Assessing the Relevance of Onset Information for Note Tracking in Piano Music Transcription
Valero-Mas, JJ; Benetos, E; Iñesta, JM; 2017 AES International Conference on Semantic Audio

© 2017 Audio Engineering Society

For additional information about this publication click this link.
http://qmro.qmul.ac.uk/xmlui/handle/123456789/22497

Information about this research object was correct at the time of download; we occasionally make corrections to records, please therefore check the published record when citing. For more information contact scholarlycommunications@qmul.ac.uk
Assessing the Relevance of Onset Information for Note Tracking in Piano Music Transcription

Jose J. Valero-Mas\textsuperscript{1}, Emmanouil Benetos\textsuperscript{2}, and José M. Iñesta\textsuperscript{1}

\textsuperscript{1}Pattern Recognition and Artificial Intelligence Group, University of Alicante, Spain
\textsuperscript{2}Centre for Digital Music, Queen Mary University of London, UK

Correspondence should be addressed to Jose J. Valero-Mas (jjvalero@dlsi.ua.es)

ABSTRACT

In Automatic Music Transcription, onset information is useful for correcting timing issues of multi-pitch estimation processes and obtain note-level representations of audio signals. Although this idea has been often used in transcription systems, it is still unclear to which degree its use is beneficial. We address this question by studying the influence of the accuracy of onset information in piano music transcription. Results indicate that note tracking results improve when the onset information provided is accurately estimated and properly included with the correct strategy. Additionally, results depict an important accuracy gap in note tracking when considering ground-truth onset information compared to using an automatic onset estimation algorithm, showing the need for more accurate onset detection methods for music transcription systems.

1 Introduction

Automatic Music Transcription (AMT) stands for the process of automatically retrieving a high-level symbolic representation of the music present in an audio signal [1]. Such ambitious task requires a large number of disparate processes which complementary describe the signal, being pitch estimation, rhythm analysis or instrument detection some possible examples, among many others [2].

Most AMT systems comprise two stages [3]: multi-pitch estimation (MPE), in which the system estimates the active pitches in each frame of the signal; and note tracking, which processes the frame-based MPE result to produce a list of note events in terms of a pitch, onset and offset. While both representations constitute abstractions of the music signal, namely frame-level and note-level transcriptions respectively, the note tracking step is the one for obtaining musically-meaningful representations.

Frame-level transcription has been widely addressed over the years. Examples of successful systems span data-driven techniques, with a remarkable use of neural networks for modelling such extraction [4, 5], to signal processing methodologies, with the spectrogram factorisation paradigm a very successful example [3] of such family of methods. The idea behind spectrogram factorisation techniques resides on decomposing the initial spectrogram into a series of pitch templates and pitch activations. Non-negative Matrix Factorisation (NMF) and Probabilistic Latent Component Analysis (PLCA) [6] constitute practical and successful examples of such factorisation principles.
On the contrary, note tracking has not received much attention [7]. Note-level transcriptions are commonly obtained by processing frame-level representations with minimum-length pruning processes for eliminating spurious detections and gap-filling stages for removing small gaps between consecutive pitches. Implementations of these ideas range from rule-based systems [6] to hidden Markov models (HMMs) [8] or dynamic Bayesian networks (DBNs) [9].

In general, MPE systems are imprecise in terms of timing. Typical issues comprise their tendency to miss note starts, mainly due to the irregularity of the signal in the attack stage, or the over-segmentation of long notes or the merge of repeated notes (e.g., tremolo passages) into single events. The use of timing information has been considered for tackling the aforementioned issues; examples include [10] which uses onset events for segmenting the signal before the pitch estimation stage, or [11] which post-processes the MPE results with onset information for correcting timing issues.

In this paper we study the potential of onset information for improving note tracking performance for automatic piano transcription. The idea is to assess how relevant the goodness of the onset information is in this context: on the one hand, we use ground truth onset information (oracle approach) to study a possible upper bound in performance; on the other hand, we consider onset events obtained with state-of-the-art onset detection algorithms (practical approach) and compare those results with the oracle ones to point out the limitations found. For that, we model the note tracking task as a sequence-to-sequence transduction problem (raw estimation to onset-based corrected estimation) and thus consider the use of Finite State Transducers (FSTs) for performing it, which constitutes a paradigm that, to our best knowledge, has not been previously considered. Also note that, while onset information has previously been incorporated in some AMT systems, to the authors’ knowledge no existing work has thoroughly studied the influence of onset information for improving automatic music transcription performance.

The rest of the paper is structured as follows: Section 2 explains the scheme proposed for the experiment; Section 3 introduces the multipitch estimation algorithms considered; Section 4 describes the onset detection strategies contemplated; Section 5 presents the proposed note tracking process; Section 6 details the evaluation methodology and describes the results; finally, Section 7 provides conclusions and future directions.

2 Scheme proposed

To carry out the proposed study, we have implemented the scheme shown in Figure 1. Audio signals undergo an MPE process which outputs frame-level transcription $T_F(p, t)$, that is a binary representation depicting whether pitch $p$ at time frame $t$ is active. Simultaneously, onset events $(o_i)_{i=1}^L$ are estimated with an onset detection algorithm. Eventually both analyses are merged in a note tracking stage obtaining the note-level abstraction $T_N(p, t)$.

The details concerning each of the processes in Figure 1 will be explained in the following sections.

3 Multipitch estimation

We have considered two MPE approaches in our work: the system by Vincent et al. [12] based on adaptive NMF and the one by Benetos et al. [13] based on dictionary-based PLCA. Both models output a pitch activation probability $P(p, t)$, where $p$ stands for pitch in the MIDI scale and $t$ for time instant. We set a temporal resolution of 10 ms for the input time-frequency representation and output pitch activation.

Vincent et al. [12] decompose a spectrogram with an NMF-like method by modelling each template spectrum as a weighted sum of narrowband spectra that represents a group of adjacent harmonic partials. This enforces harmonicity and spectral smoothness while it allows adapting the spectral envelope to the instruments in the piece.

Benetos et al. [13] take as input a constant-Q transform (CQT) spectrogram with a resolution of 60 bins per octave and decomposes it into a series of pre-extracted log-spectral templates per pitch, instrument source, and tuning deviation from ideal tuning. Model parameters are estimated using Expectation-Maximization (EM) [14].
In both cases, \( P(p, t) \) is processed to obtain the \( T_F(p, t) \) binary representation: \( P(p, t) \) is normalised to its global maximum so that \( P(p, t) \in [0, 1] \) and a 7-element median filter is applied over time to smooth it. Then, the function is binarised using a threshold value \( \theta = 0.1 \), which is obtained taking the work in [12] as a reference and refining it for the data used in this work. Finally, a pruning stage with a minimum-length filter of 50 ms is applied to remove spurious note detections. These values were obtained by performing initial exploratory experiments to optimise the parameters for the data considered.

4 Onset Detection

As mentioned, our aim is to study the relevance of the onset information accuracy when considered for note tracking. Thus, we distinguish two situations: a first one considering ground-truth onset events and a second one with estimated onset information. For the latter case we have selected three different algorithms given their good results reported in literature: Semitone Filter-Bank (SFB) [15], SuperFlux (SF), and ComplexFlux (CF) [16, 17]. These processes output a list \((o_i)_{i=1}^L\) whose elements represent the time positions of the \( L \) onsets detected. We shall now introduce them.

SFB applies a harmonic semitone filter bank to each analysis window of the magnitude spectrogram and retrieves the energy of each band (root mean square value); a first-order derivative is then applied to each band; negative results are filtered out as only energy increases may point out onset information; finally, all bands are summed to obtain a function whose peaks represent the onset events.

SF and CF expand the idea of the spectral flux signal descriptor by substituting the difference between consecutive analysis windows by tracking spectral trajectories in the spectrum together with a morphological dilation filtering process. This suppresses vibrato articulations in the signal which tend to increase false positives.

The time-frequency analysis parameters of the algorithms have been set to their default values. As all of them comprise a final thresholding stage, where 25 different values equally spaced in the range \((0, 1)\) have been tested to check the influence of that parameter. Onset lists \((o_i)_{i=1}^L\) have been filtered with an averaging 30 ms to avoid overestimation issues by the algorithms following [18].

5 Note Tracking

\( T_F(p, t) \) can be considered a set of \(|\mathcal{P}|\) binary sequences of \(|I|\) symbols. Hence, elements \((o_i)_{i=1}^L\) may be used as delimiters for segmenting each sequence \( p_i \in \mathcal{P} \) in \( L + 1 \) subsequences, resulting in a frame-level abstraction quantised by the onset information:

\[
T_F(p_i, t) = T_F(p_i, 0 : o_1) \ldots || T_F(p_i, o_L : |I| - 1)
\]

where \(||\) represents the concatenation operator, \( p_i \) the pitch band at issue and \( L \) the total number of onsets.

Once onset information has been included in \( T_F(p, t) \) we can process each subsequence for each pitch value \( p_i \in \mathcal{P} \) separately for correcting the errors committed. For that, we have considered the use of Finite State Transducers (FSTs), a type of automaton which transforms a sequence of symbols \(x_0, x_1, ..., x_N\) into another sequence \(y_0, y_1, ..., y_Y\) [19]. The input to the FST is each single onset-based subsequence whereas the output is another sequence in which some of the elements have been changed following a particular policy.

Given that each subsequence is a series of ones and zeros representing pitch activations and silences respectively, the two possible actions to model are either activating or deactivating sections. We focus on the former case, i.e. assuming the MPE process misses active areas. Thus, this note tracking approach tackles the MPE issues of missing onset events in attack phases and the breaking of notes. The main reason for only tackling one of the two types of errors is to assess how beneficial can be the use of onset information for post-processing an MPE estimation when considering a very simplistic note tracking approach. This may somehow depict a lower limit in the note tracking figures that may be surpassed if more sophisticated approaches are considered.

Let the 6-tuple \( \Pi = (Q, \Sigma, \Lambda, \delta, \lambda, q_1) \) define our transducer. As we are dealing with binary sequences, the input alphabet is \( \Sigma = \{0, 1\} \). Its possible states are \( Q = \{q_1, q_2\} \) connected with transitions \( \delta(q_1, 0) = q_1, \delta(q_1, 1) = q_2 \) and \( \delta(q_2, a) = q_2 \) where \( a \in \Sigma \). The output alphabet \( \Lambda = \{v_1, v_2\} \) is a non-binary representation which is parsed once the subsequence has been processed to model different FST behaviours. The outputs are given by \( \lambda(q_1, 0) = v_1, \lambda(q_2, 0) = v_2 \) and \( \lambda(b, 1) = 1 \) where \( b \in Q \). Finally, \( q_1 \) represents the initial state of the process. This transducer \( \Pi \) is graphically shown in Figure 2.
To model different performances of the FST we parse symbols $v_1$ and $v_2$ to values of the input alphabet $\Sigma$ following three different policies. For clarity, let $\zeta(v_1) \in \Sigma$ be the first element after $v_1$, which is different to it and let # represent the end-of-string character. All policies fill the gaps in-between active areas and two of them additionally fill other gaps which may be present. Policy (i) fixes $v_1 = 1$ and $v_2 = 1$ if $\zeta(v_2) = 1$ or, alternatively, $v_2 = 0$ if $\zeta(v_2) = #$, which fills the possible gap between the onset and the first active area. Policy (ii) fixes $v_1 = 0$ and $v_2 = 1$, thus filling the possible gap between the last active area and the end of the sequence. Policy (iii) is equivalent to (i) but setting $v_1 = 0$, thus not filling any other type of gap. Figure 3 graphically shows their behaviour.

Finally, before the FST processes the subsequences, they undergo a pruning stage of 50 ms for removing spurious detections.

6 Evaluation

6.1 Datasets

We consider the use of the MAPS database [20] containing audio piano performances (both from real and synthesised pianos) synchronised with MIDI annotations. From that we have taken the pieces of the MUS set recorded with the Disklavier piano in both “ambient” and “close” configurations (i.e., recording microphones near and far from the source, respectively). We have also used the Saarland Music Data (SMD) collection [21] that comprises 50 piano pieces (audio and MIDI aligned) also recorded with a Disklavier. As in other AMT works, we only considered the first 30 seconds of each piece. Table 1 provides a summary of these sets.

<table>
<thead>
<tr>
<th>Set</th>
<th>Pieces</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPS-Close</td>
<td>30</td>
<td>7,353</td>
</tr>
<tr>
<td>MAPS-Ambient</td>
<td>30</td>
<td>8,764</td>
</tr>
<tr>
<td>Saarland SMD</td>
<td>50</td>
<td>12,231</td>
</tr>
</tbody>
</table>

6.2 Evaluation metrics

Since we aim at assessing the relevance of using proper onset information for note tracking, we shall evaluate both tasks.

An estimated onset is considered to be correct if its corresponding ground-truth annotation is within a ±50 ms window of it [22].

In terms of note tracking, we shall restrict ourselves to the onset-based figure of merit as we are not considering note offsets. Thus, a detected note event is assumed to be correct if its pitch matches the corresponding ground-truth pitch and its onset is within a ±50 ms lapse of the corresponding ground-truth onset [23].

For assessing the tasks we may define the figures of merit precision (P), recall (R) and F-measure ($F_1$) as follows: $P = \frac{N_{OK}}{N_{DET}}$, $R = \frac{N_{OK}}{N_{GT}}$, $F_1 = \frac{2 \cdot P \cdot R}{P + R}$. $N_{OK}$ stands for the number of correctly detected events (onsets or notes, depending on the case), $N_{DET}$ for the number of total events detected and $N_{GT}$ the total amount of ground-truth events.

6.3 Results and discussion

Table 2 shows the results obtained for the onset detection process, which constitute the average and deviation of the figures obtained when evaluating each dataset using the 25 threshold values considered.

The high precision figures obtained state the robustness of these algorithms against false alarm detections in these data. Recall figures, though, are not that consistent: SFB commits a number of false positive errors while SF and CF seem to properly deal with them. The $F_1$ figures obtained show the performance limitations of these methods. For instance, the best-case scenarios are the SF and CF algorithms when tackling the MAPS-Close set (average $F_1 = 0.82$, possibly due to being the dataset recorded in the most favourable conditions, i.e. close to the source) which are far from a score of 1.
onset information for note tracking. Nevertheless, the actual point here is the need for accurate onset information. As shown, SF and CF improve results when compared to a simple pruning stage (e.g., an improvement around 5% to 10% in $F_1$ may be achieved in the MAPS-Ambient set depending on the MPE method with respect to the single pruning stage), but these figures are far from results achieved with ground-truth onset information (e.g., ground-truth onset information implies a further improvement of up to 5% in $F_1$ on top of the improvement achieved by SF and CF in the Saarland set). Furthermore, there seems to be more room for improvement in the MAPS-Ambient set than in the rest, possibly due to being the set with the most unfavourable recording conditions (far from the source) and thus the one with the lowest figures in both onset estimation (cf. Table 2) and the MPE process (qualitatively reflected on the note tracking scores when not considering onset information, i.e. $F_1 \approx 0.45$). Additionally, threshold values maximising onset estimation in the entire collections (for all sets, these threshold values are around 0.5 for SF and CF, reporting $F_1 \approx 0.8$, and 0.15 for SFB, achieving $F_1 \approx 0.7$) also exhibit the maximum for note tracking results. This reveals a relation between the accuracy of onset information and the success of the note tracking process (i.e., the better onset detection, the better note tracking), with the ideal case being the one considering ground-truth onset information.

Figures obtained when considering Policy (ii) (Fig. 5) and Policy (iii) (Fig. 6) of the FST do not show such improvement for the results in note tracking. For policies (ii) and (iii), results obtained when onset information is not considered outperform all other cases. Clearly,
Fig. 4: Note tracking results (F₁ score) obtained when applying Policy (i) in the FST for the MPE systems considered.

Fig. 5: Note tracking results (F₁ score) obtained when applying Policy (ii) in the FST for the MPE systems considered.

Fig. 6: Note tracking results (F₁ score) obtained when applying Policy (iii) in the FST for the MPE systems considered.
the fact that Policy (i) is able to correct missed attack events by the MPE stage makes it stand as a better alternative for note tracking than the other policies considered. Moreover, this fact states the relevance of the note tracking stage: when providing onset information to the system, a proper strategy has to be followed to correctly incorporate that knowledge and take advantage of it. Thus, the use of more elaborated tracking processes which may take advantage of the particularities of piano notes should report an improvement.

Additionally, it can be checked that the MPE method by Vincent et al. [12] consistently improves results with respect to Benetos et al. [13]: the figures obtained by the former method outperform the latter in around 5% to 10% in F1, which suggests that the former method is more precise in terms of timing than the latter one. Finally, the improvement in the note tracking results of both MPE methods when onset information is considered states the robustness of onset-based tracking when compared to a basic pruning approach.

Future work considers the further exploration of Finite State Transducers for note tracking by studying more complex architectures so that, false alarm errors may be tackled. Also, we aim at exploring human-computer interactive methods to obtain accurate onset information with the least annotation effort to narrow the accuracy gap between ground-truth and estimated onset events. Moreover, combining pitch and onset salience information rather than their binarised versions might report some improvements. Finally, machine learning could be considered to automatically infer the proper policy to combine pitch and onset events.

Acknowledgements

This work was supported by the FPU program of the University of Alicante (UAFPU2014–5883) and the Spanish Ministerio de Economía y Competitividad through project TIMuL (No. TIN2013–48152–C2–1–R, supported by EU FEDER funds). EB is supported by a UK Royal Academy of Engineering Research Fellowship (grant no. RF/128).

References


