



# Panako: a scalable audio search system

Joren Six <sup>1</sup>

1 IPEM, Ghent University, Belgium

DOI: [10.21105/joss.04554](https://doi.org/10.21105/joss.04554)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Brian McFee](#)  

## Reviewers:

- [@dpwe](#)
- [@hendriks73](#)

Submitted: 01 July 2022

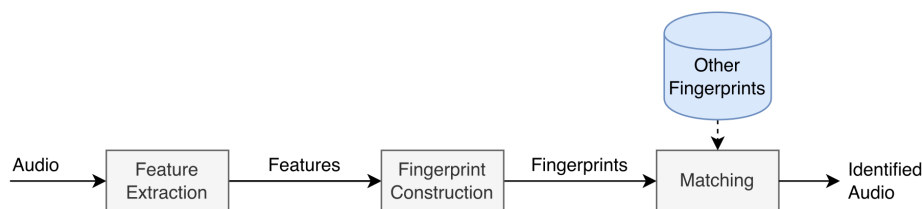
Published: 11 October 2022

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

## Summary

Panako solves the problem of finding short audio fragments in large digital audio archives. The content based audio search algorithm implemented in Panako is able to identify a short audio query in a large database of thousands of hours of audio using an acoustic fingerprinting technique.



**Figure 1:** A general acoustic fingerprinting system. Features are extracted from audio and combined into fingerprints. The fingerprints are matched with fingerprints in a reference database. Finally a match is reported.

The dataflow in Panako closely resembles the flow depicted in [Figure 1](#). Audio is transformed to features which are grouped in recognizable fingerprints. The fingerprints are compared with a database of reference fingerprints. If a match is found, it is reported or, in case of a true negative, the system reports that the audio is not present in the database. The properties of the acoustic fingerprinting system mainly depend on the selection of features, the information captured by the fingerprints and the performance of the matching step.

There are three algorithms implemented in Panako. The first is based on pitch class histograms [Six & Cornelis \(2012\)](#). The other two are based on peaks in a spectral representation: a baseline algorithm ([Six, 2020](#); [Avery L. Wang, 2003](#)) and the main Panako algorithm ([Six & Leman, 2014](#); [Six, Joren, 2021](#); [A. Li-chun Wang & Culbert, 2003](#)).

The main algorithm in Panako finds matches between a short query fragment and the audio in the database even if the query is time-stretched, pitch-shifted, slowed down or sped up while maintaining other desirable features such as scalability, robustness and reliability. This is important since changes in replay speed do occur commonly, they are either introduced by accident during an analog to digital conversion or are introduced deliberately.

Accidental audio speed changes are often the result of varying or uncalibrated recording or playback speeds of analogue physical media such as wax cylinders, wire recordings, magnetic tapes or gramophone records. To identify duplicates in a digitized archive, a music search algorithm should be robust against replay speed. Deliberate speed manipulations are sometimes introduced during radio broadcasts: occasionally songs are played a bit faster to fit into a time-slot. During a DJ-set speed changes are almost always present. To correctly identify audio in these cases as well, a music search algorithm must be robust against pitch shifting, time stretching and speed changes.

## Statement of need

Audio search algorithms have been described for decades (Cano et al., 2005; Fenet et al., 2011; Haitsma & Kalker, 2002; Herre et al., 2002; Sonnleitner & Widmer, 2014; Avery L. Wang, 2003) but have not been accessible for researchers due to the lack of proper, scalable, freely available implementations. Panako solves this problem by providing an acoustic fingerprinting system which can be used by researchers for DJ-set analysis, digital music archive management and audio-to-audio alignment.

In DJ-set analysis the aim is to automatically identify music in sets and how that music was modified and mixed (Kim et al., 2020; Schwarz & Fourer, 2018; Sonnleitner et al., 2016). With its robustness for time-stretch/pitch-shift and speed-up Panako is ideally suited to gather large scale insights in DJ performance practice.

Six et al. (2018) describes the applications of acoustic fingerprints for digital music archive management. These range from meta-data quality verification - through the identification of duplicates - to merging archives with potential duplicate material.

A less straightforward application of Panako is audio-to-audio alignment and synchronization (Six & Leman, 2015, 2017). In that case the matching fingerprints are used to align e.g. multiple video recordings of the same event by aligning the audio attached to each video.

Alternative systems with available implementations are by Chang et al. (2021) and audfprint by Ellis (2014). Both systems however lack robustness to significant speed changes of more than 5%. Note that there are two implementations of the audfprint system: there is a MatLab<sup>1</sup> and Python<sup>2</sup> version. The matlab version has support for small amounts time scaling. The details on how Panako handles time-scaling are described in two papers (Six & Leman, 2014; Six, Joren, 2021).

## Design

Simplicity and maintainability are two keywords in the design of Panako. The code aims to be as readable and simple as possible. The second version of Panako is a complete rewrite to ensure this simplicity while still keeping query and computational performance in check.

Relying on conservative platforms with a long history of backwards compatibility should allow Panako to stand the test of time. Panako is developed in Java and targets the long term support release Java SE 11. Panako also relies on software in C and C++. Java, C and C++ have been around for decades and it is reasonable to assume that these platforms will be supported for decades to come. Boring technology enables longevity.

Next to Java 11, Panako depends on three libraries: a DSP library, a key-value store and a spectral transform library. The first is a pure Java DSP library called TarsosDSP<sup>3</sup> (Six et al., 2014). LMDB<sup>4</sup> is used as a high performance key-value store. It is a C library and accessible through lmdbjava. The third and final dependency is JGaborator<sup>5</sup>: a wrapper around the Gaborator<sup>6</sup> library which implements a constant-Q non-stationary Gabor transform in C++11 (Velasco, Gino Angelo and Holighaus, Nicki and Dörfler, Monika and Grill, Thomas, 2011). The last two have native compiled parts and need to be ported to new or exotic platforms if the need arises. The transition to aarch64 (Apple M1), for example consisted of a straightforward compilation step and repackaging of this native library. Panako can be

<sup>1</sup><https://www.ee.columbia.edu/~dpwe/resources/matlab/audfprint/>

<sup>2</sup><https://github.com/dpwe/audfprint>

<sup>3</sup><https://github.com/JorenSix/TarsosDSP>

<sup>4</sup><https://www.symas.com/lmdb>

<sup>5</sup><https://github.com/JorenSix/JGaborator>

<sup>6</sup><https://gaborator.com>

containerized and the Docker file supports both ARM and x86 platforms and always compiles these native dependencies.

The code of Panako is hosted in a publicly available GitHub repository. Internal documentation follows the JavaDoc standards. Two papers give the rationale behind the algorithms (Six & Leman, 2014; Six, Joren, 2021). Panako can be installed using a Gradle wrapper script which automatically downloads the Gradle build-system if it is not present on the system. The Gradle wrapper compiles and installs Panako.

## Acknowledgements

Development of Panako is partially funded by the Ghent University BOF Project PaPiOM.

## References

- Cano, P., Batlle, E., Kalker, T., & Haitsma, J. (2005). A review of audio fingerprinting. *The Journal of VLSI Signal Processing*, 41, 271–284. <https://doi.org/10.1007/s11265-005-4151-3>
- Chang, S., Lee, D., Park, J., Lim, H., Lee, K., Ko, K., & Han, Y. (2021). Neural audio fingerprint for high-specific audio retrieval based on contrastive learning. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3025–3029. <https://doi.org/10.1109/ICASSP39728.2021.9414337>
- Ellis, D. (2014). The 2014 labrosa audio fingerprint system. *MIREX abstracts of the 15th International Symposium on Music Information Retrieval (ISMIR 2014)*.
- Fenet, S., Richard, G., & Grenier, Y. (2011). A Scalable Audio Fingerprint Method with Robustness to Pitch-Shifting. *Proceedings of the 12th International Symposium on Music Information Retrieval (ISMIR 2011)*, 121–126. <https://doi.org/10.5281/zenodo.1417593>
- Haitsma, J., & Kalker, T. (2002). A highly robust audio fingerprinting system. *Proceedings of the 3th International Symposium on Music Information Retrieval (ISMIR 2002)*. <https://doi.org/10.5281/zenodo.1417973>
- Herre, J., Hellmuth, O., & Cremer, M. (2002). Scalable robust audio fingerprinting using MPEG-7 content description. *Multimedia Signal Processing, 2002 IEEE Workshop on*, 165–168. <https://doi.org/10.1109/MMSP.2002.1203273>
- Kim, T., Choi, M., Sacks, E., Yang, Y.-H., & Nam, J. (2020). A computational analysis of real-world DJ mixes using mix-to-track subsequence alignment. <https://doi.org/10.5281/zenodo.4245544>
- Schwarz, D., & Fourer, D. (2018). Unmixdb: A dataset for DJ-mix information retrieval. *Proceedings of the 19th International Symposium on Music Information Retrieval (ISMIR 2018)*. <https://doi.org/10.5281/zenodo.1422385>
- Six, J. (2020). OLAF: Overly lightweight acoustic fingerprinting. *Extended abstracts for the Late-Breaking Demo Session of the 21st International Society for Music Information Conference (ISMIR 2020)*.
- Six, J., Bressan, F., & Leman, M. (2018). Applications of duplicate detection in music archives: From metadata comparison to storage optimisation. In G. Serra & C. Tasso (Eds.), *Digital libraries and multimedia archives* (pp. 101–113). Springer International Publishing. [https://doi.org/10.1007/978-3-319-73165-0\\_10](https://doi.org/10.1007/978-3-319-73165-0_10)
- Six, J., & Cornelis, O. (2012). A Robust Audio Fingerprinter Based on Pitch Class Histograms - Applications for Ethnic Music Archives. *Proceedings of the Folk Music Analysis conference (FMA 2012)*.

- Six, J., Cornelis, O., & Leman, M. (2014). TarsosDSP, a Real-Time Audio Processing Framework in Java. *Proceedings of the 53rd AES Conference (AES 53rd)*.
- Six, J., & Leman, M. (2014). Panako - A scalable acoustic fingerprinting system handling time-scale and pitch modification. *Proceedings of the 15th ISMIR Conference (ISMIR 2014)*, 1–6. <https://doi.org/10.5281/zenodo.1416190>
- Six, J., & Leman, M. (2015). Synchronizing multimodal recordings using audio-to-audio alignment. *Journal on Multimodal User Interfaces*, 9(3), 223–229. <https://doi.org/10.1007/s12193-015-0196-1>
- Six, J., & Leman, M. (2017). A framework to provide fine-grained time-dependent context for active listening experiences. *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio*.
- Six, Joren. (2021). Panako 2.0 : updates for an acoustic fingerprinting system. *Extended abstracts for the Late-Breaking Demo Session of the 22st International Society for Music Information Conference (ISMIR 2021)* .
- Sonnleitner, R., Arzt, A., & Widmer, G. (2016). Landmark-based audio fingerprinting for DJ mix monitoring. *Proceedings of the 17th International Symposium on Music Information Retrieval (ISMIR 2016)*, 185–191. <https://doi.org/10.5281/zenodo.1417008>
- Sonnleitner, R., & Widmer, G. (2014). Quad-based Audio Fingerprinting Robust To Time And Frequency Scaling. *Proceedings of the 17th International Conference on Digital Audio Effects (DAFx-14)*.
- Velasco, Gino Angelo and Holighaus, Nicki and Dörfler, Monika and Grill, Thomas. (2011). Constructing an invertible constant-q transform with non-stationary gabor frames. *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, 33.
- Wang, Avery L. (2003). An Industrial-Strength Audio Search Algorithm. *Proceedings of the 4th International Symposium on Music Information Retrieval (ISMIR 2003)*, 7–13. <https://doi.org/10.5281/zenodo.1416340>
- Wang, A. Li-chun, & Culbert, D. (2003). Robust and invariant audio pattern matching (Patent No. US7627477 B). In *Patent US7627477 B (US7627477 B)*.