Skipping the Frame-Level: Event-Based Piano Transcription With Neural Semi-CRFs

Yujia Yan, Frank Cwitkowitz, Zhiyao Duan
University of Rochester

In this work...

- We demonstrate **a simple, fast, yet effective formulation** for automatic piano transcription.

- The proposed paradigm may be beneficial to other areas that rely on frame-level estimates, e.g., DCASE (detection and classification of acoustic scenes and events).

Automatic Piano Transcription

Input

Transcribe

Output

Each event is also associated with a *velocity*, representing how hard the key is pressed.

Previous v.s. Proposed

- Previous neural network based approaches (e.g., Hawthorne et al. [2018], Kong et al. [2020], Kwon et al.[2020]):
  - Predict different stages of a note event, i.e., the onset, offset, and pitch activations, separately.
  - Combine disjoint predictions into events using post-processing (e.g., thresholding, peak picking, or hidden Markov model filtering).

- Key idea of the proposed approach
  - Formulate piano transcription as the direct prediction of note events in one-stage using semi-CRFs (conditional random fields)

Proposed Semi-CRF Approach

Target output: a set of events such as notes and pedals, represented as <*onset*, *offset*, *eventType*> tuples.
Events of a certain *eventType* (a specific note/pedal) are non-overlapping (onset/offset frame indices are allowed to overlap), e.g., [0,0], [2,4], [4,5].
Posterior probability of events is evaluated through two **score functions**, which are parameterized by neural networks. The model is trained end to end via **MLE**.

A set of events

Scores how likely [i, j] is an event of *eventType*

$$p_\theta(\mathcal{Y}_{eventType}|\mathcal{X}) = \frac{1}{Z(eventType)} \exp\left[\sum_{(i,j,eventType)\in\mathcal{Y}_{eventType}} score(i,j,eventType) + \sum_{[i-1,i] \text{ not covered in } \mathcal{Y}_{eventType}} score_\epsilon(i-1,i,eventType)\right], \quad (1)$$

Input audio

Scores how likely [i-1, i] is **not covered** by any event of *eventType.*

Training/Inference is quadratic in complexity. However, we show that it is still a fast solution for event-based prediction. See paper for details.

System Overview

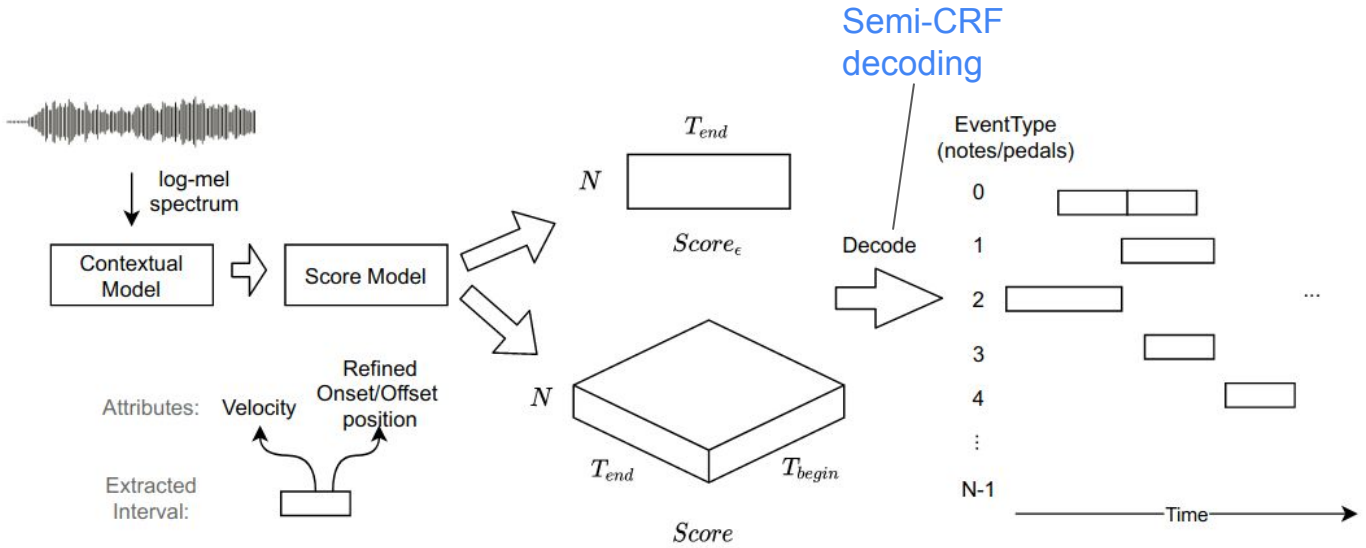


Figure 1: Proposed system overview. For the middle part of the figure, T_{begin} and T_{end} are the number of beginning positions and ending positions, respectively, and N is the number of *eventType*(s).

Experiments: Transcription Performance

	Activation			Note Onset			Note w/ Offset			Note w/ Offset & Vel.		
Method	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Hawthorne et al. [2019]	86.84	89.24	87.82	97.88	92.26	94.93	82.09	77.44	79.65	78.37	73.94	76.05
Kong et al. [2020]	90.09	**90.42**	90.15	98.16	**95.46**	**96.77**	85.65	83.32	84.45	84.18	81.92	83.02
Proposed	**93.84**	88.48	**90.98**	**98.78**	94.18	96.39	**90.79**	**86.62**	**88.63**	**89.78**	**85.68**	**87.65**

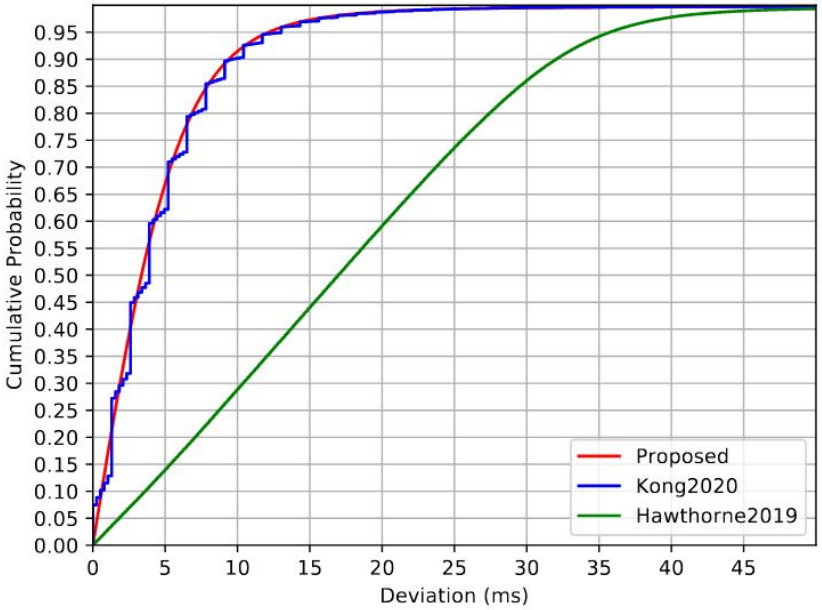
Table 1: Piano transcription note results for the proposed methods and various related works.

	Activation			Onset			Onset & Offset		
Method	P	R	F_1	P	R	F_1	P	R	F_1
Kong et al. [2020]	94.14	**94.29**	**94.11**	77.43	**78.19**	77.71	73.56	**74.21**	73.81
Proposed	**95.13**	87.71	90.73	**82.14**	74.91	**78.10**	**78.48**	71.72	**74.71**

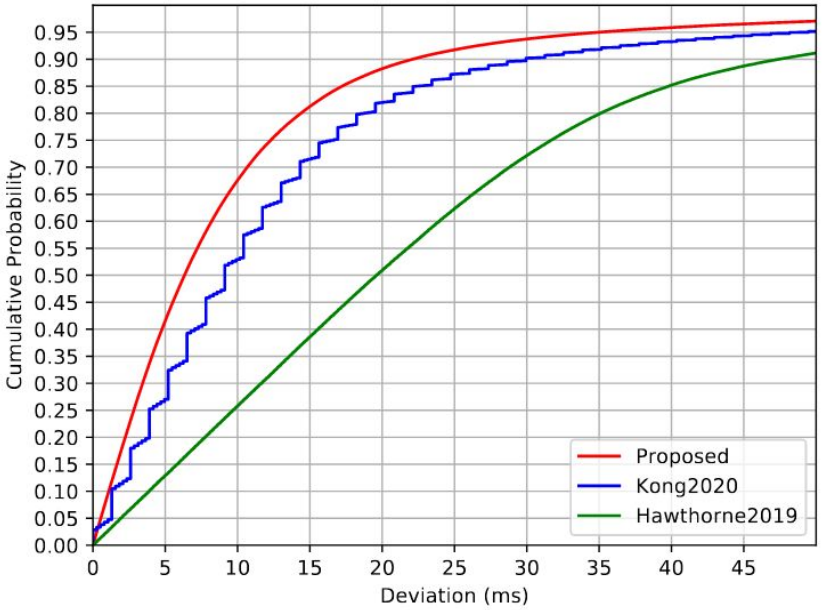
Table 2: Sustain pedal detection results for the proposed methods and various related works.

See paper for additional results.

Experiments: Timing Precision of Matched Notes



(a) Onset time deviation distribution.



(b) Offset time deviation distribution.

Figure 3: Empirical cumulative distribution functions of time deviations of estimated onsets and offsets from ground-truth notes.

Experiments: Running time

With Intel(R) i7-7800X@3.5GHz and Nvidia 1080TI, on *Carl Czeny Grand Sonata Op.145 No.9*:

System	Running time
Kong et al. [2020]	353s
Proposed	95s

Table 4: Running time for transcribing the same 33.3 minutes audio file.

Thank you!