

TP12 – Shazam

Dans ce TP, vous allez coder une version simplifiée de l'application Shazam, qui permet de reconnaître un morceau de musique et de donner à l'utilisateur des informations telles que le nom du morceau et son auteur.

Base d'empreintes sonores

La **création d'une base d'empreintes sonores** consiste à numériser un certain nombre de morceaux de musique (la base de l'application Shazam est gigantesque), et à calculer le sonagramme de chaque morceau. Seules les fréquences positives comprises entre deux bornes f_{\min} et f_{\max} sont conservées, afin de limiter la quantité d'information à traiter. Dans l'exercice 4 du TP11, vous avez recherché, pour chaque mesure, les n fréquences correspondant aux n plus grandes valeurs du module du spectre. Avec un nombre n de l'ordre de 100, vous avez constaté qu'il était possible de restituer un signal sonore quasiment identique à l'original. Vu que la quantité de données devient alors environ 30 fois moindre, cette idée est à l'origine de la compression MP3.

Il est possible de détecter encore moins de fréquences par mesure, si le but n'est pas de restituer le son original, mais de produire une « marque », appelée *empreinte sonore*, qui puisse caractériser un morceau de musique de manière unique : l'application Shazam détecte $n = 6$ fréquences seulement par mesure. Mais au lieu de chercher ces fréquences dans la bande $[f_{\min}, f_{\max}]$, elle découpe cette bande en $n = 6$ sous-bandes, et cherche dans chaque sous-bande la fréquence correspondant à la plus grande valeur du module du spectre.

Vous avez observé, lors du TP11, que les basses fréquences contenaient généralement plus d'énergie que les hautes fréquences (la couleur du sonagramme y est plus claire). De ce fait, au lieu d'effectuer une partition régulière de la bande de fréquences $[f_{\min}, f_{\max}]$, c'est sur l'intervalle $[\log(f_{\min}/f_{\min}), \log(f_{\max}/f_{\min})]$ que cette partition est effectuée. Les sous-bandes fréquentielles des basses fréquences sont effectivement moins larges en procédant ainsi. Si toutes les valeurs détectées étaient retenues, la taille de l'empreinte sonore serait très élevée. En réalité, dans chaque sous-bande, parmi les fréquences détectées, seules celles qui correspondent à une valeur du module du spectre supérieure à un seuil S (qui reste à définir) sont retenues. Dans une mesure de silence, il se peut donc que, parmi les fréquences détectées, aucune ne soit retenue.

L'empreinte sonore est une liste de couples de valeurs (t_i, f_i) , $i \in [1, p]$, où t_i est un instant (relativement au début du morceau, en secondes) et f_i une fréquence (en Hertz). L'ordre de cette liste n'a pas d'importance. Un exemple d'empreinte sonore est donné sur la figure 1, où les valeurs $\log(f/f_{\min})$ sont affichées en ordonnée. L'empreinte sonore constitue donc un nuage de points 2D présentant une certaine similitude avec une partition.

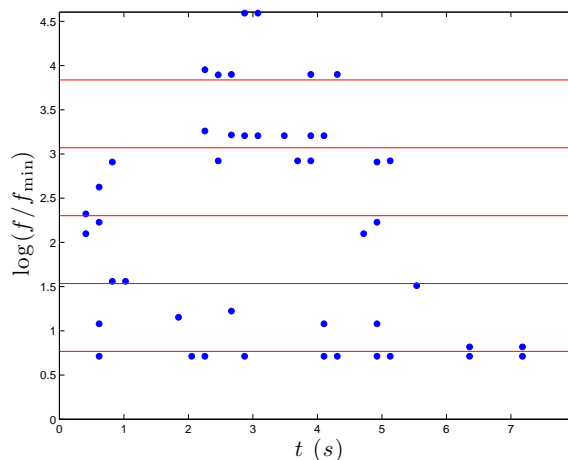


FIGURE 1 – Exemple d'empreinte sonore d'un extrait musical de 8 secondes.

L'**interrogation d'une base d'empreintes sonores** consiste ensuite à calculer l'empreinte sonore d'un morceau dont on cherche le titre et l'auteur, et à comparer celle-ci avec toutes les empreintes sonores de la base.

Exercice 1 : calcul d'une empreinte sonore

Complétez le script `exercice_1.m`, qui est censé calculer une empreinte sonore selon la procédure décrite dans le paragraphe précédent. En l'occurrence, calculez l'empreinte sonore de l'extrait musical `007.wav`, situé dans le répertoire `Donnees_WAV`. En guise de vérification, cette empreinte sonore est celle de la figure 1. Les limites de la bande fréquentielle doivent être fixées à $f_{\min} = 20 \text{ Hz}$ et $f_{\max} = 2000 \text{ Hz}$. Dans chaque sous-bande, fixez le seuil S à la somme de la moyenne et de l'écart-type (utilisez la fonction `std` de Matlab pour calculer ce dernier). La matrice `empreinte_sonore` comporte deux colonnes : la première contient des instants t_i , $i \in [1, p]$, la seconde les fréquences f_i correspondantes. Comme l'empreinte sonore de l'extrait musical `007.wav` comporte $p = 46$ points 2D, le nombre de lignes de cette matrice doit être égal à 46.

Exercice 2 : comparaison de deux empreintes sonores

Écrivez un script, de nom `exercice_2.m`, qui calcule l'empreinte sonore de l'extrait musical `solo.wav`, situé dans le répertoire `Donnees_WAV`, et la compare avec l'empreinte sonore du morceau dont provient cet extrait, contenue dans le fichier `nuages.mat`. Pour tous les décalages égaux à un nombre entier de mesures (la durée de la mesure est fixée à $0,2 \text{ s}$), calculez un score permettant de « caler » l'extrait dans le morceau entier. Vous devez trouver un décalage de 80 secondes environ. Pour écrire ce script, il est conseillé d'utiliser la fonction `dsearchn` de recherche du plus proche voisin (lisez sa documentation!).

Exercice 3 : interrogation de la base d'empreintes sonores

Cet exercice vise à reproduire (en beaucoup plus lent) le fonctionnement de l'application Shazam. Le répertoire `Donnees_WAV/Extraits` contient $m = 9$ extraits musicaux au format WAV. Un de ces morceaux, tiré au hasard, est lu. Complétez le script `exercice_3.m` de manière à calculer l'empreinte sonore de cet extrait (cf. exercice 1) et à chercher sa meilleure coïncidence avec chacune des m empreintes sonores de la base (cf. exercice 2), ce qui fournit m scores. Le morceau de la base correspondant au meilleur score est désigné.