COMPENSATING FOR ASYNCHRONIES BETWEEN MUSICAL VOICES IN SCORE-PERFORMANCE ALIGNMENT

Siying Wang *Sebastian Ewert* *Simon Dixon*

Queen Mary University of London, UK

ABSTRACT

The goal of score-performance synchronisation is to align a given musical score to an audio recording of a performance of the same piece. A major challenge in computing such alignments is to account for musical parameters including the local tempo or playing style. To increase the overall robustness, current methods assume that notes occurring simultaneously in the score are played concurrently in a performance. Musical voices such as the melody, however, are often played asynchronously to other voices, which can lead to significant local alignment errors. In this paper, we present a novel method that handles asynchronies between the melody and the accompaniment by treating the voices as separate timelines in a multi-dimensional variant of dynamic time warping (DTW). Constraining the alignment with information obtained via classical DTW, our method measurably improves the alignment accuracy for pieces with asynchronous voices and preserves the accuracy otherwise.

***Index Terms*—** score-audio alignment, multi-dimensional dynamic time warping, asynchrony, melody lead.

1. INTRODUCTION

Methods for the automatic alignment of different versions of a piece of music have a long history in music signal processing. In particular, the score-performance alignment problem has seen significant efforts in recent years. Applications include real-time score following [1–5] and automatic page-turning [6], musical expression analysis [7, 8], navigation in large music collections [9], informed audio editing and source separation [10]. In general, given a symbolic score representation (MIDI, MusicXML) and an audio recording of a performance of a piece of music, score-performance synchronisation methods aim at linking each note event in the score to its corresponding position in the recording.

A main difficulty in computing such alignments stems from the diversity of possible interpretations of a piece by a musician, i.e. not only the acoustic conditions can change considerably between recordings but also musical parameters including the playing style, expressive timing, or embellishments. To increase the overall robustness, state-of-the-art methods typically make simplifying assumptions about the problem, and in particular, that notes occurring simultaneously in the score are also played concurrently during a performance [11–13]. However, introducing asynchronies between simultaneous notes is considered an important part of musical expression. For example, emphasising a musical voice such as the melody by playing it earlier compared to other voices is a form of expression typically referred to as melody lead [14]. While such asynchronies usually do not have a strong effect on the alignment on a coarse level, the alignment accuracy on a finer, local level can drop measurably as the asynchrony is not expected by current methods.

S. Wang is funded by the China Scholarship Council (CSC). S. Ewert is funded by EPSRC Grant EP/J010375/1.

To cope with possible asynchronies between the melody and the accompaniment, the main idea in this paper is to separate the two voices in the score and to compute a joint three-dimensional alignment between the two score timelines and the audio timeline. While, in this basic form, the additional degree of freedom in the alignment can lead to measurable improvements in alignment accuracy on a fine level, it can also cause a loss of accuracy on a coarser, global level. Therefore, to exploit the overall robustness of existing alignment methods, we employ a state-of-the-art method to compute a coarser alignment in a first step, which is then used to constrain and guide the alignment in our proposed method. This way, our method not only combines the robustness of current methods with an improved alignment accuracy, but also drastically lowers the computational cost for computing a three-dimensional alignment (given the guiding alignment, from cubic to linear in the length of the recording or score).

The paper is organized as follows. Technical details of our method are described in Section 2. We report on some of our experiments in Section 3. Conclusions and discussions of future work are given in Section 4. Related work is discussed in the respective sections.

2. ALIGNMENT METHOD

A general procedure to synchronise a score and a performance can be summarized in three simple steps. First, the score and the audio are converted to a suitable, common feature representation. Second, by comparing each element in the score feature sequence with each element in the audio sequence using a distance measure, one obtains a distance or cost matrix. Third, based on such a matrix, a synchronisation method is applied to obtain a cost-minimizing alignment. In this context, various alignment methods have been proposed, including Dynamic Time Warping (DTW) [15], Hidden Markov Models (HMM) [16], Conditional Random Fields (CRF) [11], general graphical models [17], and Particle Filter / Monte-Carlo Sampling (MCS) based methods [3, 5]. While all these approaches typically yield robust alignments, none of them accounts for asynchronies between voices. An exception was presented in [18] but only for aligning MIDI files. Further, in [19], a greedy, post-processing method is introduced, which locally refines the alignment on a note level.

To model possible asynchronies between voices, we need to modify all three steps of the procedure above. First, the score can no longer be treated as a single data stream. Instead, the voices have to be isolated from the score and features have to be derived for each voice separately. Second, the comparison of features from all three sequences leads to a three-dimensional cost matrix (or cost tensor). Third, an extended alignment method is needed, which is able to deal with three sequences. Interpreting the alignment as a multi-dimensional data series synchronisation problem leads to two existing methods: the Asynchronous Hidden Markov Model (AHMM) [20] and the Multi-Dimensional Dynamic Time Warping

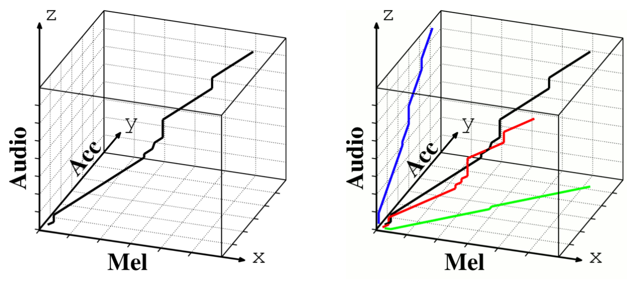

Fig. 1. A three-dimensional cost tensor. (a) Three-dimensional alignment path of the melody (Mel), accompaniment (Acc) and audio; (b) Projections of the path (black) onto x-z (red), y-z (blue) and x-y (green) planes.

(MD-DTW) [21]. These methods have been applied to various problems, including audio-visual speech recognition, and in particular for bi-modal speech and gesture fusion [22]. Both approaches share similar algorithmic roots (dynamic programming). In the following, we introduce our method as an extension to MD-DTW.

2.1. Computing Features for Individual Voices

While a musical score can often be separated into various voices, we focus in the following on the melody and accompaniment parts. From a musical point of view, these voices are particularly important for us as asynchrony between them has been reported and analysed in musicological studies [14]. Also from a numerical point of view this is beneficial, as only three timelines need to be aligned, which limits the computational complexity of the alignment problem.

We separate the melody and the accompaniment notes from the score using the skyline algorithm [23], which can be replaced by more complicated methods, such as the contig mapping [24], in future work. Once separated, the feature computation itself is essentially identical to previous methods. We compute the feature sequences $X := (x_1, x_2, \ldots, x_K)$ and $Y := (y_1, y_2, \ldots, y_K)$ for the two score voices as well as $Z := (z_1, z_2, \ldots, z_L)$ for the audio, with $x_n, y_m, z_\ell \in \mathcal{F}$ where $\mathcal{F}$ is a space containing two types of features similar to the approach described in [12]. The first type is a 88-dimensional log-frequency feature, whose entries encode a short-time intensity in spectral bands with centre frequencies corresponding to the 88 keys on a grand piano, see [25, 26] for information on how to derive such features from audio and MIDI representations. Additionally, we include a second 88-dimensional feature type which indicates possible onset positions separately for each key, see [12] for details. As shown previously [12], such a combination of features can lead to a substantial increase in alignment accuracy.

2.2. Three-Dimensional Dynamic Time Warping

In previous alignment approaches, each element of one feature sequence is compared with that of another sequence, which results in a cost matrix. With three feature sequences, we now extend this idea to a three dimensional cost tensor, see also Fig 1(a). More precisely, given the three feature sequences X, Y and Z, we define a $(K \times K \times L)$ cost tensor C by $C(n, m, \ell) := c(x_n + y_m, z_\ell)$, where $c : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_{\geq 0}$ denotes a local cost measure on $\mathcal{F}$. For $n \neq m$ we combine a melody and an accompaniment feature from different positions in the two score timelines into a single score feature, which is then compared to the audio feature. In this case, the difference $n - m$ encodes the asynchrony between the two voices. In particular, the *diagonal plane* in the cost tensor ($C(n, n, \ell)$ for $n \in [1:N]$ and $\ell \in [1:L]$) is essentially identical to a cost matrix between the complete score and the audio as used in classical two-dimensional DTW. All entries in the cost tensor on planes parallel to the diagonal plane have the same asynchrony between the two score voices (i.e. $n - m$ is constant), compare Fig. 2(a).

An alignment between X, Y and Z is defined as a sequence $p = (p_1, \ldots, p_Q)$ with $p_q = (n_q, m_q, \ell_q) \in [1:K] \times [1:K] \times [1:L]$ for $q \in [1:Q]$ satisfying $p_1 = (1, 1, 1)$ and $p_Q = (K, K, L)$ as well as $p_{q+1} - p_q \in \{(1,0,0), (0,1,0), (0,0,1), (1,1,0), (1,0,1), (0,1,1), (1,1,1)\}$ (step size condition). An alignment through the cost tensor aligning X, Y and Z is illustrated in Fig 1(a).

The cost of an alignment is defined as $\sum_{q=1}^{Q} C(n_q, m_q, \ell_q)$ and an alignment having minimal cost among all possible alignments is called an *optimal alignment*. To determine such an optimal alignment, one can employ MD-DTW [21]. In summary, one recursively computes a $(K \times K \times L)$-tensor D, where the entry $D(n, m, \ell)$ is the cost of an optimal alignment between $(x_1, \ldots, x_n)$, $(y_1, \ldots, y_m)$ and $(z_1, \ldots, z_\ell)$. Using dynamic programming, this tensor can be recursively computed as follows:

$$D(n,m,l) := \min \begin{cases} D(n-1,m,l) + w_1\, C(n,m,l), \\ D(n,m-1,l) + w_2\, C(n,m,l), \\ D(n,m,l-1) + w_3\, C(n,m,l), \\ D(n-1,m-1,l) + w_4\, C(n,m,l), \\ D(n-1,m,l-1) + w_5\, C(n,m,l), \\ D(n,m-1,l-1) + w_6\, C(n,m,l), \\ D(n-1,m-1,l-1) + w_7\, C(n,m,l), \end{cases}$$

for $n, m, l > 1$. Furthermore, $D(n,1,1) := \sum_{k=1}^{n} w_1\, C(k,1,1)$ for $n > 1$, $D(1,m,1) = \sum_{k=1}^{m} w_2\, C(1,k,1)$ for $m > 1$, $D(1,1,l) = \sum_{k=1}^{l} w_3\, C(1,1,k)$ for $l > 1$, and $D(1,1,1) := C(1,1,1)$. Calculations of entries on the x-y, x-z and y-z planes, i.e., $D(n,m,1)$, $D(n,1,l)$ and $D(1,m,l)$, are equivalent to the accumulated cost matrix calculation in classical two-dimensional DTW [21]. The weights $(w_1, w_2, w_3, w_4, w_5, w_6, w_7) \in \mathbb{R}_+^7$ can be set to adjust the preferences for the seven step sizes. Note that a bias for any direction is removed by setting these weights to $(w_1, w_2, w_3, w_4, w_5, w_6, w_7) = (1,1,1,2,2,2,3)$. An optimal alignment is obtained by tracing the minimising argument backwards from $D(K,K,L)$ to $D(1,1,1)$. Its projections onto the x-z and y-z planes yield alignments between the melody and the audio as well as the accompaniment and the audio, respectively. The projection onto the x-y plane corresponds to an alignment between the two score voices and thus encodes the estimated local asynchrony between them, see Fig. 1(b).

2.3. Path Constraints

In principle, an asynchronous alignment could be computed using MD-DTW as described above. In practice, however, there are additional factors which render this approach practically infeasible. On the one hand, the computational complexity of MD-DTW is considerable. Assuming the sequences to be aligned are roughly of the same length L, the memory and time complexity of an N-dimensional dynamic programming algorithm is $O(L^N)$ and $O(2^N L^N)$, respectively [27]. Since our application requires a high temporal resolution for the features, the value of L is typically high and the alignment becomes practically infeasible even for pieces of average length. On the other hand, splitting the score into two independent voices results in the number of notes in each voice to

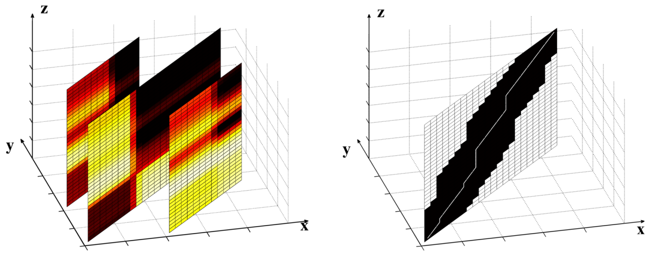

Fig. 2. Constraining the alignment. (a) Diagonal plane in the cost tensor corresponding to no asynchrony between voices, surrounded by two parallel planes corresponding to regions with constant asynchrony. (b) The alignment is contrained to run in a neighbourhood of a reference alignment, illustrated only on the diagonal plane.

be lower compared to the full score. This becomes a problem, if the remaining notes do no provide enough information to be discriminative in time. For example, if a chord is repeated consecutively in the accompaniment, an asynchronous alignment might easily confuse one instance of the chord for another, resulting in a substantial alignment error. We refer to this issue as the loss-of-structure problem in the following. Note that previous approaches will not suffer from this issue if the melody is discriminative enough.

In the following, we describe two extensions to MD-DTW, which aim at constraining the alignment in a meaningful way. As we will see, we can not only drastically lower the computational cost this way but also combine the robustness of previous approaches with an increased alignment accuracy resulting from our asynchronous alignment.

Asynchrony Constraints

With the first constraint we account for the observation that asynchronies between musical voices are in practice not arbitrarily high [14]. Musicians typically employ asynchronies to highlight certain elements in a piece, and if used in an extreme way, the asynchrony might render the piece unrecognisable by the audience. To limit the amount of asynchronies between voices to a musically meaningful range, we force the alignment to run closely to the diagonal plane in the cost tensor, compare Section 2.2. More precisely, in order to compute the diagonal plane features are combined without any asynchrony between them ($n = m$ as described above), and parallel planes use a non-zero but constant asynchrony $n - m$. To implement a constraint on the asynchrony, we only compute entries in C and D, for which $|n - m| <= A$ is satisfied, where $A \geq 0$ denotes the maximally allowed asynchrony. All entries not satisfying this constraint are formally set to infinity. Fig. 2(a) shows the diagonal plane as well as two parallel planes which satisfy the boundary case $|n-m| = A$ for a given value of A. Note that, since A is a fixed parameter, the number of parallel planes is fixed and the computational complexity is lowered from $O(L^3)$ to $O(L^2)$. The second constraint to be described next lowers the complexity even further.

Reference Alignment Constraints

While the main purpose of the simple asynchrony constraint just introduced is to lower the computational complexity, it also affects the loss-of-structure issue as certain degenerate alignments are automatically eliminated. However, depending on the value of A, our asynchronous alignment might still be less robust than previous approaches. Additionally, the computational costs are still about $2 \cdot A$ times higher than classical DTW. Therefore, we now introduce a second constraint, which guides the alignment using a reference alignment computed using a method based on classical DTW [12]. More precisely, given a 2D reference alignment $\tilde{p} = (\tilde{p}_1, \ldots, \tilde{p}_R)$ with $\tilde{p}_r = (n_r, \ell_r)$, we compute only the entry (n, m, ℓ) in C and D if there is a $\tilde{p}_r$ with $\ell_r = \ell$ and $|n_r - n| < B$, where $B > 0$ is the size of the constraint region. This way, we essentially project the reference alignment into the 3D cost tensor and use it there to define a neighbourhood which the alignment is forced to run in, see Fig 2(b) for an illustration. This approach resembles the general principle behind multiscale and FastDTW [28, 29], which are methods to accelerate classical DTW.

Overall, since B is fixed, the alignment in now restricted to a fixed size in a further dimension, which further reduces the computational complexity from $O(L^2)$ to $O(L)$ given the reference alignment. Our method to compute the reference alignment employs a multiscale version of DTW as well, which additionally lowers the computational cost of the entire system. Furthermore, by limiting both the allowed asynchrony and the displacement from a reference alignment effectively mitigates the loss-of-structure problem, which is demonstrated by our experiments to be discussed next.

3. EXPERIMENTS

3.1. Data Set

The experiments were conducted with recordings of three pieces which are known to contain strong asynchronies and three pieces played without asynchrony, to illustrate the performance of our proposed method in both cases. The former three pieces are Chopin Etude op. 10/3 (first 21 measures), Chopin Prelude op. 28/15 (first 27 measures) and Chopin Nocturne op. 48/1 (first 24 measures). The other three are picked from Bach's Well-Tempered Clavier, BWV 848, BWV 849 and BWV 889. The corresponding scores were obtained from the Mutopia project[1], the KernScores website[2] and the MuseScore website[3] as MIDI files.

For Chopin Etude op. 10/3, we used a data set consisting of 22 performances by skilled pianists recorded on a Bösendorfer computer-monitored piano, which includes both an audio recording as well as a corresponding MIDI version for each performance [14]. For the remaining five pieces, we downloaded MIDI versions of performances from the website of the Minnesota International Piano-e-Competition[4]. These MIDI files were recorded on Yamaha Disklavier Pro pianos during annual competitions for over ten years, which capture the detailed nuances of the performances. We generated audio versions from the MIDI files using Native Instruments's Vienna Concert Grand VST plugin comprising samples for a Boesendorfer 290 with an uncompressed size of almost 14 GB. The total number of performances for each piece is given in Table 1.

3.2. Evaluation Measure

To evaluate the accuracy of an alignment, we exploit that each audio recording is accompanied by a performance MIDI file, which annotates when and which notes are played. By manually aligning the performance MIDI with the corresponding score MIDI on a note level, we obtain a ground truth alignment between the audio and the score. Using the computed alignment, we can then locate for each note onset in the score the corresponding position in the audio. By

[1] http://www.mutopiaproject.org
[2] http://kern.ccarh.org
[3] http://musescore.org
[4] http://www.piano-e-competition.com/

	Piece	No. Rec	2D-DTW [12] Mel	2D-DTW [12] Acc	2D-DTW [12] OA	3D-DTW Mel	3D-DTW Acc	3D-DTW OA
w/ Asyn	Op. 10/3	22	16	23	21	16	18 (-22%)	17 (-19%)
	Op. 28/15	5	16	45	37	16	25 (-44%)	22 (-41%)
	Op. 48/1	4	27	64	49	25	56 (-13%)	44 (-10%)
	Average		18	31	27	17	24 (-23%)	22 (-19%)
w/o Asyn	BWV 848	3	11	14	12	11	14	12
	BWV 849	2	21	29	26	21	28	25
	BWV 889	2	11	15	13	11	17	14
	Average		14	19	16	14	19	16

Table 1. Experimental results for three pieces played with strong asynchrony (upper) and three pieces without asynchrony (lower). This table shows the number of performances available and statistics over the alignment error in milliseconds for the respective pieces. Both results for the 2D-DTW [12] and our 3D-DTW alignment method are computed separately for the melody (Mel) and accompaniment (Acc). The error values of these two voices are averaged over the number of notes to get the overall (OA) alignment error.

comparing the positions obtained from the computed and the ground truth alignment, we get an alignment error for each note. The error of an alignment is then the average over the differences for all notes, which we specify in milliseconds. To see the influence of our method in aligning each voice, we perform separate evaluations on melody and accompaniment notes and get an overall alignment error for a score-audio pair by averaging error values of the two voices over the number of notes.

3.3. Results

We compare the results of our method with a synchronization method based on classical 2D-DTW [12], which is also used to generate the 2D reference alignment for our 3D-DTW. To improve comparability, we use the same feature types and cost measures in the reference and our proposed method. In particular, we use a temporal resolution of 20 ms for the features. We set the weights for our 3D-DTW to $(w_1, w_2, w_3, w_4, w_5, w_6, w_7) = (1.5, 1.5, 1.5, 2.5, 2.5, 2.5, 3)$, and the weights for the 2D-DTW to $(w_1, w_2, w_3) = (2, 1.5, 1.5)$. The maximally allowed asynchrony between the two voices is set to 15 time frames (300 ms). The size of the constraint region is set to 50 time frames (1 second) around the guiding path.

Experimental results are summarized in Table 1. Comparing the results for the 2D-DTW and our 3D-DTW alignment method for the three pieces with strong asynchrony, we see that our method mostly improves the alignment accuracy for the accompaniment part. For op. 10/3, the overall alignment error for the accompaniment is 22% lower using 3D-DTW (23ms down to 18ms) while the error for the melody remains the same on average. The decrease in alignment error for the accompaniment is even greater for op. 28/15, by 44% (45ms down to 25ms). For op. 48/1, our 3D-DTW method reduces the alignment error for the accompaniment by 13%, and slightly for the melody. A possible explanation for the improvement being limited to the accompaniment could be that the melody is often played louder than the rest to emphasize it. This way, the melody dominates the energy distribution in the features and, not being able to differentiate between the two voices, classical DTW thus tends to focus on the dominating voice. In contrast, the two score voices are treated as independent timelines in our 3D-DTW alignment method, which reduces the local alignment error for the accompaniment. Moreover, for the three pieces without asynchrony, the average alignment error

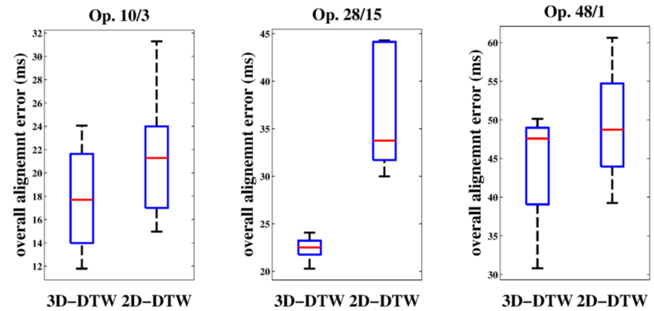

Fig. 3. Comparison of the 2D-DTW alignment results with our 3D-DTW alignment results. The boxplots illustrate the distribution of the alignment results in milliseconds for each piece separately.

of our proposed method is the same as that of the classical 2D-DTW alignment.

These results indicate that the improvement in alignment accuracy provided by our method depends on the characteristics of the music piece to be aligned and the amount of asynchrony played in the performances. That is exactly what we wanted to achieve by our proposed method, i.e., compensating for the asynchronies between two voices while preserving both the alignment accuracy of non-asynchronous parts and the overall alignment robustness.

The overall alignment error for the three pieces with strong asynchrony, drops from 27ms using 2D-DTW alignment to 22 ms using 3D-DTW alignment on average (decreases by 19%). This drop can also be seen from the boxplots[5] in Fig 3, which show the distribution of the alignment error for all score-audio pairs for the three pieces. Note that the above results were obtained by separating the melody and accompaniment notes from the score using the skyline algorithm. Compared with results obtained using a manual separation, the overall alignment error remained the same on average.

4. CONCLUSION AND FUTURE WORK

In this paper, we introduced a score-audio alignment method that can compensate for an asynchrony between the melody and accompaniment. A 3D-DTW algorithm was employed in which the two score voices are treated as independent timelines. Further, the alignment was constrained by a guiding alignment obtained via a classical 2D-DTW, providing improved robustness and a reduced computational complexity. Our experiments demonstrated that our proposed method can indeed improve the alignment accuracy for pieces with strong asynchrony and preserves the accuracy otherwise, compared to a previously proposed alignment method using classical DTW.

As a by-product, the resulting alignment can be used to indicate the positions where asynchrony occurs. In initial experiments, our method achieved a precision of 0.44 and recall of 0.58 on average in detecting positions with strong asynchrony. In the future, we plan to further investigate how to improve the performance of our method in this respect in order to develop an assistive tool for musical expression analysis. Furthermore, we will also apply multidimensional DTW to different asynchronous data stream alignment problems, such as the asynchrony between different instruments in a musical ensemble.

[5] We use standard boxplots: the red bar indicates the median, the blue box gives the $25th$ and $75th$ percentiles (p_{25} and p_{75}), the black bars correspond to the smallest data point greater than $p_{25} - 1.5(p_{75} - p_{25})$ and the largest data point less than $p_{75} + 1.5(p_{75} - p_{25})$. The red crosses are called outliers.

5. REFERENCES

[1] Roger B. Dannenberg and Christopher Raphael, "Music score alignment and computer accompaniment," *Communications of the ACM, Special Issue: Music Information Retrieval*, vol. 49, no. 8, pp. 38–43, 2006.

[2] Andreas Arzt, Sebastian Böck, Sebastian Flossmann, Harald Frostel, Martin Gasser, and Gerhard Widmer, "The complete classical music companion v0.9," in *Proceedings of the AES International Conference on Semantic Audio*, London, UK, 18–20 2014, pp. 133–137.

[3] Nicola Montecchio and Arshia Cont, "A unified approach to real time audio-to-score and audio-to-audio alignment using sequential Montecarlo inference techniques," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011, pp. 193–196.

[4] Robert Macrae and Simon Dixon, "A guitar tablature score follower," in *ICME*, 2010, pp. 725–726.

[5] Zhiyao Duan and Bryan Pardo, "A state space model for online polyphonic audio-score alignment," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011, pp. 197–200.

[6] Andreas Arzt, Gerhard Widmer, and Simon Dixon, "Automatic page turning for musicians via real-time machine listening," in *Proceedings of the European Conference on Artificial Intelligence*, 2008, pp. 241–245.

[7] Gerhard Widmer, Simon Dixon, Werner Goebl, Elias Pampalk, and Asmir Tobudic, "In search of the Horowitz factor," *AI Magazine*, vol. 24, no. 3, pp. 111–130, 2003.

[8] Meinard Müller, Verena Konz, Andi Scharfstein, Sebastian Ewert, and Michael Clausen, "Towards automated extraction of tempo parameters from expressive music recordings," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, 2009, pp. 69–74.

[9] Meinard Müller, Michael Clausen, Verena Konz, Sebastian Ewert, and Christian Fremerey, "A multimodal way of experiencing and exploring music," *Interdisciplinary Science Reviews (ISR)*, vol. 35, no. 2, pp. 138–153, 2010.

[10] Sebastian Ewert, Bryan Pardo, Meinard Müller, and Mark D. Plumbley, "Score-informed source separation for musical audio recordings," *IEEE Signal Processing Magazine - Special Issue on Source Separation and Applications*, 2014.

[11] Cyril Joder, Slim Essid, and Gaël Richard, "A conditional random field framework for robust and scalable audio-to-score matching," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2385–2397, 2011.

[12] Sebastian Ewert, Meinard Müller, and Peter Grosche, "High resolution audio synchronization using chroma onset features," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 1869–1872.

[13] Simon Dixon and Gerhard Widmer, "MATCH: A music alignment tool chest," in *ISMIR*, London, GB, 2005, pp. 492–497.

[14] Werner Goebl, "Melody lead in piano performance: expressive device or artifact?," *The Journal of the Acoustical Society of America*, vol. 110, pp. 563–572, 2001.

[15] Ning Hu, Roger B. Dannenberg, and George Tzanetakis, "Polyphonic audio matching and alignment for music retrieval," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, 2003.

[16] Nicola Orio and Francois Déchelle, "Score following using spectral analysis and hidden markov models," in *Proceedings of the International Computer Music Conference*, 2001, pp. 125–129.

[17] Christopher Raphael, "A hybrid graphical model for aligning polyphonic audio with musical scores," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Barcelona, Spain, 2004, pp. 387–394.

[18] Henkjan Honing Hank Heijink, Peter Desain and Luke Windsor, "Make me a match: An evaluation of different approaches to scoreperformance matching," *Computer Music Journal*, vol. 24 (1), pp. 43–56, 2000.

[19] Bernhard Niedermayer, "Improving accuracy of polyphonic music-to-score alignment," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, 2009, pp. 585–590.

[20] Samy Bengio, "An asynchronous hidden markov model for audio-visual speech recognition," in *Advances in Neural Information Processing Systems*, 2002, pp. 1213–1220.

[21] G.A.ten Holt, M.J.T. Reinders, and E.A. Hendriks, "Multidimensional dynamic time warping for gesture recognition," in *The Advanced School for Computing and Imaging*, 2002, vol. 300.

[22] Martin Wöllmer, Marc Al-Hames, Florian Eyben, Björn Schuller, and Gerhard Rigoll, "A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams," *Neurocomputing*, vol. 73, pp. 366–380, 2009.

[23] Alexandra Uitdenbogerd and Justin Zobel, "Melodic matching techniques for large music databases," in *Proceedings of the ACM International Conference on Multimedia*, 1999, pp. 57–66.

[24] Elaine Chew and Xiaodan Wu, "Separating voices in polyphonic music: A contig mapping approach," in *CMMR*, 2005, pp. 1–20.

[25] Judith C. Brown and Miller S. Puckette, "An efficient algorithm for the calculation of a constant Q transform," *Journal of the Acoustic Society of America (JASA)*, vol. 92, pp. 2698–2698, 1992.

[26] Meinard Müller and Sebastian Ewert, "Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Miami, FL, USA, 2011, pp. 215–220.

[27] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison, *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, New York, USA, 1999.

[28] S. Salvador and P. Chan, "FastDTW: Toward accurate dynamic time warping in linear time and space," in *Proceedings of the KDD Workshop on Mining Temporal and Sequential Data*, 2004.

[29] Meinard Müller, Henning Mattes, and Frank Kurth, "An efficient multiscale approach to audio synchronization," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Victoria, Canada, 2006, pp. 192–197.