M2-DS - AMIR
Lab - Music recognition

F. Angulo, D. Perera, G. Richard

1 Lab objective

The objective of the Lab is to implement a simple, yet efficient, music recognition method based on the original ™Shazam algorithm [1] and to evaluate its performance on a given music database. Additional information on audio fingerprint with more recent references can also be found in [2].

For the evaluation, it is proposed to study the robustness of the method to a variety of perturbations (noise, speed changes, ...). A final part of the lab will consist in proposing and implementing solutions to obtain a more robust algorithm to the different perturbations.

2 Principle of the method

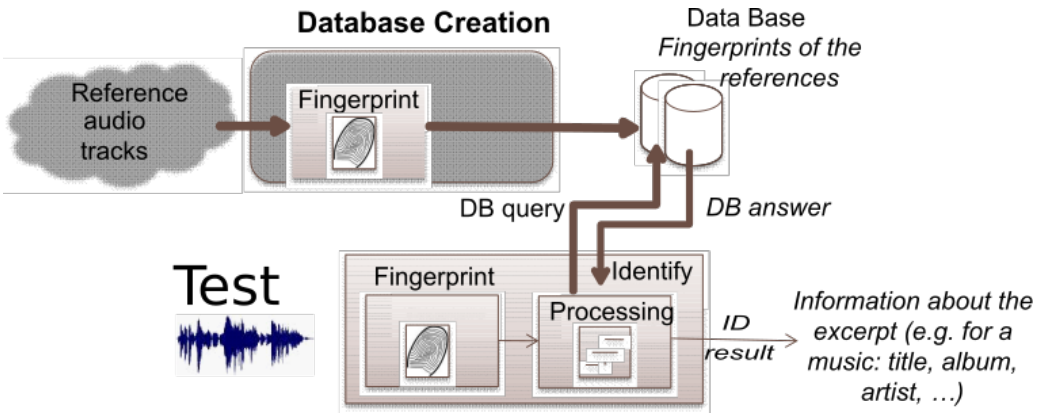

FIG. 1 – *Principle of the method (image from S. Fenet)*

The whole method is based on several important steps.

- **Database creation**, which includes the computation and storage of the audio fingerprints for the reference audio tracks.
  For this lab, two databases of references have been created:
  - REFSmallSet: a database of 10 references;
  - REFLargeSet: a database of 500 references.
- **Recognition**: For a given test audio excerpt, the recognition is done by computing the audio fingerprint, querying the database of references, processing of the database answer and finally identifying the recognized excerpt

2.1 Fingerprint computation

The first important step is therefore to compute the fingerprint
The different modules of the fingerprint computation includes:

1. **Computation of the spectrogram** (already implemented in the python notebook). The spectrogram is computed using a window size of *nfft=2048*, and *hopsize of 1024*. In this lab, to limit overall complexity, we only keep the first 100 frequency bins (*nbins=100*) which corresponds to a frequency range of 2205 Hz (since the original audio signals are sampled at 22,05 kHz);
2. **Design of the fixed grid** placed on the spectrogram (see figure below). The parameters for the fixed grid can be taken such as one cell corresponds to a rectangle of 6 time frames and 14 frequency bins [*n_frames=6; n_bins=14*];

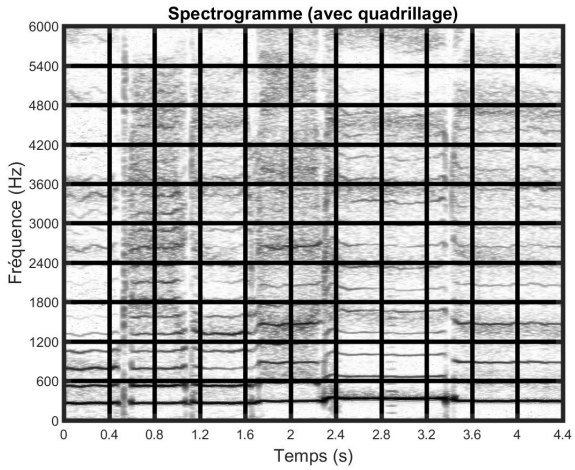

FIG. 2 – *Representation of the fixed grid*

3. **Obtention of the maxima:** only retain one maximum per grid cell and store the frequency index and frame index of each maximum in an array. (see figure 3).

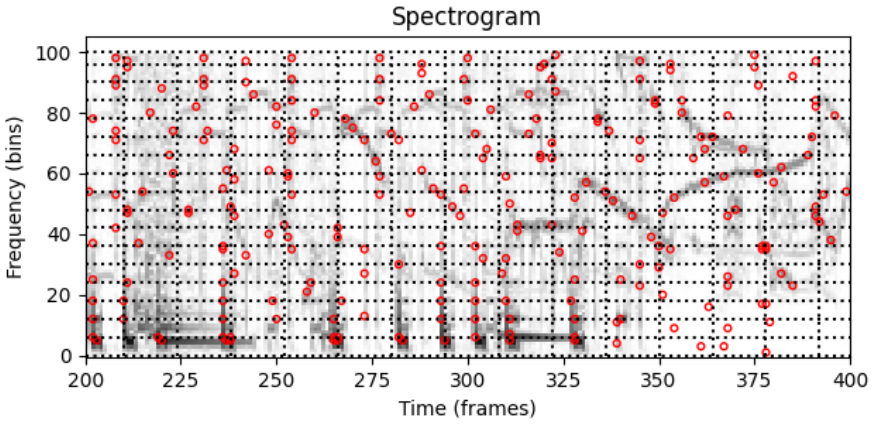

FIG. 3 – *Maxima found on the spectrogram in each grid cell.*

4. **Extraction of the keys:** For each maximum detected, build pairs of maxima by searching for the other maxima which are located in a neighbouring Time-Frequency (TF) region of predefined size [*n_frames=30; n_bins=40*] (see figure 4).
   The bottom-left corner of the TF region is placed at [*frames=*t_1*+5;bins=*f_1*-20*]. Each pair then constitutes a key and is encoded in the form of a vector [f_1,f_2,t_2-t_1] where (f_1,t_1) (resp. (f_2,t_2))

is the spectrotemporal position of the first maximum (resp. the second maximum);

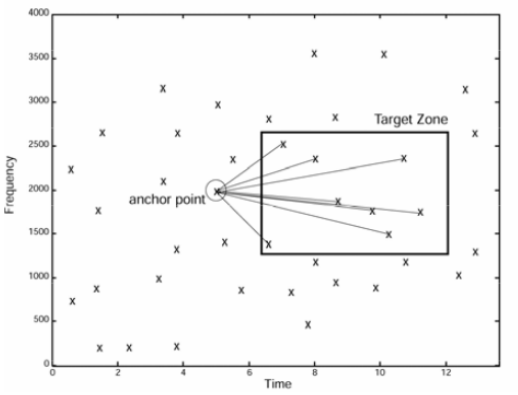

FIG. 4 – *Illustration of the neighbouring TF region*

2.2 Search and histogram computation

1. **Search:** The next step is to search in the database of the set of candidate references that have keys in common with those of the piece to be recognized. For each extracted key, we will note the time shift $\Delta t = t_1 - t_{ref}$ where t_1 and t_{ref} are respectively the instants of the first maximum of the key in the test excerpt to be recognized and in the candidate reference. The search itself in the database is already implemented
2. **Histogram computation and recognition:** A histogram is then built for each reference in the database by counting the number of common keys that occur at the same Δt (with $\Delta t \in [O,T]$). The reference that has the most keys in common at a constant time shift Δt is the recognized music recording (see Figure 5 below for an example histogram for 3 candidate references).

3 Experiments

In this lab, it is proposed to implement the method described above by completing the provided notebook template. The database of references have already been built and the fingerprint should then only be computed for the test signals.

3.1 Experiment 1

Evaluate your algorithm by measuring the recognition rate on a small dataset. This will be done by running the recognition for each test audio excerpt of the test set "TESTSmallSet" which is composed of 10 excerpt of 5s from the 10 music recordings of the reference test (REFSmallSet)

3.2 Experiment 2

Evaluate your algorithm by measuring the recognition rate on a larger dataset. This is the same experiment as Experience 1 by replacing the test set by "TESTLargeSet" and the reference set by (REFLargeSet)

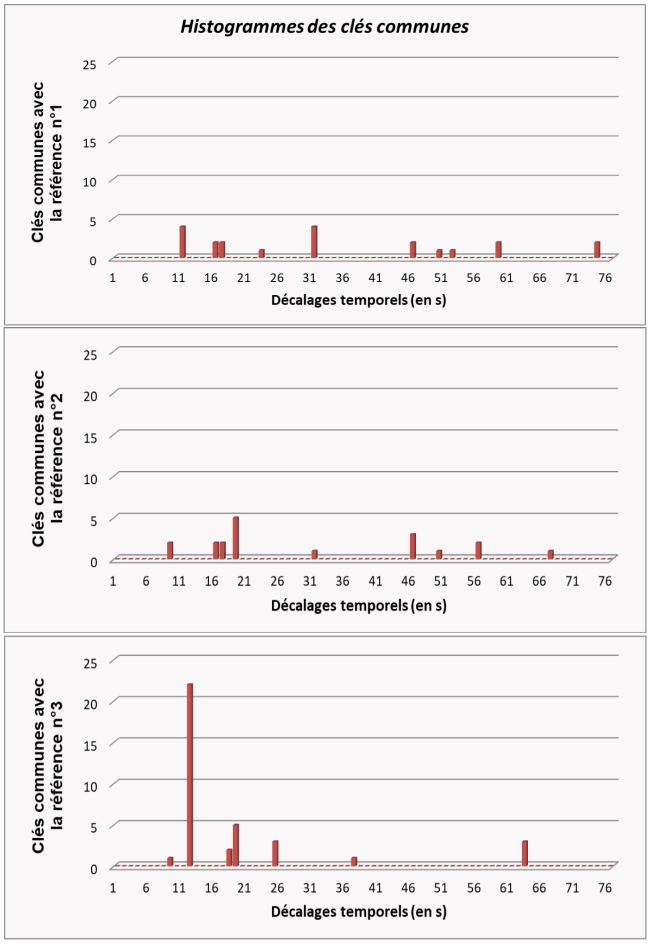

FIG. 5 – *Histogram for 3 reference. Here, reference 3 is the recognized music example*

3.3 Experiment 3

The goal of this experiment is to evaluate your algorithm by measuring its robustness to potential modifications or effects (noise addition and speed changes, in particular)

1. Use the provided functions to generate new test sets from the "TESTSmallSet" with varying level of additional noise (white noise or recorded background noise from $-10db$ to $+10db$), varying level of speed modifications (from $\pm 1\%$ to $\pm 10\%$) and varying pitch shifting, and measure the recognition rates as a function of the strength of the modification.
2. Propose and implement potential solutions to limit this robustness problem.

3.4 Experiment 4

In the last experiment, we propose to test the algorithm of a provided "enigma" test set called "TESTEnigmaSet" which contains 10 unknown audio samples (i.e. samples which have no references in the reference database).

1. Run your algorithm and observe the result
2. Propose and implement a solution which will permit to identify that these samples are unknown samples.
3. Evaluate your new algorithm on the TESTSmallSet to check that the recognition rate is not degraded.

3.5 (Optional) Experience 5: to go further

The fingerprint computation is based on computing 1 maximum per cell of a fixed grid. One of the drawback of this implementation is that two maxima can be very close to each other at both sides of a cell limit. An alternative solution is to compute the maxima on a dynamic cell (of similar size) that is slid on all time/frequency point. Evaluate the method with this alternative dynamic grid design for the different experiments run above. What are the drawbacks/advantages of this new method?

Références

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

[2] G. Richard, S. Fenet, and Y. Grenier, "De Fourier à la reconnaissance musicale," *Interstices*, Feb. 2019. [Online]. Available: https://interstices.info/de-fourier-a-la-reconnaissance-musicale/