Towards Reliable Partial Music Alignments Using Multiple Synchronization Strategies

Sebastian Ewert[1], Meinard Müller[2], and Roger B. Dannenberg[3]

[1] Bonn University, Bonn, Germany,
Multimedia Signal Processing Group,
ewerts@iai.uni-bonn.de
[2] Saarland University and MPI Informatik,
Saarbrücken, Germany,
meinard@mpi-inf.mpg.de
[3] Carnegie Mellon University, Pittsburgh, USA,
School of Computer Science,
rbd@cs.cmu.edu

Abstract. The general goal of music synchronization is to align multiple information sources related to a given piece of music. This becomes a hard problem when the various representations to be aligned reveal significant differences not only in tempo, instrumentation, or dynamics but also in structure or polyphony. Because of the complexity and diversity of music data, one can not expect to find a universal synchronization algorithm that yields reasonable solutions in all situations. In this paper, we present a novel method that allows for automatically identifying the reliable parts of alignment results. Instead of relying on one single strategy, our idea is to combine several types of conceptually different synchronization strategies within an extensible framework, thus accounting for various musical aspects. Looking for consistencies and inconsistencies across the synchronization results, our method automatically classifies the alignments locally as reliable or critical. Considering only the reliable parts yields a high-precision partial alignment. Moreover, the identification of critical parts is also useful, as they often reveal musically interesting deviations between the versions to be aligned.

1 Introduction

As a result of massive digitization efforts, there is an increasing number of relevant digital documents for a single musical work comprising audio recordings, MIDI files, digitized sheet music, music videos, and various symbolic representations. In order to coordinate the multiple information sources related to a given musical work, various alignment and synchronization procedures have been proposed with the common goal to automatically link several types of music representations, [1–6, 8–10, 13–15, 18–24]. In a retrieval context, this linking information allows for an easy and intuitive formulation of a query. For example, in [13] the query is created by selecting multiple bars in a score representation. As the score is linked to an audio recording the query in the score domain can be

translated into a query in the audio domain, which can be used in an underlying audio retrieval system. This way, the user can make use of a semantically oriented high-level representation while the low-level representation needed only for technical reasons is hidden from the user.

In general terms, *music synchronization* denotes a procedure which, for a given position in one representation of a piece of music, determines the corresponding position within another representation. Even though recent synchronization algorithms can handle significant variations in tempo, dynamics, and instrumentation, most of them rely on the assumption that the two versions to be aligned correspond to each other with respect to their overall global temporal and polyphonic structure. In real-world scenarios, however, this assumption is often violated [11]. For example, for a popular song there often exists various structurally different album, radio, or extended versions. Live or cover versions may contain improvisations, additional solos, and other deviations from the original song [21]. Poor recording conditions, interfering screams and applause, or distorted instruments may introduce additional serious degradations in the audio recordings. On the other side, MIDI and other symbolic descriptions often convey only a simplistic view of a musical work, where, e. g., certain voices or drum patterns are missing. Furthermore, symbolic data as obtained from optical music recognition is often corrupted by recognition errors. In general, the synchronization of two strongly deviating representations of a piece of music constitutes an ill-posed problem. Here, without further model assumptions on the type of similarity, the synchronization task becomes infeasible.

In this paper, we address the problem of reliable partial music synchronization with the goal to automatically identify those passages within the given music representations that allow for a reliable alignment. Given two different representations of the same piece, the idea is to use several types of conceptually different synchronization strategies to obtain an entire family of temporal alignments. Now, consistencies over the various alignments indicate a high reliability in the encoded correspondences, whereas inconsistencies reveal problematic passages in the music representations to be aligned. Based on this automated local classification of the synchronization results, we segment the music representations into passages, which are then further classified as *reliable* and *critical*. Here, the reliable passages have a high confidence of being correctly aligned with a counterpart, whereas the critical passages are likely to contain variations and artifacts. The reliable passages can then be used as anchors for subsequent improvements and refinements of the overall synchronization result. Conversely, our automated validation is also useful in revealing the critical passages, which often contain the semantically interesting and surprising parts of a representation.

The remainder of this paper is organized as follows. In Sect. 2, we describe three conceptually different synchronization strategies. Then, in Sect. 3, we introduce our novel concept that allows for locally classifying the computed alignment results as reliable or critical. Finally, we report on our experiments in Sect. 4 and sketch some future work in Sect. 5. Further related work is discussed in the respective sections.

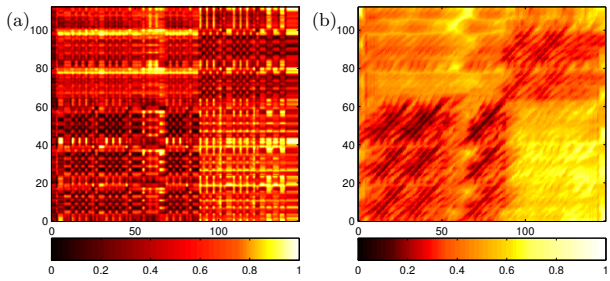

Fig. 1. Chroma based cost matrices for an audio recording (vertical axis) and a MIDI version (horizontal axis) of the song 'And I love her' by the Beatles. Times are given in seconds. **(a)** Cost Matrix **(b)** Smoothed Cost Matrix.

2 Music Synchronization Strategies

Most synchronization methods can be summarized in three simple steps. In a first step, the data streams to be aligned are converted to a suitable feature representation. Next, a local cost measure is used to compare features from the two streams. Based on this comparison, the actual alignment is then computed using an alignment strategy in a final step. In the following, we describe three conceptually different approaches for synchronizing a given MIDI-audio pair of a piece of music. Here, exemplarily using chroma features, we fix the parameters of the first two steps as described in Sect. 2.1 and focus on the third step, the alignment strategy (Sect. 2.2). As a first approach, we consider classical dynamic time warping (DTW), which allows for computing a global alignment path. We then introduce a recursive variant of the Smith-Waterman algorithm, which yields families of local path alignments. As a third approach, we use a partial matching strategy, which yields the least constrained alignment. While these three approaches share similar algorithmic roots (dynamic programming) they produce fundamentally different types of alignments. Intuitively, one may think of two extremes. On the one hand, DTW relies on strong model assumptions, but works reliably in the case that these assumptions are fulfilled. On the other hand, partial matching offers a high degree of flexibility, but may lead to alignments being locally misguided or split into many fragments. The Smith-Waterman approach can be thought of being in between these two extremes. As a complete description of the three alignment strategies would go beyond the scope of this paper, we summarize their properties while highlighting the conceptual differences among the approaches in Sect. 2.2. References to literature with exact implementation details are given as necessary.

2.1 Local Cost Measure

For comparing a MIDI file and an audio recording of the same song, we convert both representations into a common mid-level representation. Depending on the type of this representation the comparison can be based on musical properties

like harmony, rhythm or timbre. Since our focus is on alignment strategies, we exemplarily fix one type of representation and revert to chroma-based music features, which have turned out to be a powerful tool for relating harmony-based music, see [8, 10]. For details on how to derive chroma features from audio and MIDI files, we refer to the cited literature. In the subsequent discussion, we employ normalized 12-dimensional chroma features with a temporal resolution of 2 Hz (2 features per second).

Let $V := (v^1, v^2, \ldots, v^N)$ and $W := (w^1, w^2, \ldots, w^M)$ be two chroma feature sequences. To relate two chroma vectors we use the cosine distance defined by $c(v^n, w^m) = 1 - \langle v^n, w^m \rangle$ for normalized vectors. By comparing the features of the two sequences in a pairwise fashion, one obtains an $(N \times M)$-*cost matrix* C defined by $C(n, m) := c(v^n, w^m)$, see Fig. 1a. Each tuple (n, m) is called a *cell* of the matrix. To increase the robustness of the overall alignment procedure, we further enhance the structure of C by using a contextual similarity measure as described in [12]. The enhancement procedure can be thought of as a multiple filtering of C along various directions given by gradients in a neighborhood of the gradient $(1, 1)$. We denote the smoothed cost matrix again by C. For an example see Fig. 1b.

2.2 Alignment Methods

In the following, an alignment between the feature sequences $V := (v^1, v^2, \ldots, v^N)$ and $W := (w^1, w^2, \ldots, w^M)$ is regarded as a set $\mathcal{A} \subseteq [1 : N] \times [1 : M]$. Here, each cell $\gamma = (n, m) \in \mathcal{A}$ encodes a correspondence between the feature vectors v^n and w^m. Furthermore, by ordering its elements lexicographically $\mathcal{A}$ takes the form of a sequence, i. e., $\mathcal{A} = (\gamma_1, \ldots, \gamma_L)$ with $\gamma_\ell = (n_\ell, m_\ell)$, $\ell \in [1 : L]$. Additional constraints on the set ensure that only semantically meaningful alignments are permitted. We say that the set $\mathcal{A}$ is *monotonic* if

$$n_1 \leq n_2 \leq \ldots \leq n_L \text{ and } \; m_1 \leq m_2 \leq \ldots \leq m_L.$$

Similarly, we say that $\mathcal{A}$ is *strictly monotonic* if

$$n_1 < n_2 < \ldots < n_L \text{ and } \; m_1 < m_2 < \ldots < m_L.$$

Note that the monotonicity condition reflects the requirement of faithful timing: if an event in V precedes a second one this also should hold for the aligned events in W. A strictly monotonic set $\mathcal{A}$ will also be referred to as *match*, denoted by the symbol $\mathcal{M} = \mathcal{A}$. To ensure certain continuity conditions, we introduce step-size constraints by requiring

$$\gamma_{\ell+1} - \gamma_\ell \in \Sigma$$

for $\ell \in [1 : L - 1]$, in which Σ denotes a set of admissible step sizes. A typical choice is $\Sigma = \Sigma_1 := \{(1, 1), (1, 0), (0, 1)\}$ or $\Sigma = \Sigma_2 := \{(1, 1), (2, 1), (1, 2)\}$. Note that when using Σ_1 (Σ_2) the set $\mathcal{A}$ also becomes monotonic (strictly monotonic). A set $\mathcal{A}$ that fulfills the step-size condition is also referred to as *path* denoted by the symbol $\mathcal{P} = \mathcal{A}$. As final constraint, the boundary condition

$$\gamma_1 = (1, 1) \text{ and } \; \gamma_L = (N, M),$$

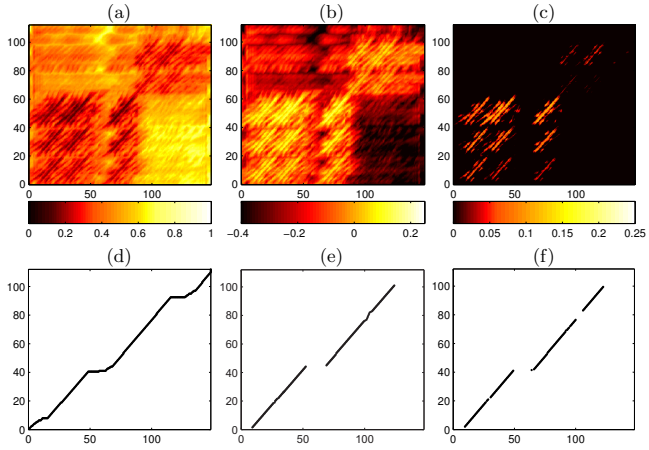

Fig. 2. Several techniques for the alignment of an audio recording (vertical axis) and a MIDI version (horizontal axis) of the song 'And I love her' by the Beatles. **(a)** Cost Matrix C. **(b)** Score Matrix S. **(c)** Thresholded Score Matrix $S_{\geq 0}$. **(d)** Optimal global path obtained via DTW based on matrix C. **(e)** Family of paths obtained via Smith-Waterman based on matrix S. **(f)** Optimal match obtained via partial matching based on matrix $S_{\geq 0}$.

ensures in combination with a step-size condition the alignment of V and W as a whole. If both the step-size as well as the boundary condition hold for a set $\mathcal{A}$, then $\mathcal{A}$ will be referred to as *global path* (or *warping path*) denoted by $\mathcal{G}$. Finally, a monotonic set $\mathcal{A}$ is referred to as *family of paths*, denoted by $\mathcal{F}$, if there exist paths $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_K$ with $\mathcal{F} = \mathcal{A} = \bigcup_{k \in [1:K]} \mathcal{P}_k$.

If it is known a-priori that the two sequences to be aligned correspond to each other globally then a global path is the correct alignment model. Here, dynamic time warping (DTW) is a standard technique for aligning two given sequences and can be used to compute a global path [17, 10]. In this context, the *cost* of an alignment $\mathcal{A}$ is defined as $\sum_{\ell=1}^{L} C(n_\ell, m_\ell)$. Then, after fixing a set of admissible step-sizes Σ, DTW yields an optimal global path having minimal cost among all possible global paths. In our experiments $\Sigma = \Sigma_1$ and $\Sigma = \Sigma_2$ yielded alignments of similar quality. However, here we choose $\Sigma = \Sigma_1$, since it leads to more reasonable results in cases where the assumption of global correspondence between the sequences is violated. For the subsequent discussion we use $\mathrm{A}(s,t)$ to refer to the segment in the audio recording starting at s seconds and terminating at t seconds. Similarly, $\mathrm{M}(s,t)$ refers to a MIDI segment. So listening to $\mathrm{M}(55,65)$ of the song 'And I love her' (used as example in Fig. 2) reveals a short bridge in the song. However in the audio recording (taken from the Anthology release) the bridge is skipped by the Beatles. Since DTW always aligns the sequences as a whole we find a semantically meaningless alignment between $\mathrm{A}(40,42)$ and $\mathrm{M}(48,65)$. A similar observation can be made at the beginning and the end of the optimal global path. Here, the intro and outro in the audio recording deviate strongly from those in the MIDI version.

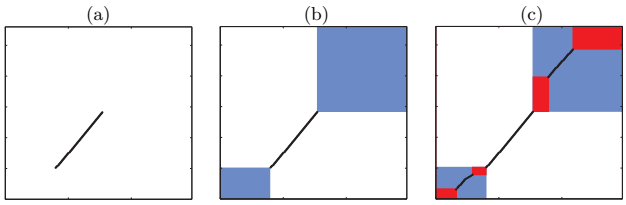

Fig. 3. First steps of our recursive Smith-Waterman variant. **(a)** Optimal path $\mathcal{P}$ derived via classical Smith-Waterman. **(b)** Submatrices defined via $\mathcal{P}$. **(c)** Result after the first recursion. Optimal paths have been derived from the submatrices and new submatrices (red) for the next recursive step are defined.

In general, using DTW in the case that elements in one sequence do not have suitable counterparts in the other sequence is problematic. Particularly, in the presence of structural differences between the two sequences, this typically leads to misguided alignments. Hence, if it is known a-priori that the two sequences to be aligned only partially correspond to each other, a path or a family of paths allows for a more flexible alignment than a global path.

Here, the Smith-Waterman algorithm is a well known technique from biological sequence analysis [16] to align two sequences that only locally correspond to each other. Instead of using the concept of a cost matrix with the goal to find a cost-minimizing alignment, we now use the concept of a *score matrix* with the goal to find a score-maximizing alignment. To this end, we fix a threshold $\tau > 0$. Then, a score matrix S is derived from C by setting $S = \tau - C$. Fig. 2b shows a score matrix derived from the cost matrix shown in Fig. 2a using the threshold $\tau = 0.25$. The *score* of an alignment $\mathcal{A}$ is defined as $\sum_{\ell=1}^{L} S(n_\ell, m_\ell)$. Then, after fixing a set of admissible step-sizes Σ, the Smith-Waterman algorithm computes an optimal path having maximal score among all possible paths using dynamic programming similar to DTW (see [16, 21]). Here, we found $\Sigma = \Sigma_2$ to deliver good alignment results.

We now introduce a recursive variant of the Smith-Waterman algorithm. In a first step, we derive an optimal path $\mathcal{P}$ as described above (see Fig. 3a). Then in a second step, we define two submatrices in the underlying score matrix S (see Fig. 3b). The first matrix is defined by the cell $(1, 1)$ and the starting cell of $\mathcal{P}$, and the second matrix by the ending cell of $\mathcal{P}$ and the cell (N, M). For these submatrices, we call the Smith-Waterman algorithm recursively to derive another optimal path for each submatrix (see Fig. 3c). These new paths define new submatrices on which Smith-Waterman is called again. This procedure is repeated until either the score of an optimal path or the size of a submatrix is below a given threshold. This results in a monotonic alignment set in form of a family of paths $\mathcal{F}$. Fig. 2e shows a family of two paths derived from the score matrix in Fig. 2b using our recursive Smith-Waterman variant. Here, the missing bridge in the audio as well as the different intros and outros in the audio and MIDI version are detected and, in this example, the recursive Smith-Waterman approach avoids a misalignment as in the DTW case (Fig. 2d).

This example illustrates another interesting property of the Smith-Waterman algorithm. Listening to A$(75, 83)$ and M$(99, 107)$ reveals a solo improvisation which is different in the audio and MIDI version, so they should not be aligned. Also, the corresponding area in the score matrix shows negative values. However, the Smith-Waterman algorithm aligns these two segments as part of the second path. The reason is that Smith-Waterman always tries to find the path with maximum score, and in this example the score for a longer path containing a few negative entries was higher than for a shorter path without negative entries. This property of the Smith-Waterman algorithm can be configured using *gap-penalty parameters*, see [21]. Essentially, these are used to define an additional weight for negative score entries. If the weight is high, then negative entries are emphasized and paths tend to be shorter and contain fewer entries with negative score. If the weight is low, paths tend to be longer and may contain longer sequences with negative score. We chose to use gap-penalty parameters being equivalent to a subtraction of 0.5 from all negative entries in the score matrix S. This is an empirically found value which worked best in our experiments.

However, even using gap-penalty parameters there is no control over the absolute length of sequences with negative score in an optimal path. If there is enough score to gain before and after a sequence with negative score, then this sequence will be bridged using Smith-Waterman and can become arbitrarily long in the resulting optimal path. So for the example depicted in Fig. 2e one could find a set of parameters to circumvent this misalignment, but in general these parameters would have to be set for each song individually. Here, a method referred to as *partial matching* allows for an even more flexible local alignment than Smith-Waterman (see [1, 10, 16]). The goal is to find a score-maximizing alignment, similar to the Smith-Waterman approach. But instead of using the Smith-Waterman score matrix S a thresholded version $S_{\geq 0}$ is used where every negative entry in S is replaced by zero (see Fig. 2c). The idea of this thresholding is to disallow the alignment of a pair of feature vectors if there is no score to gain from this alignment. Therefore, negative score can be ignored completely. In a sense, this concept is similar to finding the longest common subsequence (LCS) in string matching tasks. Based on this idea partial matching computes a score-maximizing optimal match. Again, this can be achieved efficiently using dynamic programming. See Fig. 2f for an example of an optimal match computed via partial matching based on the matrix shown in Fig. 2c. Here, the misalignment of the solo segments A$(75, 83)$ and M$(99, 107)$ as found in the Smith-Waterman case is not present. So partial matching, not enforcing any step-size or continuity conditions on the alignment, yields a more flexible alignment than the Smith-Waterman approach.

However, for other pieces, the lack of step-size constraints might lead to highly fragmented or even misguided alignments. As an example, we consider the song 'Lucy in the sky with diamonds' by the Beatles (see Fig. 4). Here, the optimal match computed via partial matching is highly fragmented (Fig. 4f). This is caused by local deviations of the audio recording (taken from the anthology release) from the MIDI version. For example, A$(133, 147)$ and M$(165, 183)$ are

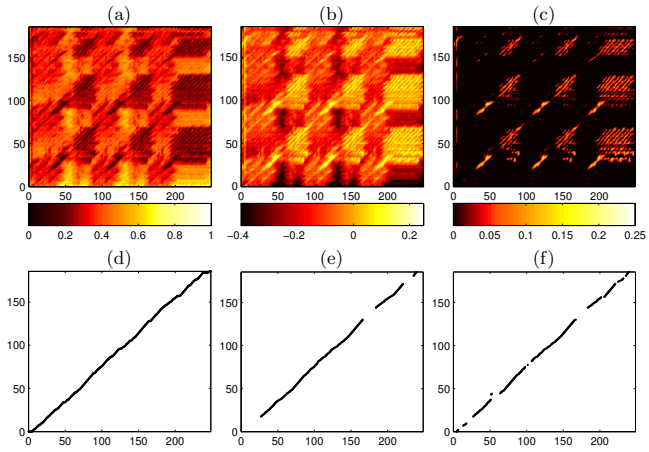

Fig. 4. Several techniques for the alignment of an audio recording (vertical axis) and a MIDI version (horizontal axis) of the Beatles song 'Lucy in the sky with diamonds', cf. Fig. 2.

semantically corresponding sections and should be aligned, but slightly differ in their arrangement. Here, the MIDI version features a very prominent bass line while in the audio recording the bass is only in the background. This leads to chroma representations for the audio and MIDI segment that differ strongly from each other thus explaining the low score in the corresponding area of the score matrix $S_{\geq 0}$, see Fig. 4c. Similar observations can be made comparing $\mathrm{A}(40, 47)$ and $\mathrm{M}(52, 62)$ as well as $\mathrm{A}(75, 82)$ and $\mathrm{M}(100, 110)$. However, the latter two pairs are correctly aligned by the recursive Smith-Waterman variant, see Fig. 4e. Here, a path is allowed to contain a few cells with negative score which helps to overcome local deviations in the feature representation. Nonetheless, using Smith-Waterman the segments $\mathrm{A}(133, 147)$ and $\mathrm{M}(165, 183)$ are still not aligned to each other. Here, the classical DTW approach, being forced to align the sequences as a whole, yields the best alignment result.

As illustrated by the examples shown in Figs. 2 and 4 each synchronization strategy sometimes yields reasonable and sometimes misguided and unsatisfying results. Therefore, without any definite a-priori knowledge about the input data none of the presented alignment methods can guarantee in general a reliable and musically meaningful alignment.

3 Proposed Method

In general, when the two sequences to be aligned correspond to each other in terms of their global temporal progression, the DTW procedure yields a robust alignment result. On the other hand, if structural differences are present, the more flexible Smith-Waterman approach or the even more flexible partial matching procedure may yield more reasonable alignments than DTW. Now, if

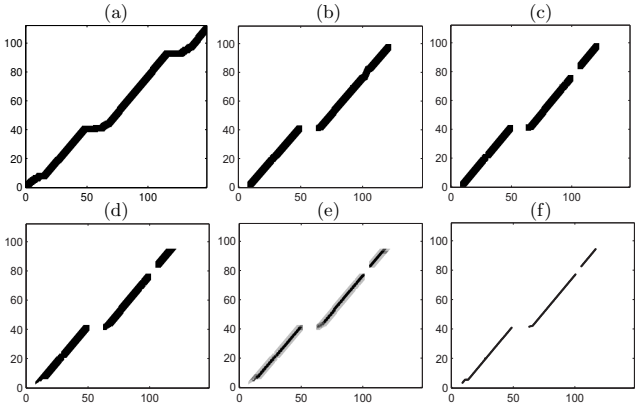

Fig. 5. Steps in our proposed method continuing the example shown in Fig. 2. **(a)-(c)** Augmented binary matrices for the optimal global path (DTW), family of paths (Smith-Waterman) and the optimal match (partial matching) using a tolerance neighborhood of 2 seconds. **(d)** Intersection matrix derived from (a)-(c). **(e)** Weighted intersection matrix. **(f)** Consistency alignment $\mathcal{C}$.

several strategies with different design goals yield similar alignment results, then there is a high probability of having semantically meaningful correspondences. Based on this simple idea, we present an automatic method towards finding passages in the MIDI and audio representations that can be synchronized in a reliable way. In contrast, this method also can be applied to identify critical passages, where the alignments disagree.

Given a MIDI-audio pair for a song, we start by computing an optimal global path using DTW, a family of paths using recursive Smith-Waterman, as well as an optimal match using partial matching. Next, we convert each alignment into a binary matrix having the same size as the cost matrix C. Here, a cell in the matrix is set to one if it is contained in the corresponding alignment and zero otherwise (see Figs. 2d-f). The next step is essentially a soft intersection of the three alignments. To this end, we augment the binary matrices by additionally setting every cell in the binary matrices to one if they are in a neighborhood of an alignment cell (see Figs. 5a-c). For Fig. 5, we used a large neighborhood of two seconds for illustrative purposes, while for the experiments in Sect. 4 we used a neighborhood of only one second. After that we derive an intersection matrix by setting each matrix cell to one that is one in all three augmented binary matrices (see Fig. 5d). The intersection matrix can be thought of as a rough indicator for areas in the cost matrix where the three alignment strategies agree with each other. However, this matrix does not encode an alignment that is constrained by any of the conditions described in Sect. 2.2. Therefore, to derive a final alignment result from this matrix, we first weight the remaining cells in the intersection matrix according to how often they are contained in one of the original three alignments (Fig. 5e). Then, interpreting the weighted matrix as a

score matrix, we use partial matching to compute an optimal match $\mathcal{C}$ referred to as *consistency alignment* (Fig. 5f).

In the following, we call a segment in the audio recording (the MIDI version) *reliable* if it is aligned via $\mathcal{C}$ to a segment in the MIDI version (in the audio recording). Similarly, we call a segment *critical* if it is not aligned. Here, $\mathrm{A}(3, 39)$, $\mathrm{A}(39, 76)$ and $\mathrm{A}(83, 95)$ as well as $\mathrm{M}(8, 45)$, $\mathrm{M}(63, 99)$ and $\mathrm{M}(106, 117)$ are examples of reliable segments in the audio recording and in the MIDI version, respectively. However, the automatic detection of critical sections can also be very useful as they often contain musically interesting deviations between two versions. For example, consider the critical segment $\mathrm{M}(45, 63)$. This segment contains the bridge found in the MIDI that was omitted in the audio recording as discussed in Sect. 2.2. Here, our method automatically revealed the inconsistencies between the MIDI version and the audio recording. Similarly, the differences between the audio and the MIDI version in the intro, outro and solo segments have also been detected. Here, not relying on a single alignment strategy leads to a more robust detection of critical segments than using just a single approach. The reasons why a segment is classified as critical can be manifold and might be an interesting subject for a subsequent musical analysis. In this context, our approach can be thought of as a supporting tool for such an analysis.

4 Experiments

In this section, we report on systematically conducted experiments to illustrate the potential of our method. To this end, we used twelve representative pieces from the classical, popular, and jazz collection of the RWC music database [7]. For each piece, RWC supplies high-quality MIDI-audio pairs that globally correspond to each other. Hence, using classical DTW allows us to synchronize each MIDI-audio pair to obtain an accurate alignment that can be used as ground-truth. The synchronization results were manually checked for errors. For the experiment, we strongly distorted and modified the MIDI versions as follows. Firstly, we temporally distorted each MIDI file by locally speeding up or slowing down the MIDI up to a random amount between $\pm 50\%$. In particular, we continuously changed the tempo within segments of 20 seconds of length with abrupt changes at segment boundaries to simulate ritardandi, accelerandi, fermata, and so on. Secondly, we structurally modified each MIDI file by replacing several MIDI segments (each having a length of 30 to 40 seconds) by sets of short 2 second snippets taken from random positions within the same MIDI file. In doing so, the length of each segment remained the same. These modified segments do not have any corresponding segments in the audio anymore. However, taken from the same piece, the snippets are likely to be harmonically related to the replaced content. Here, the idea is to simulate a kind of improvisation that fits into the harmonic context of the piece but is regarded as different between the audio and the MIDI version (similar to the differences found in $\mathrm{A}(75, 83)$ and $\mathrm{M}(99, 107)$, as discussed in Sect. 2.2). Finally, recall that we computed a ground-truth alignment via DTW between the original MIDI and the audio. Keeping

	P	R	F
DTW	**0.63**	0.96	0.76
rSW	**0.85**	0.85	0.85
PM	**0.80**	0.92	0.85
Con	**0.93**	0.85	0.88

	P	R	F
DTW	**0.51**	0.95	0.66
rSW	**0.83**	0.84	0.83
PM	**0.74**	0.92	0.82
Con	**0.95**	0.84	0.89

Table 1. Precision (P), Recall (R) and F-measure (F) for the alignment strategies DTW, recursive Smith-Waterman (rSW), Partial Matching (PM) and the consistency alignment (Con). **Left:** PR-values for modified MIDI-audio pairs. **Right:** PR-values for strongly modified MIDI-audio pairs.

track of the MIDI modifications, we derive a ground-truth alignment between the modified MIDI and the audio, in the following referred to as $\mathcal{A}^*$.

For each modified MIDI-audio pair, we compute the three alignments obtained by the three strategies described in Sect. 2.2 as well as the consistency alignment as described in Sect. 3. Let $\mathcal{A}$ be one of the computed alignments. To compare $\mathcal{A}$ with the ground-truth alignment $\mathcal{A}^*$, we introduce a quality measure that is based on precision and recall values, while allowing some deviation tolerance controlled by a given tolerance parameter $\varepsilon > 0$. The precision of $\mathcal{A}$ with respect to $\mathcal{A}^*$ is defined by

$$\mathrm{P}(\mathcal{A}) = \frac{|\{\gamma \in \mathcal{A} | \exists \gamma^* \in \mathcal{A}^* : ||\gamma - \gamma^*||_2 \leq \varepsilon\}|}{|\mathcal{A}|}$$

and the recall of $\mathcal{A}$ with respect to $\mathcal{A}^*$ is defined by

$$\mathrm{R}(\mathcal{A}) = \frac{|\{\gamma^* \in \mathcal{A}^* | \exists \gamma \in \mathcal{A} : ||\gamma - \gamma^*||_2 \leq \varepsilon\}|}{|\mathcal{A}^*|}.$$

Here, $||\gamma - \gamma^*||_2$ denotes the Euclidean norm between the elements $\gamma, \gamma^* \in [1 : N] \times [1 : M]$, see Sect. 2.2. Finally, the F-measure is defined by

$$\mathrm{F}(\mathcal{A}) := \frac{2\mathrm{P}(\mathcal{A})\mathrm{R}(\mathcal{A})}{\mathrm{P}(\mathcal{A}) + \mathrm{R}(\mathcal{A})}.$$

In our experiments, we used a threshold parameter ε corresponding to one second. The left part of Table 1 shows the PR-values averaged over all pieces for the four different alignment results. For example, when using DTW, the precision amounts to only $\mathrm{P} = 0.63$. The reason for this low value is that all time positions of the MIDI are to be aligned in the global DTW strategy, even if there is no semantically meaningful correspondence in the audio. When using Smith-Waterman or partial matching, the precision values become better. Note that the three alignments based on the three different strategies typically produce different (often random-like) correspondences in regions where the MIDI and audio differ. As a result, these correspondences are discarded in the consistency alignment yielding a high precision of $\mathrm{P} = 0.93$. This is exactly what we wanted

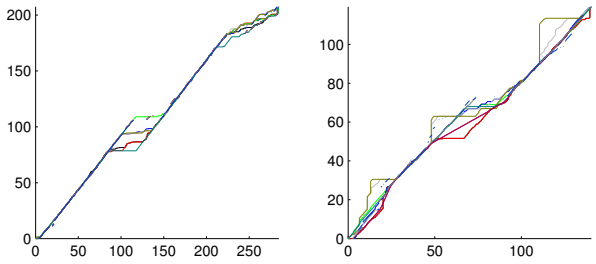

Fig. 6. Various alignments for two Beatles songs using conceptually different synchronization approaches. **Left:** 'While My Guitar Gently Weeps' **Right:** 'Norwegian Wood'

to achieve by our proposed method, where we only want to keep the reliable information, possibly at the cost of a lower recall.

In a second experiment, we modified the MIDI version even further by not only distorting and replacing randomly chosen MIDI segments as described above, but by inserting additional MIDI snippet segments. These additional structural modifications make the synchronization task even harder. The corresponding PR-values averaged over all pieces are shown in the right part of Table 1. As the task is harder now, the quality measures for all three alignment strategies drop except for our consistency alignment, where the precision (P = 0.95) essentially remains the same as in the first experiment.

5 Conclusions and Future Work

In this paper, we introduced a novel method for locally classifying alignments as reliable or critical. Here, our idea was to look for consistencies and inconsistencies across various alignments obtained from conceptually different synchronization strategies. Such a classification constitutes an essential step not only for improving current synchronization approaches but also for detecting artifacts and structural differences in the underlying music material. In the latter sense, our approach may be regarded as a supporting tool for musical analysis.

To cope with the richness and variety of music, we plan to incorporate many more competing strategies by not only using different alignment strategies but also by considering different feature representations, various feature resolutions, several local cost (similarity) measures, and different enhancement strategies for cost (score) matrices. Here, additional alignment strategies can be based on approaches algorithmically different from the ones presented here, like HMM-based methods as known from online score following and computer accompaniment [2, 18, 20], but also on approaches describes in Sect. 2 in combination with varying values for important parameters like τ or Σ. Fig. 6 shows first illustrating results for two Beatles songs, where we computed a large number of alignments using many different synchronization approaches. Despite of significant differences in structure, timbre, and polyphony, the consistencies of the various alignments

reveal the reliable passages that can then serve as anchor for subsequent improvements and refinements of the overall synchronization result.

Acknowledgements: The first author is funded by the German Research Foundation (DFG CL 64/6-1). The second author is with the Cluster of Excellence on *Multimodal Computing and Interaction* at Saarland University. The third is partially supported by the National Science Foundation (0534370).

References

1. Vlora Arifi, Michael Clausen, Frank Kurth, and Meinard Müller. Synchronization of music data in score-, MIDI- and PCM-format. *Computing in Musicology*, 13, 2004.
2. Arshia Cont. Real time audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical hmms. In *Proc. IEEE ICASSP, Toulouse, France*, 2006.
3. Roger Dannenberg and Ning Hu. Polyphonic audio matching for score following and intelligent audio editors. In *Proc. ICMC, San Francisco, USA*, pages 27–34, 2003.
4. Roger Dannenberg and Christopher Raphael. Music score alignment and computer accompaniment. *Special Issue, Commun. ACM*, 49(8):39–43, 2006.
5. Simon Dixon and Gerhard Widmer. Match: A music alignment tool chest. In *Proc. ISMIR, London, GB*, 2005.
6. Hiromasa Fujihara, Masataka Goto, Jun Ogata, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Automatic synchronization between lyrics and music cd recordings based on viterbi alignment of segregated vocal signals. In *ISM*, pages 257–264, 2006.
7. Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. Rwc music database: Popular, classical and jazz music databases. In *ISMIR*, 2002.
8. Ning Hu, Roger Dannenberg, and George Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *Proc. IEEE WASPAA, New Paltz, NY*, October 2003.
9. F. Kurth, M. Müller, C. Fremerey, Y. Chang, and M. Clausen. Automated synchronization of scanned sheet music with audio recordings. In *Proc. ISMIR, Vienna, AT*, 2007.
10. Meinard Müller. *Information Retrieval for Music and Motion*. Springer, 2007.
11. Meinard Müller and Daniel Appelt. Path-constrained partial music synchronization. In *Proc. IEEE ICASSP, Las Vegas, USA*, 2008.
12. Meinard Müller and Frank Kurth. Enhancing similarity matrices for music audio analysis. In *Proc. IEEE ICASSP, Toulouse, France*, 2006.
13. Meinard Müller, Frank Kurth, David Damm, Christian Fremerey, and Michael Clausen. Lyrics-based audio retrieval and multimodal navigation in music collections. In *Proc. 11th European Conference on Digital Libraries (ECDL)*, 2007.
14. Meinard Müller, Frank Kurth, and Tido Röder. Towards an efficient algorithm for automatic score-to-audio synchronization. In *Proc. ISMIR, Barcelona, Spain*, 2004.
15. Meinard Müller, Hennig Mattes, and Frank Kurth. An efficient multiscale approach to audio synchronization. In *Proc. ISMIR, Victoria, Canada*, pages 192–197, 2006.

16. Pavel A. Pevzner. *Computational Molecular Biology: An Algorithmic Approach*. MIT Press, 2000.
17. Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series, 1993.
18. Christopher Raphael. Automatic segmentation of acoustic musical signals using hidden markov models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:360–370, 1998.
19. Christopher Raphael. A hybrid graphical model for aligning polyphonic audio with musical scores. In *Proc. ISMIR, Barcelona, Spain*, 2004.
20. Diemo Schwarz, Nicola Orio, and Norbert Schnell. Robust polyphonic midi score following with hidden markov models. In *International Computer Music Conference*. 2004.
21. Joan Serrà, Emilia Gómez, Perfecto Herrera, and Xavier Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech and Language Processing*, 16, 2008.
22. Ferréol Soulez, Xavier Rodet, and Diemo Schwarz. Improving polyphonic and poly-instrumental music to score alignment. In *Proc. ISMIR, Baltimore, USA*, 2003.
23. Robert J. Turetsky and Daniel P.W. Ellis. Force-Aligning MIDI Syntheses for Polyphonic Music Transcription Generation. In *Proc. ISMIR, Baltimore, USA*, 2003.
24. Y. Wang, M.-Y. Kan, T. L. Nwe, A. Shenoy, and J. Yin. LyricAlly: automatic synchronization of acoustic musical signals and textual lyrics. In *MULTIMEDIA '04: Proc. 12th annual ACM international conference on Multimedia*, pages 212–219, New York, NY, USA, 2004. ACM Press.