

## Robust Joint Alignment of Multiple Versions of a Piece of Music

Wang, S; Ewert, S; Dixon, S

arXiv <https://arxiv.org/abs/1604.08516>

For additional information about this publication click this link.

<http://qmro.qmul.ac.uk/xmlui/handle/123456789/12554>

Information about this research object was correct at the time of download; we occasionally make corrections to records, please therefore check the published record when citing. For more information contact [scholarlycommunications@qmul.ac.uk](mailto:scholarlycommunications@qmul.ac.uk)

# ROBUST JOINT ALIGNMENT OF MULTIPLE VERSIONS OF A PIECE OF MUSIC

Siying Wang

Sebastian Ewert

Simon Dixon

Queen Mary University of London, UK

{siying.wang, s.ewert, s.e.dixon}@qmul.ac.uk

## ABSTRACT

Large music content libraries often comprise multiple versions of a piece of music. To establish a link between different versions, automatic music alignment methods map each position in one version to a corresponding position in another version. Due to the leeway in interpreting a piece, any two versions can differ significantly, for example, in terms of local tempo, articulation, or playing style. For a given pair of versions, these differences can be significant such that even state-of-the-art methods fail to identify a correct alignment. In this paper, we present a novel method that increases the robustness for difficult to align cases. Instead of aligning only pairs of versions as done in previous methods, our method aligns multiple versions in a joint manner. This way, the alignment can be computed by comparing each version not only with one but with several versions, which stabilizes the comparison and leads to an increase in alignment robustness. Using recordings from the Mazurka Project, the alignment error for our proposed method was 14% lower on average compared to a state-of-the-art method, with significantly less outliers (standard deviation 53% lower).

## 1. INTRODUCTION

Recent years have seen significant efforts to create large, comprehensive music collections. Music content providers (e.g. Spotify, iTunes, Pandora) rely on their existence, while national libraries and charitable organizations create and curate them in order to provide access to cultural heritage. For a given piece of music, large collections often contain various related recordings (cover songs, different interpretations), videos (official clip, live concert) and musical scores (in different formats such as MIDI and MusicXML, covering several editions). To identify and link these different versions, various automatic alignment methods have been proposed in recent years. Such *synchronization methods* have been used to facilitate navigation in large collections [1], to implement score following in real-time [2–5], to compare different interpretations of

a piece [6], to identify cover songs [7] or to simplify complex audio processing tasks [8].

In general, the goal of music synchronization is, given a position in one version of a piece of music, to locate the corresponding position in another version. To compute a synchronization, existing methods align two versions of a piece at a time, even if several relevant versions are available. For example, in [9, 10] a score of a piece is automatically aligned to a corresponding audio recording, while in [11] two acoustic realizations are being synchronized. As shown previously, current methods yield in many cases alignments of high accuracy [9–11]. However, musicians can interpret a piece in diverse ways, which can lead to significant local differences in terms of articulation and note lengths, ornamental notes, or the relative loudness of notes (balance). If such differences are substantial, the alignment accuracy of state-of-the-art methods can drop significantly.

To increase alignment robustness for difficult cases, the main idea in this paper is to exploit the fact that multiple versions of a piece are often available and can be aligned in a joint way. This way, we can exploit the additional information that each version provides about how a certain position in a piece can be realized by a musician. As a consequence, while two given recordings might be rather different and hard to align, both of them might actually be more similar to a third recording and including such a recording within the alignment process can lead to an increase in overall robustness. To compute our joint synchronization, we modify a multiple sequence alignment method typically employed in biological signal processing and combine it with strategies developed in a musical context based on Multiscale-DTW (FastDTW) and chroma-based onset features for increased computational efficiency and synchronization accuracy. In the following, we describe technical details of this method in Section 2. Then, we report on some of our experiments in Section 3. Conclusions and prospects for future work are given in Section 4.

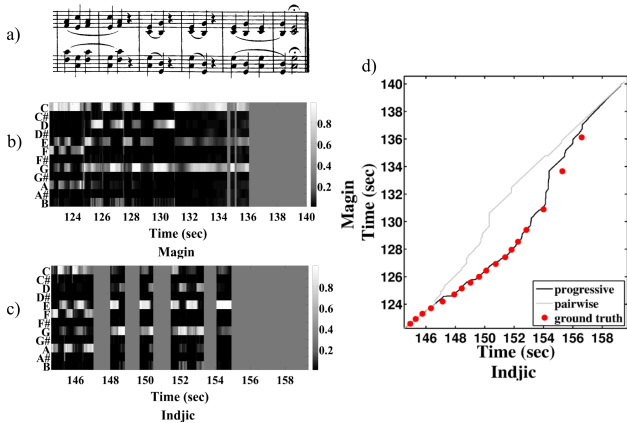
## 2. ALIGNMENT METHOD

Various methods have been proposed to align two given data sequences, including Dynamic Time Warping (DTW) and Hidden Markov Models (HMM) [2], Conditional Random Fields (CRF) [9], and Particle Filter / Monte-Carlo Sampling (MCS) based methods [4, 5]. With the exception of MCS methods, which are online methods, the remaining three methods operate in an offline fashion and are quite



© Siying Wang, Sebastian Ewert, Simon Dixon.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Siying Wang, Sebastian Ewert, Simon Dixon. “Robust Joint Alignment of Multiple Versions of a Piece of Music”, 15th International Society for Music Information Retrieval Conference, 2014.



**Figure 1.** Alignment of two interpretations of Chopin Op. 24 No. 2, measures 115-120: (a) Score for the six measures. (b)/(c) Chroma features for an interpretation by Magin and Indjic, respectively; chroma features with uniform energy distribution are the result of silence in the recording. (d) Alignment results for our baseline pairwise (gray) and proposed method (black).

similar from an algorithmic point of view. We describe our proposed method as an extension to DTW. However, the underlying ideas are applicable in HMM and CRF contexts as well.

## 2.1 Baseline Pairwise Alignment

To summarize DTW-based alignment, let  $X := (x_1, x_2, \dots, x_N)$  and  $Y := (y_1, y_2, \dots, y_M)$  be two feature sequences with  $x_n, y_m \in \mathcal{F}$ , where  $\mathcal{F}$  denotes a suitable feature space. Furthermore, let  $c : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}$  denote a local cost measure on  $\mathcal{F}$ . We define a resulting  $(N \times M)$  cost matrix  $C$  by  $C(n, m) := c(x_n, y_m)$ . An alignment between  $X$  and  $Y$  is defined as a sequence  $p = (p_1, \dots, p_L)$  with  $p_\ell = (n_\ell, m_\ell) \in [1 : N] \times [1 : M]$  for  $\ell \in [1 : L]$  satisfying  $1 = n_1 \leq n_2 \leq \dots \leq n_L = N$  and  $1 = m_1 \leq m_2 \leq \dots \leq m_L = M$  (boundary and monotonicity condition), as well as  $p_{\ell+1} - p_\ell \in \{(1, 0), (0, 1), (1, 1)\}$  (step size condition). An alignment  $p$  having minimal total cost among all possible alignments is called an *optimal alignment*. To determine such an optimal alignment, one recursively computes an  $(N \times M)$ -matrix  $D$ , where the matrix entry  $D(n, m)$  is the total cost of the optimal alignment between  $(x_1, \dots, x_n)$  and  $(y_1, \dots, y_m)$ :

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

for  $n, m > 1$ . Furthermore,  $D(n, 1) := \sum_{k=1}^n w_2 C(k, 1)$  for  $n > 1$ ,  $D(1, m) = \sum_{k=1}^m w_3 C(1, k)$  for  $m > 1$ , and  $D(1, 1) := C(1, 1)$ . The weights  $(w_1, w_2, w_3) \in \mathbb{R}_+^3$  can be used to adjust the preference over the three step sizes. By tracking the choice for the minimum starting from  $D(N, M)$  back to  $D(1, 1)$ , an optimal alignment can be derived in a straightforward way [2]. In a musical context,  $\mathcal{F}$  typically denotes the space of normalized chroma features,  $c$  is usually a cosine (or Euclidean) distance with

weights set to  $(w_1, w_2, w_3) = (2, 1, 1)$  to remove a bias for the diagonal direction [2, 11].

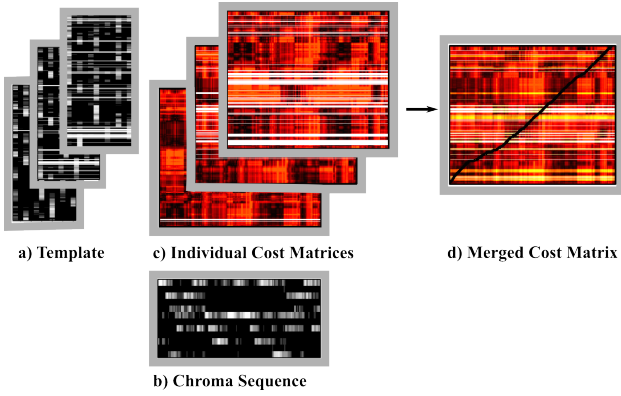
A main difficulty in aligning music stems from the degree of freedom a musician has in interpreting a score, in particular regarding the local tempo, balance (relative loudness of concurrent notes), articulation and playing style. If several differences occur together, standard alignment methods sometimes fail to identify the musically correct alignment. In Fig. 1(b)/(c), we see chroma features for two interpretations of Chopin Op. 24 No. 2 measures 115-120 (Fig. 1(a)) as performed by Magin and by Indjic, respectively. Besides the tempo, we see differences in the interpretation of pauses (the uniform energy distributions in the features correspond to silence), articulation and in the balance (relative loudness of notes). In this case, the differences are significant such that pairwise DTW-based approaches [10, 11] fail to compute the correct alignment, see upper path in Fig. 1(d). The red dots indicate corresponding beat positions in the two versions.

## 2.2 Joint Alignment of Multiple Versions

Comparing several versions of a piece, interpretations vary in different ways and to different extents. If several versions of a piece are available, each version provides an example of how a specific position in a piece can be realized, and this additional information can be used to stabilize the alignment for difficult sections. A straightforward strategy to compute a joint alignment could be to extend DTW to allow for more than two versions. For example, to align three versions, one can define an order-3 cost tensor in a straightforward way and apply the same dynamic programming techniques as used in DTW [12] (note that a cost matrix for two versions is an order-2 tensor). However, assuming that each feature sequence to be aligned is roughly of length  $N$ , the time and memory requirement to align  $K$  recordings would be in  $O(N^K)$ , which prohibits the alignment of more than a very few recordings.

In computational biology, multiple sequence alignment is a well-studied problem. Most popular are so called *profile-based methods* and *progressive alignment methods* [12]. Profile-based methods employ a specific type of HMM, which is trained via Expectation-Maximization (EM) on the set of feature sequences to be aligned. Each state of the resulting *profile-HMM* corresponds to a position in a so called average-sequence: the sequence of means of the observation probabilities of the HMM-states, see [12] for details. A multiple synchronization is then computed by aligning each sequence to the average-sequence via the Viterbi algorithm. This procedure has been attempted in a musical context with limited success [13]. We believe this is due to, using EM training, whereby aligned features are essentially averaged (with Gaussian observation probabilities), which results in a loss of information and can lead to a loss of alignment accuracy.

Using progressive alignment such averaging is not necessary. The underlying idea is to successively build a data structure referred to as a *template*, which provides efficient access to several aligned feature sequences, see Fig. 2(a).



**Figure 2.** Progressive alignment: Three aligned chroma sequences contained in the template (a) are compared to the chroma sequence (b). The resulting individual cost matrices (c) are merged into one (d), which is used to compute the alignment. The white lines in (a) and (c) indicate the positions of gap symbols.

By comparing a given feature sequence (Fig. 2(b)) to the sequences contained in the template, the alignment can be computed not only using one cost matrix (as in pairwise alignment) but several matrices in parallel - one for each sequence in the template (Fig. 2(c)). By suitably combining the information provided by each individual cost matrix, the influence of strong local differences on the alignment, that often only occur between specific pairs of versions, can be attenuated. As shown in Section 3, this can lead to a significant boost in alignment robustness.

To describe this procedure in more detail, we assume that we have  $K$  different versions of a piece and that their feature sequences are denoted by  $X^k = (x_1^k, \dots, x_{N_k}^k)$  for  $k \in [1 : K]$ . In each step of the progressive alignment, the template  $Z$  contains several of these feature sequences that have been stretched to have the same length. Initially,  $Z$  only consists of  $X^1$ . The remaining feature sequences are then successively aligned to  $Z$ , and after each alignment  $Z$  is updated by adding one more sequence. To this end, let  $\tilde{Z} = (\tilde{z}_1, \dots, \tilde{z}_{\tilde{L}})$  denote the current template which contains  $k - 1$  sequences of length  $\tilde{L}$  (i.e. each  $\tilde{z}_\ell$  contains  $k - 1$  features),  $X^k$  the sequence to be aligned, and  $p = (p_1, \dots, p_L) = ((n_1, m_1), \dots, (n_L, m_L))$  an alignment between  $\tilde{Z}$  and  $X^k$ . Intuitively, to add  $X^k$  to  $\tilde{Z}$ , we use  $p$  to stretch  $\tilde{Z}$  and  $X^k$  such that corresponding features are aligned and become part of the same element of  $Z$ . However, whenever features need to be copied to do the stretching (step sizes  $(1, 0)$  and  $(0, 1)$ ), we rather insert a special *gap* symbol instead of the features themselves. More precisely, let  $Z = (z_1, \dots, z_L)$  denote the updated template,  $z_n(k)$  denote the  $k$ -th feature in the  $n$ -th element of  $Z$ , and  $G$  denote the gap symbol<sup>1</sup>. Set  $z_1 = (\tilde{z}_1(1), \dots, \tilde{z}_1(k-1), x_1^k)$ , then for  $l = (2, 3, \dots, L)$ :

$$z_l = \begin{cases} (\tilde{z}_{n_\ell}(1), \dots, \tilde{z}_{n_\ell}(k-1), x_{m_\ell}^k), & p_\ell - p_{\ell-1} = (1, 1) \\ (\tilde{z}_{n_\ell}(1), \dots, \tilde{z}_{n_\ell}(k-1), G), & p_\ell - p_{\ell-1} = (1, 0) \\ (G, \dots, G, x_{m_\ell}^k), & p_\ell - p_{\ell-1} = (0, 1) \end{cases}$$

<sup>1</sup> Since chroma features contain only non-negative entries, the gap symbol can often be encoded as a pseudo-feature having negative entries.

The gap symbol and its influence will be further discussed in Section 3.

The alignment procedure itself is almost identical to standard DTW; only the local cost measure has to be adjusted to take the properties of the template into account. For a template  $Z$  comprising  $k - 1$  feature sequences and a feature sequence  $X$ , we define a template-aware cost function  $c_T : (\mathcal{F} \cup G)^{k-1} \times \mathcal{F} \rightarrow \mathbb{R}$  as

$$c_T(z_n, x_m) = \sum_{r=1}^{k-1} \begin{cases} c(z_n(r), x_m), & z_n(r) \neq G, \\ \mathcal{C}_G, & z_n(r) = G, \end{cases}$$

where  $\mathcal{C}_G > 0$  is a constant referred to as the *gap penalty*.

The influence a single additional recording can have using progressive alignment is illustrated in Fig. 1(d). Here, we included a third performance by Poblocka in the alignment, which could be considered as being “between” the two versions shown in Fig. 1 in terms of articulation style and balance. As we can see, the resulting path (black) follows the ground-truth markings (red dots) quite closely and improves significantly over the pairwise result.

### 2.3 Order of Alignments and Iterative Processing

The alignment of the first two versions in our progressive approach is equivalent to standard pairwise alignment. Errors in this first step influence to some degree all subsequent alignment steps. We discuss now two strategies that can help to increase the reliability of the first few alignments in our progressive approach. First, the order in which the alignments are computed is of importance, and we should start with recordings that are easy to align. In computational biology, a common approach to identify a reasonable order is referred to as the *guide tree approach* [12]. While there are various ways to implement such an approach, we consider the following procedure. First, for each pair of recordings, we compute the total cost of an optimal alignment between the pair to identify the pair having the lowest *average cost*, which is defined as the total cost of the alignment divided by its length  $L$ . We call the feature sequences for the recordings in this pair  $X^1$  and  $X^2$ . For the next recording, we identify the one being jointly closest to  $X^1$  and  $X^2$ . To this end, we sum for each of the remaining recordings the average cost of the alignments between the recording and  $X^1$ , and the recording and  $X^2$ . We call the feature sequence of the recording with the lowest sum  $X^3$ . We continue with this procedure until all recordings are in order. We refer to this strategy as *DTW-cost-based order*.

While this strategy leads to a useful order, its computational costs are significant. In our experiments, we found an alternative based on a much simpler strategy: We sorted the versions according to their length, starting with the shortest recordings. In the following, we refer to this strategy as *length-based order*. In Section 3, we compare both ordering strategies and discuss their behavior.

A second strategy to improve the reliability of the first alignments is referred to as *iterative progressive alignment*. The idea behind this strategy is, after all versions are aligned and included in the template, to remove one version from

ID	Piece	No. Rec.	No. Pairs
M17-4	Opus 17 No. 4	62	1891
M24-2	Opus 24 No. 2	62	1891
M30-2	Opus 30 No. 2	34	561
M63-3	Opus 63 No. 3	81	3240
M68-3	Opus 68 No. 3	49	1176

**Table 1.** Chopin Mazurkas and their identifiers used in our experiments. The last two columns indicate the number of performances available for the respective piece and the number of evaluated unique pairs.

the template and realign it, starting with the first version that was aligned. This way, errors made early in the progressive alignment can potentially be corrected. We implemented this extension as well and discuss it in Section 3.

## 2.4 Increasing the Computational Efficiency and Alignment Accuracy

Since progressive alignment shares its algorithmic roots with standard DTW, we can incorporate extensions that were successfully used with DTW-based methods. In particular, the methods described in [10, 14] employ a variant of DTW referred to as multiscale DTW (FastDTW) to increase the computational efficiency. The general idea is to recursively project an alignment computed at a coarse feature resolution level to a next higher resolution, and to refine the projected alignment on that resolution. This way, the matrix  $D$  only has to be evaluated around the projected path. This multiscale approach typically leads to a significant drop in runtime by up to a factor of 30, see [14].

Furthermore, the authors in [10] introduce a type of features that indicate onset positions separately for each chroma. These chroma-based onset features (DLNCO features) are then combined with normalized chroma features. As shown by the experiments in [10], these combined features can lead to a significant increase in alignment accuracy for pairwise methods. In the following, we employ the same features and cost measure as used in [10].

## 3. EXPERIMENTS

To illustrate the performance of our proposed method as well as the influence of certain parameters, we conducted a series of experiments using recordings taken from the Mazurka Project<sup>2</sup>, which compiled a database of over 2700 recorded performances by more than 130 distinct pianists for 49 Mazurkas composed by Frédéric Chopin. The recordings are dated between 1902 and today, and were made under strongly varying recording conditions. For our experiments, we employ a subset of five Mazurkas and 288 recordings, for which manually annotated beat positions are available, see Table 1. Performances with structural differences compared to the majority of recordings (such as additional repetitions of a part of a piece) were excluded from our experiments.

<sup>2</sup><http://www.mazurka.org.uk>

## 3.1 Evaluation Measure

To evaluate the accuracy of an alignment between two different versions of a piece, we employ the beat annotations as ground truth. To this end, we use the alignment to locate for each annotated beat position in the one version a corresponding position in the other version. Using the manual beat annotations for the other version, we can then compute the absolute difference between the correct beat position and the one obtained from the alignment. By averaging these differences for all beats, we obtain the *average beat deviation (ABD)* for a given alignment, which we measure in milli-seconds. For our evaluation, we compute this measure for each Mazurka and each pair of recordings. For example, for M17-4 our setup contains 62 recordings, which results in  $\binom{62}{2} = 1891$  unique pairs and corresponding average beat deviation values, see Table 1.

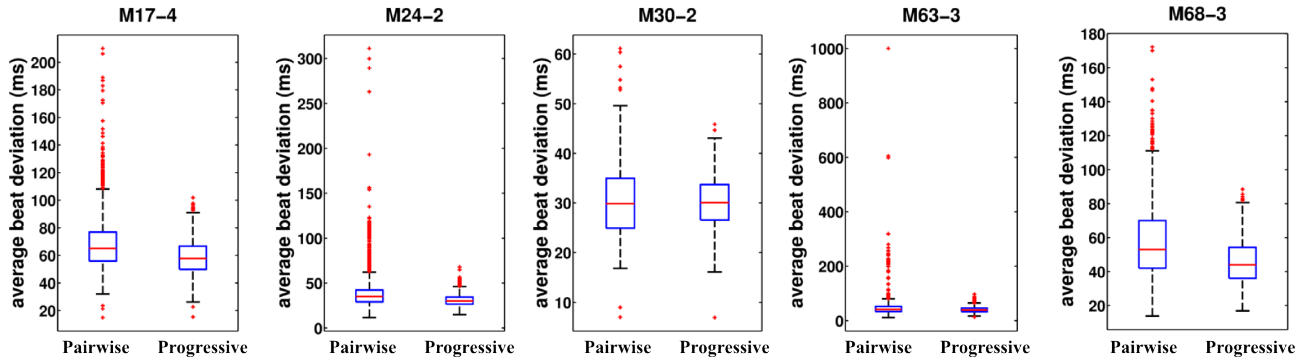
## 3.2 Pairwise vs Progressive Alignment

In a first experiment, we compare the alignment accuracy for pairwise and progressive alignment. Since the pairwise method described in [10] employs the same features and cost measure as our proposed progressive method, we use [10] as a baseline (other pairwise methods [11] showed a similar behavior). In particular, we use a temporal resolution of 20ms for both chroma and onset-indicator (DLNCO) features. The DTW weights are set to  $(w_1, w_2, w_3) = (2, 1.5, 1.5)$ . As proposed in [10], we use the cosine distance for the chroma features and the Euclidean distance for the DLNCO features. Moreover, for our proposed progressive alignment, we use the length-based alignment order and set the gap penalty parameter to the highest value the cost measure  $c$  can assume. The distribution of the average beat deviation (ABD) values for all pairs is summarized for each of the five Mazurkas separately in the boxplots<sup>3</sup> shown in Fig. 3, as well as in column A and B in Table 2.

Comparing the results for pairwise and progressive alignment, we can see that the mean ABD drops slightly using the progressive approach for most examples. For example, the mean ABD for M17-4 drops from 68ms using pairwise alignment to 59ms using our progressive method (decrease by 13%). On average, the mean ABD drops by 14%. More importantly though, the progressive alignment is significantly more stable. In particular, the inter-quartile range is smaller for all five Mazurkas using the progressive alignment (Fig. 3). Further, the number of alignments with a very high ABD is significantly reduced. This can be measured by the standard deviation (std), which for M17-4 using pairwise alignment is 19ms, while progressive alignment leads to an std of 12ms. This difference is even greater for other Mazurkas (M24-2 and M63-3). On average, the std is reduced by more than 50%. So overall, while our proposed procedure also led to an increase in

<sup>3</sup> 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  $p_{25} - 1.5(p_{75} - p_{25})$  and  $p_{75} + 1.5(p_{75} - p_{25})$ , and the red crosses are called outliers.





**Figure 3.** Comparison of the baseline pairwise alignment method with our proposed progressive alignment method. The boxplots illustrate the distribution of the average beat deviation values for each Mazurka separately.

M17-4	[A]	[B]	[C]	[D]	[E]	[F]	[G]
min	15	15	17	15	15	15	19
mean	68	59	68	63	76	80	91
max	210	102	118	116	789	129	252
std	19	12	13	13	94	13	22
M24-2							
min	12	15	17	12	15	16	11
mean	39	31	38	33	31	46	56
max	311	68	118	59	68	98	320
std	20	6	12	7	6	9	22
M30-2							
min	7	7	7	7	16	6	6
mean	30	30	31	29	31	40	43
max	61	46	49	53	46	64	80
std	8	5	6	6	5	7	9
M63-3							
min	11	13	15	12	13	14	9
mean	46	40	46	40	40	53	62
max	1000	97	99	99	97	109	1000
std	32	11	12	11	11	11	33
M68-3							
min	14	17	21	15	17	21	12
mean	58	46	57	53	46	71	86
max	172	89	144	105	89	179	335
std	23	13	18	15	13	21	34

**Table 2.** Statistics over the average beat deviation (ABD) values for the five Mazurkas and for 7 different alignment approaches (see text). [A]: Pairwise alignment. [B]: Proposed progressive alignment. [C]: Proposed without gap symbols. [D]: Proposed using DTW-cost-based alignment order. [E]: Proposed using iterative alignment. [F]: Proposed without DLNCO features. [G]: Pairwise without DLNCO features. All values in milli-seconds.

alignment accuracy on average, the main effect is a gain in robustness against strongly incorrect alignments.

### 3.3 Gap Penalties

In the next experiment, we investigate the influence of the gap penalty parameter by testing a slightly modified version of our proposed method. To this end, we modify the way the template is created by setting  $z_\ell = (\tilde{z}_{n_\ell}(1), \dots, \tilde{z}_{n_\ell}(k-1), x_{m_\ell}^k)$  for  $\ell \in [1:L]$ , i.e. we do not insert gap symbols but copy features as necessary to create the new template (comparing to Section 2.2). The results using this modification are shown in column C in Table 2. Comparing these values to our proposed method (column B) and the reference pairwise method (column A), we see

that this gap-less version typically improves over pairwise alignment in terms of maximum ABD values and the standard deviation, just as the proposed method. For example, for M17-4, the max ABD in column A is 210ms, while the max ABD in column C is 118ms. However, we do not observe a decrease in the mean ABD compared to pairwise alignment. For example, for M17-4, while using gaps the mean ABD drops from 68ms (column A) to 59ms (column B), it stays on a similar level in column C (68ms). The reason could be that by copying the features to create the template, some temporal precision is lost and this results in a minor loss of alignment accuracy.

### 3.4 Alignment Order

Next, we investigate the influence of the order in which we compute the progressive alignment, comparing the length-based and the DTW-cost-based strategy (see Section 2.3). The results are given in columns B and D of Table 2, respectively. As we can see, there are no significant differences between both strategies. For example, for M17-4, the mean ABD using the length-based strategy is 59ms (column B), while using the DTW-cost-based strategy the ABD slightly increases to 63ms. The other statistical values show a similar behavior. Since these results do not disclose any obvious advantages for the DTW-cost-based strategy, we therefore propose to simply use the length-based strategy. Interestingly, using the length-based strategy but starting with the longest recordings led to worse results.

Since (local) tempo differences can usually be handled quite well using DTW, it is not obvious why sorting by length yields a useful order. However, the fact that it does could indicate that there might be a correlation between the chosen tempo and other expressive parameters, such as articulation or balance, as strong differences in these parameters typically lead to difficulties for the alignment. Furthermore, the fact that according to our evaluation the shorter recordings were easier to align, could indicate that a high tempo could limit the range of possible realizations of expressive parameters in a performance. However, further studies would be necessary to confirm such theories.

### 3.5 Iterative Alignment

In a further experiment, we investigate whether iterative processing could further improve the alignment accuracy, compare Section 2.3. To this end, we use two iterations: the first iteration corresponds to progressive alignment, and in the second iteration, each version is removed from the template once and is then realigned. The results for this extension are given in column E of Table 2. Overall, the iterative variant led to a slight decrease in ABD in almost all examples, which is not even visible in Table 2 as we rounded all values. On the contrary, we observed a significant increase in ABD for M17-4 using the iterative variant. Here, the realignment led to a misalignment of several shorter recordings. Therefore, the results do not indicate any significant advantages of using iterative alignment.

### 3.6 Influence of Onset-Indicator Features

In a final experiment, we investigate the influence of the chroma-based onset-indicator (DLNCO) features [10] on the alignment accuracy when using progressive alignment. To this end, we disabled the DLNCO features in our proposed method, and computed the alignment only based on the normalized chroma features. The results of this experiment are given in column F in Table 2. As a further reference, we disabled the DLNCO features in our baseline pairwise method as well (column G).

As we can see, the minimum over the ABD values remains unaffected for most of the Mazurkas, which means that easy to align pairs can be aligned with chroma features alone just as well. For example, for M17-4, the minimum value in column F is identical to the one in column B. However, we see a significant increase in ABD in all other statistical values. For example, the mean ABD for M17-4 for our proposed method including DLNCO features is 59ms (column B), while disabling the DLNCO leads to a mean ABD of 80ms (column F). Similar observations can be made comparing the pairwise results. Overall, the results seem to indicate that including onset-indicator features indeed leads to a significant increase in alignment accuracy also for progressive alignments.

## 4. CONCLUSION

In this paper, we introduced a method for aligning multiple versions of a piece of music in a joint way. The availability of multiple versions to compare against during the alignment, stabilized the comparison for hard-to-align recordings and led to an overall increase in alignment accuracy and, in particular, in alignment robustness. Our experiments using real-world recordings from the Mazurka Project demonstrated that our proposed method can indeed be used to raise the alignment accuracy compared to previous methods that are limited to pairwise alignments. For the future, we plan to further investigate the behaviour of our procedure. In particular, we plan to analyze how other ordering strategies influence the alignment accuracy. We will also further explore different strategies to implement a cost for the gap symbol and to make it more adaptive.

**Acknowledgements:** This work was partly funded by the China Scholarship Council (CSC), EPSRC Grant EP/J010375/1, and the Queen Mary Postgraduate Research Fund (PGRF).

## 5. REFERENCES

- [1] M. Müller, M. Clausen, V. Konz, S. Ewert, and C. Fremerey, “A multimodal way of experiencing and exploring music,” *Interdisciplinary Science Reviews (ISR)*, vol. 35, no. 2, pp. 138–153, 2010.
- [2] R. B. Dannenberg and C. Raphael, “Music score alignment and computer accompaniment,” *Communications of the ACM, Special Issue: Music Information Retrieval*, vol. 49, no. 8, pp. 38–43, 2006.
- [3] A. Arzt, S. Böck, S. Flossmann, H. Frostel, M. Gasser, and G. 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.
- [4] N. Montecchio and A. Cont, “A unified approach to real time audio-to-score and audio-to-audio alignment using sequential Montecarlo inference techniques,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011, pp. 193–196.
- [5] Z. Duan and B. 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] G. Widmer, S. Dixon, W. Goebel, E. Pampalk, and A. Tobudic, “In search of the Horowitz factor,” *AI Magazine*, vol. 24, no. 3, pp. 111–130, 2003.
- [7] J. Serrà, E. Gómez, P. Herrera, and X. Serra, “Chroma binary similarity and local alignment applied to cover song identification,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, pp. 1138–1151, 2008.
- [8] S. Ewert, B. Pardo, M. Müller, and M. D. Plumbley, “Score-informed source separation for musical audio recordings: An overview,” *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 116–124, May 2014.
- [9] C. Joder, S. Essid, and G. 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.
- [10] S. Ewert, M. Müller, and P. 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.
- [11] S. Dixon and G. Widmer, “MATCH: A music alignment tool chest,” in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, London, GB, 2005, pp. 492–497.
- [12] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. New York, USA: Cambridge University Press, 1999.
- [13] H. I. Robertson, “Testing a new tool for alignment of musical recordings,” Master’s thesis, McGill University, 2013.
- [14] M. Müller, H. Mattes, and F. 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.