Audio Engineering Society

Convention Paper

Presented at the 147th Convention
2019 October 16 – 19, New York

Alignment and Timeline Construction for Incomplete Analogue Audience Recordings of Historical Live Music Concerts

Thomas Wilmering, Florian Thalmann, and Mark B. Sandler

Centre for Digital Music (C4DM), School of Electronic Engineering and Computer Science, Queen Mary University of London

Correspondence should be addressed to Thomas Wilmering (t.wilmering@qmul.ac.uk)

ABSTRACT

Analogue recordings pose specific problems during automatic alignment, such as distortion due to physical degradation, or differences in tape speed during recording, copying, and digitisation. Oftentimes, recordings are incomplete, exhibiting gaps with different lengths. In this paper we propose a method to align multiple digitised analogue recordings of same concerts of varying quality and song segmentations. The process includes the automatic construction of a reference concert timeline. We evaluate alignment methods on a synthetic dataset and apply our algorithm to real-world data.

1 Introduction

Prior to the widespread adoption of digital recording technology many live music concerts in the second half of the 20th century were recorded on analogue tape, either directly from the mixing desk or with amateur equipment by audience members. The alignment of different recordings of the same concert pose specific problems not occurring in the digital domain. These include distortion due to physical degradation, as well as inconsistent tape speed during recording and digitisation. Moreover, live recordings often include crowd noise and many recordings are incomplete with gaps of different lengths, varying from brief interruptions, e.g. due to turning over a tape, to longer gaps in partial recordings.

In this paper we propose a method to align such recordings which differ in coverage and song segmentation, and construct a reference timeline for the visualisation and convenient comparison of the recordings, both by listening and visual information. We propose a method based on iterative application of Subsequence Dynamic Time Warping (S-DTW), and evaluate different variants of the technique for the alignment on a constructed dataset and a real-world dataset taken from the Grateful Dead Collection of the Live Music Archive (LMA)[1]. Our work builds on previous work aligning and clustering individual songs from historic live music recordings based on various audio characteristics and editorial metadata, to create an immersive virtual space that can be imported into a multichannel web or mobile audio

[1] https://archive.org/details/GratefulDead

application [1]. We illustrate how the alignment of concert recordings works for a large database and how it can be used in a novel Web application for the exploration of collections such as the one we used in our experiment. Furthermore, the analysis data can help to correct and add missing editorial metadata about the live music events.

The rest of this paper is structured as follows. In Section 2 we give an overview of the background to our research including digitised analogue live music recordings, related work in the field of audio alignment and the Dynamic Time Warping (DTW) algorithm. This is followed by a description of the alignment algorithm used in our experiments (Section 3). We evaluate the algorithm both on a synthetic dataset (Section 4.1) and on a real-world example (Section 4.2). In a separate Section (5), we describe the construction of the concert timeline from the results of the alignment of the individual tracks of the different recordings. Before concluding we discuss our algorithm and discuss future work for the optimisation of the alignment procedure (Section 6).

2 Background

2.1 Digitised Analog Live Music Recordings

Before the advent of portable digital recording devices, recordings were made on analog tape which poses several problems for the curation and distribution of archival material: Over time, analog recording suffer from degradation, and generation loss occurs when copying the recordings to compact cassette [2, 3]. Furthermore, the recordings of a concert may exhibit different speeds, resulting in changes of temporal and spectral information. Live music recorded by members of the audience may include applause and crowd noise, further complicating the automated analysis using MIR techniques [4]. Sturm [5] discusses incorrect and incomplete editorial metadata and repetitions in music corpuses on machine learning tasks in the context of genre classification.

We apply our alignment approach on material from the LMA, a collection of over 200,000 concert recordings. The content is provided by members of the etree[2] community, dedicated to the preservation and trading of legally tradeable concert recordings in lossless digital audio formats. At the time of writing, the Grateful Dead collection of the LMA holds more than 13,000 recordings of over 2,000 shows spanning the years 1965 to 1995. The improvisatory nature of Grateful Dead performances and the continued interest by both fans and scholars in the band's music and cultural impact [6] makes this collection particularly interesting for our work.

[2] http://wiki.etree.org/

2.2 Alignment of Unsynchronised Audio Sequences

Related work includes several methods proposed for the alignment of crowd-sourced digital audio/video recordings [7, 8, 9, 10]. Many proposed techniques are based on audio fingerprinting [11, 12, 13], a method to extract compact content-based signatures from perceptually relevant features of audio material [14]. Kennedy and Naaman [13] and Subramanian and Lerch [15] specifically focus on the alignment of rock concerts. Basaran et al. [10] employ a graph-based approach using sequential Monte Carlo samplers to align unsynchronised user generated audio recordings. For a comprehensive review of audio synchronisation techniques see [9, 10].

In many cases, the proposed algorithms involve cross-correlation of sequences of audio feature vectors which assume the audio sequences to be continuous and of the same pitch and speed, which may not be the case for digitised analog recordings that may exhibit temporal differences due to tape speed and local temporal variations due to errors in the tape playback. Despite its robustness against noise and distortion audio fingerprinting has been shown to be inadequate for collections whose material exhibits greater fluctuation and variance in playback speed [1, 15].

2.3 Dynamic Time Warping

Dynamic Time Warping (DTW), first introduced by Sakoe and Chiba [16] in the context of speech recognition, is a method to compare two time-dependent sequences and has been successfully applied to the alignment of audio recordings [17, 18]. For music alignment applications the sequences are usually audio feature vectors, for instance based on chroma features of two different performances of the same piece that differ in length and exhibit local temporal and spectral variations.

The classic DTW algorithm can be explained as follows. Let $X = (x_1, x_2, ..., x_I)$ with length $I \in \mathbb{N}$ and

$Y = (y_1, y_2, ..., y_J)$ with length $J \in \mathbb{N}$. In order to compute the optimal warping path between X and Y, the minimal distances are obtained from a cost matrix $I \times J$. The cost matrix is computed by evaluating the local cost measure based on a given distance measure for each pair of elements of X and Y. In classic DTW there are three constraints on the construction of the warping path $w = (w_1, ..., w_N)$ of length N [19]:

- *Boundary condition.* x_1 must be mapped with y_1 and x_N must be matched with y_J, i.e. the warping path spans the lengths of both sequences:

$$(w_1 = (1,1)) \wedge (w_N = (I,J)) \tag{1}$$

- *Monotonicity condition.* The warping path is monotonically increasing for both X and Y:

$$(i_1 \leq i_2 \leq ... \leq i_N) \wedge (j_1 \leq j_2 \leq ... \leq j_N) \tag{2}$$

- *Step size condition.* While each element of X must be matched with at least one element Y and vice versa, all index pairs must be pairwise distinct:

$$\begin{aligned} w_n + 1 - w_n \in \{(1,0),(0,1),(1,1)\} \\ \mid n \in [1 : N-1] \end{aligned} \tag{3}$$

There are several techniques to modify the DTW, for instance constraining the slope of the warping paths or defining local weights to introduce a bias in favour of paths of a specified direction. Global constraints limit the admissible values for the warping path to a subset of $[1 : I] \times [1 : J]$.

3 Alignment Algorithm

The alignment algorithm used in our experiments consist of the following steps:

1) Identifying the longest available recording of a given concert to be used as the initial reference timeline.

2) Establishing the relative difference in tape speed of another recording and the reference by computing the tuning difference and resampling the audio according to the information obtained.

3) Computing the alignment between the segmented tracks of a recording with a concatenated reference recording using a S-DTW with chroma feature vectors.

4) Finding a consensus between the individual alignment results and constructing a reference timeline along which all of them can be placed (see Section 5).

In this section we describe the modified S-DTW algorithm we apply to the audio data.

3.1 Chroma Features

The S-DTW is applied on chroma vectors corresponding to the audio signals. Chroma features are computed by adding up the subbands obtained from a spectral representation, such as fast fourier transform (FFT) or constant-Q transform (CQT), that correspond to a defined pitch-class. For example, for the western equal-tempered scale we use in our work, we construct a 12-dimensional vector for every window, where the dimensions represent the notes C to B in semitone steps. The window length in our experiments is 512 samples corresponding to 23.2ms at 22.05kHz.

We test our algorithm with three different chroma representations. In addition to the CQT based chroma (chroma CQT), we use the Chroma Energy Normalised Statistics (CENS) feature, a chroma variant developed by Müller et al. [17] to increase the robustness against temporal microdeviations and local variations in dynamics, timbre and articulation in audio alignment tasks. The construction of the CENS features involves the following steps:

1. L1-Normalisation across each chroma vector
2. Quantisation of the amplitudes based on logarithmic amplitude thresholds
3. Smoothing with sliding window

Additionally, In our experiments we smoothed the features over a length of 21 (CENS 21) and 41 (CENS 41) windows.

3.2 Subsequence DTW

As opposed to the comparison of two different performances of the same piece, where it is assumed that the two follow the same score with different timings and articulation, in our case we compare the exact same music recorded under different conditions. Thus, we can assume a relatively constant slope close to 1 for the warping path. However, due to the nature of audio material on the LMA, we found the standard DTW obeying the conditions described in Section 2.3 to be unsuitable for our alignment task. In our work we base the alignment algorithm on the subsequence DTW (S-DTW), where successive DTW steps are applied on the data to identify matching sequences of acoustic frames [19]. We evaluate the individual matching paths

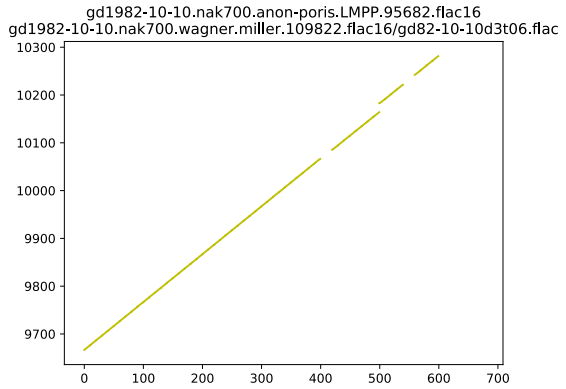

Fig. 1: Combined S-DTW warping path indicating skipping audio at around 500s on the x-axis.

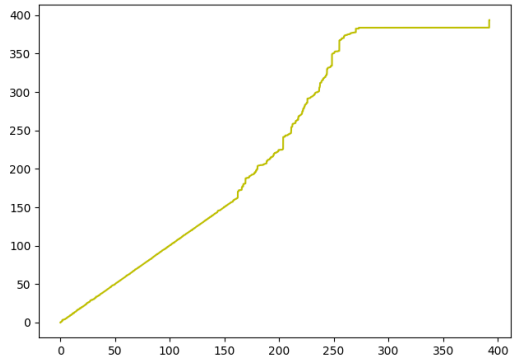

Fig. 2: DTW producing unrealistic warping path with noisy audio.

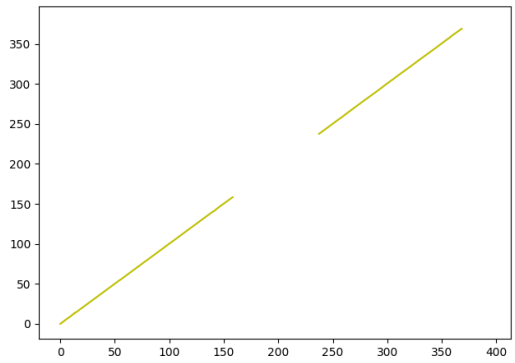

Fig. 3: Modified S-DTW warping path omitting invalid subsequence paths.

using linear regression and omit paths where either $1.05 > slope > 0.95$ or $R^2 < 0.99$, where the tolerance allows for local errors and time variations. When a segment is matched we first compare the next segment to a shorter segment of the reference following the matched time, which speeds up the process considerably.

The different recordings of a concert in the LMA vary in completeness, for instance, a recording may not contain all the songs, or pauses between songs are cut while they are included in others. Moreover, the concerts are segmented into different tracks with inconsistent boundaries. We may also encounter recordings that omit sections within one track, e.g. when a tape came to and end in the middle of a song recording resulting in skipping the part of the song for the duration in took to replace the tape. Such a case is illustrated in Figure 1, where skipping occurs around 500 on the x-axis. Since incomplete sequences are not anticipated in classic DTW, we cannot assume that every element of the sequences to be compared can be matched as dictated by the boundary and step-size conditions (eq. 1 and 3).

Live music recorded in the audience may contain parts that are predominantly crowd noise, resulting in audio features not to be found in the reference recording. Furthermore, analog tapes may have degraded over time leading to sections of heavy distortion. These factors make it difficult to obtain the correct warping path with classic DTW, resulting in the matching of short features from one time series with long features of the second time-series. The monotonicity condition (eq. 2) of the DTW prevents the warping path turning back on itself , thus, the optimal match for later sections may not be found. Figure 2 shows the warping path computed with standard DTW of two different recordings of the same performance, one exhibiting a section of added noise from approximately 160s to 240s into the signal. In this example, as the spectral differences increase, the path follows an unrealistic direction to such an extent that the later parts of the sequences are not optimally matched. Figure 3 shows the combined warping path computed by the successive application of the S-DTW with sections of 10s length. Warping paths that are not within the slope constraints computed with linear regression are omitted, resulting in a gap of the path signifying large spectral differences between the two sequences. The later parts of the sequences are matched correctly.

3.3 Resampling

As a first step prior to the feature extraction we resample the audio material to compensate for varying tape speeds during the recording and the transfer or digitisation of the audio material. The speed difference between the recordings is calculated from the estimated tuning difference computed by comparing the chroma features of a segment of the reference input with rotated chroma features of a segment of the recording to be aligned[3]. We also estimate the actual tuning of the reference audio[4] and adjust the pitch classes in the computation of the chroma features for the DTW accordingly.

4 Evaluation

In the following we test the our S-DTW algorithm under different conditions. We evaluate the performance using feature vectors constructed with different chroma variants and subsequence lengths of 10s and 20s.

4.1 Synthetic Dataset

The artificial dataset is based on a soundboard recording from the LMA[5]. First, we constructed a 10min version, downsampled to 22.05kHz, of the concert by concatenating 1min segments reflecting typical musical content from the bands performances. This includes sections of pauses between songs that includes tuning and percussive music. To simulate recordings we applied different audio effects to the recording to investigate how the different effects impact the S-DTW. The effects include added row noise, lowpass filter (4-pole), highpass filter (4-pole), reverberation , wow and flutter. For the reverberation we used the *IR_GreatHall* impulse response from the Audio Degradation Toolbox by Mauch and Ewert [20] and controlled the level of the non-direct sound. The wow and flutter effects are also taken from the Audio Degradation toolbox, while the filters are implemented with *scipy* versions of *pydub*[6]. We created 200 versions per effect applying the transformations with incremental parameter settings. The cutoff frequencies are increased on a logarithmic scale. The effects and parameter ranges are given in Table 1. We also created a separate dataset of 500 versions applying all effects with randomised parameter settings within the specified ranges.

parameter	range
crowd noise	0 to 80 dB SNR
flutter (100Hz)	0 to 0.2 intensity
wow (4Hz)	0 to 0.2 intensity
lowpass filter	2 to 11.025kHz cutoff
highpass filter	20 to 1500Hz cutoff
reverberation	-80 to 0dB level

Table 1: Effects and parameter ranges for the creation of the synthetic dataset.

Figures 4 to 7 visualise the results for the different effects (20 degree polynomial fitting curve). The Y-axes show the total matched time for the 600s recording with the effect parameter setting given on the X-axes. The values are Figures for flutter and lowpass filtering are omitted since the effect did not impact the results, which indicates that we could downsample the audio further to speed up the algorithm. The graphs indicate that the DTW with the CQT-based chroma vector in most cases performs best. However, in the case of high reverberation levels the the CENS chroma proves to be more suitable. When interpreting the numbers for the total matched time, we have to consider that our algorithm omits all subsequence warping paths that do not meet the restrictions described in Section 3. Therefore, when using 20s sequences instead of 10s sequences, each omitted path accounts for a larger chunk of the audio material.

For the real-world application of algorithm the total matched time is not the only factor to consider. For instance, a small number of matched paths per track may be enough for the alignment of the complete track, while tracks with no subsequence match or false positives introduce additional problems leading to ambiguity when matching a complete concert segmented into a number of tracks. The audio data processed with reverberation and highpass filter led to false positives (fpos) for subsequence warping paths and entire audio files for which no match could be identified (no match). The results are given in Table 2. Here, the CQT-based S-DTW produced the highest number of false positives. The CENS-based S-DTW with subsequences of 20s length produced the most unmatched items (items with no valid partial warping path).

[3] https://github.com/cannam/tuning-difference

[4] http://www.isophonics.net/nnls-chroma

[5] https://archive.org/details/gd1982-10-10.sbd.fixed.miller.110784.flac16

[6] https://github.com/jiaaro/pydub

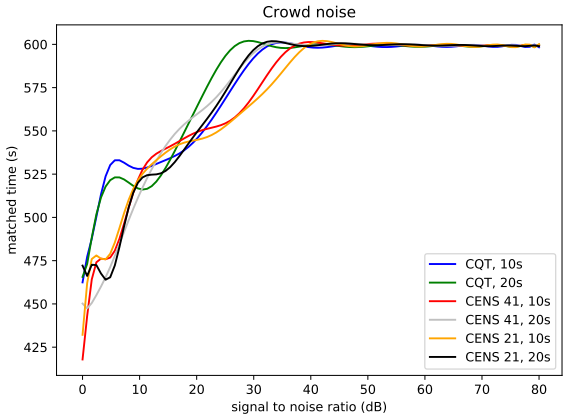

Fig. 4: Alignment performance on audio with added crowd noise with decreasing signal-to-noise ratio.

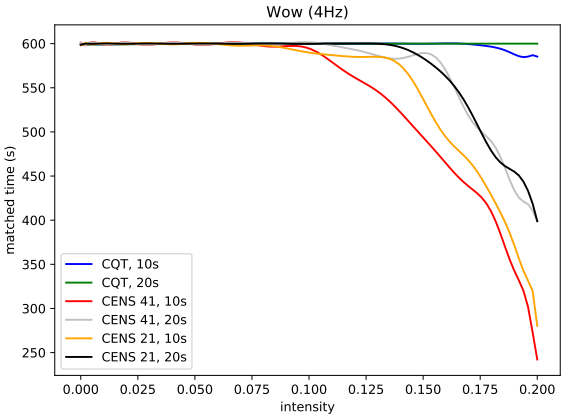

Fig. 5: Alignment performance on audio with applied wow effect with increasing intensity.

	highpass		reverberation	
chroma type	fpos	no match	fpos	no match
chroma CQT 10s	30	3.6%	1	0%
chroma CQT 20s	44	5.6%	9	0%
CENS 41 10s	15	4.2%	0	0%
CENS 41 20s	1	6.2%	0	0%
CENS 21 10s	14	5.0%	0	0%
CENS 21 20s	5	5.8%	0	0%

Table 2: False positive sequences (fpos) and tracks without any subsequence match (no match) under highpass and reverberation conditions using different chroma vectors.

chroma type	matched time	fpos	no match
chroma CQT 10s	37.4%	125	37.0%
chroma CQT 20s	40.3%	124	36.8%
CENS 21 10s	31.9%	8	37.8%
CENS 21 20s	35.5%	8	41.8%
CENS 41 10s	30.8%	9	36.8%
CENS 41 20s	37.0%	5	40.2%

Table 3: S-DTW performance for the synthetic dataset with random parameter settings.

4.2 Real-World Data

We applied the our algorithm to the Grateful Dead concert of October 10^{th} 1982. The recordings available in the LMA for this date are given in Table 4. We choose the longest recording as the reference against which the tracks of the other recordings are compared. In the experiment we used 10s segments and constructed feature vectors with the CQT-based chroma and CENS with 21 and 41 window smoothing. While we found larger tuning differences during initial exploratory experiments with concert recordings from the LMA, the recordings in this experiment varied between -6 and 14 cents compared to the reference. As opposed to the artificial dataset which does not exhibit any playback speed differences, for the real-world examples we resampled the audio material as described in Section 3.3.

Table 5 shows the combined matched time and the number of tracks with no valid subsequence for all recordings using 10s segments in the S-DTW. The total number of tracks compared with the reference is 249. It should be noted that the unmatched tracks under "no match" songs also include songs that cannot be matched because they are not present in reference. For instance some audience recordings contain a separate first track containing the tuning of instruments by the band not present in the reference soundboard recording. Nevertheless, the results show that the CENS 21 chroma based DWT produces the fewest unmatched items, while performing better than CENS 41 with respect to total matched time. The experiment further reveals that the S-DTW performs better on the real-world data than on our synthetic dataset. This maybe explained with the fact that the intensity of the audio degradation in the synthetic dataset is often greater than

etree ID	length	tracks	type
gd1982-10-10.nak700.anon-poris.LMPP.95682.flac16	3h:06m:41s	26	AUD
gd1982-10-10.nak700.wagner.miller.109822.flac16	3h:03m:57s	26	AUD
gd1982-10-10.sonyecm220T.keshavan.miller.93732.sbeok.flac16	3h:03m:12s	26	AUD
gd1982-10-10.beyers.stankiewicz.128808.flac16	3h:03m:08s	26	AUD
gd1982-10-10.123624.senn421.gans.miller.flac16	3h:03m:01s	25	AUD
gd1982-10-10.sony-ecm220.keshavan.tzuriel.29315.sbeok.flac16	3h:01m:40s	23	AUD
gd1982-10-10.aud.keshavan.bertha.77325.sbeok.flac16	3h:01m:10s	23	AUD
gd1982-10-10.111039.nak300.hoey.flac16	2h:59m:41s	24	AUD
gd1982-10-10.sbd.fixed.miller.110784.flac16	2h:58m:37s	25	SBD
gd1982-10-10.sbd.tetzeli.ayers.79903.sbeok.flac16	2h:58m:26s	26	SBD
gd1982-10-10.sbd.miller.110626.flac16	2h:58m:13s	25	SBD

Table 4: Audience (AUD) and soundboard (SBD) recordings of the Grateful Dead concert, October 10 1982.

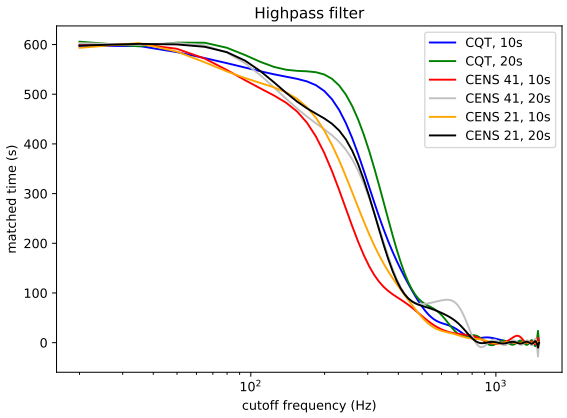

Fig. 6: Alignment performance on audio with applied high pass filter with increasing cutoff frequency.

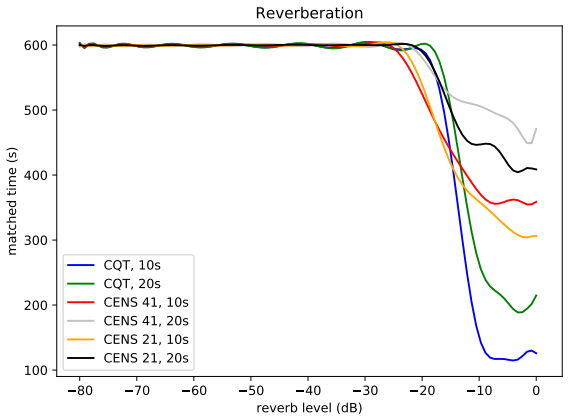

Fig. 7: Alignment performance on audio with applied reverberation with increasing reverberation level.

what we encounter in the recordings of the LMA.

Since a low number of unmatched tracks is a crucial factor in the alignment of the complete recording, we use CENS 21 feature vectors for the timeline construction described in Section 5 .

5 Timeline Construction

Our next task is to find a consensus between the local pairwise alignments obtained as described above. The goal is to construct a reference timeline along which we can situate all the time points occurring in the set of analysed recordings. After applying the process described in Section 3.2 to N recordings $R_0, \ldots, R_N$ including reference recording R_0, each recording R_i being composed of a sequence of n_i tracks $r_0^i, \ldots, r_{n_i}^i$, we obtain for each aligned track r_j^i a sequence of $k_{i,j}$ continuous aligned segments $s_0^{i,j}, \ldots, s_{k_{i,j}}^{i,j}$ consisting of pairs of time points $\in R_i \times R_0$, typically over a length of a multiple of 10 seconds.

While the points in each of the segments $s_0^{i,j}, \ldots, s_{k_{i,j}}^{i,j}$ for track r_j^i are distributed very closely to a line of slope 1 (Section 3.2), they may have different intercepts, for example due to a cut in either of the recordings R_i or R_0, as in the alignment shown in Figure 1. We thus first use another application of linear regression to split every track's segments into partitions of adjacent segments with the same intercept. This can be done using a recursive algorithm, each application of which di-

chroma type	matched time	no match
CENS 21	83.4%	2.4%
CENS 41	75.4%	2.8%
chroma CQT	88.6%	5.2%

Table 5: S-DTW performance (10s subsequences) for the concert recordings.

vides a given sequence of segments $s = (s_0, \ldots, s_k)$ into two partitions $p_0 = (s_0, \ldots, s_d)$ and $p_1 = (s_d, \ldots, s_k)$ for which the linear regression r value is maximal, i.e. $\arg\max_{0 \leq d \leq k}(max(rval(p_0), rval(p_1)))$. The recursion's base case is reached if no r value is greater than the one over all input segments s.

We then infer positions for all the parts of the recordings for which there is no match. First, we search for all subsequent pairs of partitions between which there is a gap, i.e. all pairs p with last time point (x_1, y_1) and q with first time point (x_2, y_2) where $x_1 < x_2$, and append a point $(x_2, y_1 + (x_2 - x_1)/2)$ to the last segment of p and prepend a point $(x_2, y_2 - (x_2 - x_1)/2)$ to the first segment of q, assuming p and q to continue on with slope 1 and the break point to be in the middle of the gap. Similarly, we fill the gaps before the first partition p_f and after the last partition p_l of each track with track length l by prepending a point $(0, y_1 - x_1)$ to the first segment of p_f, (x_1, y_1) being the previous first point of p_f, and by appending a point $(l, y_2 + (l - x_2))$ to the last segment of p_l, (x_2, y_2) being the previous last point of p_l.

Finally, we map all points $\in R_i \times R_0$ to a new space $R_i \times T$ where T is a new global reference timeline. This can be done satisfactorily using a simple algorithm that iterates through all subsequent pairs of partitions p, q of each recording and finds all places where p and q overlap. First, for each point in $R_i \times R_0$ with $i \neq 0$ we initialise a corresponding point in $R_i \times T$ and for each track in R_0 we initialise segments with two points $((t_s, t_s), (t_e, t_e))$ in $R_0 \times T$, one for the start point t_s and one for the end point t_e of each track.

Then, for each overlapping $p = (\ldots, (\ldots, (x_1, y_1)))$ and $q = (((x_2, y_2), \ldots), \ldots)$ with $x_2 < x_1$ we push back all partitions of the current recording with first time point $\geq x_2$ which includes q by adding $x_1 - x_2$ to their positions in T. We do the same for all partitions in other recordings including R_0 whose average position between first and last point is $> x_2$, i.e. the majority of the partition occurs after x_2.

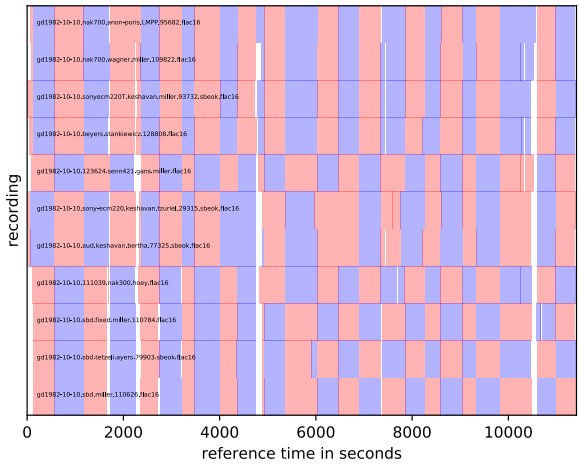

Fig. 8: The recordings from Table 4 situated along the constructed reference timeline. Odd-numbered tracks red, even-numbered tracks blue, gaps between tracks or track partitions white.

Figure 8 visualises the positions of all partitions of the recordings in our example real-world data set from Section 4.2 along the constructed reference timeline. One can identify three positions where a significant part was missing in R_0 resulting in a gap in the first row. Major gaps can also be observed occurring simultaneously in all recordings between the different sets of the concert and before the encore section (last two songs). Furthermore, one can see the different ways in which recordings are partitioned into tracks, e.g. two songs are merged into one at two different locations in the *keshavan.tzuriel* and *keshavan.bertha* recordings, visible as large blue and red areas around 4000 and 6000 seconds. Many recordings also have short inserted segments which often include parts where the band is tuning instruments or addressing the audience.

Note that this algorithm cannot guarantee that all pairs of points between tracks be aligned, for which we would need to calculate further pairwise alignments between pairs of tracks where overlaps occur and between overlapping tracks and recordings other than reference recording R_0 which may contain relevant audio material that may be missing in R_0. Also, tracks that are completely displaced or mislabeled may not be discovered since nonaligned tracks are simply prepended or appended in their respective recordings. Again, additional alignments could solve this problem.

However, if one is ready to accept these deficiencies,

this procedure is highly efficient for real world situations due to the fact that every recording only needs to be aligned once with the reference recording R_0. It can thus be applied to large collections of audio recordings if they are decently well organised and labeled.

6 Conclusions and Future Work

In this paper we have shown that we can align different digitised analogue recordings of the same concert using the modified S-DTW described in this work. In future work we aim to optimise several parts of the process. For instance, the application of the S-DTW can be optimised by not aligning all tracks with entire concerts, but pairwise with individual tracks in an iterative way, which would significantly speed up the procedure and reduce the memory load. The robustness against lowpass filtering indicates that we can furher downsample the audio material which would further lower the computational cost.

The process of identifying connected aligned parts with the same intercept can be simplified and done in one step instead of first identifying short continuous segments (Section 3.2) and then merging them into partitions (Section 5). Moreover, adding additional steps of comparison between the different recordings other than the reference recording can increase the precision in the construction of the timeline. In order to improve the alignment accuracy beyond the window length of the feature analysis Subramanian and Lerch [15] additionally applies cross-correlation over a short segment around detected overlaps.

Moreover, a formal evaluation of the proposed timeline construction algorithm can be conducted with a synthetic dataset of segmented audio material. Finally, we will integrate an audio player for the playback of synchronised recordings in the Web application for the exploration of Grateful Dead concerts extending the work described in [21, 22].

Acknowledgments

This paper is supported by EPSRC Grant EP/ L019981/1, Fusing Audio and Semantic Technologies for Intelligent Music Production and Consumption. Mark B. Sandler acknowledges the support of the Royal Society as a recipient of a Wolfson Research Merit Award.

References

[1] Wilmering, T., Thalmann, F., and Sandler, M. B., "Grateful Live: Mixing Multiple Recordings of a Dead Performance into an Immersive Experience," in *Proceedings of the Audio Engineering Society Convention 141*, 2016.

[2] Van Bogart, J. W., *Magnetic Tape Storage and Handling: A Guide for Libraries and Archives*, The Commission on Preservation and Access, Washington, DC and National Media Laboratory, St. Paul, MN, 1995.

[3] Anderton, C., "Beating the Bootleggers: Fan Creativity, 'Lossless' Audio Trading, and Commercial Opportunities," in M. D. Ayers, editor, *Cybersounds. Essays on Virtual Music Culture*, Peter Lang Publishing, 2006.

[4] Wilmering, T., Fazekas, G., Dixon, S., Page, K., and Bechhofer, S., "Towards High Level Feature Extraction from Large Live Music Recording Archives," *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML), Lille, France*, 2015.

[5] Sturm, B., "An Analysis of the GTZAN Music Genre Dataset." *Proceedings of the 2nd international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, 2012.

[6] Benson, M., *Why the Grateful Dead Matter*, ForeEdge Press, 2016.

[7] Kammerl, J., Birkbeck, N., Inguva, S., Kelly, D., Crawford, A. J., Denman, H., Kokaram, A., and Pantofaru, C., "Temporal synchronization of multiple audio signals," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014.

[8] Casanovas, A. L. and Cavallaro, A., "Audio-visual events for multi-camera synchronization," *Multimedia Tools and Applications*, 74(4), 2015.

[9] Bano, S. and Cavallaro, A., "Discovery and organization of multi-camera user-generated videos of the same event," *Information Sciences*, 302, 2015.

[10] Basaran, D., Cemgil, A. T., and Anarim, E., "Multiresolution alignment for multiple unsynchronized audio sequences using sequential Monte Carlo samplers," *Digital Signal Processing*, 77, 2018.

[11] Bryan, N. J., Smaragdis, P., and Mysore, G. J., "Clustering and Synchronizing Multi-camera Video via Landmark Cross-Correlation," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[12] Mendes C. Segundo, R., de Amorim, M. N., and Santos, C. A. S., "Automatic Mashup Generation from Multiple-Camera Concert Recordings," *International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2017.

[13] Kennedy, L. and Naaman, M., "Less Talk, More Rock: Automated Organization of Community-Contributed Collections of Concert Videos," *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*, 2009.

[14] Wang, A., "An Industrial Strength Audio Search Algorithm," *International Society for Music Information Retrieval Conference (ISMIR), Washington, D.C., USA*, 2003.

[15] Subramanian, V. and Lerch, A., "Concert Stitch: Organization and Synchronization of Crowd Sourced Recordings." in *ISMIR*, 2018.

[16] Sakoe, H. and Chiba, S., "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1), 1978.

[17] Müller, M., Kurth, F., and Clausen, M., "Chroma-based Statistical Audio Features for Audio Matching," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005.

[18] Dixon, S. and Widmer, G., "MATCH: A Music Alignment Tool Chest," *International Society for Music Information Retrieval Conference (ISMIR)*, 2005.

[19] Müller, M., *Information Retrieval for Music and Motion*, Springer, 2007.

[20] Mauch, M. and Ewert, S., "The Audio Degradation Toolbox and Its Application to Robustness Evaluation," in *ISMIR*, 2013.

[21] Wilmering, T., Thalmann, F., and Sandler, M. B., "Exploration of Grateful Dead Concerts and Memorabilia on the Semantic Web," *17th International Semantic Web Conference (ISWC)*, 2018.

[22] Thalmann, F., Wilmering, T., and Sandler, M. B., "Cultural Heritage Documentation and Exploration of Live Music Events with Linked Data," *1st International Workshop on Semantic Applications for Audio and Music (SAAM '18)*, 2018.