AudioAlign - Synchronization of A/V-Streams based on Audio Data

Mario Guggenberger, Mathias Lux, Laszlo Böszörmenyi
Institute of Information Technology
Alpen-Adria-Universität Klagenfurt
9020 Klagenfurt, Austria
{*mario.guggenberger,mathias.lux,laszlo.boeszoermenyi*}*@aau.at*

***Abstract*—Manual synchronization of audio and video recordings is a very annoying and time consuming task, especially if the tracks are very long and/or of large quantity. If the tracks aren't just short clips (of a few seconds or minutes) and recorded from heterogeneous sources, an additional problem comes into play - time drift - which arises if different recording devices aren't synchronized. This demo paper presents the experimental software *AudioAlign*, which aims to simplify the manual synchronization process with the ultimate goal to automate it altogether. It gives a short introduction to the topic, discusses the approach, method, implementation and preliminary results and gives an outlook at possible improvements.**

***Keywords*-audio; video; synchronization; alignment; fingerprinting; time-warping; clock drift; time drift;**

I. Introduction

The application area of this work are particular events like speeches or concerts, where the whole local area in which the event takes place is usually dominated by the same "audio signal". A common situation is that various persons are recording clips with their cameras and no matter where they point their lenses to, the video tracks might be different, but the audio tracks will roughly capture the same content. Settings all device clocks to exactly the same time and starting all recordings at exactly the same point would allow a simple synchronization by metadata timestamps and probably work for very short recordings, but as soon as they exceed a few minutes, another serious problem comes into play. Time drift, also called clock drift, resulting from a combination of jitter and production tolerance leads to the problem that the clocks of the recording devices, which control the analog-to-digital conversion, don't run at exactly the same speed. The implication is that one point in time isn't enough for a correct synchronization, and as this drift phenomenon doesn't necessarily need to be linear, not even synchronization at the start and end points of two recordings suffices. In the end, only professional equipment that is constantly synchronized by some sort of a central clock generator saves the manual work.

This work presents a software to align/synchronize multiple audio and video recordings from heterogeneous sources based on overlap in the corresponding audio streams, which eliminates the need of expensive professional hardware as usually used for multitrack recordings. The presented approach doesn't rely on metadata at all. It takes a number of files as input, analyzes their audio content and takes matches in content (detected overlaps) as synchronization points.

II. Related Work

Commercially, two very similar products are available from *Singular Software*[1]. *PluralEyes*, a plug-in for non-linear audio/video editing systems, which can synchronize an arbitrary number of tracks, but isn't able to detect or correct time drift, and *DualEyes*, a standalone tool which claims to be able to correct time drift but is specialized at the use-case of substituting audio tracks of video camera recordings with externally recorded audio. Some research papers have been published on this issue recently, e.g. in [1] the authors suggest to synchronize videos based on their audio tracks by computing the cross-correlation between pairs of tracks, which is, among other problems, much too slow due to the algorithm's complexity. More promising are the approaches in [2] and [3] which - similarly to our approach - use audio fingerprinting as a tool to find potential synchronization points, but they aren't addressing the issue of time drift. In summary, all of these approaches disregard least one of the aspects described above.

III. Synchronization Process

Our tool takes an arbitrary number of audio tracks as input, which can either be pure audio recordings or audio tracks of videos. The synchronization process starts by generating a fingerprint [4] sequence for each track and storing them into a database. Then this database is used to determine matching points between the tracks, which are tuples each containing references to two tracks and the corresponding time-based positions. Such a tuple translates to e.g. track *A* at position 13'24" contains the "same" aural information as track *B* at position 53'15", therefore this matching point can be used to synchronize the tracks at those positions. The minimum goal of this step is to find at least $n-1$ matches for n overlapping tracks, such that all tracks are interconnected by the matches in a way that a spanning tree can be built with tracks as vertices and matches as edges. Tracks that don't overlap with any other track should not

[1] http://www.singularsoftware.com

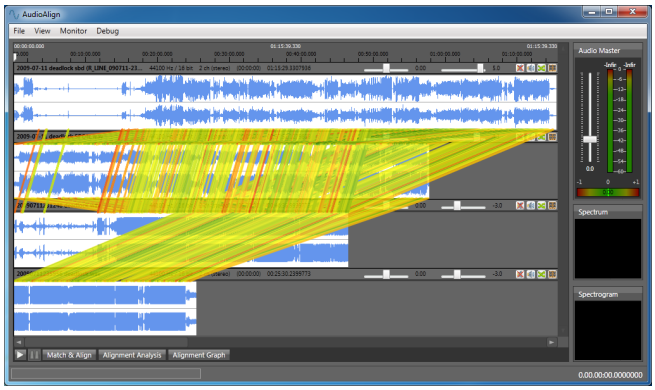

Figure 1. Screenshot of *AudioAlign* showing four tracks and their calculated matching points.

have any matches assigned. Next follows the only step where manual user interaction is required. The user needs to look at the matching points and decide whether they are correct (and optionally filter out false positives), whether time drift correction is required, and whether sufficiently many matches are available for compensating the drift. Ultimately this should be automated, and concepts for possible solutions have already been developed [5]. For the rare case that not a single match for a track has been found, the user can also manually add one or more. If the users decides that drift correction is required but not enough matches have been found, additional matches can be automatically calculated by the use of the dynamic time warping algorithm [6]. Finally, the synchronization can be finished by temporally arranging the tracks according to their matches and time stretching drifted tracks to achieve a common timebase.

IV. User Interface

The user interface of our tool (figure 1) is built around a typical multitrack timeline like software in this field is often offering. The timeline features (1) waveform rendering of the added tracks, (2) the possibility to freely drag and position the tracks, (3) seamless zooming down to single sample values, (4) adjust volume and balance, and (5) playback of the composition. This functionality makes it easy to judge the quality of the synchronization, determine if time drift occurs and do manual corrections if desired. Matching points can be optionally displayed as an overlay on the timeline, visualized as connection lines between the matching positions of the tracks (cp. figure 1, color indicates matching quality). The distribution of matching points and related offsets between tracks can also be visualized, making it even easier to manually detect possible drifts. An export of the timeline to *Sony Vegas Pro*[2] is also available, which allows further processing of the synchronized tracks in a number of ways.

[2]http://www.sonycreativesoftware.com/vegaspro

V. Experimental Results

A custom test set has been built to evaluate the accuracy of our approach. The set contains more than 250 recordings of 15 events with runtimes between a few seconds and 100 minutes, totalling at a length of more than 40 hours, from various sources like video and still cameras, mobile phones, TV broadcasts, audio recorders and internet sources like YouTube[3]. The results are very promising and the method even succeeds in situations where the audio quality is so bad that manual synchronization is getting hard for a human [5].

VI. Conclusion and Future Work

Tests have shown the applicability of our approach both in terms of accuracy and speed [5]. The tool can be applied to many real-world situations like synchronizing multitrack audio recordings made with cheap consumer devices or synchronizing multicamera video shots. It is also capable of synchronizing YouTube clips recorded at events like concerts and easily enables people to create long running continuous multicamera footage out of those clips. There's still room for improvement of the fingerprint matching and thus the synchronization rate by evaluating further algorithms and optimizations. Also we want to automate the detection of time drift to achieve a completely unsupervised synchronization process.

References

[1] N. Hasler, B. Rosenhahn, T. Thormahlen, M. Wand, J. Gall, and H.-P. Seidel, "Markerless motion capture with unsynchronized moving cameras," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 224–231.

[2] P. Shrstha, M. Barbieri, and H. Weda, "Synchronization of multi-camera video recordings based on audio," in *Proceedings of the 15th international conference on Multimedia*, ser. MULTIMEDIA '07. New York, NY, USA: ACM, 2007, pp. 545–548.

[3] L. Kennedy and M. Naaman, "Less talk, more rock: automated organization of community-contributed collections of concert videos," in *Proceedings of the 18th international conference on World wide web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 311–320.

[4] J. Haitsma and T. Kalker, "A highly robust audio fingerprinting system," in *ISMIR*, 2002. [Online]. Available: http://dblp.uni-trier.de/db/conf/ismir/ismir2002.html#HaitsmaK02

[5] M. Guggenberger, "Synchronisation von Multimediadaten auf Basis von Audiospuren," Master's thesis, Alpen-Adria-Universität Klagenfurt, Austria, 2012.

[6] S. Dixon, "An on-line time warping algorithm for tracking musical performances," in *Proceedings of the 19th international joint conference on Artificial intelligence*, ser. IJCAI'05. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005, pp. 1727–1728.

[3]http://www.youtube.com