TP11 – Reconnaissance musicale

Empreinte sonore d'un signal acoustique

Dans l'exercice 3 du TP10, il était demandé de sélectionner, dans chaque colonne du sonagramme d'un signal acoustique, les m plus grands coefficients de Fourier (au sens du module complexe). En ne conservant que 20% des fréquences positives, et pour un nombre m de l'ordre de 20, vous avez constaté qu'il était possible de restituer un signal acoustique fidèle à l'original, avec un coefficient de compression avoisinant la dizaine. Cette technique de compression avec perte constitue une version simplifiée de la première étape du format MP3 (la deuxième étape consisterait à encoder les coefficients de Fourier sélectionnés).

Comme l'application `Shazam` ne vise pas à restituer le son d'origine, mais à le caractériser par une *empreinte sonore* unique, le nombre d'informations extraites du signal peut être beaucoup plus restreint. Pour commencer, seules les fréquences de la bande $[f_{\min}, f_{\max}[= [20\ Hz, 2000\ Hz[$ sont conservées. Cette bande de fréquences est ensuite découpée en $m = 6$ sous-bandes, de manière à former une partition régulière en « log-fréquences ». Comme l'information d'un signal acoustique se concentre essentiellement dans les basses fréquences, il est logique que les sous-bandes soient plus larges dans les hautes fréquences que dans les basses fréquences.

Dans chaque colonne du sonagramme, et pour chacune des $m = 6$ sous-bandes, le coefficient de Fourier de plus grand module complexe est identifié., mais le numéro de colonne et le numéro de ligne de ce coefficient ne sont pas suffisamment informatifs pour être enregistrés tels quels. C'est pourquoi le numéro de ligne est traduit en une fréquence exprimée en Hertz, et le numéro de colonne est traduit en un temps exprimé en secondes. Cependant, si la fréquence est définie de manière absolue, il n'en va pas de même du temps, qui dépend du choix d'une origine. Cette disymétrie entre les deux dimensions d'un sonagramme annonce que la reconnaissance musicale passera par un *recalage temporel*, c'est-à-dire par la recherche d'une coïncidence entre deux empreintes sonores, en effectuant une translation de l'une relativement à l'autre le long de l'axe temporel.

Si les $m = 6$ maxima détectés dans chaque colonne du sonagramme étaient retenus, la taille de l'empreinte sonore serait encore inutilement élevée. En réalité, seuls les maxima supérieurs à un seuil sont retenus. Par exemple, dans un moment de « silence », aucun maximum ne doit être retenu. Néanmoins, il est nécessaire de définir un seuil par sous-bande, sans quoi la plupart des maxima retenus se situeraient dans les sous-bandes correspondant aux basses fréquences. En pratique, pour chaque sous-bande, le seuil de sélection est égal à la somme de la moyenne et de l'écart-type des maxima, valeurs calculées sur toute la durée du signal.

L'empreinte sonore d'un signal acoustique consiste donc en une liste de couples de valeurs (t_j, f_j), où t_j est un instant (relativement au début du morceau, exprimé en secondes) et f_j une fréquence (exprimée en Hertz). Il est notable que l'ordre de cette liste n'a pas d'importance. La figure 1 montre un exemple de sonagramme, où seules les fréquences positives inférieures à 2000 Hz ont été affichées, et l'empreinte sonore qui en découle.

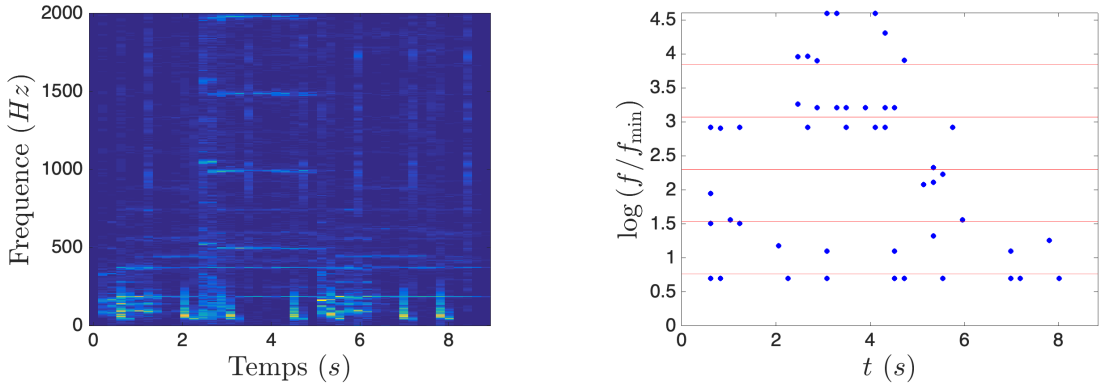

FIGURE 1 – À gauche : exemple de sonagramme. À droite : empreinte sonore qui en découle.

Exercice 1 : calcul d'une empreinte sonore

Après avoir importé une copie des fonctions `t_Gabor` et `sonagramme` du TP10, écrivez la fonction `calcul_ES`, appelée par le script `exercice_1`, qui est censé calculer le sonagramme, puis l'empreinte sonore de l'extrait musical `007.wav`. Les affichages produits par ce script doivent être identiques à ceux de la figure 1.

Il est conseillé de tenir compte des indications suivantes :

- Dans chaque sous-bande, le seuil de sélection des maxima doit être égal à la somme de la moyenne et de l'écart-type des maxima. Vous pouvez utiliser la fonction `std` pour calculer l'écart-type.
- La première colonne de la matrice `ES` contient les instants t_j, la seconde les fréquences f_j. Cette matrice doit comporter 48 lignes, ce qui signifie que l'empreinte sonore de la figure 1 contient 48 points 2D.
- Si vous ne trouvez pas le bon résultat, vérifiez que les fréquences des différentes sous-bandes forment bien une partition de l'ensemble des fréquences : l'union des intervalles doit être égale à la bande de fréquences $[20\ Hz, 2000\ Hz[$, l'intersection entre sous-bandes devant être vide.

Remarque – La largeur de la fenêtre glissante préconisée par l'application `Shazam` pour le calcul du sonagramme est $0,2\ s$. C'est effectivement la valeur attribuée à la variable `T_fenetre` dans le script `exercice_1`. En comparaison du TP10, où cette variable était égale à $0,01\ s$, les fréquences de la bande $[20\ Hz, 2000\ Hz[$ sont donc échantillonnées plus finement, au détriment de l'échantillonnage temporel, mais cette valeur de `T_fenetre` est bien adaptée à la musique, car elle correspond au tempo des morceaux les plus rapides.

Reconnaissance musicale

La notion d'empreinte sonore présente des similitudes avec celle d'*empreinte digitale*, car elle est censée caractériser un morceau de musique de manière unique. Bien entendu, on ne peut identifier un individu à partir de son empreinte digitale que si celle-ci apparaît dans une base de données. On ne pourra donc identifier un morceau de musique que si son empreinte sonore a été calculée puis stockée dans une base de données.

La base de données de `Shazam` contient les empreintes sonores précalculées de morceaux de musique complets. L'interrogation de cette base de données consiste à calculer l'empreinte sonore de l'extrait musical que l'on cherche à identifier, et à comparer celle-ci avec toutes celles de la base de données. Comme nous l'avons déjà dit, il ne suffit pas d'une simple comparaison entre l'empreinte sonore de l'extrait et chaque empreinte sonore de la base de données : il est nécessaire de procéder à un *recalage temporel* entre ces empreintes sonores, qui consiste à les translater l'une par rapport à l'autre. Il existe donc un « meilleur recalage » avec chaque empreinte sonore de la base de données, qui est très mauvais dans la plupart des cas. Le morceau identifié comme résultat de la recherche est celui qui correspond au « meilleur des meilleurs recalages ». Toutefois, il ne faut pas s'attendre à ce que ce meilleur recalage soit parfait, ce qui signifierait que chaque point 2D de l'empreinte sonore de l'extrait musical devrait coïncider parfaitement avec un point 2D de l'empreinte sonore du résultat.

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

Le script `exercice_2` calcule l'empreinte sonore de l'extrait `solo.wav`, puis cherche à le recaler temporellement sur le morceau complet, dont l'empreinte sonore est lue dans le fichier `nuages.mat`. Écrivez la fonction `decalage_ES`, appelée par `exercice_2`, qui doit superposer à l'empreinte sonore du morceau complet l'empreinte sonore de l'extrait décalé temporellement, et retourner un score permettant de quantifier le recalage.

Il est conseillé de tenir compte des indications suivantes :

- Il est recommandé d'utiliser la fonction `dsearchn` de Matlab, avec la syntaxe suivante :
  `[~,distances] = dsearchn(ES_1,ES_2);`
  qui recherche, pour chaque point du nuage `ES_2`, le point du nuage `ES_1` le plus proche, et qui retourne les distances minimales (cf. la documentation de cette fonction).
- Le score du recalage peut être estimé par la moyenne des distances retournées par `dsearchn`, mais la médiane est connue pour être un estimateur plus robuste aux « données aberrantes ». En testant ces deux possibilités, vous constaterez que le meilleur recalage n'est pas exactement le même. Le choix de l'estimateur n'est donc pas sans conséquence.

Exercice 3 : reconnaissance musicale

Cet exercice vise à reproduire, dans une version simplifiée, le fonctionnement de l'application `Shazam`. Les empreintes sonores de $n = 73$ morceaux de musique (complets) ont été calculées et stockées dans le fichier `base_donnees.mat`. Plus précisément, les informations relatives au $k^{\text{ème}}$ morceau de la base de données sont concaténées dans une structure de nom `titres_auteurs_empreintes{k}`, qui comporte les champs `titre`, `auteur` et `empreinte` (la variable `titres_auteurs_empreintes` est donc un vecteur de n structures).

Le script `exercice_3` tire au hasard un entier `k` entre 1 et n (fonction `randi`), lit l'extrait musical de nom `extrait<k>.wav` dans le répertoire `Extraits`, calcule son empreinte sonore (cf. exercice 1), cherche son meilleur recalage (cf. exercice 2) avec chacune des n empreintes sonores de la base de données, et enfin affiche le titre et l'auteur correspondant au meilleur score. Écrivez la fonction `calcul_liste_scores`, appelée par `exercice_3`, qui retourne la liste des scores de recalage de l'empreinte sonore de l'extrait musical tiré au hasard avec les n empreintes sonores de la base de données.

Dans cette base de données, le morceau intitulé *Sarah* apparaît deux fois : une fois interprété par son auteur Georges Moustaki, une autre fois par Serge Reggiani. Vous pourrez constater que les empreintes sonores de ces deux morceaux sont suffisamment différentes pour que l'interprète soit reconnu sans erreur, pour peu que vous sachiez distinguer les voix de ces deux chanteurs... Afin de vous aider à valider les résultats, vous pouvez vous référer au fichier `catalogue.txt`, mais vous pouvez également profiter de ce TP pour tester vos connaissances !

Exercice 4 : test de validation (exercice facultatif)

Dans l'exercice 3, il est sous-entendu que n désigne à la fois le nombre d'empreintes sonores de la base de données et le nombre d'extraits musicaux stockés dans le répertoire `Extraits`. Cela signifie qu'il existe une bijection entre ces deux ensembles. Même si cela est le but des concepteurs de `Shazam`, il s'agit d'un but impossible à atteindre. En d'autres termes, il existe toujours des morceaux non encore entrés dans la base.

Copiez un des extraits du répertoire `Audio` dans le répertoire `Extraits`, en veillant à lui donner un nom conforme au modèle utilisé. Sans modification du script `exercice_3`, la lecture de cet extrait musical causera inévitablement une réponse erronée, puisqu'un recalage optimal sera quand même trouvé. Proposez une modification du script `exercice_3`, de nom `exercice_4`, consistant à soumettre le résultat à un test de validation pour déterminer s'il doit être affiché ou non.