CERGY PARIS UNIVERSITÉ

Identification de titres musicaux

Méthode de fingerprinting

Othman EL HOUFI
Mohamed DIAWARA

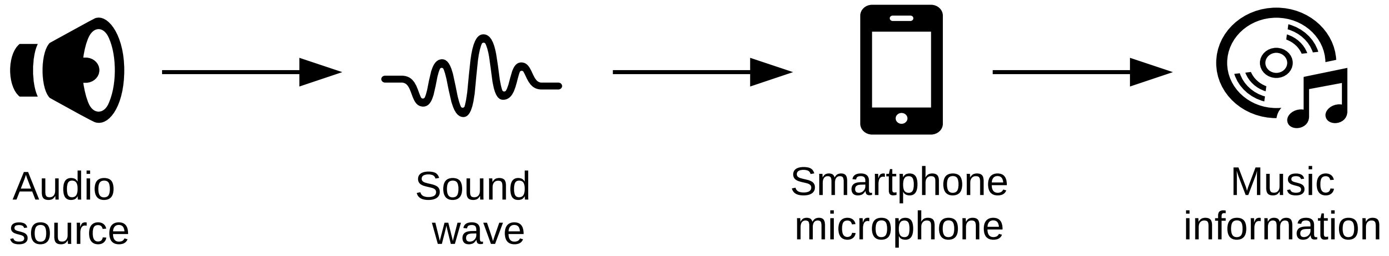
M1 Ingénierie des Systèmes Complexes et Intelligents

Tuteur technique
Dimitris KOTZINOS

Encadrant de Gestion de Projet
Tianxiao LIU

Goal

Identify music titles using a microphone.



Problems

How can we compare two sound tracks ?

How can we be able to identify tracks in the presence of noise ?

How much time and memory we need in order to identify a track ?

Let’s answer these questions

The identification of music tracks can be divided in two separate operations

Signal Processing

- Here we look at sound as a Signal.
- We look for the right representation of this Signal.
- We identify the critical points that can be present in two tracks and make similitude comparison.
- We make sure that noise won’t stop us from identifying a track.

Database

- We find a way to store our critical points – A fingerprint.
- We make sure to search and find for matches in a reasonable time.
- We make sure that memory won’t be a problem – As in minimum memory use.

Acoustic Signal Processing

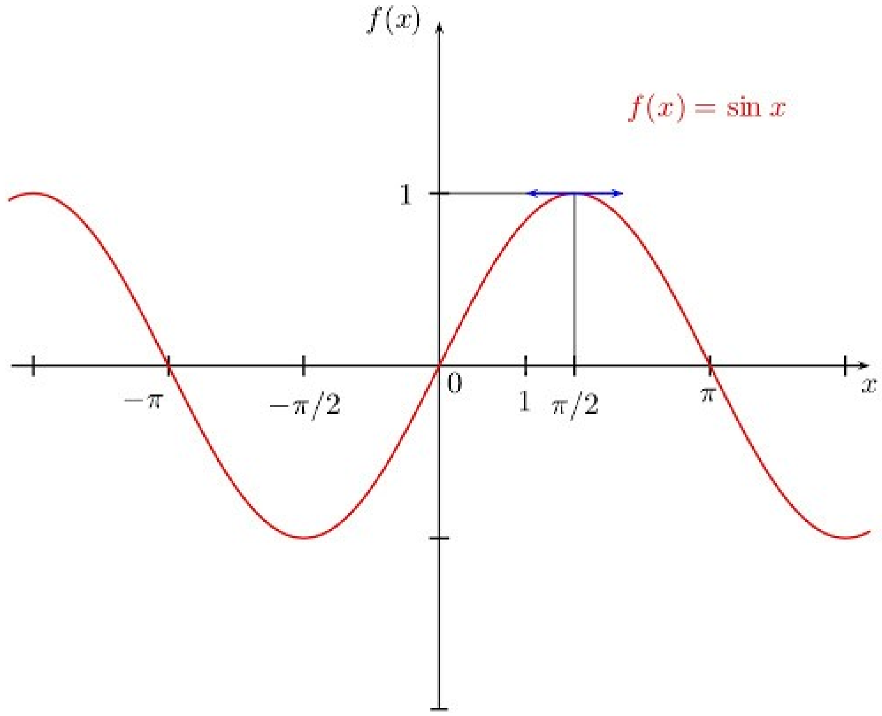
Acoustic Signal Representation

There are three main representations of a signal :

- Time representation,
- Frequency representation,
- Spectrogram representation,

Which representation is more adequate for our project ?

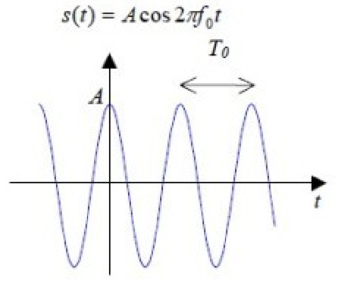
Acoustic Signal Representation

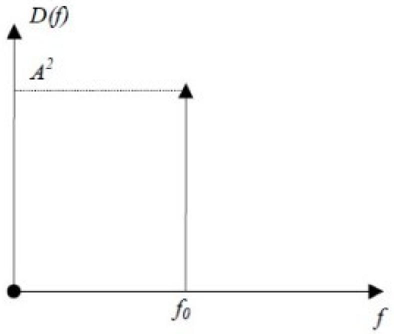


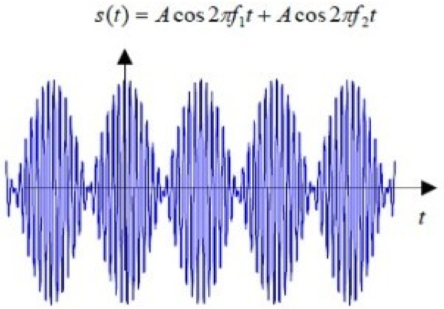
Time representation

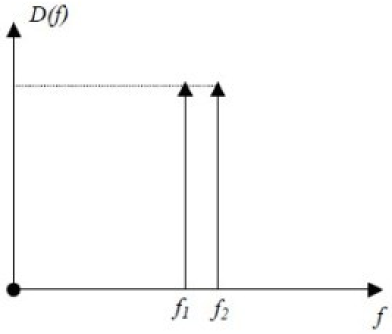
This representation gives a general idea about the amplitude variation over time. It's definitely not robust in case of noise and distortion.

Acoustic Signal Representation







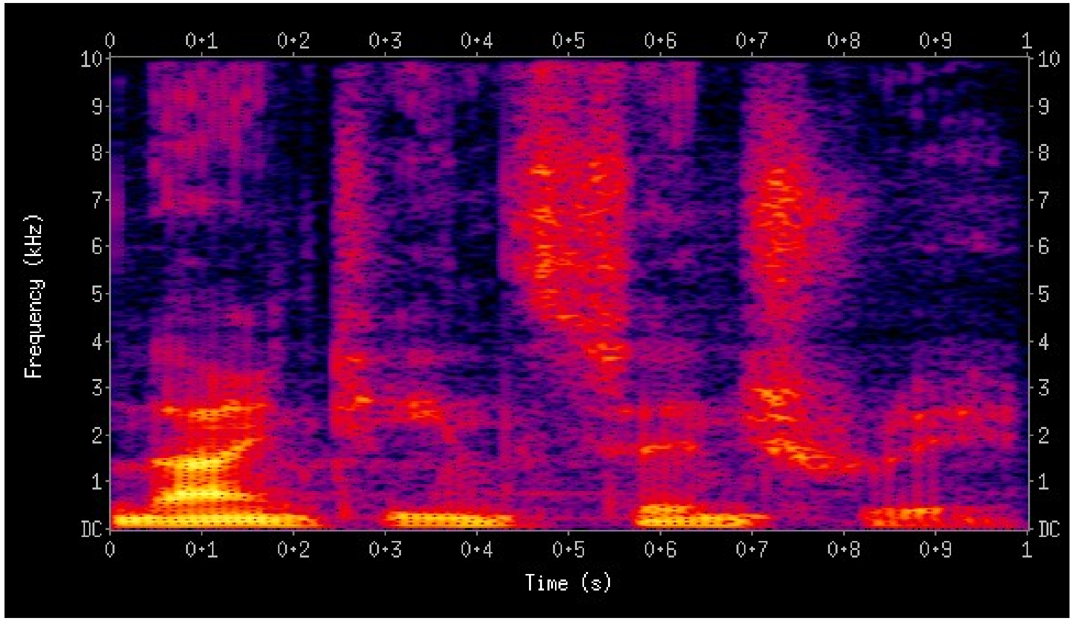


Frequency representation

This representation gives specific information about the signal frequencies.
It’s definitely not robust in case of noise and distortion.
It’s not robust in case of de-synchronization (time offsets) because we lost track of time .
It demands computational power.

$$X(f) = \int_{-\infty}^{\infty} x(t) \times e^{-i2\pi f t}\, dt$$

Acoustic Signal Representation



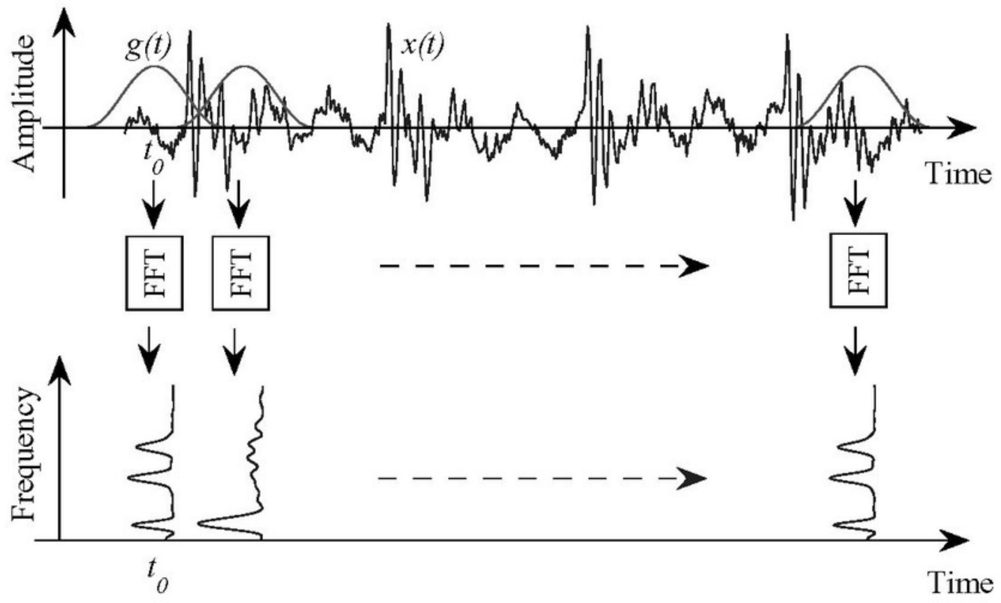
Spectrogram representation

This representation combines both frequencies and the time at each frequency occurs. It also gives information about the intensity of amplitudes.
It’s robust in case of noise and de-synchronization (time offsets) .

Acoustic Signal Representation

The Spectrogram representation is a result of a STFT (Short-time Fourier Transform) algorithm, which mean we use a filter or a window function for a time interval and we calculate the FFT of the signal at that interval.

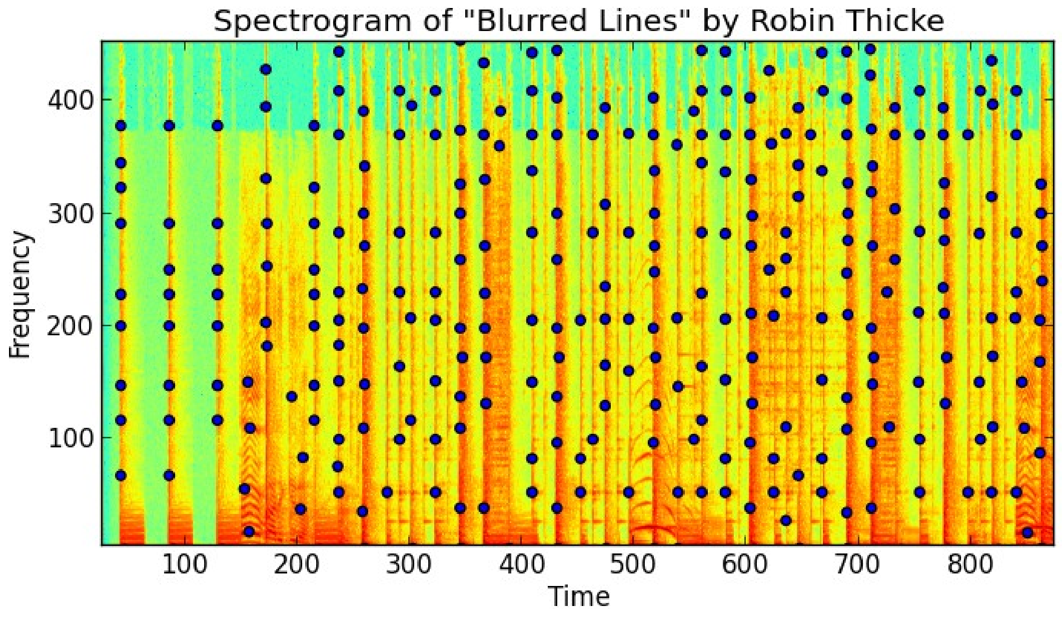
$$STFT = X(\tau, \omega) = \int_{-\infty}^{\infty} x(t) w(t-\tau) e^{-i\omega t} dt$$



Peaks Extraction

In order to address the problem of robust identification in the presence of highly significant noise and distortion, we smartly choose a set of points that are robust, as they are called Peaks.

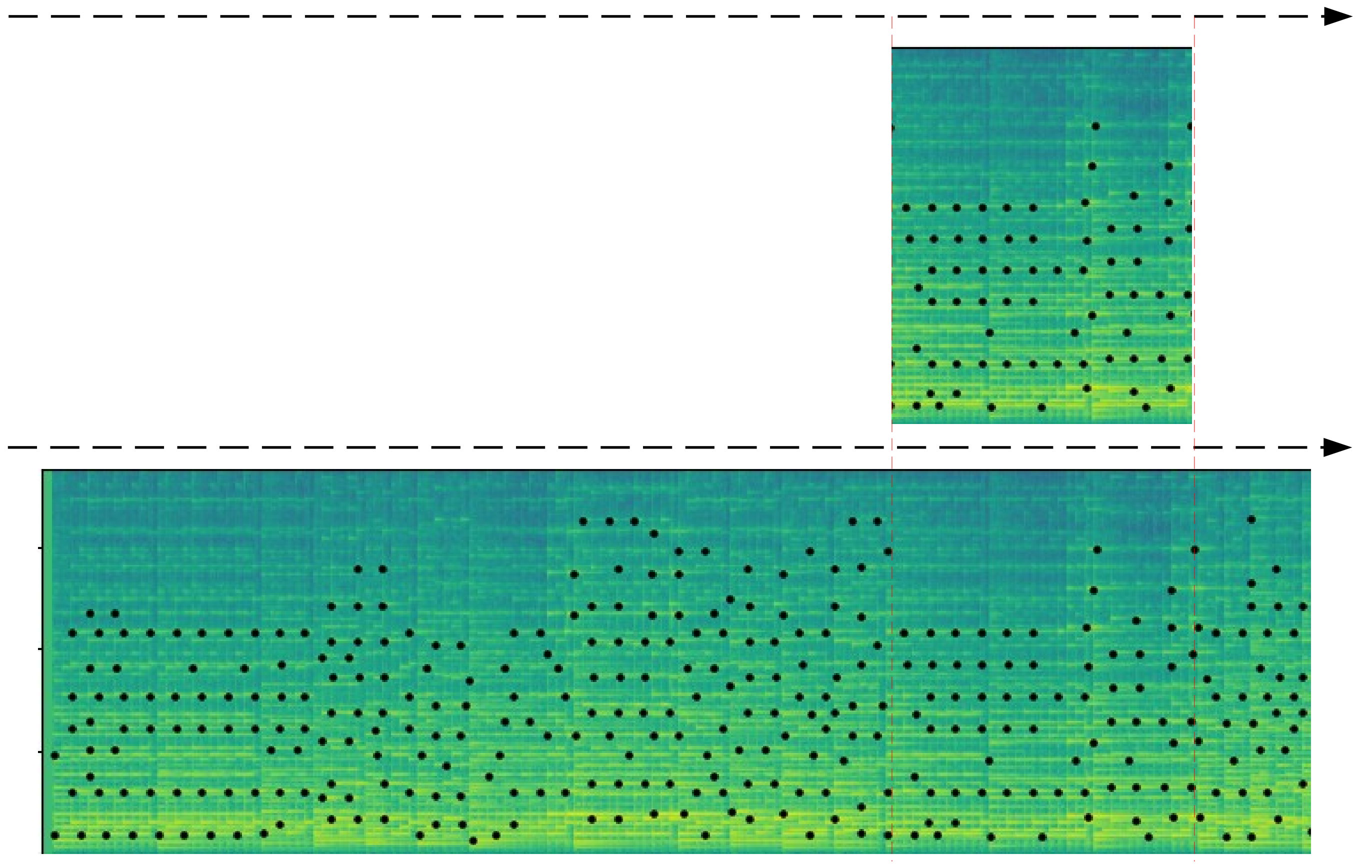
A time-frequency point is a candidate peak if it has a higher energy content than all its neighbors in a region centered around the point.



Peaks Extraction

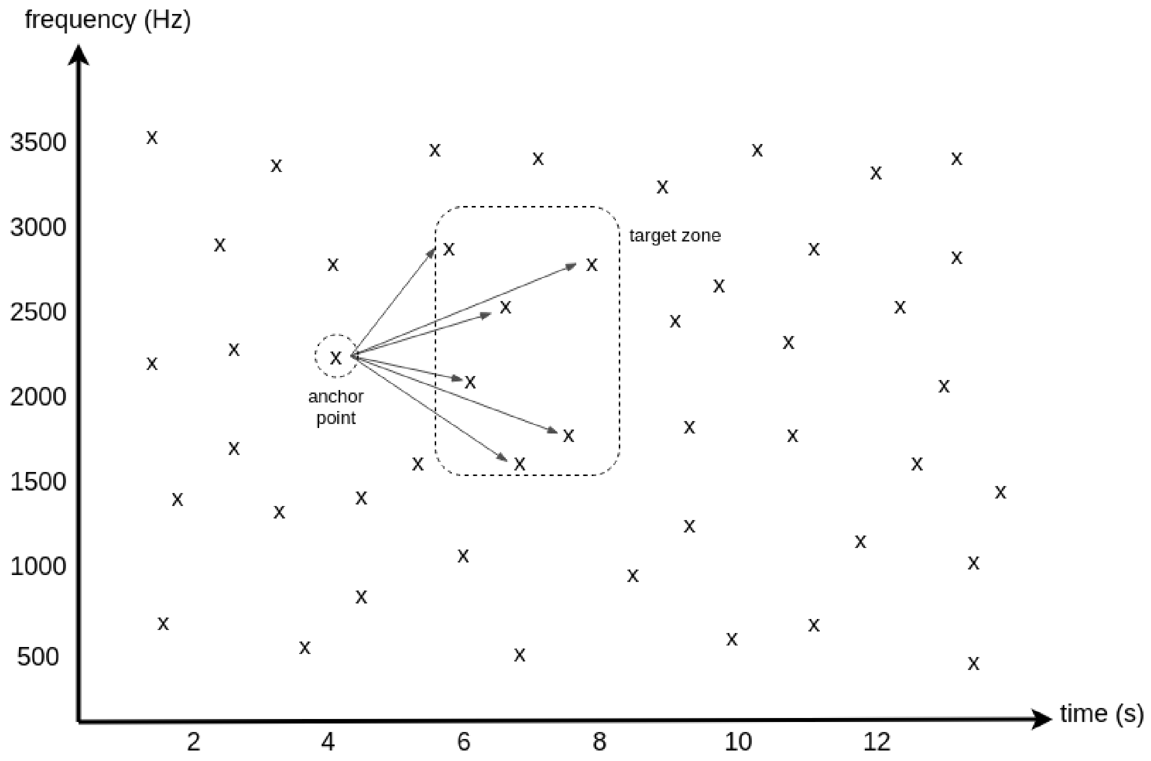
The pattern of dots should be the same for matching segments of audio. If you put the constellation map of a database song on a strip chart, and the constellation map of a short matching audio sample of a few seconds length on a transparent piece of plastic, then slide the latter over the former, at some point a significant number of points will coincide when the proper time offset is located and the two constellation maps are aligned in register.

Peaks Extraction



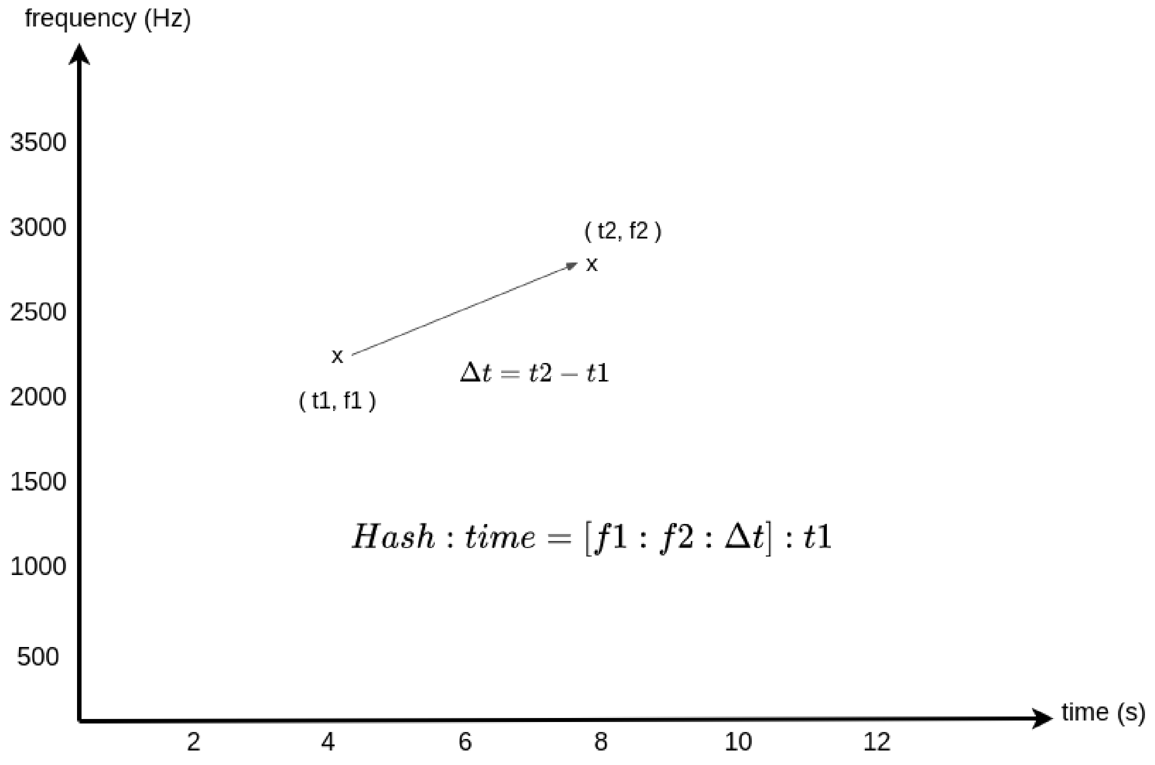
Database

Fast Combinatorial Hashing



- Fingerprint hashes are formed from the constellation map, in which time-frequency point pairs are combinatorially associated.

Fast Combinatorial Hashing



- Each hash is also associated with the time offset between the beginning of the respective file and its anchor point.

Fast Combinatorial Hashing

hash(frequencies of peaks, time difference between peaks) = fingerprint hash value

- These hashes are quite reproducible, even in the presence of noise and voice codec compression.
- Each hash is also associated with the time offset from the beginning of the respective file to its anchor point.
- By forming pairs instead of searching for matches against individual constellation points we gain a tremendous acceleration in the search process.

Search For Match

```
channels = capture_audio()

fingerprints_matching = [ ]
for channel_samples in channels
    hashes = process_audio(channel_samples)
    fingerprints_matching += find_database_matches(hashes)

predicted_song = align_matches(fingerprints_matching)
```

What does it mean for hashes to be aligned?

Search For Match

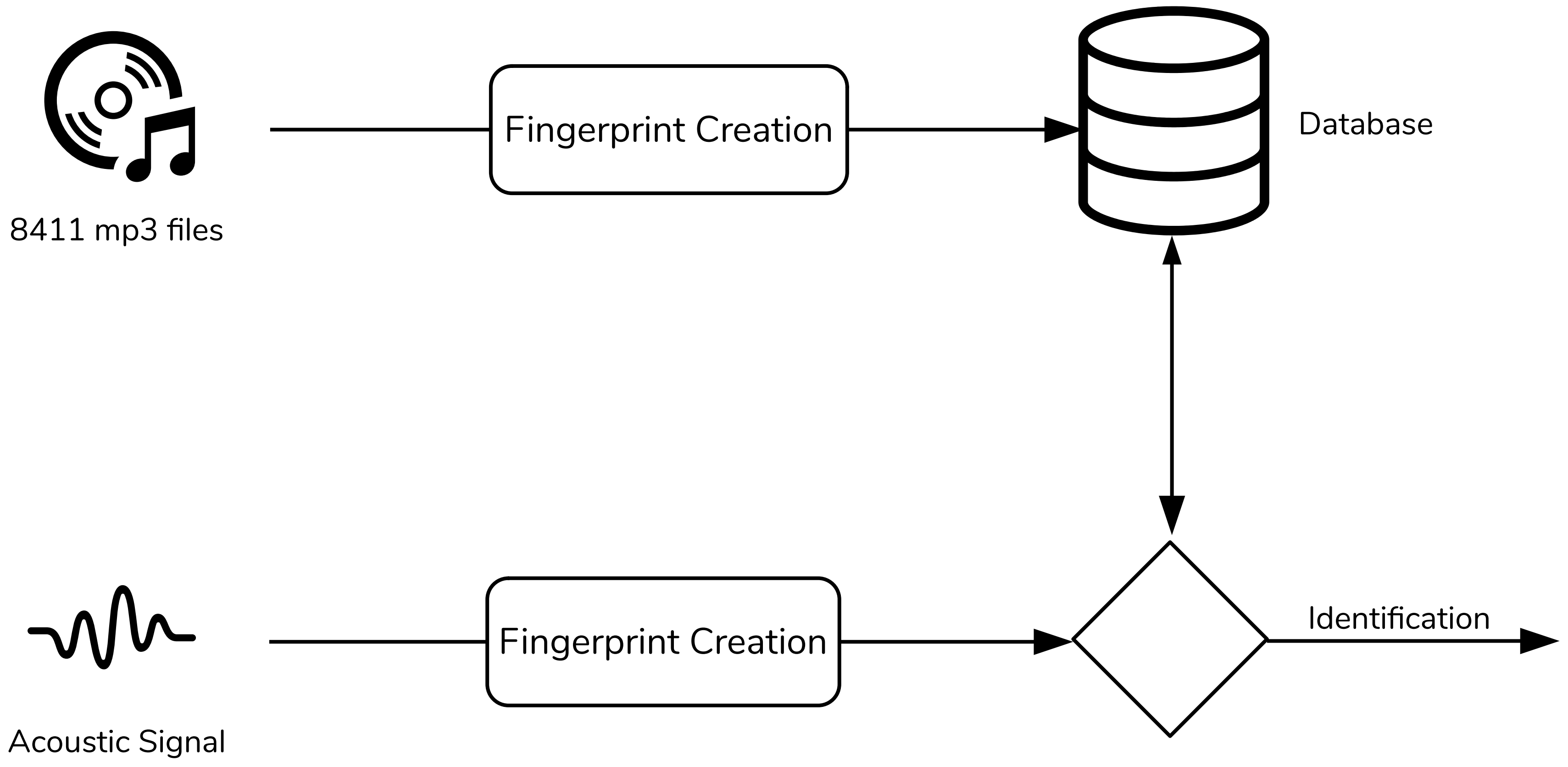
```
channels = capture_audio()

fingerprints_matching = [ ]
for channel_samples in channels
    hashes = process_audio(channel_samples)
    fingerprints_matching += find_database_matches(hashes)

predicted_song = align_matches(fingerprints_matching)
```

We know all the relative offsets will be the same distance apart.

Under this assumption, for each match we calculate a difference between the offsets:

difference = database offset from original track - sample offset from recording

Complete process



Where are we so far ?

✔ Enregistrement d’un extrait acoustique à travers un microphone.
✔ Création d’un Spectrogramme.
✔ Extraction des pics spectraux.
✔ Création d’empreinte acoustique (hachage).
✔ Mise en place d’une base de donnée relationnelle.
✔ Fonctions d’insertions et de modification de la BDD.

✗ Création des empreintes pour les 8411 musiques.
✗ Remplissage de la BDD avec les empreintes précédemment crées.
✗ Fonctions de recherche dans la BDD.
✗ Comparaison de deux extraits acoustiques.
✗ Interface interactive sur terminal.
✗ Calcul de la précision de l’identification en fonction du temps d’enregistrement.
✗ Calcul de la précision de l’identification en fonction du bruit ajouté.