{W^d}$ and α).

3.1.2 Transcription results

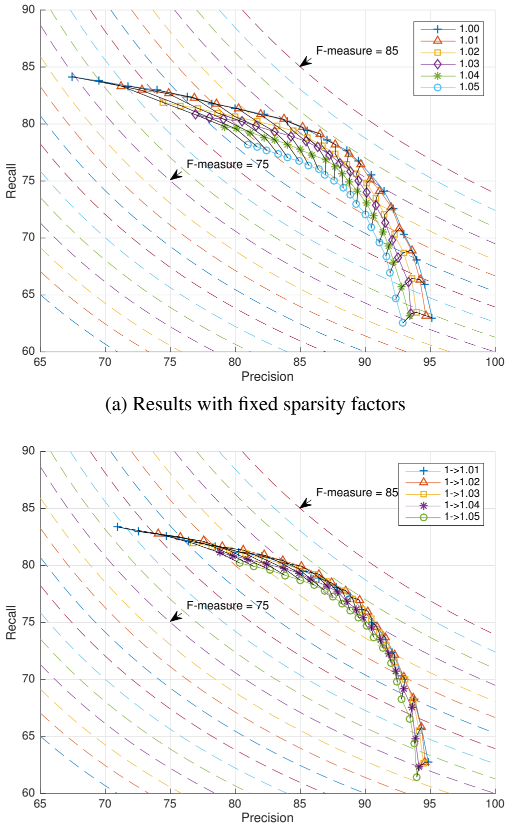
For transcription, we update note activations $\mathbf{H}$, keeping parameters ($\mathbf{W^a}, \mathbf{W^d}, \mathbf{P}$ and α) fixed from the training stage. Table 1 shows note tracking results (presented as percentage) using different sparsity factors. The optimal thresholds are shown in the last column. The top part of Table 1 are results using fixed sparsity factors. The best results are achieved without the sparsity constraint (γ = 1.00), with an F-measure of 82.24%. The performance decreases with increasing sparsity factor. The second part of the experiment gives results for using annealing sparsity. The best F-measure is 82.36% with the setting (1.00 → 1.02). The difference between the best and the worst F-measure is only 0.6 percentage points. In general, all results with different sparsity constraints are considerably good with optimal thresholds, and the optimal threshold decreases when sparsity gets higher. However, F-measures considering both onsets and offsets are quite low, around 40%.

[2] A 20ms hop size is used to reduce computation time. For frame-wise evaluation, transcription results are represented with a hop size of 10ms by duplicating every frame.

[3] The proposed model runs at about 3 × real-time using MATLAB on a MacBook Pro laptop (I7, 2.2GHz, 16GB).

In the proposed model, the activation of each note decays after its onset, as shown in the decay activations of Figure 1. Given a note is played, we consider two situations. In the first case, there is another note of the same pitch being played later. We know that in this case the first note should be ended then. If the activation of the first note has already decreased to a low level, there is little influence on detecting the second note. However, if these two notes are very close, detection of the second note might be missed because of the remaining activation of the first note. In the second case, there is another note of a different pitch being played. The activation of the first note won't be changed by the attack of the second note in our model, while for standard NMF, there is always some interference with the first note's activation.

We compute the performance using thresholds ranging from -40 to -21dB to study performance variations as a function of the threshold. Figure 6(a) shows the results for different fixed sparsity factors. It is clear that precision decreases with the increase of the threshold, while recall increases. The higher the sparsity factor is, the more robust the results are on threshold changes. This is because small peaks in activations are already discounted when imposing sparsity, as shown in Figure 7. Lowering the threshold does not bring many false positives. Results with higher sparsity are less sensitive to the decrease of the threshold. However, when the threshold becomes larger, the results with low sparsity still outperform those with high sparsity. With a larger threshold, the number of true positives decreases. There are more peaks in activations when using lower sparsity, so more true positives remain. This favours the assumption that the true positives have larger amplitudes. Figure 6(b) shows the robustness of using annealing sparsity factors. The transcription results are close to each other. With annealing sparsity, the results are better and more tolerant to threshold changes.



(a) Results with fixed sparsity factors

(b) Results with annealing sparsity factors

Figure 6: Performance (presented percentage) using different sparsity factors and thresholds. The sparsity factors are indicated by different shapes, as shown in the top-right box. Lines connecting different shapes are results achieved via the same threshold. The threshold of the top set is -40dB, and the bottom set is -21dB. The dashed lines show F-measure contours, with the values dropping from top-right to bottom-left.

3.2 Comparison with state-of-the-art methods

We apply a comparison experiment on three datasets, pieces from a real piano ('ENSTDkCl') and a synthetic piano ('AkPnCGdD') in the MAPS database [17], and another 10 synthetic piano pieces (denoted as 'PianoE') used in [12]. All experiments are performed on the first 30s of each piece. We compare to two top transcription methods. Vincent *et al.*'s method applies adaptive spectral bases generated by linear combinations of narrow-band spectra, so the spectral bases have a harmonic structure and the flexibility to adapt to different sounds [20]. Benetos and Weyde's method employs 3 templates per pitch, and the sequence of templates is constrained by a probabilistic model [21]. In the PianoE dataset, we also compare to another state-of-the art method of Ewert *et al.* [12]. This method identifies frames in NMD patterns with states in a dynamical system. Note events are detected with constant amplitudes but various durations. In the comparison experiment, the proposed system is also trained on isolated notes from the AkPnCGdD and PianoE pianos. Vincent *et al.*'s method is performed in an unsupervised way, to indicate what can be achieved without training datasets. We use the version of Benetos and Weyde's method from the MIREX competition [22]. We have access to the code and train this model on isolated notes of the corresponding pianos. For Ewert's method we only have access to the published data in [12]. These two methods are performed in a supervised way.

Based on previous analysis, we employ the following parameters for the proposed model in comparison experiments. The sparsity factor is $\gamma = 1 \rightarrow 1.04$ by balancing among note tracking results and the robustness to different thresholds. Onsets are detected with threshold $\delta = -30\text{dB}$. In the first dataset ('ENSTDkCl'), results of other methods are also reported with optimal thresholds with best note-wise F-measures. Then the same thresholds are used for two synthetic piano datasets.

Results on piano pieces from the 'ENSTDkCl' subset are shown in Table 2(a). The proposed model has a note tracking F-measure of 81.80% and a frame-wise F-measure of 79.01%, outperforming Vincent *et al.*'s unsupervised method by around 10 and 20 percentage points,

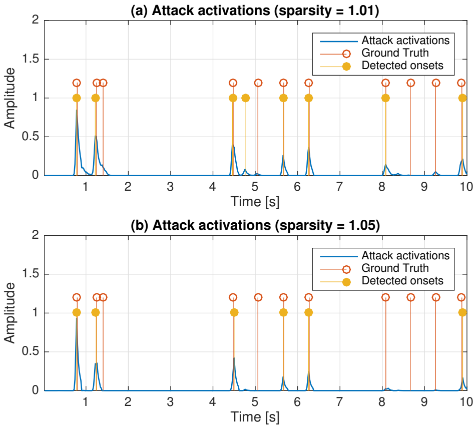


Figure 7: Detected onsets with different sparsity for pitch G4 (MIDI index 67).

Table 2: The comparison experiment

(a) Transcription results on 'ENSTDkCl'

Method	F_{on}	A_{on}	F_f	A_f
Decay	**81.80**	**69.94**	**79.01**	**65.89**
Vincent [20]	72.15	57.45	58.84	42.71
Benetos [21]	73.61	59.73	67.79	52.15

(b) Transcription results on 'AkPnCGdD'

Method	F_{on}	A_{on}	F_f	A_f
Decay	**84.63**	**74.03**	**80.81**	**68.39**
Vincent [20]	79.86	67.32	69.76	55.17
Benetos [21]	74.05	59.57	53.94	38.65

(c) Transcription results on 'PianoE'

Method	F_{on}	A_{on}	F_f	A_f
Decay	81.28	69.12	**66.77**	**51.63**
Vincent [20]	79.41	66.39	58.59	42.45
Benetos [21]	83.80	**72.82**	60.69	44.24
Ewert [12]	**88**	-	-	-

respectively. Results of Benetos and Weyde's method are in between.

Results on the synthetic piano 'AkPnCGdD' are shown in Table 2(b). In general, all methods perform better on this dataset than on the 'ENSTDkCl' dataset, especially on note tracking results. The proposed model has the best results (84.63% on note tracking F-measure and 80.81% on frame-wise F-measure), outperforming all other methods by at least 5 percentage points.

Results on the other synthetic dataset 'PianoE' are shown in Table 2(c). Note tracking results of all methods are good but frame-wise results are poor. Ewert *et al.*'s method performs the best on note tracking (88% on F-measure), and Benetos and Weyde's method is the second (83.80% on F-measure). The proposed model only outperforms Vincent *et al.*'s method, with F-measures of 81.28% and 79.41% for these two methods respectively. However, the proposed model remains the best on the frame-wise F-measure (66.77%). Pieces in this dataset are from a piano competition. Many notes have very short durations. The remaining energies of a short note in the proposed model may interfere with later notes, causing false negatives.

A supervised neural network model also works on the MAPS database for piano transcription [23]. Besides an acoustic model, the method employs a music language model to capture the temporal structure of music. Although the method is not directly comparable, it is noticeable that our method exceeds its results by at least 5 percentage points on F-measures. When tested on the real recordings using templates trained in the synthetic piano notes, the proposed method has both F-measures of around 65%, outperforming the method of [23] by 10 percentage points on note-wise F-measure in a similar experiment.

4. DISCUSSION AND CONCLUSION

In this paper we propose a piano-specific transcription system. We model a piano note as a percussive attack stage and a harmonic decay stage, and the decay stage is explicitly modelled as an exponential decay. Parameters are learned in a sparse NMF, and transcription is performed in a supervised way. The proposed model provides promising transcription results, with around 82% and 79% for note tracking and frame-wise F-measures in music pieces from a real piano in the 'ENSTDkCl' dataset. The annealing sparsity factor improves both performance and the robustness of the proposed model. The comparison experiment shows that the proposed model outperforms two state-of-the-art methods by a large margin on real and synthetic pianos in the MAPS database. On a different synthetic dataset, the other methods performs relatively better, especially on note tracking, while the proposed method remains best on frame-wise metrics.

The proposed model can also be understood as a deconvolution method in which a patch is parameterised by two sets of templates and activations. One advantage of the proposed model is that we can build a note-level system by deconvolution, which has provided good transcription results [9, 10, 12]. The other is that using parametric patches reduces the number of parameters. The model also provides us with a way to analyse piano decay rates.

In the future, we would like to represent a note's decay stage by a decay filter instead of a decay rate, which is more in line with studies on piano decay [13]. Secondly, the good performance on piano music transcription is partly due to the availability of the training datasets. We would like to build an adaptive model, which could work in a more general scenario, hence more automatically. Finally, we are keen to find a way to estimate note offsets more accurately in the proposed model.

5. ACKNOWLEDGEMENT

TC is supported by a China Scholarship Council/Queen Mary Joint PhD Scholarship. EB is supported by a Royal Academy of Engineering Research Fellowship (grant no. RF/128). We would like to thank Dr. Sebastian Ewert for his comments on this paper.

6. REFERENCES

[1] P. Smaragdis and J. C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proc. IEEE WASPAA*, pages 177–180, 2003.

[2] A. Cont. Realtime multiple pitch observation using sparse non-negative constraints. In *Proc. ISMIR*, pages 206–211, 2006.

[3] T. Virtanen. Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria. *IEEE Trans. on Audio, Speech and Language Processing*, 15(3):1066–1074, 2007.

[4] N. Bertin, R. Badeau, and E. Vincent. Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Trans. on Audio, Speech and Language Processing*, 18(3):538–549, 2010.

[5] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013.

[6] F. Rigaud, A. Falaize, B. David, and L. Daudet. Does inharmonicity improve an NMF-based piano transcription model? In *Proc. ICASSP*, pages 11–15, 2013.

[7] E. Benetos and S. Dixon. Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model. *The Journal of the Acoustical Society of America*, 133(3):1727–1741, 2013.

[8] T. Cheng, S. Dixon, and M. Mauch. Improving piano note tracking by HMM smoothing. In *Proc. EUSIPCO*, pages 2009–2013, 2015.

[9] Z. Chen, G. Grindlay, and D. Ellis. Transcribing multi-instrument polyphonic music with transformed eigeninstrument whole-note templates. In *MIREX*, 2012.

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

[11] A. Cogliati and Z. Duan. Piano music transcription modeling note temporal evolution. In *Proc. ICASSP*, pages 429–433, 2015.

[12] S. Ewert, M. D. Plumbley, and M. Sandler. A dynamic programming variant of non-negative matrix deconvolution for the transcription of struck string instruments. In *Proc. ICASSP*, pages 569–573, 2015.

[13] T. Cheng, S. Dixon, and M. Mauch. Modelling the decay of piano sounds. In *Proc. ICASSP*, pages 594–598, 2015.

[14] G. Weinreich. Coupled piano strings. *The Journal of the Acoustical Society of America*, 62(6):1474–1484, 1977.

[15] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. Advances in Neural Information Processing Systems 13*, pages 556–562, 2000.

[16] T. Cheng, S. Dixon, and M. Mauch. A deterministic annealing EM algorithm for automatic music transcription. In *Proc. ISMIR*, pages 475–480, 2013.

[17] V. Emiya, R. Badeau, and B. David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Trans. on Audio, Speech, and Language Processing*, 18(6):1643–1654, 2010.

[18] S. Dixon. On the computer recognition of solo piano music. In *Proc. Australasian Computer Music Conference*, pages 31–37, 2000.

[19] M. Bay, A. F. Ehmann, and J. S. Downie. Evaluation of multiple-F0 estimation and tracking systems. In *Proc. ISMIR*, pages 315–320, 2009.

[20] E. Vincent, N. Bertin, and R. Badeau. Adaptive harmonic spectral decomposition for multiple pitch estimation. *IEEE Trans. on Audio, Speech and Language Processing*, 18(3):528–537, 2010.

[21] E. Benetos and T. Weyde. An efficient temporally-constrained probabilistic model for multiple-instrument music transcription. In *Proc. ISMIR*, pages 701–707, 2015.

[22] E. Benetos and T. Weyde. Multiple-F0 estimation and note tracking for MIREX 2015 using a sound state-based spectrogram factorization model. In *MIREX*, 2015.

[23] S. Sigtia, E. Benetos, and S. Dixon. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5):927–939, 2016.