Master 2 ATIAM (2017-2018)
TP Structuration et résumé audio

Geoffroy.Peeters@ircam.fr
UMR STMS 9912 (IRCAM CNRS UPMC)

Table des matières

1 Introduction

Matériel nécessaire pour ce TP

- `20161016_Peeters_20162017_ATIAM_TP_Structure.pdf`
- `F_audioSummary.m`
- `F_filterSelfSimilarity.p`
- `audio_gammepno.wav`
- `audio_bep.wav`

L'objectif de ce TP est de construire un système simple mais complet de création de résumé audio.

- l'entrée du système est un fichier .wav contenant la totalité du signal audio d'un morceau de musique,
- la sortie du système est un fichier .wav contenant le résumé audio d'une durée de 20 s.

La méthode de création de résumé utilisée sera basée sur le « summary score » de [Cooper, Foote, 2002] (voir références à la fin). Cependant à la différence de [Cooper, Foote, 2002] (qui utilisent des observations de type MFCCs) nous utiliserons des observations de type Chroma (voir références à la fin).

En pratique le TP consistera à compléter le script `F_audioSummary.m` par les fonctions manquantes. Ce script effectue les appels suivants :

- Lecture d'un fichier .wav à l'aide de la fonction `audioread`
- `F_extractChroma` : extraction (à chaque instant du fichier audio) d'un vecteur d'observation de type Chroma
- `F_computeSelfSimilarity` : calcul de la matrice de similarité (temps,temps) à partir de la succession de vecteurs de Chroma
- `F_filterSelfSimilarity` : filtrage de la matrice de similarité pour renforcé les sous-diagonales et réduire le bruit de fond
- `F_computeSummaryScore` : calcul du summary score pour une longueur de 20s à partir de la matrice de similarité

- Sélection du summary score le plus élevé
- Ecriture du résumé audio par la fonction audiowrite

2 Extraction des Chromas à partir du signal audio

Fonction à écrire : `obs_m = F_extractChroma(data_v, sr_hz, L_n, STEP_n)`.

L'objectif de cette fonction est d'extraire à chaque instant du fichier audio un vecteur d'observation de type Chroma. Pour rappel, le Chroma est un vecteur à 12 dimensions représentant l'importance des 12 classes de demi-tons (classes de hauteur ou pitch-classes) à un instant donné.

La fonction à écrire comporte deux étapes :

- 1) Création d'une matrice de filtres effectuant le mapping entre les "bins" du module de la DFT et les 12 valeurs de Chroma
- 2) Analyse du signal à fenêtre glissante effectuant :
  - a) fenêtrage du signal
  - b) calcul du module de la DFT
  - c) mapping entre les valeurs du module de la DFT et les 12 valeurs de Chroma à l'aide de la matrice de filtres précédemment créée

2.1 Création de la matrice de filtres

Nous notons N le nombre de points de la DFT, nous nous intéressons uniquement au demi- axe positif des fréquences de la DTF (indices entre 1 et $N/2+1$).

Chaque filtre effectuera le mapping entre le vecteur de la DFT et un Chroma spécifique. Pour rappel, il y a 12 chromas, chacun correspondant à une classe de hauteur de demi-ton `c`. Une classe de hauteur de demi-ton `c` représente toutes les hauteurs de demi-tons `n` tel que `c = mod(n, 12)+1`.
Pour chaque chroma c, le filtre à créer est donc composé de la somme des filtres passe-bandes chacun centrés sur les demi-tons n tel que `c = mod(n, 12)+1`.

Chacun des filtres passe-bandes est centré sur une hauteur de demi-ton n et s'étend de part et d'autres de ce demi-ton jusqu'au demi-tons inférieur et supérieur.
On s'aidera de la formule suivante effectuant le mapping entre les fréquences de la DFT, $freqTF$, et les notes midi correspondantes : `midiTF = 12 * log2(freqTF / 440)+69`.

On utilisera également la formule suivante pour la forme du filtre passe-bande :
`H = 1/2 * tanh(pi * (1-2 * x))+1/2` dans lequel `x` représente la valeur absolue de la distance entre `x = abs(n - midiTF)` avec

- `midiTF` : la fréquence des bins de la TF exprimé en notes midi et

- `n` : la hauteur de demi-ton considéré

La figure suivante illustre la partie entre 0 et 2000 Hz du filtre de demi-ton pour n=69 (note A4 ou 440Hz).

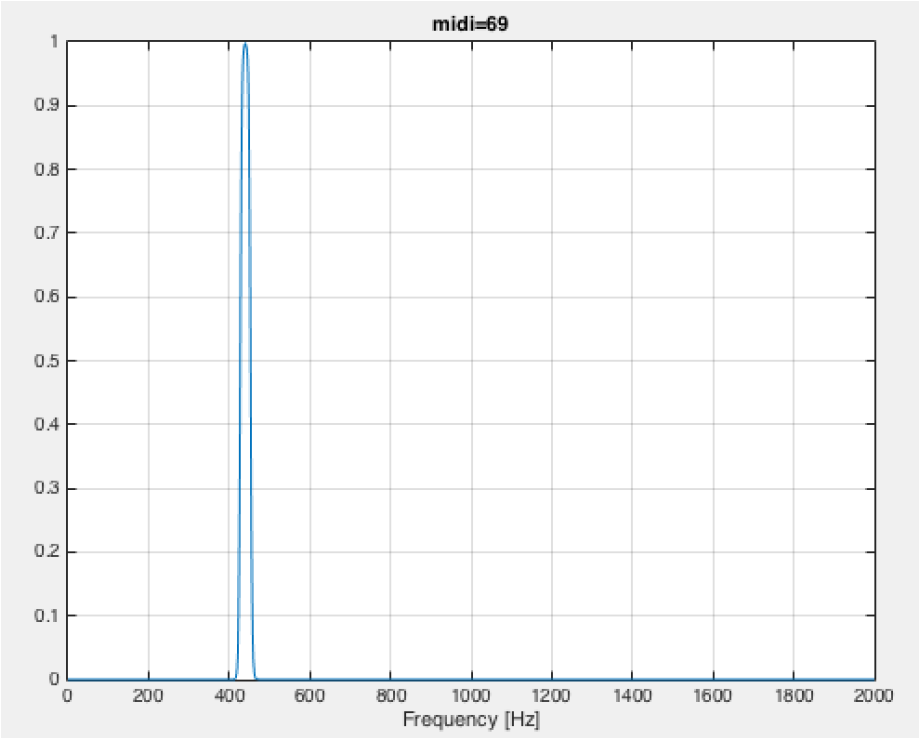

Figure :

La figure suivante illustre la partie entre 0 et 2000 Hz du filtre de chroma pour c=10 (note A=A2+A3+A4+...) somme des contributions des filtres n tel que 10 = mod(n, 12)+1.

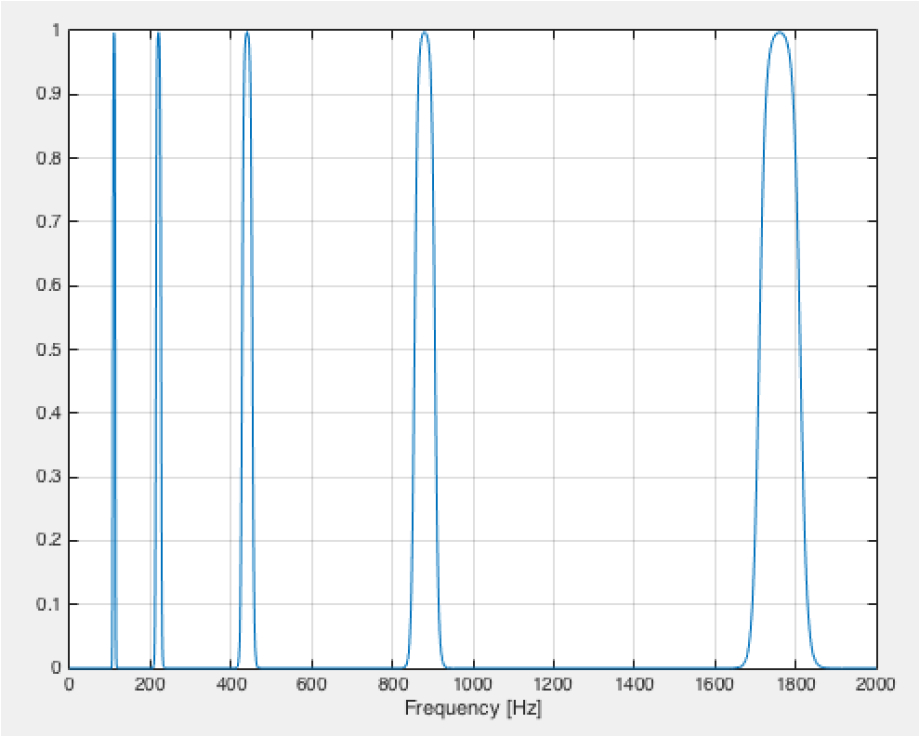

Figure :

On se limitera à l'utilisation des filtres entre n = 36 à 119.

Pour un chroma donné, son filtre est normalisé de manière à ce que sa somme soit égale à 1.

L'ensemble des filtres de chroma seront stockés dans une matrice de taille (12, N/2+1), i.e. un filtre de chroma par ligne. La figure suivante illustre la partie entre 0 et 2000 Hz de l'ensemble des filtres de chroma normalisés. L'affichage s'effectue à l'aide de la fonction `imagesc`.

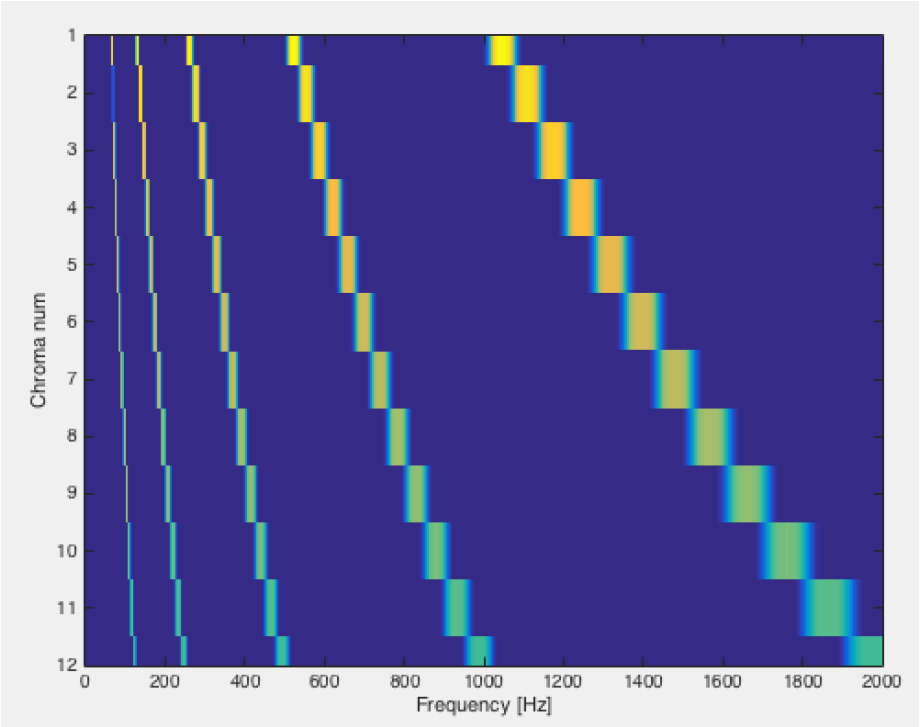

Figure :

2.2 Calcul du chromagram par analyse à fenêtre glissante

L'analyse à court terme s'effectue à l'aide d'une fenêtre de type blackman de longueur 0.2s, le pas d'avancement étant fixé à 0.2/3s.
La conversion en nombre d'échantillon s'effectue selon

- `L_n=round(0.2 * sr_hz)` et
- `STEP_n=round(0.2/3 * sr_hz)`
- dans lesquels `sr_hz` désigne le sampling rate du signal audio.

Le sampling rate dépend du signal et est donné par la fonction audioread. Pour rappel, la taille de la FFT, `N`, doit être supérieure à la longueur de la fenêtre, `L_n`, et être une puissance de 2. On s'aidera d'un facteur de zéro-padding > 1 afin de permettre l'obtention d'un nombre suffisant de points de la FFT pour les filtres de chroma en basses fréquences. A chaque trame d'analyse, on calculera

- fenêtrage du signal
- calcul du module de la DFT du signal fenêtré
- calcul du vecteur de chroma correspondant. Pour cela on multipliera le module de la DFT par la matrice de filtre de chroma : `(12, N/2+1) * (N/2+1, 1) = (12, 1)`.

La sortie de la fonction est une matrice dont les lignes correspondent aux 12 valeurs de chroma et les colonnes aux trames successives de l'analyse à court terme. Sa taille est `(12, nbFrame)`.

2.3 Test de la fonction

On testera la fonction d'observation chroma sur le signal `audio_gammepno.wav` représentant une gamme chromatique ascendante au piano.

L'affichage de la matrice d'observation chroma s'effectue à l'aide de la fonction `imagesc`. Le résultat pour le signal gammepno.wav doit ressembler à celui de la figure suivante. A chaque instant, le chroma représentant la note jouée à la valeur maximale (1, 2, 3, 4, ..., 12 au cours du temps).

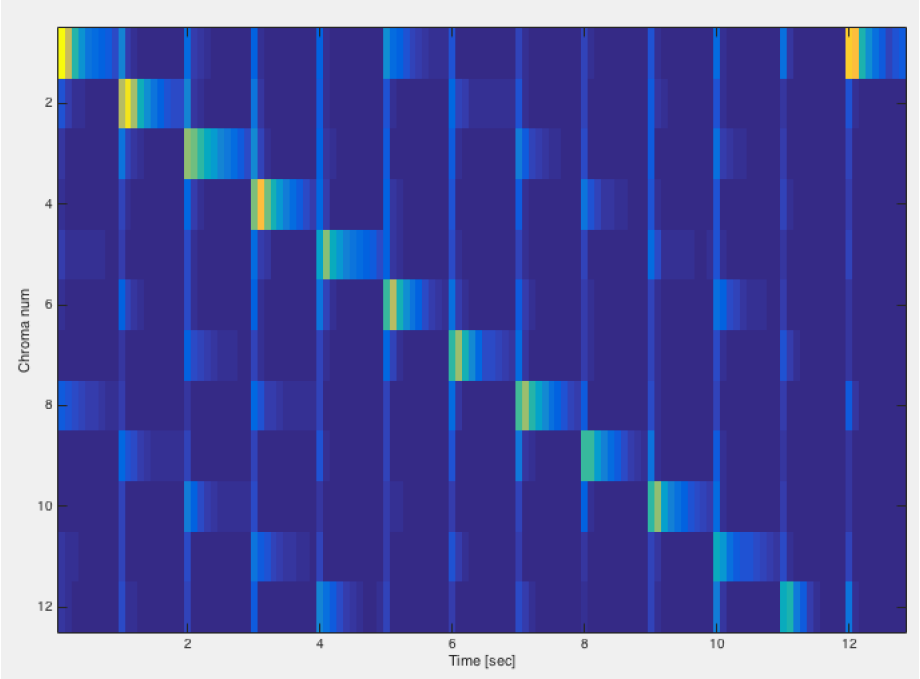

Figure :

3 Calcul de la matrice d'auto-similarité (temps,temps) des observations

Fonction à écrire : `SSM_m = F_computeSelfSimilarity(obs_m)`

A partir de la matrice d'observation de Chroma `obs_m (12,nbFrame)`, nous calculons la matrice d'auto-similarité (temps,temps) du morceau `SSM_m (nbFrame,nbFrame)`. La sortie de cette fonction est une matrice carrée symétrique de taille égale au nombre de trames du morceau.

Nous utilisons pour cela la distance cosinusoidale. Nous calculons la distance entre chaque couple de vecteurs d'observation aux temps i et j. Etant donné la nature symétrique de la matrice, on se limitera à calculer la matrice triangulaire

inférieure ou supérieure de la matrice et à en déduire l'autre partie par transposition. Pour rappel la distance cosinusoidale entre un vecteur x_i et un vecteur y_i est calculée par

$$dist(x,y) = \frac{\sum_i x(i)y(i)}{\sqrt{\sum_i x^2(i)}\sqrt{\sum_i y^2(i)}} \quad (1)$$

3.1 Test de la fonction

A titre d'exemple, on montre ci-dessous la matrice de similarité correspondant au fichier `audio_gammepno.wav`. L'affichage en Matlab s'effectue à l'aide de la fonction `imagesc`.

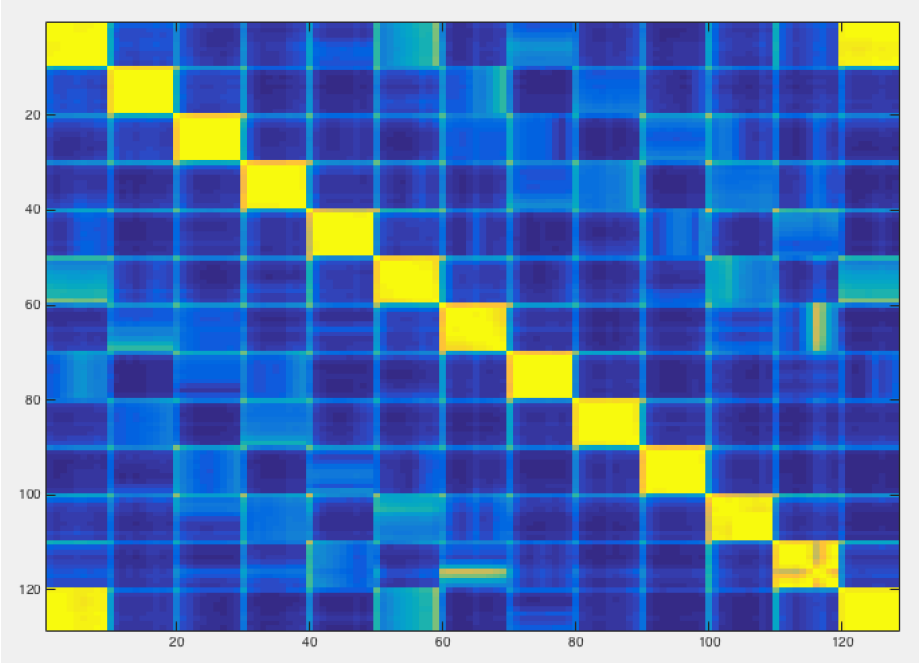

Figure :

4 Filtrage de la matrice d'auto-similarité (temps,temps)

Fonction à utiliser : `SSMfiltered_m = F_filterSelfSimilarity(SSM_m, STEP_sec)`

Afin d'accentuer la présence de diagonales dans la matrice de similarité, et donc de mettre en évidence les répétitions de type "séquences" par rapport au bruit de fond, on utilisera la fonction `Ffiltragematrice.p`. Cette fonction effectue un filtrage passe-haut dans le sens perpendiculaire aux diagonales et un filtrage passe-bas dans le sens des diagonales de manière à accentuer la présence de diagonales par rapport au bruit de fond et à lisser les discontinuités des

diagonales (répétition non-exacte). Cette partie n'ayant pas été étudiée lors du cours, nous nous contenterons d'appliquer telle quelle cette fonction.

5 Estimation du point de démarrage du résumé audio : méthode du summary score

Fonction à écrire : `start_frame = F_computeSummaryScore(SSMfiltered_m, L_frame)`

A partir de la matrice de similarité filtrée `SSMfiltered_m`, nous cherchons à estimer le point de départ dans le morceau d'un extrait continu de 20 s tel que cet extrait soit le plus représentatif du morceau (i.e. similaire au reste du morceau). Pour cela, nous calculons le "summary score" proposé par [Cooper, Foote, 2002].

Pour chaque temps (ligne q de la matrice), le summary score est calculé comme la somme des colonnes de la matrice de similarité à q donné. Il représente la similarité de l'instant q avec l'ensemble des temps du morceau. Pour un segment démarrant en q et de longueur L, le summary score est calculé comme la somme des sommes des colonnes entre q et q+L. Il représente la similarité moyenne du segment [q,q+L] avec l'ensemble du morceau. L est la longueur du résumé recherché. Nous recherchons un résumé de longueur 20s (convertir ce temps en nombre de trames en utilisant le pas d'avancement (0.1s)). Nous cherchons la valeur de q qui maximise le summary score. La sortie de la fonction est cette valeur de q.

6 Création du résumé audio

Le résumé audio est obtenu en prenant le segment du signal audio démarrant en q et de longueur L. On écrira le fichier audio résultant avec la fonction `audiowrite`.

Attention la valeur retournée par la fonction `F_computeSummaryScore` (`start_frame`) indique un numéro de trame et non un temps !

- il faut convertir cette valeur en temps en utilisant le pas d'avancement de l'analyse (0.1s)
- il faut ensuite convertir la valeur de temps en nombre d'échantillons du signal en utilisant le sampling rate du signal audio.

Le résumé final sera calculé sur le morceau : song1.wav.

7 Bibliographie

- Article libre accessible en ligne expliquant l'extraction des vecteurs de chroma

    - Peeters, G. (2006). Chroma-based estimation of musical key from audio-signal analysis. Proc. of ISMIR, Victoria, Canada.
    - Cet article est accessible à l'URL suivante : `http://ismir2006.ismir.net/PAPERS/ISMIR06134_Paper.pdf`
- Article libre accessible en ligne expliquant l'extraction du summary score
    - Cooper, M. and J. Foote (2002). Automatic Music Summarization via Similarity Analysis. Proc. of ISMIR, Paris, France.
    - Cet article est accessible à l'URL suivante : `http://ismir2002.ismir.net/proceedings/02-FP03-1.pdf`