Merged-Output Hidden Markov Model for Score Following of MIDI Performance with Ornaments, Desynchronized Voices, Repeats and Skips

Eita Nakamura
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan
eita.nakamura@gmail.com

Yasuyuki Saito
Kisarazu National College of Technology
2-11-1 Kiyomidai Higashi, Kisarazu, Chiba, Japan
saito@j.kisarazu.ac.jp

Nobutaka Ono
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan
onono@nii.ac.jp

Shigeki Sagayama*
Meiji University
4-21-1 Nakano, Nakano-ku, Tokyo, Japan
sagayama@meiji.ac.jp

ABSTRACT

A score-following algorithm for polyphonic MIDI performances is presented that can handle performance mistakes, ornaments, desynchronized voices, arbitrary repeats and skips. The algorithm is derived from a stochastic performance model based on hidden Markov model (HMM), and we review the recent development of model construction. In this paper, the model is further extended to capture the multi-voice structure, which is necessary to handle note reorderings by desynchronized voices and widely stretched ornaments in polyphony. For this, we propose merged-output HMM, which describes performed notes as merged outputs from multiple HMMs, each corresponding to a voice part. It is confirmed that the model yields a score-following algorithm which is effective under frequent note reorderings across voices and complicated ornaments.

1. INTRODUCTION

Automated matching of notes in music performances to notes in corresponding scores in real time is called score following, and it is a basic machine-listening tool for real-time applications such as automatic accompaniment and automatic turning of score pages. Since the first studies [1, 2], many studies have been carried out on score following (see [3] for a review of studies in this field, and for more recent studies, see, *e.g.*, [4, 5, 6, 7], just to mention a few). Score-following algorithms generally accept either acoustic signals or symbolic MIDI signals of performances as input. Algorithms for acoustic signals are applicable to a wider range of instruments and situations, and they have been improved over the years [8, 5, 6, 9]. On the other hand, using MIDI inputs has advantages in quick correspondences to onsets and in clean signals [10, 11, 4, 7], and it has potentially vast demand for score following of polyphonic piano performances. We focus on polyphonic MIDI signals for inputs in this paper.

A central problem in score following is to properly and efficiently capture indeterminacies and uncertainties of music performance, which are included in tempos, noise in onset times, dynamics, articulations, ornaments, and also in the way of making performance mistakes, repeats, and skips, especially in performances during practice [7]. Stochastic models are often used to derive algorithms that handle these indeterminacies and uncertainties [3]. Performance mistakes and tempo variations have been treated since the earliest studies [1, 10]. Repeats and skips to restricted score positions were discussed in [4, 12] for monophonic performance, and generalization to arbitrary repeats and skips for polyphonic performance was discussed in [13, 14, 7]. Recently, quantitative analysis and stochastic modeling of performances with ornaments were carried out [15], and an accurate score-following algorithm has been obtained. One of the purposes of this paper is to report the current status of these studies.

In [15], it was found that reorderings of performed notes across voices in complex polyphonic passages such as polyrhythmic passages and passages with many ornaments remains as a major cause of matching errors. The reordering is caused by asynchrony between voices and widely stretched ornaments, manifesting the complicated temporal structure of polyphonic performance [16]. The same problem has been addressed in studies on offline score-performamce matching [17, 18, 19]. It has been observed that the temporal structure is much simpler inside each voice part [1] [17, 18], suggesting that use of voice information is essential for precise score following. Because voice information of performed notes is implicit in piano performance, an algorithm should hold a function to estimate the voice part of each note during score following, and it must be computationally efficient for real-time processing.

In this paper, we propose a score-following algorithm using both voice information and temporal information which can further handle note reorderings due to polyphonic structure. It is derived from a hidden Markov model (HMM) of

* On leave from National Institute of Informatics.

[1] In this paper, a voice part signifies a totality of single or multiple voices.

performance which extends the model in [15] to capturing multi-voice structure. The performed notes are described as merged outputs from multiple HMMs, each corresponding to a voice part. The basic model, which is named merged-output HMM, is also potentially useful for other tasks in music information processing, and we discuss the model and its inference algorithms in detail. A part of this work was reported in [20]. Details and extended discussions of the model and algorithm will be reported elsewhere.

2. TEMPORAL HMM OF PERFORMANCE AND ARBITRARY REPEATS AND SKIPS

In this section, we briefly review our works [7, 15] to prepare for the following sections. For details, see the original papers.

2.1 Temporal HMM

A score-following algorithm should hold a set of complex rules to capture various sources of indeterminacies and uncertainties of music performance mentioned in Section 1. Use of stochastic models has been shown to be effective to derive such an algorithm [3]. One constructs a stochastic model that yields the probability of a sequence of intended score positions and of generated performed notes based on a score, and the score-following problem can be restated as finding the most probable sequence of intended score positions given a performance signal. HMM is particularly suited for this because it effectively describes the sequential, erroneous, and noisy observations of music performance, and there are computationally efficient inference algorithms [21, 8].

The use of temporal information is important for score following of performances including ornaments such as trill, arpeggio, and grace notes, since the clustering of performed notes into musical events, e.g., chords or arpeggios, often becomes ambiguous without it. An HMM was proposed to describe the temporal information explicitly. There are two equivalent representations of the model, one describes time as a dimension in the state space and the other has output probability of inter-onset intervals (IOIs). The latter representation is explained in the following.

First, let i label a unit of score notes that is represented by a state, which will be called a musical event and specified in Section 2.3. The state space of the model is represented by an intended musical event i_m, where $m = 1, \cdots, M$ indexes the performed notes with the total number M. The pitch and onset time of the m-th performed note are denoted by p_m and t_m. The music performance can be modeled as a two-stage stochastic process of choosing the intended musical events first and then outputting the observed performed notes. The first stage is described as transitions between states, and the temporal information can be described as output of IOI $\delta t_m = t_m - t_{m-1}$ at each transition. Assuming that the probability of choosing the state i_m is only dependent on the previous state as $P(i_m|i_{m-1}) = a_{i_{m-1}i_m}$ and the output probability of pitch and IOI is only dependent on the current and the previous states as $P(p_m, \delta t_m|i_{m-1}, i_m) = b_{i_{m-1}i_m}(p_m, \delta t_m)$, the probability of the performance sequence $(p_m, i_m, t_m)_{m=1}^M$ is given as

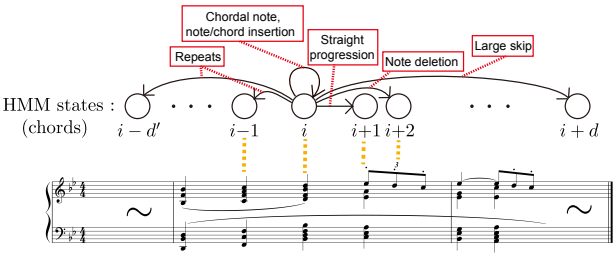


Figure 1. Transitions of the HMM for a simple passage and their interpretations [15].

$$P\left((p_m, i_m, t_m)_{m=1}^M\right) = \prod_{m=1}^{M} a_{i_{m-1}i_m} b_{i_{m-1}i_m}(p_m, \delta t_m), \tag{1}$$

where the factors for $m = 1$ mean the initial probabilities by abuse of notation.

The transition probability a_{ij} describes how players proceed in the score during performance (Figure 1), and the output probability describes how they actually produce performed notes. These probabilities can be obtained from performance data in principle. However, for efficiency of learning parameters, the dependence on the state pair is assumed to be translationally invariant in the state space, and the output probability is factorized into independent pitch and IOI probabilities. Then, $b_{ij}(p, \delta t) = b_j^{\mathrm{pitch}}(p) b_{ij}^{\mathrm{IOI}}(\delta t)$, where we further assumed that the pitch probability is only dependent on the current state for simplicity.

2.2 Repeats and skips, and computational cost

As shown in Figure 1, large repeats and skips are described by the transition probability a_{ij} with large $|j - i|$. Since it is difficult to anticipate all score positions from and to where players make repeats and skips, it is practical to consider arbitrary repeats and skips, which can be expressed as $a_{ij} \neq 0$ for all i and j. In this case, all score positions and transitions must be taken into account at every time, and the computational cost for the conventional inference algorithm is large for long scores. For example, a Viterbi update requires $\mathcal{O}(N^2)$ complexity, where N is the number of states, which is too large for real-time processing when $N \gtrsim 500$.

There are solutions to reduce the computational cost by using simplified models, one of which is the model with uniform repeat/skip probability where a_{ij} is constant for large $|j - i|$. It can be shown that the computational complexity can be reduced to $\mathcal{O}(DN)$ when a_{ij} is constant for $j < i - D_1$ or $j > i + D_2$ ($D = D_1 + D_2 + 1$). The value of D is 3–10 in practice, and hence the complexity is significantly reduced. We can generalize the model to outer-product HMM, where a_{ij} is an outer-product of two vectors for large $|j - i|$ while keeping the computational efficiency. The details of the models and analyses of ten-

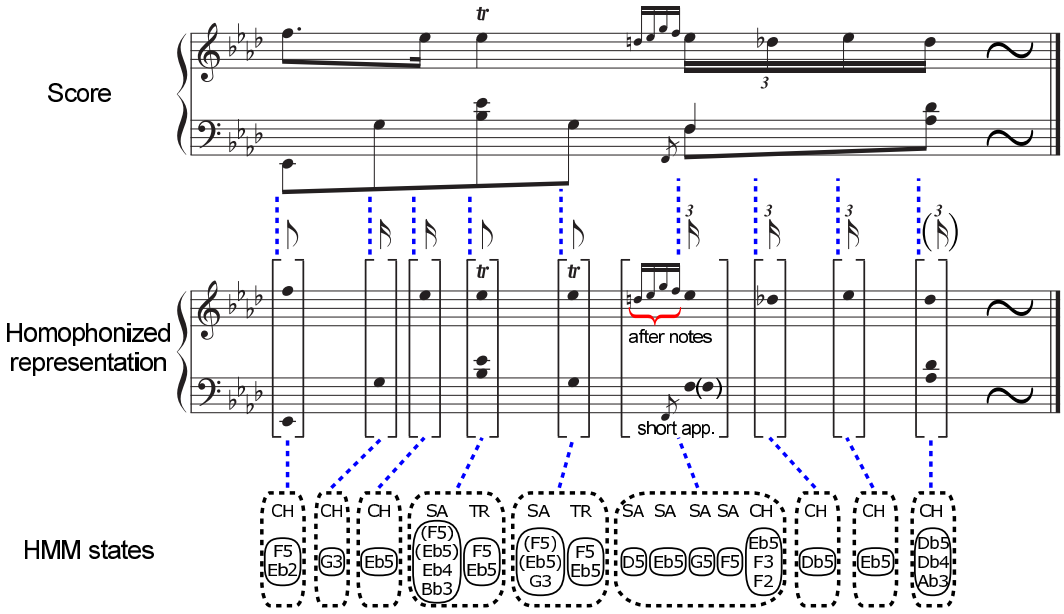


Figure 2. Example of homophonization and HMM state construction. The HMM states are illustrated with their state type and main output pitches. The large (resp. small) smoothed squares indicate top-level (resp. bottom-level) states.

dencies in repeats and skips of actual performance data are given in [7].

2.3 Score representation and state construction

An HMM state must be related to a certain unit of score notes. It can be related to a chord in a simple passage, as in Figure 1. To capture the temporal structure of polyphonic performance with ornaments properly, however, we need more labor. To explain the state construction, we begin with a score representation for a fairly general polyphonic passage. A polyphonic passage $\boldsymbol{H}$, or a score, is defined as a composition of homophonic passages $H_1, \cdots, H_V$ and written as $\boldsymbol{H} = \bigoplus_{v=1}^{V} H_v$, where each H_v ($v = 1, \cdots, V$), which is called a voice, is of the form

$$H = \alpha_1 \beta_1 y_1 \cdots \alpha_n \beta_n y_n. \tag{2}$$

Here y_i is either a chord, a rest, a tremolo, or a glissando, and α_i and β_i denotes after notes and short appoggiaturas, which can be empty if there is none. (A short appoggiatura is a note with an indeterminate short duration notated with a grace note, and an after note is a short appoggiatura which is almost definitely played in precedence to the associated metrical score time.) In the convention, α_i, β_i, and y_i have the same score time, and after notes in α_i is associated with the previous event y_{i-1}.

Given a polyphonic passage, we combine the constituent homophonic passages into a linear sequence of composite factors each containing all onset events at a score time. It is written as

$$\tilde{\boldsymbol{H}} = \tilde{\boldsymbol{\alpha}}_1 \tilde{\boldsymbol{\beta}}_1 \tilde{\boldsymbol{y}}_1 \cdots \tilde{\boldsymbol{\alpha}}_N \tilde{\boldsymbol{\beta}}_N \tilde{\boldsymbol{y}}_N. \tag{3}$$

This procedure is a generalization of Conklin's "homophonization" [22], and we call $\tilde{\boldsymbol{H}}$ the homophonization of $\boldsymbol{H}$ (Figure 2).

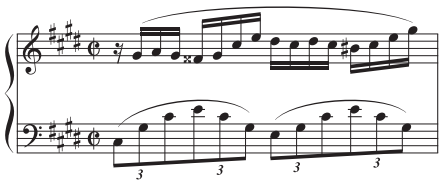
The model is described with a two-level hierarchical HMM, and a state in the top HMM corresponds to a factor $\tilde{\boldsymbol{\alpha}}_i \tilde{\boldsymbol{\beta}}_i \tilde{\boldsymbol{y}}_i$ in $\tilde{\boldsymbol{H}}$. If the factor contains trill, tremolo, or short appoggiaturas, the bottom HMM is constructed with possibly multiple substates as long as the temporal order of the substates is determinate in straight performances without mistakes. Three types for the substates, "CH", "SA", and "TR", each representing a generalized chord, short appoggiatura, and trill events, are considered, and the transition probabilities of the bottom HMM are determined through an argument on expected realizations. The transition probability in the top HMM is similar to that in the simple model in Figure 1, whose values were obtained in [7]. Explicit forms of output probabilities are explained in [15].

3. MERGED-OUTPUT HMM

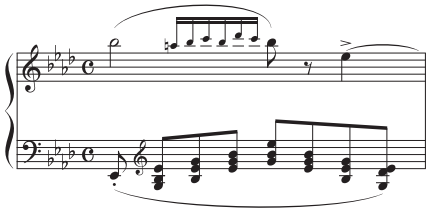
3.1 The idea of merging outputs of multiple models

A potential problem of the model in Section 2 is that it does not properly capture reorderings of performed notes due to voice asynchrony or widely stretched ornaments. Voice asynchrony influences the ordering of performed notes at different score times in different voices, especially in fast or polyrhythmic passages (Figure 3(a)). A widely stretched ornament, typically a long chain of short appoggiaturas, in polyphonic passages can overlap with notes in other voices with different score times (Figure 3(b)).

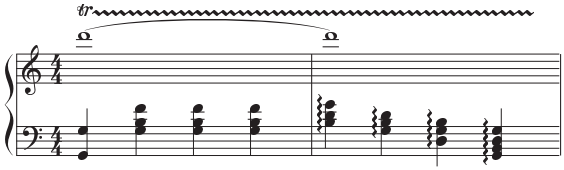
Since the note reorderings can be described by neighboring transitions similarly as insertion and deletion errors, one may wonder if they are already treated properly by the previous model. However, this is not true as long as the translationally-invariant transition probability is assumed because such erroneous transitions are rare in most passages, and probability values obtained from many performances do not reflect such reorderings well, or the whole



(a) Polyrhythmic passage



(b) Passage with a widely stretched ornament



(c) Sustained trill and repeated chords/arpeggios

Figure 3. Examples of passages which can induce errors in score following by a simple (one-part) temporal HMM.

result may be crushed if we adjust the values for particular passages. Changing the probability values for a particular set of states can help, but there remains a problem of automatically identify the corresponding score positions and giving suitable values, which requires knowledge of the structure of the note reorderings. In particular, it is difficult to recognize the structure of the reorderings from the state constructed via homophonization, since the voice structure is contracted and mostly lost in the process of homophonization. If we could preserve the voice structure in the model, it may become much easier.

Another problem arises, for example, when a trill in the right-hand voice part is superposed with a passage with a repeated chord in the left-hand voice part (Figure 3(c)). The matching of the left-hand chords becomes more ambiguous since the long inter-chord IOI in the left-hand voice part is interrupted by small IOIs of trill notes and cannot be observed directly. Of course, we could consider a higher-order Markov model to keep temporal information from far past, but it is not viable in terms of computational efficiency for real-time processing. Again, if we could preserve the voice structure and process notes in different voices separately, the problem seems much reduced.

Given the above problems as well as an observation that the sequential regularity is more well-kept inside each voice part [17, 18], which can be well described with an HMM, one can expect a solution with a model in which polyphonic performance is described with multiple HMMs and outputs of the HMMs are merged into the sequence of performed notes.

3.2 Description of the model

The idea of the following model is to first consider an HMM for each voice, or more precisely, each voice part consisting of several voices, and combine all the HMMs into one model by merging the outputs of the HMMs. The crucial point is that each output observation is emitted from one of the HMMs, and the other HMMs do not make a transition at the time. The whole model is naively a product model of HMMs, but it is shown to have efficient inference algorithms according to this condition. As we will discuss, some interactions between the HMMs can also be introduced while keeping the computational efficiency.

In the following, we describe the merged-output model of general HMMs. For simplicity, we mainly consider the simplest case of two voice parts. Let $a^{(1)}_{ii'}$ and $a^{(2)}_{jj'}$ be transition probabilities of the two models, and let $b^{(1)}_{ii'}(o)$ and $b^{(2)}_{jj'}(o)$ be output probabilities with an output symbol o. We consider the general case that output probabilities depend on both the current and previous states, and that the state spaces of the models can be different.

The state of the totality of the models is represented as a pair (i, j). Introducing a variable $\eta = 1, 2$, which indicates which of the model makes a transition at each time, the state space of the merged-output model is indexed by $k = (\eta, i, j)$. When there is no interaction between the HMMs, they are coupled only by a stochastic process of choosing which of the HMMs transits at each time, which is assumed to be a Bernoulli (coin-toss) process. Let the probability of the Bernoulli process be α_1 and α_2 ($\alpha_1+\alpha_2 = 1$), and then the transition of the merged-output model is described by a probability

$$a_{kk'} = P(k'|k) = \begin{cases} \alpha_1 a^{(1)}_{ii'}\delta_{jj'}, & \eta' = 1; \\ \alpha_2 a^{(2)}_{jj'}\delta_{ii'}, & \eta' = 2. \end{cases} \tag{4}$$

The output of the transition obeys the output probability of the chosen HMM, and it is written as

$$b_{kk'}(o) = P(o|k, k') = \begin{cases} b^{(1)}_{ii'}(o)\delta_{jj'}, & \eta' = 1; \\ b^{(2)}_{jj'}(o)\delta_{ii'}, & \eta' = 2. \end{cases} \tag{5}$$

Eqs. (4) and (5) show that the merged-output model is itself an HMM, which we call merged-output HMM. Each component HMM is called a part HMM. We emphasize that the current state of the non-transiting part HMM is kept in the state label k', and hence the voice-part structure is preserved in the merged-output HMM.

We can also introduce some interactions between the part HMMs as

$$a_{kk'} = \begin{cases} \alpha_1(k) a^{(1)}_{ii'}\delta_{jj'}\phi^{(1)}_{kk'}, & \eta' = 1; \\ \alpha_2(k) a^{(2)}_{jj'}\delta_{ii'}\phi^{(2)}_{kk'}, & \eta' = 2, \end{cases} \tag{6}$$

$$b_{kk'}(o) = \begin{cases} b^{(1)}_{ii'}(o)\delta_{jj'}\psi^{(1)}_{kk'}(o), & \eta' = 1; \\ b^{(2)}_{jj'}(o)\delta_{ii'}\psi^{(2)}_{kk'}(o), & \eta' = 2. \end{cases} \tag{7}$$

Here $\alpha_1(k) + \alpha_2(k) = 1$, and $a_{kk'}$ and $b_{kk'}(o)$ satisfy proper normalization conditions. Applicational examples of the interaction factors $\alpha_{\eta'}(k)$, $\phi^{(\eta')}_{kk'}$, and $\psi^{(\eta')}_{kk'}(o)$ will

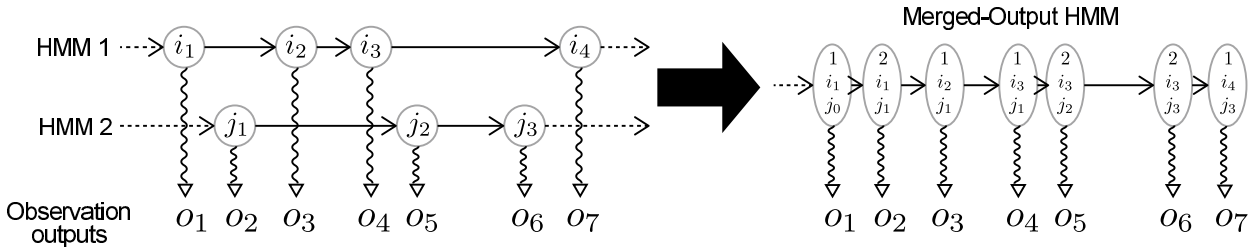


Figure 4. Schematic illustration of merged-output HMM.

be discussed in Section 3.4. The merged-output HMM can also be generalized for more than two voice parts, and we can also consider higher-order Markov models for both η and i_η. A schematic illustration of the merged-output HMM is given in Figure 4.

A similar HMM has been proposed in [23]. The most significant difference is that only one of the component HMMs transits and outputs at each time in the present model, which requires an additional process of choosing the component HMM at each time. Consequently, the way one can introduce interaction factors is also different. As we discussed above, the property is particularly important for the present model to be effectively applied for polyphonic performance.

3.3 Inference algorithms and computational complexity

The Viterbi, forward, and backward algorithms are typically used for inference of HMMs [24]. We discuss the Viterbi algorithm as an example in the following, and similar arguments are valid for the other algorithms. For an HMM with N states in which all states are connected with transitions, a Viterbi update requires $\mathcal{O}(N^2)$ computations of probability. First, suppose a two-part merged-output HMM, and let I and J be the number of states of the part HMMs. Then the number of states of the merged-output HMM is $2IJ$, and the computational complexity is naively $\mathcal{O}(4I^2J^2)$. However, since the transition and output probabilities of the merged-output HMM has a special form in Eq. (6), it is reduced to $\mathcal{O}(2IJ(I+J))$. In general, the computational complexity for an N_p-part merged-output HMM is $\mathcal{O}(N_p I_1 \cdots I_{N_p}(I_1+\cdots+I_{N_p}))$ instead of $\mathcal{O}(N_p^2 I_1^2 \cdots I_{N_p}^2)$, where I_η ($\eta = 1, \cdots, N_p$) is the number of states for each part HMM.

3.4 Merged-output HMM for score following

A performance model which preserves the voice-part structure can be obtained by applying the merged-output HMM to the model described in Section 2. There are options in what unit of voices to model as a part HMM in general. A model with more than two voice parts may be used, but the computational cost rapidly increases with the number of voice parts. For piano performance, the voice asynchrony is most evident between both hands, and we consider a merged-output HMM of two voice parts, which basically correspond to the left-hand and right-hand parts, in the rest of this paper.

Each part HMM is constructed in the same way as in Section 2, except that a score containing voices in each hand is now used. However, the IOI output needs to be considered carefully because it implicitly uses the time information of the previous state, and the information is not kept in the state of the merged-output HMM. In another view, the IOI output is equivalent to consider an additional dimension of time in the state space for each part HMM [15], and in the case of two voice parts, the two dimensions of time cannot be converted to a simple IOI output. In practice, efficient algorithms such as the Viterbi algorithm cannot simply be applied to find the optimal state, and some kind of sub-optimization method must be used. We will come back to this point in Section 4.

In the case of the performance model, the interaction factors of the merged-output HMM in Eq. (6) can be interpreted as follows. For example, when the performance by the left hand happens to be behind the right hand, it is more likely that the left hand will play the delayed note sooner. This indicates that the current state of the merged-output HMM may influence the probability of choosing the transiting part HMM, which can be incorporated in $\alpha_{\eta'}(k)$. In real piano performances, the score positions where the both hands are playing can rarely be far apart, and this can be described by appropriate values of $\phi_{kk'}^{(\eta')}$. Similarly, the factor $\psi_{kk'}^{(\eta')}(o)$ can represent the dependence of the output probability on the relative score position between both hands. Although the interaction factors can be important to improve the score-following result, we do not make full use of them in this paper, for simplicity.

4. SCORE-FOLLOWING ALGORITHM

Given the stochastic generative model of performance described in the previous sections, the score following can be done by finding the most probable hidden state sequence $(i_m)_m$ given observations of performed notes $(p_m, t_m)_m$. To improve computational efficiency for real-time working, we need several refinements of the inference algorithm. First, we need a sub-optimization method for treating the IOI output as mentioned in Section 3.4. For this, the most probable arrival time at each state is memorized and used for calculating the IOI output probability. This makes the inference algorithm as efficient as the Viterbi algorithm.

The second point in computational efficiency is the treatment of arbitrary repeats and skips. Although the method

Table 1. Error rates (%) of the score-following algorithms with the temporal HMM and HMM without modeling ornaments. Pieces indicate those described in the text.

Piece	Onsets	Temporal HMM	HMM w/o ornaments
Couperin	1763	4.71	16.5
Beethoven No.1	17587	3.28	7.83
Beethoven No.2	5861	2.73	5.49
Chopin	16241	10.4	16.2

explained in Section 2.2 can be applied to the present model, it is not sufficient since the state space is quite large. To solve the problem, we set $\phi_{kk'}^{(\eta')} = 0$ for $k' = (\eta', i', j')$ with far apart i' and j'. This in effect reduces the concerned state space significantly. Since transition paths required for large repeats and skips are also eliminated, we reconnect separate states with a small uniform probability. Note that the resulting model is no longer a merged-output HMM, strictly, but they are almost identical in terms of local transitions, for which the precise description of the voice-part structure is most important.

Finally, even after the above refinements of the algorithm, the complexity is large compared to the one-part HMM, and it can be problematic for a very long score. Generally, there is no reason to use the merged-output HMM for a passage where voice asynchrony and ornaments bring no troubling reorderings of performed notes, which is the most typical case. In practice, we can model such a passage within one of the part HMM, say, the first one, and use the second part HMM, or possibly the third, fourth, etc., only for those passages where the voice-part-structured modeling is necessary.

5. EVALUATIONS

5.1 Accuracy of the score-following algorithm

For the purpose of confirming the effectiveness of the score-following algorithms, the accuracy of the algorithms is evaluated with piano performances by several players. First, four pieces with frequently used ornaments were selected to test the algorithm with the temporal HMM [15]. The pieces are the first harpsichord part of Couperin's Allemande à deux clavecins (the first piece of the ninth ordre in the second book of pièces de clavecin), the solo piano part in the second movement of Beethoven's first piano concerto, the third movement of Beethoven's second piano concerto, and the second movement of Chopin's second piano concerto. Each piece was played by two or three pianists during practice and recorded in MIDI format.

Table 1 shows the results of score following in terms of error rates calculated by comparing the estimation result with the hand-matched result. We see that the algorithm based on the temporal HMM with ornaments yielded lower error rates than the one based on the HMM without modeling ornaments. It is confirmed that the explicit modeling of ornaments is indeed effective. Detailed analysis of the results is provided in [15].

Table 2. Error rates (%) of the score-following algorithms by one-part temporal HMM and merging-output HMM. The used test pieces are explained in the text.

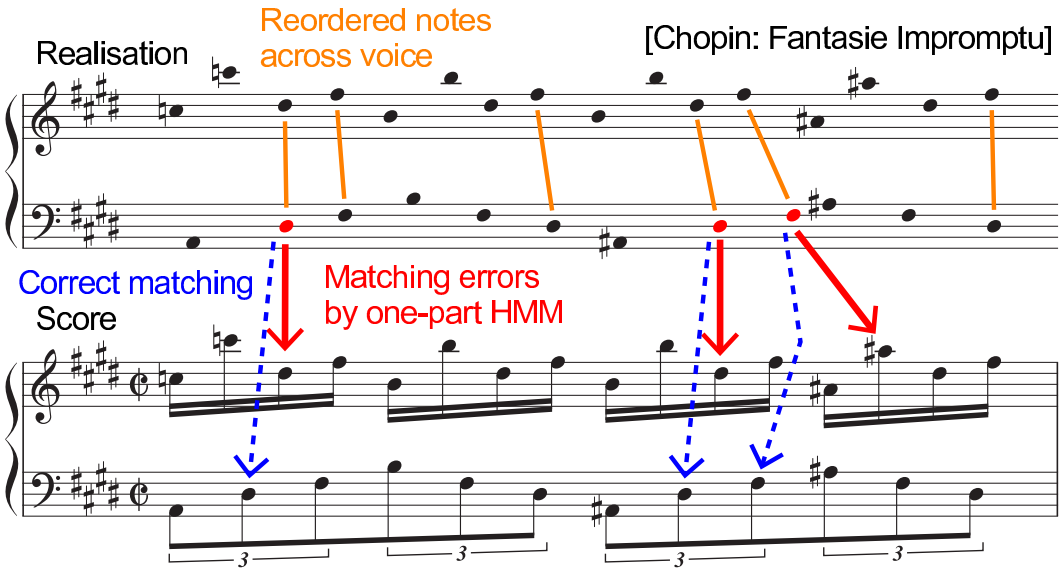
Piece	Onsets	Merged-output HMM	One-part HMM
1	2532	12.8	22.1
2	1226	11.3	23.3

Next, the score-following algorithm by the merged-output HMM is evaluated. As test pieces, we used the allegro part of Chopin's Fantasie Impromptu (piece 1), which include a fast passage with 3 against 4 polyrhythms, and an étude (piece 2) with many sustained trills played in superposition with chords and arpeggios, which was composed for the test purpose (part of it is shown in Figure 3(c) and 5(b)). The pieces were played by two pianists, and the performances were recorded in MIDI format during practice.

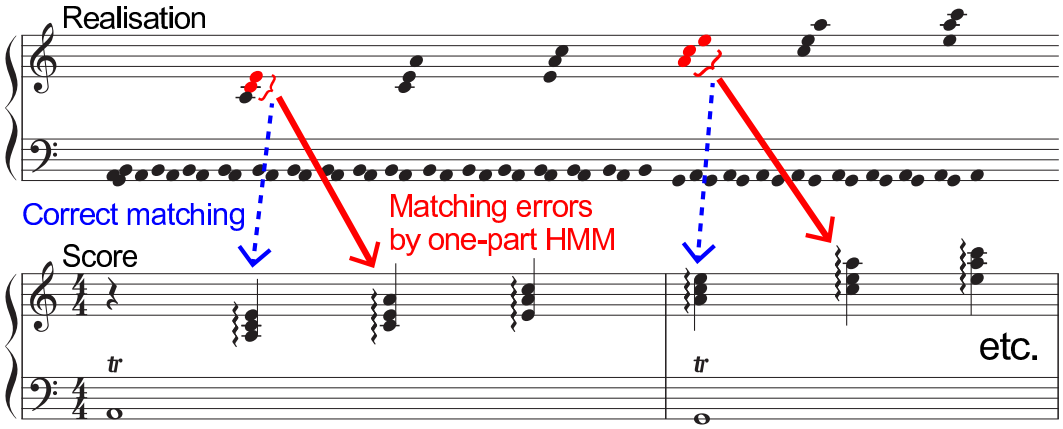
The results are shown in Table 2 and results for a score-following algorithm by a one-part temporal HMM is also shown for comparison. The error rates were calculated by comparing the estimation result with the hand-matched result. There were many trill notes in piece 2, and the error rate was calculated with chords or arpeggios other than trills since the score positions of trill notes are ambiguous in nature. The results show that the error rates are reduced by nearly 50% with the merged-output HMM, compared to the case with the one-part HMM. As examples are shown in Figure 5, there was a tendency that the merged-output HMM estimated score positions more correctly when performed notes were reordered across hands in piece 1, and when repeated chords or arpeggios were played together with sustained trills. On the other hand, the time necessary for following repeats, which we call the following time, were faster with the one-part HMM. For example, the averaged following time in terms of notes for Fantasie Impromptu was 11.8 notes for the merged-output HMM and 7.0 notes for the one-part HMM, where repeats were defined as a backward skip of more than one quater note. The reason is probably that the model uses richer information of simultaneous relations between both hands. The relatively large error rates were due to frequent mistakes, repeats, and skips in the prepared performances.

5.2 Computation time

We have confirmed that the score-following algorithm with the merged-output HMM works in real-time for pieces with roughly 1000 chords, which include the two test pieces, in a PC with moderate computation power. However, it seems hard for pieces with over a few thousands of chords, which may be a drawback of the algorithm, given that the algorithm with the one-part HMM can process pieces with about 10000 chords in real-time [7]. In practice, we can often reduce computational cost by preparing the voice-part structure of the score efficiently as we described in the last paragraph of Section 4. The computational cost mainly comes from treatment of arbitrary repeats and skips, and



(a) A passage from Chopin's Fantasie Impromptu.



(b) A passage with arpeggios and sustained trills.

Figure 5. Examples of score-following results. In each figure, the performed note onsets are written whose horizontal positions are proportional to actual onset times. Notes that are incorrectly matched by the one-part HMM is indicated in red color, and the matched results (resp. correct matchings) are indicated with red straight (resp. blue dashed) arrows. Score-following results for these examples by the merged-output HMM were all correct.

one can also possibly reduce the cost by treating repeats and skips with a simpler model, and use the merging-output HMM for local and precise score-position estimation.

6. CONCLUSIONS

In this paper, we discussed the construction of a score-following algorithm for polyphonic MIDI performance that can handle reorderings of performed notes due to voice asynchrony and widely stretched ornaments in polyphony, particularly focusing on the background model of performance which properly and efficiently capture such deformations in performance. We first reviewed the temporal HMM which is effective for performances with mistakes, ornaments, arbitrary repeats, and skips, and discussed that it is difficult to properly describe those deformations solely with the model. Pointing out the importance of preserving the voice-part structure for capturing voice asynchrony and ornaments in polyphony, we proposed a voice-part-structured model in which outputs from several part HMMs are merged, each of which being a temporal HMM. Several refinements of the score-following algorithm to improve computational efficiency are also explained. We confirmed the effectiveness of the algorithm by evaluating its accuracy.

The key point of the merged-output HMM is that loose inter-dependency between voice parts can be introduced while the sequential regularity inside a voice part is preserved. Since such fabric of inter-dependencies and sequential regularities is common in polyphonic music, the model can potentially be applied to other kinds of music information processing in the domain of both composition and performance. Discovering and extending applications of the model is an important direction in the future. An analogous model for audio signals is also attractive.

It is certainly interesting to use the score-following tech-

nique for automatic accompaniment and other applications. The voice information would also be important in generating musically successful expressive accompaniments and reflecting performer's musicality into them. We are currently working on these issues.

Acknowledgments

The author E.N. thanks Hiroaki Tanaka for useful discussions. This work is supported in part by Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science, No. 23240021 (S.S. and N.O.), No. 26240025 (S.S., Y.S., and N.O.), and No. 25880029 (E.N.).

7. REFERENCES

[1] R. Dannenberg, "An on-line algorithm for real-time accompaniment," *Proc. ICMC*, pp. 193–198, 1984.

[2] B. Vercoe, "The synthetic performer in the context of live performance," *Proc. ICMC*, pp. 199–200, 1984.

[3] N. Orio, S. Lemouton and D. Schwarz, "Score following: State of the art and new developments," *Proc. NIME*, pp. 36–41, 2003.

[4] B. Pardo and W. Birmingham, "Modeling form for online following of musical performances," *Proc. of the 20th National Conf. on Artificial Intelligence*, 2005.

[5] A. Cont, "A coupled duration-focused architecture for realtime music to score alignment," *IEEE Trans. PAMI*, **32(6)**, pp. 974–987, 2010.

[6] A. Arzt, G. Widmer and S. Dixon, "Adaptive distance normalization for real-time music tracking," *Proc. EUSIPCO*, pp. 2689–2693, 2012.

[7] E. Nakamura, T. Nakamura, Y. Saito, N. Ono and S. Sagayama, "Outer-product hidden Markov model and polyphonic MIDI score following," *JNMR*, **43(2)**, pp. 183–201, 2014.

[8] C. Raphael, "Automatic segmentation of acoustic musical signals using hidden Markov models," *IEEE Trans. PAMI*, **21(4)**, pp. 360–370, 1999.

[9] T. Nakamura, E. Nakamura and S. Sagayama, "Acoustic score following to musical performance with errors and arbitrary repeats and skips for automatic accompaniment," *Proc. SMC*, pp. 299–304, 2013.

[10] J. Bloch and R. Dannenberg, "Real-time computer accompaniment of keyboard performances," *Proc. ICMC*, pp. 279–290, 1985.

[11] D. Schwarz, N. Orio and N. Schnell, "Robust polyphonic MIDI score following with hidden Markov models," *Proc. ICMC*, pp. 442–445, 2004.

[12] C. Oshima, K. Nishimoto and M. Suzuki, "A piano duo performance support system to motivate children's practice at home (in Japanese)," *J. Information Processing Society of Japan (IPSJ)*, **46(1)**, pp. 157–171, 2005.

[13] H. Takeda, T. Nishimoto and S. Sagayama, "Automatic accompaniment system of MIDI performance using HMM-based score following (in Japanese)," *Tech. Rep. IPSJ SIGMUS*, pp. 109–116, 2006.

[14] E. Nakamura, H. Takeda, R. Yamamoto, Y. Saito, S. Sako and S. Sagayama, "Score following handling performances with arbitrary repeats and skips and automatic accompaniment (in Japanese)," *J. IPSJ*, **54(4)**, pp. 1338–1349, 2013.

[15] E. Nakamura, N. Ono, S. Sagayama and K. Watanabe, "A stochastic temporal model of polyphonic MIDI performance with ornaments," submitted to *JNMR*.

[16] C. Palmer and C. van de Sande, "Units of knowledge in music performance," *J. Exp. Psych.*, **19(2)**, pp. 457–470, 1993.

[17] P. Desain, H. Honing and H. Heijink, "Robust score-performance matching: Taking advantage of structural information," *Proc. ICMC*, pp. 337–340, 1997.

[18] H. Heijink, L. Windsor and P. Desain, "Data processing in music performance research: Using structural information to improve score-performance matching," *Behavior Research Methods, Instruments, & Computers*, **32(4)**, pp. 546–554, 2000.

[19] B. Gingras and S. McAdams, "Improved score-performance matching using both structural and temporal information from MIDI recordings," *JNMR*, **40(1)**, pp. 43–57, 2011.

[20] E. Nakamura, Y. Saito and S. Sagayama, "Merged-output hidden Markov model and its applications in score following and hand separation of polyphonic keyboard music (in Japanese)," *Tech. Rep. IPSJ SIGMUS*, Mar., 2013.

[21] P. Cano, A. Loscos and J. Bonada, "Score-performance matching using HMMs," *Proc. ICMC*, pp. 441–444, 1999.

[22] D. Conklin, "Representation and discovery of vertical patterns in music," in A. Smaill (eds.), *Music and Artificial Intelligence*, Lecture Notes in Artificial Intelligence, Springer, pp. 32–42, 2002.

[23] Z. Ghahramani and M. Jordan, "Factorial Hidden Markov Models," *Machine Learning*, **29**, pp. 245–273, 1997.

[24] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, **77(2)**, pp. 257–286, 1989.