TP12 – Shazam

Dans ce TP, vous allez coder une version simplifiée de l'application Shazam, qui permet de reconnaître un morceau de musique et retourne à l'utilisateur des informations telles que le nom du morceau ou 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 3 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 200, vous avez constaté qu'il était possible de restituer un signal sonore relativement fidèle à l'original. Vu que la quantité de données devient alors environ 15 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 seulement $n = 6$ fréquences 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, dans le 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, car les sous-bandes fréquentielles correspondant aux basses fréquences sont moins larges en procédant ainsi. Néanmoins, si toutes les fréquences détectées étaient retenues, la taille de l'empreinte sonore serait très élevée. En réalité, parmi les fréquences détectées, seules celles qui correspondent à une valeur du module du spectre supérieure à un seuil sont retenues (ce seuil peut varier d'une sous-bande à l'autre). Dans une mesure de silence, il est donc probable que, parmi les fréquences détectées, aucune ne sera 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 forme donc un nuage de points 2D ressemblant plus ou moins à une partition musicale.

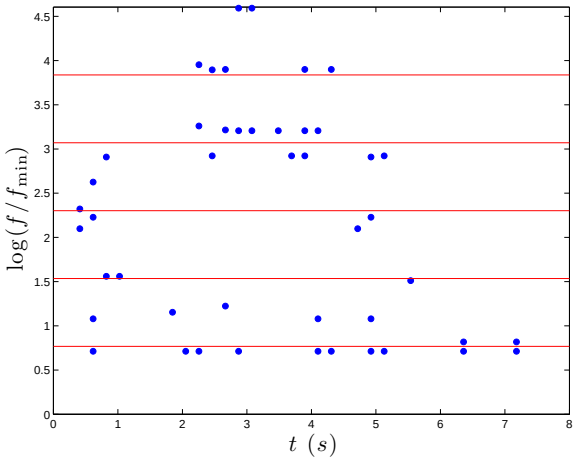

FIGURE 1 – Exemple d'empreinte sonore d'un enregistrement musical de 8 secondes.

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

Écrivez la fonction `calcul_ES`, appelée par le script `exercice_1.m`, et importez la fonction `T_Gabor` du TP11, afin de calculer l'empreinte sonore d'un enregistrement musical (en l'occurrence, il s'agit de l'extrait `007.wav`). En guise de vérification, cette empreinte sonore doit être celle de la figure 1.

Les limites de la bande fréquentielle utile sont fixées à $f_{\min} = 20\ Hz$ et $f_{\max} = 2000\ Hz$. Dans chaque sous-bande : le maximum du module complexe de chaque colonne du sonagramme est calculé ; le seuil de sélection des maxima est égal à la somme de la moyenne et de l'écart-type, calculés sur les maxima (utilisez la fonction `std` pour calculer l'écart-type). La première colonne de la matrice `ES` doit contenir des instants t_i, la seconde les fréquences f_i correspondantes. Comme l'empreinte sonore de l'extrait `007.wav` comporte 46 points 2D, le nombre de lignes de cette matrice doit être égal à 46. **Attention :** veillez à ce que les sous-bandes forment bien une partition de l'ensemble des fréquences, donc en particulier à ce qu'elles soient disjointes.

Exercice 2 : recalage d'un extrait sur le morceau entier

Le script `exercice_2.m` calcule l'empreinte sonore de l'extrait `solo.wav`, puis cherche à le recaler sur le morceau entier, dont l'empreinte sonore est lue dans le fichier `nuages.mat`.

Écrivez la fonction `decalage_ES`, appelée par ce script, qui doit superposer à l'empreinte sonore du morceau entier l'empreinte sonore de l'extrait, après décalage, et retourne un score permettant de quantifier l'écart entre ces deux empreintes sonores. Pour écrire cette fonction, il est conseillé d'utiliser `dsearchn`, qui est une fonction de recherche du plus proche voisin, dont l'utilisation a déjà été préconisée pour la fonction `priorites` du TP8. Vous devez trouver un décalage optimal de 70 secondes environ.

Remarque - Les écarts n'étant pas calculés en échelle logarithmique, cela revient à accepter des écarts plus élevés dans les hautes fréquences que dans les basses fréquences, ce que vous devez effectivement observer sur le meilleur recalage obtenu.

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 `Base` accessible sur Moodle contient $m = 9$ morceaux de musique au format WAV, dont les empreintes sonores ont été pré-calculées et stockées dans le fichier `base_donnees.mat`.

Écrivez un script, de nom `exercice_3.m`, qui tire au hasard un nombre entre 1 et m, lit le morceau correspondant, calcule son empreinte sonore (cf. exercice 1), et enfin cherche sa meilleure correspondance avec chacune des m empreintes sonores de la base (cf. exercice 2), ce qui fournit m scores. Le nom du morceau de la base obtenant le meilleur score est affiché.