FRAME-LEVEL AUDIO SEGMENTATION FOR ABRIDGED MUSICAL WORKS

Thomas Prätzlich, Meinard Müller
International Audio Laboratories Erlangen
{thomas.praetzlich, meinard.mueller}@audiolabs-erlangen.de

ABSTRACT

Large-scale musical works such as operas may last several hours and typically involve a huge number of musicians. For such compositions, one often finds different arrangements and abridged versions (often lasting less than an hour), which can also be performed by smaller ensembles. Abridged versions still convey the flavor of the musical work containing the most important excerpts and melodies. In this paper, we consider the task of automatically segmenting an audio recording of a given version into semantically meaningful parts. Following previous work, the general strategy is to transfer a reference segmentation of the original complete work to the given version. Our main contribution is to show how this can be accomplished when dealing with strongly abridged versions. To this end, opposed to previously suggested segment-level matching procedures, we adapt a frame-level matching approach for transferring the reference segment information to the unknown version. Considering the opera "Der Freischütz" as an example scenario, we discuss how to balance out flexibility and robustness properties of our proposed frame-level segmentation procedure.

1. INTRODUCTION

Over the years, many musical works have seen a great number of reproductions, ranging from reprints of the sheet music to various audio recordings of performances. For many works this has led to a wealth of co-existing versions including arrangements, adaptations, cover versions, and so on. Establishing semantic correspondences between different versions and representations is an important step for many applications in Music Information Retrieval. For example, when comparing a musical score with an audio version, the goal is to compute an alignment between measures or notes in the score and points in time in the audio version. This task is motivated by applications such as score following [1], where the score can be used to navigate through a corresponding audio version and vice versa. The aligned score information can also be used to parameterize an audio processing algorithm such as in score-informed source separation [4, 12]. When working with two audio versions, alignments are useful for comparing different performances of the same piece of music [2,3]. In cover song identification, alignments can be used to compute the similarity between two recordings [11]. Alignment techniques can also help to transfer meta data and segmentation information between recordings. In [7], an unknown recording is queried against a database of music recordings to identify a corresponding version of the same musical work. After a successful identification, alignment techniques are used to transfer the segmentation given in the database to the unknown recording.

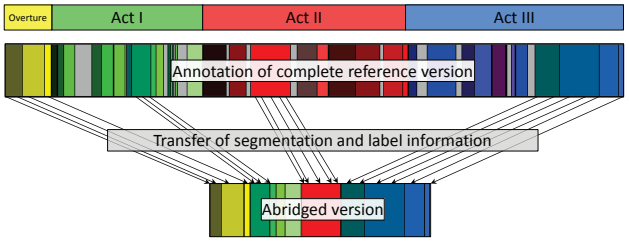

Figure 1. Illustration of the proposed method. Given the annotated segments on a complete reference version of a musical work, the task is to transfer the segment information to an abridged version.

A similar problem was addressed in previous work, where the goal was to transfer a labeled segmentation of a reference version onto an unknown version of the same musical work [10]. The task was approached by a segment-level matching procedure, where one main assumption was that a given reference segment either appears more or less in the same form in the unknown version or is omitted completely.

In abridged versions of an opera, however, this assumption is often not valid. Such versions strongly deviate from the original by omitting a large portion of the musical material. For example, given a segment in a reference version, one may no longer find the start or ending sections of this segment in an unknown version, but only an intermediate section. Hence, alignment techniques that account for structural differences are needed. In [5], a music synchronization procedure accounting for structural differences in recordings of the same piece of music is realized with an adaption of the Needleman-Wunsch algorithm. The algorithm penalizes the skipping of frames in the alignment by adding an additional cost value for each skipped frame.

Thus, the cost for skipping a sequence of frames is dependent on the length of the sequence. In abridged versions, however, omission may occur on an arbitrary scale, ranging from several musical measures up to entire scenes of an opera. In such a scenario, a skipping of long sequences should not be more penalized as a skipping of short sequences. In this work, we will therefore use a different alignment strategy.

In this paper, we address the problem of transferring a labeled reference segmentation onto an unknown version in the case of abridged versions, see Figure 1. As our main contribution, we show how to approach this task with a frame-level matching procedure, where correspondences between frames of a reference version and frames of an unknown version are established. The labeled segment information of the reference version is then transferred to the unknown version only for frames for which a correspondence has been established. Such a frame-level procedure is more flexible than a segment-level procedure. However, on the downside, it is less robust. As a further contribution, we show how to stabilize the robustness of the frame-level matching approach while preserving most of its flexibility.

The remainder of this paper is structured as follows: In Section 2, we discuss the relevance of abridged music recordings and explain why they are problematic in a standard music alignment scenario. In Section 3, we review the segment-level matching approach from previous work (Section 3.2), and then introduce the proposed frame-level segmentation pipeline (Section 3.3). Subsequently, we present some results of a qualitative (Section 4.2) and a quantitative (Section 4.3) evaluation and conclude the paper with a short summary (Section 5).

2. MOTIVATION

For many musical works, there exists a large number of different versions such as cover songs or different performances in classical music. These versions can vary greatly in different aspects such as the instrumentation or the structure. Large-scale musical works such as operas usually need a huge number of musicians to be performed. For these works, one often finds arrangements for smaller ensembles or piano reductions. Furthermore, performances of these works are usually very long. Weber's opera "Der Freischütz", for example, has an average duration of about two hours. Taking it to an extreme, Wagner's epos "Der Ring der Nibelungen", consists of four operas having an overall duration of about 15 hours. For such large-scale musical works, one often finds abridged versions. These versions usually present the most important material of a musical work in a strongly shortened and structurally modified form. Typically, these structural modifications include omissions of repetitions and other "non-essential" musical passages. Abridged versions were very common in the early recording days due to space constraints of the sound carriers. The opera "Der Freischütz" would have filled 18 discs on a shellac record. More recently, abridged versions or excerpts of a musical work can often be found as bonus tracks on CD records. In a standard alignment scenario, abridged versions are particularly problematic as they omit material on different scales, ranging from the omission of several musical measures up to entire parts.

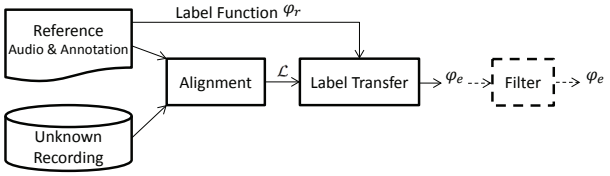

Figure 2. Illustration of the proposed frame-level segmentation pipeline. A reference recording with a reference label function φ_r is aligned with an unknown version. The alignment $\mathcal{L}$ is used to transfer φ_r to the unknown version yielding φ_e.

3. METHODS

In this section, we show how one can accomplish the task of transferring a given segmentation of a reference version, say X, onto an unknown version, say Y. The general idea is to use alignment techniques to find corresponding parts between X and Y, and then to transfer on those parts the given segmentation from X to Y.

After introducing some basic notations on alignments and segmentations (Section 3.1), we review the segment-level matching approach from our previous work (Section 3.2). Subsequently, we introduce our frame-level segmentation approach based on partial matching (Section 3.3).

3.1 Basic Notations

3.1.1 Alignments, Paths, and Matches

Let $[1:N] := \{1, 2, \ldots, N\}$ be an index set representing the *time line* of a discrete signal or feature sequence $X = (x_1, x_2, \ldots, x_N)$. Similarly, let $[1:M]$ be the time line of a second sequence $Y = (y_1, \ldots, y_M)$. An *alignment* between two time lines $[1:N]$ and $[1:M]$ is modeled as a set $\mathcal{L} = (p_1, \ldots, p_L) \subseteq [1:N] \times [1:M]$. An element $p_\ell = (n_\ell, m_\ell) \in \mathcal{L}$ is called a *cell* and encodes a correspondence between index $n_\ell \in [1:N]$ of the first time line and index $m_\ell \in [1:M]$ of the second one. In the following, we assume $\mathcal{L}$ to be in lexicographic order. $\mathcal{L}$ is called a *match* if $(p_{\ell+1} - p_\ell) \in \mathbb{N} \times \mathbb{N}$ for $\ell \in [1:L-1]$. Note that this condition implies strict monotonicity and excludes the possibility to align an index of the first time line with many indices of the other and vice versa. An alignment can also be constrained by requiring $(p_{\ell+1} - p_\ell) \in \Sigma$ for a given set Σ of admissible step sizes. A typical choice for this set is $\Sigma = \{(1,1), (1,0), (0,1)\}$, which allows to align an index of one time line to many indices of another, and vice versa. Sometimes other sets such as $\Sigma = \{(1,1), (1,2), (2,1)\}$ are used to align sequences which are assumed to be structurally and temporally mostly consistent. If $\mathcal{L}$ fulfills a given step size condition, $\mathcal{P} = \mathcal{L}$ is called a *path*. Note that alignments that fulfill Σ_1 and Σ_2 are both paths, but only an alignment fulfilling Σ_2 is also a match.

3.1.2 Segments and Segmentation

We formally define a *segment* to be a set $\alpha = [s:t] \subseteq [1:N]$ specified by its start index s and its end index t. Let $|\alpha| := t-s+1$ denote the length of α. We define a (partial) *segmentation* of size K to be a set $\mathcal{A} := \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ of pairwise disjoint segments: $\alpha_k \cap \alpha_j = \emptyset$ for $k, j \in [1:K], k \neq j$.

3.1.3 Labeling

Let $[0:K]$ be a set of labels. The label 0 plays a special role and is used to label everything that has not been labeled otherwise. A *label function* φ maps each index $n \in [1:N]$ to a label $k \in [0:K]$:

$$\varphi : [1:N] \to [0:K].$$

The pair $([1:N], \varphi)$ is called a *labeled time line*. Let $n \in [1:N]$ be an index, $\alpha = [s:t]$ be a segment, and $k \in [0:K]$ be a label. Then the pair (n,k) is called a *labeled index* and the pair (α, k) a labeled segment. A labeled segment (α, k) induces a labeling of all indices $n \in \alpha$. Let $\mathcal{A} := \{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ be a segmentation of $[1:N]$ and $[0:K]$ be the label set. Then the set $\{(\alpha_k, k) \,|\, k \in [1:K]\}$ is called a *labeled segmentation* of $[1:N]$. From a labeled segmentation one obtains a label function on $[1:N]$ by setting $\varphi(n) := k$ for $n \in \alpha_k$ and $\varphi(n) := 0$ for $n \in [1:N] \setminus \bigcup_{k\in[1:K]} \alpha_k$. Vice versa, given a label function φ, one obtains a labeled segmentation in the following way. We call consecutive indices with the same label a *run*. A segmentation of $[1:N]$ is then derived by considering runs of maximal length. We call this segmentation the *segmentation induced by* φ.

3.2 Segment-Level Matching Approach

The general approach in [10] is to apply segment-level matching techniques based on dynamic time warping (DTW) to transfer a labeled reference segmentation to an unknown version. Given a labeled segmentation $\mathcal{A}$ of X, each $\alpha_k \in \mathcal{A}$ is used as query to compute a ranked list of matching candidates in Y. The matching candidates are derived by applying a subsequence variant of the DTW algorithm using the step size conditions $\Sigma = \{(1,1),(1,2),(2,1)\}$, see [8, Chapter 5]. The result of the subsequence DTW procedure is a matching score and an alignment path $\mathcal{P} = (p_1, \ldots, p_L)$ with $p_\ell = (n_\ell, m_\ell)$. $\mathcal{P}$ encodes an alignment of the segment $\alpha_k := [n_1 : n_L] \subseteq [1:N]$ and the corresponding segment $[m_1 : m_L] \subseteq [1:M]$ in Y. To derive a final segmentation, one segment from each matching candidate list is chosen such that the sum of the alignment scores of all chosen segments is maximized by simultaneously fulfilling the following constraints. First, the chosen segments have to respect the temporal order of the reference segmentation and second, no overlapping segments are allowed in the final segmentation. Furthermore, the procedure is adapted to be robust to tuning differences of individual segments, see [10] for further details.

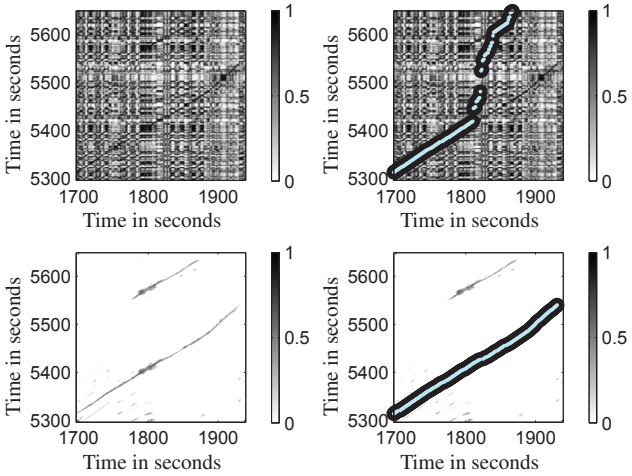

Figure 3. Excerpt of similarity matrices of the reference `Kle1973` and `Kna1939` before (top) and after enhancement (bottom), shown without match (left) and with match (right).

3.3 Frame-Level Segmentation Approach

The basic procedure of our proposed frame-level segmentation is sketched in Figure 2. First, we use a *partial matching* algorithm (Section 3.3.1) to compute an alignment $\mathcal{L}$. Using $\mathcal{L}$ and the reference label function φ_r obtained from the reference annotation $\mathcal{A}$ of X, an *induced label function* φ_e to estimate the labels on Y is derived (Section 3.3.2). Finally, we apply a mode filter (Section 3.3.3) and a filling up strategy (Section 3.3.4) to derive the final segmentation result.

3.3.1 Partial Matching

Now we describe a procedure for computing a partial matching between two sequences as introduced in [8]. To compare the two feature sequences X and Y, we compute a similarity matrix $S(n,m) := s(x_n, y_m)$, where s is a suitable similarity measure. The goal of the partial matching procedure is to find a score-maximizing match through the matrix S. To this end, we define the *accumulated score matrix* D by $D(n,m) := \max\{D(n-1,m), D(n,m-1), D(n-1,m-1) + S(n,m)\}$ with $D(0,0) := D(n,0) := D(0,m) := 0$ for $1 \leq n \leq N$ and $1 \leq m \leq M$. The score maximizing match can then be derived by backtracking through D, see [8, Chapter 5]. Note that only diagonal steps contribute to the accumulated score in D. The partial matching algorithm is more flexible in aligning two sequences than the subsequence DTW approach, as it allows for skipping frames at any point in the alignment. However, this increased flexibility comes at the cost of loosing robustness. To improve the robustness, we apply path-enhancement (smoothing) on S, and suppress other noise-like structures by thresholding techniques [9, 11]. In this way, the algorithm is less likely to align small scattered fragments. Figure 3 shows an excerpt of a similarity matrix before and after path-enhancement together with the computed matches.

3.3.2 Induced Label Function

Given a labeled time line $([1:N], \varphi_r)$ and an alignment $\mathcal{L}$, we derive a label function φ_e on $[1:M]$ by setting:

$$\varphi_e(m) := \begin{cases} \varphi_r(n) & \text{if } (n,m) \in \mathcal{L} \\ 0 & \text{else,} \end{cases}$$

for $m \in [1:M]$. See Figure 4 for an illustration.

3.3.3 Local Mode Filtering

The framewise transfer of the labels may lead to very short and scattered runs. Therefore, to obtain longer runs and a more homogeneous labeling, especially at segment boundaries, we introduce a kind of smoothing step by applying a mode filter. The *mode* of a sequence $\mathcal{S} = (s_1, s_2, \ldots, s_N)$ is the most frequently appearing value and is formally defined by $\mathrm{mode}(\mathcal{S}) := \arg\max_{s \in \mathcal{S}} |\{n \in [1:N] \colon s_n = s\}|$. A *local mode filter* of length $L = 2q+1$ with $q \in \mathbb{N}$ replaces each element $s_n \in \mathcal{S}$, $n \in [1:N]$, in a sequence by the mode of its neighborhood $(s_{n-q}, \ldots, s_{n+q})$:

$$\mathrm{modefilt}_q(\mathcal{S})(n) := \mathrm{mode}(s_{n-q}, \ldots, s_{n+q}).$$

Note that the mode may not be unique. In this case, we apply the following strategy in the mode filter. If the element s_n is one of the modes, s_n is left unmodified by the filter. Otherwise, one of the modes is chosen arbitrarily.

In our scenario, we apply the local mode filter on a labeled time line $([1:N], \varphi_e)$ by inputting the sequence $\varphi_e([1:N]) := (\varphi_e(1), \varphi_e(2), \ldots, \varphi_e(N)))$ into the filter, see Figure 4 for an illustration. The reason to use the mode opposed to the median to filter segment labels, is that labels are nominal data and therefore have no ordering (integer labels were only chosen for the sake of simplicity).

3.3.4 From Frames to Segments (Filling Up)

In the last step, we derive a segmentation from the label function φ_e. As indicated in Section 3.1.3, we could simply detect maximal runs and consider them as segments. However, even after applying the mode filter, there may still be runs sharing the same label that are interrupted by non-labeled parts (labeled zero). In our scenario, we assume that all segments have a distinct label and occur in the same succession as in the reference. Therefore, in the case of a sequence of equally labeled runs that are interrupted by non-labeled parts, we can assume that the runs belong to the same segment. Formally, we assign an index in between two indices with the same label (excluding the zero label) to belong to the same segment as these indices. To construct the final segments, we iterate over each $k \in [1:K]$ and construct the segments $\alpha_k = [s_k : e_k]$, such that $s_k = \min\{m \in [1:M] \colon \varphi(m) = k\}$, and $e_k = \max\{m \in [1:M] \colon \varphi(m) = k\}$, see Figure 4 for an example.

4. EVALUATION

In this section, we compare the previous segment-level matching procedure with our novel frame-level segmentation approach based on experiments using abridged versions of the opera "Der Freischütz". First we give an overview of our test set and the evaluation metric (Section 4.1). Subsequently, we discuss the results of the segment-level approach and the frame-level procedure on the abridged versions (Section 4.2). Finally, we present an experiment where we systematically derive synthetic abridged versions from a complete version of the opera (Section 4.3).

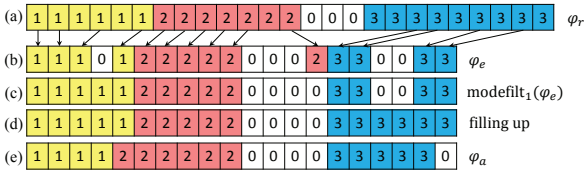

Figure 4. Example of frame-level segmentation. The arrows indicate the match between the reference version and the unknown version. **(a):** Reference label function. **(b):** Induced label function. **(c):** Mode filtered version of (b) with length $L = 3$. **(d):** Filling up on (c). **(e):** Ground truth label function.

4.1 Tests Set and Evaluation Measure

In the following experiments, we use the recording of Carlos Kleiber performed in 1973 with a duration of 7763 seconds as reference version. The labeled reference segmentation consists of 38 musical segments, see Figure 5. Furthermore, we consider five abridged versions that were recorded between 1933 and 1994. The segments of the opera that are performed in these versions are indicated by Figure 5. Note that the gray parts in the figure correspond to dialogue sections in the opera. In the following experiments, the dialogue sections are considered in the same way as non-labeled (non-musical) parts such as applause, noise or silence. In the partial matching algorithm, they are excluded from the reference version (by setting the similarity score in these regions to minus infinity), and in the segment-level matching procedure, the dialogue parts are not used as queries.

Throughout all experiments, we use CENS features which are a variant of chroma features. They are computed with a feature rate of 1 Hz (derived from 10 Hz pitch features with a smoothing length of 41 frames and a downsampling factor of 10), see [8]. Each feature vector covers roughly 4.1 seconds of the original audio.

In our subsequent experiments, the following segment-level matching (M4) and frame-level segmentation (F1–F4) approaches are evaluated:

(M4) – Previously introduced segment-level matching, see Section 3.2 and [10] for details.

(F1) – Frame-level segmentation using a similarity matrix computed with the cosine similarity s defined by $s(x,y) = \langle x, y \rangle$ for features x and y, see Section 3.3.

(F2) – Frame-level segmentation using a similarity matrix with enhanced path structures using the SM Toolbox [9]. For the computation of the similarity matrix, we used forward/backward smoothing with a smoothing length of 20

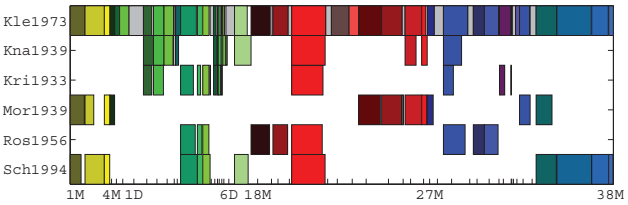

Figure 5. Visualization of relative lengths of the abridged versions compared to the reference version `Kle1973`. The gray segments indicate dialogues whereas the colored segments are musical parts.

frames (corresponding to 20 seconds) with relative tempi between $0.5-2$, sampled in 15 steps. Afterwards, a thresholding technique that retained only 5% of the highest values in the similarity matrix and a scaling of the remaining values to $[0,1]$ is applied. For details, we refer to [9] and Section 3.3.

(F3) – The same as in F2 with a subsequent mode filtering using a filter length $L = 21$ frames, see Section 3.3.3 for details.

(F4) – The segmentation derived from F3 as described in Section 3.3.4.

4.1.1 Frame Accuracy

To evaluate the performance of the different segmentation approaches, we calculate the *frame accuracy*, which is defined as the ratio of correctly labeled frames and the total number of frames in a version. Given a ground truth label function φ_a and an induced label function φ_e, the frame accuracy A_f is computed as following:

$$A_f := \frac{\sum_{k\in[0:K]} \left|{\varphi_a}^{-1}(k) \cap {\varphi_e}^{-1}(k)\right|}{\sum_{k\in[0:K]} \left|{\varphi_a}^{-1}(k)\right|}$$

We visualize the accuracy by means of an *agreement sequence* $\Delta(\varphi_a, \varphi_e)$ which we define as $\Delta(\varphi_a, \varphi_e)(m) := 1$ (white) if $\varphi_a(m) = \varphi_e(m)$ and $\Delta(\varphi_a, \varphi_e)(m) := 0$ (black) otherwise. The sequences $\Delta(\varphi_a, \varphi_e)$ visually correlates well with the values of the frame accuracy A_f, see Table 1 and the Figure 6. Note that in structural segmentation tasks, it is common to use different metrics such as the pairwise precision, recall, and f-measure [6]. These metrics disregard the absolute labeling of a frame sequence by relating equally labeled pairs of frames in an estimate to equally labeled frames in a ground truth sequence. However, in our scenario, we want to consider frames that are differently labeled in the ground truth and the induced label function as wrong. As the pairwise f-measure showed the same tendencies as the frame accuracy (which can be easily visualized), we decided to only present the frame accuracy values.

4.2 Qualitative Evaluation

In this section, we qualitatively discuss the results of our approach in more detail by considering the evaluation of the version `Kna1939`. For each of the five approaches, the results are visualized in a separate row of Figure 6,

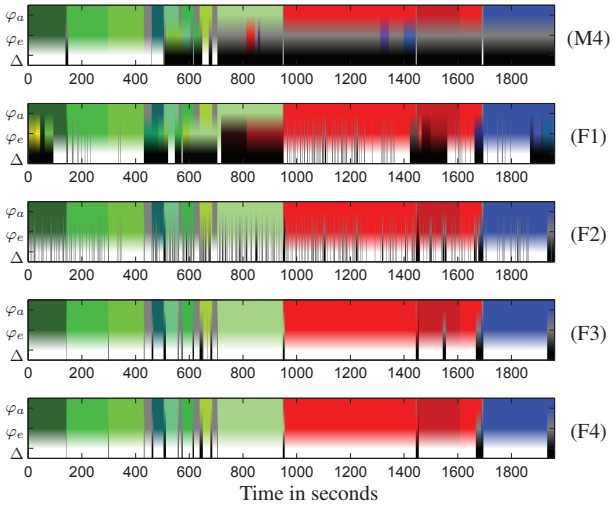

Figure 6. Segmentation results on `Kna1939` showing the ground truth label function φ_a, the induced label function φ_e, and the agreement sequence $\Delta := \Delta(\varphi_a, \varphi_e)$. White encodes an agreement and black a disagreement between φ_a and φ_e. **(M4),(F1),(F2),(F3),(F4):** See Section 4.1.

showing the ground truth φ_a, the induced label function φ_e and the agreement sequence $\Delta(\varphi_a, \varphi_e)$.

For `Kna1939`, the segment-level matching approach M4 does not work well. Only 28% of the frames are labeled correctly. The red segment, for example, at around 1500 seconds is not matched despite the fact that it has roughly the same overall duration as the corresponding segment in the reference version, see Figure 5. Under closer inspection, it becomes clear that it is performed slower than the corresponding segment in the reference version, and that some material was omitted at the start, in the middle and the end of the segment. The frame-level matching approach F1 leads to an improvement, having a frame accuracy of $A_f = 0.520$. However, there are still many frames wrongly matched. For example, the overture of the opera is missing in `Kna1939`, but frames from the overture (yellow) of the reference are matched into a segment from the first act (green), see Figure 6. Considering that the opera consists of many scenes with harmonically related material and that the partial matching allows for skipping frames at any point in the alignment, it sometimes occurs that not the semantically corresponding frames are aligned, but harmonically similar ones. This problem is better addressed in approach F2, leading to an improved frame accuracy of 0.788. The enhancement of path structures in the similarity matrix in this approach leads to an increased robustness of the partial matching. Now, all high similarity values are better concentrated in path structures of the similarity matrix.

As a result, the algorithm is more likely to follow sequences of harmonically similar frames, see also Figure 3. However, to follow paths that are not perfectly diagonal, the partial matching algorithm needs to skip frames in the alignment, which leads to a more scattered label function. This is approached by F3 which applies a mode filter on the label function from F2, resulting in an improved frame

	$dur.(s)$	**M4**	F1	F2	F3	**F4**
Kna1939	1965	0.283	0.520	0.788	0.927	0.934
Kri1933	1417	0.390	0.753	0.777	0.846	0.870
Mor1939	1991	0.512	0.521	0.748	0.841	0.919
Ros1956	2012	0.887	0.749	0.817	0.850	0.908
Sch1994	2789	0.742	0.895	0.936	0.986	0.989
mean	2035	0.563	0.687	0.813	0.890	0.924

Table 1. Frame accuracy values on abridged versions. M4: Segment-level matching, F1: Frame-level segmentation, F2: Frame-level segmentation with path-enhanced similarity matrix, F3: Mode filtering with $L = 21$ seconds on F2. F4: Derived Segmentation on F4.

accuracy of 0.927. In F4, the remaining gaps in the label function of F3 are filled up, which leads to a frame accuracy of 0.934.

4.3 Quantitative Evaluation

In this section, we discuss the results of Table 1. Note that all abridged versions have less than 50% of the duration of the reference version (7763 seconds). From the mean frame accuracy values for all approaches, we can conclude that the segment-level matching (0.563) is not well suited for dealing with abridged versions, whereas the different strategies in the frame-level approaches F1 (0.687) – F4 (0.924) lead to a subsequent improvement of the frame accuracy. Using the segment-level approach, the frame accuracies for the versions `Ros1956` (0.887) and `Sch1994` (0.742) stand out compared to the other versions. The segments that are performed in these versions are not shortened and therefore largely coincide with the segments of the reference version. This explains why the segment-level matching still performs reasonably well on these versions.

In Figure 7, we show the frame accuracy results for the approaches M4 and F4 obtained from an experiment on a set of systematically constructed abridged versions. The frame accuracy values at 100% correspond to a subset of 10 segments (out of 38) that were taken from a complete recording of the opera "Der Freischütz" recorded by Keilberth in 1958. From this subset, we successively removed 10% of the frames from each segment by removing 5% of the frames at the start, and 5% of the frames at the end sections of the segments. In the last abridged version, only 10% of each segment remains. This experiment further supports the conclusions that the segment-level approach is not appropriate for dealing with abridged versions, whereas the frame-level segmentation approach stays robust and flexible even in the case of strong abridgments.

5. CONCLUSIONS

In this paper, we approached the problem of transferring the segmentation of a complete reference recording onto an abridged version of the same musical work. We compared the proposed frame-level segmentation approach based on partial matching with a segment-level matching strategy. In experiments with abridged recordings, we have shown that our frame-level approach is robust and flexible when enhancing the path structure of the used similarity matrix and applying a mode filter on the labeled frame sequence before deriving the final segmentation.

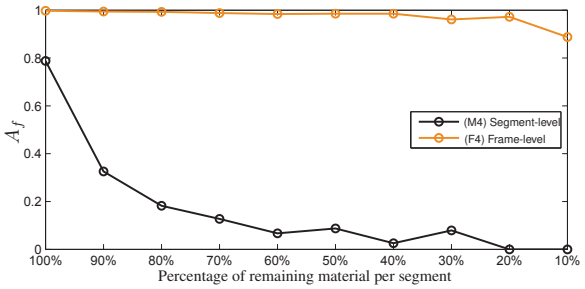

Figure 7. Performance of segment-level approach (M4) versus frame-level approach (F4) on constructed abridged versions. See Section 4.3

Acknowledgments: This work has been supported by the BMBF project *Freischütz Digital* (Funding Code 01UG1239A to C). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer IIS.

6. REFERENCES

[1] Roger B. Dannenberg and Ning Hu. Polyphonic audio matching for score following and intelligent audio editors. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 27–34, San Francisco, USA, 2003.

[2] Simon Dixon and Gerhard Widmer. MATCH: A music alignment tool chest. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, London, GB, 2005.

[3] Sebastian Ewert, Meinard Müller, Verena Konz, Daniel Müllensiefen, and Geraint Wiggins. Towards cross-version harmonic analysis of music. *IEEE Transactions on Multimedia*, 14(3):770–782, 2012.

[4] Sebastian Ewert, Bryan Pardo, Meinard Müller, and Mark D. Plumbley. Score-informed source separation for musical audio recordings: An overview. *IEEE Signal Processing Magazine*, 31(3):116–124, May 2014.

[5] Maarten Grachten, Martin Gasser, Andreas Arzt, and Gerhard Widmer. Automatic alignment of music performances with structural differences. In *ISMIR*, pages 607–612, 2013.

[6] Hanna Lukashevich. Towards quantitative measures of evaluating song segmentation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 375–380, Philadelphia, USA, 2008.

[7] Nicola Montecchio, Emanuele Di Buccio, and Nicola Orio. An efficient identification methodology for improved access to music heritage collections. *Journal of Multimedia*, 7(2):145–158, 2012.

[8] Meinard Müller. *Information Retrieval for Music and Motion*. Springer Verlag, 2007.

[9] Meinard Müller, Nanzhu Jiang, and Harald Grohganz. SM Toolbox: MATLAB implementations for computing and enhancing similiarty matrices. In *Proceedings of the AES Conference on Semantic Audio*, London, GB, 2014.

[10] Thomas Prätzlich and Meinard Müller. Freischütz digital: A case study for reference-based audio segmentation of operas, to appear. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 589–594, Curitiba, Brazil, 2013.

[11] 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*, 16:1138–1151, 2008.

[12] John Woodruff, Bryan Pardo, and Roger B. Dannenberg. Remixing stereo music with score-informed source separation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 314–319, Victoria, Canada, 2006.