JOINT STRUCTURE ANALYSIS WITH APPLICATIONS TO MUSIC ANNOTATION AND SYNCHRONIZATION

Meinard Müller
Saarland University and MPI Informatik
Campus E1 4, 66123 Saarbrücken, Germany
meinard@mpi-inf.mpg.de

Sebastian Ewert
Bonn University, Computer Science III
Römerstr. 164, 53117 Bonn, Germany
ewerts@iai.uni-bonn.de

ABSTRACT

The general goal of music synchronization is to automatically align different versions and interpretations related to a given musical work. In computing such alignments, recent approaches assume that the versions to be aligned correspond to each other with respect to their overall global structure. However, in real-world scenarios, this assumption is often violated. For example, for a popular song there often exist various structurally different album, radio, or extended versions. Or, in classical music, different recordings of the same piece may exhibit omissions of repetitions or significant differences in parts such as solo cadenzas. In this paper, we introduce a novel approach for automatically detecting structural similarities and differences between two given versions of the same piece. The key idea is to perform a single structural analysis for both versions simultaneously instead of performing two separate analyses for each of the two versions. Such a joint structure analysis reveals the repetitions within and across the two versions. As a further contribution, we show how this information can be used for deriving musically meaningful partial alignments and annotations in the presence of structural variations.

1 INTRODUCTION

Modern digital music collections contain an increasing number of relevant digital documents for a single musical work comprising various audio recordings, MIDI files, or symbolic score representations. In order to coordinate the multiple information sources, various synchronization procedures have been proposed to automatically align musically corresponding events in different versions of a given musical work, see [1, 7, 8, 9, 14, 15] and the references therein. Most of these procedures rely on some variant of dynamic time warping (DTW) and assume a global correspondence of the two versions to be aligned. In real-world scenarios, however, different versions of the same piece may exhibit significant structural variations. For example, in the case of Western classical music, different recordings often exhibit omissions of repetitions (e. g., in sonatas and symphonies) or significant differences in parts such as solo cadenzas of concertos. Similarly, for a given popular, folk, or art song, there may be various recordings with a different number of stanzas. In particular for popular songs, there may exist structurally different album, radio, or extended versions as well as cover versions.

A basic idea to deal with structural differences in the synchronization context is to combine methods from music structure analysis and music alignment. In a first step, one may partition the two versions to be aligned into musically meaningful segments. Here, one can use methods from automated structure analysis [3, 5, 10, 12, 13] to derive similarity clusters that represent the repetitive structure of the two versions. In a second step, the two versions can then be compared on the segment level with the objective for matching musically corresponding passages. Finally, each pair of matched segments can be synchronized using global alignment strategies. In theory, this seems to be a straightforward approach. In practise, however, one has to deal with several problems due to the variability of the underlying data. In particular, the automated extraction of the repetitive structure constitutes a delicate task in case the repetitions reveal significant differences in tempo, dynamics, or instrumentation. Flaws in the structural analysis, however, may be aggravated in the subsequent segment-based matching step leading to strongly corrupted synchronization results.

The key idea of this paper is to perform a single, joint structure analysis for both versions to be aligned, which provides richer and more consistent structural data than in the case of two separate analyses. The resulting similarity clusters not only reveal the repetitions within and across the two versions, but also induce musically meaningful partial alignments between the two versions. In Sect. 2, we describe our procedure for a joint structure analysis. As a further contribution of this paper, we show how the joint structure can be used for deriving a musically meaningful partial alignment between two audio recordings with structural differences, see Sect. 3. Furthermore, as described in Sect. 4, our procedure can be applied for automatic annotation of a given audio recording by partially available MIDI data. In Sect. 5, we conclude with a discussion of open problems and

The research was funded by the German Research Foundation (DFG) and the Cluster of Excellence on Multimodal Computing and Interaction.

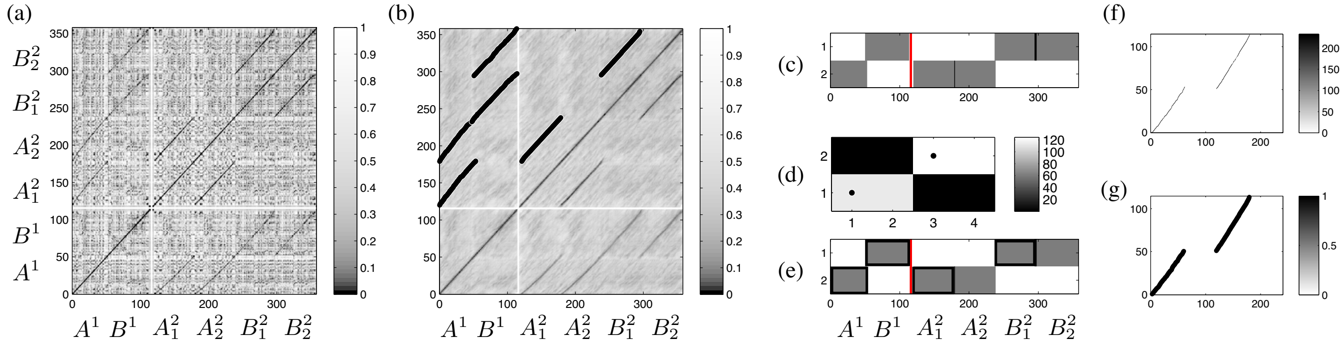

Figure 1. Joint structure analysis and partial synchronization for two structurally different versions of the Aria of the Goldberg Variations BWV 988 by J.S. Bach. The first version is played by G. Gould (musical form A^1B^1) and the second by M. Perahia (musical form $A_1^2A_2^2B_1^2B_2^2$). **(a)** Joint similarity matrix $\mathcal{S}$. **(b)** Enhanced matrix and extracted paths. **(c)** Similarity clusters. **(d)** Segment-based score matrix $\mathcal{M}$ and match (black dots). **(e)** Matched segments. **(f)** Matrix representation of matched segments. **(g)** Partial synchronization result.

prospects on future work.

The problem of automated partial music synchronization has been introduced in [11], where the idea is to use the concept of path-constrained similarity matrices to enforce musically meaningful partial alignments. Our approach carries this idea even further by using cluster-constraint similarity matrices, thus enforcing structurally meaning partial alignments. A discussion of further references is given in the subsequent sections.

2 JOINT STRUCTURE ANALYSIS

The objective of a joint structure analysis is to extract the repetitive structure within and across two different music representations referring to the same piece of music. Each of the two versions can be an audio recording, a MIDI version, or a MusicXML file. The basic idea of how to couple the structure analysis of two versions is very simple. First, one converts both versions into common feature representations and concatenates the resulting feature sequences to form a single long feature sequence. Then, one performs a common structure analysis based on the long concatenated feature sequence. To make this strategy work, however, one has to deal with various problems. First, note that basically all available procedures for automated structure analysis have a computational complexity that is at least quadratic in the input length. Therefore, efficiency issues become crucial when considering a single concatenated feature sequence. Second, note that two different versions of the same piece often reveal significant local and global tempo differences. Recent approaches to structure analysis such as [5, 12, 13], however, are built upon the constant tempo assumption and cannot be used for a joint structure analysis. Allowing also tempo variations between repeating segments makes the structure analysis problem a much harder problem [3, 10]. We now summarize the approach used in this paper closely following [10].

Given two music representations, we transform them into suitable feature sequences $U := (u^1, u^2, \ldots, u^L)$ and $V := (v^1, v^2, \ldots, v^M)$, respectively. To reduce different types of music data (audio, MIDI, MusicXML) to the same type of representation and to cope with musical variations in instrumentation and articulation, chroma-based features have turned out to be a powerful mid-level music representation [2, 3, 8]. In the subsequent discussion, we employ a smoothed normalized variant of chroma-based features (CENS features) with a temporal resolution of 1 Hz, see [8] for details. In this case, each 12-dimensional feature vector u^ℓ, $\ell \in [1 : L]$, and v^m, $m \in [1 : M]$, expresses the local energy of the audio (or MIDI) distribution in the 12 chroma classes. The feature sequences strongly correlate to the short-time harmonic content of the underlying music representations. We now define the sequence W of length $N := L + M$ by concatenating the sequences U and V:

$$W := (w^1, w^2, \ldots, w^N) := (u^1, \ldots, u^L, v^1, \ldots, v^M).$$

Fixing a suitable local similarity measure— here, we use the inner vector product—the $(N \times N)$-*joint similarity matrix* $\mathcal{S}$ is defined by $\mathcal{S}(i, j) := \langle w^i, w^j \rangle$, $i, j \in [1 : N]$. Each tuple (i, j) is called a *cell* of the matrix. A *path* is a sequence $p = (p_1, \ldots, p_K)$ with $p_k = (i_k, j_k) \in [1 : N]^2$, $k \in [1 : K]$, satisfying $1 \leq i_1 \leq i_2 \leq \ldots \leq i_K \leq N$ and $1 \leq j_1 \leq j_2 \leq \ldots \leq j_K \leq N$ (monotonicity condition) as well as $p_{k+1} - p_k \in \Sigma$, where Σ denotes a set of admissible step sizes. In the following, we use $\Sigma = \{(1, 1), (1, 2), (2, 1)\}$.

As an illustrative example, we consider two different audio recordings of the Aria of the Goldberg Variations BWV 988 by J.S. Bach, in the following referred to as *Bach example*. The first version with a duration of 115 seconds is played by Glen Gould without repetitions (corresponding to the musical form A^1B^1) and the second version with a duration of 241 seconds is played by Murray Perahia with repetitions (corresponding to the musical form $A_1^2A_2^2B_1^2B_2^2$). For the feature sequences hold $L = 115$, $M = 241$, and $N = 356$. The resulting joint similarity matrix is shown in

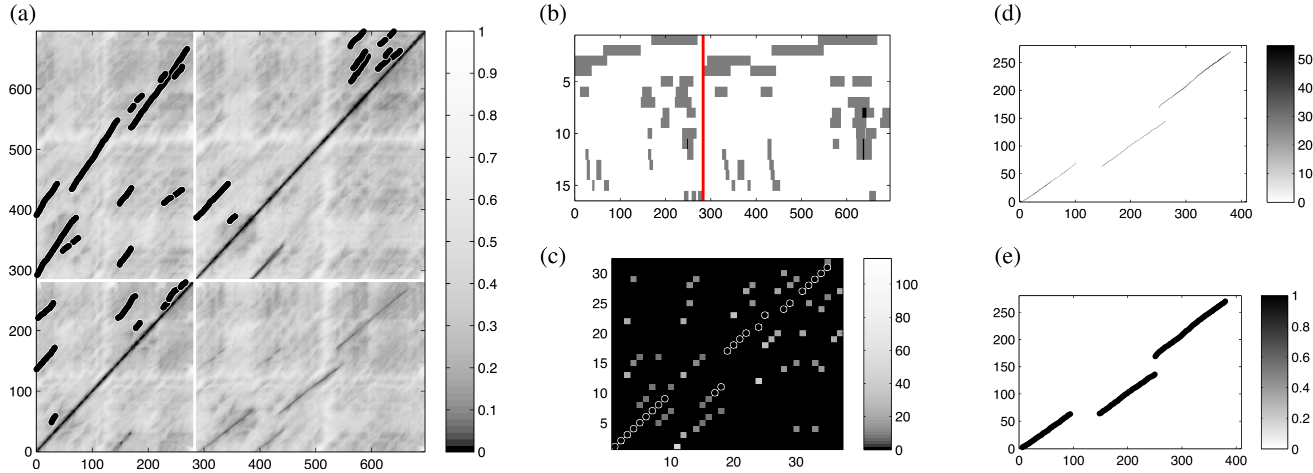

Figure 2. Joint structure analysis and partial synchronization for two structurally modified versions of Beethoven's Fifth Symphony Op. 67. The first version is a MIDI version and the second one an audio recording by Bernstein. **(a)** Enhanced joint similarity matrix and extracted paths. **(b)** Similarity clusters. **(c)** Segment-based score matrix $\mathcal{M}$ and match (indicated by black dots). **(d)** Matrix representation of matched segments. **(e)** Partial synchronization result.

Fig. 1a, where the boundaries between the two versions are indicated by white horizontal and vertical lines.

In the next step, the path structure is extracted from the joint similarity matrix. Here, the general principle is that each path of low cost running in a direction along the main diagonal (gradient $(1,1)$) corresponds to a pair of similar feature subsequences. Note that relative tempo differences in similar segments are encoded by the gradient of the path (which is then in a neighborhood of $(1,1)$). To ease the path extraction step, we enhance the path structure of $\mathcal{S}$ by a suitable smoothing technique that respects relative tempo differences. The paths can then be extracted by a robust and efficient greedy strategy, see Fig. 1b. Here, because of the symmetry of $\mathcal{S}$, one only has to consider the upper left part of $\mathcal{S}$. Furthermore, we prohibit paths crossing the boundaries between the two versions. As a result, each extracted path encodes a pair of musically similar segments, where each segment entirely belongs either to the first or to the second version. To determine the global repetitive structure, we use a one-step transitivity clustering procedure, which balances out the inconsistencies introduced by inaccurate and incorrect path extractions. For details, we refer to [8, 10].

Altogether, we obtain a set of similarity clusters. Each similarity cluster in turn consists of a set of pairwise similar segments encoding the repetitions of a segment within and across the two versions. Fig. 1c shows the resulting set of similarity clusters for our Bach example. Both of the clusters consist of three segments, where the first cluster corresponds to the three B-parts B^1, B^2_1, and B^2_2 and the second cluster to the three A-parts A^1, A^2_1, and A^2_2. The joint analysis has several advantages compared to two separate analyses. First note that, since there are no repetitions in the first version, a separate structure analysis for the first version would not have yielded any structural information. Second, the similarity clusters of the joint structure analysis naturally induce musically meaningful partial alignments between the two versions. For example, the first cluster shows that B^1 may be aligned to B^2_1 or to B^2_2. Finally, note that the delicate path extraction step often results in inaccurate and fragmented paths. Because of the transitivity step, the joint clustering procedure balances out these flaws and compensates for missing parts to some extent by using joint information *across* the two versions.

On the downside, a joint structural analysis is computationally more expensive than two separate analyses. Therefore, in the structure analysis step, our strategy is to use a relatively low feature resolution of 1 Hz. This resolution may then be increased in the subsequent synchronization step (Sect. 3) and annotation application (Sect. 4). Our current MATLAB implementation can easily deal with an overall length up to $N = 3000$ corresponding to more then forty minutes of music material. (In this case, the overall computation time adds up to 10-400 seconds with the path extraction step being the bottleneck, see [10]). Thus, our implementation allows for a joint analysis even for long symphonic movements of a duration of more than 20 minutes.

Another drawback of the joint analysis is that local inconsistencies across the two versions may cause an over-fragmentation of the music material. This may result in a large number of incomplete similarity clusters containing many short segments. As an example, we consider a MIDI version as well as a Bernstein audio recording of the first movement of Beethoven's Fifth Symphony Op. 67. We structurally modified both versions by removing some sections. Fig. 2a shows the enhanced joint similarity matrix and Fig. 2b the set of joint similarity clusters. Note that some of the resulting 16 clusters contain semantically meaningless segments stemming from spuriously extracted path fragments. At this point, one could try to improve the overall structure result by a suitable postprocessing procedure. This itself constitutes a difficult research problem and is not in the scope of this paper. Instead, we introduce a procedure for partial music alignment, which has some degree of robustness to inaccuracies and flaws in the previously extracted structural data.

3 PARTIAL SYNCHRONIZATION

Given two different representations of the same underlying piece of music, the objective of *music synchronization* is to automatically identify and link semantically corresponding events within the two versions. Most of the recent synchronization approaches use some variant of dynamic time warping (DTW) to align the feature sequences extracted from the two versions, see [8]. In classical DTW, all elements of one sequence are matched to elements in the other sequence (while respecting the temporal order). This is problematic when elements in one sequence do not have suitable counterparts in the other sequence. In the presence of structural differences between the two sequences, this typically leads to corrupted and musically meaningless alignments [11]. Also more flexible alignment strategies such as subsequence DTW or partial matching strategies as used in biological sequence analysis [4] do not properly account for such structural differences.

A first approach for partial music synchronization has been described in [11]. Here, the idea is to first construct a path-constrained similarity matrix, which a priori constricts possible alignment paths to a semantically meaningful choice of admissible cells. Then, in a second step, a path-constrained alignment can be computed using standard matching procedures based on dynamic programming.

We now carry this idea even further by using the segments of the joint similarity clusters as constraining elements in the alignment step. To this end, we consider pairs of segments, where the two segments lie within the same similarity cluster and belong to different versions. More precisely, let $\mathcal{C} = \{C_1, \ldots, C_M\}$ be the set of clusters obtained from the joint structure analysis. Each similarity cluster C_m, $m \in [1 : M]$, consists of a set of segments (i.e., subsequences of the concatenated feature sequence W). Let $\alpha \in C_m$ be such a segment. Then let $\ell(\alpha)$ denote the length of α and $c(\alpha) := m$ the cluster affiliation. Recall that α either belongs to the first version (i.e., α is a subsequence of U) or to the second version (i.e., α is a subsequence of V). We now form two lists of segments. The first list $(\alpha_1, \ldots, \alpha_I)$ consists of all those segments that are contained in some cluster of $\mathcal{C}$ and belong to the first version. The second list $(\beta_1, \ldots, \beta_J)$ is defined similarly, where the segments now belong to the second version. Both lists are sorted according to the start positions of the segments. (In case two segments have the same start position, we break the tie by also considering the cluster affiliation.) We define a segment-based $I \times J$-score matrix $\mathcal{M}$ by

$$\mathcal{M}(i,j) := \begin{cases} \ell(\alpha_i) + \ell(\beta_j) & \text{for } c(\alpha_i) = c(\beta_j), \\ 0 & \text{otherwise,} \end{cases}$$

$i \in [1 : I], j \in [1 : J]$. In other words, $\mathcal{M}(i,j)$ is positive if and only if α_i and β_j belong to the same similarity cluster. Furthermore, $\mathcal{M}(i,j)$ depends on the lengths of the two segments. Here, the idea is to favor long segments in the synchronization step. For an illustration, we consider the Bach example of Fig. 1, where $(\alpha_1, \ldots, \alpha_I) = (A^1, B^1)$ and $(\beta_1, \ldots, \beta_J) = (A_1^2, A_2^2, B_1^2, B_2^2)$. The resulting matrix $\mathcal{M}$ is shown in Fig. 1d. For another more complex example, we refer to Fig. 2c.

Now, a segment-based *match* is a sequence $\mu = (\mu_1, \ldots, \mu_K)$ with $\mu_k = (i_k, j_k) \in [1 : I] \times [1 : J]$ for $k \in [1 : K]$ satisfying $1 \leq i_1 < i_2 < \ldots < i_K \leq I$ and $1 \leq j_1 < j_2 < \ldots < j_K \leq J$. Note that a match induces a partial assignment of segment pairs, where each segment is assigned to at most one other segment. The *score* of a match μ with respect to $\mathcal{M}$ is then defined as $\sum_{k=1}^{K} \mathcal{M}(i_k, j_k)$. One can now use standard techniques to compute a score-maximizing match based on dynamic programming, see [4, 8]. For details, we refer to the literature. In the Bach example, the score-maximizing match μ is given by $\mu = ((1,1),(2,3))$. In other words, the segment $\alpha_1 = A^1$ of the first version is assigned to segment $\beta_1 = A_1^2$ of the second version and $\alpha_2 = B^1$ is assigned to $\beta_3 = B_1^2$.

In principle, the score-maximizing match μ constitutes our partial music synchronization result. To make the procedure more robust to inaccuracies and to remove cluster redundancies, we further clean the synchronization result in a postprocessing step. To this end, we convert the score-maximizing match μ into a sparsepath-constrained similarity matrix $\mathcal{S}^{\text{path}}$ of size $L \times M$, where L and M are the lengths of the two feature sequences U and V, respectively. For each pair of matched segments, we compute an alignment path using a global synchronization algorithm [9].Each cell of such a path defines a non-zero entry of $\mathcal{S}^{\text{path}}$, where the entry is set to the length of the path (thus favoring long segments in the subsequent matching step). All other entries of the matrix $\mathcal{S}^{\text{path}}$ are set to zero. Fig. 1f and Fig. 2d show the resulting path-constrained similarity matrices for the Bach and Beethoven example, respectively. Finally, we apply the procedure as described in [11] using $\mathcal{S}^{\text{path}}$ (which is generally much sparser than the path-constrained similarity matrices as used in [11])to obtain a purified synchronization result, see Fig. 1g and Fig. 2e.

To evaluate our synchronization procedure, we performed similar experiments as described in [11]. In one experiment, we formed synchronization pairs each consisting of two different versions of the same piece. Each pair

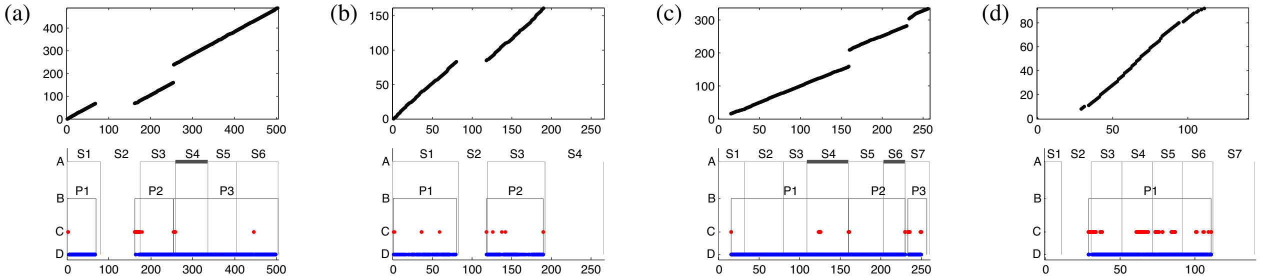

Figure 3. Partial synchronization results for various MIDI-audio synchronization pairs. The top figures show the final path components of the partial alignments and the bottom figures indicate the ground truth (Row A), the final annotations (Row B), and a classification into correct (Row D) and incorrect annotations (Row C), see text for additional explanations. The pieces are specified in Table 1. **(a)** Haydn (RWC C001), **(b)** Schubert (RWC C048, distorted), **(c)** Burke (P093), **(d)** Beatles ("Help!", distorted).

consists either of an audio recording and a MIDI version or of two different audio recordings (interpreted by different musicians possibly in different instrumentations). We manually labeled musically meaningful sections of all versions and then modified the pairs by randomly removing or duplicating some of the labeled sections, see Fig. 3. The partial synchronization result computed by our algorithm was analyzed by means of its path components. A path component is said to be *correct* if it aligns corresponding musical sections. Similarly, a match is said to be *correct* if it covers (up to a certain tolerance) all semantically meaningful correspondences between the two versions (this information is given by the ground truth) and if all its path components are correct. We tested our algorithm on more than 387 different synchronization pairs resulting in a total number of 1080 path components. As a result, 89% of all path components and 71% of all matches were correct (using a tolerance of 3 seconds).

The results obtained by our implementation of the segment-based synchronization approach are qualitatively similar to those reported in [11]. However, there is one crucial difference in the two approaches. In [11], the authors use a combination of various ad-hoc criteria to construct a path-constrained similarity matrix as basis for their partial synchronization. In contrast, our approach uses only the structural information in form of the joint similarity clusters to derive the partial alignment. Furthermore, the availability of structural information within and across the two versions allows for recovering missing relations based on suitable transitivity considerations. Thus, each improvement of the structure analysis will have a direct positive effect on the quality of the synchronization result.

4 AUDIO ANNOTATION

The synchronization of an audio recording and a corresponding MIDI version can be regarded as an automated annotation of the audio recording by means of the explicit note events given by the MIDI file. Often, MIDI versions are used as a kind of score-like symbolic representation of the underlying musical work, where redundant information such as repetitions are not encoded explicitly. This is a further setting with practical relevance where two versions to be aligned have a different repetitive structure (an audio version with repetitions and a score-like MIDI version without repetitions). In this setting, one can use our segment-based partial synchronization to still obtain musically adequate audio annotations.

We now summarize one of our experiments, which has been conducted on the basis of synchronization pairs consisting of structurally equivalent audio and MIDI versions.[1] We first globally aligned the corresponding audio and MIDI versions using a temporally refined version of the synchronization procedure described in [9]. These alignments were taken as ground truth for the audio annotation. Similar to the experiment of Sect. 3, we manually labeled musically meaningful sections of the MIDI versions and randomly removed or duplicated some of these sections. Fig. 3a illustrates this process by means of the first movement of Haydn's Symphony No. 94 (RWC C001). **Row A** of the bottom part shows the original six labeled sections S1 to S6 (warped according to the audio version). In the modification, S2 was removed (no line) and S4 was duplicated (thick line). Next, we partially aligned the modified MIDI with the original audio recording as described in Sect. 3. The resulting three path components of our Haydn example are shown in the top part of Fig. 3a. Here, the vertical axis corresponds to the MIDI version and the horizontal axis to the audio version. Furthermore, **Row B** of the bottom part shows the projections of the three path components onto the audio axis resulting in the three segments P1, P2, and P3. These segments are aligned to segments in the MIDI thus being annotated by the corresponding MIDI events. Next, we compared these partial annotations with the ground truth annotations on the MIDI note event level. We say that an alignment of a note event to a physical time position of the audio version is correct in a *weak* (*strong*) sense, if there is

[1] Most of the audio and MIDI files were taken from the RWC music database [6]. Note that for the classical pieces, the original RWC MIDI and RWC audio versions are not aligned.

Composer	Piece	RWC	Original		Distorted	
			weak	strong	weak	strong
Haydn	Symph. No. 94, 1st Mov.	C001	98	97	97	95
Beethoven	Symph. Op. 67, 1st Mov.	C003	99	98	95	91
Beethoven	Sonata Op. 57, 1st Mov.	C028	99	99	98	96
Chopin	Etude Op. 10, No. 3	C031	93	93	93	92
Schubert	Op. 89, No. 5	C048	97	96	95	95
Burke	Sweet Dreams	P093	88	79	74	63
Beatles	Help!	—	97	96	77	74
Average			96	94	91	87

Table 1. Examples for automated MIDI-audio annotation (most of files are from the RWC music database [6]). The columns show the composer, the piece of music, the RWC identifier, as well as the annotation rate (in %) with respect to the weak and strong criterion for the original MIDI and some distorted MIDI.

a ground truth alignment of a note event of the same pitch (and, in the strong case, additionally lies in the same musical context by checking an entire neighborhood of MIDI notes) within a temporal tolerance of 100 ms. In our Haydn example, the weakly correct partial annotations are indicated in **Row D** and the incorrect annotations in **Row C**.

The other examples shown in Fig. 3 give a representative impression of the overall annotation quality. Generally, the annotations are accurate—only at the segment boundaries there are some larger deviations. This is due to our path extraction procedure, which often results in “frayed” path endings. Here, one may improve the results by correcting the musical segment boundaries in a postprocessing step based on cues such as changes in timbre or dynamics. A more critical example (Beatles example) is shown Fig. 3d, where we removed two sections (S2 and S7) from the MIDI file and temporally distorted the remaining parts. In this example, the MIDI and audio version also exhibit significant differences on the feature level. As a result, an entire section (S1) has been left unannotated leading to a relatively poor rate of 77% (74%) of correctly annotated note events with respect to the weak (strong) criterion.

Finally, Table 1 shows further rates of correctly annotated note events for some representative examples. Additionally, we have repeated our experiments with significantly temporally distorted MIDI files (locally up to ±20%). Note that most rates only slightly decrease (e. g., for the Schubert piece, from 97% to 95% with respect to the weak criterion), which indicates the robustness of our overall annotation procedure to local tempo differences. Further results as well as audio files of sonifications can be found at `http://www-mmdb.iai.uni-bonn.de/projects/partialSync/`

5 CONCLUSIONS

In this paper, we have introduced the strategy of performing a joint structural analysis to detect the repetitive structure within and across different versions of the same musical work. As a core component for realizing this concept, we have discussed a structure analysis procedure that can cope with relative tempo differences between repeating segments. As further contributions, we have shown how joint structural information can be used to deal with structural variations in synchronization and annotation applications. The tasks of *partial* music synchronization and annotation is a much harder then the *global* variants of these tasks. The reason for this is that in the partial case one needs *absolute* similarity criteria, whereas in the global case one only requires *relative* criteria. One main message of this paper is that automated music structure analysis is closely related to partial music alignment and annotation applications. Hence, improvements and extensions of current structure analysis procedures to deal with various kinds of variations is of fundamental importance for future research.

6 REFERENCES

[1] V. Arifi, M. Clausen, F. Kurth, and M. Müller. Synchronization of music data in score-, MIDI- and PCM-format. *Computing in Musicology*, 13, 2004.

[2] M.A. Bartsch, G.H. Wakefield: Audio thumbnailing of popular music using chroma-based representations. IEEE Trans. on Multimedia **7**(1) (2005) 96–104.

[3] R. Dannenberg, N. Hu, *Pattern discovery techniques for music audio*, Proc. ISMIR, Paris, France, 2002.

[4] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*, Cambridge Univ. Press, 1999.

[5] M. Goto, *A chorus section detection method for musical audio signals and its application to a music listening station*, IEEE Transactions on Audio, Speech & Language Processing **14** (2006), no. 5, 1783–1794.

[6] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Popular, classical and jazz music databases. Proc. ISMIR, Paris, France, 2002.

[7] N. Hu, R. Dannenberg, and G. Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *Proc. IEEE WASPAA, New Paltz, NY*, October 2003.

[8] M. Müller: Information Retrieval for Music and Motion. Springer (2007).

[9] M. Müller, H. Mattes, and F. Kurth. An efficient multiscale approach to audio synchronization. In *Proc. ISMIR, Victoria, Canada*, pages 192–197, 2006.

[10] M. Müller, F. Kurth, *Towards structural analysis of audio recordings in the presence of musical variations*, EURASIP Journal on Advances in Signal Processing 2007, Article ID 89686, 18 pages.

[11] M. Müller, D. Appelt: Path-constrained partial music synchronization. In: Proc. International Conference on Acoustics, Speech, and Signal Processing, Las Vegas, USA (2008).

[12] G. Peeters, *Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach*, Proc. ISMIR, Vienna, Austria, 2007.

[13] C. Rhodes, M. Casey, *Algorithms for determining and labelling approximate hierarchical self-similarity*, Proc. ISMIR, Vienna, Austria, 2007.

[14] F. Soulez, X. Rodet, and D. Schwarz. Improving polyphonic and poly-instrumental music to score alignment. In *Proc. ISMIR, Baltimore, USA*, 2003.

[15] R. J. Turetsky and D. P. Ellis. Force-Aligning MIDI Syntheses for Polyphonic Music Transcription Generation. In *Proc. ISMIR, Baltimore, USA*, 2003.