LANDMARK-BASED SYNCHRONIZATION FOR DRIFTING AUDIO RECORDINGS

Technical Report

Manu Harju, Annamaria Mesaros

Signal Processing Research Centre, Tampere University, Tampere, Finland

ABSTRACT

This technical report describes our submissions to the multichannel alignment task in the BioDCASE 2025 Challenge. Our system is based on matching and aligning audio landmarks, which are simple structures extracted from the spectrogram representations. Our code and the configuration used is available on GitHub [1].

***Index Terms*—** STFT, audio landmark, audio fingerprint, audio syncronization

1. AUDIO LANDMARKS

Audio landmarks were originally developed for finding audio tracks from a large database. The idea is to generate an acoustic fingerprint of audio, which can be encoded as simple integers. In the end the whole problem of searching audio tracks becomes just hash matching [1].

The original method for audio landmarks uses STFT magnitude spectrogram as the input. First, the method extracts local peaks from the spectral representation, and then combines two closely placed peaks to landmarks. The second peak must reside in a certain target area of the first peak. A landmark is a quartet $t; (f_1, f_2, \tau)$, where t is the time step of the first peak, f_1 and f_2 are the frequency channels of the peaks, and τ is the time difference between the peaks. The peaks can be chosen so that the time difference is always a non-negative number, and generally the whole landmark can be coded in a relatively small number of bits [1]. A Python implementation of the method is Audfprint, available online [2].

The landmarks form an acoustic fingerprint that can be used to recognize the audio from a noisy input signal [1].

In this work, we use the landmarks to find the corresponding locations of two different signals recorded in the same scene. Although the spectral representations of the two recordings may differ quite significantly from each other, the peaks and their relations should be relatively recognizable, as the signals still have common audio events.

2. METHOD

Our method for synchronizing two audio signals is as follows.

1. Extract audio landmarks from both channels
2. Match landmarks between the channels taking into account the maximum time shift
3. Filter the matching landmarks with a moving window
4. Fit a polynomial to the filtered time shifts

The first step is to find the landmarks of the input signals. We establish our method on Audfprint [2]. However, instead of using STFT magnitudes, we use mel spectrograms to further reduce the number of frequency bins. The second step is to find matching pairs of landmarks in both signals. We introduce parameters to allow small changes in the landmarks, as a drifting clock can introduce pitch shifting. Furthermore, we only match landmarks that are within the given maximum desynchronization range, which was stated to be five seconds [3]. However, because the matching procedure also creates erroneous matches, after finding matching landmarks we filter the matches within a moving window. The aim of this filtering procedure is to produce a single pair of timestamps to denote the shift between the signals within one window. There are various possible filtering strategies, for example simply taking median, mean or mode of the shifts. We use mode if the mode count is at least 3, otherwise we revert to median. Finally, we fit a polynomial to the windowed timestamp pairs to smoothen the output and to be able to output the shift at any given time point. Based on experiments with the training data, we use a degree 3 polynomial for the final predictions.

parameter	ARU	Zebra Finch
F_S	16 kHz	32 kHz
FFT len	256	512
FFT hop	128	256
n_mels	32	48
density	80.0	80.0
max Δc	0	30
max $\Delta \tau$	1	5
win len	1200	400
win hop	400	200

Table 1: Used parameters for both datasets.

3. RESULTS

We tested different parameter configurations on the provided training data [3] to find the most suitable parameter values for our method. However, the changes in the system performance are very nonlinear with respect to the parameters, and the number of combinations is too large to test. We perform bootstrap sampling of individual file errors and report the averages of the bootstrap distributions with 95 % confidence intervals.

The two datasets in the task data are quite different from each other; we individually optimize the parameters for each dataset. The list of parameters is shown in Table 1. The "density" regulates peak

This work was supported by Academy of Finland grant 332063 "Teaching machines to listen".

[1] https://github.com/mnuhurr/biodcase2025-task1-landmarks

Model	Dataset	MSE	MAE
No sync	ARU	0.97 (0.61 - 1.68)	0.80 (0.59 - 1.06)
Baseline DL	ARU	0.80 (0.59 - 1.10)	0.59 (0.51 - 0.68)
Landmarks	ARU	0.41 (0.23 - 0.71)	0.48 (0.33 - 0.63)
No sync	Zebra Finch	1.31 (0.71 - 2.49)	0.89 (0.62 - 1.28)
Baseline DL	Zebra Finch	1.97 (1.54 - 2.55)	0.88 (0.81 - 0.98)
Landmarks	Zebra Finch	1.27 (0.74 - 2.15)	0.86 (0.63 - 1.20)

Table 2: Results of the proposed method on the provided development validation set. The estimates are averages of the bootstrap distribution, with confidence intervals for 95 % confidence level.

generation, and "max Δ" values limit how much difference is allowed in the matching of the landmarks: Δc denotes the allowed amount of shifting the landmark in frequency bands, and $\Delta\tau$ denotes the change of time difference between the peaks.

We evaluate the systems on the provided validation set, and use bootstrap [4] to estimate confidence intervals from the errors on individual files. The numbers are presented in Table 2. For comparison we also include the errors for no syncing and the baseline deep learning model [5] which was trained ten times to get some insight of the variance in the final models.

4. CONCLUSIONS

This technical report presents our submission to BioDCASE 2025 Challenge Task 1. The proposed system uses a simple yet effective approach to synchronize two audio signals. The method is based on matching acoustic landmarks between the signals, which gives a robust and effective way to find matching segments.

5. REFERENCES

[1] A. L.-C. Wang, "An industrial-strength audio search algorithm," in *Proc. ISMIR*, 2003.

[2] D. Ellis. Audfprint. [Online]. Available: https://github.com/dpwe/audfprint

[3] B. Heath, L. Gill, B. Hoffman, and G. Narula, "BioDCASE 2025 task 1: Multichannel alignment development and evaluation set," May 2025. [Online]. Available: https://doi.org/10.5281/zenodo.15492592

[4] B. Efron, "Bootstrap Methods: Another Look at the Jackknife," *Annals Statist.*, vol. 7, no. 1, pp. 1–26, 1979.

[5] B. Hoffman, L. Gill, B. Heath, and G. Narula, "Baseline system for BioDCASE 2025: Task 1," Apr. 2025. [Online]. Available: https://github.com/earthspecies/biodcase_2025_task1_public