Audio-to-Score Singing Transcription Based on Joint Estimation of Pitches, Onsets, and Metrical Positions With Tatum-Level CTC Loss

Tengyu Deng, Eita Nakamura, and Kazuyoshi Yoshii
Graduate School of Informatics, Kyoto University, Japan
E-mail: deng@sap.ist.i.kyoto-u.ac.jp, eita.nakamura@i.kyoto-u.ac.jp, yoshii@i.kyoto-u.ac.jp

***Abstract*—This paper describes an end-to-end singing transcription method that directly estimates a musical score (a sequence of sung notes with metrical positions) from a music audio signal. The monotonicity of audio-to-score mapping naturally calls for the use of connectionist temporal classification (CTC). Inspired by the success of character-level automatic speech recognition, previous studies on CTC-based music transcription represent a musical score as a sequence of various kinds of symbols (*e.g.*, note pitches and values and barlines) defined in some music notation. Such a naive notation-respecting representation, however, does not fit the non-overlapping monotonic audio-to-symbol alignment and the positions of barlines and the durations of beats tend to be incoherent in the estimated score. To solve this problem, we propose a tatum-level singing transcription method that jointly estimates the pitch (including rest), onset flag, and metrical position at each tatum. Our approach enables the tatums to be monotonically aligned with regularly-spanned intervals of the music signal and the estimated notes are located on the estimated metrical positions that are encouraged to be periodic. Experimental results clearly showed that our proposed model reached comparable accuracy on score-level singing transcription with only unaligned training data, and the proposed tatum-level representation significantly improved the stability of the metrical structures in the estimated scores.**

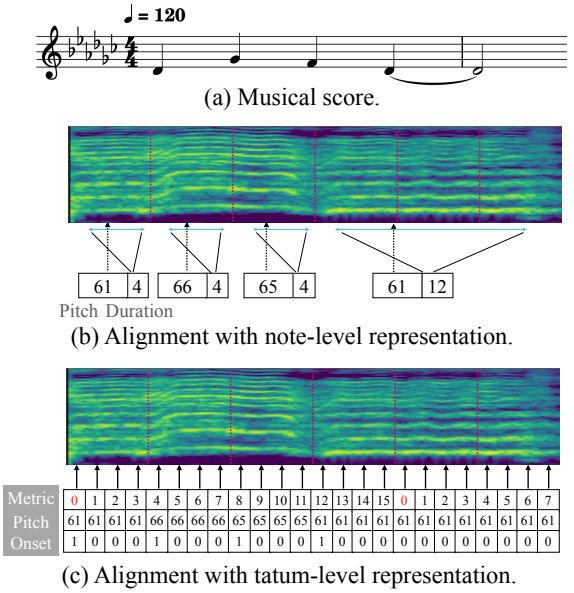


Fig. 1. Comparison of (b) conventional and (c) proposed methods of CTC-based audio-to-score transcription.

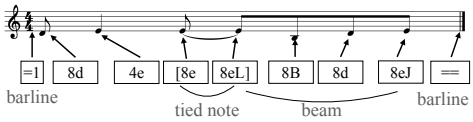
I. Introduction

Automatic singing transcription (AST) is one of the most fundamental music analysis tasks in the field of music information retrieval (MIR). Its ultimate goal is to estimate a *musical score* of the vocal part (described in the MusicXML format) from a music audio signal. The estimated scores can be used for melody-based music search (*e.g.*, query-by-humming [1], [2]), score-guided singing voice separation [3], [4], and interpretable emotion recognition [5].

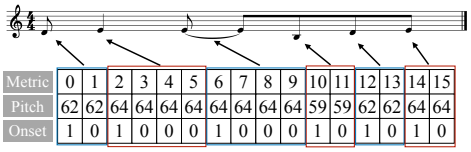
Most studies on AST have attempted to estimate a *piano roll* of the vocal part (described in the MIDI format) [6]–[10], where each sung note is represented by a semitone-level pitch and onset and offset times in seconds. The basic strategy is to estimate a continuous contour of fundamental frequencies (F0s) in Hz (*a.k.a.*, melody extraction) with a deep neural network (DNN) and then segment it to a series of sung and rest notes (discontinuous step function) with a note tracking method [7], [8]. Note that the piano roll differs from the musical score in the time units of the note onsets and offsets. The onsets and offsets are represented in frames or seconds in the piano roll representation, whereas they are represented in quantized units such as beats or fractions of beats in the score representation. To obtain a musical score from a piano roll, rhythm transcription methods [11] are required for quantizing the onset and offset times of notes onto a grid represented in musical metrics such as beats or tatums.

Only a few studies have tackled audio-to-score transcription [12], [13], which typically involve audio-to-MIDI transcription and MIDI-to-audio transcription. Such a cascading approach, however, suffers from several problems. Since F0 annotations aligned with music audio signals are limited both in size and variation, the accuracy of supervised F0 estimation is hard to improve. The complicated dynamics of the F0 contour (*e.g.*, vibrato and glissando) make it hard to perform frequency and temporal quantization without referring to the audio data [12]. This calls for the end-to-end (*i.e.*, audio-to-score) approach to AST that can make effective use of non-aligned pairs of musical scores and audio signals.

Considering the monotonicity of audio-to-score alignment, connectionist temporal classification (CTC) [14] has been used for audio-to-score automatic music transcription (AMT) [15],



(a) The Kern representation.



(b) The proposed tatum-level representation.

Fig. 2. Comparison of score representations.

[16] (Fig. 1). In the analogy with CTC-based automatic speech recognition (ASR) that converts a speech signal to a sequence of characters [17], one may try to convert a music signal to a sequence of various kinds of symbols (*e.g.*, note pitches, note values, and barlines) defined in some music notation (*e.g.*, Lilypond [18] and Kern [19], Fig. 2 (a)). It is, however, unclear how such instantaneous and temporal symbols are monotonically aligned with the audio signal in a non-overlapping manner (Fig. 1 (b)). Note that the basic CTC is intended for instantaneous symbols (*e.g.*, phonemes and note pitches) that can be estimated in local regions. To estimate temporal symbols (*e.g.*, note values and barlines), acoustic features should be aggregated from a nonlocal region. The mixture of these modes harms the stability of CTC-based training, leading to incoherent metrical structure in the estimated score. Indeed, previous CTC-based methods were tested only on synthetic data and evaluations on real audio data have not been reported in the literature.

To solve this problem, we propose a CTC-based AST method that converts a music audio signal to a tatum-level score representation with a metrical structure. The tatum is a minimum time unit on a score, which is assumed to be the sixteenth note length in this study. Representing a musical score as a sequence of tatums, each having a pitch (MIDI note number or 128 (rest)), an onset flag ($\in [0, 1]$), and a metrical position ($\in [0, 15]$) (Fig. 2 (b)), we can achieve stable monotonic audio-to-tatum alignment (Fig. 1 (c)). Specifically, a convolutional recurrent neural network (CRNN) that predicts the probabilities of pitches, onset flag, and metrical positions from a music signal at the frame level is trained such that the CTC loss is minimized. At run-time, the most likely tatum-level sequence is estimated with a Viterbi algorithm that explicitly considers the durations of sung notes and the metrical structure.

II. Related Work

Conventional AST approaches combine frame-level F0 estimation and note tracking, where the piano-roll representation is a transcription target [20], [21]. Recently, deep learning techniques have been used for F0 extraction tasks, remarkably improving the estimation accuracy [6]–[8]. A hidden semi-Markov model (HSMM) [12] was employed to quantize F0 contours into sequences of musical notes. Attempts to directly transcribe the musical signals into note events have also be made [9], using deep neural networks to overcome the error accumulation problem in cascading methods. To improve the accuracy of piano-roll transcription, the CTC loss was used as an auxiliary loss function in addition to the standard frame-wise cross-entropy loss [10].

In the context of AST that aims to estimate musical scores, rhythm transcription methods have been explored to quantize piano-roll representations into musical scores [11], [22], [23]. A hybrid DNN-HSMM model [13], for example, was proposed to avoid depending on the preprocessing step of F0 estimation, in the same way as the DNN-HMM approach to ASR. In these studies, representations of rhythmic information for music transcription have been studied. In general, it is possible to represent rhythmic information either at the tatum level using metrical positions or at the note level using note values. Studies using HMMs have shown that the tatum-level representation generally performs better than the note-level representation because certain logical structures of musical rhythms cannot be represented in the latter [23].

Several recent studies have attempted end-to-end AMT to address the error accumulation issue in cascading methods. Inspired by the success of end-to-end ASR with either the attention mechanisms or CTC, studies on end-to-end AMT have employed both attention mechanism [24] and CTC [15], [16] to directly transcribe audio signals into musical scores. To facilitate sequence-to-sequence learning in these methods, it is necessary to represent a musical score as a sequence of symbols. Previous CTC-based end-to-end AMT methods [15], [16] used a *note-level* score representation, where quantized durations of notes were directly estimated as output symbols. However, these methods have been shown to work only on synthetic audio signals made from MIDI-like data. Besides, an encoder-decoder model with an attention mechanism and a *tatum-level* score representation was also employed [24]. This study proposed a regularized training method that encourages the attention matrix to have monotonic and regular structures, aligning the tatums with the frames in a beat-synchronous manner. However, this approach often failed to find correct alignments because the attention matrix does not necessarily be monotonic in early epochs of non-regularized training.

III. Proposed Method

This section describes the proposed CTC-based end-to-end AST method that considers metrical structure.

A. Problem Specification

Given a music audio signal as input, the goal is to estimate a musical score of sung notes. Let $\mathbf{X} \triangleq \{\mathbf{x}_t\}_{t=1}^{T}$ be a frame-level sequence of acoustic features obtained by stacking the mel-spectrogram of the music signal and that of the singing voice separated with a source separation method called Open-Unmix [25], where T is the number of frames. The input to our system is $\mathbf{X}$.

In this paper, the music signal is assumed to have a time signature of 4/4, and the duration of the sixteenth note is regarded

as the minimum time unit called a tatum, *i.e.*, each measure includes 16 metrical positions (tatums). Let $\mathbf{Y} \triangleq \{y_n\}_{n=1}^N$ be a tatum-level score representation, where N is the number of tatums and $y_n \triangleq (b_n, p_n, o_n)$ is a triplet of a metrical position $b_n \in [0, 15]$ (relative position in the measure), a semitone-level pitch $p_n \in [0, 128]$ represented by a MIDI note number ($p_n = 128$ indicates the rest), and an onset flag $o_n \in \{0, 1\}$. To reconstruct a musical score from $\mathbf{Y}$, we first determine the positions of barlines from $\mathbf{B} \triangleq \{b_n\}_{n=1}^N$ under the assumption of the time signature and then the note pitches and values are determined from $\mathbf{P} \triangleq \{p_n\}_{n=1}^N$ and $\mathbf{O} \triangleq \{o_n\}_{n=1}^N$.

B. End-to-End Training

In the training phase, given paired data of the input $\mathbf{X}$ and the ground-truth output $\mathbf{Y} \triangleq (\mathbf{B}, \mathbf{P}, \mathbf{O})$, we train a DNN with the CTC loss in a supervised manner. The input length is assumed to be longer than the output length ($T > N$), which always holds in AST.

1) Single-Label CTC: We first explain the standard CTC. Suppose we aim to estimate only $\mathbf{B}$ from $\mathbf{X}$ ($\mathbf{P}$ and $\mathbf{O}$ are not considered here). Let K be the number of kinds of original output symbols (metrical positions) (*i.e.*, $K = 16$). The special blank symbol "*" working as a wild card is introduced.

A DNN is trained such that the posterior probability of the *tatum-level* sequence $\mathbf{B}$ given as the ground-truth data, denoted by $p(\mathbf{B}|\mathbf{X})$, is maximized. Let $\boldsymbol{\pi} \triangleq \{\pi_t\}_{t=1}^T$ be a *frame-level* sequence such that $\mathbf{B} = \mathcal{M}(\boldsymbol{\pi})$, where $\pi_t \in [0, K-1] \cup \{*\}$ represents the output symbol at frame t, and $\mathcal{M}(\boldsymbol{\pi})$ is a reduction operator that annexes repeated symbols and removes all blank symbols from $\boldsymbol{\pi}$. If $\mathbf{B} = \{1, 2, 3\}$, for example, $\{1, 1, 1, *, *, 2, *, *, 3, *\}$ is a possible example of $\boldsymbol{\pi}$. The DNN is used for inferring metrical positions in frames as follows:

$$(\boldsymbol{\phi}^{\mathrm{B}}, \boldsymbol{\phi}^{*}) = \mathrm{DNN}(\mathbf{X}), \tag{1}$$

where $\boldsymbol{\phi}^{\mathrm{B}} \triangleq \{\phi_{tk}^{\mathrm{B}}\}_{t=1,k=1}^{T,K-1}$, $\boldsymbol{\phi}^{*} \triangleq \{\phi_t^{*}\}_{t=1}^{T}$, ϕ_{tk}^{B} is the posterior probability of metrical position k when $\pi_t \neq *$, and ϕ_t^{*} is that of $\pi_t = *$ at frame t.

$p(\mathbf{B}|\mathbf{X})$ is obtained by accumulating the posterior probabilities over all possible $\boldsymbol{\pi}$'s (frame-to-tatum alignments) that can be reduced to $\mathbf{B}$ as follows (Fig. 3 (a)):

$$p(\mathbf{B}|\mathbf{X}) = \sum_{\boldsymbol{\pi} \in \mathcal{M}^{-1}(\mathbf{B})} p(\boldsymbol{\pi}|\mathbf{X}), \tag{2}$$

where $\mathcal{M}^{-1}(\mathbf{B})$ is an expansion operator that maps $\mathbf{B}$ to a frame-level sequence and $p(\boldsymbol{\pi}|\mathbf{X})$ is the posterior probability of a sequence $\boldsymbol{\pi}$. Assuming the conditional independence over frames, $p(\boldsymbol{\pi}|\mathbf{X})$ is given by

$$p(\boldsymbol{\pi}|\mathbf{X}) = \prod_{t=1}^{T} p(\pi_t|\mathbf{X}), \tag{3}$$

where $p(\pi_t|\mathbf{X})$ is the local posterior probability given by

$$p(\pi_t|\mathbf{X}) = \begin{cases} (1 - \phi_t^{*})\phi_{t\pi_t}^{\mathrm{B}} & \text{if} \quad \pi_t \neq *, \\ \phi_t^{*} & \text{otherwise.} \end{cases} \tag{4}$$

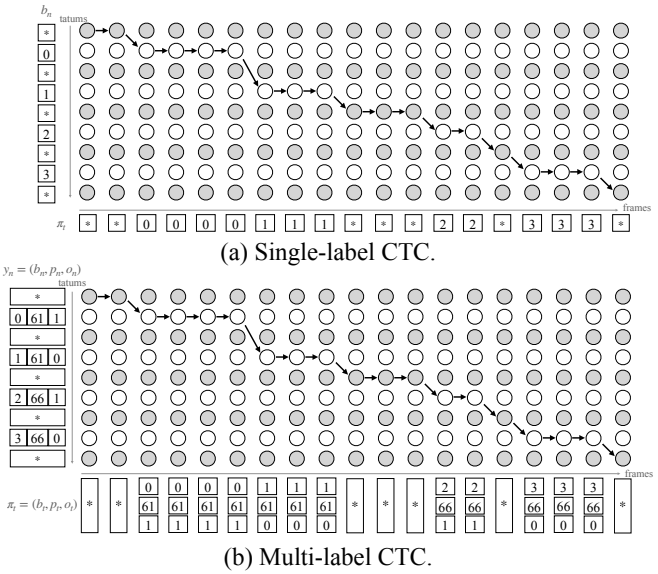


Fig. 3. Possible alignment paths in CTC.

2) Multi-Label CTC: In our study on joint pitch, onset, and metrical estimation, we use the multi-label CTC (MCTC) loss [26]. Let I be the size of the pitch vocabulary ($I = 129$).

A DNN is trained such that the posterior probability $p(\mathbf{Y}|\mathbf{X})$ of the *tatum-level* sequence $\mathbf{Y}$ given as the ground-truth data is maximized. Let $\boldsymbol{\pi} \triangleq \{\pi_t\}_{t=1}^T$ be a *frame-level* sequence (alignment path) such that $\mathbf{Y} = \mathcal{M}(\boldsymbol{\pi})$, where $\pi_t \triangleq (b_t, p_t, o_t) \in \{(k, i, \{0, 1\})\}_{k=1,i=1}^{K-1,I-1} \cup \{*\}$ denotes the output symbol (triplet or $*$) at frame t, and $\mathcal{M}(\boldsymbol{\pi})$ is a reduction operator. The DNN is used for jointly inferring metrical positions, note pitches, and onset flags at the frame level as follows:

$$(\boldsymbol{\phi}^{\mathrm{B}}, \boldsymbol{\phi}^{\mathrm{P}}, \boldsymbol{\phi}^{\mathrm{O}}, \boldsymbol{\phi}^{*}) = \mathrm{DNN}(\mathbf{X}), \tag{5}$$

where $\boldsymbol{\phi}^{\mathrm{P}} \triangleq \{\phi_{ti}^{\mathrm{P}}\}_{t=1,i=1}^{T,I-1}$, $\boldsymbol{\phi}^{\mathrm{O}} \triangleq \{\phi_t^{\mathrm{O}}\}_{t=1}^{T}$, ϕ_{ti}^{P} is the posterior probability of pitch i when $\pi_t \neq *$, and ϕ_t^{O} is that of the onset presence at frame t.

In the same ways as Eq. (2), $p(\mathbf{Y}|\mathbf{X})$ can be computed with dynamic programming (DP) as follows:

$$p(\mathbf{Y}|\mathbf{X}) = \sum_{\boldsymbol{\pi} \in \mathcal{M}^{-1}(\mathbf{Y})} p(\boldsymbol{\pi}|\mathbf{X}), \tag{6}$$

where $\mathcal{M}^{-1}(\mathbf{Y})$ is an expansion operator. Assuming the conditional independence over frames and labels, the posterior probability $p(\boldsymbol{\pi}|\mathbf{X})$ can be computed in the same ways as Eq. (3), where $p(\pi_t|\mathbf{X})$ is given by

$$p(\pi_t|\mathbf{X}) = \begin{cases} (1 - \phi_t^{*})\phi_{tb_t}^{\mathrm{B}}\phi_{tp_t}^{\mathrm{P}}(\phi_t^{\mathrm{O}})^{o_t}(1 - \phi_t^{\mathrm{O}})^{1-o_t} & \text{if} \quad \pi_t \neq *, \\ \phi_t^{*} & \text{otherwise.} \end{cases} \tag{7}$$

C. Network Architecture

The DNN used in this study is based on an encoder-decoder architecture (Fig. 4). Since wide-band harmonic structures and long-term temporal structures are considered to be useful clues for pitch estimation and metrical analysis, respectively, two

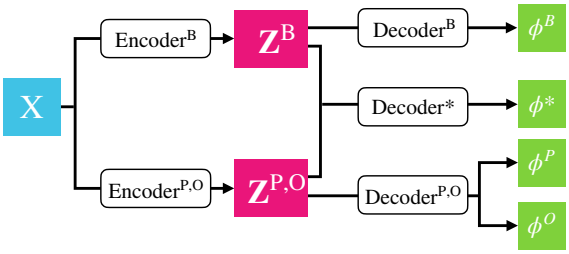


Fig. 4. Network architecture.

separate encoders are used to extract latent metrical features $\mathbf{Z}^{\mathrm{B}}$ and latent pitch features $\mathbf{Z}^{\mathrm{P,O}}$ as follows:

$$\mathbf{Z}^{\mathrm{B}} = \mathrm{Encoder}^{\mathrm{B}}(\mathbf{X}), \tag{8}$$

$$\mathbf{Z}^{\mathrm{P,O}} = \mathrm{Encoder}^{\mathrm{P,O}}(\mathbf{X}). \tag{9}$$

To estimate the pitch and onset presence at each frame, it is sufficient to focus on the local region of the singing voice around the frame. To estimate the metrical position, in contrast, it is necessary to collect long-term periodic information over multiple measures from the background music.

We then compute the posterior probabilities of the metrical positions, pitches, onset presence, and blank symbol, denoted by $\boldsymbol{\phi}^{\mathrm{B}}$, $\boldsymbol{\phi}^{\mathrm{P}}$, $\boldsymbol{\phi}^{\mathrm{O}}$, and $\boldsymbol{\phi}^{*}$, as follows:

$$\boldsymbol{\phi}^{\mathrm{B}} = \mathrm{Decoder}^{\mathrm{B}}(\mathbf{Z}^{\mathrm{B}}), \tag{10}$$

$$(\boldsymbol{\phi}^{\mathrm{P}}, \boldsymbol{\phi}^{\mathrm{O}}) = \mathrm{Decoder}^{\mathrm{P,O}}(\mathbf{Z}^{\mathrm{P,O}}), \tag{11}$$

$$\boldsymbol{\phi}^{*} = \mathrm{Decoder}^{*}(\mathbf{Z}^{\mathrm{B}}, \mathbf{Z}^{\mathrm{P,O}}). \tag{12}$$

Since the blank symbol tends to be selected at frames with small confidence about the pitch, onset, or metrical estimation, all the latent features are considered.

D. Decoding

After the CTC-based training, one can estimate the frame- and tatum-level sequences $\hat{\boldsymbol{\pi}}$ and $\hat{\mathbf{Y}}$ as follows:

$$\hat{\boldsymbol{\pi}} = \arg\max_{\boldsymbol{\pi}} p(\boldsymbol{\pi}|\mathbf{X}), \tag{13}$$

$$\hat{\mathbf{Y}} = \mathcal{M}(\hat{\boldsymbol{\pi}}). \tag{14}$$

According to this method, however, the estimated metrical positions may not have a monotonically increasing order.

To impose the constraint of monotonically increasing metrical positions, we propose a refined decoding method based on an HSMM. Generally in music performances including singing, the durations of beats or tatums keep stable because of the regularity of metrical structure in music. To utilize this temporal stability, we introduce the durations of tatums $\mathbf{D} \triangleq \{\mathbf{d}_n\}_{n=1}^{N}$ measured in frames. Consider the frame-level input features and tatum-level output features described in Section. III-A, namely, input spectrograms $\mathbf{X} \triangleq \{\mathbf{x}_t\}_{t=1}^{T}$, and output metrical positions $\mathbf{B} \triangleq \{b_n\}_{n=1}^{N}$, pitches $\mathbf{P} \triangleq \{p_n\}_{n=1}^{N}$ and onset flags $\mathbf{O} \triangleq \{o_n\}_{n=1}^{N}$. We formulate the generative process of $\mathbf{B}, \mathbf{P}, \mathbf{O}, \mathbf{D}$ and $\mathbf{X}$ as an HSMM, which only allows the metrical positions to make periodic progress, that is,

$$b_n = \begin{cases} b_{n-1} + 1 & b_{n-1} < 15, \\ 0 & b_{n-1} = 15, \end{cases} \tag{15}$$

and encourages the durations of metrical positions to be stable by assuming

$$p(d_n|d_{n-1}) \propto \exp\left(-\lambda \left| \frac{d_n}{d_{n-1}} - 1 \right| \right). \tag{16}$$

With the HSMM, one can efficiently decode the sequence of metrical positions, pitch values and onset flags with the Viterbi algorithm (see details in Appendix A).

IV. Evaluation

This section reports experiments conducted for evaluating the effectiveness of the proposed method.

A. Experimental Conditions

1) Dataset: We made a dataset consisting of 343 Japanese popular (JPOP) songs with the corresponding musical scores in the MusicXML format transcribed by experts and real audio recordings extracted from the original CDs. It was randomly split into a training set of 308 songs and a test set of 35 songs. Although this JPOP dataset is useful for evaluating the practical performance, it cannot be released due to copyright issues. We thus also used the publicly available RWC Popular Music dataset [27] for evaluation. In detail, we randomly selected 12 songs with the signature of 4/4 to form a test dataset (Nos. 7, 8, 13, 18, 20, 47, 63, 79, 80, 84, 90, and 100). The vocal scores of these songs were transcribed by experts in the MusicXML format as well.

Note that most common datasets (*e.g.*, MIR-ST500 [28]) consist of singing voices with annotations in the MIDI format, where the onset and offset times of each note are given in seconds. These datasets have thus been used for audio-to-MIDI AST, but are hard-to-use for audio-to-score AST (ours) due to the lack of ground-truth musical scores.

To reduce the computational cost, all the audio recordings were resampled at 22050 Hz and cut into segments of 8 seconds. To find the corresponding segments of scores, we synthesized waveforms from the scores and aligned them with the original signals with dynamic time warping (DTW). These alignment results were then used to segment the ground truth scores. The spectrogram of a music signal was obtained with short-time Fourier transform (STFT) with a window of 2048 samples and a hop length of 256 samples (11.61 ms). The mel spectrogram was then obtained with 128 mel filter banks.

2) Network Configuration: The CRNN used for evaluation was configured as follows. $\mathrm{Encoder}^{\mathrm{B}}$ and $\mathrm{Encoder}^{\mathrm{P,O}}$ were implemented with CNNs of the same structure. Each CNN had five layers and the numbers of channels for these layers were 64, 32, 32, 32, and 32, respectively. The kernel sizes for the hidden layers were 5, 5, 3, 3, and 3, respectively. Following the CNN, a fully-connected layer was used for reducing the hidden dimension to 256 at each frame. $\mathrm{Decoder}^{\mathrm{B}}$, $\mathrm{Decoder}^{\mathrm{P,O}}$ and $\mathrm{Decoder}^{*}$ were implemented with RNNs. Each RNN had two layers of BLSTMs, and the number of channels in the hidden layers was all 256. The CRNN was trained by an Adam optimizer with a learning rate of 1×10^{-4} for 20 epochs with the training set.

TABLE I
COMPARISON OF EDIT-DISTANCE-BASED ERROR RATES (%) OF DIFFERENT METHODS.

Test Dataset	Method	Pitch Error	Miss Rate	Extra Rate	Onset Error	Offset Error	Mean Error
JPOP-test	VOCANO [9] + quantization [22]	9.0	47.8	14.2	47.4	31.2	29.9
	Note-level CTC (baseline)	**6.9**	24.6	**7.5**	37.3	27.3	**20.7**
	Tatum-level CTC (proposed)	8.8	**23.4**	14.1	**35.1**	**26.4**	21.6
RWC-test	VOCANO [9] + quantization [22]	12.7	16.5	15.6	40.6	29.5	23.0
	Note-level CTC (baseline)	**9.1**	**12.1**	**11.7**	35.5	**24.0**	**18.5**
	Tatum-level CTC (proposed)	9.4	13.3	18.9	**32.3**	26.9	20.2

TABLE II
COMPARISON OF BEAT AND DOWNBEAT F SCORES (%) FOR DIFFERENT METHODS.

Test Dataset	Method	Beat			Downbeat		
		Precision	Recall	F score	Precision	Recall	F score
JPOP-test	VOCANO [9] + quantization [22]	53.3	42.8	45.7	15.3	15.7	14.8
	Note-level CTC (baseline)	50.7	28.0	34.6	13.0	6.9	8.6
	Tatum-level CTC (proposed)	**80.5**	**73.6**	**75.2**	**67.9**	**68.8**	**67.3**
RWC-test	VOCANO [9] + quantization [22]	39.5	39.3	39.0	24.7	25.2	24.6
	Note-level CTC (baseline)	41.0	26.7	31.5	15.4	8.1	10.37
	Tatum-level CTC (proposed)	**77.0**	**82.0**	**79.3**	**66.5**	**70.2**	**68.0**

The HSMM used in the decoding phase was configured as follows. The hyperparameter λ in the transition model was set to 100, the minimum and maximum pitch values were $p_{min} = 43$, $p_{max} = 79$, and the minimum and maximum tatum durations were $d_{min} = 6$, $d_{max} = 30$, corresponding to tempos from 215 bpm to 43 bpm.

B. Compared Methods

To evaluate the efficiency of the proposed method, we compared it with VOCANO [9], a singing transcription system which combined pitch extraction and note segmentation. The original training data used for VOCANO have no overlap with the test data in our experiments. Since VOCANO was designed for estimating a pitch contour from singing voice, we processed the frame-based result obtained by VOCANO with a rhythm quantization method based on an HMM [22].

For comparison, we configured a baseline model based on a note-level score representation that estimates the note value of each note instead of the metrical position. This model was similar to ones used in the literature [15], [16] and was almost the same as the proposed method, except that each output symbol was given by $y_n \triangleq (d_n, p_n, o_n)$, where p_n and o_n are the same as ones described in Section III-A and $d_n \in [1, 16]$ represents the note duration. Note that only the duration within a measure was considered, and notes longer than a measure were split into different symbols. For example, a note with the pitch value of 64 and a duration of 18 tatums was represented as $(16, 64, 1), (2, 64, 0)$. The note-level baseline model was trained with the same procedure and the same dataset as the proposed CTC-based method and decoded with the greedy algorithm as Eqs. 13,14.

C. Evaluation Measures

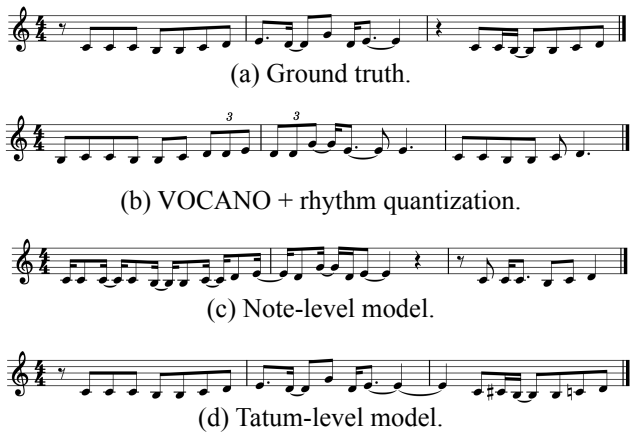
We evaluated the estimated results with edit-distance-based metrics [29] in a way similar to the word error rate used for ASR. These metrics were computed by performing alignment between the estimated notes and ground-truth notes with DP and then identifying five exclusive types of errors: pitch errors, missing notes, extra notes, onset time errors, and offset time errors. The onset time and offset time errors are defined in terms of score times in units of tatums, but they do not take into account metrical positions. To evaluate the metrical structure of estimated scores, we additionally calculated the precision, recall, and F-score by inspecting the beat and downbeat positions of the correctly aligned notes.

D. Experimental Results

Tables I and II show the evaluation results. The multi-step combination of VOCANO and rhythm quantization had a low overall accuracy in terms of both the edit-distance-based metrics and the beat and downbeat F scores. This shows the difficulty of quantizing the estimated frame-level F0 contours. The CTC-based method with the note-level representation and its variant with the tatum-level representation had similar mean error rates in Table I, and the latter had higher beat and downbeat F scores (Table II). This shows the effectiveness of the end-to-end approach to AST based on CTC with the tatum-level representation.

Fig. 5 shows examples of musical scores estimated for the same piece of music from the JPOP-test dataset. As shown in Fig. 5(b), the score estimated by VOCANO contained extra notes and missing notes as well as a number of rhythmic errors. The score estimated by the baseline model with the note-level representation had mostly correct pitches and note values, but suffered from severe misalignment of metrical positions. Since the note-level representation does not consider beat and downbeat positions, rhythmic errors were accumulated when estimated notes were concatenated together. Compared with the baseline model, the proposed tatum-level model significantly reduced errors in metrical positions while keeping errors in pitches and note values at the same level.

Fig. 6 shows the histograms of the durations of the estimated notes obtained by the note-level model and the tatum-level model on the JPOP-test dataset. We found that the CTC-based



(a) Ground truth.

(b) VOCANO + rhythm quantization.

(c) Note-level model.

(d) Tatum-level model.

Fig. 5. Examples of estimated scores.

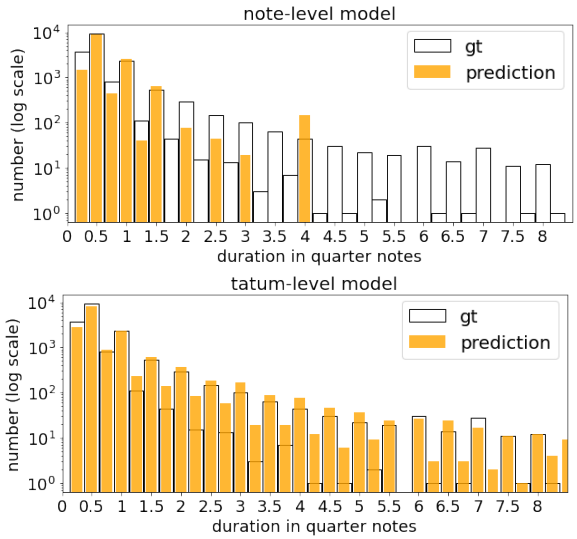


Fig. 6. Histograms of the note values estimated by the note-level (above) and tatum-level (below) CTCs, where "gt" stands for the ground truth data.

model with the note-level representation failed to correctly predict note values longer than a half note. Although the method was designed for dealing with note values longer than a whole note, the model failed to learn this property. With the proposed tatum-level representation, the histogram of the estimated note values was similar to that of the ground truth data. The proposed representation increased the rate of correctly estimating note values and contributed to correctly predicting metrical positions.

V. Conclusion

In this paper, we proposed an end-to-end AST method based on the CTC-based training with a tatum-level score representation consisting of positions of the tatums within measures, note pitches, and onset flags. The experimental results showed that the proposed method achieved better accuracy on the estimation of metrical positions of notes, and a mean edit-distance-based error rate similar to the CTC-based baseline method with a note-level score representation. We also found that the cascade of F0 extraction and rhythm quantization yielded a significantly worse result.

There are several issues left for future work. First, we assumed that the minimum time unit of a score is the sixteenth note, and we thus need to extend the model for fine-grained time units. Second, the proposed method can only transcribe songs with the time signature of 4/4, and other common meters should be properly dealt with. Additionally, popular music can have multiple meters within a song, and such variable meters should also be considered. Finally, in the decoding with the HSMM, a note-level musical language model is expected to be effective to improve the result, as in the case for ASR.

Acknowledgment

This study was partially supported by JSPS KAKENHI Nos. 21K12187, 21K02846, 22H03661, 20H00602, and 21H03572, JST PRESTO No. JPMJPR20CB, and JST FOREST No. JPMJPR226X. We thank Ryo Nishikimi for the constructive discussion with him on this study.

Appendix

A. HSMM-Based Decoding

1) Formulation of the HSMM: To impose the constraint of monotonically increasing metrical positions, we propose a refined decoding method based on an HSMM. Generally in music performances including singing, the durations of beats or tatums keep stable because of the regularity of metrical structure in music. To utilize this temporal stability, we introduce the durations of tatums $\mathbf{D} \triangleq \{\mathbf{d}_n\}_{n=1}^N$ measured in frames in the proposed HSMM. Consider the frame-level input features and tatum-level output features described in Section. III-A, namely, input spectrograms $\mathbf{X} \triangleq \{\mathbf{x}_t\}_{t=1}^T$, and output metrical positions $\mathbf{B} \triangleq \{b_n\}_{n=1}^N$, pitches $\mathbf{P} \triangleq \{p_n\}_{n=1}^N$ and onset flags $\mathbf{O} \triangleq \{o_n\}_{n=1}^N$. We formulate the generative process of $\mathbf{B}, \mathbf{P}, \mathbf{O}, \mathbf{D}$ and $\mathbf{X}$ as

$$p(\mathbf{X}, \mathbf{B}, \mathbf{P}, \mathbf{O}, \mathbf{D}) = p(\mathbf{X}|\mathbf{B}, \mathbf{P}, \mathbf{O}) p(\mathbf{B}, \mathbf{P}, \mathbf{O}, \mathbf{D}). \quad (17)$$

In Eq. 17, $p(\mathbf{B}, \mathbf{P}, \mathbf{O}, \mathbf{D})$ represents the language model of the hidden variables. We assume the distributions of pitches $\mathbf{P}$ and onsets $\mathbf{O}$ to be uniform, and formulate the transition probabilities as

$$p(\mathbf{B}, \mathbf{P}, \mathbf{O}, \mathbf{D}) = p(b_1, p_1, o_1, d_1) \times \prod_{n=2}^{N} p(b_n, p_n, o_n, d_n | b_{n-1}, p_{n-1}, o_{n-1}, d_{n-1}). \quad (18)$$

We assume the initial distribution of $\mathbf{B}$ and $\mathbf{D}$ is uniform, namely

$$p(b_1, p_1, o_1, d_1) = \frac{1}{K_b} \frac{1}{K_o} \frac{1}{p_{max} - p_{min} + 2} \frac{1}{d_{max} - d_{min} + 1}, \quad (19)$$

where $K_b = 16, K_o = 2$ are the number of distinct metrical positions and onset flags, p_{min} and p_{max} are assumed minimum and maximum pitch values other than the rest, and d_{min}

TABLE III
AN EXAMPLE OF THE FRAME-LEVEL VARIABLES.

Metrical Position $\bar{b}_t$	15	15	15	15	0	0	0	0	0	1	1	1	1	...
Tatum Duration $\bar{d}_t$	4	4	4	4	5	5	5	5	5	4	4	4	4	...
Tatum Counter $\bar{c}_t$	3	2	1	0	4	3	2	1	0	3	2	1	0	...
Pitch $\bar{p}_t$	62	62	62	62	62	62	62	62	62	59	59	59	59	...
Onset $\bar{o}_t$	1	1	1	1	0	0	0	0	0	1	1	1	1	...

and d_{max} are assumed minimum and maximum lengths of a tatum. The transition probabilities of b_n, d_n from b_{n-1}, d_{n-1} are determined by

$$\begin{aligned} &p(b_n, p_n, o_n, d_n | b_{n-1}, p_{n-1}, o_{n-1}, d_{n-1}) \\ &= \frac{1}{K_o} \frac{1}{p_{max} - p_{min} + 2} \delta_{16}(b_n, b_{n-1} + 1) p(d_n | d_{n-1}), \end{aligned} \quad (20)$$

where δ_{16} is the Kronecker's function modulo 16, that is

$$\delta_{16}(x, y) = \begin{cases} 1 & \text{if } x \equiv y \pmod{16}, \\ 0 & \text{if } x \not\equiv y \pmod{16}. \end{cases} \quad (21)$$

As suggested in [30], the transition probabilities of d_n from d_{n-1} are given by

$$p(d_n | d_{n-1}) \propto \exp\left(-\lambda \left| \frac{d_n}{d_{n-1}} - 1 \right| \right). \quad (22)$$

2) Frame-Level Formulation of the HSMM: The probabilistic model proposed in Section. A1 contains temporally asynchronous variables, $\mathbf{X}$ on the frame level and $\mathbf{B}, \mathbf{P}, \mathbf{O}, \mathbf{D}$ on the tatum level. To formulate the model in a synchronous way, we introduce the frame-synchronized versions of metrical positions $\bar{\mathbf{B}} \triangleq \{\bar{b}_t\}_{t=1}^T$, pitches $\bar{\mathbf{P}} \triangleq \{\bar{p}_t\}_{t=1}^T$, onset flags $\bar{\mathbf{O}} \triangleq \{\bar{o}_t\}_{t=1}^T$, and tatum durations $\bar{\mathbf{D}} \triangleq \{\bar{d}_t\}_{t=1}^T$. These variables can be constructed from the corresponding frame-level variables with $\bar{b}_t = b_n$, $\bar{p}_t = p_n$, $\bar{o}_t = o_n$ and $\bar{d}_t = d_n$ when $u(n-1) + 1 \le t \le u(n)$, where

$$u(n) \triangleq \begin{cases} 0 & n = 0, \\ \sum_{j=1}^n d_j & n > 0, \end{cases} \quad (23)$$

is the index of the last frame for each tatum. In addition, a tatum counter $\bar{\mathbf{C}} \triangleq \{\bar{c}_t\}_{t=1}^T$ is used to represent the position of the current frame within the tatum, counting down from $d_n - 1$ to 0. Table III shows an example of the frame-level variables.

With the tatum-level variables the HSMM can be rewritten as

$$\begin{aligned} p(\mathbf{X}, \mathbf{B}, \mathbf{P}, \mathbf{O}, \mathbf{D}) &= p(\mathbf{X}, \bar{\mathbf{B}}, \bar{\mathbf{P}}, \bar{\mathbf{O}}, \bar{\mathbf{D}}, \bar{\mathbf{C}}) \\ &= p(\mathbf{X} | \bar{\mathbf{B}}, \bar{\mathbf{P}}, \bar{\mathbf{O}}) p(\bar{\mathbf{B}}, \bar{\mathbf{P}}, \bar{\mathbf{O}}, \bar{\mathbf{D}}, \bar{\mathbf{C}}). \end{aligned} \quad (24)$$

In Eq. 17, $p(\bar{\mathbf{B}}, \bar{\mathbf{P}}, \bar{\mathbf{O}}, \bar{\mathbf{D}}, \bar{\mathbf{C}})$ represents the language model of the frame-level hidden variables. It is similar to Eq. 18, except that the transition probabilities become

$$\begin{aligned} &p(\bar{b}_t, \bar{p}_t, \bar{o}_t, \bar{d}_t, \bar{c}_t | \bar{b}_{t-1}, \bar{p}_{t-1}, \bar{o}_{t-1}, \bar{d}_{t-1}, \bar{c}_{t-1}) \\ &= \begin{cases} \delta(\bar{b}_t, \bar{b}_{t-1}) \delta(\bar{p}_t, \bar{p}_{t-1}) \delta(\bar{o}_t, \bar{o}_{t-1}) \delta(\bar{d}_t, \bar{d}_{t-1}) \delta(\bar{c}_t, \bar{c}_{t-1} - 1), \\ \bar{c}_{t-1} > 0, \\ \frac{1}{K_o} \frac{1}{p_{max} - p_{min} + 2} \delta_{16}(\bar{b}_t, \bar{b}_{t-1} + 1) p(\bar{d}_t | \bar{d}_{t-1}) \delta(\bar{c}_t, \bar{d}_t - 1), \\ \bar{c}_{t-1} = 0. \end{cases} \end{aligned} \quad (25)$$

In Eq. 24, $p(\mathbf{X} | \bar{\mathbf{B}}, \bar{\mathbf{P}}, \bar{\mathbf{O}})$ represents the emission probabilities of the observed spectrograms given the metrical positions, pitch values and onset flags. We assume the conditional independence of $\mathbf{X}$ given $\bar{\mathbf{B}}, \bar{\mathbf{P}}, \bar{\mathbf{O}}$, and we have

$$p(\mathbf{X} | \bar{\mathbf{B}}, \bar{\mathbf{P}}, \bar{\mathbf{O}}) = \prod_{t=1}^T p(\mathbf{x}_t | \bar{b}_t, \bar{p}_t, \bar{o}_t), \quad (26)$$

According to Bayes' theorem, the emission probability of a frame can be written as

$$\begin{aligned} p(\mathbf{x}_t | \bar{b}_t, \bar{p}_t, \bar{o}_t) &= \frac{p(\bar{b}_t, \bar{p}_t, \bar{o}_t | \mathbf{x}_t) p(\mathbf{x}_t)}{p(\bar{b}_t, \bar{p}_t, \bar{o}_t)} \\ &\propto \frac{p(\bar{b}_t, \bar{p}_t, \bar{o}_t | \mathbf{x}_t)}{p(\bar{b}_t, \bar{p}_t, \bar{o}_t)}. \end{aligned} \quad (27)$$

We assume that $p(\bar{b}_t, \bar{p}_t, \bar{o}_t)$ is a uniform distribution and that $p(\bar{b}_t, \bar{p}_t, \bar{o}_t | \mathbf{x}_t)$ can be estimated with a DNN trained in an end-to-end manner as described in Section III-B. Under the assumption of conditional independence of $\bar{b}_t, \bar{p}_t, \bar{o}_t$ given $\mathbf{x}_t$, we put that

$$\begin{aligned} p(\bar{b}_t, \bar{p}_t, \bar{o}_t | \mathbf{x}_t) &= p(\bar{b}_t, | \mathbf{x}_t) p(\bar{p}_t, | \mathbf{x}_t) p(\bar{o}_t, | \mathbf{x}_t), \\ &= \phi^{\mathrm{B}}_{t\bar{b}_t} \phi^{\mathrm{P}}_{t\bar{p}_t} (\phi^{\mathrm{O}}_t)^{\bar{o}_t} (1 - \phi^{\mathrm{O}}_t)^{1-\bar{o}_t}, \quad (28) \\ p(b_n | \mathbf{x}_t) &= \phi^{\mathrm{B}}_{t\bar{b}_t}, \quad (29) \\ p(p_n | \mathbf{x}_t) &= \phi^{\mathrm{P}}_{t\bar{p}_t}, \quad (30) \\ p(o_n | \mathbf{x}_t) &= (\phi^{\mathrm{O}}_t)^{\bar{o}_t} (1 - \phi^{\mathrm{O}}_t)^{1-\bar{o}_t}. \quad (31) \end{aligned}$$

It is notable that only the outputs $\phi^{\mathrm{B}}, \phi^{\mathrm{P}}$ and ϕ^{O} are used here since they can be interpreted as the posterior probabilities when the neural network judges a frame to be non-blank.

3) Viterbi Decoding: With the probabilistic model described above, we can estimate the optimal metrical positions $\mathbf{B}^*$, pitch values $\mathbf{P}^*$ and onset flags $\mathbf{O}^*$ by first estimating

$$\bar{\mathbf{B}}^*, \bar{\mathbf{P}}^*, \bar{\mathbf{O}}^*, \bar{\mathbf{D}}^*, \bar{\mathbf{C}}^* = \operatorname{argmax} p(\mathbf{X}, \bar{\mathbf{B}}, \bar{\mathbf{P}}, \bar{\mathbf{O}}, \bar{\mathbf{D}}, \bar{\mathbf{C}}), \quad (32)$$

and then obtain $\mathbf{B}^*, \mathbf{P}^*, \mathbf{O}^*$ from $\bar{\mathbf{B}}^*, \bar{\mathbf{P}}^*, \bar{\mathbf{O}}^*$. Eq. 32 can be solved efficiently with the Viterbi algorithm.

REFERENCES

[1] Akinori Ito, Yu Kosugi, Shozo Makino, and Masashi Ito. A query-by-humming music information retrieval from audio signals based on multiple F0 candidates. In *ICALIP*, pp. 1–5, 2010.

[2] Martín Rocamora, Pablo Cancela, and Alvaro Pardo. Query by humming: Automatically building the database from music recordings. *Pattern Recognition Letters*, Vol. 36, pp. 272–280, 2014.

[3] Zhiyao Duan and Bryan Pardo. Soundprism: An online system for score-informed source separation of music audio. *IEEE Journal of Selected Topics in Signal Processing*, Vol. 5, No. 6, pp. 1205–1215, 2011.

[4] Sebastian Ewert, Bryan Pardo, Meinard Muller, and Mark D. Plumbley. Score-informed source separation for musical audio recordings: An overview. *IEEE Signal Processing Magazine*, Vol. 31, No. 3, pp. 116–124, 2014.

[5] Shreyan Chowdhury, Andreu Vall Portabella, Verena Haunschmid, and Gerhard Widmer. Towards explainable music emotion recognition: The route via mid-level features. In *ISMIR*, pp. 237–243, 2019.

[6] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. Crepe: A convolutional representation for pitch estimation. In *ICASSP*, pp. 161–165, 2018.

[7] Tsung-Han Hsieh, Li Su, and Yi-Hsuan Yang. A streamlined encoder/decoder architecture for melody extraction. In *ICASSP*, pp. 156–160, 2019.

[8] Shuai Yu, Xiaoheng Sun, Yi Yu, and Wei Li. Frequency-temporal attention network for singing melody extraction. In *ICASSP*, pp. 251–255, 2021.

[9] Jui-Yang Hsu and Li Su. VOCANO: A note transcription framework for singing voice in polyphonic music. In *ISMIR*, 2021.

[10] Jun-You Wang and Jyh-Shing Roger Jang. Training a singing transcription model using connectionist temporal classification loss and cross-entropy loss. *TASLP*, 2023.

[11] Eita Nakamura and Kazuyoshi Yoshii. Musical rhythm transcription based on Bayesian piece-specific score models capturing repetitions. *Information Sciences*, Vol. 572, pp. 482–500, 2021.

[12] Ryo Nishikimi, Eita Nakamura, Masataka Goto, Katsutoshi Itoyama, and Kazuyoshi Yoshii. Scale- and rhythm-aware musical note estimation for vocal F0 trajectories based on a semi-tatum-synchronous hierarchical hidden semi-Markov model. In *ISMIR*, pp. 376–382, 2017.

[13] Ryo Nishikimi, Eita Nakamura, Masataka Goto, and Kazuyoshi Yoshii. Audio-to-score singing transcription based on a CRNN-HSMM hybrid model. *APSIPA*, Vol. 10, p. e7, 2021.

[14] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pp. 369–376, 2006.

[15] Miguel A Román, Antonio Pertusa, and Jorge Calvo-Zaragoza. An end-to-end framework for audio-to-score music transcription on monophonic excerpts. In *ISMIR*, pp. 34–41, 2018.

[16] Victor Arroyo, Jose J. Valero-Mas, Jorge Calvo-Zaragoza, and Antonio Pertusa. Neural audio-to-score music transcription for unconstrained polyphony using compact output representations. In *ICASSP*, pp. 4603–4607, 2022.

[17] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, pp. 1764–1772, 2014.

[18] Han-Wen Nienhuys and Jan Nieuwenhuizen. Lilypond, a system for automated music engraving. In *CIM*, pp. 167–171, 2003.

[19] Craig Stuart Sapp. Online database of scores in the humdrum file format. In *ISMIR*, pp. 664–665, 2005.

[20] Matthias Mauch and Simon Dixon. Pyin: A fundamental frequency estimator using probabilistic threshold distributions. In *ICASSP*, pp. 659–663, 2014.

[21] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Dai, J. Bello, and S. Dixon. Computer-aided melody note transcription using the tony software: Accuracy and efficiency. In *TENOR*, 2015.

[22] Kentaro Shibata, Eita Nakamura, and Kazuyoshi Yoshii. Non-local musical statistics as guides for audio-to-score piano transcription. *Information Sciences*, Vol. 566, pp. 262–280, 2021.

[23] Eita Nakamura, Kazuyoshi Yoshii, and Shigeki Sagayama. Rhythm transcription of polyphonic piano music based on merged-output hmm for multiple voices. *TASLP*, Vol. 25(4), pp. 794–806, 2017.

[24] Ryo Nishikimi, Eita Nakamura, Masataka Goto, and Kazuyoshi Yoshii. End-to-end melody note transcription based on a beat-synchronous attention mechanism. In *WASPAA*, pp. 26–30, 2019.

[25] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji. Open-Unmix — a reference implementation for music source separation. *Journal of Open Source Software*, 2019.

[26] Curtis Wigington, Brian Price, and Scott Cohen. Multi-label connectionist temporal classification. In *ICDAR*, pp. 979–986, 2019.

[27] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. Rwc music database: Popular, classical and jazz music databases. In *ISMIR*, 2002.

[28] Jun-You Wang and Jyh-Shing Roger Jang. On the preparation and validation of a large-scale dataset of singing transcription. In *ICASSP*, pp. 276–280, 2021.

[29] Eita Nakamura, Emmanouil Benetos, Kazuyoshi Yoshii, and Simon Dixon. Towards complete polyphonic music transcription: Integrating multi-pitch detection and rhythm quantization. In *ICASSP*, pp. 101–105, 2018.

[30] Florian Krebs, Sebastian Böck, and Gerhard Widmer. An efficient state-space model for joint tempo and meter tracking. In *ISMIR*, pp. 72–78, 2015.