Note Value Recognition for Piano Transcription Using Markov Random Fields

Eita Nakamura, *Member, IEEE*, Kazuyoshi Yoshii, *Member, IEEE*, and Simon Dixon

***Abstract*—This paper presents a statistical method for use in music transcription that can estimate score times of note onsets and offsets from polyphonic MIDI performance signals. Because performed note durations can deviate largely from score-indicated values, previous methods had the problem of not being able to accurately estimate offset score times (or note values) and, thus, could only output incomplete musical scores. Based on observations that the pitch context and onset score times are influential on the configuration of note values, we construct a context-tree model that provides prior distributions of note values using these features and combine it with a performance model in the framework of Markov random fields. Evaluation results show that our method reduces the average error rate by around 40 percent compared to existing/simple methods. We also confirmed that, in our model, the score model plays a more important role than the performance model, and it automatically captures the voice structure by unsupervised learning.**

***Index Terms*—Markov random field, model for polyphonic musical scores, music transcription, statistical music language model, symbolic music processing.**

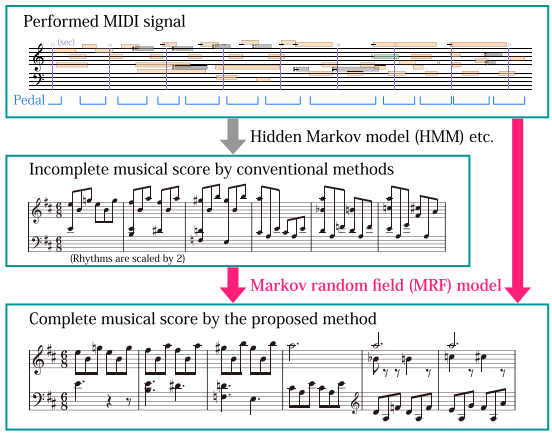

Fig. 1. An outcome obtained by our method (Mozart: Piano Sonata K576). While previous rhythm transcription methods could only estimate onset score times accurately from MIDI performances, our method can also estimate offset score times, providing a complete representation of polyphonic musical scores.

I. Introduction

MUSIC transcription is one of the most fundamental and challenging problems in music information processing [1], [2]. This problem, which involves conversion of audio signals into symbolic musical scores, can be divided into two subproblems, pitch analysis and rhythm transcription, which are often studied separately. Pitch analysis aims to convert the audio signals into the form of a piano roll, which can be represented as a MIDI signal, and multi-pitch analysis methods for polyphonic music have been extensively studied [3]–[6]. Rhythm transcription, on the other hand, aims to convert a MIDI signal into a musical score by locating note onsets and offsets in musical time (*score time*) [7]–[16]. In order to track time-varying tempo, beat tracking is employed to locate beat positions in music audio signals [17]–[21].

Although most studies on rhythm transcription and beat tracking have focused on estimating onset score times, to obtain complete musical scores it is necessary to locate note offsets, or equivalently, identify *note values* defined as the difference between onset and offset score times. The configuration of note values is especially important to describe the acoustic and interpretative nature of polyphonic music where there are multiple voices and the overlapping of notes produces different harmonies. Note value recognition has been addressed only in a few studies [10], [14] and the results of this study reveal that it is a non-trivial problem.

The difficulty of the problem arises from the fact that observed note durations in performances deviate largely from the score-indicated lengths so that the use of a prior (language) model for musical scores is crucial. Because of its structure with overlapping multiple streams (voices), construction of a language model for polyphonic music is challenging and gathers increasing attention [6], [14], [16], [22], [23]. In particular, building a model at the symbolic level of musical notes (as opposed to the frame level of audio processing) that properly describes the multiple-voice structure while retaining computational tractability is an open problem.

Manuscript received March 23, 2017; revised June 9, 2017; accepted June 20, 2017. Date of publication June 30, 2017; date of current version August 15, 2017. This work was supported in part by JSPS KAKENHI under Grants 24220006, 26280089, 26700020, 15K16054, 16H01744, and 16J05486, in part by JST ACCEL under Grant JPMJAC1602, and in part by the long-term overseas research fund by the Telecommunications Advancement Foundation. The work of E. Nakamura was supported by the JSPS research fellowship (PD). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Matthew E. P. Davies. *(Corresponding author: Eita Nakamura.)*

E. Nakamura is with the Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan. This work was done while he was a visiting researcher at Queen Mary University of London, London E1 4NS, U.K. (e-mail: enakamura@sap.ist.i.kyoto-u.ac.jp).

K. Yoshii is with the Graduate School of Informatics, Kyoto University, Kyoto 606-8501 Japan, and also with AIP Center for Advanced Intelligence Project, Tokyo 103-0027, Japan (e-mail: yoshii@kuis.kyoto-u.ac.jp).

S. Dixon is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (e-mail: s.e.dixon@qmul.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2017.2722103

The purpose of this paper is to investigate the problem of note value recognition using a statistical approach (Fig. 1). We formulate the problem as a post-processing step of estimating offset score times given onset score times obtained by rhythm transcription methods for note onsets. Firstly, we present results of statistical analyses and point out that the information of onset score times and the pitch context together with interdependence between note values provide clues for model construction. Secondly, we propose a Markov random field model that integrates a prior model for musical scores and a performance model that relates note values and actual durations (Section IV). To determine an optimal set of contexts/features for the score model from data, we develop a statistical learning method based on context-tree clustering [24]–[26], which is an adaptation of statistical decision tree analysis. Finally, results of systematic evaluations of the proposed method and baseline methods are presented (Section V).

The contributions of this study are as follows. We formulate a statistical learning method to construct a highly predictive prior model for note values and quantitatively demonstrate its importance for the first time. The discussions cover simple methods and more sophisticated machine learning techniques and the evaluation results can serve as a reference for the state-of-the-art. Our problem is formulated in a general setting following previous studies on rhythm transcription and the method is applicable to a wide range of existing methods of onset rhythm transcription. Results of statistical analyses and learning in Sections III and IV can also serve as a useful guide for research using other approaches such as rule-based methods and neural networks. Lastly, source code of our algorithms and evaluation tools is available from the accompanying web page [27] to facilitate future comparisons and applications.

II. Related Work

Before beginning the main discussion, let us review previous studies related to this paper.

There have been many studies on converting MIDI performance signals into a form of musical score. Older studies [7], [8] used rule-based methods and networks in attempts to model the process of human perception of musical rhythm. Since around 2000, various statistical models have been proposed to combine the statistical nature of note sequences in musical scores and that of temporal fluctuations in music performance. A popular approach is to use hidden Markov models (HMMs) [9]–[12], [16]. The score is described either as a Markov process on beat positions (metrical Markov model) [9], [11], [12] or a Markov model of notes (note Markov model) [10], and the performance model is often constructed as a state-space model with latent variables describing locally defined tempos. Recently a merged-output HMM incorporating the multiple-voice structure has been proposed [16]. Temperley [14] proposed a score model similar to the metrical Markov model in which the hierarchical metrical structure is explicitly described. There are also studies that investigated probabilistic context-free grammar models [15].

A recent study [16] reported results of systematic evaluation of (onset) rhythm transcription methods. Two data sets, polyrhythmic data and non-polyrhythmic data, were used and it was shown that HMM-based methods generally performed better than others and the merged-output HMM was most effective for polyrhythmic data. In addition to the accuracy of recognising onset beat positions, the metrical HMM has the advantage of being able to estimate metrical structure, i.e. the metre (duple or triple) and bar (or down beat) positions, and to avoid grammatically incorrect scores that appeared in other HMMs.

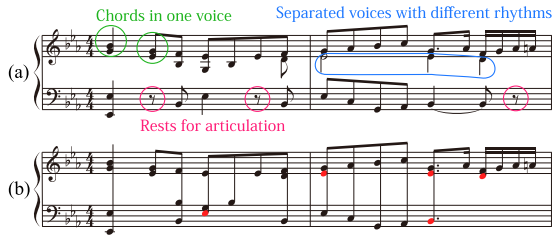

Fig. 2. Example of (a) a polyphonic piano score (Mozart: Sonata KV570) and (b) a reduced score represented with one voice. Notes that have different note values in the two representations are indicated with red note heads.

As mentioned above, there have been only a few studies that discussed the recognition of note values in addition to onset score times. Takeda *et al.* [10] applied a similar method of estimating onset score times to estimating note values of monophonic performances and reported that the recognition accuracy dropped from 97.3% to 59.7% if rests are included. Temperley's Melisma Analyzer [14], based on a statistical model, outputs estimated onset and offset beat positions together with voice information for polyphonic music. There, offset score times are chosen from one of the following tactus beats according to some probabilities, or chosen as the onset position of the next note of the same voice. The recognition accuracy of note values has not been reported.

III. Preliminary Observations and Analyses

We explain here basic facts about the structure of polyphonic piano scores and discuss how it is important and non-trivial to recognise note values for such music based on observations and statistical analyses. This provides motivations for the architecture of our model. Some terminology and notions used in this paper are also introduced. We consider the music style of the common practice period and similar music styles such as popular and jazz music in this paper.

A. Structure of Polyphonic Musical Scores

To discuss recognition of note values in polyphonic piano music, we first explain the structure of polyphonic scores. The left-hand and right-hand parts are usually written in separate staffs and each staff can contain several *voices*[1], or streams of notes [Fig. 2(a)]. In piano scores, each voice can contain chords and the number of voices can vary locally. Hereafter we use the word *chords* to indicate those within one voice. Except for rare cases of partial ties in chords, notes in a chord must have simultaneous onset and offset score times. This means that the offset score time of a note must be equal to or earlier than

[1]Our "voice" corresponds to the voice information defined in music notation file formats such as MusicXML and Finale file format.

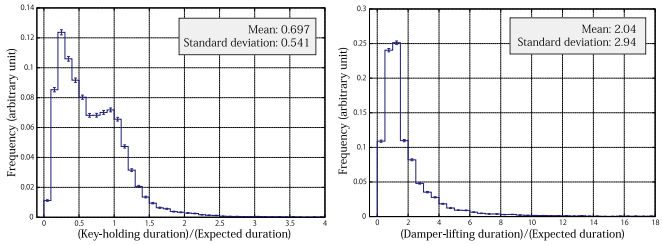

Fig. 3. Distributions of the ratios of actual duration, (a) key-holding durations and (b) damper-lifting durations, to the expected duration.

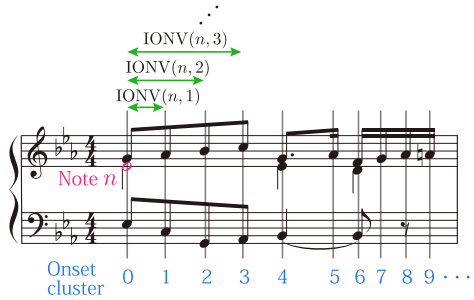

Fig. 4. Onset clusters and inter-onset note values (IONVs).

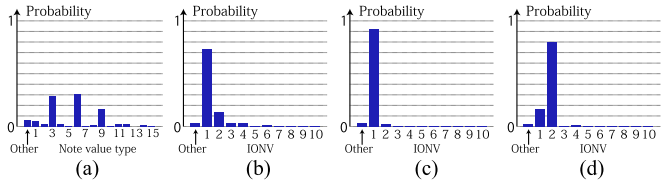

Fig. 5. Distributions of note values. In (a), note values are categorised into 15 types in (1) and another type including all others; in (b), (c), (d), they are categorised into the first ten IONVs and others. Samples in (c)(d) were selected by conditions on the pitch context described in the text.

the onset score time of the next note/chord of the same voice. In the latter case, the note is followed by a rest. Such rests are rare [14] and thus the configuration of note values and the voice structure are inter-related.

The importance of voice structure in the description of note values can also be understood by comparing a polyphonic score with a reduced score obtained by putting all notes with simultaneous onsets into a chord and forming one 'big voice' without any rests as in Fig. 2(b). Since these two scores are the same in terms of onset score times, the differences are only in offset score times. One can see that appropriate voice structure is necessary to recover correct note values from the reduced score. It can also be confirmed that note values are influential to realise the expected acoustic effect of polyphonic music. As one can automatically obtain the reduced score given the onset score times, recovering the polyphonic score as in Fig. 2(a) from the reduced score as in Fig. 2(b) is the aim of note value recognition.

B. Distribution of Durations in Music Performances

A natural approach to recover note values from MIDI performances is finding those note values that best fit the actual note durations in the performances. In this paper, *duration* always means the time length measured in physical time, and a score-written note length is called a note value. To relate durations to note values, one needs the (local) tempo that provides the conversion ratio. Although estimating tempos from MIDI performances is a nontrivial problem (see Section IV), let us suppose here they are given, for simplicity. Given a local tempo and a note value, one can calculate an expected duration, and conversely, one can estimate a note value given a local tempo and actual duration.

Fig. 3 shows distributions of the ratios of actual durations in performances and the durations expected from note values and tempos estimated from onset times (used performance data is described in Section IV-D). Because information of key-press and key-release times for each note and pedal movements can be obtained from MIDI signals, one can define the following two durations. The *key-holding duration* is the time interval between key-press and key-release times and the *damper-lifting duration* is obtained by extending the offset time as long as the sustain/sostenuto pedal is held. As can be seen from the figure, both distributions have large variances and thus precise prediction of note values is impossible by using only the observed values. As mentioned previously [12], [14], this makes note value recognition a difficult problem and it has often been avoided in previous studies. Additionally, due to the large deviations of durations, most tempo estimation methods use only onset time information.

A similar situation happens in speech recognition where the presence of acoustic variations and noise makes it difficult to extract symbolic text information by pure feature extraction. Similarly to using a prior language model, which was the key to improve the accuracy of speech recognition [28], a prior model for musical scores (*score model*) would be a key to solving our problem, which we seek in this paper.

C. Hints for Constructing a Score Model

The simplest score model for note value recognition would be a discrete probability distribution over a set of note values. For example, one can consider the following 15 types of note values (e.g. $1/2 =$ half note, $3/16 =$ dotted eighth note, etc.):

$$\left\{\frac{1}{32}, \frac{1}{48}, \frac{1}{16}, \frac{1}{24}, \frac{3}{32}, \frac{1}{8}, \frac{1}{12}, \frac{3}{16}, \frac{1}{4}, \frac{1}{6}, \frac{3}{8}, \frac{1}{2}, \frac{1}{3}, \frac{3}{4}, 1\right\}. \quad (1)$$

The distribution taken from a score data set (see Section IV-D) is shown in Fig. 5(a). Although the distribution has clear tendencies, it is not sufficiently sharp to compensate the large variance of the duration distributions. We will confirm that this simple model yields a poor recognition accuracy in Section V-B.

Hints for constructing a score model can be obtained by again observing the example in Fig. 2. It is observed that most notes in the reduced score have the same note values as in the original score, and even when they do not, the offset score times tend to correspond with one of the onset score times of following notes. To explain this more precisely in a statistical way, we define an *onset cluster* as the set of all notes with simultaneous onsets in the score and *inter-onset note values* (*IONVs*) as the intervals between onset score times of succeeding onset clusters (Fig. 4). As in the figure, for later convenience, we define IONVs for each note, even though they are same for all notes in an onset cluster. If one counts frequencies that each note value matches one of the first ten IONVs (or none of them), the result is as shown in

Fig. 5(b). We see that the distribution has lower entropy than that in Fig. 5(a) and the probability that note values would be different from any of the first ten IONVs is small (3.50% in our data). This suggests that a more efficient search space for note values can be obtained by using the onset score time information.

Even more predictive distributions of note values can be obtained by using the pitch information. This is because neighbouring notes (either horizontally or vertically) in a voice tend to have close pitches, as discussed in studies on voice separation [29]–[31]. For example, if we select notes that have a note within five semitones in the next onset cluster, the distribution of note values in the space of IONVs becomes as in Fig. 5(c), reflecting the fact that inserted rests are rare. On the other hand, if we impose a condition of having a note with five semitones in the second next onset cluster but not having any notes within 14 semitones in the next cluster, then the distribution becomes as in Fig. 5(d), which reflects the fact that this condition implies that the note has an adjacent note in the same voice in the second next onset cluster. These results suggest on one side that pitch information together with onset score time information can provide distributions of note values with more predictive ability and on the other side that those distributions are highly dependent on the pitch context.

Although so far we have considered note values as independent distributions, their interdependence can also provide clues in estimating note values. One such interdependence can be inferred from the logical constraint of voice structure described in Section III-A. As chordal notes have the same note values and they also tend to have close pitches, notes with simultaneous onset score times and close pitches tend to have identical note values. This is another case where pitch information has influence on the distribution of note values.

D. Summary of the Section

Here we summarise the findings in this section:

- The voice structure and the configuration of note values are inter-related and the logical constraints for musical scores induce interdependence between note values.
- Performed durations contain large deviations from those implied by the score and a score model is crucial to accurately estimate note values from performance signals.
- Information about onset score times provides an efficient search space for note values through the use of IONVs. In particular, the probability that a note value falls into one of the first ten IONVs is quite high.
- The distribution of note values is highly dependent on the pitch context, which would be useful for improving their predictability.

In the rest of this paper, we construct a computational model to incorporate these findings and examine by numerical experiments how they quantitatively influence the accuracy of note value recognition.

IV. Proposed Method

A. Problem Statement

For rhythm transcription, the input is a MIDI performance signal, represented as a sequence of pitches, onset times and offset times $(p_n, t_n, t_n^{\mathrm{off}}, \bar{t}_n^{\mathrm{off}})_{n=1}^N$ where n is an index for notes and N is the number of notes. As explained in Section III-B, we can define two offset time, the key-release time and damper-drop time, denoted by t_n^{off} and $\bar{t}_n^{\mathrm{off}}$. The corresponding key-holding and damper-lifting duration will be denoted by $d_n = t_n^{\mathrm{off}} - t_n$ and $\bar{d}_n = \bar{t}_n^{\mathrm{off}} - t_n$. The aim is to recognise the score times of the note onsets and offsets, which are denoted by $(\tau_n, \tau_n^{\mathrm{off}})_{n=1}^N$. In general, τ_n and τ_n^{off} take values in the set of rational numbers in units of a beat unit, say, the whole-note length. For example, $\tau_1 = 0$ and $\tau_1^{\mathrm{off}} = 1/4$ means that the first note is at the beginning of the score and has a quarter-note length. We use the following notations for sequences: $\boldsymbol{d} = (d_n)_{n=1}^N$, $\boldsymbol{\tau}^{\mathrm{off}} = (\tau_n^{\mathrm{off}})_{n=1}^N$, etc. We call the difference $r_n = \tau_n^{\mathrm{off}} - \tau_n$ the *note value*. Frequently used mathematical symbols are listed in Table I.

TABLE I
List of Frequently Used Mathematical Symbols

Variable	Notation
Index for note	n
Pitch	p_n
Onset time	t_n
Key-release [Damper-drop] (offset) time	t_n^{off} $[\bar{t}_n^{\mathrm{off}}]$
Key-holding [Damper-lifting] duration	d_n $[\bar{d}_n]$
Onset [offset] score time	τ_n $[\tau_n^{\mathrm{off}}]$
Note value	r_n
Local tempo	v_n
Sequence of variables	$\boldsymbol{p} = (p_n)_{n=1}^N$ etc.

In this paper, we consider the situation that the onset score times $\boldsymbol{\tau}$ are given as estimations from conventional onset rhythm transcription algorithms. In addition, we assume that a local tempo v_n, which gives a smoothed ratio of the time interval and score time interval at each note n, is given. Local tempos $\boldsymbol{v} = (v_n)_{n=1}^N$ can be obtained from the sequences $\boldsymbol{t}$ and $\boldsymbol{\tau}$ by applying some smoothing methods such as Kalman smoothing and local averaging, and typically they can be obtained as outputs of onset rhythm transcription algorithms.

In summary, we set up the problem of note value recognition as estimating the sequence $\boldsymbol{\tau}^{\mathrm{off}}$ (or $\boldsymbol{r}$) with inputs $\boldsymbol{p}, \boldsymbol{d}, \bar{\boldsymbol{d}}, \boldsymbol{\tau}$ and $\boldsymbol{v}$. For concreteness, in this paper, we mainly use as $\boldsymbol{\tau}$ and $\boldsymbol{v}$ the outputs from a method based on a metrical HMM (Section IV-B), but our method is applicable as a post-processing step for any rhythm transcription method that outputs $\boldsymbol{\tau}$.

B. Estimation of Onset Score Times and Local Tempos

To estimate onset score times $\boldsymbol{\tau}$ and local tempos $\boldsymbol{v}$ from a MIDI performance $(\boldsymbol{p}, \boldsymbol{t}, \boldsymbol{t}^{\mathrm{off}}, \bar{\boldsymbol{t}}^{\mathrm{off}})$, we use a metrical HMM [9], which is one of the most accurate onset rhythm transcription methods (Section II). Here we briefly review the model.

In the metrical HMM, the probability $P(\boldsymbol{\tau})$ of the score is generated from a Markov process on periodically defined beat positions denoted by $(s_n)_{n=1}^N$ with $s_n \in \{1, \ldots, G\}$ (G is a period of beats such as a bar). The sequence $\boldsymbol{s}$ is generated with the initial and transition probabilities as

$$P(\boldsymbol{s}) = P(s_1) \prod_{n=2}^{N} P(s_n | s_{n-1}). \tag{2}$$

We interpret s_n as τ_n modulo G, or more explicitly, we obtain $\boldsymbol{\tau}$ incrementally as follows:

$$\tau_1 = s_1, \tag{3}$$

$$\tau_{n+1} = \tau_n + \begin{cases} s_{n+1} - s_n, & \text{if } s_{n+1} > s_n; \\ G + s_{n+1} - s_n, & \text{if } s_{n+1} \leq s_n. \end{cases} \tag{4}$$

That is, if $s_{n+1} \leq s_n$, we interpret that s_{n+1} indicates the beat position in the next bar. With this understood, $P(\boldsymbol{\tau})$ is equivalent to $P(\boldsymbol{s})$ as long as $r_n \leq G$ for all n. An extension is possible to allow note onset intervals larger than G [32].

In constructing the performance model, local tempo variables $\boldsymbol{v}$ are introduced to describe the indeterminacy and temporal variations of tempos. The probability $P(\boldsymbol{t}, \boldsymbol{v}|\boldsymbol{\tau})$ is decomposed as $P(\boldsymbol{t}|\boldsymbol{\tau}, \boldsymbol{v})P(\boldsymbol{v})$ and each factor is described with the following Gaussian Markov process:

$$P(v_n|v_{n-1}) = N(v_n; v_{n-1}, \sigma_v^2), \tag{5}$$

$$P(t_{n+1}|t_n, \tau_{n+1}, \tau_n, v_n) = N\big(t_{n+1}; t_n + (\tau_{n+1} - \tau_n)v_n, \sigma_t^2\big) \tag{6}$$

where $N(\,\cdot\,; \mu, \Sigma)$ denotes a normal distribution with mean μ and variance Σ, and σ_v and σ_t are standard deviations representing the degree of tempo variations and onset time fluctuations, respectively. An initial distribution for v_1 is described similarly as a Gaussian $N(v_1; v_{\mathrm{ini}}, \sigma_{v,\mathrm{ini}}^2)$.

An algorithm to estimate onset score times and local tempos can be obtained by maximising the posterior probability $P(\boldsymbol{\tau}, \boldsymbol{v}|\boldsymbol{t}) \propto P(\boldsymbol{t}, \boldsymbol{v}|\boldsymbol{\tau})P(\boldsymbol{\tau})$. This can be done by a standard Viterbi algorithm after discretisation of the tempo variables [20], [32]. Note that this method does not use the pitch and offset information, which is typical in conventional onset rhythm transcription methods. Since the period G and rhythmic properties encoded in $P(s_n|s_{n-1})$ are dependent on the metre, in practice it is effective to consider multiple metrical HMMs corresponding to different metres, such as duple metre and triple metre, and choose the one with the maximum posterior probability in the stage of inference.

C. Markov Random Field Model

Here we describe our main model. As explained in Section III, it is essential to combine a score model that enables prediction of note values given the input information of onset score times and pitches and a performance model that relates note values to actual durations realised in music performances. To enable tractable inference and efficient parameter estimation, one should typically decompose each model into component models that involve a smaller number of stochastic variables.

As a framework to combine such component models, we consider the following Markov random field (MRF):

$$P(\boldsymbol{r}|\boldsymbol{p}, \boldsymbol{d}, \bar{\boldsymbol{d}}, \boldsymbol{\tau}, \boldsymbol{v}) \propto \exp\Bigg[-\sum_{n=1}^{N} H_1(r_n; \boldsymbol{\tau}, \boldsymbol{p}) - \sum_{(n,m)\in\mathcal{N}} H_2(r_n, r_m) - \sum_{n=1}^{N} H_3(r_n; d_n, \bar{d}_n, v_n) \Bigg]. \tag{7}$$

Here H_1 (called the *context model*) represents the prior model for each note value that depends on the onset score times and pitches, H_2 (the *interdependence model*) represents the interdependence of neighbouring pairs of note values ($\mathcal{N}$ denotes the set of neighbouring note pairs specified later) and H_3 (the *performance model*) represents the likelihood model. Each term can be interpreted as an energy function that has small values when the arguments have higher probabilities. The explicit forms of these functions are given as follows:

$$H_1 = -\beta_1 \ln P(r_n; \boldsymbol{\tau}, \boldsymbol{p}), \tag{8}$$

$$H_2 = -\beta_2 \ln P(r_n, r_m), \tag{9}$$

$$H_3 = -\beta_{31} \ln P(d_n; r_n, v_n) - \beta_{32} \ln P(\bar{d}_n; r_n, v_n). \tag{10}$$

Each energy function is constructed with a negative log probability function multiplied by a positive weight. These weights β_1, β_2, β_{31} and β_{32} are introduced to represent the relative importance of the component models. For example, if we take $\beta_1 = \beta_{31} = \beta_{32} = 1$ and $\beta_2 = 0$, the model reduces to a Naive Bayes model with the durations considered as features. For other values of β s, the model is no longer a generative model for the durations but still a generative model for the note values, which are the only unknown variables in our problem. In the following we explain the component models in detail. Learning parameters including β s is discussed in Section IV-D.

1) Context Model: The context model H_1 describes a prior distribution for note values that is conditionally dependent on given onset score times and pitches. To construct this model, one should first specify the sample space of r_n, or, the set of possible values that each r_n can take. Based on the observations in Section III, we consider the first ten IONVs as possible values of r_n. Since r_n can take other values in reality, we also consider a formally defined value '**other**', which represents all other values of r_n. Let

$$\Omega_r(n) = \{\mathrm{IONV}(n, 1), \ldots, \mathrm{IONV}(n, 10), \mathbf{other}\}$$

denote the sample space. Therefore $P(r_n; \boldsymbol{\tau}, \boldsymbol{p})$ is considered as an 11-dimensional discrete distribution.

As we saw in Section III, the distribution $P(r_n; \boldsymbol{\tau}, \boldsymbol{p})$ depends heavily on the pitch context. Based on our intuition that for each note n the succeeding notes with a close pitch are most influential on the voice structure, in this paper we use the feature vector $c_n = (c_n(1), \ldots, c_n(10))$ as a context of note n, where $c_n(k)$ denotes the unsigned pitch interval between note n and the closest pitch in its k-th next onset cluster. An example of the context is given in Fig. 6. Thus we have

$$P(r_n; \boldsymbol{\tau}, \boldsymbol{p}) = P\big(r_n; c_n(1), \ldots, c_n(10)\big). \tag{11}$$

We remark that in general we can additionally consider different features (for example, metrical features) and our formulation in this section and in Section IV-D is valid independently of our particular choice of features.

Due to the huge number of different contexts for notes, it is not practical to use (11) directly. With 88 pitches on a piano keyboard, each $c_n(k)$ can take 87 values and thus the right-hand side (RHS) of (11) has $11 \cdot 87^{10}$ parameters (or slightly less free parameters after normalisation), which is computationally infeasible. (If one uses additional features, the number of parameters increases further.) To solve this problem, we use a

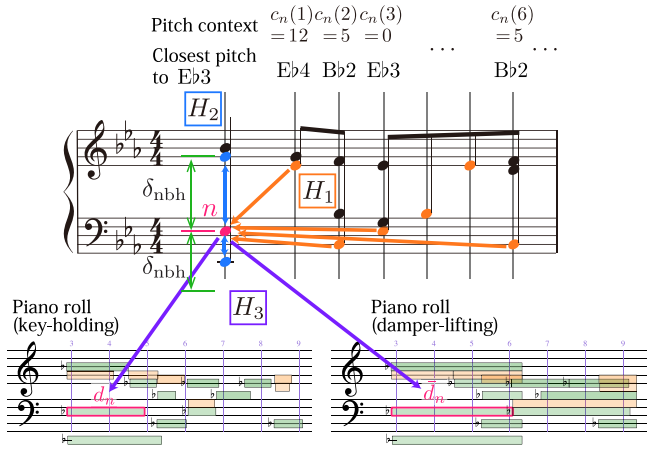

Fig. 6. Statistical dependencies in the Markov random field model.

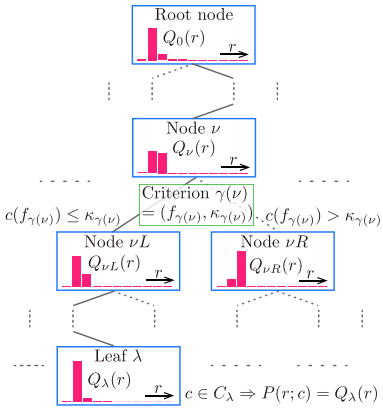

Fig. 7. In a context-tree model, the distribution of a quantity r is categorised with a set of criteria on the context c.

context-tree model [24], [25], in which contexts are categorised according to a set of criteria that are represented as a tree (as in decision tree analysis) and all contexts in one category have the same probability distribution.

Formally, a context-tree model is defined as follows. Here we consider a general *context* $c = (c(1), \ldots, c(F))$, which is an F-dimensional feature vector. We assume that the set of possible values for $c(f)$ is an ordered set for all $f = 1, \ldots, F$ and denote it by R_f. Let us denote the leaf nodes of a binary tree T by ∂T. Each node $\nu \in T$ is associated with a set of contexts denoted by C_ν. In particular, for the root node $0 \in T$, C_0 is the set of all contexts $(R_1 \times \cdots \times R_F)$. Each internal node $\nu \in T \backslash \partial T$ is associated with a criterion $\gamma(\nu)$ for selecting a subset of C_ν. A *criterion* $\gamma = (f_\gamma, \kappa_\gamma)$ is defined as a pair of a feature dimension $f_\gamma \in \{1, \ldots, F\}$ and a cut $\kappa_\gamma \in R_{f_\gamma}$. The criterion divides a set of contexts C into two subsets as

$$C^L(\gamma) = \{c \in C \,|\, c(f_\gamma) \le \kappa_\gamma\}, \tag{12}$$

$$C^R(\gamma) = \{c \in C \,|\, c(f_\gamma) > \kappa_\gamma\}, \tag{13}$$

so that $C^L(\gamma) \cap C^R(\gamma) = \phi$ and $C^L(\gamma) \cup C^R(\gamma) = C$. Now denoting the left and right child of $\nu \in T \backslash \partial T$ by νL and νR, their sets of contexts are defined as $C_{\nu L} = C_\nu \cap C_0^L(\gamma(\nu))$ and $C_{\nu R} = C_\nu \cap C_0^R(\gamma(\nu))$, which recursively defines a *context tree* (T, f, κ) (Fig. 7). By definition, a context is associated to a unique leaf node: for all $c \in C_0$ there exists a unique $\lambda \in \partial T$ such that $c \in C_\lambda$. We denote such a leaf by $\lambda(c)$. Finally, for each node $\nu \in T$, a probability distribution $Q_\nu(\,\cdot\,)$ is associated. Now we can define the probability $P_{\mathscr{T}}(\,\cdot\,; c)$ as

$$P_{\mathscr{T}}(\,\cdot\,; c) = Q_{\lambda(c)}(\,\cdot\,). \tag{14}$$

The tuple $\mathscr{T} = (T, f, \kappa, Q)$ defines a *context-tree model*.

For a context-tree model with L leaves, the number of parameters for the distribution of note values is now reduced to $11L$. In general a model with a larger tree size has more ability to approximate (11) at the cost of an increasing number of model parameters. The next problem is to find the optimal tree size and the optimal criterion for each internal node. We will explain this in Section IV-D1.

2) Interdependence Model: Although the distribution of note values in the context model is dependent on the pitch context, it is independently defined for each note value. As explained in Section III, interdependence of note values is also important since it arises from logical constraint on the voice structure. Such interdependence can be described with a joint probability of note values of a pair of notes in H_2. As in the context model, we consider the set Ω_r as a sample space for note values so that the joint probability $P(r_n, r_m)$ for notes n and m has 11^2 parameters.

The choice of the set of neighbouring note pairs $\mathscr{N}$ in (7) is most important for the interdependence model. In order to capture the voice structure we define $\mathscr{N}$ as

$$\mathscr{N} = \{(n, m) \,|\, \tau_n = \tau_m, \, |p_n - p_m| \le \delta_{\mathrm{nbh}}\} \tag{15}$$

where δ_{nbh} is a parameter to define the vicinity of the pitch. The value of δ_{nbh} is determined from data (see Section IV-D4).

3) Performance Model: The performance model is constructed with the probability of actual durations in performances given a note value and a local tempo. Since we can use two durations d_n and $\bar{d}_n$, two distributions, $P(d_n; r_n, v_n)$ and $P(\bar{d}_n; r_n, v_n)$, are considered for each note as in the RHS of (10). To regulate the effect of varying tempos and avoid the increase in the complexity of the model to handle possibly many types of note values, we consider distributions over normalised durations, $d'_n = d_n/(r_n v_n)$ and $\bar{d}'_n = \bar{d}_n/(r_n v_n)$, as we did in Section III. We therefore assume

$$P(d_n; r_n, v_n) = g(d'_n) \quad \text{and} \quad P(\bar{d}_n; r_n, v_n) = \bar{g}(\bar{d}'_n) \tag{16}$$

where g and $\bar{g}$ are one-dimensional probability distributions supported on positive real numbers.

The histograms corresponding to g and $\bar{g}$ taken from performance data described in Section IV-D are illustrated in Fig. 3. One can recognise two (one) peak(s) for the distribution of normalised key-holding (damper-lifting) durations. Since theoretical forms of these distributions are unknown, we use as phenomenologically fitting distributions the following generalised inverse-Gaussian **(GIG)** distribution:

$$\mathbf{GIG}(x; a, b, h) = \frac{(a/b)^{h/2}}{2K_h(2\sqrt{ab})} x^{h-1} e^{-(ax + b/x)} \tag{17}$$

where $a, b > 0$ and $h \in \mathbb{R}$ are parameters and K_h denotes the modified Bessel function of the second kind. The **GIG** distributions are supported on positive real numbers and include the

gamma ($a \to 0$), inverse-gamma ($b \to 0$) and inverse-Gaussian ($h = -1/2$) distributions as special cases. Since a **GIG** distribution has only one peak, we use a mixture of **GIG** distributions to represent g. We parameterise g and $\bar{g}$ as

$$g(x) = w_1 \mathbf{GIG}(x; a_1, b_1, h_1) + w_2 \mathbf{GIG}(x; a_2, b_2, h_2), \tag{18}$$

$$\bar{g}(x) = \mathbf{GIG}(x; a_3, b_3, h_3) \tag{19}$$

where w_1 and $w_2 = 1 - w_1$ are mixture weights. Parameter values obtained from data are given in Section IV-D3.

D. Model Learning

Similarly as the language model and the acoustic model for a speech recognition system are generally trained separately with different data, our three component models can be trained separately and combined afterwards to determine the optimal weights (the β s). The context model and the interdependence model can be learned with musical score data and we used a dataset of 148 classical piano pieces (with 3.4×10^6 notes) by various composers[2]. On the other hand, the performance model requires performance data aligned with reference scores. The used data consisted of 180 performances (60 phrases $\times$ 3 different players) by various composers and various performers that are mostly collected from publicly available MIDI performances recorded in international piano competitions [33]. Due to the lack of abundant data, we used the same performance data for training and evaluation. Because the number of parameters for the performance model is small (ten independent parameters in g and $\bar{g}$ and two weight parameters) and they are not fine-tunable, there should be little concern about overfitting here and most comparative evaluations in Section V are done with equal conditions. (See also the discussion in Sections IV-D3 and V-C.) To avoid overfitting, the score data and the performance data contained no overlapping musical pieces (at least in units of movements). Learning methods for the component models are described in the following sections and Section IV-D4 describes the optimisation of the β s.

1) Learning the Context Model: The context-tree model can be learned by growing the tree based on the maximum likelihood (ML) principle, which is called *context-tree clustering*. This is usually done by recursively splitting a node that minimises the likelihood [24]. Although it is not essentially new, we describe the learning method here for the readers' convenience because context-tree clustering is not commonly used in the field of music informatics and in articles for speech processing (where it is widely used) the notations are adapted for the case with Gaussian distributions, which is not ours.

Let $x_i = (r_i, c_i)$ denote a sample extracted from score data, where i denotes a note in the score data, r_i denotes an element in $\Omega_r(i)$ and c_i denotes the context of note i. The set of all samples will be denoted by $\boldsymbol{x} = (x_i)_{i=1}^I$. The log likelihood $L_{\mathscr{T}}(\boldsymbol{x})$ of a context-tree model $\mathscr{T} = (T, f, \kappa, Q)$ is given as

$$\begin{aligned} L_{\mathscr{T}}(\boldsymbol{x}) &= \sum_{i=1}^{I} \ln P_{\mathscr{T}}(x_i) = \sum_{i=1}^{I} \ln Q_{\lambda(c_i)}(x_i) \\ &= \sum_{\lambda \in \partial T} \sum_{i:\, c_i \in C_\lambda} q_\lambda(x_i) \end{aligned} \tag{20}$$

where in the second line we decomposed the samples according to the criteria of the leaves and hereafter we denote $q_\nu(\,\cdot\,) = \ln Q_\nu(\,\cdot\,)$ for each node ν. The parameters for each distribution Q_ν for node $\nu \in T$ are learned from the samples $\{x_i | c_i \in C_\nu\}$ according to the ML method. We implicitly understand that all Q s are already learned in this way.

Given a context tree $\mathscr{T}^{(m)}$ (one begins with a tree $\mathscr{T}^{(0)}$ containing only the root node and proceeds $m = 0, 1, 2, \ldots$ as follows), one of the leaves $\lambda \in \partial T^{(m)}$ is split according to some additional criterion $\gamma(\lambda)$. Let us denote the expanded context-tree model by $\mathscr{T}_\lambda^{(m)}$. Since $\mathscr{T}_\lambda^{(m)}$ is same as $\mathscr{T}^{(m)}$ except for the new leaves λL and λR, the difference of log likelihoods $\Delta L(\lambda) = L_{\mathscr{T}_\lambda^{(m)}}(\boldsymbol{x}) - L_{\mathscr{T}^{(m)}}(\boldsymbol{x})$ is given as

$$\sum_{i:\, c_i \in C_{\lambda L}} q_{\lambda L}(x_i) + \sum_{i:\, c_i \in C_{\lambda R}} q_{\lambda R}(x_i) - \sum_{i:\, c_i \in C_\lambda} q_\lambda(x_i). \tag{21}$$

Note that $\Delta L(\lambda) \geq 0$ since $Q_{\lambda L}$ and $Q_{\lambda R}$ have the ML. Now the leaf λ^* and the criterion $\gamma(\lambda^*)$ that maximise $\Delta L(\lambda)$ are selected for growing the context tree: $\mathscr{T}^{(m+1)} = \mathscr{T}_{\lambda^*}^{(m)}$.

According to the above ML criterion, the context tree can be expanded to the point where all samples are completely separated by contexts, for which the model often suffers from overfitting. To avoid this and find an optimal tree size according to the data, the minimal description length (MDL) criterion for model selection can be used [26], [34]. The MDL $\ell_{\mathcal{M}}(\boldsymbol{x})$ for a model $\mathcal{M}$ with parameters $\theta_{\mathcal{M}}$ is given as

$$\ell_{\mathcal{M}}(\boldsymbol{x}) = -\log_2 P(\boldsymbol{x}; \hat{\theta}_{\mathcal{M}}) + \frac{|\mathcal{M}|}{2} \log_2 I \tag{22}$$

where I is the length of $\boldsymbol{x}$, $|\mathcal{M}|$ is the number of free parameters of model $\mathcal{M}$ and $\hat{\theta}_{\mathcal{M}}$ denotes the ML estimate of $\theta_{\mathcal{M}}$ according to data $\boldsymbol{x}$. Here, the first term in the RHS is the negative log likelihood, which in general decreases when the model's complexity increases. On the other hand, the second term increases when the number of model parameters increases. Thus a model that minimises the MDL is chosen by a trade off of the model's precision and complexity. The MDL criterion is justified by an information-theoretic argument [34].

For our context-tree model, each Q is an 11-dimensional discrete distribution and has ten free parameters, and therefore the increase of parameters by expanding a node is ten. Substituting this into (22), we find

$$\begin{aligned} \Delta \ell(\lambda^*) &= \ell_{\mathscr{T}^{(m+1)}}(\boldsymbol{x}) - \ell_{\mathscr{T}^{(m)}}(\boldsymbol{x}) \\ &= -\Delta L(\lambda^*)/(\ln 2) + (10/2)\log_2 I. \end{aligned} \tag{23}$$

In summary, the context tree is expanded by splitting the optimal leaf λ^*, up to a step where $\Delta \ell(\lambda^*)$ becomes positive.

With our score data of 3.4×10^6 musical notes, the learned context tree had 132 leaves. A subtree is illustrated in Fig. 8

[2]The lists of used pieces for the score data and the performance data are available at the accompanying web page [27].

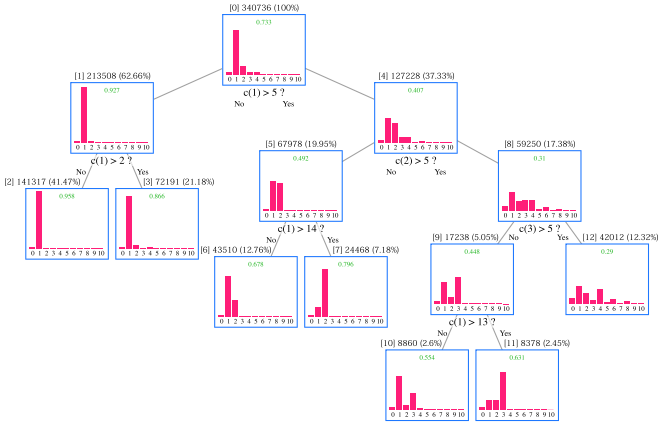

Fig. 8. A subtree of the obtained context-tree model. Above each node are indicated the node ID, number of samples and their proportion in the whole data and the green number indicates the highest probability in each distribution. See text for explanation of the labels for each distribution.

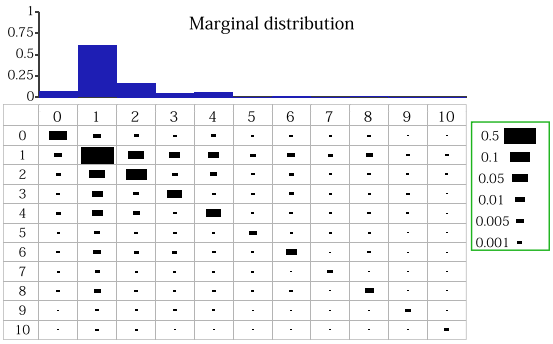

Fig. 9. Joint probability distribution of note values obtained for the interdependence model for $\delta_{\mathrm{nbh}} = 12$. See text for explanation of the labels.

where the node ID is shown in square brackets and the labels $1, \ldots, 10$ in the distribution show those probabilities correspond to $\mathrm{IONV}(1), \ldots, \mathrm{IONV}(10)$ and the label 0 is assigned to the '**other**'. For example, one finds a distribution with a sharp peak at $\mathrm{IONV}(1)$ in node 2 whose contexts satisfy $c(1) \leq 2$. This can be interpreted as follows: if note n has a pitch within 2 semitones in the next onset cluster, then it is highly probable that they are in the same voice and note n has $r_n = \mathrm{IONV}(n, 1)$. On the other hand, the $\mathrm{IONV}(2)$ has the largest probability in node 7 [the distribution is the same one as in Fig. 5(d)] with contexts satisfying $c(2) \leq 5$ and $c(1) > 14$, whose interpretation was explained in Section III-C. Similar interpretations can be made for node 11 and other nodes. These results show that the context tree tries to capture the voice structure through the pitch context. As this is induced from data in an unsupervised way, it serves as an information-scientific confirmation that the voice structure has a strong influence on the configuration of note values.

2) Learning the Interdependence Model: The interdependence model for each δ_{nbh} can be directly learned from score data: for all note pairs defined by (15), one obtains the joint probability of their note values. The obtained results for $\delta_{\mathrm{nbh}} = 12$ is shown in Fig. 9 where the same labels are used as in Fig. 8. The diagonal elements, which have the largest probability in each row and column, clearly reflect the constraint of chordal notes having the same note values.

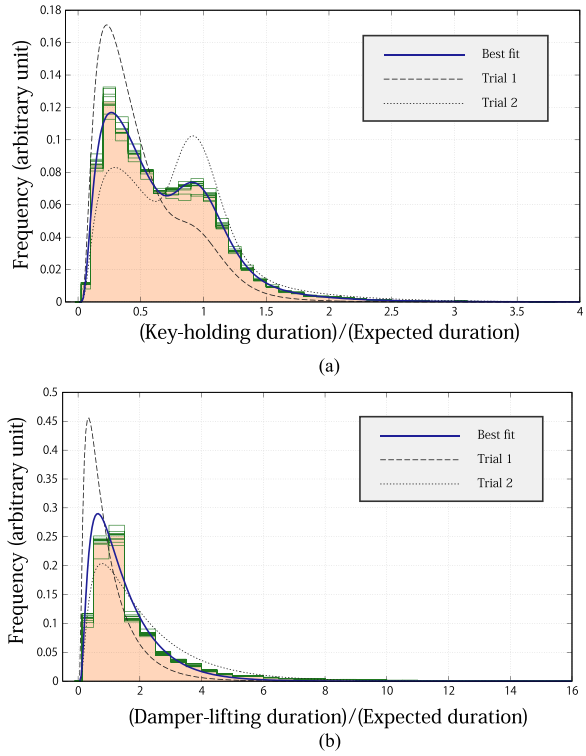

Fig. 10. Distributions used for the performance model for (a) key-holding durations and (b) damper-lifting durations. In each figure, the background histogram is the one obtained from the whole training data (same as Fig. 3) and the superposed histograms are obtained from 10-fold training datasets.

Since the interdependence model is by itself not as precise a generative model as the context model and these models are not independent, we optimise δ_{nbh} in combination with the context model. This is described in Section IV-D4, together with the optimisation of the weights. In preparation for this, we learned the joint probability for each of $\delta_{\mathrm{nbh}} = 0, 1, \ldots, 15$.

3) Learning the Performance Model: The parameters for the performance model in (18) and (19) are learned from the distributions given in Fig. 3. We performed a grid search for minimising the squared fitting error for each distribution. The obtained values are the following:

$$a_1 = 2.24 \pm 0.02, \quad b_1 = 0.24 \pm 0.01, \quad h_1 = 0.69 \pm 0.01,$$
$$a_2 = 13.8 \pm 0.1, \quad b_2 = 15.2 \pm 0.1, \quad h_2 = -1.22 \pm 0.04,$$
$$w_1 = 0.814 \pm 0.004, \quad w_2 = 0.186 \pm 0.004,$$
$$a_3 = 0.94 \pm 0.01, \quad b_3 = 0.51 \pm 0.01, \quad h_3 = 0.80 \pm 0.01.$$

The fitting curves are illustrated in Fig. 10. In the figure, we also show histograms of normalised durations obtained from ten different subsets of the training data that are constructed similarly as the 10-fold cross-validation method: i.e. we split the training data into ten separate sets (each containing 10% of the performances) and the remaining 90% of the data were used as one of the 10-fold training datasets. We can see in Fig. 10 that the differences among these histograms are not large. Two other parameter sets for g and $\bar{g}$ were chosen as trial distributions shown in the figure, which deviate from the best fit distribution more than the differences among the 10-fold histograms. These

distributions are used in Section V-C3 to examine the influence of the parameter values for the performance model.

4) Optimisation of the Weights: Since the three component models for the MRF model in (7) are not independent, the weights β should be obtained by simultaneous optimisation using performance data in general. However, since the amount of score data at hand is significantly larger than that of the performance data, we optimise the weights in a more efficient way. Namely, we first optimise β_1 and β_2 with the score data and then optimise β_{31} and β_{32} with the performance data (with fixed β_1 and β_2). When examining the influence of varying these weights in Section V-C, we will discuss that the influence of this sub-optimisation procedure is seemingly small.

We obtained the first two weights simultaneously with δ_{nbh} by the ML principle with the following results:

$$\hat{\beta}_1 = 0.965 \pm 0.005, \quad \hat{\beta}_2 = 0.03 \pm 0.005, \quad \hat{\delta}_{\mathrm{nbh}} = 12. \tag{24}$$

The result $\hat{\beta}_2 \ll \hat{\beta}_1$ indicates that the interdependence model has little influence in the score model. Although it seems somewhat contradictory to the results in Section IV-D2 at first sight, we can understand this by noticing that both the context model and interdependence model make use of pitch proximity to capture the voice structure. The former model uses pitch proximity in the horizontal (time) direction and the latter model does so in the vertical (pitch) direction, and they have overlapping effects since whenever a note pair (say, note n and n') in an onset cluster have close pitches, they tend to share notes in succeeding onset clusters with close pitches (see e.g. the chords in the left-hand part in the score in Fig. 16). Thus note n and n' tend to obey similar distributions in the context model. Since the interdependence model is weaker in terms of predictive ability, this results in small $\hat{\beta}_2$.

We optimised β_{31} and β_{32} according to the accuracy of note value recognition (more precisely, the average error rate defined in Section V-A) and the obtained values are as follows:

$$\hat{\beta}_{31} = 0.21 \pm 0.01, \quad \hat{\beta}_{32} = 0.003 \pm 0.001. \tag{25}$$

One can notice that $\hat{\beta}_{32} \ll \hat{\beta}_{31}$, which can be explained by the significantly larger variance of the distribution of damper-lifting durations than that of key-holding durations in Fig. 3. On the other hand, the result that $\hat{\beta}_{31}$ is considerably smaller than $\hat{\beta}_1$ can be interpreted as that the score model has more importance for estimating note values (in our model). The effect of varying weights is examined in Section V-C.

E. Inference Algorithm and Implementation

We can develop a note value recognition algorithm based on the maximisation of the probability in (7) with respect to $\boldsymbol{r}$. As a search space, we consider $\Omega_r(n)\backslash\{other\}$ for each r_n. Without H_2, the probability is independent for each r_n and the optimisation is straightforward. With H_2, we should optimise those r_n s connected in $\mathscr{N}$ simultaneously. Since there are only vertical interdependencies in our model, the optimisation can be done independently for each onset cluster. With J notes in an onset cluster, the set of candidate note values has size 10^J. Typically $J \leq 6$ for piano scores and the global search can be done directly. Occasionally, however, J can be ten or more and the computation time can be too large. To reduce the size of search space in this case, cutoffs are placed on the order of IONVs when $J > 6$ in our implementation: instead of the first ten IONVs, we use the first $(14 - J)$ IONVs for $6 < J \leq 10$ and two IONVs for $J > 10$. Although with this approximation we lose a certain proportion of possible solutions, we know that this proportion is small from the small probability of r having higher IONVs in Fig. 5(b).

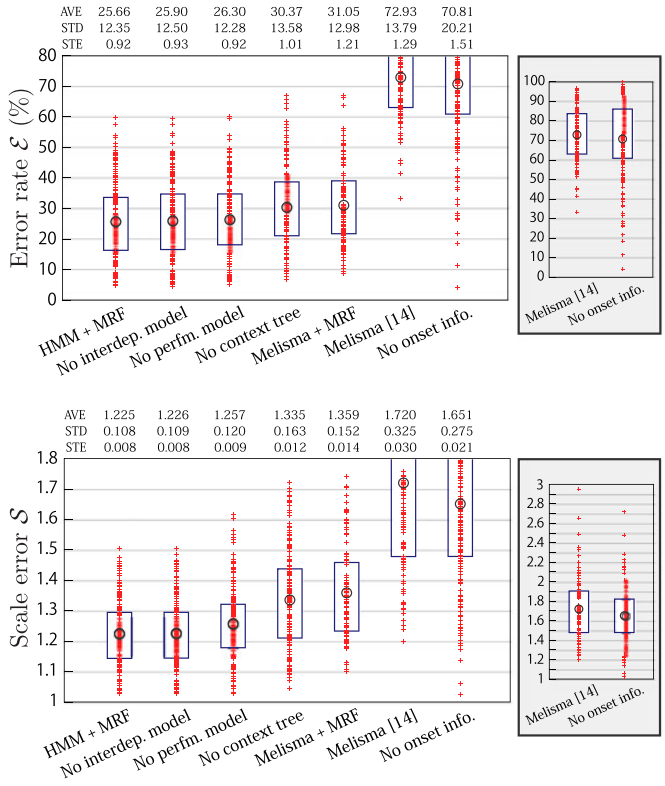

Fig. 11. Piece-wise average error rates and scale errors of note value recognition. Each red cross corresponds to one performance. The circle indicates the average (AVE), the blue box the range from the first to third quartiles, STD the standard deviation, and STE the standard error.

Our implementation of the MRF model and the metrical HMM for onset rhythm transcription and tempo estimation is available [27]. A tempo estimation algorithm based on a Kalman smoother is also provided for applying our method to results of other onset rhythm transcriptions that do not include tempo information as output.

V. Evaluation

A. Evaluation Measures

We first define evaluation measures used in our study. For each note $n = 1, \ldots, N$, let r_n^{c} and r_n^{e} be the correct and estimated note values. Then the *error rate* $\mathcal{E}$ is defined as

$$\mathcal{E} = \frac{1}{N} \sum_{n=1}^{N} \mathbb{I}(r_n^{\mathrm{e}} \neq r_n^{\mathrm{c}}) \tag{26}$$

where $\mathbb{I}(\mathcal{C})$ is 1 if condition $\mathcal{C}$ is true and 0 otherwise. This measure does not take into account how close the estimation is to the correct value when they are not exactly equal. Alternatively one

can consider the averaged 'distance' between the estimated and correct note values. As such a measure we define the following *scale error* $\mathcal{S}$:

$$\mathcal{S} = \exp\left[\frac{1}{N}\sum_{n}\left|\ln(r_n^{\mathrm{e}}/r_n^{\mathrm{c}})\right|\right]. \tag{27}$$

The difference and average is defined in the logarithmic domain to avoid bias for larger note values. $\mathcal{S}$ is unity if all note values are correctly estimated, and for example, $\mathcal{S} = 2$ if all estimations are doubled or halved from the correct values.

Because of the ambiguity of defining the beat unit, score times estimated by rhythm transcription methods often have doubled, halved or other scaled values [16], [35], which should not be treated as complete errors. To handle such scaling ambiguity, we normalise note values with the first IONV as

$$r_n'^{\mathrm{e}} = r_n^{\mathrm{e}}/\mathrm{IONV}^{\mathrm{e}}(n,1), \tag{28}$$

$$r_n'^{\mathrm{c}} = r_n^{\mathrm{c}}/\mathrm{IONV}^{\mathrm{c}}(n,1) \tag{29}$$

where $\mathrm{IONV}^{\mathrm{e}}(n,1)$ and $\mathrm{IONV}^{\mathrm{c}}(n,1)$ is the first IONV defined for the estimated and correct score, respectively. Scale-invariant evaluation measures can be obtained by applying (26) and (27) for $r_n'^{\mathrm{e}}$ and $r_n'^{\mathrm{c}}$.

B. Comparative Evaluations

In this section, we evaluate the proposed method, a previously studied method [14] and a simple model discussed in Section III on our data set and compare them in terms of the accuracy of note value recognition.

1) Setup: To study the contribution of the component models of our MRF model, we evaluated the full model, a model without the interdependence model ($\beta_2 = 0$), a model without the performance model ($\beta_{31} = \beta_{32} = 0$) and an MRF model with a context model having no (or a trivial) context tree, all applied to the result of onset rhythm transcription by the metrical HMM. For the metrical HMM, we use the parameter values taken from a previous study [16]. These parameters were learned with the same score data and different performance data.

In addition, we evaluated a method based on a simple prior distribution on note values [Fig. 5(a)] combined with an output probability $P(d_n; r_n, v_n)$ in (16), which uses no information of onset score times. For comparison, we evaluated the Melisma Analyzer (version 2) [14], which is to our knowledge the only major method that can estimate onset and offset score times, and we also applied post-processing by the proposed method on the onset score times obtained by the Melisma Analyzer. The used data is described in Section IV-D.

2) Results: The piece-wise average error rates and scale errors are shown in Fig. 11 where the mean (AVE) over all pieces and the standard error for the mean (corresponding to 1σ deviation in the t-test) are also given. Out of the 180 performances, only 115 performances were properly processed by the Melisma Analyzer and are depicted in the figure. In addition, 30.0% of the note values estimated by the method were zero and scale errors were calculated without these values. One can see that the Melisma Analyzer and the simple model without using the onset score time information have high error rates and the proposed methods clearly outperformed them.

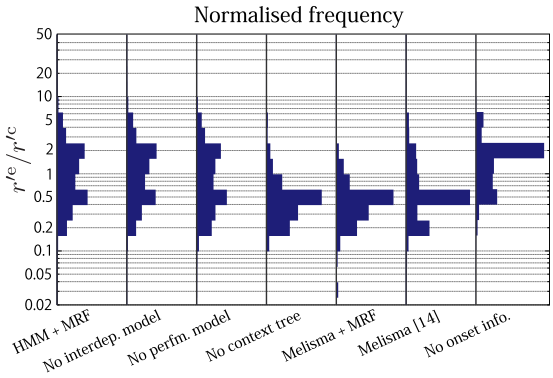

Fig. 12. Distributions of note-wise scale errors $r'^{\mathrm{e}}/r'^{\mathrm{c}}$ for notes with $r'^{\mathrm{e}}/r'^{\mathrm{c}} \neq 1$.

The distributions of note-wise scale errors $r'^{\mathrm{e}}/r'^{\mathrm{c}}$ for incorrect estimations ($r'^{\mathrm{e}}/r'^{\mathrm{c}} \neq 1$) in Fig. 12 show that the Melisma Analyzer (simple model) more often estimates note values shorter (longer) than the correct ones. For the simple model, this is because it mostly relies on, other than a relatively weak prior distribution in Fig. 5(a), the distribution of key-holding durations in Fig. 3(a), which has the highest peak position lower than its mean. For the Melisma Analyzer, the short and zero note values arise because the method quantises the onset and (key-release) offset times into analysis frames of 50 ms. Whereas the comparison is not fair in that the Melisma Analyzer can potentially identify grace notes with zero note values, which our data did not contain and our method cannot recognise, the rate (30.0%) is considerably higher than their typical frequency in piano scores.

Among the different conditions for the proposed method, the full model had the best accuracy and the case with no context tree had significantly worse results, showing a clear effect of the context model. Compared to the full model, the average error rate for the model without the performance model was worse but within 1σ deviation and the average scale error was significantly worse, indicating that the performance model has an effect in approximating the estimated note values to the correct ones. Results without the interdependence model were slightly worse but almost the same as the full model, which is because of the small $\hat{\beta}_2$. The last result indicates that one can remove the interdependence model without much increase of estimation errors, which simplifies the inference algorithm as the distributions of note values become independent for each note.

C. Examining the Proposed Model

Here we examine the proposed model in greater depth.

1) Error Analyses: To examine the effect of the component models, let us look at the distribution of the estimated note values in the space of IONVs (Fig. 13). Note that the distribution for the ground truth is essentially the same as that in Fig. 5(b) but slightly different because the data is different and the onset clusters here are defined with the result of onset rhythm transcription by the metrical HMM.

Firstly, the model without a context tree assigns the first IONV to note values with a high probability ($>98\%$), indicating that estimated results by the model are almost the same as for the one-voice representation in Fig. 2(b). This is consistent with the

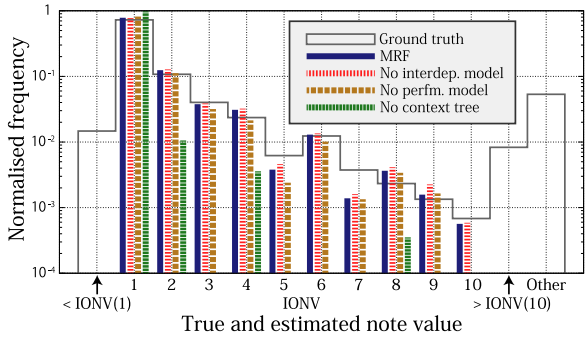

Fig. 13. Distributions of true and estimated note values relative to IONVs.

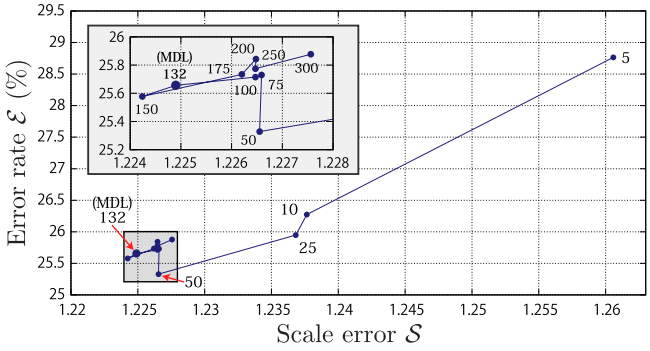

Fig. 14. Average error rates and scale errors for various sizes of the context tree. The figure close to each point indicates the number of leaves. 132 is the optimal number predicted by the MDL criterion. All data points have statistical errors of order 1% for error rate and order 0.01 for scale error.

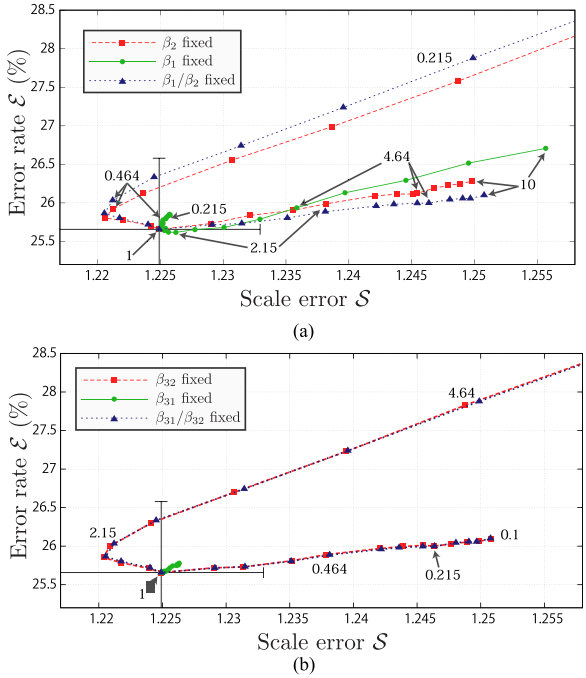

Fig. 15. Average error rates and scale errors with (a) varying β_1 and β_2 and (b) varying β_{31} and β_{32}. The β s are scaled in logarithmically equally spaced scaling factors, which are partly indicated by numbers, and the centre values (indicated by '1') are given in (24) and (25). All data points have statistical errors of order 1% for error rate and order 0.01 for scale error.

results in Fig. 12 that this model tends to estimate note values shorter than the correct values. Secondly, one can notice that the model without the performance model has a higher probability for the first IONV and smaller probabilities for most of the later IONVs compared with the full model. This suggests that the performance model uses the information of actual durations to correct (or better approximate) the estimated note values more frequently to larger values, leading to decreased scale errors. Finally, the proportion of errors corresponding to note values that are larger than $\mathrm{IONV}(10)$ is about 0.8%, indicating that the effect of enlarging the search space of note values by including higher IONVs is limited.

2) Influence of the Context-Tree Size and Weights: Fig. 14 shows the average error rates and scale errors for various sizes of the context tree. The case with only one leaf (not shown in the figure) is the same as the case without a context tree explained above. The errors rapidly decreased as the tree size increased for small numbers of leaves and but changed only slightly above 50 leaves. There was a gap between the error rates for the cases with 50 and 75 leaves, which we confirmed is caused by a discontinuity of results for 52 and 53 leaves. We have not succeeded in finding a good explanation for this gap. Far above the predicted value (132 leaves) by the MDL criterion, the errors tended to increase slightly, confirming that it is close to the optimal choice.

Fig. 15 shows the average error rate and scale error when varying the weights from the values in (24) and (25). The context tree had 132 leaves. First, variations by increasing and decreasing the weights by 50% are within 1σ statistical significance, showing that the error rates are not very sensitive to these parameters. Second, the values $\hat{\beta}_1$ and $\hat{\beta}_2$, which were optimised based on ML using the score data, are found to be optimal with respect to the error rate. Finally, the similar shapes of the curves when fixing β_1/β_2 and fixing β_{31}/β_{32} show that their relative values influences the results more than their absolute values in the examined region. The results together with the large-variance nature of the distributions of durations in Fig. 3 suggest that it is likely that more elaborate fitting functions for the performance model would not improve the results significantly and also that the sub-optimisation procedure for β s described in Section IV-D4 did not deteriorate the results much.

3) Influence of the Parameters of the Performance Model: To examine the influence of the parameter values of the performance model in (18) and (19), we run the proposed model for each of three distributions shown in Fig. 10(a) and (b). The other parameters were set to the optimal values and the size of the context tree was 132. Results in Table II show that despite the differences among distributions, the average scale error was almost constant and the variation of the average error rate is also smaller than the standard error. More precisely, the influence of the choice of parameters for $\bar{g}$ is negligible, which can be explained by the small value of β_{32}. This confirms that the influence of the performance model is small and there is little effect of overfitting in using the test data for learning.

4) Example Result: Let us discuss an example[3] in Fig. 16, which has a typical texture of piano music with the left-hand

[3]Sound files are available at the accompanying web page [27].

TABLE II
AVERAGE ERROR RATES AND SCALE ERRORS FOR DIFFERENT DISTRIBUTIONS FOR THE PERFORMANCE MODEL

Key-holding g	Damper-lifting $\bar{g}$	Error rate $\mathcal{E}$ (%)	Scale error $\mathcal{S}$
Best fit	Best fit	25.66	1.225
Best fit	Trial 1	25.67	1.225
Best fit	Trial 2	25.67	1.225
Trial 1	Best fit	25.97	1.225
Trial 1	Trial 1	25.98	1.225
Trial 1	Trial 2	25.97	1.225
Trial 2	Best fit	25.46	1.225
Trial 2	Trial 1	25.46	1.225
Trial 2	Trial 2	25.46	1.225

The best fit and trial distributions are shown in Fig. 10.

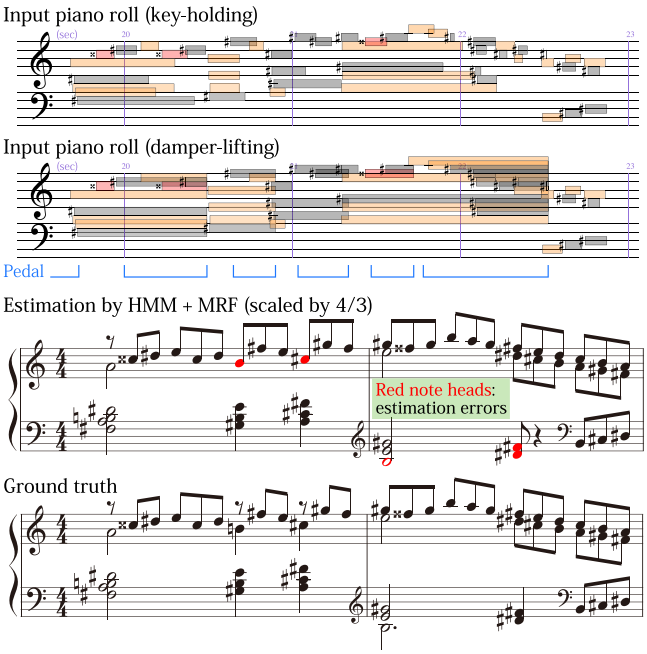

Fig. 16. Example result of rhythm transcription by the metrical HMM and the proposed MRF model (Beethoven: Waldstein sonata 1st mov.). Voice, staff and time signature are added manually to the estimated result for the purpose of this illustration.

part having harmonising chords and the right-hand part having melodic notes, both of which have multiple voices inside. By comparing the performed durations to the score, we can see that overall the damper-lifting durations are closer to the score-indicated durations for the left-hand notes and the key-holding durations are closer for the right-hand notes. This is because pianists tend to lift the pedal when harmonising chords change. This example shows that the two types of durations provide complementary information and one should not rely on one of them. On the other hand, for most notes, the offset score time matches to the onset score time of a succeeding note with a close pitch, which is what our context model describes.

The result by the MRF model shows that the model uses the score and performance models complementarily to find the optimal estimation. The correctly estimated half notes (as $\mathrm{IONV}(6)$), A4 in the first bar and E5 in the second bar, have a close pitch in the next onset cluster and the incorrect estimates as $\mathrm{IONV}(1)$ are avoided by using the duration (and perhaps because of the existence of very close pitches at the sixth next onset clusters). On the other hand, the quarter-note F#4 and D#4 in the left-hand part in the second bar could not be correctly estimated probably because the voice makes a big leap here, closer notes in the right-hand part succeed them and the key-holding durations are short.

VI. CONCLUSION AND DISCUSSION

We discussed note value recognition of polyphonic piano music based on an MRF model combining the score model and the performance model. As suggested in the discussion in Section III and confirmed by evaluation results, performed durations can deviate greatly from the score-indicated lengths and thus the performance model aline has little predictive ability. The construction of the score model is then the key to solve the problem. We formulated a context-tree model that can learn highly predictive distributions of note values from data, using onset score times and the pitch context. It was demonstrated that this score model brings significant improvements on the recognition accuracy.

Refinement of the score model is possible in a number of ways. Using more features for the context-tree model could improve the results. Using other feature-based model learning schemes such as deep neural networks are similarly possible. The refinement and extension of the search space for note values is another issue since the set of the first ten IONVs used in this study loses a certain proportion of solutions. The result that the context-tree model learned to capture the voice structure suggests that building a model with explicit voice structure is also interesting for creating generative models to reduce reliance on arbitrarily chosen features.

Remaining issues to obtain musical scores in a fully automatic way include the assignment of voice and staff to the transcribed notes. Voice separation methods and staff estimation methods exist (e.g. [29]–[31]) and the information of transcribed note values can be useful to identify chordal notes within each voice. Another issue is the recognition of time signature. Using multiple metrical HMMs learned with score data for each metres is one possibility and we could also apply other metre detection methods (e.g. [36]) to the transcribed result.

To apply this work, the construction of a complete polyphonic music transcription system from audio signals to musical scores is attractive. The framework developed in this study can be combined with existing multi-pitch analysers [3]–[6] for this purpose. It is worth mentioning that the performance model should be trained on piano rolls obtained with these methods since the distribution of durations would differ from that of recorded MIDI signals. Extension of the model to correct audio transcription errors such as note insertions and deletions would also be of great importance.

ACKNOWLEDGMENT

We are grateful to D. Temperley for providing source code for the Melisma Analyzer. E. Nakamura would like to thank S. Takaki for useful discussions about context-tree clustering.

REFERENCES

[1] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*. New York, NY, USA: Springer, 2006.

[2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: Challenges and future directions," *J. Intell. Inf. Syst.*, vol. 41, no. 3, pp. 407–434, 2013.

[3] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 3, pp. 528–537, Mar. 2010.

[4] K. O'Hanlon and M. D. Plumbley, "Polyphonic piano transcription using non-negative matrix factorisation with group sparsity," in *Proc. Int. Conf. Acoust. Speech, Signal Process.*, pp. 2214–2218, 2014.

[5] K. Yoshii, K. Itoyama, and M. Goto, "Infinite superimposed discrete all-pole modeling for source-filter decomposition of wavelet spectrograms," in *Proc. Int. Soc. Music Inf. Retrieval*, pp. 86–92, 2015.

[6] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 5, pp. 927–939, May 2016.

[7] H. Longuet-Higgins, *Mental Processes: Studies in Cognitive Science*. Cambridge, MA, USA: MIT Press, 1987.

[8] P. Desain and H. Honing, "The quantization of musical time: A connectionist approach," *Comput. Music J.*, vol. 13, no. 3, pp. 56–66, 1989.

[9] C. Raphael, "A hybrid graphical model for rhythmic parsing," *Artif. Intell.*, vol. 137, pp. 217–238, 2002.

[10] H. Takeda, T. Otsuki, N. Saito, M. Nakai, H. Shimodaira, and S. Sagayama, "Hidden Markov model for automatic transcription of MIDI signals," in *Proc. Multimedia Signal Process.*, 2002, pp. 428–431.

[11] M. Hamanaka, M. Goto, H. Asoh, and N. Otsu, "A learning-based quantization: Unsupervised estimation of the model parameters," in *Proc. Int. Comput. Music Conf.*, 2003, pp. 369–372.

[12] A. Cemgil and B. Kappen, "Monte Carlo methods for tempo tracking and rhythm quantization," *J. Artif. Intell. Res.*, vol. 18, no. 1, pp. 45–81, 2003.

[13] E. Kapanci and A. Pfeffer, "Signal-to-score music transcription using graphical models," in *Proc. Int. Joint Conf. Artif. Intell.*, 2005, pp. 758–765.

[14] D. Temperley, "A unified probabilistic model for polyphonic music analysis," *J. New Music Res.*, vol. 38, no. 1, pp. 3–18, 2009.

[15] M. Tsuchiya, K. Ochiai, H. Kameoka, and S. Sagayama, "Probabilistic model of two-dimensional rhythm tree structure representation for automatic transcription of polyphonic MIDI signals," in *Proc. Asia-Pacific Signal Inf. Process. Assoc.*, 2013, pp. 1–6.

[16] E. Nakamura, K. Yoshii, and S. Sagayama, "Rhythm transcription of polyphonic piano music based on merged-output HMM for multiple voices," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 4, pp. 794–806, Apr. 2017.

[17] S. Dixon and E. Cambouropoulos, "Beat tracking with musical knowledge," in *Proc. Eur. Conf. Artif. Intell.*, 2000, pp. 626–630.

[18] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *J. New Music Res.*, vol. 30, no. 1, pp. 39–58, 2001.

[19] G. Peeters and H. Papadopoulos, "Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 6, pp. 1754–1769, Aug. 2011.

[20] F. Krebs, A. Holzapfel, A. T. Cemgil, and G. Widmer, "Inferring metrical structure in music using particle filters," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 5, pp. 817–827, May 2015.

[21] S. Durand, J. P. Bello, B. David, and G. Richard, "Downbeat tracking with multiple features and deep neural networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 409–413.

[22] H. Kameoka, K. Ochiai, M. Nakano, M. Tsuchiya, and S. Sagayama, "Context-free 2D tree structure model of musical notes for bayesian modeling of polyphonic spectrograms," in *Proc. Int. Soc. Music Inf. Retrieval*, 2012, pp. 307–312.

[23] S. Raczynski, E. Vincent, and S. Sagayama, "Dynamic Bayesian networks for symbolic polyphonic pitch modeling," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 9, pp. 1830–1840, Sep. 2013.

[24] S. Young, J. J. Odell, and P. Woodland, "Tree-based state tying for high accuracy acoustic modelling," *Proc. Human Lang. Technol.*, 1994, pp. 307–312.

[25] S. Takaki, Y. Nankaku, and K. Tokuda, "Contextual additive structure for HMM-based speech synthesis," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 2, pp. 229–238, Apr. 2014.

[26] K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," *J. Acoust. Soc. Jpn E*, vol. 21, no. 2, pp. 79–86, 2000.

[27] E. Nakamura, K. Yoshii, and S. Dixon, "Note value recognition for polyphonic music," 2017. [Online]. Available: http://anonymous574868.github.io/demo.html, Accessed on: Mar. 18, 2017.

[28] S. Levinson, L. Rabiner, and M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *The Bell Sys. Tech. J.*, vol. 62, no. 4, pp. 1035–1074, 1983.

[29] E. Cambouropoulos, "Voice and stream: Perceptual and computational modeling of voice separation," *Music Perception*, vol. 26, no. 1, pp. 75–94, 2008.

[30] A. McLeod and M. Steedman, "HMM-based voice separation of MIDI performance," *J. New Music Res.*, vol. 45, no. 1, pp. 17–26, 2016.

[31] E. Nakamura, N. Ono, and S. Sagayama, "Merged-output HMM for piano fingering of both hands," in *Proc. Int. Soc. Music Inf. Retrieval*, 2014, pp. 531–536.

[32] E. Nakamura, K. Itoyama, and K. Yoshii, "Rhythm transcription of MIDI performances based on hierarchical Bayesian modelling of repetition and modification of musical note patterns," in *Proc. EUSIPCO*, 2016, pp. 1946–1950.

[33] School of Music, University of Minesota, *International Piano e-Competition*. [Online]. Available: http://www.piano-e-competition.com/ecompetition/default.asp, Accessed on: Feb. 24, 2017.

[34] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. Inf. Theory*, vol. IT-30, no. 4, pp. 629–636, 1984.

[35] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing, "On tempo tracking: Tempogram representation and Kalman filtering," *J. New Music Res.*, vol. 29, no. 4, pp. 259–273, 2000.

[36] W. B. de Haas and A. Volk, "Meter detection in symbolic music using inner metric analysis," in *Proc. Int. Soc. Music Inf. Retrieval*, 2016, pp. 441–447.

Eita Nakamura (M'15) received the Ph.D. degree in physics from the University of Tokyo, Tokyo, Japan, in 2012. After having been a Postdoctoral Researcher at the National Institute of Informatics, Meiji University and Kyoto University, he is currently a JSPS Research Fellow in the Speech and Audio Processing Group at Kyoto University, Kyoto, Japan. His research interests include music modelling and analysis, music information processing, and statistical machine learning.

Kazuyoshi Yoshii (M'05) received the Ph.D. degree in informatics from Kyoto University, Kyoto, Japan, in 2008. He is currently a Senior Lecturer at Kyoto University. His research interests include music signal processing and machine learning. He is a member of the Information Processing Society of Japan and Institute of Electronics, Information, and Communication Engineers.

Simon Dixon received the Ph.D. degree in computer science and the LMusA diploma in classical guitar. He is a Reader (Associate Professor), the Director of Graduate Studies, and the Deputy Director of the Centre for Digital Music at Queen Mary University of London, London, U.K. His research interests include high-level music signal analysis, computational modelling of musical knowledge, and the study of musical performance. Particular areas of focus include automatic music transcription, beat tracking, audio alignment, and analysis of intonation and temperament. He was the President (2014–2015) of the International Society for Music Information Retrieval (ISMIR), is the founding Editor of the *Transactions of International Society for Music Information Retrieval*, and member of the Editorial Board of the *Journal of New Music Research* (since 2011), and has published over 160 refereed papers in the area of music informatics.