CONCERT STITCH: ORGANIZATION AND SYNCHRONIZATION OF CROWD-SOURCED RECORDINGS

Vinod Subramanian
Center for Music Technology
Georgia Institute of Technology
vsubramanian32@gatech.edu

Alexander Lerch
Center for Music Technology
Georgia Institute of Technology
alexander.lerch@gatech.edu

ABSTRACT

The number of audience recordings of concerts on the internet has exploded with the advent of smartphones. This paper proposes a method to organize and align these recordings in order to create one or more complete renderings of the concert. The process comprises two steps: first, using audio fingerprints to represent the recordings, identify overlapping segments, and compute an approximate alignment using a modified Dynamic Time Warping (DTW) algorithm and second, applying a cross-correlation around the approximate alignment points in order to improve the accuracy of the alignment. The proposed method is compared to two baseline systems using approaches previously proposed for similar tasks. One baseline cross-correlates the audio fingerprints directly without DTW. The second baseline replaces the audio fingerprints with pitch chroma in the DTW algorithm. A new dataset annotating real-world data obtained from the Live Music Archive is presented and used for evaluation of the three systems.

1. INTRODUCTION

Crowd-sourcing is the concept of presenting a problem to a large group of people and utilizing the best combination of the solutions received [12]. Although a large group of people can be used to obtain data, the data needs to be organized and labeled in a logical way to be useful. For instance, there has been an explosion in the number of audio and video recordings available online in the last few years. For large events such as concerts, speeches, and sports events, there are many recordings of (parts of) the same event. These recordings, however, are not annotated in a way that would allow a reconstruction of the complete timeline of the event. The focus of this research is, therefore, on the automatic organization and synchronization of the multiple recordings available of the same event.

Marshall and Shipman [16] analyze the people's reasons for recording events and report personal memorabilia, sharing on social platforms, creation of remixes, and online republishing as the main reasons. This indicates that there is value attached to these recordings. Vihavainen et al. [24] showed in their work that a human-computer collaborative approach to remixing concerts is of interest to a concert audience. Although the subjects favored the manually edited concerts in this instance, it still emphasizes the value of recombining audience recordings

While recombining audience recordings creates a better audience experience beyond the concert, a tool for automatic concert "stitching", faces several challenges. For example, each recording will have different audio quality due to different recording devices, distance from the stage, local disturbances etc.

After meeting these challenges, the application of this research enables (a) improved audience experience through personalized, collaborative, or theme-driven reconstruction of the event thus creating a platform for derivative work, (b) analysis and improvement of stage setups by venues and performers through audience videos from a large variety of recording angles, and, more generally, (c) audio forensics to reconstruct a scene by synchronizing multiple recordings for surveillance and investigation.

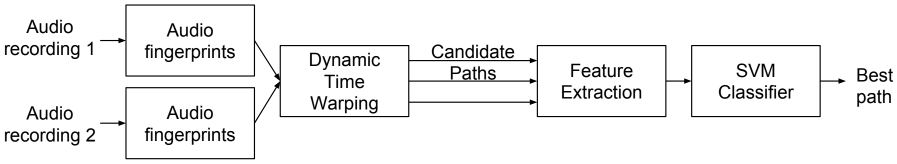
The goal of this study is to present a method that can (a) reliably identify if multiple recordings from an event have common audio content and (b) provide a precise alignment between all pairs of recordings. In the hope of encouraging more research on this task, we also present a new dataset for training and evaluation.

2. RELATED WORK

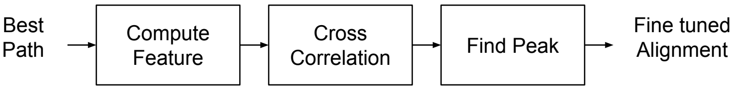
The task of aligning multiple recordings of an event can be divided into two steps: first, using a representation the recordings to identify overlapping segments, and compute an approximate alignment and second, applying a cross-correlation around the approximate alignment points in order to improve the accuracy of the alignment.

In tasks such as speech recognition [6,11] and music similarity [1,8], Mel-Frequency Cepstral Coefficients (MFCCs) are widely used to measure similarity between audio files. The Mel-Cepstrum captures timbral information and the spectral shape of the audio file [3]. However, MFCCs do not contain musically meaningful information such as melody or rhythm which could be argued to be crucial for computing music similarity.

Music Information Retrieval tasks such as cover song



(a) First part of algorithm for approximate alignment



(b) Second part of algorithm for fine tuning alignment using cross-correlation

Figure 1: Proposed method block diagram

detection [19], audio thumbnailing [2], and genre classification [23] use pitch-based features such as a pitch chroma to compute a measure of similarity. The pitch chroma [15] is an octave-invariant representation of the tonal content of an audio file and is usually computed in intervals of approx. 10 ms. A useful property of the pitch chroma is its robustness to timbre variations, allowing it to compare the pitch content of two different versions of the same song without being strongly influenced by timbre variations.

Determining the similarity of two recordings is closely related to audio fingerprinting, which aims at identifying a recording from a large database of recordings. An audio fingerprint is a highly compressed and unique representation of a (part of a) song [10, 25]. Wang [25] introduced an audio fingerprinting technique based on so-called landmarks. A landmark is identified as the spatial relationship of the salient spectral peaks. This representation is also used for the task of audio alignment of concert recordings [4, 13]. Most audio fingerprinting methods are temporally sensitive, meaning that they are not designed to handle variations in playback speed — a scenario that is likely in the case of analog recordings of concerts. The audio fingerprinting method introduced by Haitsma and Kalker calculates a 32 bit sub-fingerprint for every block of audio by looking at the energy differences along the frequency and time axes. This fingerprint method is used by Shrestha et al. [21] in their work on alignment of concert recordings. Alternately, Wilmering et al. [26] use high-level audio features such as tempo and chords in combination with low-level audio features such as MFCCs and pitch chroma to detect audio similarity for audio alignment of different versions of concerts.

Identifying and aligning overlapping segments requires the computation of a similarity or distance measure across a sequence of signal descriptors. One way of doing this is cross-correlation. Most of the research in aligning concert recordings apply this approach [4, 5, 13, 21, 22]. One constraint of cross-correlation is that the two sequences are assumed to be at the same speed. It is apparent that cross-correlation cannot be easily applied to the task of aligning analog recordings because there may be tempo variations and temporal fluctuations in the data. Another issue with cross-correlation is that a threshold needs to be set for what constitutes an alignment. To set the threshold some publications use heuristic methods based on their data [4, 5, 13, 22], while others [21] use a threshold determined by Haitsma and Kalker [10]. Using fixed thresholds bears the risk of errors when applying the system to unseen data.

Another method for computing overlaps is the use of Dynamic Time Warping (DTW), as it is able to handle temporal fluctuations between the signals [14]. Wilmering et al. apply DTW twice, the first time for aligning a recording to a reference audio file in order to identify the different playback speeds. Based on the result, the audio files are processed to mirror the playback speed of the reference. The second alignment is then applied to improve the accuracy of the first alignment. DTW is also used in the related task of sample detection, where it can help to identify the location of a sample in a song [9].

3. ALGORITHM DESCRIPTION

The first part of the algorithm, as shown in Figure 1a, computes audio fingerprints for each recording and uses these fingerprints to compute pairwise distance matrices. For each distance matrix, a DTW algorithm determines multiple possible path candidates representing the potentially overlapping region between that pair of recordings. For each of these candidates, features are extracted and an SVM classifier determines which path is the most likely. In the case that the pair is not overlapping, no path should be selected from the candidates. The second part of the algorithm as shown in Figure 1b takes the most likely path and computes a cross-correlation of the overlapping regions to determine the exact alignment of the pairs and to improve accuracy.

3.1 Audio Fingerprint Computation

The motivation for using audio fingerprints is that it is a representation of audio robust to noise and timbre [10,

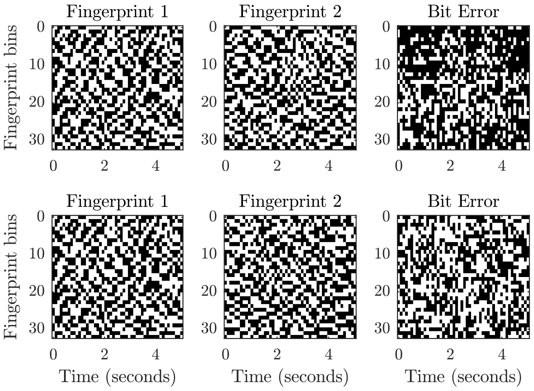


Figure 2: The top row shows the fingerprints from two recordings of the same 5 second snippet. The second row shows the fingerprints from two recordings of different 5 second snippets. For the Bit Error, the black regions indicate the fingerprints match and the white regions indicate the fingerprints are different.

25]. The audio fingerprinting technique utilized here is the Haitsma and Kalker algorithm [10]. The audio fingerprints are computed at a sampling rate of 5 kHz with a block size 2048 and a hop size of 512.

Figure 2 visualizes the robustness of audio fingerprints to noise distortion with an example. The upper row shows the bit error (in white) between the fingerprints of two matching but distorted recordings, the lower row shows the same for two different recordings. We can clearly see how the fingerprints retain the essential information even in the case of heavy distortion.

3.2 Modified Dynamic Time Warping

Dynamic Time Warping (DTW) is designed to align sequences with similar content but are temporally different. In the case of aligning concert recordings, the temporal fluctuations might occur due to inaccuracies in the sampling rate; in the case of analog recordings, the temporal fluctuations might be caused due to varying playback speeds.

The classical DTW algorithm introduced by Sakoe and Chiba works under the assumption that the start and end points of the two sequences are aligned [20]. A modification of the standard approach allows the algorithm to detect subsequences [18]; however, in the case of real life recordings, the most likely scenario is that a pair of recordings might have overlapping regions. Therefore, a pair of recordings will neither have the same start and end points, nor will one recording necessarily be a subsequence of the other. To address this issue, the subsequence DTW algorithm is modified to look for overlapping regions by doing the traceback from all possible end points.

The distance matrix is computed as the pairwise distance of two audio fingerprint matrices corresponding to two recordings. The dimension of one fingerprint matrix is $32 \times M$ and of the second is $32 \times N$ where M and N correspond to the number of blocks of audio that each recording was divided into. Using the Hamming distance, the result is

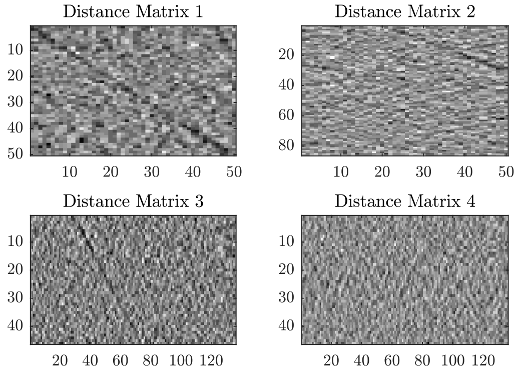


Figure 3: Distance matrix examples. The dark line indicates high similarity. For Distance matrix 4, there is no overlap, so there is no high similarity region

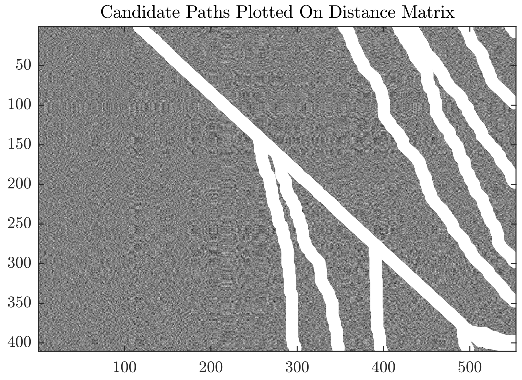


Figure 4: Different candidate path examples. The straightest line in the image represents the correct path.

a distance matrix D with the dimensions $M \times N$. Figure 3 shows examples of the distance matrix for different pairs of recordings; the top left matrix shows a standard DTW case with start and end points of both sequences aligned, the top right and bottom left are computed from pairs of recordings with overlapping regions and the bottom right matrix corresponds to a pair of recordings without overlapping regions.

A cost matrix is computed from the distance matrix as is done for the subsequence DTW algorithm [18]. In short, the initialization of the cost matrix computation is modified– as opposed to accumulating the distance across both the first row and first column, only the first column is accumulated.

We use the standard DTW technique to traceback the path; however, instead of doing this on just the minimum cost point, the traceback is performed on all possible path end points from the last row and last column. This results in multiple paths. Figure 4 illustrates a few paths that are computed for an example cost matrix.

3.3 Feature Extraction

To identify the most likely candidate path, we extract features from each path. Each possible path has three features:

(a) the DTW cost normalized by path length, (b) the slope of the line connecting the starting and ending points, and (c) the deviation of the path from the line connecting the start and end points. These paths are then clustered such that each cluster contains paths that share a start point; the end point for each cluster is the path with the lowest normalized cost. From each cluster, the minimum, mean, and standard deviation of the three path features are taken along with the number of paths in the cluster. These cluster features are similar to the ones proposed by Gururani and Lerch in the context of sample detection [9]. The extracted features per cluster have a dimensionality of 1×10 per cluster and are the input of a classifier estimating whether a path candidate represents a true overlap or not.

3.4 Classifier

A binary classifier is trained to determine which of the candidate paths is the most likely path for the alignment. A Support Vector Machine algorithm (SVM) with a linear kernel is used as this classifier. In the event that the classifier doesn't identify any of the candidate paths as a path for alignment, it is assumed that that pair of recordings do not have overlapping content. In the case of two or more paths being classified as true overlapping paths, the classifier's output probability is used to choose the most probable path.

3.5 Sample-Accurate Alignment

The audio fingerprinting technique used [10] downsamples the audio to 5000 Hz and blocks the audio by 1024 samples so the DTW alignment has a low resolution. As a more accurate result is desirable to reconstruct the timeline artifact-free (without 'jumps') when splicing two recordings together, a post-processing step is applied. One audio file is resampled based on the approximate alignment; then, the cross-correlation of overlapping regions of the pair of recordings is computed for 5 seconds around the detected start point. The result should then provide a synchronization point with improved accuracy.

3.6 Baseline

We compare the results of the proposed method to two baseline systems– one looking at the audio features and the other looking at the alignment stage.

3.6.1 Pitch chroma baseline

In order to investigate the effect of audio descriptors on the alignment accuracy, the pitch chroma is used as the audio representation instead of audio fingerprints. For the pitch chroma, a euclidean distance is used instead of the Hamming distance for calculating the distance matrix. Pitch chroma is a feature of interest as it is a typical feature used for audio similarity [2, 19, 23]. It has also been used in previous work on aligning concert recordings [26]. The pitch chroma is computed at a sampling rate of 11 kHz with a block size of 4096 and a hop size of 1024.

3.6.2 Cross-correlation baseline

The cross-correlation on audio fingerprints is the most established approach in the field of aligning noisy concert recordings [4, 13, 21]. For this process, the Hamming distance is computed at different levels of overlap and a threshold of 0.35 Bit Error Rate (BER) is set according to the recommendation by Haitsma and Kalker [10]. If the distance falls below the threshold then the pair of recordings are aligned at that overlap.

4. EXPERIMENTS

We run several experiments to investigate our algorithm. We evaluate the audio (feature) representation, approaches to alignment, and alignment accuracy.

4.1 Dataset

Two datasets are used in this study– a synthetic dataset and a real world dataset. The synthetic dataset created for simulating a real world scenario; the advantage is a sample-accurate ground truth. The synthetic dataset will be used as the training and validation set, as well as to provide some preliminary results with high accuracy. The real-world dataset is manually annotated from existing recordings and is used to test the overall performance of the algorithm.

4.1.1 Synthetic Dataset

The synthetic dataset is a collection of audio recordings downloaded from YouTube [1] consisting of live recordings of concerts. There are a 100 songs available in this dataset.

In order to create training data for the classifier, each song of the dataset is divided into 17 (can be varied) recordings with the constraint that each recording is longer than 20 s and the entire song is covered. Each recording is modified by (a) resampling randomly between 42.9 kHz to 45.2 kHz, (b) either low pass filtering with a cutoff between 5000 Hz to 11600 Hz or high pass filtering with a cutoff between 200 Hz to 5000 Hz, (c) adding crowd sounds obtained from freesound.org [7], and (d) adding distortion using the 'live recording' and 'smart phone recording' simulations in the audio degradation toolbox [17]. The code for generating the synthetic dataset is available online [2] .

4.1.2 Real World Dataset

The real world dataset consists of 5 audience recordings of a Grateful Dead concert performed on 1977-05-08. The audio data was obtained from the Live Music Archive [3] . The first 5 songs from the concert were selected and each of the 5 versions of the 5 songs were annotated. The annotations indicate the start and end points of the song. In case a part of the song is missing, the duration and location of the missing location is indicated. Since these recordings were made on analog devices, the data is prone to tempo and playback speed variation in addition to the usual filtering and distortion heard in audience recordings. The real world

[1] https://www.youtube.com/ accessed March 1st 2018

[2] https://github.com/VinodS7/ConcertStitch-dataset

	precision	recall	f-measure
Fingerprints	0.9697	0.6732	0.8145
Pitch Chroma	0.6753	0.3191	0.4335

Table 1: Experiment 1: Overlap detection for audio fingerprints vs. pitch chroma on real world data

dataset is augmented by splitting each version of a song into 10 recordings, resulting in 50 simulated audience recordings per song. The songs are split in the same way as for the synthetic dataset.

4.2 Metrics

There are two metrics used for the evaluation of this task. The first metric is using the precision, recall, and f-measure to provide an understanding of whether an alignment is correctly detected for a pair of recordings. The second metric is the statistical analysis of the alignment accuracy in seconds where the median, standard deviation, and maximum values are used to measure how accurate the alignment is.

4.3 Experiment 1: Audio fingerprints vs. pitch chroma

The aim of this experiment is to compare the audio representation on which the distance computation is based. We investigate audio fingerprints and pitch chroma for the task of aligning noisy recordings.

To train the SVM classifier for the algorithm, the abovementioned cluster features are extracted from the synthetic dataset for 25 songs. To extract the features for each pair of recordings, the DTW algorithm computes multiple possible paths corresponding to all unique starting points. All paths are labeled incorrect except the path that is closest to the ground truth in the case of overlapping recordings. The extracted feature matrix thus consists of the cluster features along with a label of whether those features correspond to an overlap or not. This process is applied to both audio fingerprints and pitch chromas.

Once the feature matrix is available, it is divided into an 80-20 split for training and validation, respectively. As each pair of recordings has multiple candidate paths with a maximum of only one being correct, there are far more negative observations in the feature matrix than positive observations. To counteract the high number of negative observations, the training data is sampled to reduce the number of negative observations. The ratio of negative to positive observations is 50:1 for the audio fingerprints classifier and 30:1 for the pitch chroma classifier.

The evaluation is performed on the real world dataset. Only the start points of the alignment are taken into account because the audio files are not modified or resampled based on the end points.

4.3.1 Results

Table 1 reports the precision, recall, and f-measure of the audio fingerprints and the pitch chroma. The fingerprint outperforms the pitch chroma considerably for all metrics. This result is expected as the fingerprint is specifically designed to work in conditions with severe quality impairments. The poor performance of the pitch chroma can be traced back to computing the candidate paths in the DTW algorithm. Due to the noise, the candidate paths frequently do not contain the correct path for the pitch chroma. This adversely affects the training process for the SVM classifier and subsequently the performance on the real world data.

Real World	precision	recall	f-measure
DTW	0.9697	0.6732	0.8145
cross-correlation	0.4132	0.2534	0.3141
Synthetic	precision	recall	f-measure
DTW	0.9570	0.9319	0.9443
cross-correlation	0.6936	0.8956	0.7818

Table 2: Experiment 2: DTW vs. cross-correlation using audio fingerprints for real world and synthetic data

4.4 Experiment 2: DTW vs. cross-correlation

The aim of this experiment is to compare the performance of the DTW and the cross-correlation techniques when audio fingerprints are used as the audio representation. The audio fingerprints for the cross-correlation method are almost the same as for the DTW algorithm, the only difference is that the hop size is now 64 instead of 512.

The classifier for the DTW algorithm is set up the same way as in Experiment 1. The evaluation is performed on both the synthetic dataset with no temporal fluctuations and the real world dataset.

4.4.1 Results

Table 2 reports the precision, recall, and f-measure of the DTW method and the cross-correlation method on the two datasets. We observe that the DTW method clearly outperforms the cross-correlation method. This is especially true for the real-world data because the DTW is designed to handle temporal fluctuations while cross-correlation is not. On the synthetic dataset containing no temporal fluctuations, the cross-correlation method performs much better; however, it still does not perform as well as the DTW method. One possible reason might be that the cross-correlation method uses a strict threshold to identify alignment so the cross-correlation method does not scale well to different types of noise.

4.5 Experiment 3: DTW performance analysis

The goal of this experiment is to understand the strengths and weaknesses of the proposed algorithm.

For the first part of the experiment, the precision, recall, and f-measure are reported for difference tolerance thresholds on the real world dataset. The tolerance threshold gives maximum allowable deviation of the alignment provided by the algorithm from the ground truth. If the alignment exceeds the threshold then it means the algorithm predicted the alignment incorrectly.

[3] https://archive.org/details/GratefulDead accessed January 15th 2018

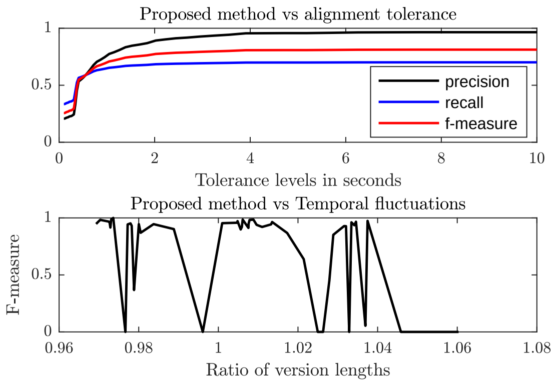


Figure 5: Experiment 3: Analyzing the performance of the proposed method

The second part of the experiment tests how robust to time stretching and pitch shifting the algorithm is. First, the sample rates for each of the recordings are identified using the same technique as Wilmering et al. [26]. Then, the alignment is calculated between each pair of recordings. Finally, the f-measure of the alignment is compared to the ratio of sample rates(or version lengths).

4.5.1 Results

The first part of Figure 5 shows the precision, recall, and f-measure at different tolerance thresholds. The plot shows that the performance decreases drastically for tolerances below 2 s. These results indicate a need to refine the alignment in order to provide a more accurate measure of alignment.

For the second part of Figure 5, we expect the algorithm to perform better if the ratio of lengths is closer to 1 and the performance to get worse the further away from 1. The reason is that if the ratio of sample rates is further away from one the pitch shifting becomes more significant which this algorithm is not designed to handle. However, the plot does not reflect this hypothesis because the pitch shifts in certain audio files is greater than expected. In addition to resampling there is more pitch shifting which causes the algorithm to fail since both the pitch chroma and fingerprints are sensitive to pitch shifting.

4.6 Experiment 4: Analysis of improved alignment accuracy

For a pair of recordings using the alignment, a resampling factor is calculated using a ratio of the length of the two paths. One recording is resampled so it has the same length as the other. We investigate and compare the spectral flux, spectral centroid, and time-domain raw audio for their ability to improve the alignment accuracy when cross-correlating a small segment around the previously estimated alignment points. For reference, the same features are computed without resampling the audio. The spectral flux and spectral centroid are calculated at a block size of 128 with a hop size of 32. The alignment accuracy for the raw audio, spectral flux, and spectral centroid for the original and resampled audio are compared against the original DTW algorithm to evaluate the accuracy improvement. The evaluation for this task is done on the synthetic dataset because the annotations are more accurate than for the real world data.

	median	std	max
DTW alignment	5240	11503	125221
Raw audio	9043	49091	823906
Spectral Flux	7073	12554	108950
Spectral Centroid	7161	9919	54695
Res. Raw Audio	5801	23185	227631
Res. Spec. Flux	5078	13092	125846
Res. Spec. Centroid	5006	11128	100165

Table 3: Raw Audio vs Spectral Flux to improve alignment accuracy. The results are reported as deviation in samples at 44.1 kHz

4.6.1 Results

The results of Experiment 4 are reported in Table 3. The numbers indicate how close to the ground truth alignment the algorithm performs in samples at a sample rate of 44100 Hz None of the finer alignment algorithms are able to significantly improve the alignment of the algorithm. However, it is important to note that by using the approximate alignment to resample the audio files, the results are much better than without resampling. One explanation for the limited improvement in performance is that the spectral centroid and spectral flux might not be too susceptible to noise.

5. CONCLUSION

This paper presented a method for accurately aligning recordings of a concert event given that these recordings are noisy snippets. The results show that audio fingerprints are better suited than pitch chroma for the task of representing noisy audio and that dynamic time warping performs better than cross-correlation for the alignment. Using a finer alignment on the resampled audio shows promise; however, the results are still unsatisfactory. The real world data has been made publicly available, and the used modifications of the data is published online [4] .

The biggest drawback of the algorithm is its inability to handle pitch shifts in audio recordings very well– a known issue with many fingerprinting systems. If the current audio fingerprinting algorithm is replaced with an algorithm that is robust to noise as well as to pitch shifts, we expect the performance of the system would improve considerably on our real world dataset.

Future work on this task will focus on the actual rendition of the complete event once the alignment is known and possibly combine audio with video. Selecting the segments, determining fade points, durations, and type in the overlapping regions, are all interesting and challenging tasks that have not been researched in depth yet.

[4] https://github.com/VinodS7/ConcertStitch-dataset

6. REFERENCES

[1] Jean-Julien Aucouturier, Francois Pachet, et al. Music similarity measures: What's the use? In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 13–17, 2002.

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

[3] A. Berenzweig, B. Logan, D. P. W. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, 2004.

[4] N. J. Bryan, P. Smaragdis, and G. J. Mysore. Clustering and synchronizing multi-camera video via landmark cross-correlation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2389–2392, 2012.

[5] C. V. Cotton and D. P. W. Ellis. Audio fingerprinting to identify multiple videos of an event. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2386–2389, March 2010.

[6] Steven B Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In *Readings in speech recognition*, pages 65–74. Elsevier, 1990.

[7] Frederic Font, Gerard Roma, and Xavier Serra. Freesound technical demo. In *ACM International Conference on Multimedia (MM'13)*, pages 411–412, Barcelona, Spain, 21/10/2013 2013.

[8] Jonathan T. Foote. Content-based retrieval of music and audio, 1997.

[9] Siddharth Gururani and Alexander Lerch. Automatic Sample Detection in Polyphonic Music. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, 2017.

[10] Jaap Haitsma and Ton Kalker. A highly robust audio fingerprinting system. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 107–115, 2002.

[11] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, Nov 2012.

[12] Jeff Howe. The rise of crowdsourcing. *Wired magazine*, 14:1–4, 2006.

[13] Lyndon Kennedy and Mor Naaman. Less talk, more rock: Automated organization of community-contributed collections of concert videos. In *Proc. of the 18th International Conference on World Wide Web*, pages 311–320, New York, 2009.

[14] Holger Kirchhoff and Alexander Lerch. Evaluation of Features for Audio-to-Audio Alignment. *Journal of New Music Research*, 40(1):27–41, 2011.

[15] Alexander Lerch. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. Wiley-IEEE Press, Hoboken, 2012.

[16] Catherine C. Marshall and Frank M. Shipman. Saving, reusing, and remixing web video: Using attitudes and practices to reveal social norms. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 885–896, New York, NY, USA, 2013.

[17] Matthias Mauch, Sebastian Ewert, et al. The audio degradation toolbox and its application to robustness evaluation. 2013.

[18] Meinard Müller. *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[19] S. Ravuri and D. P. W. Ellis. Cover song detection: From high scores to general classification. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 65–68, March 2010.

[20] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, Feb 1978.

[21] Prarthana Shrestha, Peter H.N. de With, Hans Weda, Mauro Barbieri, and Emile H.L. Aarts. Automatic mashup generation from multiple-camera concert recordings. In *Proc. of the 18th ACM International Conference on Multimedia*, pages 541–550, New York, 2010.

[22] Joren Six and Marc Leman. Synchronizing multimodal recordings using audio-to-audio alignment. *Journal on Multimodal User Interfaces*, 9(3):223–229, Sep 2015.

[23] George Tzanetakis, Andrey Ermolinskyi, and Perry Cook. Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, 32(2):143–152, 2003.

[24] Sami Vihavainen, Sujeet Mate, Lassi Seppälä, Francesco Cricri, and Igor D.D. Curcio. We want more: Human-computer collaboration in mobile social video remixing of music concerts. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 287–296, New York, 2011.

[25] Avery Wang. An industrial strength audio search algorithm. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, volume 2003, pages 7–13. Washington, D.C., 2003.

[26] Thomas Wilmering, Florian Thalmann, and Mark B. Sandler. Grateful live: Mixing multiple recordings of a dead performance into an immersive experience. In *Audio Engineering Society Convention 141*, 2016.