THE 2014 LABROSA AUDIO FINGERPRINT SYSTEM

Daniel P. W. Ellis
LabROSA, Columbia University
New York, USA
dpwe@ee.columbia.edu

ABSTRACT

For the first MIREX Audio Fingerprinting evaluation, we are submitting the current version of the audio fingerprint system `audfprint` that we have been developing since 2008. The system is based on finding and matching local "landmarks" (prominences) in a spectrogram representation as described by Wang [1].

1. AUDFPRINT

Our implementation of audio fingerprinting, `audfprint`, has been developed as a tool for using noise-robust audio fingerprinting in academic research projects since 2007 [2, 3]. It is closely based on the fingerprinting system of [1], and, briefly, works by recording the precise frequency bins and relative timing (but not the magnitudes) of nearby strong "onsets" in a spectrogram, then encoding these landmarks in pairs to facilitate efficient search for reference items matching a query. By recording only frequency and relative timing, the representation is robust to variations in channel caused by microphones, speakers, or room acoustics. By limiting the representation to the most prominent time-frequency peaks, the system maximizes its robustness to additive noise, since the highest energy peaks of the source material are the least likely to be obscured or distorted by added interference. Matching essentially counts the number of correctly-aligned landmarks shared between query and reference item. Since the prior probability of a random match of a single landmark is smaller than 1%, the probability of a false alarm falls rapidly as the number of matching landmarks increases, and is essentially zero for more than 10 matching landmarks over a query of 30 s. Thus, only a few percent of the reference landmarks need to be matched to achieve a confident, true identification, further increasing the robustness to noise and channel distortion. Figure 1 illustrates the raw landmark pairs, and the matching landmarks, for a short, noisy query matched against a reference item.

`audfprint` was originally developed in Matlab [4], but the current system is implemented in Python and is freely available [5].

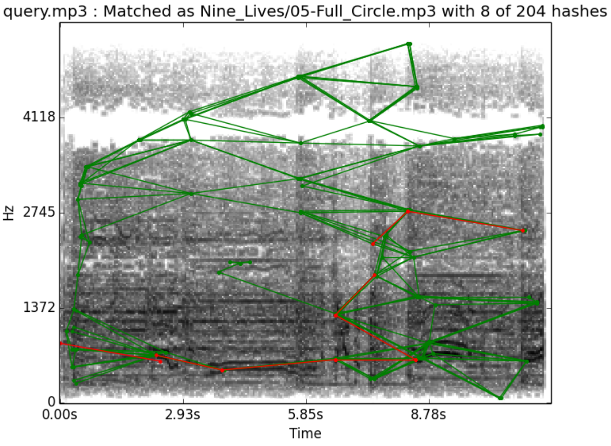

Figure 1. Example illustration of landmark matching from `audfprint`. Each line connects a pair of landmarks extracted from the query; red lines indicate pairs matched to reference item.

2. CONFIGURATION

`audfprint` provides a number of configuration options. The "density" option controls how rapidly the adaptive magnitude threshold decays, and hence how many landmarks are found; larger number of landmarks generally lead to greater matching accuracy (particularly for high noise and/or short duration queries), at the cost of larger reference databases and more computationally expensive matching. Another option is "fanout", which limits the number of nearby landmarks used to form pairs to store in the reference index. The default settings are density of 20 with fanout of 3, giving a reference database of around 8KB per minute of audio. As described below, for this evaluation we experimented with considerably higher values, to improve recognition accuracy in noise.

Another option enables the use of multiple cores, when available. "ncores" will parallelize both database building and matching, using Python's multiprocessing and joblib modules. As a result of the overhead associated with managing multiple threads, we find that using four cores leads to approximately doubling the speed of computation.

3. MIREX

The 2014 MIREX Audio Fingerprinting task [6] provided a collection of noisy development queries based on sam-

Configuration	Density	Fanout	Build time	Dbase size	Match time	Accuracy
tiny	20	3	0.80% RT	7.8 kB/min	1.0% RT	60.8%
low	50	6	0.97% RT	22.5 kB/min	2.0% RT	77.7%
main	70	8	1.07% RT	34.2 kB/min	3.4% RT	78.0%
high	100	10	1.19% RT	52.4 kB/min	7.8% RT	78.2%

Table 1. Parameters and performance of various configurations. Execution times are as a percentage of the audio duration (reference items for build, query items for match), and correspond to using 4 cores on a 12-core Xeon E5-2420 (1.9 GHz) machine (Dell R520).

ples from the GTZAN genre dataset [7]. For development, we built a reference dataset composed of the 1000 $\times$ 30 s clips in the dataset, ran each of the 1062 $\times$ 10 s queries from the `queryPublic_George` development set, and measured database size, execution times (for database building, and for querying), and final matching accuracy. These are shown in table 1 for four different configurations. "tiny" corresponds to the default settings of `audfprint`, which are intended for reasonable performance on relatively clean queries. "main" is our much denser configuration aimed to give good performance within the original constraints specified for the MIREX task (2% RT for reference database building, 50% RT for query matching, and 50 kB per minute of reference audio for the reference database). "low" and "high" are provided as contrast conditions to indicate how rapidly accuracy varies with resource consumption around this operating point; we see that for this query set, doubling the reference database size gives only slight improvement in accuracy.

4. CONCLUSIONS

We have described the audio fingerprinting system we submitted to MIREX 2014. The full code to run this system is available at `https://github.com/dpwe/audfprint/`.

5. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation (NSF) under Grant No. IIS-1117015. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

6. REFERENCES

[1] A. Wang, "The Shazam music recognition service," *Comm. ACM*, vol. 49, no. 8, pp. 44–48, Aug. 2006.

[2] J. Ogle and D. P. W. Ellis, "Fingerprinting to identify repeated sound events in long-duration personal audio recordings," in *Proc. ICASSP*, Hawai'i, 2007, pp. I–233–236. [Online]. Available: http://www.ee.columbia.edu/~dpwe/pubs/OgleE07-pershash.pdf

[3] C. Cotton and D. P. W. Ellis, "Audio fingerprinting to identify multiple videos of an event," in *Proc. IEEE ICASSP*, Dallas, 2010, pp. 2386–2389. [Online]. Available: http://www.ee.columbia.edu/~dpwe/pubs/CottonE10-fingerprint.pdf

[4] D. Ellis, "Audfprint - audio fingerprint database creation + query," web resource, Dec 2011. [Online]. Available: http://labrosa.ee.columbia.edu/matlab/audfprint/

[5] ——, "`audfprint`: Audio landmark-based fingerprinting," web resource, Jun 2014. [Online]. Available: https://github.com/dpwe/audfprint

[6] C.-C. Wang and J.-S. R. Jang, "MIREX 2014: Audio fingerprinting," web resource, Jul 2014. [Online]. Available: http://www.music-ir.org/mirex/wiki/2014:Audio_Fingerprinting

[7] G. Tzanetakis, "Gtzan genre collection," web resource, 2001. [Online]. Available: http://marsyas.info/download/data_sets/