Autoregressive hidden semi-Markov model of symbolic music performance for score following

Eita Nakamura, Philippe Cuvillier, Arshia Cont, Nobutaka Ono, Shigeki Sagayama

To cite this version:

Eita Nakamura, Philippe Cuvillier, Arshia Cont, Nobutaka Ono, Shigeki Sagayama. Autoregressive hidden semi-Markov model of symbolic music performance for score following. 16th International Society for Music Information Retrieval Conference (ISMIR), Oct 2015, Malaga, Spain. hal-01183820

HAL Id: hal-01183820

https://hal.inria.fr/hal-01183820

Submitted on 11 Aug 2015

AUTOREGRESSIVE HIDDEN SEMI-MARKOV MODEL OF SYMBOLIC MUSIC PERFORMANCE FOR SCORE FOLLOWING

Eita Nakamura[1] **Philippe Cuvillier**[2] **Arshia Cont**[2]
Nobutaka Ono[1] **Shigeki Sagayama**[3]
[1] National Institute of Informatics, Tokyo 101-8430, Japan
[2] Institut de Recherche et Coordination Acoustique/Musique (IRCAM), 75004 Paris, France
[3] Meiji University, Tokyo 164-8525, Japan
eita.nakamura@gmail.com, philippe.cuvillier@ircam.fr, Arshia.Cont@ircam.fr
onono@nii.ac.jp, sagayama@meiji.ac.jp

ABSTRACT

A stochastic model of symbolic (MIDI) performance of polyphonic scores is presented and applied to score following. Stochastic modelling has been one of the most successful strategies in this field. We describe the performance as a hierarchical process of performer's progression in the score and the production of performed notes, and represent the process as an extension of the hidden semi-Markov model. The model is compared with a previously studied model based on hidden Markov model (HMM), and reasons are given that the present model is advantageous for score following especially for scores with trills, tremolos, and arpeggios. This is also confirmed empirically by comparing the accuracy of score following and analysing the errors. We also provide a hybrid of this model and the HMM-based model which is computationally more efficient and retains the advantages of the former model. The present model yields one of the state-of-the-art score following algorithms for symbolic performance and can possibly be applicable for other music recognition problems.

1. INTRODUCTION

For the last thirty years the real-time matching of music performance to the corresponding score (called score following) has been a popular field of study motivated by applications such as automatic music accompaniment and score-page turning system [1, 2, 3, 4, 5, 6, 7, 8]. We study here score following of polyphonic symbolic (MIDI) performance. A central problem in score following is to properly capture the variety of music performance in a computationally efficient manner. A commonly studied way to capture this variety and develop an effective score-following algorithm is to use stochastic models of music performance (Sec. 2.1, see also [3]).

Hidden Markov models (HMMs) have been applied to score following of symbolic performance and provided currently best results [4, 7, 9]. In these models, a musical event in the score, i.e. note, chord, trill, etc., is represented as a state, and the performed notes are described as outputs of an underlying state transition process. Memoryless statistical dependence is assumed for both output and transition probabilities for the sake of computational efficiency. Due to these simplifications the models cannot well describe significant features of performance data such as the number of performed notes per event and the total duration of a trill.

Phenomenologically, music performance can be regarded as a hierarchical process of producing musical notes: The higher level describes performer's progression in the score in units of musical events, and the lower level describes the production of individual notes [9, 10]. We describe this process in terms of a hidden semi-Markov model (HSMM) [11] with an autoregressive extension [12] (Sec. 2) and incorporate the above features into the model. With some simplifications, the model is reduced to a previously studied HMM [9]. We compare these models in the informational and algorithmic aspects and argue that the present model is advantageous for score following especially for scores with trills, tremolos, and arpeggios (Sec. 3). Empirical confirmation of this fact is given by comparing the accuracy of score following and analysing the errors (Sec. 4). Finally remaining problems and future prospects are discussed (Sec. 5).

2. AUTOREGRESSIVE HIDDEN SEMI-MARKOV MODEL OF SYMBOLIC PERFORMANCE

2.1 Stochastic description of music performance

Music performances based on a score have a wide variety because of indeterminacies inherent in musical score descriptions and uncertainties in movements of performers and musical instruments. These indeterminacies and uncertainties are included in tempos, noise in onset times, dynamics, articulations, ornaments, and also in the way of making performance errors, repeats, and skips [7]. In order to perform accurate and robust score following, we need to incorporate (maybe implicit) rules into the algorithm to capture this variety.

A way to do this is to construct a stochastic model of music performance and describe those indeterminacies and uncertainties in terms of probability. A score-following algorithm can be developed as an inference problem of the model. We shall take this approach in the following, which has been proved to be successful in score following.

2.2 Model of performer's progression in the score

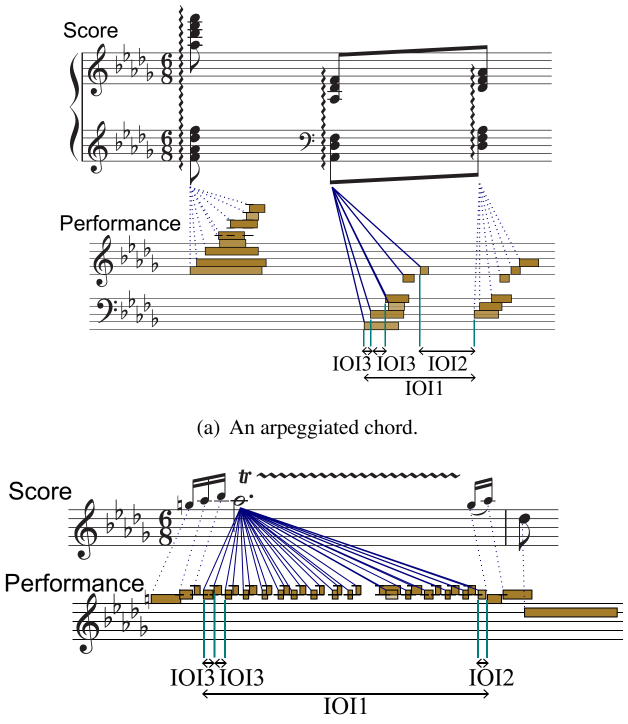
Let us present the model. We model music performance as a combination of subprocesses in two levels. The higher-level (top-level) process describes the performer's progression in the score in units of musical events that are well-ordered in performances without errors. We take a chord (possibly arpeggiated), a trill/tremolo, a short appoggiatura, or an after note [1] as a unit and represent it with a state (top state). Let i label a top state. Then the performer's progression can be described as successive transitions between these states denoted by $i_{1:N} = (i_1, \cdots, i_N)$ (N is the number of performed MIDI notes). We will use the symbol $n(= 1, \cdots, N)$ to index the performed notes that are ordered according to the onset time, and i_n represents the corresponding musical event.

The probability $P(i_{1:N})$ describes statistical tendencies of performances. Simplifications are necessary to construct a performance model yielding a computationally tractable algorithm. A typical assumption is that the probability is decomposed into transition probabilities: $P(i_{1:N}) = \mathbf{\Pi}_{n=1}^{N} P(i_n|i_{n-1})$ ($P(i_1|i_0) \equiv P(i_1)$ denotes the initial distribution). The probability $P(j|i)$ represents the relative frequency of straight progressions to the next event ($j = i + 1$), insertions of events ($j = i$), deletions of an event ($j = i + 2$), and repeats or skips (if $|j - i - 1| > 1$). These probability values can be estimated from performance data. With the assumption that $P(i|j)$ is only dependent on $i - j$, the probability values have been estimated with piano performance data in a previous study ([7], Table 3).

2.3 Model of production of performed notes

The lower-level process describes the production of performed notes during each musical event. Because dynamics and articulations are generically highly indeterminate, we focus on pitch and onset time which are denoted by p_n and t_n. For example, multiple notes are performed at a chord or a trill (Fig. 1). Note that where as chords are written in musical scores as simultaneous notes, performed MIDI notes are serialised and never exactly simultaneous. Thus p_n is always a single pitch.

Let us first consider the number of performed notes per event. For "chords" (meaning a set of all simultaneous notes in the score), short appoggiaturas, and after notes, the expected number of notes is determinate, but it can be modified as a result of added or deleted notes by mistake. For trills and (unmeasured) tremolos, the number of notes are indeterminate since the speed of ornaments varies among realisations. We describe this situation with a probability distribution $d_i(s)$ where s denotes the number of performed notes ($\Sigma_{s=1}^{\infty} d_i(s) = 1$). For example, the function $d_i(s)$ peaks at the indicated number of notes when event i is a chord. When event i is a one-note trill, the peak can be written as $s_i^{\rm peak} \simeq \nu_i v / \delta t_{\rm trill}$, where $\delta t_{\rm trill}$, ν_i, and v denote the average inter-onset time interval (IOI) of successive notes of a trill, the note value of event i, and the (inverse) tempo in units of "second per unit note value". Because currently we do not have a strong empirical basis for determining the shape of $d_i(s)$, we simply assume it is a normal distribution $d_i(s) = N(s; s_i^{\rm peak}, \sigma_i)$ with $s_i^{\rm peak}$ given in Sec. 2.3, and leave σ_i as an adjustable parameter.



(a) An arpeggiated chord.

(b) Trill with preceding short-appoggiaturas and after notes.

Figure 1. Examples of musical events and performed notes. The three types of time intervals IOI1, IOI2, and IOI3 are explained in the text.

Next the pitch of each performed note of event i can be described with a probability $P_i^{\rm pitch}(p)$, which is assumed to be independent for each note for the sake of computational efficiency. The probability values for incorrect pitches represent the possibility and frequencies of pitch errors. An approximate distribution of $P_i^{\rm pitch}(p)$ has been estimated previously (Eq. (30) of [7]) with piano performance data, where the probability of pitch errors is assumed to be uniform for all score notes.

Finally we consider the description of onset times. A natural assumption of time translational invariance requires the model to be only dependent of time intervals. There

[1] Here 'after notes' are defined as grace notes that are played in precedence over the associated beat. A typical example is grace notes after a trill.

are (at least) three different kinds of time intervals relevant in locally describing onset times of music performance: (IOI1) The time interval between the first notes of succeeding events, which is typically the duration of an event, (IOI2) the time interval between the first note of an event and the last note of its previous event, and (IOI3) the time interval between succeeding performed notes within an event (Fig. 1). Assuming that the probability of these time intervals depends only on the current and previous states for simplicity and computational efficiency, it has the form $P_\kappa(\delta t|i_{n-1}, i_n, v)$ ($\kappa = \text{IOI1}, \text{IOI2}, \text{IOI3}$) where δt and v denote the relevant time interval and the tempo. Based on the experience that time interval IOI3 is mostly dependent on the relevant event and almost independent of tempo and other contexts, we further simplify the functional form as $P_{\text{IOI3}}(\delta t|i_n)$. Note that the time intervals IOI1 and IOI2 are not independent quantities if we retain all historical information on time, but they have different importance when we take the Markovian description explained below.

2.4 Autoregressive hidden semi-Markov model

The integration of the models in Secs. 2.2 and 2.3 can be described in terms of an extension of the HSMM. In one of equivalent formulations [13] (also Sec. 3.3 of Ref. [11]), a semi-Markov model can be represented as a Markov model on an extended state space. The extended state space is indexed by a pair (i, s) of the top state i (corresponding to a musical event) and a counter of performed notes $s = 1, 2, \cdots$ [2] with a transition probability

$$P(i_n, s_n|i_{n-1}, s_{n-1}) = \delta_{s_n,1} P(i_n|i_{n-1}) P^{\text{exit}}_{i_{n-1}}(s_{n-1}) + \delta_{s_n,s_{n-1}+1}\delta_{i_n,i_{n-1}} \left(1 - P^{\text{exit}}_{i_{n-1}}(s_{n-1})\right) \quad (1)$$

where

$$P^{\text{exit}}_i(s) = d_i(s)/\Sigma^\infty_{s'=s} d_i(s'). \quad (2)$$

Here δ in Eq. (1) denotes Kronecker's delta. The exiting probability in Eq. (2) represents the probability that the performer moves to another event given that she has already played s notes at event i. The first term in the right-hand side of Eq. (1) describes the probability that the performer moves to event i_n after having played s_{n-1} notes of event i_{n-1}. The second term describes the probability that the performer stays at event i_n and sound another note after having played s_{n-1} notes. In this way, this model describes the integrated process of performer's progression in the score and the production of performed notes.

The pitches and onset times of the performed notes can be described with output probabilities associated with this semi-Markov process. We assume the statistical independence of pitch and onset time for simplicity. The output probability of pitch is given by $P(p_n|i_n, s_n) = P^{\text{pitch}}_{i_n}(p_n)$.

The output probability of the onset time of the n-th note is given as

$$P(t_n|i_n, s_n, i_{n-1}, s_{n-1}, v, t_{1:n-1}) = \begin{cases} w_1 P_{\text{IOI1}} + w_2 P_{\text{IOI2}}, & s_n = 1; \\ P_{\text{IOI3}}, & s_n \neq 1 \end{cases} \quad (3)$$

where

$$P_{\text{IOI1}} = P_{\text{IOI1}}(t_n - t_{n-s[n-1]}|i_n, i_{n-1}, v), \quad (4)$$

$$P_{\text{IOI2}} = P_{\text{IOI2}}(t_n - t_{n-1}|i_n, i_{n-1}, v), \quad (5)$$

$$P_{\text{IOI3}} = P_{\text{IOI3}}(t_n - t_{n-1}|i_n)\delta_{i_n i_{n-1}}. \quad (6)$$

(Here we have written $s[n-1] = s_{n-1}$ to display the equation with clarity.) The three cases correspond to the three kinds of time intervals explained in Sec. 2.3. Because both probabilities for IOI1 and IOI2 have relevance in score following, we have used a mixture probability of them ($w_1 + w_2 = 1$). Such output probabilities with conditional dependence on the previous outputs have been considered in some studies on speech processing, and we call the model autoregressive semi-Markov model based on the convention of previous studies [12]. A graphical representation of the model is given in Fig. 2.

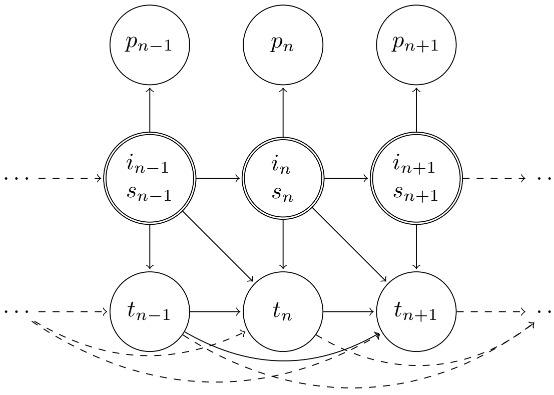


Figure 2. Graphical representation of the autoregressive hidden semi-Markov model of symbolic music performance. The stochastic variables are explained in the text.

The distributions P_{IOI1}, P_{IOI2}, and P_{IOI3} can be estimated by analysing performance data. The functions P_{IOI2} and P_{IOI3} have previously been estimated with piano performance data [9]. It has been shown there that, in the most important case that $i_n = i_{n-1}+1$ (straight transition to the next event), $P_{\text{IOI2}}(\delta t|i+1, i, v)$ is well approximated by a Cauchy distribution of the form

$$\text{Cauchy}(\delta t; v(\tau^{\text{end}}_i - \tau_i) - \text{dev}_i, 0.4\text{ s}). \quad (7)$$

Here $\text{Cauchy}(x; \mu, \Gamma)$ denotes the Cauchy distribution with mean μ and width Γ, and τ_i is the onset score time of event i, τ^{end}_i is the score time after which no new onsets of event i can occur, and dev_i describes the 'stolen time' of event i whose expectation value is given as the number of short appoggiaturas and arpeggiated notes times the average IOI of the corresponding notes. Using this result, we can estimate P_{IOI1} in the case that $i_n = i_{n-1}+1$ as

$$P_{\text{IOI1}}(\delta t|i+1, i, v) = \text{Cauchy}(\delta t; v\nu_i, 0.4\text{ s}) \quad (8)$$

[2] Remark: In the present model, s counts the number of notes played during a musical event. This is not the durational time (in seconds) spent on that event, which is described with time interval IOI1.

where $\nu_i = \tau_{i+1} - \tau_i$ is the note value of event i. The distribution $P_{\rm IOI3}$ was estimated with measurements on IOIs of chordal notes and ornaments (see Secs. 3.3 and 4.2 of [9]).

Finally, tempo v_n is estimated online with a separate model, for which we use a method based on switching Kalman filter (see Sec. 3.4 of [9]). In summary the complete-data probability $P(i_{1:n}, s_{1:n}, t_{1:n}, p_{1:n})$ is given as the following recursive product:

$$\prod_{m=1}^{n} \Big[P(t_m | i_m, s_m, i_{m-1}, s_{m-1}, v_{m-1}, t_{1:m-1}) \cdot P(i_m, s_m | i_{m-1}, s_{m-1}) P^{\rm pitch}_{i_m}(p_m) \Big]. \quad (9)$$

3. COMPARISON WITH OTHER MODELS

3.1 Relation to the HMM-based model

So far the state-of-the-art method for symbolic score following is developed with a performance model based on a standard HMM [9]. The current model can be seen as an extension of this performance model in two ways. First the transition probability of the HMM is realised as a special case of the transition probability in Eq. (1) with exiting probabilities $P^{\rm exit}_i(s)$ constant in s. Specifically, it is given as the inverse of the expected number of performed notes in event i. As is well known, this constraint leads to a geometrically distributed $d_i(s)$ with a peak at $s = 1$, which is a bad approximation for a large chord or a long trill/tremolo.

The second difference is the structure of output probabilities for onset times. In the standard HMM, the Markovian condition is assumed on the output probability of onset times. Thus the model describes only time intervals IOI2 and IOI3, and the probability distribution for IOI1 in Eq. (3) is ignored. In other words, the IOI output probability of the HMM assumes $w_1 = 0$ and $w_2 = 1$ in that equation. This means that the total duration of a trill/tremolo or an arpeggios is poorly captured with the HMM.

These differences have important effects when the models are applied to score following. For score following, the pitch information is generically most important. When there are musical events with similar pitch contents in succession, however, the information on onset times and the number of performed notes play more significant roles in correctly matching notes. For example, to correctly match performed notes of succeeding trills/tremolos, the number of notes and the duration of each trill/tremolo are important viewpoints. Since they are not well captured in the HMM, the autoregressive HSMM would work better in this case. Similar situations arise for successions of arpeggios, where the time intervals IOI2 and IOI3 are largely variable among realisations. On the other hand, the time intervals IOI1 and IOI2 are almost same for successive normal chords and these IOIs carry much information necessary to cluster them. Thus the models are expected to have similar effects for passages without ornaments.

3.2 Comparison with the preprocessing method

To solve the problems with ornaments for score following, a preprocessing method has been proposed long ago [14]. The idea is to preprocess performed notes so that ornamental notes are not sent to the matching module directly. While the method can work for scores with not-heavy polyphonic ornamentation and performances with infrequent errors, the preprocessing can fail when there are errors or unexpected repeats or skips near ornaments. Because a direct comparison showed that the HMM outperformed the preprocessing method for piano performances with errors, repeats, and skips [9], we compare our model only with the HMM in Sec. 4.

3.3 Computational cost

For score following, we find the most probable hidden state sequence given the input performance. In order to realise real-time processing, the computational cost of the estimation algorithm must be sufficiently small. We here compare the present model and the HMM discussed in Sec. 3.1 in terms of the computational cost.

The Viterbi algorithm can be applied for HMMs to estimate states. Let us denote the product of the transition probability and the output probability as $a_{ij}(o) = P(j|i) \cdot P(o|i,j)$ where o represents pitch and onset time. The Viterbi update equation can be expressed as the following recursive equation

$$\hat{p}_N(i_N) \equiv \max_{i_1,\cdots,i_{N-1}} \left[\prod_{n=1}^{N} a_{i_{n-1}i_n}(o_n) \right] \quad (10)$$

$$= \max_{i_{N-1}} \big[\hat{p}_{N-1}(i_{N-1}) a_{i_{N-1}i_N}(o_N) \big]. \quad (11)$$

The number of states is N since a state corresponds to a musical event in the score. If we allow arbitrary progressions in the score including repeats and skips, a direct application of the Viterbi algorithm requires $\mathcal{O}(N^2)$ computations of probability for each update. When the probability matrix $a_{ij}(o)$ can be represented as a sum of a band matrix α_{ij} of width D and an outer product of two vectors S_i and r_j, the computational complexity can be reduced to $\mathcal{O}(DN)$ with a recombination method [7]. Intuitively, α_{ij} describes probabilities corresponding to transitions between neighbouring states, which have larger probabilities, and S_i and r_j represent probabilities corresponding to large repeats and skips, which typically have very small probabilities. Substituting $a_{ij}(o) = \alpha_{ij} + S_i r_j$ into Eq. (11), we see α_{ij} induces $O(DN)$ complexity and $S_i r_j$ induces $O(N)$ complexity by a recombination. This simplified transition probability matrix is used in previous studies to enable real-time processing for long scores.

It is clear from the formulation of the autoregressive HSMM in Sec. 2.4 that the standard Viterbi algorithm can also be applied to the model. In practice, we put an upper bound on the number of performed notes $s^{\rm max}_i$ for each event i, and the number of states of the HSMM is $\Sigma_i s^{\rm max}_i \equiv SN$ where S is the average of $s^{\rm max}_i$. Because of the special form of transition probabilities in Eq. (1), the computational complexity for one Viterbi update is generically

Table 1. Error rates (%) of score following with the autoregressive HSMM ("HSMM"), the hybrid model ("Hybrid"), and the HMM [9]. The first four pieces indicate Couperin's Allemande à deux clavecins, the solo piano part of Beethoven's first piano concerto, Beethoven's second piano concerto, and Chopin's second piano concerto [9], and the last two pieces are explained in the text.

Piece	# Notes	HSMM	Hybrid	HMM
Couperin	1763	5.50	6.02	6.66
Beethoven 1	17587	3.16	3.13	3.16
Beethoven 2	5861	2.01	2.20	2.35
Chopin	16241	9.22	9.22	11.1
Debussy	3294	3.64	3.58	4.66
Tchaikovsky	2245	0.40	0.40	4.55

$\mathcal{O}(SN^2)$. When we apply the recombination method in Ref. [7], the complexity can be reduced to $\mathcal{O}(DSN)$ for the outer-product type transition probability. Note that the width D in the top-level transition probability matrix induces SD transitions between HSMM states. Consequently the computational cost of the model is about S times larger than its reduced HMM. For example, if we set $s_i^{\max}$ as twice the number of expected notes per event, $S \simeq 3$–10 for a score with a modest degree of polyphony, and it increases if there are many large chords or long trills/tremolos.

3.4 Hidden hybrid Markov/semi-Markov model

As discussed in Sec. 3.1, there are reasons that the present model yields better results for score following than the HMM, but it is at the cost of increased computational cost, which is unwanted for long scores. On the other hand, most of the musical events in scores are normal chords (or single notes) for which the HMM already yields good results. Therefore if we combine the HMM state representation for normal chords and the autoregressive HSMM state representation for other ornamented events, it would be possible to obtain an improved score-following algorithm with minimal increase in computational cost. Such a combination of HMM and HSMM can be achieved in the framework of hidden hybrid Markov/semi-Markov model [5, 15]. In the hybrid model, normal chords are represented with HMM states and other events (i.e. trill, tremolo, arpeggio, short appoggiatura, and after notes) are represented with HSMM states. For this model the computational complexity of the Viterbi algorithm takes the same form as the autoregressive HSMM, by substituting $s_i^{\max} = 1$ for HMM states in $S = \Sigma_i s_i^{\max}/N$.

4. COMPARING THE ACCURACY OF SCORE FOLLOWING

To evaluate and compare the discussed models with respect to the accuracy of score following, we implemented three score-following algorithms based on the autoregressive HSMM (Sec. 2.4), the hybrid model (Sec. 3.4), and the HMM [9], and run these algorithms for music performance data containing various ornaments. In addition to the piano performance data used in Ref. [9] which contain performance errors, repeats and skips, we used collected piano performances of passages in Debussy's En Blanc et Noir with successions of tremolos (the first piano part in the second movement) and the solo piano part of Tchaikovsky's first piano concerto with his typical successions of wide arpeggios (the last section of the second movement).

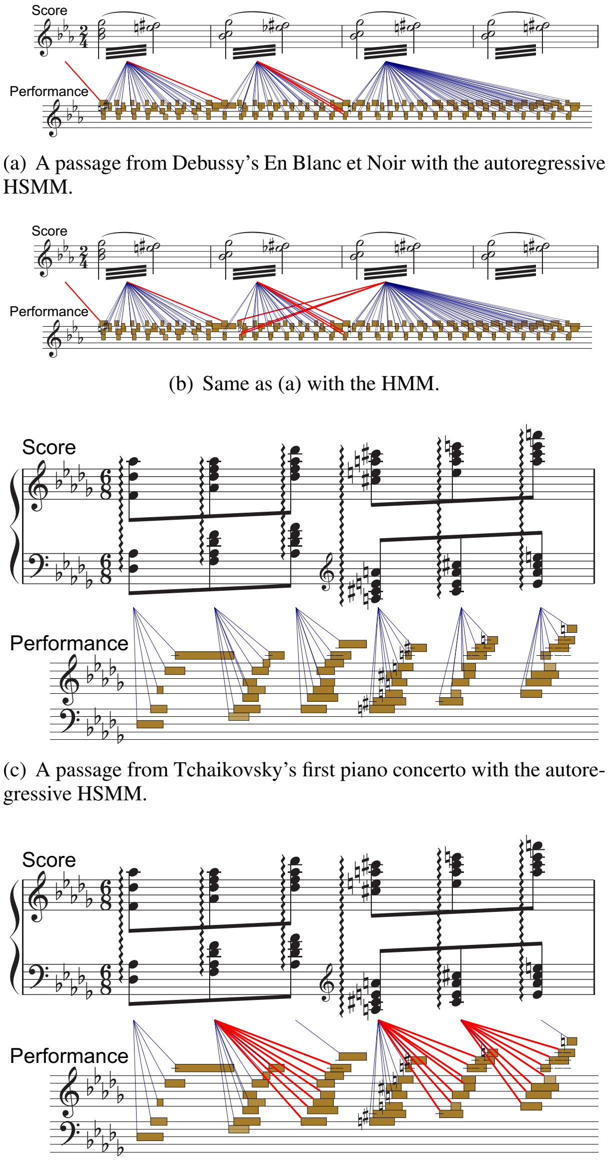
Table 2. Number of mismatched notes of various types. Each type is explained in the text. The same abbreviations for the models as in Table 1 are used.

Type	# Notes	HSMM	Hybrid	HMM
Trill	8159	282	281	508
Tremolo	2603	115	115	151
Arpeggio	1081	36	33	127
Other ornaments	2401	340	339	362
Other	32030	1580	1599	1673

The additional parameters σ_i for the autoregressive HSMM and the hybrid model were set as follows: $\sigma_i = 0.4 s_i^{\mathrm{peak}}$ for trills and tremolos and $\sigma_i = 1$ otherwise. The mixture weights for the output probability for time intervals IOI1 and IOI2 were set as $w_1 = w_2 = 1/2$. These parameters were used as a benchmark and there is a room for further optimisation.

For the evaluation measure, we calculated the error rate, which is defined as the proportion of mis-matched notes to the total number of performed notes. There were performed notes that are difficult to associate with any score notes even for humans, which naturally appear in real data. While they were included in the input data, they were not used in the calculation of error rates. Results are shown in Table 1, where we see that the autoregressive HSMM and the hybrid model had similar accuracies, and the HMM had the worst accuracy overall. (Slight differences in the values for the HMM compared to those in Ref. [9] are mainly due to slight corrections of the implementation.) For detailed error analysis, we list the frequencies of classified matching errors in Table 2. Here the numbers indicate the total number of matching errors in the whole data for each type. Ornaments are classified into the first four types, and other notes are gathered in the last type. Significant reduction of matching errors is observed in the first three types (trill, tremolo, and arpeggio), and other types of matching errors are also reduced but rather slightly in the reduction rate.

Two example results of score following are shown in Fig. 3, which represent typical situations where the autoregressive HSMM worked better than the HMM. In the first example, the passage includes a succession of tremolos with similar pitch contents. We see some of the mismatched notes with the HMM are correctly matched with the autoregressive HSMM. Similarly the mismatched notes with the HMM are all correctly matched with the autoregressive HSMM for a succession of wide arpeggios in the



(a) A passage from Debussy's En Blanc et Noir with the autoregressive HSMM.

(b) Same as (a) with the HMM.

(c) A passage from Tchaikovsky's first piano concerto with the autoregressive HSMM.

(d) Same as (c) with the HMM.

Figure 3. Example results of score following with the autoregressive HSMM and the HMM [9]. Mismatched notes are indicated with bold red lines.

Table 3. Averaged computation time (ms) required for one Viterbi update. The same abbreviations for the models and the musical pieces as in Table 1 are used.

Piece	HSMM	Hybrid	HMM
Couperin	1.6	1.1	0.3
Beethoven 1	5.9	2.9	1.1
Beethoven 2	7.0	3.0	1.6
Chopin	7.1	3.5	1.2
Debussy	0.9	0.8	0.1
Tchaikovsky	1.2	1.0	0.1

second example. These results are consistent with the discussion in Sec. 3.1.

We also measured the required computation time (Table 3). The computation time for each Viterbi update is constant over time, and the algorithms were run on a laptop with moderate computation power. The results confirm our expectation that the use of hybrid model for score following has practical advantages over the autoregressive HSMM in the computation time and the HMM in the accuracy.

5. CONCLUSION

We explained reasons that the present model of symbolic music performance based on autoregressive HSMM is more advantageous for score following than previously studied HMMs, and we have confirmed this empirically by comparing the accuracy of score following and analysing the matching errors. Because a semi-Markov model can be seen as a Markov model with an extended state space as we have explained, we can apply to the present model the methods for HMMs to improve score following [7, 16]. In particular, this is important to reduce matching errors occurring after repeats and skips and those due to reordered notes in the performance, which were the main factors of remaining errors.

It would be interesting to apply the present model for music/rhythm transcription and related problems. Because the model describes both the total duration and the internal temporal structure of ornaments, it would be possible to detect ornaments from performances without a score and integrate the results into music transcription.

6. ACKNOWLEDGEMENTS

This work is partially supported by NII MOU Grant in fiscal year 2014 and Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science, No. 26240025 (S.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] P. Cuvillier and A. Cont, "Coherent time modeling of semi-Markov models with application to real-time audio-to-score alignment," *Proc. IEEE MLSP*, 6 pages, 2014.

[9] E. Nakamura, N. Ono, S. Sagayama and K. Watanabe, "A stochastic temporal model of polyphonic MIDI performance with ornaments," to appear in *JNMR*, 2015.

[10] N. Orio and F. Déchelle, "Score following using spectral analysis and hidden Markov models," *Proc. ICMC*, pp. 1708–1710, 2001.

[11] S.-Z. Yu, "Hidden semi-Markov models," *Artificial Intelligence*, **174**, pp. 215–243, 2010.

[12] J. Bilmes, "Graphical models and automatic speech recognition," in *Mathematical foundations of speech and language processing* (Springer New York), pp. 191–245, 2004.

[13] M. Russel and A. Cook, "Experimental evaluation of duration modelling techniques for automatic speech recognition," *Proc. ICASSP*, pp. 2376–2379, 1987.

[14] R. Dannenberg and H. Mukaino, "New techniques for enhanced quality of computer accompaniment," *Proc. ICMC*, pp. 243–249, 1988.

[15] Y. Guédon, "Hidden Hybrid Markov/Semi-Markov Chains," *Computational Statistics and Data Analysis*, **49**, pp. 663–688, 2005.

[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," *Proc. Joint ICMC|SMC 2014*, pp.1185-1192, 2014.