PERFORMANCE ERROR DETECTION AND POST-PROCESSING FOR FAST AND ACCURATE SYMBOLIC MUSIC ALIGNMENT

Eita Nakamura
Kyoto University
enakamura@sap.ist.i.kyoto-u.ac.jp

Kazuyoshi Yoshii
Kyoto University/RIKEN AIP
yoshii@kuis.kyoto-u.ac.jp

Haruhiro Katayose
Kwansei Gakuin University
katayose@kwansei.ac.jp

ABSTRACT

This paper presents a fast and accurate alignment method for polyphonic symbolic music signals. It is known that to accurately align piano performances, methods using the voice structure are needed. However, such methods typically have high computational cost and they are applicable only when prior voice information is given. It is pointed out that alignment errors are typically accompanied by performance errors in the aligned signal. This suggests the possibility of correcting (or *realigning*) preliminary results by a fast (but not-so-accurate) alignment method with a refined method applied to limited segments of aligned signals, to save the computational cost. To realise this, we develop a method for detecting performance errors and a realignment method that works fast and accurately in local regions around performance errors. To remove the dependence on prior voice information, voice separation is performed to the reference signal in the local regions. By applying our method to results obtained by previously proposed hidden Markov models, the highest accuracies are achieved with short computation time. Our source code is published in the accompanying web page, together with a user interface to examine and correct alignment results.

1. INTRODUCTION

To computationally analyse music performances or to construct performance databases, it is needed to match notes in a music performance signal (called an *aligned signal*) to those in a reference musical score or another performance signal (*reference signal*). This process is called music alignment and automating it is a fundamental technique for music information processing and has been a popular field of research [1–16]. This study deals with offline symbolic music alignment, with particular focus on piano performances. Both score-to-MIDI alignment and MIDI-to-MIDI alignment are considered in this paper. We consider Western classical music or similar music styles where musical scores exist behind the performances.

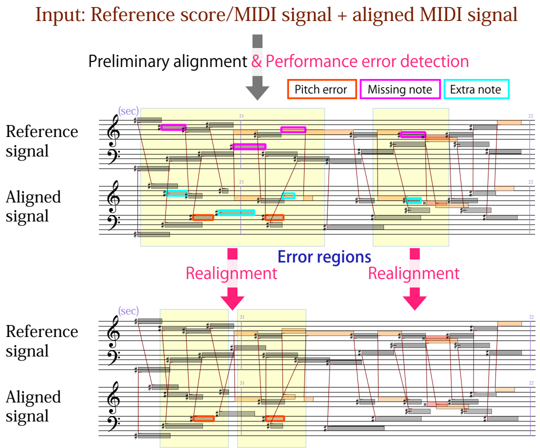

Figure 1. An outcome of the proposed method. Errors in preliminary alignment caused by reordered note pairs in the aligned signal are corrected by the realignment method.

Since music alignment between two identical performances is trivial, the central issue of automatic music alignment is to handle deviations in music performances. Possible deviations include tempo changes, performance errors (e.g. pitch errors, note insertions and deletions), ornamentation, and global structural differences (repeats and skips). To find the optimal alignment, various extensions of sequence matching methods such as hidden Markov models (HMMs) [7–9] and dynamic time warping (DTW) [1,2,6] have been studied. In the method using HMMs [7], for example, an HMM is constructed for each reference signal, in which note insertions and deletions, repeats, and skips are described by transition probabilities, and pitch errors are described by output probabilities. The aligned signal is considered as an output sequence from the HMM and the most probable sequence of latent states is estimated with the Viterbi algorithm for alignment.

It has been found that, in the case of polyphonic piano performances, deviations in performances due to asynchronies between hands/voices require special treatments [4, 5, 7, 9]. Such asynchronies result in reordering of notes with different score times, which is the main cause of alignment errors for HMMs or DTWs that are not specially designed to handle them. Models with explicit voice structure have been proposed and proved effective to solve this [4, 5, 9]. However, prior voice information of the reference signal is needed for applying these methods, which imposes limitations on usability since voice information is not given in single-channel MIDI signals and in some score file formats. Moreover, these methods have high computational cost compared to standard HMMs or DTWs [5–7, 9]. As is empirically known, those reordered notes appear only occasionally and in most cases the standard alignment methods work as accurately as the refined methods using voice information. Thus, the high computational cost would be reduced if parts of aligned signals, for which special treatments are necessary, can be selected.

Because significant deviations in music performances can usually be interpreted as performance errors, alignment errors are often connected with performance errors. For example, a pair of extra and missing notes as in Fig. 1 typically appear as a result of alignment errors. Based on the authors' experience, displaying performance errors enables human annotators to easily find alignment errors and greatly improves the efficiency of examining and correcting automatic alignment results. Likewise, by detecting performance errors in a given result of automatic alignment, it would be possible to select limited regions in the aligned signal that may contain alignment errors.

Based on these observations, this study aims to develop an automatic post-processing method for correcting given symbolic music alignment results. We first develop a performance error detection algorithm that recognises pitch errors, extra notes, and missing notes in a given alignment result. *Error regions* are then defined as segments of aligned and reference signals around performance errors and we investigate how much alignment errors are contained in these regions with various sizes of the regions. Next we develop a post-processing *realignment* method that can handle hand/voice asynchrony based on a voice-structured model. Since both music alignment and recognition of performance errors involve searches for an optimal choice among possible candidate solutions, we formulate them based on statistical models whose parameters can be optimised from data. To construct a realignment method that does not require prior voice information, we combine the method using merged-output HMMs [9] with a voice (hand) separation method [17]. For concreteness, we use as a preliminary alignment method the one based on temporal HMMs [7]. The results of the proposed method are evaluated in comparison with the state-of-the-art methods.

The contributions of this study are as follows. First, our alignment method achieves the highest accuracy and its computational cost is much smaller than previous methods with comparable accuracies. The method works without prior voice information and can be applied for a wide class of performance and score data. The source code for our algorithms and a user interface to examine and correct the results is published in the accompanying web page [18]. To our knowledge, this is currently the only publicly available alignment tool of comparable accuracies. Second, this is the first paper that quantitatively investigates the relation between performance errors and alignment errors, which can be used generally to reduce high computational cost that is typically required in elaborated methods. Lastly, our alignment algorithm of the merged-output HMMs yields better sub-optimisation than a previous one [9].

1.1 Current State-of-the-Art Methods

The method by Gingras and McAdams [5] (*GM algorithm*), which takes into account the voice structure and timing information, is regarded as one of the most accurate methods for symbolic music alignment, with 99.978% of accuracy on their data. The method based on the temporal HMM by Nakamura et al. [7] (*NOSW algorithm*) also uses the timing information but not the voice structure. The method can handle arbitrary repeats and skips, but the accuracy was lower than the GM algorithm in a direct comparison. For online alignment, the method using merged-output HMMs for incorporating the voice structure had better accuracies than the temporal HMMs [9].

Recently, Chen et al. [6] reported a significant lower accuracy ($\leq 91.93\%$) for the GM algorithm on other data and proposed a method based on DTW (*CJL1 algorithm*) with a better accuracy ($\leq 98.51\%$)[1]. Another method (*CJL2 algorithm*) is proposed in the paper, which is less accurate but more efficient than the CJL1 algorithm. The CJL algorithms neither use voice information nor have a special architecture to utilise the voice structure.

2. PERFORMANCE ERROR DETECTION

2.1 Problem Statement

Both reference and aligned signals can be represented as a sequence of musical notes (called *reference notes* and *aligned notes*) with a pitch and an onset time described as physical or score time. For MIDI-to-MIDI alignment, the reference signal can be a performed MIDI signal that has (almost) continuous onset times. In this case, we cluster notes according to onset times to obtain a reference signal with quantised onset times, which enables us to discuss score-to-MIDI and MIDI-to-MIDI alignment in a unified way. Specifically, we put a threshold of 35 ms, which is known to well discriminate chordal notes [19], to form clusters of notes and then quantise onset times (e.g. in units of ms etc.). An *alignment result* is a sequence of labels that indicates for each aligned note the corresponding reference note. If there is no corresponding note (as is the case for extra notes), a distinguished label 'EXTRA' is given.

As performance errors we consider pitch errors, extra notes, and missing notes. Extra notes are aligned notes that are not matched to any of the reference notes and missing notes are reference notes that do not appear in the aligned signal. In this study we consider the *strict alignment*, for which each reference note can be matched to at most one aligned note[2]. For a strict alignment result, performance errors are automatically determined: aligned notes without corresponding reference notes are extra notes; aligned notes with pitches different from the corresponding reference notes have pitch errors; reference notes not appearing

[1] A different evaluation measure was used in Ref. [6] and these upper bounds have been derived as conservative limits.

[2] This condition must be relaxed and apply only locally if we allow global repeats and skips in the aligned signal. In addition, trills and tremolos are exceptions where multiple aligned notes correspond to each reference note. For simplicity and for the lack of space, we concentrate on the case without ornaments, repeats, and skips in this paper.

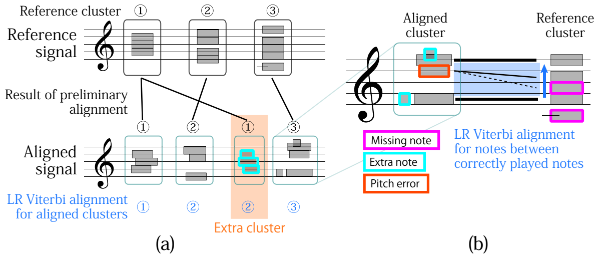

Figure 2. Steps for performance error detection. After (a) cluster-wise LR Viterbi alignment is performed, (b) note-wise LR Viterbi alignment is performed for each cluster.

in the alignment result are missing notes.

Standard alignment methods such as HMMs and DTWs often output alignment results that are not strict (so that a reference note can appear more than once) and performance error indications are not given. Therefore, the aim of performance error detection is to obtain a strict alignment result from a non-strict alignment result, which is equivalent to identifying extra notes in the latter one. In fact, our method uses only the information of matched score times for each note in an input alignment result.

2.2 Model

Our approach for the identification of extra notes is to carry out two left-to-right (LR) Viterbi alignments, first in units of 'chords' and second in units of notes within each 'chord' (Fig. 2). To be precise, we define a *reference cluster* as a set of all notes with the same score time in the reference signal. Aligned notes are clustered so that all successive notes form a cluster (*aligned cluster*) as long as their reference labels are in the same onset cluster. The first LR Viterbi alignment is then performed on the sequence of aligned clusters and those clusters assigned a reference cluster different from the original one are identified as extra clusters. Note that after this procedure each non-extra aligned cluster is matched to a unique reference cluster.

In the next step, extra notes in each (non-extra) aligned cluster are identified. Based on our intuition that aligned notes with correct pitches play a pivot role and the assigned reference labels should respect the pitch order, we first identify aligned notes with correct pitches and then match other notes, which are either extra notes or notes with pitch errors. Since in general there are multiple notes with the same pitch in one aligned cluster, the onset time information should be used here. As a reference point of onset time, the expected onset time $\tilde{t}$ of the reference onset cluster is computed by local averaging, similarly as tempo estimation [19]. If there are multiple candidates with the correct pitch, the one with an onset time nearest to $\tilde{t}$ is chosen and the other candidates are identified as extra notes.

Let $(q_1,\dots,q_C)$ denote an ordered set of notes in the concerned reference cluster, where q_c is the integral pitch of the c-th note and satisfies $q_1 \le q_2 \le \dots \le q_C$, and let $Q^{\rm corr}$ denote the set of reference notes matched to aligned notes with correct pitches. Similarly, let us order notes in the concerned aligned cluster according to pitch first and then onset time. Denoting the pitch and onset time of the b-th note by p_b and t_b, we thus have for all $b \in \{1,\dots,B\}$ (B is the number of notes in the aligned cluster) $p_{b-1} \le p_b$ and $t_{b-1} \le t_b$ if $p_{b-1} = p_b$. Now suppose that a pair of pivot notes (c, c') $(c, c' \in \{1,\dots,C\})$ satisfies that $q_c, q_{c'} \in Q^{\rm corr}$, $q_c < q_{c'}$, and $q_j \notin Q^{\rm corr}$ for each q_j with $c < j < c'$. For such a pair we define $Q = \{q_j \,|\, c < j < c'\}$ and $S = \{b \in \{1,\dots,B\} \,|\, q_c < p_b < q_{c'}\}$. The next step is to match Q and S for each pair (c, c') of pivot notes. For aligned notes with pitches higher or lower than the highest or lowest pivot note, we can similarly define Q and S as half-bounded sets, and for the case with no pivot notes, we define $Q = \{1,\dots,C\}$ and $S = \{1,\dots,B\}$, and carry out the following procedure.

The matching is trivial when $\#Q \le 1$ and $\#S \le 1$. In other cases, multiple interpretations of pitch errors exist and some principle must be introduced to find the optimal choice (Fig. 2(b)). We solve this optimisation problem with a statistical performance model including temporal fluctuations and pitch errors, similar to the model in Ref. [7]. The mapping $z : Q \ni j \mapsto z_j \in S$ is optimised by LR Viterbi alignment with the following probability:

$$P(z_j = b \,|\, z_{j-1} = b') = \theta_{b'b}\, \psi^{\rm pitch}(p_b - q_b)\psi^{\rm time}(t_b - \tilde{t})$$

where θ is a $\#S \times \#S$ LR transition probability matrix,

$$\theta_{b'b} = \begin{cases} 1/\#\{l \in S \,|\, l > b'\}, & b' < b; \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

and $\psi^{\rm pitch}(\delta p)$ is the probability of pitch errors in δp semitones (given by Eq. (30) of Ref. [19]), and $\psi^{\rm time}(\delta t)$ is the probability of onset time fluctuation given as

$$\psi^{\rm time}(\delta t) = \mathsf{N}(\delta t; 0, \rho^2). \tag{2}$$

Here, $\mathsf{N}(\,\cdot\,; \mu, \Sigma)$ denotes a normal distribution with mean μ and variance Σ. The value of ρ is taken as 100 ms in our implementation. Aligned notes without matched reference notes are classified as extra notes.

2.3 Error Regions and Alignment Errors

Having identified the performance errors, we now define error regions in the aligned signal around them. To do this, we first calculate the *synchronised onset time* for each reference cluster by averaging onset times of corresponding aligned notes, or if there are no such notes, by interpolating/extrapolating neighbouring synchronised onset times. We consider, for each extra note n with onset time t_n, a time interval of the form $[t_n - \Delta, t_n + \Delta)$ with width Δ (called an extra note region) and construct the set $\mathcal{R}_{\rm e}$ of such time intervals for all extra notes. Likewise, the set of pitch error regions $\mathcal{R}_{\rm p}$ is constructed. The set of missing note regions $\mathcal{R}_{\rm m}$ is similarly constructed by using the synchronised onset times to define each time region. Finally, the error region $\mathcal{R}$ is constructed by combining elements in $\mathcal{R}_{\rm e}$, $\mathcal{R}_{\rm p}$, and $\mathcal{R}_{\rm m}$. If there are overlapping time regions, they are expanded/unified to one time region at this step (Fig. 3). Thus, $\mathcal{R}$ is a set $\{[t_r, t'_r)\}_{r=1}^{N_{\mathcal{R}}}$ of non-overlapping

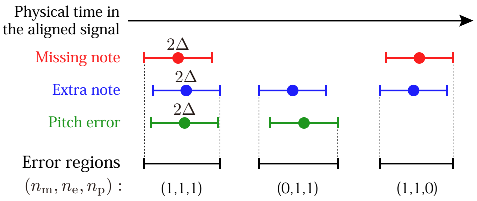

Figure 3. Examples of error regions and their indices.

time regions where we have $t'_{r-1} < t_r$ for all r. The number of missing notes, extra notes, and pitch errors in each region r is denoted by $n_{\rm m}(r)$, $n_{\rm e}(r)$, and $n_{\rm p}(r)$.

Let us discuss the relation between performance errors and alignment errors. An *alignment error* is defined as a reference label in an alignment result that is different from the ground-truth label. We say that an alignment error is contained in the error regions if the onset time of the incorrectly aligned note is contained in one of the regions in $\mathcal{R}$. The proportion of alignment errors contained in the error regions for varying time width Δ, calculated for alignment results of the temporal HMM on the three datasets explained in Sec. 4, is shown in Fig. 4, together with the proportion of aligned notes contained in the error regions. More than 90% of the performance errors are contained in the error regions for Δ as small as 0.1 s, while contained aligned notes remain less than 20% for $\Delta \leq 0.3$ s.

For the alignment errors to be corrected by realignment carried out on each error region, not only incorrectly aligned notes but also their corresponding reference notes must be contained in the region. To be precise, for each time region $[t_r, t'_r)$ in $\mathcal{R}$, we choose segments of the aligned and reference signals and use them as the aligned and reference signals for realignment. For the segment of the aligned signal, the subsequence of aligned notes whose onset times belong to the time region is used. For these aligned notes, we obtain the maximal and minimal score times ($\tau_{\rm max}$ and $\tau_{\rm min}$) of corresponding reference notes. The subsequence of all reference notes whose onset score times are in the range $[\tau_{\rm min}, \tau_{\rm max}]$ is used as the segment of the reference signal. We call an alignment error in an error region *correctable* if its ground-truth label is ‘EXTRA’ or is a reference note in the reference signal segment. We see in Fig. 4 that the proportion of correctable errors increases rapidly for $\Delta < 0.3$ s and gradually for $\Delta > 0.3$ s.

Although it is not always the case, naively we expect that the number of performance errors is reduced when alignment results are corrected, as is evident in the case of correcting a mismatched pair of missing and extra notes. On the other hand, if only one performance error exists or only missing notes exist in an error region, the number of performance errors cannot be reduced by realigning notes. By expanding this idea, we can impose conditions on error regions so that most of the alignment errors remain contained in the selected error regions but the contained alignment notes are reduced significantly. Results in Table 1, where error regions were imposed the condition of containing at least two types of performance errors, show an example of this fact. Such conditions can be used to increase the efficiency of realignment, as we see in Sec. 4.

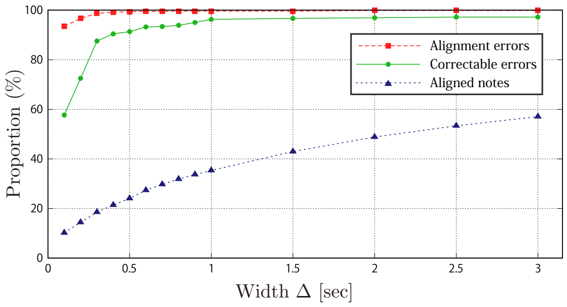

Figure 4. Proportion of alignment errors and aligned notes contained in the error regions.

Conditions	Alignment errors	Correctable errors	Aligned notes
None	98.7%	87.5%	18.5%
$n_{\rm m}n_{\rm e}+n_{\rm e}n_{\rm p}+n_{\rm p}n_{\rm m} > 0$	88.5%	80.3%	9.2%

Table 1. Same as Fig. 4 with and without imposed conditions on the error regions ($\Delta = 0.3$ s).

3. REALIGNMENT

Here we develop a realignment method based on merged-output HMMs, which is applied to the error regions to correct the preliminary alignment result. The overall procedure of realignment is illustrated in Fig. 5. We first apply hand separation for the reference signal segment to estimate the voice structure and then carry out alignment based on the merged-output HMM using the estimated voices.

3.1 Hand Separation

To formulate a method that does not require prior voice information, we apply voice separation to each reference signal segment. Because voice asynchrony in piano performances usually appears between the left- and right-hand parts and a larger number of voices increases the computational cost for realignment, we use a technique that separates a performance signal into two hand parts [17].

Voice information is described with a binary variable s_m for each note m in the reference signal segment. If $s_m = L$ (or R), the m-th note is in the left-hand (or right-hand) part. Let us denote the pitches of the reference signal segment by $\boldsymbol{x} = x_{1:M} = (x_1, \ldots, x_M)$, where the notes are ordered according to the onset score time. (Similar notations appear throughout the paper.) To estimate the sequence $\boldsymbol{s} = s_{1:M}$ from the input $\boldsymbol{x}$, we construct a merged-output HMM. The Markov model for each voice is described with transition probabilities on pitches, denoted by $\chi^L_{yy'}$ and $\chi^R_{yy'}$. Introducing pitch variables for the two voices, y^L_m and y^R_m for each m, the latent state variable for the merged-output HMM is given as $Y_m = (s_m, y^L_m, y^R_m)$ and the transition and output probabilities are given as

$$\begin{aligned} &P(Y_m = Y \,|\, Y_{m-1} = Y') \\ &\quad = \tfrac{1}{2}(\delta_{sL}\,\chi^L_{y'^L y^L}\,\delta_{y'^R y^R} + \delta_{sR}\,\chi^R_{y'^R y^R}\,\delta_{y'^L y^L}), \quad (3) \\ &P(x_m \,|\, Y_m = Y) = \delta_{sL}\delta_{y^L x_m} + \delta_{sR}\delta_{y^R x_m}, \quad (4) \end{aligned}$$

where $\delta_{yy'}$ denotes Kronecker’s delta. To complete the

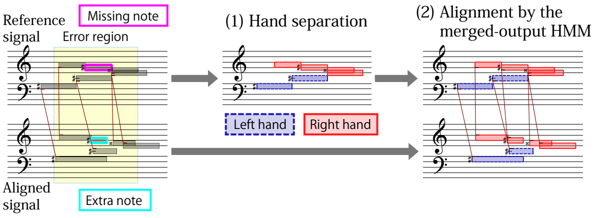

Figure 5. The realignment step consists of hand separation and alignment by the merged-output HMM.

stochastic process, we should specify the initial probability, which is given similarly as in Eq. (3) with initial pitch values denoted by y_0^L and y_0^R. We use as y_0^L and y_0^R the lowest and highest pitch in the reference signal segment. We can estimate $\boldsymbol{s}$ with the maximal posterior probability using an efficient Viterbi algorithm [17].

3.2 Realignment Based on Merged-Output HMM

For realignment we use the merged-output HMM proposed previously [9]. Modifications to the model are introduced to reduce computational cost and an inference algorithm that is more rigorous than the original one is derived.

Let us first briefly review the temporal HMM [7] for music alignment. The aligned signal (segment) can be can be described as a sequence $(\boldsymbol{p}, \boldsymbol{t})$, where $\boldsymbol{p} = (p_1, \dots, p_N)$ denotes the integral pitches and $\boldsymbol{t} = (t_1, \dots, t_N)$ denotes the onset times (N is the number of aligned notes). The reference signal (segment) is represented as a sequence of reference clusters indexed by $i \in \{1, \dots, I\}$ (I is the number of reference clusters), and the corresponding onset score time is denoted by τ_i. Local tempos are denoted by $\boldsymbol{v} = (v_1, \dots, v_N)$. The corresponding reference cluster of the n-th aligned note is denoted by $i_n \in \{1, \dots, I\}$. The latent state of the temporal HMM is indexed by (i_n, v_n) for each $n \in \{1, \dots, N\}$ and the output symbol is the pair (p_n, t_n). Transition and output probabilities are given as

$$\begin{aligned}
&P(i_n, v_n \,|\, i_{n-1}, v_{n-1}) = \pi(i_{n-1}, i_n)\mathsf{N}(v_n; v_{n-1}, \sigma_v^2),\\
&P(p_n|i_n) = \phi(i_n, p_n), &(5)\\
&P(t_n \,|\, t_{n-1}, i_{n-1} = i', i_n = i, v_n)\\
&\quad = (1 - \delta_{ii'})\mathsf{N}(t_n; t_{n-1} + v_n(\tau_i - \tau_{i'}); \sigma_t^2)\\
&\qquad + \delta_{ii'}\mathsf{Exp}(t_n - t_{n-1}; \lambda), &(6)
\end{aligned}$$

where we have assumed the statistical independence for the pairs i_n and v_n, and p_n and t_n. The probability π stochastically describes how the performance proceeds in the reference signal. The standard deviation σ_v represents the amount of tempo variation during the performance. The pitch output probability ϕ stochastically describes pitch errors (similarly as ψ^{pitch} in Sec. 2.2); it depends on the pitch context of reference cluster i. The form of the output probability for onset times reflects the fact that inter-onset intervals between chordal notes obey an exponential distribution (denoted by Exp) and those between onset clusters are approximately given as the product of the local tempo and the score time interval [7]. The scale parameter λ and the standard deviations σ_t and σ_v have been measured [7].

We can now construct the merged-output HMM for music alignment using voice information, by describing each voice by the temporal HMM and merging outputs from the two HMMs [9]. To reduce computational cost, we introduce two simplifications to the model. First, since the error region is considered to span a small time range (less than a few seconds), the variation of tempos should be relatively small. We therefore assume a constant tempo v in each error region, which can be obtained from the preliminary alignment result. This removes the dynamics of tempos and reduces the state space of the temporal HMM to that indexed only by i_n. Second, again because of the locality of error regions, we can assume LR transition probabilities for π. This reduces the number of possible state transition paths and thus reduces the computational cost. With these simplifications, the state space of the merged-output HMM is indexed by $k = (s, i^L, i^R, t^L, t^R)$ ($s \in \{L, R\}$) and the transition and output probabilities are

$$\begin{aligned}
&P(k_n = k \,|\, k_{n-1} = k')\\
&\quad = \tfrac{1}{2} A_s(i^s, t^s \,|\, i'^s, t'^s, v)\\
&\qquad \cdot \left[\delta_{sL}\delta_{i'^R i^R}\delta(t'^R - t^R) + \delta_{sR}\delta_{i'^L i^L}\delta(t'^L - t^L)\right],\\
&A_s(i^s, t^s \,|\, i'^s, t'^s; v) = \pi(i'^s, i^s)P(t'^s \,|\, t^s, i'^s, i^s, v), &(7)\\
&P(p_n \,|\, k_n = k) = \phi(i^s, p_n), &(8)\\
&P(t_n \,|\, k_n = k) = \delta(t_n - t^s), &(9)
\end{aligned}$$

where $\delta(\,\cdot\,)$ is the Dirac delta function. Notating $\boldsymbol{k} = k_{1:N}$, the complete-data probability is given as

$$P(\boldsymbol{k}, \boldsymbol{p}, \boldsymbol{t}) = \prod_{n=1}^{N} P(k_n|k_{n-1})P(p_n|k_n)P(t_n|k_n). \quad (10)$$

The alignment result is obtained by inferring $\boldsymbol{k}$ that maximises $P(\boldsymbol{k}, \boldsymbol{p}, \boldsymbol{t})$. The direct application of the Viterbi algorithm is impossible since the temporal HMM is of autoregressive type, i.e. the output probability of onset times depends on past values. Instead of the rough sub-optimisation method used in Ref. [9], we use a trick of introducing an auxiliary variable that encodes the historical path, as in Ref. [20], which enables almost exact optimisation. Introduce $h_n = 1, 2, \cdots$, which is defined as the smallest $h \geq 1$ satisfying $s_n \neq s_{n-h}$ for each n. We have

$$h_n = \begin{cases} h_{n-1}{+}1, & s_n = s_{n-1};\\ 1, & s_n \neq s_{n-1}, \end{cases} \quad t_n^s = \begin{cases} t_n, & s = s_n;\\ t_{n-h_n}, & s \neq s_n. \end{cases}$$

With a change of variables ($\boldsymbol{h} = h_{1:N}$, $\boldsymbol{i}^L = i^L_{1:N}$, etc.),

$$\begin{aligned}
&P(\boldsymbol{k}, \boldsymbol{p}, \boldsymbol{t}) = P(\boldsymbol{s}, \boldsymbol{h}, \boldsymbol{i}^L, \boldsymbol{i}^R, \boldsymbol{p}, \boldsymbol{t})\\
&= \prod_n \Bigg\{ \tfrac{1}{2}\left[\delta_{s_n L}\delta_{i'^R i^R} + \delta_{s_n R}\delta_{i'^L i^L}\right]\\
&\cdot \left[\delta_{s_n s_{n-1}}\delta_{h_n(h_{n-1}+1)}A_n^{\text{same}} + (1 - \delta_{s_n s_{n-1}})\delta_{h_n 1}A_n^{\text{diff}}\right]\Bigg\},\\
&A_n^{\text{same}} = A_{s_n}(i_n^{s_n}, t_n \,|\, i_{n-1}^{s_n}, t_{n-1}; v),\\
&A_n^{\text{diff}} = A_{s_n}(i_n^{s_n}, t_n \,|\, i_{n-1}^{s_n}, t_{\tilde{n}}; v) &(11)
\end{aligned}$$

where $\tilde{n} = n - h_{n-1} - 1$. It is now possible to derive the Viterbi algorithm for the state space of $(\boldsymbol{s}, \boldsymbol{h}, \boldsymbol{i}^L, \boldsymbol{i}^R)$. A

Algorithm	GM data	CJL data	Our data
Proposed	$\mathbf{0.18 \pm 0.08}$	$\mathbf{0.79 \pm 0.06}$	$\mathbf{0.48 \pm 0.03}$
NOSW+	1.46 ± 0.23	1.81 ± 0.08	0.64 ± 0.04
NOSW [7]	1.78 ± 0.25	2.33 ± 0.10	2.24 ± 0.07
GM [5]	0.28 ± 0.10 [7]	$8.07^{\dagger} \pm 0.18$ [6]	N/A
CJL1 [6]	N/A	$1.49^{\dagger} \pm 0.08$ [6]	N/A
CJL2 [6]	N/A	$2.20^{\dagger} \pm 0.09$ [6]	N/A

Table 2. Alignment error rates (%) with 1σ statistical errors. The best values within 1σ significance are displayed in bold font. Daggers indicate lower bounds (see Sec. 1.1).

cutoff (~ 50) on the maximum value of h_n can be put to reduce the search space with little loss of optimality [20]. Finally, the performance error detection described in Sec. 2 is performed separately on each voice (hand part).

During testing the method, we noticed that alignment errors as simple as a pair of missing and extra notes as in Fig. 1 sometimes remain after applying the described realignment step. This is often because the result of hand separation is not completely correct. To handle this, we carried out a simple processing step (called *pairing step*) of matching trivially corresponding missing and extra note pairs. For each missing note, an extra note with the same pitch is searched within the time region of half width Δ, and if found, they are matched. The pairing step can also be applied before the realignment step to correct trivial alignment errors and thus reduce the cost of realignment.

4. EVALUATION

As explained in Sec. 1.1, the state-of-the-art methods are the NOSW algorithm [7], the GM algorithm [5], and the CJL algorithms (CJL1 and CJL2) [6]. For comparison, we run the performance error detection on the results of the NOSW algorithm (*NOSW+ algorithm*), and the proposed realignment was applied to its results. Since the GM and CJL algorithms were not available from their authors but the used data were provided, we run the proposed method and temporal HMM on their data and directly compared the accuracies. The *GM data* consisted of seven performances of two excerpts of Chopin's piano pieces (total of 2,815 aligned notes). The *CJL data* consisted of 21 pairs of piano MIDI files (total of 25,656 aligned notes), most of which are synthetic (not human-played) performances. We also tested the proposed method on the human-played performance data that we prepared. Our data consisted of 60 excerpts of classical piano pieces each played by three different pianists (total of 43,608 aligned notes) [3] . For the GM data and our data it was score-to-MIDI alignment and for the CJL data it was MIDI-to-MIDI alignment. For the proposed method, $\Delta = 0.3$ s was used, error regions satisfying the condition ($n_{\rm m}n_{\rm e} + n_{\rm e}n_{\rm p} + n_{\rm p}n_{\rm m} > 0$) were selected for realignment, and the pairing step was applied before and after the realignment step.

The rates of alignment errors in Table 2 show that for all data the realignment method significantly improved the preliminary alignment results: in total 47% ($= 369$ aligned notes) of alignment errors made by the NOSW+ algorithm were reduced. The proposed method had the highest accuracies for all datasets. To evaluate computational efficiency, the processing time was measured. Our algorithms were implemented in C++ on a computer with 3.1 GHz CPU and 16 GB memory running Mac OS X 10.11. The measured time for the CJL data was 8.25 s for the NOSW algorithm, 17.76 s for the performance error detection, and 1.12 s for the realignment. Compared to the reported values [6], 342.70 s and 3535.36 s for the CJL1 and GM algorithm, the computational efficiency of the proposed method is evident, although direct comparison is not possible because of different computer environments. Examples demonstrating the effect of the realignment method are shown in the accompanying web page [18].

	GM data	CJL data	Our data	Time (s)
(a)	0.18	0.79	0.48	5.54 ± 0.07
(b)	0.25	1.17	0.51	6.31 ± 0.07
(c)	0.14	0.85	0.61	6.32 ± 0.05

Table 3. Alignment error rates (%) and processing time (averaged over five trials) for the proposed method with (a) both paring steps and conditions on error regions, (b) only conditions on error regions, and (c) only pairing steps.

To examine the effect of the paring step and the conditions imposed on error regions, the proposed method without these modifications was compared in terms of accuracies and processing time for all data (Table 3). In addition to the expected reduction of computation time, these modifications were also effective in reducing overall alignment errors. This suggests that the realignment by the merged-output HMM increases alignment errors in some error regions and these modifications have effects in avoiding this. Detailed analyses are currently being undertaken.

5. CONCLUSION

We have described a realignment method for symbolic music signals based on merged-output HMMs, which can deal with reordered notes due to voice asynchrony. To reduce the high computational cost, performance errors are detected and the merged-output HMMs are applied to regions around the performance errors rather than to the whole signal. In all tested data and for both score-to-MIDI and MIDI-to-MIDI alignment cases, the proposed realignment method combined with an HMM-based method achieved the highest accuracies, with short computation time.

The principle of using performance errors to select regions in the aligned signals that possibly contain alignment errors is generally applicable to save computation time. For example, when a further refined alignment method is found in the future, we can apply it to the error regions of the results by the proposed method, instead of doing alignment from scratch. In addition, since the realignment can be done locally, it can be applied to performance signals with global repeats and skips [7]. For future work, refinements for the model for performance error detection by examining human-annotated data would be possible to further improve the accuracy and efficiency.

[3] The data could be provided upon requests to the authors.

6. ACKNOWLEDGEMENTS

We are grateful to Bruno Gingras and Chun-Ta Chen for providing evaluation data. E. Nakamura is supported by the JSPS Postdoctoral Research Fellowship and this studied was done when he was a visiting researcher at the Queen Mary University of London. This work is in part supported by JSPS KAKENHI Nos. 24220006, 26280089, 26700020, 15K16054, 16H01744, 16H02917, 16K00501, and 16J05486 and JST ACCEL No. JPMJAC1602.

7. REFERENCES

[1] R. Dannenberg, "An On-Line Algorithm for Real-Time Accompaniment," *Proc. ICMC*, pp. 193–198, 1984.

[2] J. Bloch and R. Dannenberg, "Real-Time Computer Accompaniment of Keyboard Performances," *Proc. ICMC*, pp. 279–290, 1985.

[3] P. Desain, H. Honing, and H. Heijink, "Robust Score-Performance Matching: Taking Advantage of Structural Information," *Proc. ICMC*, pp. 337–340, 1997.

[4] 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*, vol. 32, no. 4, pp. 546–554, 2000.

[5] B. Gingras and S. McAdams, "Improved Score-Performance Matching Using Both Structural and Temporal Information from MIDI Recordings," *J. New Music Res.*, vol. 40, no. 1, pp. 43–57, 2011.

[6] C. Chen, J. R. Jang, and W. Liou, "Improved Score-Performance Alignment Algorithms on Polyphonic Music," *Proc. ICASSP*, pp. 1365–1369, 2014.

[7] 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.

[8] B. Pardo and W. Birmingham, "Modeling Form for On-line Following of Musical Performances," *Proc. NCAI*, 2005.

[9] 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.

[10] C. Raphael, "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models," *IEEE TPAMI*, vol. 21, no. 4, pp. 360–370, 1999.

[11] A. Cont, "A Coupled Duration-Focused Architecture for Realtime Music to Score Alignment," *IEEE TPAMI*, vol. 2, no. 6, pp. 974–987, 2010.

[12] C. Fremerey, M. Müller, and M. Clausen, "Handling Repeats and Jumps in Score-Performance Synchronization," *Proc. ISMIR*, pp. 243–248, 2010.

[13] C. Joder, S. Essid, and G. Richard, "A Conditional Random Field Framework for Robust and Scalable Audio-to-Score Matching, *IEEE TASLP*, vol. 19, no. 8, pp. 2385–2397, 2011.

[14] M. Grachten, M. Gasser, A. Arzt, and G. Widmer, "Automatic Alignment of Music Performances with Structural Differences," *Proc. ISMIR*, pp. 607–612, 2013.

[15] A. Maezawa, K. Itoyama, K. Yoshii, and H. G. Okuno, "Bayesian Audio Alignment Based on a Unified Model of Music Composition and Performance," *Proc. ISMIR*, pp. 233–238, 2014.

[16] W. Siying, S. Ewert, and S. Dixon, "Robust and Efficient Joint Alignment of Multiple Musical Performances," *IEEE/ACM TASLP*, vol. 24, no. 11, pp. 2132–2145, 2016.

[17] E. Nakamura, N. Ono, and S. Sagayama, "Merged-Output HMM for Piano Fingering of Both Hands," *Proc. ISMIR*, pp. 531–536, 2014.

[18] E. Nakamura, K. Yoshii, and H. Katayose, *Symbolic Music Alignment Tool*, 2017. `http://anonym9382.github.io/demo.html` [Online]

[19] 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.

[20] E. Nakamura, K. Yoshii, and S. Sagayama, "Rhythm Transcription of Polyphonic Piano Music Based on Merged-Output HMM for Multiple Voices," *IEEE/ACM TASLP*, vol. 25, no. 4, pp. 794–806, 2017.