

Real-Time Audio-to-Score Alignment of Music Performances Containing Errors and Arbitrary Repeats and Skips

Tomohiko Nakamura, *Student Member, IEEE*, Eita Nakamura, *Member, IEEE*, and Shigeki Sagayama, *Member, IEEE*

Abstract—This paper discusses real-time alignment of audio signals of music performance to the corresponding score (a.k.a. score following) which can handle tempo changes, errors and arbitrary repeats and/or skips (repeats/skips) in performances. This type of score following is particularly useful in automatic accompaniment for practices and rehearsals, where errors and repeats/skips are often made. Simple extensions of the algorithms previously proposed in the literature are not applicable in these situations for scores of practical length due to the problem of large computational complexity. To cope with this problem, we present two hidden Markov models of monophonic performance with errors and arbitrary repeats/skips, and derive efficient score-following algorithms with an assumption that the prior probability distributions of score positions before and after repeats/skips are independent from each other. We confirmed real-time operation of the algorithms with music scores of practical length (around 10000 notes) on a modern laptop and their tracking ability to the input performance within 0.7 s on average after repeats/skips in clarinet performance data. Further improvements and extension for polyphonic signals are also discussed.

Index Terms—Arbitrary repeats and skips, audio-to-score alignment, fast Viterbi algorithm, hidden Markov model, music signal processing, score following.

I. INTRODUCTION

REAL-TIME alignment of an audio signal of a music performance to a given score, also known as score following, has been gathering attention since its first appearance in 1984 [1], [2]. Score following is a basic technique for real-time

Manuscript received March 11, 2015; revised September 22, 2015; accepted December 04, 2015. Date of publication December 10, 2015; date of current version January 05, 2016. The work of T. Nakamura was supported in part by the Japan Society for the Promotion of Science (JSPS) Research Fellowships for Young Scientists No. 15J0992. The work of E. Nakamura was supported by the JSPS under Grant-in-Aid 15K16054. The work of S. Sagayama was supported by the JSPS under Grant-in-Aid 26240025. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Matthew Davies.

T. Nakamura is with the Department of Information Physics and Computing, Graduate School of Information Science and Technology, University of Tokyo, Tokyo 113-8656, Japan (e-mail: tomohiko_nakamura@ipc.i.u-tokyo.ac.jp).

E. Nakamura is with the Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan (e-mail: enakamura@sap.ist.i.kyoto-u.ac.jp).

S. Sagayama is with the University of Tokyo, Tokyo, 113-8656, Japan, and also with the School of Interdisciplinary Mathematical Sciences, Meiji University, Tokyo 164-8525, Japan (e-mail: sagayama@meiji.ac.jp).

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.2015.2507862

musical applications such as automatic accompaniment, automatic score page-turning [3] and automatic captioning to music videos. The technique is particularly essential for automatic accompaniment, which synchronizes an accompaniment to a performer on the fly, referring to performance and accompaniment scores. Automatic accompaniment enables live performance of ensemble music by one or a few performers. Many studies of score following have been carried out (see [4] for a review and [5]–[13] for recent progress).

Automatic accompaniment is particularly useful for practices, rehearsals and personal enjoyment of ensemble music. In these situations, performers often make errors. Moreover, performers may want to start playing from the middle of a score and generally make repeats and/or skips (repeats/skips). Since errors and repeats/skips are hard to predict, a score-following algorithm capable of handling arbitrary errors and repeats/skips is necessary to realize an automatic accompaniment system effective in those situations. Our aim is to develop such an algorithm.

Treatment of errors in score following is discussed in some studies [4], [5], [13], [14]. However, a detailed discussion and a systematic evaluation of the effectiveness of the methods for audio score following have not been given in the literature. Score-following algorithms that can follow repeats/skips have been proposed in [5], [11], [15]. The targets of these algorithms are predetermined repeats/skips from and to specific score positions, and treatment of arbitrary repeats/skips is not discussed nor guaranteed. In fact, as we will show in this paper, simple extensions of these algorithms have the problem of large computational cost and cannot work in real time for long scores of practical length. Unless the problem is solved, score-following systems can only work with limited scores with very short length or we must give up following arbitrary repeats/skips as most of the current systems do, both of which sacrifice the vast potential application of score following. Therefore, it is essential to reduce the computational complexity to follow arbitrary repeats/skips.

The authors have presented a new type of hidden Markov model (HMM) that describes musical instrument digital interface (MIDI) performances with errors and arbitrary repeats/skips, and derived a computationally efficient algorithm for the HMM [13]. It reduces the computational complexity with an assumption to simplify a probability distribution of score positions before and after repeats/skips. While a similar model would be applicable to the audio case, further discussions are required since audio inputs (frame-wise discrete in

time and continuous in features) significantly differ with MIDI inputs (continuous in time and discrete in pitches) in nature.

The main contribution of this paper is to present real-time algorithms that can follow monophonic audio performances containing arbitrary repeats/skips and errors. Although monophonic score following has been addressed since [1], [2], arbitrary repeats/skips have never been discussed despite the practical importance of their treatment as the above mentioned. Because polyphonic score following is still an active field of research and the extension of the present method for polyphonic performances requires many additional issues discussed in Section V, we confine ourselves to monophonic performances.

We develop a model of music performances containing errors and arbitrary repeats/skips with an HMM. We first discuss how various types of errors can be incorporated into the model (Section II). Next, we extend the model to incorporate arbitrary repeats/skips. In order to solve the problem of large computational cost for following arbitrary repeats/skips, two HMMs with refined topologies are presented. We derive efficient score-following algorithms with reduced computational complexity based on both HMMs (Section III). We demonstrate that both algorithms can work in real time with scores of practical length on a modern laptop computer and are effective in following performances with errors and arbitrary repeats/skips through evaluations using clarinet performances during practice (Section IV). We discuss possible improvements and extensions of the proposed algorithms for polyphonic inputs (Section V). Part of this study (Section III and a part of Section IV) was reported in our previous conference paper [12].

II. SCORE FOLLOWING FOR PERFORMANCES WITH ERRORS

A. Variety in Audio Performance and Statistical Approach

Score following is generally challenging since audio signals of music performances widely vary even if the same score is used. Four typical sources of variety in monophonic audio performance are listed below.

- (a) **Acoustic variations:** Spectral features of audio performances depend on musical instruments and are not stationary. In addition, audio performances usually include noise caused by the surrounding environment and musical instruments (e.g. resonance, background noise, breath noise and other acoustics).
- (b) **Temporal fluctuations:** The tempo of the performance and onset times and durations of performed notes deviate from those indicated in scores due to performer's skills, physical limitations of musical instruments and musical expressions. For example, performances during practice are often rendered in slow tempo to avoid errors.
- (c) **Performance errors:** Performers may make errors due to lack of performance skills or mis-readings of the score. Errors are categorized into pitch errors (substitution errors), dropping notes (deletion errors), adding extra notes (insertion errors) [1]. Besides, performers may make pauses between notes, for example, to turn a page of the score and to check the next note.

- (d) **Repeats/skips:** Performers may repeat and/or skip phrases in particular during practice. Furthermore, the performers generally add or delete a repeated section.

These four sources of variety in monophonic audio performance make score following difficult and motivate us to study it. In particular, it is essential to adapt automatic accompaniment systems to the variety in order to keep synchronization to live performances. Although it is out of the scope of this paper, there are other sources of variety in music performance such as ornaments [6], [13], [16], [17] and improvisation [18], [19].

Recent score-following systems commonly use probabilistic models such as HMM to capture the variety of audio performances, and their effectiveness has been well confirmed [4] (and references in the Introduction). They are particularly advantageous to capture continuous variations of audio features and to handle errors which are hard to predict. Therefore, we take the statistical approach in this study.

B. Performance HMM

We represent the performance score with N musical events, each of which is a note or a rest. A performer reads the score from event to event and keeps making a sound corresponding to an event. This process of performance can be modeled with a hierarchical HMM with two levels [20], [21], which we call the performance HMM. The top level describes the progression of performed events, and the bottom level expresses temporal structure of the audio signal in a performed event.

Events correspond to states (top states) of the top-level HMM (top HMM), and the performance is described as transitions between the top states. Let $z_t^{(\text{top})} = 0, \dots, N - 1$ denote the random variable describing the top state at the t th frame ($t = 0, \dots, T - 1$), and let i and j label a top state. The top HMM is parameterized by state transition probabilities $a_{j,i}$ and initial probabilities π_i :

$$a_{j,i} := P(z_t^{(\text{top})} = i | z_{t-1}^{(\text{top})} = j), \quad (1)$$

$$\pi_i := P(z_0^{(\text{top})} = i), \quad (2)$$

which satisfy $\sum_{i=0}^{N-1} \pi_i = 1$ and $\sum_{i=0}^{N-1} a_{j,i} = 1$ for all j .

Each top state is itself an HMM (bottom HMM), whose states (bottom states) correspond to subevents in an event, for example, sustain of an instrumental sound, pauses between notes, etc. Let L denote the number of bottom states in the top state, $z_t^{(\text{bot})} = 0, \dots, L - 1$ denote the random variable describing the bottom state at the t th frame, and let l and l' label a bottom state. The state transitions of the bottom HMM are characterized by three kinds of probabilities. The initial probability $\pi_l^{(i)}$ describes the probability of a transition to bottom state l when top state i is entered, the exiting probability $e_l^{(i)}$ describes the probability of exiting top state i from bottom state l , and the transition probability $a_{l',l}^{(i)} := P(z_t^{(\text{bot})} = l | z_{t-1}^{(\text{bot})} = l')$ represents the transition from bottom state l' to bottom state l in top state i . These probabilities satisfy $\sum_{l'=0}^{L-1} \pi_{l'}^{(i)} = 1$ and $\sum_{l'=0}^{L-1} a_{l',l}^{(i)} + e_l^{(i)} = 1$ for all l and i . Thus, the performance is modeled as a sequence of T pairs of random variables $\{(z_t^{(\text{top})}, z_t^{(\text{bot})})\}_{t=0}^{T-1}$ (Fig. 1).

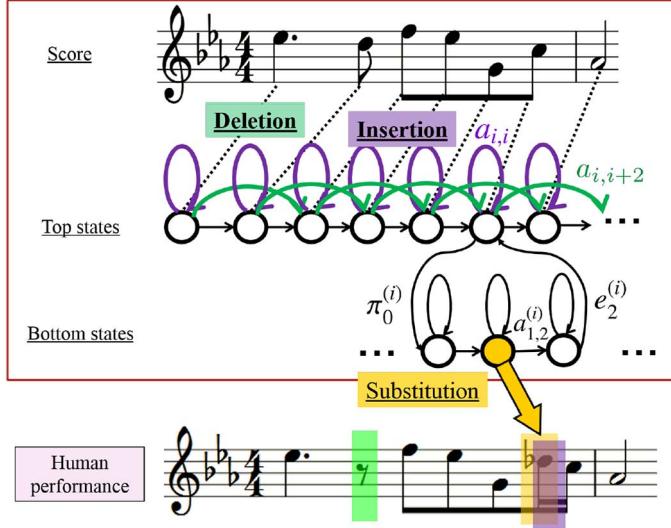


Fig. 1. A hierarchical hidden Markov model with two levels that describes a music performance with deletion, insertion and substitution errors. See text.

For example, if the pair $z_t := (z_t^{(\text{top})}, z_t^{(\text{bot})})$ equals to (i, l) , the score position at frame t is at bottom state l of top state i .

Observed audio features are described as being stochastically generated from a bottom state. Given an audio feature $\mathbf{y}_t := [y_{t,0}, y_{t,1}, \dots, y_{t,D-1}]^\top$ at frame t as a D -dimensional real vector, the emission probability of state (i, l) is defined as

$$b_l^{(i)}(\mathbf{y}_t) := P(\mathbf{y}_t | z_t = (i, l)). \quad (3)$$

C. Emission Probability and Substitution Error

From here to Section II-E, we consider the performance HMM with $L = 1$ for simplicity, but the case for $L > 1$ can be treated similarly. To extract pitch information from the input signal, we need a suitable feature representation. In the comparison of some audio features in [7], [22], the magnitude of a constant-Q transform (CQT) [23] with a quality factor set to one semitone yielded the best result of score following for monophonic audio input. Furthermore, normalizing magnitudes of CQTs such that $\sum_{d=0}^{D-1} y_d = 1$ makes them insusceptible to dynamic variations. Although one may think that the normalization makes it difficult to discriminate pauses from notes, the difference in spectral shape between pauses and notes can help the discrimination: The CQT of a pitched sound have clear peaks at its fundamental frequency and harmonics, whereas the CQTs at pauses are relatively flat. We use normalized magnitudes of CQTs (normalized CQTs) as audio features.

Let k be the pitch index and \mathcal{K} be the set of possible pitches. For convenience, we indicate the pitches A0 to C8 in the range of a standard piano as $k = 21$ to $k = 108$ and silence as $k = -1$, and $\mathcal{K} = \{21, 22, \dots, 108\} \cup \{-1\}$. We assume that normalized CQTs corresponding to pitch k follow a D -dimensional normal distribution with mean $\boldsymbol{\mu}_k$ and covariance matrix Σ_k , denoted by $\mathcal{N}(\mathbf{y}_t | \boldsymbol{\mu}_k, \Sigma_k)$. The emission probability $b_0^{(i)}(\mathbf{y}_t)$ of bottom state 0 of top state i is given as

$$b_0^{(i)}(\mathbf{y}_t) = \sum_{k \in \mathcal{K}} w_{k,0}^{(i)} \mathcal{N}(\mathbf{y}_t | \boldsymbol{\mu}_k, \Sigma_k). \quad (4)$$

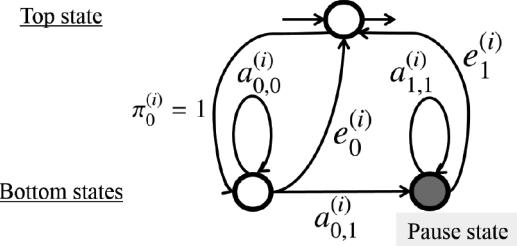


Fig. 2. A pause between notes is described with the pause state (gray disk) which emits audio features corresponding to silence.

Here $w_{k,0}^{(i)} \in [0, 1]$ is a mixture weight of pitch k of bottom state 0 of top state i , which satisfies $\sum_{k \in \mathcal{K}} w_{k,0}^{(i)} = 1$ for all i .

When substitution errors are not made, $w_{k,0}^{(i)} = 0$ unless $k = p_i$, where $p_i \in \mathcal{K}$ denotes the pitch of event i ($p_i = -1$ for a rest). On the other hand, to describe a performance with substitution errors, we have small positive values of $w_{k,0}^{(i)}$ for $k \neq p_i$ since a substitution error is represented by an emission of an audio feature with an incorrect pitch.

D. Transition Probability and Deletion and Insertion Errors

Transition probabilities in the top level $a_{j,i}$ represent the frequency of the transitions between the events. If performances do not contain insertion and deletion errors, $a_{j,i} = 0$ unless $i = j + 1$. We can express an insertion error and a deletion error with a self transition and a transition to the second next top state, which correspond to $a_{j,j}$ and $a_{j,j+2}$.

The self-transition probability $a_{0,0}^{(i)}$ of bottom state 0 of top state i describes the expected duration of the corresponding event d_i , which is computed as a product of the note value of the event and the score-notated tempo:

$$d_i = \sum_{k=1}^{\infty} k \left(a_{0,0}^{(i)} \right)^{k-1} \left(1 - a_{0,0}^{(i)} \right) = \frac{1}{1 - a_{0,0}^{(i)}}. \quad (5)$$

If d_i is shorter than a processing time interval, we put $a_{0,0}^{(i)} = 0$. This probabilistic representation of the event duration describes the temporal fluctuations of music performance.

E. Pauses Between Notes

Pauses between notes can be introduced into the performance HMM by adding an extra bottom state with index 1, which we call a pause state (Fig. 2). The occurrence of the pause is expressed as a transition to the pause state, which corresponds to $a_{0,1}^{(i)}$. The duration of the extra pause is represented by the self-transition probability of the pause state $a_{1,1}^{(i)}$, which can be set similarly to Eq. (5). We put $a_{1,0}^{(i)} = 0$ and $\pi_1^{(i)} = 0$ for all i . We assume that $b_1^{(i)}(\mathbf{y}_t) = \mathcal{N}(\mathbf{y}_t | \boldsymbol{\mu}_{-1}, \Sigma_{-1})$.

F. Estimation of Score Positions

For the convenience of estimating score positions, we convert the performance HMM into an equivalent standard HMM. Its state corresponds to a bottom state of the performance HMM and is labeled with (i, l) . The standard HMM is parameterized by emission probabilities $\tilde{b}_{(i,l)}(\mathbf{y}_t)$, initial probabilities $\tilde{\pi}_{(i,l)}$,

and transition probabilities $\tilde{a}_{(j,l'),(i,l)}$, defined by $\tilde{b}_{(i,l)}(\mathbf{y}_t) := b_l^{(i)}(\mathbf{y}_t)$, $\tilde{\pi}_{(i,l)} := \pi_i \pi_l^{(i)}$, and

$$\tilde{a}_{(j,l'),(i,l)} := \begin{cases} a_{l',l}^{(i)} + e_{l'}^{(i)} a_{i,i} \pi_l^{(i)} & (i = j) \\ e_{l'}^{(j)} a_{j,i} \pi_l^{(i)} & (i \neq j) \end{cases}, \quad (6)$$

Given observed normalized CQTs up to the t th frame $\mathbf{y}_{0:t} = \{\mathbf{y}_\tau\}_{\tau=0}^t$, the score position at frame t is estimated with the standard HMM by solving

$$\arg \max_{z_t} P(z_t | \mathbf{y}_{0:t}) = \arg \max_{z_t} P(\mathbf{y}_{0:t}, z_t), \quad (7)$$

where

$$P(\mathbf{y}_{0:t}, z_t) = \sum_{z_{0:t-1}} \left(\prod_{\tau=1}^t \tilde{b}_{z_\tau}(\mathbf{y}_\tau) \tilde{a}_{z_{\tau-1}, z_\tau} \right) \tilde{b}_{z_0}(\mathbf{y}_0) \tilde{\pi}_{z_0}. \quad (8)$$

Here $z_{0:t-1}$ denotes $\{z_\tau\}_{\tau=0}^{t-1}$. Eq. (7) is derived from the Bayes' theorem.

This maximization problem can be solved efficiently with the forward algorithm. It computes the forward variable $\alpha_{t,z_t} := P(\mathbf{y}_{0:t}, z_t)$ in a recursive manner:

$$\alpha_{t,(i,l)} = \begin{cases} \tilde{b}_{(i,l)}(\mathbf{y}_t) \sum_{\substack{j=0, \dots, N-1 \\ i'=0, \dots, L-1}} \alpha_{t-1,(j,l')} \tilde{a}_{(j,l'),(i,l)} & (t \geq 1), \\ \tilde{b}_{(i,l)}(\mathbf{y}_0) \tilde{\pi}_{(i,l)} & (t = 0). \end{cases} \quad (9)$$

Since $\tilde{a}_{(j,l'),(i,l)} = 0$ unless $0 \leq i - j \leq 2$, the complexity of computing $\alpha_{t,(i,l)}$ is of $\mathcal{O}(LN)$ at each time step.

III. INCORPORATING ARBITRARY REPEATS/SKIPS AND FAST SCORE-FOLLOWING ALGORITHMS

A. Incorporating Arbitrary Repeats/Skips and Computational Complexity for Inference

So far, the top HMM is left-to-right and its states are connected only to their neighboring states. However, all top states must be connected to describe arbitrary repeats/skips, i.e. $a_{j,i} > 0$ for all j and i . The model is a generalization of the performance models in previous studies [5], [11], [15].

Assuming $L = 1$ for simplicity and dropping the subscripts l, l' from the parameters of the standard HMM and the forward variables as $\tilde{a}_{j,i} := \tilde{a}_{(j,0),(i,0)}$, $\tilde{b}_i(\mathbf{y}_t) := \tilde{b}_{(i,0)}(\mathbf{y}_t)$, $\tilde{\pi}_i := \tilde{\pi}_{(i,0)}$ and $\alpha_{t,i} := \alpha_{t,(i,0)}$, Eq. (9) can be rewritten as

$$\alpha_{t,i} = \begin{cases} \tilde{b}_i(\mathbf{y}_t) \sum_{j=0}^{N-1} \alpha_{t-1,j} \tilde{a}_{j,i} & (t \geq 1), \\ \tilde{b}_i(\mathbf{y}_0) \tilde{\pi}_i & (t = 0). \end{cases} \quad (10)$$

Eq. (10) for $t \geq 1$ contains a summation over N states for each i , and the complexity is of $\mathcal{O}(N^2)$. As we will experimentally show in Section IV-A, this complexity is too large to run in real time with scores of practical length on a modern laptop. Therefore, it is crucial to reduce the complexity. It is noteworthy that a similar large complexity can emerge even if only specific repeats/skips are allowed (e.g. transitions between the first notes

of bars in a score), since the number of such specific transitions often increases in proportion to N .

One may think that pruning techniques can be used to reduce the computational complexity. However, pruning is ineffective here since repeats/skips seldom occur, and it is necessary to take all transitions into account. Computing all transitions has a benefit also in following performances without repeats/skips. When an estimation error of score position occurs, a score follower may fail to track the performance and become lost. It often happens that a score follower with a pruning technique (e.g. with a limited search window) cannot recover from being lost. By contrast, if a score follower searches all transitions, it can return to find the correct score position after a while if the performer continues the performance.

B. Reduction of Computational Complexity by Factorizing Probabilities of Repeats/Skips

One method to reduce the computational complexity while computing all transitions is to introduce some constraints on the transition probabilities. In [13], reduction of the computational complexity is achieved with an assumption that the probability of score positions where performers stop before repeats/skips (stop positions) is the same regardless of where they resume performing after repeats/skips (resumption positions).

We shall introduce this assumption to the performance HMM. The transition probability of a repeat/skip from event j to event i is then written as a product of two probabilities s_j and r_i . s_j is the probability of stopping at event j before a repeat/skip, and r_i is the probability of resuming a performance at event i after a repeat/skip. The transition probability of the top HMM is then written as

$$a_{j,i} = a_{j,i}^{(\text{nbh})} + s_j r_i. \quad (11)$$

where $a_{j,i}^{(\text{nbh})}$ is a band matrix satisfying $a_{j,i}^{(\text{nbh})} = 0$ unless $0 \leq i - j \leq 2$. The parameter $a_{j,i}^{(\text{nbh})}$ characterizes transitions within neighboring states and is determined according to the normalization constraint of $a_{j,i}$, which is written as $1 = \sum_i a_{j,i} = \sum_i a_{j,i}^{(\text{nbh})} + s_j \sum_i r_i$ for all j . Without loss of generality, we can assume $\sum_i r_i = 1$ and then we have $\sum_i a_{j,i}^{(\text{nbh})} = 1 - s_j$.

Let us denote the set of neighboring states of top state i by $\text{nbh}(i) := \{j; j = 0, \dots, N-1, 0 \leq i - j \leq 2\}$. The transition probability of the standard HMM $\tilde{a}_{j,i}$ for $j \notin \text{nbh}(i)$ is written as

$$\tilde{a}_{j,i} = e_0^{(j)} s_j r_i \pi_0^{(i)}. \quad (12)$$

With Eqs. (12) and (10), we have

$$\alpha_{t,i} = \tilde{b}_i(\mathbf{y}_t) \left\{ \sum_{j \in \text{nbh}(i)} \alpha_{t-1,j} \tilde{a}_{j,i} + r_i \pi_0^{(i)} \left(\sum_{j=0}^{N-1} \alpha_{t-1,j} e_0^{(j)} s_j - \sum_{j \in \text{nbh}(i)} \alpha_{t-1,j} e_0^{(j)} s_j \right) \right\}. \quad (13)$$

Since the first summation in the parentheses of the second term is independent of i , it is sufficient to calculate it once at each

time step. This term and the rest of Eq. (13) are of $\mathcal{O}(N)$, and hence the total computational complexity is $\mathcal{O}(N)$. The space complexity is also reduced: The transition probability matrix in the top level is now parameterized by $4(N - 1)$ parameters (s_j , r_i and $a_{i,j}^{(\text{nbh})}$). It has $N(N - 1)$ parameters originally.

This result can be generalized for the performance HMM with $L > 1$. The standard HMM has LN states and updating $\alpha_{t,(i,l)}$ at each time step is of $\mathcal{O}((LN)^2)$ according to Eq. (9). If we introduce the above assumption, the transition probability of the standard HMM $\tilde{a}_{(j,l'),(i,l)}$ can also be divided into a component dependent only on i, l and a component dependent only on j, l' . Therefore, the total computational complexity is reduced to $\mathcal{O}(LN)$ (see Appendix B for details). Importantly, this reduction method can be used regardless of the topology of the bottom HMMs, and it is compatible with the pause states and applicable to performance HMMs with more complex structure of bottom HMMs (e.g. [6], [20], [24], [25]).

A similar reduction method is valid for the Viterbi algorithm and the backward algorithm. The method can be applied to any HMM and similar dynamic programming techniques as well, and it can be useful for applications other than score following, (e.g. timbre editing of music signals [26]).

C. Explicit Description of Silent Breaks at Repeats/Skips

We can achieve a similar reduction of the computational complexity by using another assumption on arbitrary repeats/skips. Performers frequently make silent breaks at repeats/skips to get ready for resuming the performance. In fact, 59 of 63 repeats/skips accompanied the breaks longer than 500 ms in actual performances used in Section IV-C1

Let us represent the silent breaks by introducing an additional state (the break state) as the N th top state. The duration of the breaks is described with the self-transition probability of the bottom state of the break state $a_{0,0}^{(N)}$, and its value is determined similarly to Eq. (5). Repeats/skips are represented as two-step transitions via the break state (Fig. 3). Stopping (resuming) a performance is expressed as transitions to (from) the break state whose probability is denoted by s_j (r_i , respectively). We note that the top states excluding the break state are connected only to neighboring top states, and thus $\tilde{a}_{j,i} = 0$ if $j \notin \text{nbh}(i)$ for all $i, j \neq N$. On the other hand, the break state is connected to all top states except itself. We put $\tilde{a}_{N,N} = 0$. The transition probability of the standard HMM from or to the break state is written as

$$\tilde{a}_{j,N} = e_0^{(j)} s_j \pi_0^{(N)} \quad (j \neq N), \quad (14)$$

$$\tilde{a}_{N,i} = \begin{cases} e_0^{(N)} r_i \pi_0^{(i)} & (i \neq N), \\ a_{0,0}^{(N)} & (i = N) \end{cases} \quad (15)$$

where $e_0^{(N)} (= 1 - a_{0,0}^{(N)})$ and $\pi_0^{(N)} (= 1)$ denote the exiting probability and the initial probability of state $(N, 0)$.

For this model, Eq. (10) for $t \geq 1$ can be written as

$$\alpha_{t,i} = \begin{cases} \tilde{b}_i(\mathbf{y}_t) \left(\sum_{j \in \text{nbh}(i)} \alpha_{t-1,j} \tilde{a}_{j,i} + \alpha_{t-1,N} \tilde{a}_{N,i} \right) & (i \neq N) \\ \tilde{b}_N(\mathbf{y}_t) \sum_{j=0}^{N-1} \alpha_{t-1,j} \tilde{a}_{j,N} & (i = N). \end{cases} \quad (16)$$

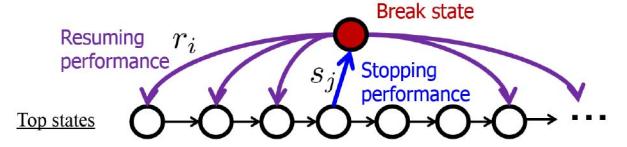


Fig. 3. A repeat/skip can be described with two-step transitions via the break state representing silent breaks.

We see that updating $\alpha_{t,i}$ involves summation of at most four terms for each $i \neq N$ and N terms for $i = N$. The total complexity is thus $\mathcal{O}(N)$ for each time step. This reduction method can also be extended to the case of $L > 1$ (see Appendix C).

It is noteworthy that the performance HMM with the break state is related to the performance HMM presented in Section III-B. If we assume that transitions go through the break state in no time, the two-step transition from top state j to top state i via the break state is reduced to the direct transition from top state j to top state i , and its probability is written as a product of s_j and r_i . In other words, the difference between these models is whether breaks are explicitly described. Since it is difficult to quantify its effect on the performance of score following analytically, we will evaluate the effect through an experiment in Section IV-C2.

IV. EXPERIMENTAL EVALUATION OF THE PROPOSED SCORE-FOLLOWING ALGORITHMS

A. Processing Time

We measured processing times in order to evaluate the reduction of the computational complexity with the proposed algorithms. The processing time depends on the number of events N and virtually not on other score content and signal content. We used synthetic scores with 10 to 10^6 events¹ and a random signal of two seconds length with a sampling rate of 16 kHz as an audio input. Normalized CQTs were computed with a frame length of 128 ms and a hopsize of 20 ms. Their center frequencies ranged from 55 to 7040 Hz at a semitone interval, and the quality factor was set to 16, which approximately corresponds to one semitone. Algorithms were implemented in C++ on a computer with 3.30 GHz CPU (Intel(R) Core(TM) i3-2120 CPU) and 8 GB memory running Debian.

Processing times averaged over 100 frames with standard errors are shown in Fig. 4 for the algorithms proposed in Section III-C (*break algorithm*) with and without the pause states ($L = 2$ and $L = 1$) and the algorithm that calculates $\alpha_{t,i}$ according to Eq. (10) (baseline algorithm). (The results for the algorithm proposed in Section III-B (*no-break algorithm*) did not significantly differ with the results for the *break algorithms*.) It can be confirmed that the average processing times increased asymptotically in proportion to N^2 (N) with the baseline algorithm (the *break algorithms*, respectively). The result shows that the proposed algorithms significantly suppress the increase of processing times. The processing times for $N \geq 1000$ were larger than the hopsize with the baseline algorithm, and the algorithm can work in real time

¹Practical scores contain $\mathcal{O}(10^3)$ to $\mathcal{O}(10^4)$ notes. For instance, there are around 2200 events in the clarinet part of the first movement in the Mozart's Clarinet Quintet.

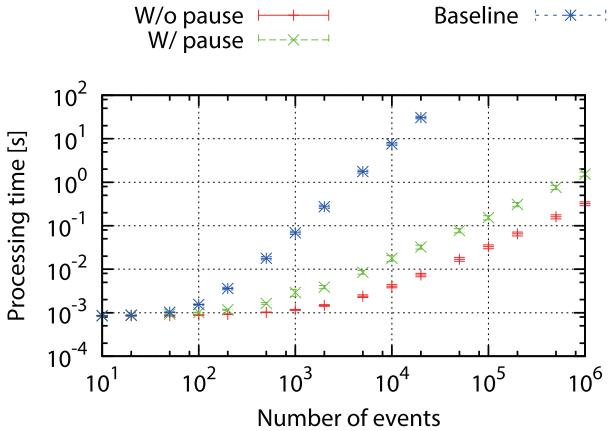


Fig. 4. Average processing times with standard errors with respect to the number of events. “W/ pause” and “W/o pause” represent the *break algorithm* with and without the pause states, respectively. “Baseline” represents a simple extension of the algorithms proposed in previous studies [5], [11], [15].

with scores with only up to $\mathcal{O}(10^2)$ events, which is the size of short music pieces. By contrast, the average processing times were below the hopsize for $N \leq 10000$ ($N \leq 50000$) with the *break algorithm* with (without, respectively) the pause states. Therefore, the proposed algorithms with and without the pause states can work in real time with scores with up to $\mathcal{O}(10^3)$ events and $\mathcal{O}(10^4)$ events, respectively. Note that processing times depend on the computing power, but their relative values remain almost the same and the proposed algorithms are always effective in reducing the computational complexity.

B. Score-Following Accuracy for Performances with Errors

1) Data Preparation: To evaluate the score-following accuracy for performances with errors, we conducted an experiment using the Bach10 dataset [27]. It consists of audio recordings of ten four-part chorales by J. S. Bach. The soprano, alto, tenor and bass parts of each piece were separately recorded and performed by the violin, clarinet, saxophone and bassoon, respectively. Their durations ranged from 25 to 41 seconds.

Since the performances did not contain errors, we simulated errors by randomly inserting, dropping and substituting notes in each score, which correspond to deletion, insertion and substitution errors in the performance, respectively. Their probability values were obtained from the MIDI piano performances during practice in [13]: 0.0034 for deletion errors and 0.0245 for insertion errors. For simplicity, substitution errors were restricted to three types typical in clarinet performances, namely errors in semitone, whole-tone and perfect 12th. The first two errors are often caused by fingering errors and mis-readings of the score, and the last error is caused by overblowing on a clarinet. The probability values of the three pitch errors were 0.0145, 0.0224 and 0.0047 in the simulation, where the probability of the perfect 12th pitch error was substituted by that of the octave pitch errors obtained in [13].

2) Experimental Conditions: We conducted a preliminary experiment and set the parameter for performance errors as follows: $a_{i,i+2} = 1.0 \times 10^{-50}$ for deletion errors, $a_{i,i} = 0$ for insertion errors, and $a_{1,1}^{(i)} = 0.999$ and $a_{0,1}^{(i)} = 1.0 \times 10^{-100}$ for pauses between notes. Although the mixture weight $w_{k,0}^{(i)}$ can be learned from audio signals at each k and i in principle, it

is difficult to obtain them independently for the lack of enormous performance data. To reduce the number of parameters, we considered only the most important three substitution errors described in the previous section. The mixture weights $w_{k,0}^{(i)}$ for the errors were designed in proportion to their frequencies used in the simulation:

$$w_{k,0}^{(i)} = \begin{cases} 1 - C & (k = p_i) \\ C \times 0.175 & (k = p_i \pm 1) \\ C \times 0.270 & (k = p_i \pm 2) \\ C \times 0.055 & (k = p_i \pm 19) \\ 0 & (\text{otherwise}) \end{cases} \quad (17)$$

for all $p_i \neq -1$, where C is the probability of pitch errors. The value of C was optimized in a preliminary experiment and we set $C = 1.0 \times 10^{-50}$. For $p_i = -1$, we put $w_{k,0}^{(i)} = 0$ unless $k = -1$. The probabilities of stopping and resuming a performance s_j, r_i were set uniformly in i, j : $s_0 = s_1 = \dots = s_{N-1} = 1.0 \times 10^{-x}$ for some positive x and $r_0 = r_1 = \dots = r_{N-1} = 1/N$. Since the value of $a_{0,0}^{(N)}$ did not significantly change the result in a preliminary experiment, we fixed $a_{0,0}^{(N)} = 0.996$.

The accuracy of score following generally depends on the parameters of the emission probabilities. It has been reported that learning them from audio performances improves the accuracy [10], [22], and thus we learned the parameters μ_k and Σ_k from audio signals. The parameters can be learned from every musical instrument if necessary data is available and we can form a detailed model for a specific instrument. Alternatively, we can use a set of data consisting of several musical instruments to form a “general model” that can be applied for a wider class of instruments. Such a learning method is applicable for any instruments in principle, and it can be even more effective for musical instruments with complex signals, for which physical modeling or manual spectrum-template construction is more difficult. In general, there is a tradeoff between the generalization capability and the adaptation ability. Here, we learned the parameters with performance data of several musical instruments and used them to measure the accuracy of score following.

The learning data consisted of performances played by the violin and clarinet in RWC musical instrument database [28]. To reduce overfitting, we assumed that Σ_k is diagonal and introduced a lower bound, or a flooring value F , on the diagonal elements of Σ_k . The introduction of F is called the flooring method and generally used for speech recognition (e.g. see [29]). We conducted a preliminary experiment and found the optimal $F = 1.0 \times 10^{-4}$. The initial probabilities were set as $\pi_i = 0$ for $i \neq 0$ and $\pi_0 = 1$.

We compared the proposed algorithms with *Antescofo* [6], which is one of the most known score-following systems applied to various musical pieces and used in the most severe artistic situations. Antescofo was not developed to cope with repeats/skips in monophonic performances, and is without special treatments for repeats/skips. It had the best accuracy in the music information retrieval evaluation exchange (MIREX 2006) [30], which is the most famous evaluation contest in this field. Since Antescofo ended score following when the last note in the score was estimated, estimated score positions were assumed to be the last note from the time when Antescofo

TABLE I

AVERAGE PIECEWISE PRECISION RATES AND STANDARD ERRORS FOR VIOLIN, CLARINET, SAXOPHONE AND BASSOON PERFORMANCES WITH ERRORS. “PROPOSED ($s_j = 0$)” (“ANTESCOFO”) DENOTES THE *BREAK ALGORITHM* WITH $s_j = 0$ (ANTESCOFO [6], RESPECTIVELY)

Musical instrument	Proposed ($s_j = 0$)	Antescofo
Violin	0.72 ± 0.03	0.66 ± 0.06
Clarinet	0.61 ± 0.05	0.57 ± 0.08
Saxophone	0.63 ± 0.06	0.64 ± 0.06
Bassoon	0.76 ± 0.04	0.79 ± 0.04

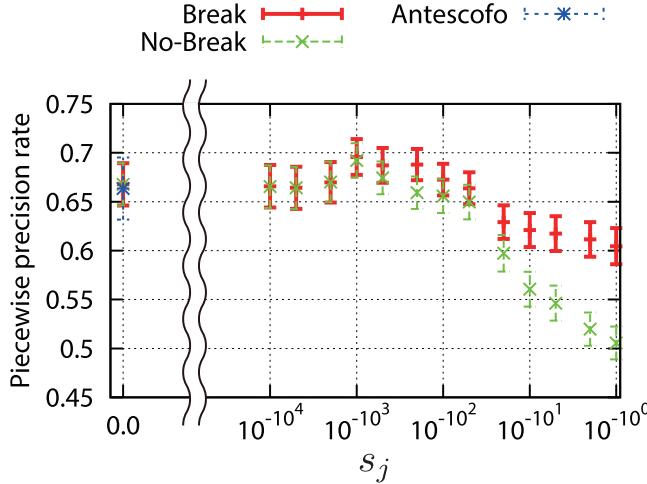


Fig. 5. Average piecewise precision rates and standard errors with respect to s_j for audio performances obtained by simulating errors. The *break algorithm* (“Break”) and the *no-break algorithm* (“No-Break”) without the pause states are compared to Antescofo [6].

ended score following. The overall accuracy of score following was measured by piecewise precision rate (PPR), defined as the piecewise average rate of onsets correctly detected within Δ ms error. The PPR has been used with $\Delta = 2000$ ms in MIREX [30], [31].

3) *Results*: Table I summarizes average PPRs and standard errors with $\Delta = 300$ ms for every musical instrument. The results for the *no-break algorithm* did not significantly differ with the results for the *break algorithm* when $s_j = 0.0$. We found that the proposed algorithms provided similar accuracies for the saxophone and bassoon data, which were not contained in the learning data, compared to the clarinet and violin data. The PPRs obtained with the proposed algorithms were similar to those obtained with Antescofo in all data.

Fig. 5 illustrates average PPRs and standard errors with $\Delta = 300$ ms. As described in Section III-A, computing all transitions help that the score follower returns to recover from being lost. The benefit can be confirmed from that the proposed algorithms with $s_j = 1.0 \times 10^{-1000}$ provided around 0.05 higher accuracy than Antescofo, which searches only local transitions. On the other hand, s_j s larger than 1.0×10^{-500} caused the frequent overdetection of repeats/skips and the accuracy became lower than $s_j = 0$. A similar tendency was observed in PPR with $\Delta = 500$ and 2000 ms.

Large values of s_j deteriorated the score-following accuracy of the present algorithms as shown in Fig. 5. This is because the larger s_j , the more frequently the algorithms may misestimate insertion/deletion/substitution errors as repeats/skips. We indeed confirmed that the number of misdected repeats/skips increased with larger s_j .

TABLE II

THE NUMBER OF ERRORS AND REPEATS/SKIPS IN THE USED CLARINET PERFORMANCES

	Pauses between notes	Deletion error	Insertion error	Substitution error	Repeats/skips
Count	21	1	21	33	63

TABLE III

STATISTICS OF DIFFERENCES IN SCORE TIMES BEFORE AND AFTER REPEATS/SKIPS IN THE PERFORMANCE DATA.
“QU.” IS AN ABBREVIATION FOR QUARTILE

Score time	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
In second	-84.83	-15.5	-7.75	-8.775	-1.875	45.750
In event	-331	-44	-25	-23.35	-4	178

There was around 0.1 difference in PPR between the algorithms when s_j is large. We found that the total number of misdected repeats/skips by the *no-break algorithm* was around 1.2 times larger than that of the *break algorithm* for $s_j \geq 10^{-10}$. Since the *break algorithm* assumes that repeats/skips always accompany breaks and simulated errors did not accompany pauses, the results suggest that the explicit description of the breaks reduced misestimations of the errors as repeats/skips.

C. Score-Following Accuracy for Performances with Errors and Repeats/Skips

1) *Performance Data During Practice*: We collected 16 audio recordings of clarinet performances with a time range of 31 to 213 s (totally 28 min 48 s). We requested an amateur clarinetist to freely practice seven music pieces containing classical and popular music pieces and nursery rhymes, partially from RWC music database [28]. His performances were recorded with a vibration microphone attached to the clarinet.

The performances were aligned to the notes in the scores by one of the authors. The total number of performed notes was 2672, and Table II lists the count of errors and repeats/skips. Table III summarizes differences in score times before and after repeats/skips in the performance data, and we see that they contain repeats/skips between remote score positions. Here, only breaks and pauses between notes longer than 500 ms were counted since it is difficult to accurately annotate offsets of performed notes and short silent breaks and pauses between notes. All transitions with $j \notin \text{nbh}(i)$ were counted as repeats/skips, where i and j denote stop and resumption positions.

2) *Results*: The parameters were same as in Section IV-B2. To measure how well the algorithms followed repeats/skips, we calculated a detection rate of repeats/skips and the time interval between a repeat/skip and its detection, which we call following time. A repeat/skip was defined to be detected if there was a correctly estimated frame until the next repeat/skip or the end of the audio recording.

Fig. 6 illustrates average PPRs with standard errors for $\Delta = 300$ ms. Both proposed algorithms outperformed Antescofo at all s_i s, clearly showing that the proposed algorithms are effective in following performances with errors and repeats/skips. A similar tendency was observed in PPR with $\Delta = 100, 500$ and 2000 ms. We also measured the effect of adding the pause states

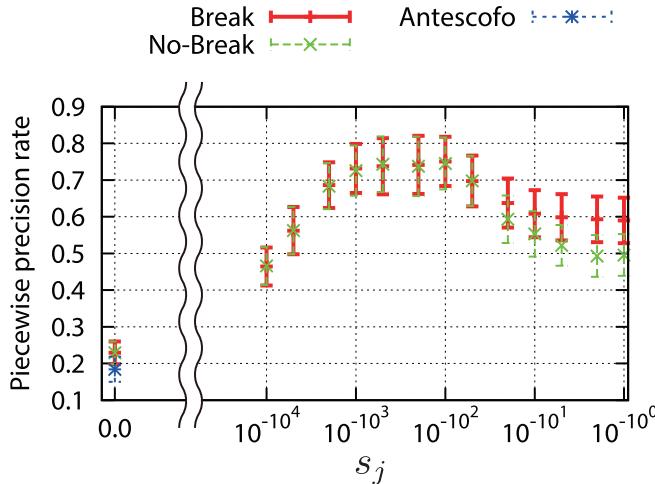


Fig. 6. Average piecewise precision rates with standard errors with respect to s_j . The algorithms are same as in Fig. 5.

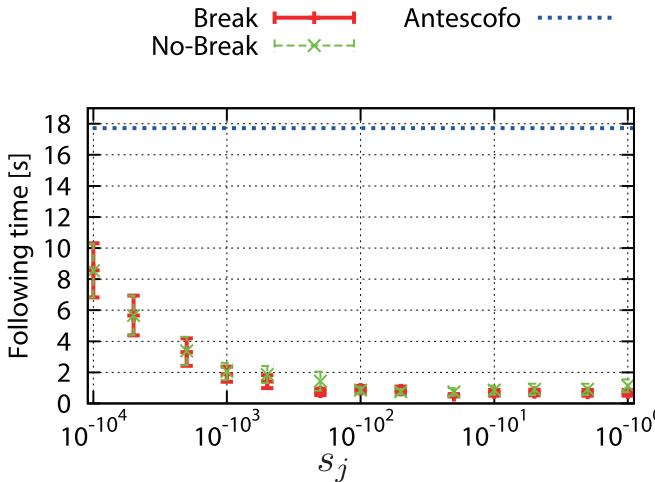


Fig. 7. Average following time and standard error for varying s_j . The algorithms are same as in Fig. 5. For Antescofo, only the average following time is shown.

in the proposed algorithms with $s_i = 1.0 \times 10^{-100}$, and found that it increased PPRs by 0.05 on average.

Table IV summarizes the detection rates of repeats/skips, and Fig. 7 illustrates averages of following times over all detected repeats/skips (average following times) and standard errors in second. Since the standard error for Antescofo was too large to display in the figure, only the average value is shown. Both proposed algorithms clearly outperformed Antescofo in the detection rate and the following time. For example, compared to Antescofo, both proposed algorithms with $s_j = 1.0 \times 10^{-100}$ detected 14 times more repeats/skips and caught up with them 20 times faster in second. These results show that the proposed models are effective for repeats/skips.

The *break algorithm* (the *no-break algorithm*) with $s_j = 1.0 \times 10^{-100}$ detected 56 (57) repeats/skips, but failed to detect seven (six, respectively) repeats/skips. These failures were caused by the existence of sections and phrases similar to each other in the scores (e.g. choruses in popular music) and considerably short performances between repeats/skips. For example, nine performances between repeats/skips were below five seconds.

TABLE IV
DETECTION RATES OF REPEATS/SKIPS FOR VARYING s_j .
THE ALGORITHMS ARE SAME AS IN FIG. 5

s_j	Break	No-Break	Antescofo
1.0×10^{-1}	58/63	60/63	-
1.0×10^{-5}	59/63	59/63	-
1.0×10^{-10}	59/63	60/63	-
1.0×10^{-50}	58/63	59/63	-
1.0×10^{-100}	56/63	57/63	-
1.0×10^{-500}	55/63	55/63	-
1.0×10^{-1000}	56/63	55/63	-
1.0×10^{-5000}	43/63	43/63	-
0.0	13/63	13/63	4/63

Most of the repeats/skips accompanied silent breaks, but the *break algorithm* provided similar results to the *no-break algorithm*. This is because the top states associated with rests can play the same role of the break state since these top states were connected to all top states.

Furthermore, we measured following times and detection rates for performances played by other musical instruments. The audio recordings in the Bach10 dataset did not contain repeats/skips, and we synthesized performances containing repeats/skips by randomly jumped between breaks in each recording with a probability of 0.1 and inserting silent breaks at repeats/skips. The durations of the breaks were sampled uniformly from 0.5 to 30 seconds and each synthesized performance was forced to contain at least one repeat/skip. After the synthesis, errors were simulated in the same way as in Section IV-B1. Table V summarizes detection rates of repeats/skips for every musical instrument. The proposed algorithms with $s_j = 1.0 \times 10^{-1000}$ outperformed Antescofo in the detection rate, and we found similar tendency in the PPR and the following time as shown in Fig. 8(a) and (b), respectively. These results show that the proposed algorithms are also effective in following performances with errors and repeats/skips for various musical instruments.

A demonstration video of an automatic accompaniment system using the *break algorithm* without the pause states is available at <https://www.youtube.com/watch?v=fW6VKiC4k34> on Youtube [32]. In the video, the *break algorithm* successfully follows the performances during practice and catches up the performances after repeats/skips within a few seconds.

V. DISCUSSIONS

A. Improvement of the Proposed algorithms

We now discuss possible extensions of the proposed algorithms. The stop and resumption positions are not completely random, and their distributions have certain tendencies in actual performances [13]. For example, performers frequently resume from the first beats of bars and the beginning of phrases, which reflects performers' understanding of musical structures. These tendencies can be incorporated in s_j, r_i in our performance HMMs, and the accuracy and following times of the proposed algorithms would improve [13].

Another method to improve the proposed algorithms is to refine the model of the durations of performed events. For this

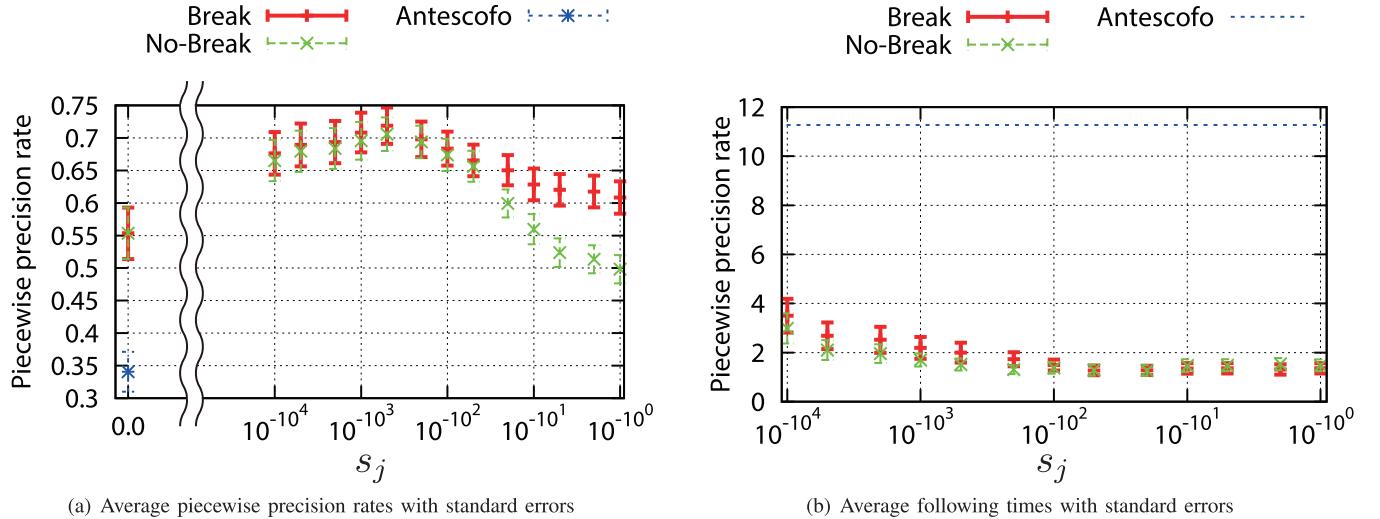


Fig. 8. (a) Average piecewise precision rates and (b) average following times with respect to s_j for audio performances with simulated errors and repeats/skips. The algorithms are same as in Fig. 5, and only the average following time is shown for Antescofo in the right panel. (a) Average piecewise precision rates with standard errors (b) Average following times with standard errors.

TABLE V
DETECTION RATES OF REPEATS/SKIPS FOR VIOLIN, CLARINET, SAXOPHONE AND BASSOON DATA WITH SIMULATED ERRORS AND REPEATS/SKIPS. THE ALGORITHMS ARE SAME AS IN FIG. 5, AND $s_j = 1.0 \times 10^{-1000}$
WAS USED IN BOTH PROPOSED ALGORITHMS

Musical instrument	Break	No-Break	Antescofo
Violin	13/13	13/13	2/13
Clarinet	11/11	10/11	4/11
Saxophone	11/12	11/12	2/12
Bassoon	10/10	10/10	0/10

purpose, we can assign multiple bottom states to model the duration [20], [24], [25] or explicitly introduce its probability distribution [6]. This refinement is compatible with the proposed methods to reduce the computational cost since they can be used regardless of the topology of the bottom HMMs.

The proposed algorithms successfully followed clarinet performances against tempo changes in the experiment and the demonstration video in Section IV-C. However, the accuracy may deteriorate for the performances with large tempo changes. To suppress the deterioration, it would be effective to adequately change d_i on the fly, referring to estimated tempos.

B. Extension to Polyphonic Music

Although we have confined ourselves to monophonic performances, let us briefly discuss the polyphonic case. We can construct a performance HMM for polyphonic scores similarly to the monophonic case. By associating top states with musical events (chords, notes and rests) in a polyphonic score, the top HMM can be used without any change, and insertions and deletions of chords, pauses between chords and repeats/skips can be incorporated in the same way. Importantly, the present methods to reduce the computational complexity can be applied to the polyphonic case since it is independent of details of the bottom HMMs. On the other hand, we need to extend the bottom HMMs to include chords. Especially, errors may occur at every note in a chord, and there are a combinatorially large number of possible forms of errors for a large chord. Although we could prepare spectral templates for all possible forms of played chords and use a mixture distribution similarly to Eq. (4) in principle, it

TABLE VI
IMPORTANT PARAMETERS AND THEIR MEANINGS OF THE PROPOSED MODELS

Mathematical notation	Meaning
$a_{i,j}$	Transition probability of the top HMM
π_i	Initial probability of the top HMM
$a_{l,l'}^{(i)}$	Transition probability of the i -th bottom HMM
$\pi_{(i)}^{(i)}$	Initial probability of the i -th bottom HMM
$e_l^{(i)}$	Exiting probability of the i -th bottom HMM
$b_l^{(i)}(\mathbf{y}_t)$	Emission probability of state (i, l) for observation \mathbf{y}_t
$\tilde{a}_{(i,l),(j,l')}$	Transition probability of the standard HMM obtained by flattening the two-level HMM
$\tilde{\pi}_{(i,l)}$	Initial probability of the standard HMM obtained by flattening the two-level HMM
$\tilde{b}_{(i,l)}(\mathbf{y}_t)$	Initial probability of the states of the standard HMM obtained by flattening the two-level HMM
s_j	The probability of stopping at event j before a repeat/skip
r_i	The probability of resuming a performance at event i after a repeat/skip

requires large computational cost in estimating score positions. However, the influence of note-wise errors in spectral differences is generally less significant for a large chord, and a bold approximation of neglecting note-wise errors would work relatively well for such a case, which can serve as a practical method to avoid the large computational cost.

There are other issues for polyphonic performances. For example, notes in a chord are indicated to be performed simultaneously in the score, but they can be actually performed at different times. Also, relative energy of notes in a chord depends on the performer. Their treatment requires additional discussions and experiments, and the extension to polyphonic performances is now under investigation.

VI. CONCLUSION

We discussed score following of monophonic music performances with errors and arbitrary repeats/skips by constructing a stochastic model of music performance. We incorporated possible errors in audio performances into the model. In order to solve the problem of large computational cost for following arbitrary repeats/skips, we presented two HMMs

that describe a probability of repeats/skips with a probability of stop positions and a probability of resumption positions, and derived computationally efficient algorithms. We demonstrated real-time working of the algorithms with scores of practical length ($\mathcal{O}(10^3)$ to $\mathcal{O}(10^4)$ events). Experimental evaluations using clarinet performance data showed that the algorithms outperformed Antescofo in the accuracy of score following and the tracking ability of repeats/skips. In addition, we briefly discussed methods to improve the proposed algorithms and extend them for polyphonic inputs.

APPENDIX

A. List of Important Parameters

Important parameters of the proposed models are listed in Table VI.

B. Derivation of the No-Break Algorithm for $L > 1$

We now derive an efficient algorithm of computing $\alpha_{t,(i,l)}$ for the performance HMM without the break state in the case of $L > 1$. Assuming that the transition probability of repeats/skips is described as a product of s_j and r_i , the transition probability of the standard HMM $\tilde{a}_{(j,l'),(i,l)}$ for $j \notin \text{nbh}(i)$ can be written as

$$\tilde{a}_{(j,l'),(i,l)} = e_{l'}^{(j)} s_j r_i \pi_l^{(i)}, \quad (18)$$

and Eq. (9) for $t \geq 1$ is rewritten as

$$\begin{aligned} \alpha_{t,(i,l)} = & \tilde{b}_{(i,l)}(\mathbf{y}_t) \left(\sum_{\substack{j \in \text{nbh}(i) \\ l'=0, \dots, L-1}} \alpha_{t-1,(i,l')} \tilde{a}_{(j,l'),(i,l)} \right. \\ & \left. + r_i \pi_l^{(i)} \sum_{\substack{j \notin \text{nbh}(i) \\ l'=0, \dots, L-1}} \alpha_{t-1,(j,l')} e_{l'}^{(j)} s_j \right). \end{aligned} \quad (19)$$

The first summation in the parentheses of Eq. (19) is of $\mathcal{O}(L)$. The second summation can be converted into

$$\begin{aligned} & \sum_{\substack{j \notin \text{nbh}(i) \\ l'=0, \dots, L-1}} \alpha_{t-1,(j,l')} e_{l'}^{(j)} s_j \\ = & \sum_{\substack{j=0, \dots, N-1 \\ l'=0, \dots, L-1}} \alpha_{t-1,(j,l')} e_{l'}^{(j)} s_j - \sum_{\substack{j \in \text{nbh}(i) \\ l'=0, \dots, L-1}} \alpha_{t-1,(j,l')} e_{l'}^{(j)} s_j. \end{aligned} \quad (20)$$

The first summation of the right-hand side of Eq. (20) is independent of i and thus it is sufficient to compute it once at each time step. Hence, the total computational complexity at each time step is of $\mathcal{O}(LN)$.

C. Derivation of the Break Algorithm for $L > 1$

Let us consider the performance HMM with the break state and with L bottom states in each top state. In the same way as Section III-C, silent breaks at repeats/skips can be introduced as top state N (the break state) and arbitrary repeats/skips are described with two-step transitions via the break state. Since the transition probability of the standard HMM $\tilde{a}_{(j,l'),(i,l)}$ is zero

unless $j \in \text{nbh}(i) \cup \{N\}$, Eq. (9) for $t \geq 1$ and $i \neq N$ can be rewritten as

$$\begin{aligned} \alpha_{t,(i,l)} = & \tilde{b}_{(i,l)}(\mathbf{y}_t) \left(\sum_{\substack{j \in \text{nbh}(i) \\ l'=0, \dots, L-1}} \alpha_{t-1,(j,l')} \tilde{a}_{(j,l'),(i,l)} \right. \\ & \left. + \sum_{l'=0}^{L-1} \alpha_{t-1,(N,l')} \tilde{a}_{(N,l'),(i,l)} \right), \end{aligned} \quad (21)$$

The second term in the parentheses of Eq. (21) for each $i \neq N$ is of a constant computational complexity. On the other hand, Eq. (9) for $t \geq 1$ and $i = N$ is converted into

$$\alpha_{t,(N,l)} = \tilde{b}_{(N,l)}(\mathbf{y}_t) \sum_{\substack{j=0, \dots, N-1 \\ l'=0, \dots, L-1}} \alpha_{t-1,(N,l')} \tilde{a}_{(j,l'),(N,l)}. \quad (22)$$

This computation is of $\mathcal{O}(LN)$ and hence the total computational complexity is of $\mathcal{O}(LN)$ at each time step.

ACKNOWLEDGMENT

We thank Yuu Mizuno and Kosuke Suzuki for participating in the early stage of this work, Naoya Ito for playing the clarinet, and Hirokazu Kameoka for useful discussions.

REFERENCES

- [1] R. B. Dannenberg, “An on-line algorithm for real-time accompaniment,” in *Proc. Int. Comput. Music Conf.*, 1984, pp. 193–198.
- [2] B. Vercoe, “The synthetic performer in the context of live performance,” in *Proc. Int. Comput. Music Conf.*, 1984, pp. 199–200.
- [3] A. Arzt, G. Widmer, and S. Dixon, “Automatic page turning for musicians via real-time machine listening,” in *Proc. Eur. Conf. Artif. Intell.*, 2008, pp. 241–245.
- [4] N. Orio, S. Lemouton, D. Schwarz, and N. Schnell, “Score following: State of the art and new developments,” in *Proc. New Interfaces for Musical Express.*, 2003, pp. 36–41.
- [5] B. Pardo and W. Birmingham, “Modeling form for on-line following of musical performances,” in *Proc. AAAI*, 2005, vol. 2, pp. 1018–1023.
- [6] A. Cont, “A coupled duration-focused architecture for real-time music-to-score alignment,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, pp. 974–987, Jun. 2010.
- [7] C. Joder, S. Essid, and G. Richard, “A comparative study of tonal acoustic features for a symbolic level music-to-score alignment,” in *Proc. IEEE Workshop Applicat. Signal Process. Audio Acoust.*, 2010, pp. 409–412.
- [8] Z. Duan and B. Pardo, “A state space model for online polyphonic audio-score alignment,” in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 197–200.
- [9] T. Otsuka, K. Nakadai, T. Takahashi, T. Ogata, and H. G. Okuno, “Real-time audio-to-score alignment using particle filter for coplayer music robots,” *EURASIP J. Appl. Signal Process.*, vol. 2011, no. 384651, pp. 1–13, 2011.
- [10] C. Joder, S. Essid, and G. Richard, “A conditional random field framework for robust and scalable audio-to-score matching,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2385–2397, Nov. 2011.
- [11] N. Montecchio and A. Cont, “A unified approach to real time audio-to-score and audio-to-audio alignment using sequential Montecarlo inference techniques,” in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 193–196.
- [12] T. Nakamura, E. Nakamura, and S. Sagayama, “Acoustic score following to musical performance with errors and arbitrary repeats and skips for automatic accompaniment,” in *Proc. Sound Music Comput. Conf.*, Aug. 2013, pp. 299–304.
- [13] E. Nakamura, T. Nakamura, Y. Saito, N. Ono, and S. Sagayama, “Outer-product hidden Markov model and polyphonic MIDI score following,” *J. New Music Res.*, vol. 43, no. 2, pp. 183–201, 2014.

- [14] D. Schwarz, N. Orio, and N. Schnell, “Robust polyphonic MIDI score following with hidden Markov models,” in *Proc. Int. Comput. Music Conf.*, 2004.
- [15] C. Oshima, K. Nishimoto, and M. Suzuki, “A piano duo performance support system to motivate children’s practice at home,” *Trans. Info. Process. Soc. Jpn.*, vol. 46, no. 1, pp. 157–170, 2005, in Japanese.
- [16] E. Nakamura, Y. Saito, N. Ono, and S. Sagayama, “Merged-output hidden Markov model for score following of MIDI performance with ornaments, desynchronized voices, repeats and skips,” in *Proc. Joint Conf. 40th Int. Comput. Music Conf. and 11th Sound and Music Comput. Conf.*, 2014, pp. 1185–1192.
- [17] E. Nakamura, N. Ono, S. Sagayama, and K. Watanabe, “A stochastic temporal model of polyphonic MIDI performance with ornaments,” *J. New Music Res.*, vol. 44, no. 4, pp. 287–304, 2015.
- [18] C. Fremerey, M. Müller, and M. Clausen, “Handling repeats and jumps in score-performance synchronization,” in *Proc. Int. Symp. Music Info. Retrieval*, 2010, pp. 243–248.
- [19] Z. Duan and B. Pardo, “Aligning semi-improvised music audio with its lead sheet,” in *Proc. Int. Symp. Music Inf. Retrieval*, 2011, pp. 513–518.
- [20] N. Orio and F. Déchelle, “Score following using spectral analysis and hidden Markov models,” in *Proc. Int. Comput. Music Conf.*, 2001, vol. 1001, pp. 1708–1710.
- [21] A. Cont, “Realtime audio to score alignment for polyphonic music instruments, using sparse non-negative constraints and hierarchical HMMs,” in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2006, vol. 5, pp. 245–248.
- [22] C. Joder, S. Essid, and G. Richard, “Learning optimal features for polyphonic audio-to-score alignment,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 2118–2128, Oct. 2013.
- [23] J. Brown and M. Puckette, “An efficient algorithm for the calculation of a constant Q transform,” *J. Acoust. Soc. Amer.*, vol. 92, pp. 2698–2701, 1992.
- [24] P. Cano, A. Loscos, and J. Bonada, “Score-performance matching using HMMs,” in *Proc. Int. Comput. Music Conf.*, 1999, pp. 441–444.
- [25] C. Raphael, “Automatic segmentation of acoustic musical signals using hidden Markov models,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 360–370, Apr. 1999.
- [26] T. Nakamura, H. Kameoka, K. Yoshii, and M. Goto, “Timbre replacement of harmonic and drum components for music audio signals,” in *Proc. Int. Conf. Acoust. Speech Signal Process.*, 2014, pp. 7520–7524.
- [27] Z. Duan and B. Pardo, “Soundprism: An online system for score-informed source separation of music audio,” *IEEE J. Sel. Topics. Signal Process.*, vol. 5, no. 6, pp. 1205–1215, 2011.
- [28] M. Goto, “Development of the RWC music database,” in *Proc. Int. Congr. Acoust.*, 2004, vol. 1, pp. 553–556.
- [29] HTK Speech Recognition Toolkit [Online]. Available: <http://htk.eng.cam.ac.uk/>, accessed 11-February-2015
- [30] MIREX HOME - MIREX Wiki [Online]. Available: http://www.music-ir.org/mirex/wiki/MIREX_HOME, accessed 11-February-2015
- [31] A. Cont, D. Schwarz, N. Schnell, and C. Raphael, “Evaluation of real-time audio-to-score alignment,” in *Proc. Int. Symp. Music Inf. Retrieval*, 2007.
- [32] S. Sagayama, T. Nakamura, E. Nakamura, Y. Saito, H. Kameoka, and N. Ono, “Automatic music accompaniment allowing errors and arbitrary repeats and jumps,” in *Proc. Meetings Acoust.*, 2014, vol. 21, 035003, pp. 1–11, Acoustical Society of America.



Tomohiko Nakamura received his B.E. and M.S. degrees from the University of Tokyo, Japan, in 2011 and 2013, respectively. He is currently a Ph.D. student at the University of Tokyo and a Research Fellow of the Japan Society for the Promotion of Science (JSPS). His research interests involve audio signal processing and statistical machine learning. He received the International Award from the Society of Instrument and Control Engineers (SICE) Annual Conference 2011, SICE Best Paper Award (Takeda Award) in 2015, and Yamashita SIG Research Award from the Information Processing Society of Japan (IPSJ) in 2015.

Eita Nakamura received a Ph.D. in physics from the University of Tokyo in 2012. After having been a Post-Doctoral Researcher at the National Institute of Informatics and Meiji University, he is currently a Post-Doctoral Researcher at the Speech and Audio Processing Group at Kyoto University. His research interests include music information processing and statistical machine learning.



Shigeki Sagayama received the B.E., M.S., and Ph.D. degrees from the University of Tokyo, Tokyo, Japan, in 1972, 1974, and 1998, respectively, all in mathematical engineering and information physics. He joined Nippon Telegraph and Telephone Public Corporation (currently, NTT) in 1974 and started his career in speech analysis, synthesis, and recognition at NTT Labs in Musashino, Japan. In 1990, he was Head of the Speech Processing Department, ATR Interpreting Telephony Laboratories, Kyoto, Japan, where he was in charge of an automatic speech translation project. In 1993, he was responsible for speech recognition, synthesis, and dialog systems at NTT Human Interface Laboratories, Yokosuka, Japan. In 1998, he became a Professor of the Graduate School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa. In 2000, he was appointed Professor at the Graduate School of Information Science and Technology (formerly, Graduate School of Engineering), the University of Tokyo. After his retirement from the University of Tokyo, he is a Professor of Meiji University from 2014. His major research interests include the processing and recognition of speech, music, acoustic signals, handwriting, and images. He was the leader of anthropomorphic spoken dialog agent project (Galatea Project) from 2000 to 2003. Prof. Sagayama received the National Invention Award from the Institute of Invention of Japan in 1991, the Director General’s Award for Research Achievement from the Science and Technology Agency of Japan in 1996, and other academic awards including Paper Awards from the Institute of Electronics, Information and Communications Engineers, Japan (IEICEJ) in 1996 and from the Information Processing Society of Japan (IPSJ) in 1995. He is a member of the Acoustical Society of Japan, IEICEJ, and IPSJ.