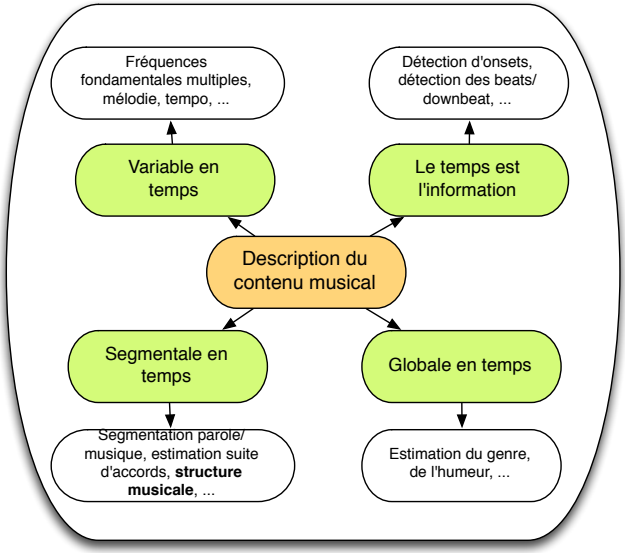
Master 2 ATIAM (2018-2019)
Extraction automatique d'une structure musicale et génération d'un résumé audio

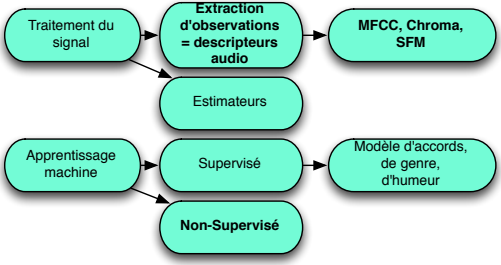
Geoffroy Peeters

LTCI - Télécom ParisTech

2018-2019

Différents types de description du contenu musical





Détection d'une structures musicale d'un morceau de musique

Objectifs

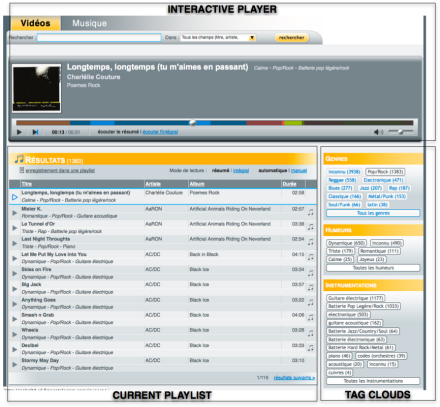
- Estimer une structure d'un morceau de musique
- Créer automatiquement un résumé audio représentatif du contenu du morceau

Applications

- Ecoute inter-active : création d'interface de lecture (player) interactif,
- Pré-écoute rapide d'un morceau
- Exemples audio et vidéo

Méthode

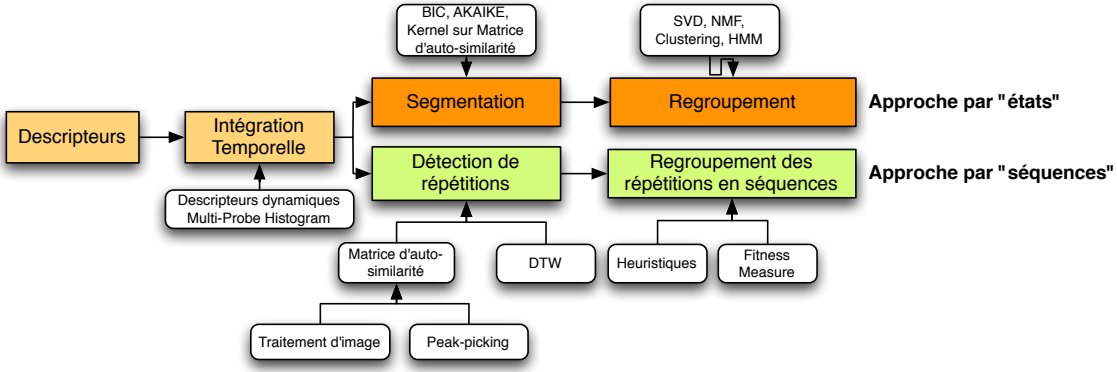
- Extraction de descripteurs audio
- Visualisation de la structure
- Estimation de la structure
  - Trdictionnalement : apprentissage non-supervisé
  - Récemment : apprentissage supervisé



source : Quaero project, MSSE-Orange interface

Méthodes d'estimation de la structure

- 1) Extraction d'observations pertinentes du signal audio
  - Descripteurs audio : mise en évidence de différents contenus (timbre, harmonique, bruité, ...)
- 2) Analyse des observations afin de détecter une structure
  - Approche par états
    - segmentation temporelle et
    - regroupement des segments homogènes identiques
  - Approche par séquences
    - détection des répétitions non-homogènes et
    - regroupement des segments répétés en séquences



2- Descripteurs audio

Descripteurs audio

[G. Peeters. A large set of audio features for sound description (similarity and classification) in the cuidado project. Cuidado project report, Ircam, 2004.]

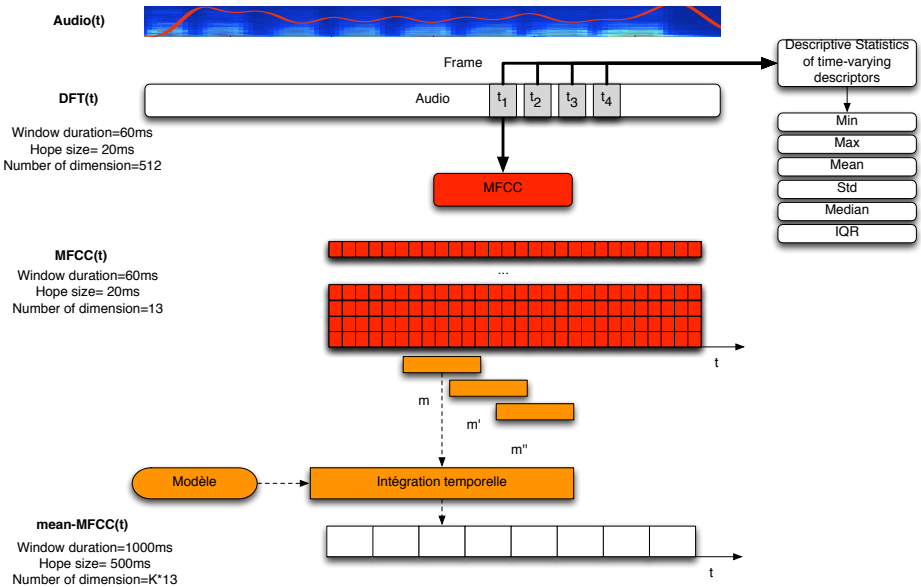
- Valeurs numériques extraites du signal audio dont le but est de représenter une propriété particulière de son contenu
  - Tout est dans la forme d'onde, dans la TFCT, diffiçile à lire, trop grande dimension
- Contrainte :
  - on veut le même nombre de dimensions pour toutes les données
- Extraction ?
  - Algorithme d'estimation
  - Opérateurs mathématique

Descripteurs audio

[G. Peeters. A large set of audio features for sound description (similarity and classification) in the cuidado project. Cuidado project report, Ircam, 2004.]

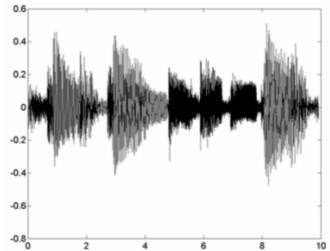
- Différentes formes :
  - scalaire : Centroide spectral, étendue spectrale, fréquence fondamentale, spectral roll-off, spectral flux, zero-crossing rate, RMS, ...
  - vecteur : Mel Frequency Cesptral Coefficients, coefficients LPC, coefficients PLP ...
- Différentes temporalité :
  - représente une trame du signal audio → descripteurs "instantanés"
  - représente le résumé du contenu d'un ensemble local de trame → texture windows
  - représente globalement le signal audio
- Mise en évidence de différents contenus (, harmonique, bruité, ...)
  - contenu timbral : Mel Frequency Cesptral Coefficients, coefficients LPC, coefficients PLP ...
  - contenu harmonique : Pitch Class Profiles/ Chroma ...
  - contenu bruité : Spectral Flatness Measure
  - contenu rythmique : ...

Descripteurs audio

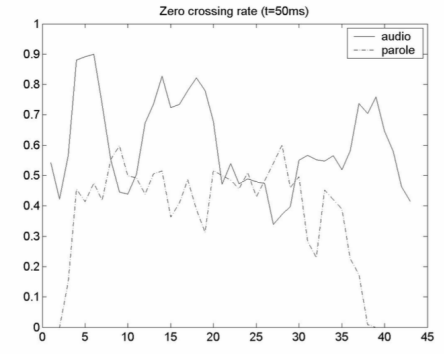


Taux de passage par zéro / zero-crossing rate (zcr)

- Mesure le nombre de fois que la forme d'onde croise l'axe zéro
  - $zcr = 0.5 \sum_{n=1}^{N} |sign(x(n)) - sign(x(n-1))|$
- Utilisation :
  - permet de distinguer les signaux bruités → zcr élevé
  - permet de distinguer les signaux harmoniques → zcr bas



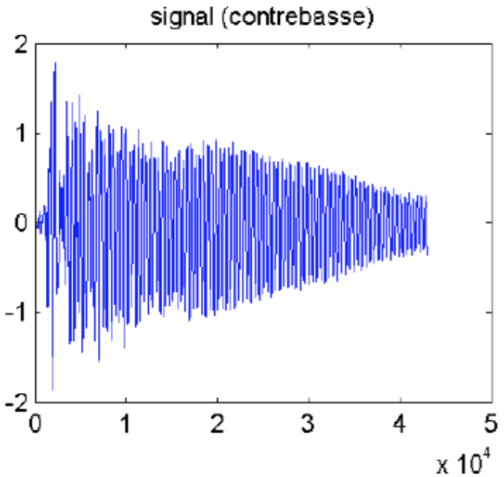
source : Gaël Richard

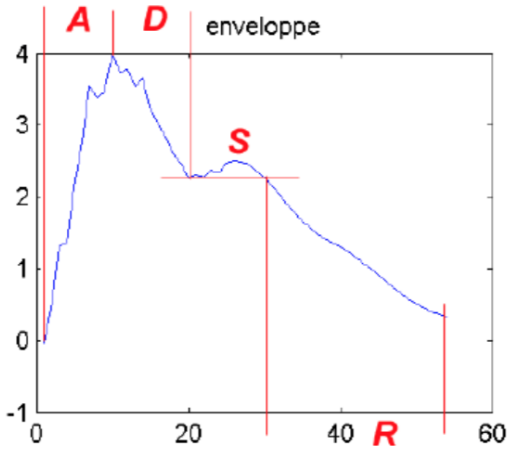


source : Gaël Richard

Enveloppe ADSR (Attack, Decay, Sustain, Release)

- Modèle réprésentant l'évolution (l'enveloppe) d'énergie d'une note de musique
- Utilisation :
  - permet de distinguer les attaques rapides (sons percussifs) / lentes
  - permet de distinguer les décroissances rapides (sons non-tenus) / lentes (sons tenus)



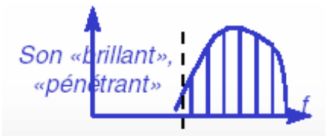


source : Gaël Richard

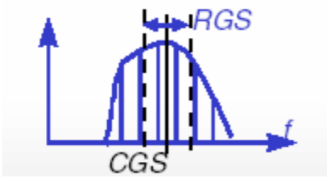
Description du spectre (barycentre, étendue spectral)

- **Centroid spectral**
  - $cs = \frac{\sum_k f_k A_k}{\sum_k A_k}$
  - Utilisation :
    - permet de distinguer les sons terne des sons brilliant
- **Etendue spectral**
  - $es = \sqrt{\frac{\sum_k (f_k - cs)^2 A_k}{\sum_k A_k}}$
  - Utilisation :
    - permet de distinguer les sons pauvres des sons riches
- **Flux spectral**
  - Mesure la variation temporel du spectre
  - $fs = \sum_k (A_k(t) - A_k(t-1))^2$
  - Utilisation :
    - permet de distinguer les sons pauvres des sons riches

source : Gaël Richard



source : Gaël Richard



source : Gaël Richard

2- Descripteurs audio

2.5- Mel Frequency Cepstral Coefficients (MFCCs)

Cepstre complexe $c(\tau)$

Objectif

- décrire la forme du spectre (du timbre) d'un signal à l'aide d'un nombre réduit de coefficients

Cepstre complexe $c(\tau)$

$$\begin{aligned} c(\tau) &= TF^{-1}\left[\log(X(\underline{w}))\right] \\ &= \frac{1}{2\pi}\int_{\omega} \log(X(\underline{w}))e^{j\omega\tau}d\omega \end{aligned}$$

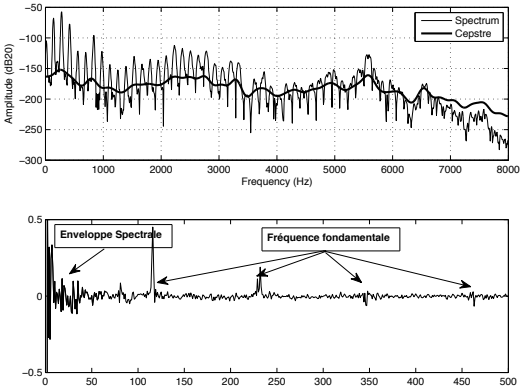
- τ est appelé "céfrence"
- $x(t) \xrightarrow{TF} X(\omega) \xrightarrow{\log} \log(X(\omega)) \xrightarrow{TF^{-1}} c(\tau)$

Modèle source/ filtre

- Source : signal périodique
- Filtre : résonant/ anti-résonant

$$x(t) = e(t) \circledast g(t)$$

$$\xrightarrow{TF} X(\underline{w}) = E(\underline{w}) \cdot G(\underline{w})$$



$$\xrightarrow{\log} \log(X(\underline{w})) = \underbrace{\log(E(\underline{w}))}_{\text{variation rapide à travers } \underline{w}} + \underbrace{\log(G(\underline{w}))}_{\text{variation lente à travers } \underline{w}}$$

$$\xrightarrow{TF^{-1}} TF^{-1}\left[\log(X(\underline{w}))\right] = \underbrace{TF^{-1}\left[\log(E(\underline{w}))\right]}_{\text{énergie aux céfrences } \tau>>} + \underbrace{TF^{-1}\left[\log(G(\underline{w}))\right]}_{\text{énergie aux céfrences } \tau<<}$$

Cepstre réel

- Cepstre calculé sur la partie réelle du log-spectrum

$$
\begin{aligned}
X(\underline{w}) &= A(\underline{w}) \cdot e^{j\phi(\underline{w})} \\
\log(X(\underline{w})) &= \log(A(\underline{w})) + j\phi(\underline{w}) \\
\Re(\log(X(\underline{w}))) &= \log(A(\underline{w}))
\end{aligned}
$$

$$
\begin{aligned}
\text{cepstre réel} &= TF^{-1}\left[\Re(\log(X(\underline{w})))\right] \\
&= TF^{-1}\left[\log(A(\underline{w}))\right] \\
c(\tau) &= \frac{1}{2\pi}\int_{\omega} \log(A(\underline{w}) e^{j\omega\tau} d\omega
\end{aligned}
$$

- Le spectre d'amplitude étant réel etsymétrique
  - sa TF se réduit à sa partie réelle
    - donc à la projection de $\log(A(\underline{w}))$ sur un ensemble de cosinus $\rightarrow$ DCT

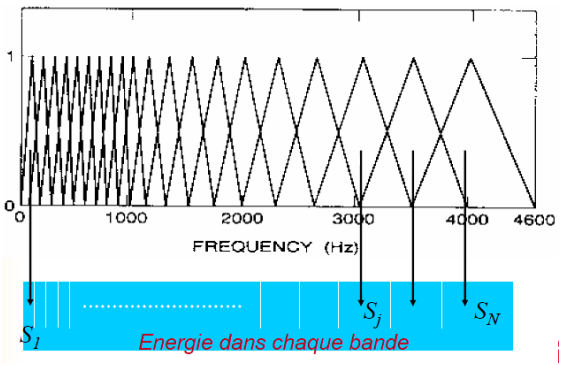
Mel Frequency Cepstral Coefficients (MFCCs)

- Cepstre réel calculé sur un spectre d'énergie exprimé en convertissant l'énergie $|X(\underline{w})|^2$ en échelle perceptive (échelle de Mel)
- Pouquoi ?
  - La transformée de Fourier :
    - décomposition sur une série de sinusoides linéairement espacées (10Hz, 20Hz, 30Hz, ... Hz)
  - L'oreille :
    - décomposition sur une série de filtres de fréquences logarithmiquement espacé (10, 20, 40, 80, ... Hz).
    - meilleure résolution en basses fréquences que en hautes fréquences.
    - résonances de l'enveloppe spectrale sont plus rapprochées en basse fréquence.
  - MFCCs permet une représentation plus compacte que le cepstre réel
- Comment ?
  - On utilise des échelles dites perceptives : échelles de Mel, de Bark, filtres ERB, Gamma tone
- Utilisation ?
  - Les coefficients les plus utilisés dans le monde de la reconnaissance audio : parole, musique, sons environnementaux, ...

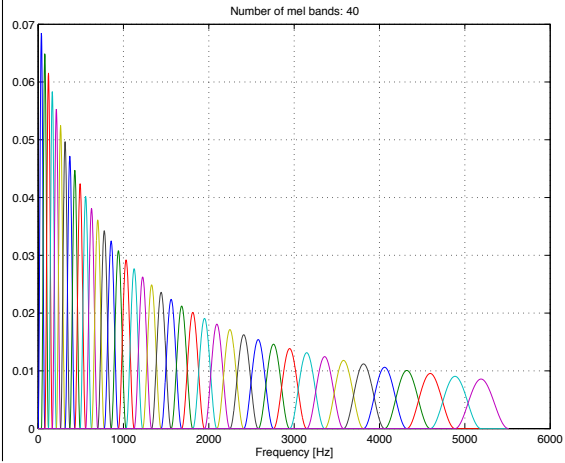
Echelle de Mel

$$M = f \quad \text{pour} \quad f < 1000Hz$$

$$M = f_c \left(1 + \log_{10}\left(\frac{f}{f_c}\right)\right) \quad \text{pour} \quad f \geqslant 1000Hz$$

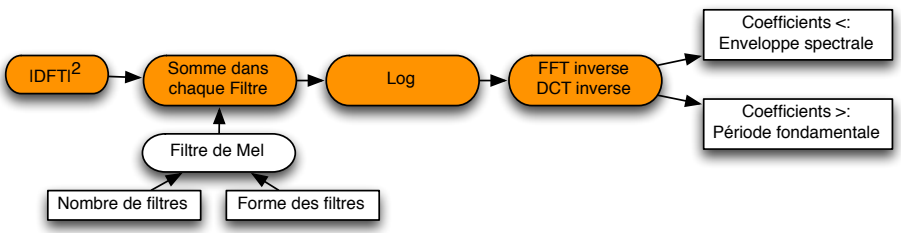


source : Gaël Richard

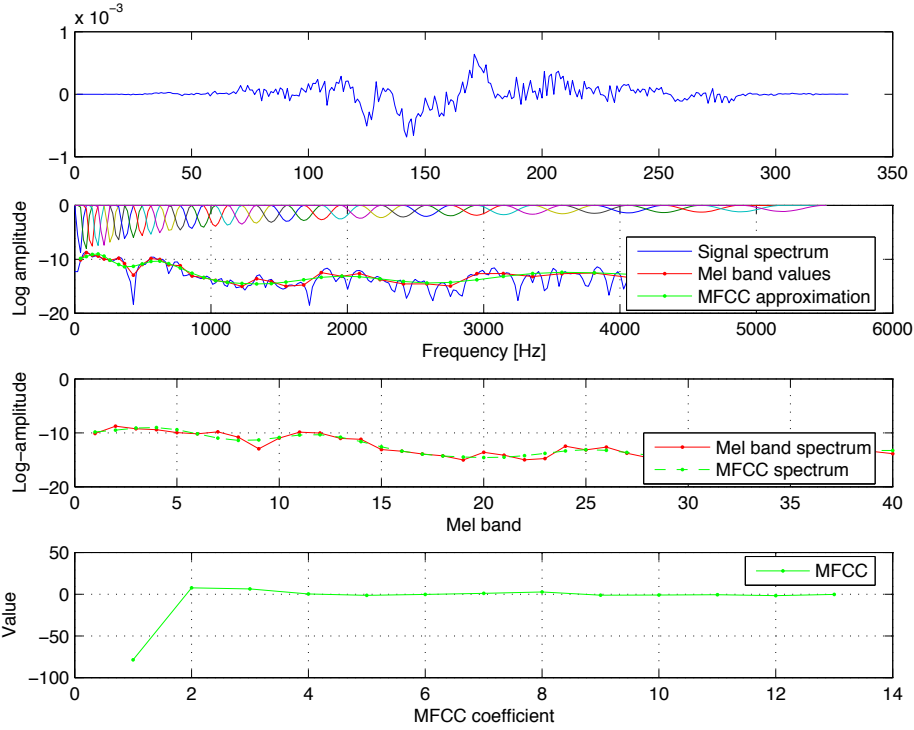


Echelle de Mel

- Calcul du spectre de puissance : $|X(\underline{w})|^2$
- Calcul des filtres de Mel : $H_b(\underline{w})$ avec $b \in [1, B]$
  - choix du nombre de filtres B : 40
  - choix de la forme des filtres : triangulaire, hanning, tanh, ...
- Conversion du spectre de puissance en bandes de Mel : $S(b) = \sum_{\underline{w}} |X(\underline{w})|^2 \cdot H_b(\underline{w})$
- Passage en échelle logarithmique : $\log(S(b))$
- Calcul de la IFFT (ou de la IDCT) :
- Sélection des coefficients de la IDCT proches de zéro (jusqu'à 13)
  - les coefficients proches de zéro représentent la décomposition du spectre en échelle de Mel sur un ensemble de cosinus à variation lente



Exemple de calcul de MFCCs

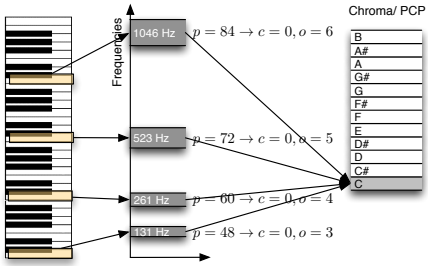


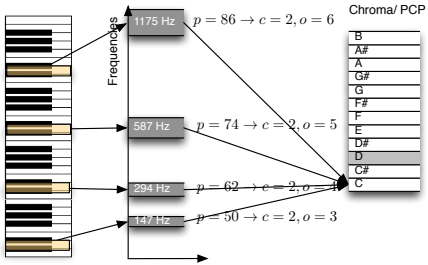
2- Descripteurs audio

2.6- Chroma - Pitch Class Profile (PCP)

Définition des Chroma - Pitch Class Profile (PCP)

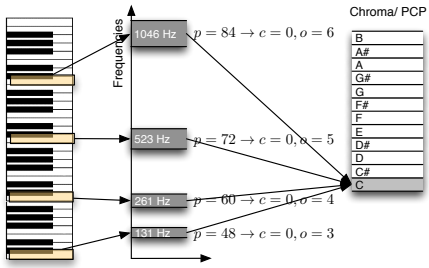
- Objectif :
  - le spectre à l'instant n : $X(k, n)$
  - représenter son contenu harmonique sous forme d'un vecteur : $C(c, n)$ $c \in [0, 12[$
- Utilisations :
  - reconnaissance de tonalité,
  - reconnaissance de suite d'accords,
  - détection de "cover versions"
- Shepard-1964 :
  - représenter la hauteur d'une note p comme une structure bi-dimensionelles :
  - $p = c + o \cdot 12$
    - le chroma c (classe de hauteur).
    - la hauteur tonale o (numéro d'octave),

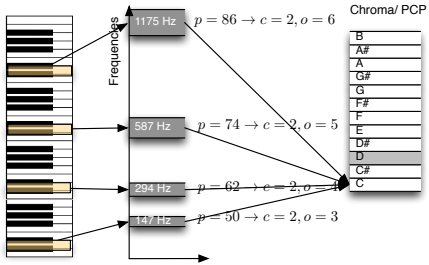




Calcul des Chromas - Pitch Class Profile (PCP)

- Relation entre les fréquences f_k de la DFT et les hauteurs de note p (hauteurs de demi-tons en échelle de notes MIDI)
  - $p(f_k) = 12 \log_2 \left(\frac{f_k}{440}\right) + 69, \quad p \in \mathbb{R}^+$
  - $f(p) = 440 \cdot 2^{\frac{p-69}{12}}$
- Calcul des chromas $C(c, n)$
  - On additionne toutes les valeurs du spectre $X(k, n)$ tel que f_k correspondent à un c donné
  - Hard-mapping
  - Soft-mapping

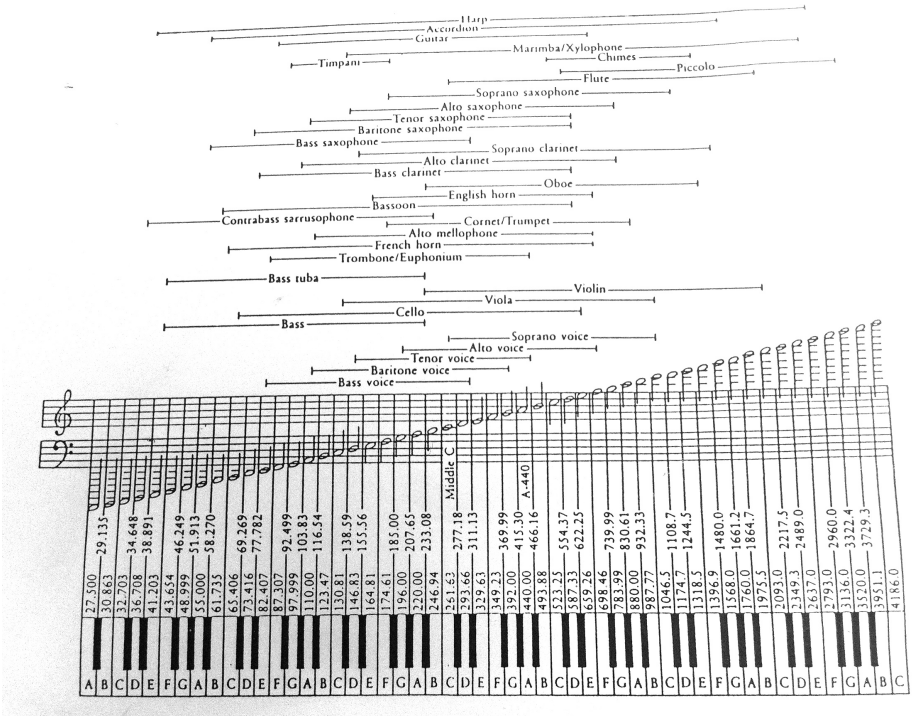




Calcul des Chromas - Pitch Class Profile (PCP)

- Résolution fréquentielle ?
  - Elle doit permettre la séparation des notes voisines
  - On définit la largeur (à -6 dB) : $Bw = \frac{Cw}{L_{sec}}$
  - Si $f_{\min}$ (la fréquence la plus basse considérée dans le sectre) est 50 Hz
    - on veut séparer G#1 (51.91Hz) et A1 (55Hz) $\rightarrow L_{sec} = \frac{Cw}{Bw} = \frac{2.35}{3.0869Hz} = 0.7613s$
  - Si $f_{\min}$ est 100 Hz
    - on veut séparer G#2 (103.82Hz)de A2 (110Hz) $\rightarrow L_{sec} = \frac{Cw}{Bw} = \frac{2.35}{6.1738Hz} = 0.3806s$
- Deux possibilités :
  - Choisir L_{sec} en fonction $f_{\min}$
  - Choisir $f_{\min}$ en fonction de L_{sec}

Calcul des Chromas - Pitch Class Profile (PCP)

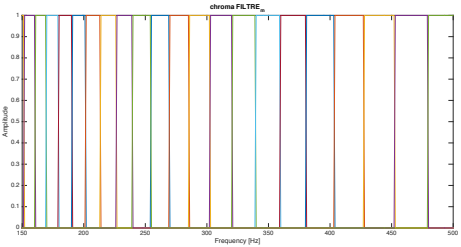


Calcul des Chromas - Pitch Class Profile (PCP)

- Calcul des chromas $C(c, n)$
  - On additionne toutes les valeurs du spectre $X(k, n)$ tel que f_k correspondent à un c donné

Calcul des Chromas - Pitch Class Profile (PCP)/ Hard-mapping

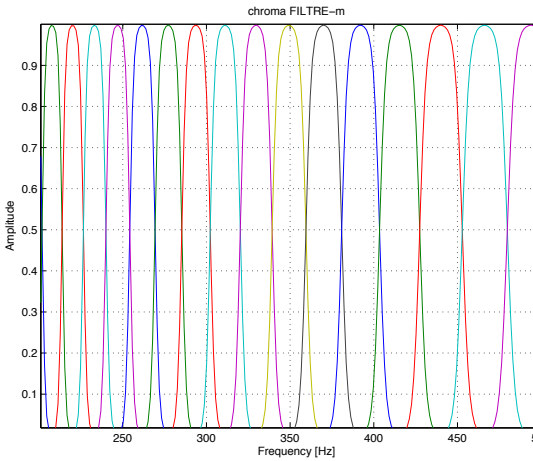
- Hard-mapping ?
  - Une fréquence f_k de la DFT contribue uniquement à la note la plus proche
  - Par exemple,
    - l'énergie à f_k=452 Hz ($p(f_k)$=69.4658) contribue entièrement à la note p=69 (c=10)
    - alors que f_k=453 Hz ($p(f_k)$=69.5041) à p=70 (c=11).
  - Création d'un banc de filtres $H_{p'}$ centrés sur les hauteurs de demi-tons $p' \in [43, 44, \ldots, 95]$:



- Soft-mapping ?
  - Une fréquence f_k de la DFT contribue à différents chroma avec un poids inversement proportionnel à la distance entre $p(f_k)$ et les p les plus proches
  - Par exemple,
    - l'énergie à f_k=452 Hz ($p(f_k)$=69.4658) contribuera de manière presque égale à p=69 (c=10) et p=70 (c=11).
  - Création d'un banc de filtres $H_{p'}$ centrés sur les hauteurs de demi-tons $p' \in [43, 44, \ldots, 95]$:
    - Chaque filtre est défini par la fonction

$$H_{p'}(f_k) = \frac{1}{2}\tanh\left(\pi(1-2x)\right) + \frac{1}{2}$$

      dans lequel x= distance relative entre centre du filtre et fréquences de la TF $x = R\,|p' - p(f_k)|$.
    - Les filtres sont équi-répartis et symétriques sur l'échelle logarithmique des hauteurs de demi-tons, non-nulles entre $p'-1$ et $p'+1$ et à valeur maximale en p'.



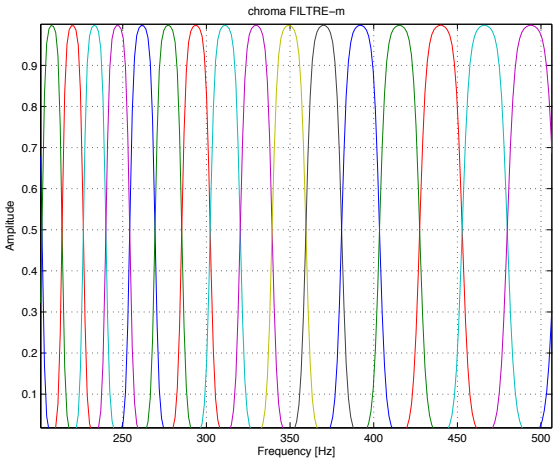
Calcul des Chromas - Pitch Class Profile (PCP)/ Soft-mapping

- La valeur du spectre de hauteur de demi-ton $N(n')$ est obtenue en multipliant les valeurs de la transformée de Fourier $A(f_k)$ par l'ensemble des filtres $H_{n'}$:

$$P(p') = \sum_{f_k} H_{p'}(f_k) A(f_k)$$

- Le mapping entre les hauteurs de demi-tons n et les classes de hauteurs de demi-ton (chroma) c est défini par $c(p) = mod\,(p, 12)$.
- La valeur du vecteur de chroma est obtenue en additionnant les valeurs de classes de hauteur équivalentes

$$C(c) = \sum_{p' \text{ tel que } c(p')=I} P(n') \quad c \in [0, 12[$$



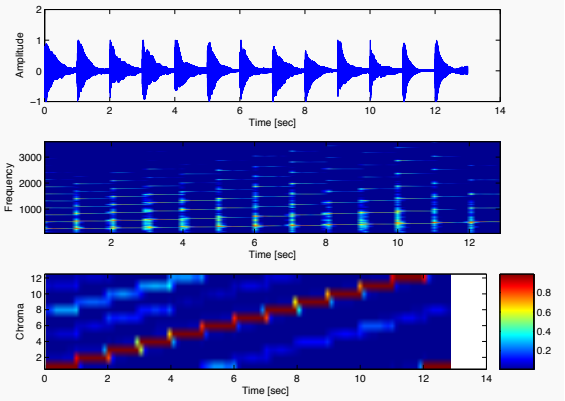
Limitations des Chromas - Pitch Class Profile (PCP)

- Présence des harmoniques supérieures de chaque note
  - En pratique pour une note C on a pas $[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$
  - mais plutôt $[a_1 + a_2 + a_4, 0, 0, 0, a_5, 0, 0, a_4, 0, 0, 0, 0]$
- Influence de l'enveloppe spectrale

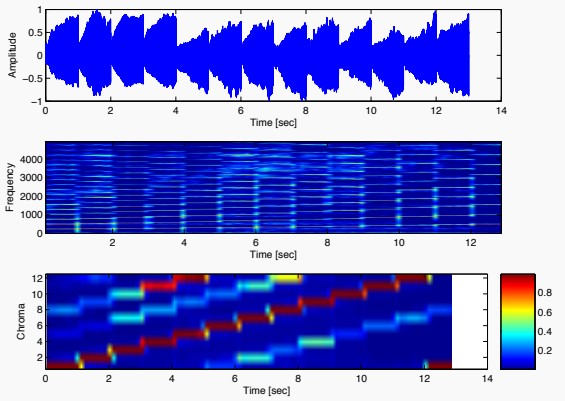
Pitch	Harmonic	Frequency f_μ	MIDI-scale m_μ	Chroma/PCP p
c3	f_0	130.81	48	1 (=c)
	$2f_0$	261.62	60	1 (=c)
	$3f_0$	392.43	67.01	8.01 ($\simeq$ g)
	$4f_0$	523.25	72	1 (=c)
	$5f_0$	654.06	75.86	4.86 ($\simeq$ e)
	...	...	...	...

Limitations des Chromas - Pitch Class Profile (PCP)

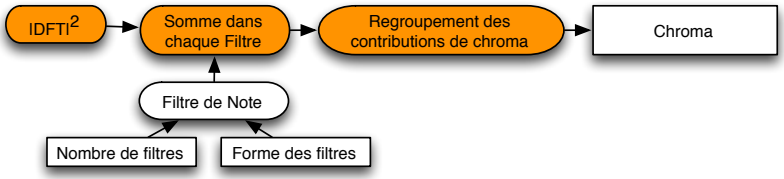
Exemple piano

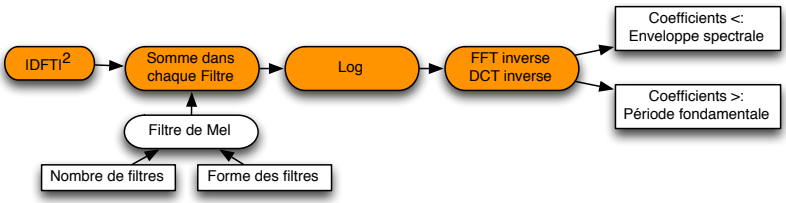


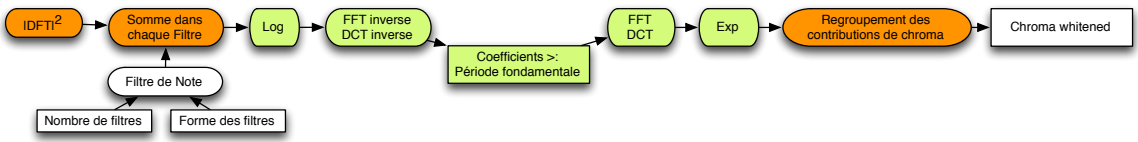
Exemple violon



Variante du calcul des Chromas - Pitch Class Profile (PCP) : blanchissement/ whitening







Spectral Flatness Measure (SFM)

- **Objectif :** distinguer la présence de contenu harmonique ou bruité dans chaque bande
  - avec les MFCCs/PCP même valeur si le contenu est harmonique ou bruité dans une bande du spectre
- **Spectral Flatness Measure** : mesure de la platitude d'une bande du spectre
  - Si la bande du spectre contient du bruit → spectre plat (flat)
  - Si la bande du spectre contient des sinusoides → spectre avec des piques (peaky)
  - Calcul [**?**] : rapport moyenne géométrique / moyenne arithmétique

$$SFM = \frac{\left(\prod_{k \in K} a(k)\right)^{1/K}}{\frac{1}{K}\sum_{k \in K} a(k)} \tag{1}$$

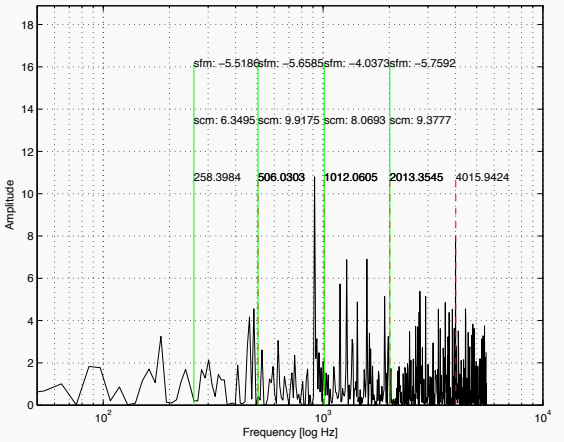
    - SFM$\simeq$0 pour signaux tonaux, SFM$\simeq$1 pour signaux bruités
    - Calcul effectué dans plusieurs bandes de fréquence : - - $[250 - 500]$, $[500 - 1000]$, $[1000 - 2000]$, $[2000 - 4000]$ Hz (MPEG-7)
- **Mesure de tonalité** :

$$SFM_{dB} = 10 \log_{10}(SFM) \qquad Tonality = \min\left(\frac{SFM_{dB}}{-60}, 1\right) \tag{2}$$

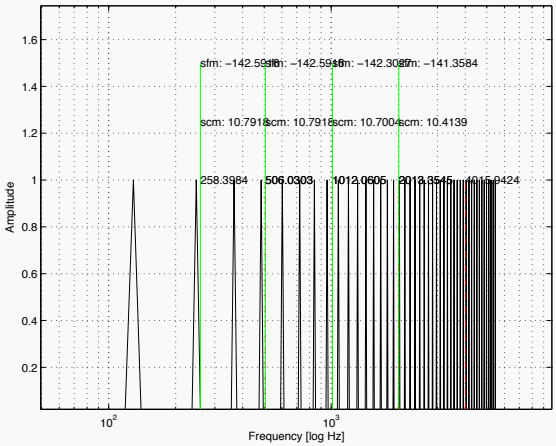
  - Tonality$\simeq$0 pour signaux bruités, Tonality$\simeq$1 pour signaux tonaux

Spectral Flatness Measure (SFM)

Exemple cas bruité



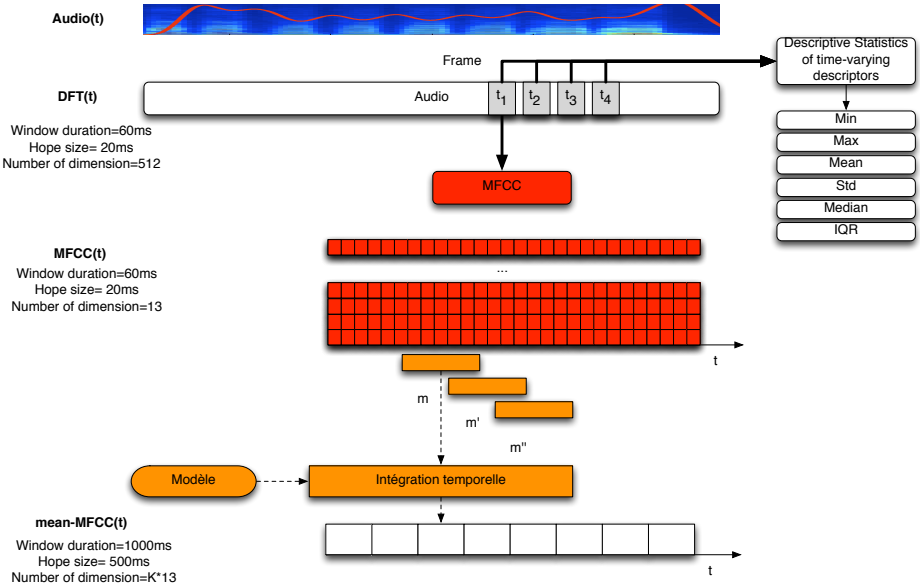
Exemple cas non-bruité



Intégration temporelle des descripteurs audio

- Objectifs :
  - Représenter le comportement temporel des observations
    - Calcul des dérivées et accélérations des observations (Δ-MFCC, Δ-Δ-MFCC) $\rightarrow$ permet de représenter le comportement temporel du descripteur au cours du temps
  - Réduire la quantité de données à traiter
    - Si le pas d'avancement = 20 ms, un morceau de 4 m. = 12.000 trames
    - $\rightarrow$ matrice d'auto-similarité = 12.000 x 12.000 $\rightarrow$ c'est beaucoup !
- Intégration sans-modèles
  - Analyse des descripteurs sur une fenêtre de durée plus longue (0.5 s., 1 s., ...)
    - Calcul des moments statistiques (μ, σ) de chaque dimension k d'un descripteur (chaque coefficient MFCC, PCP, SFM, ...)
    - modulation spectrum, scattering transform
    - modèles AR
- Intégration avec modèles
  - Multi-prob histogram
  - Universal Background Model, iVector

Intégration temporelle des descripteurs audio



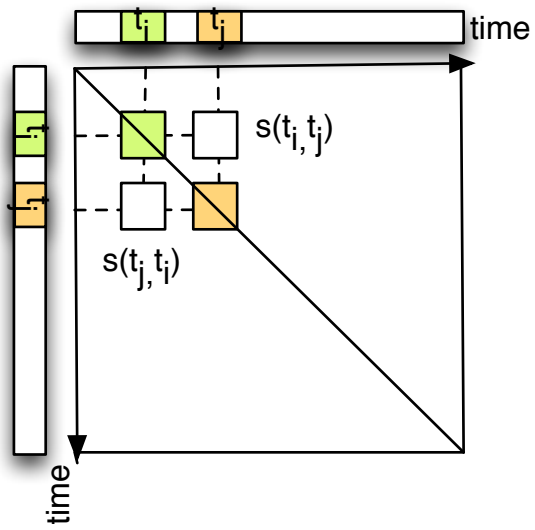
Représentation visuelle de la structure temporelle de la musique

La matrice d'auto-similarité

- Similarité entre deux instants t_i et t_j
- Similarité entre les observations du signal aux trames i et j : $s(t_i, t_j) = s(\underline{d}^i, \underline{d}^j)$
- Matrice d'auto-similarité= les valeurs $s(t_i, t_j)$ sont représentées sous forme d'une matrice $\underline{\underline{S}} = s(t_i, t_j) \ \ \forall i, j$

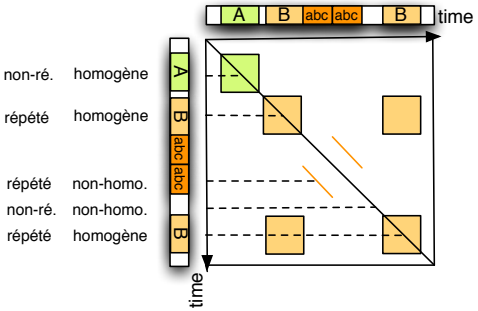
Lecture/ interprétation

- Une valeur élevée dans $S(t_i, t_j)$ représente une similarité importante entre les instants t_i et t_j.
- Si $t_i \simeq t_{i+1} \simeq t_{i+2}$ nous observons un bloque homogène
- Si une séquence de temps $t_i, t_{i+1}, t_{i+2}, ...$ est similaire à une séquence de temps $t_j, t_{j+1}, t_{j+2}, ...$ nous observons une diagonale supérieure/ inférieure dans S.

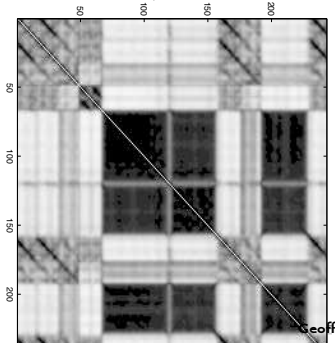


Hypothèse 1 : homogénéité

- Hypothèse :
  - le morceau est formé d'une succession de segments temporels homogènes
    $t_i \simeq t_{i+1} \simeq t_{i+2}, \ldots$ et de segments non homogènes
- Homogène ?
  - contenant une information similaire au sens d'un critère d'observation)
  - "A" et "B" sur la Figure
- Exemple :
  - arrangements d'un couplet ou d'un refrain
- Méthode :
  - approche par "état"

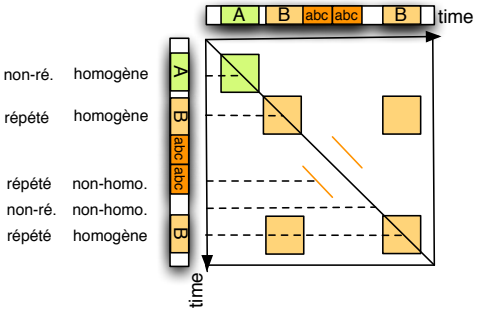


	homogène	non-homogène
répété	état	séquence
non-répété	état	état

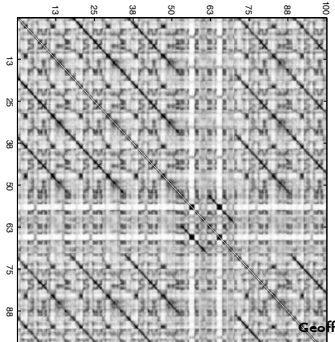


Hypothèse 2 : répétition

- Hypothèse :
  - le morceau renferme des répétitions temporelles
- Répétition ?
  - elles peuvent correspondre à des répétitions de segments homogènes
    - $\{t_j, t_{j+1}, t_{j+2}\} \simeq \{t_i, t_{i+1}, t_{i+2}\}$ et $t_i \simeq t_{i+1} \simeq t_{i+2}$
    - "B" dans la figure
    - Méthode : approche par "état"
  - elles peuvent correspondre à des répétitions de segments non homogènes
    - $\{t_j, t_{j+1}, t_{j+2}\} \simeq \{t_i, t_{i+1}, t_{i+2}\}$ et $t_i \neq t_{i+1} \neq t_{i+2}$
    - séquence "abc" dans la Figure
- Méthode :
  - approche par "séquence"



	homogène	non-homogène
répété	état	séquence
non-répété	état	état

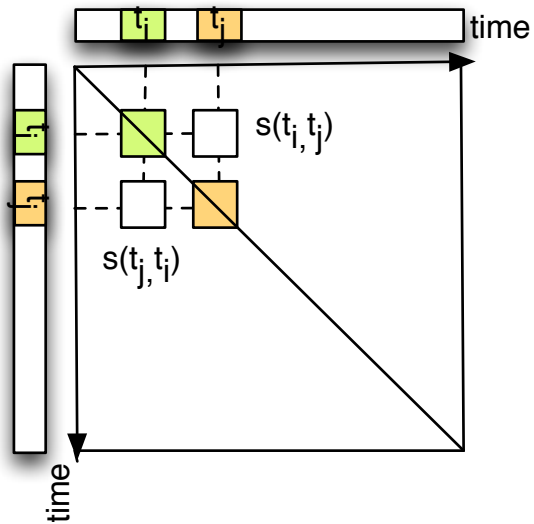


Matrice d'auto-similarité/distance (temps,temps)

- Similarité entre deux instants t_i et t_j
- Similarité entre les observations du signal à deux trames i et j : $s(t_i, t_j) = s(\underline{d}^i, \underline{d}^j)$
- Descripteurs audio multi-dimensionnels $\underline{d} = d_k \; k \in K$

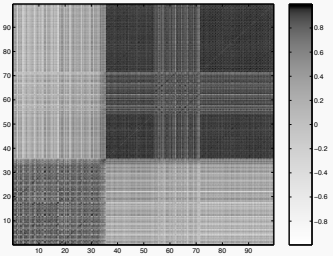
Choix d'une distance

- Distance euclidéenne : $\sqrt{\sum_k (d_k^i - d_k^j)^2}$
- Corrélation : $\sum_k (d_k^i \cdot d_k^j)$
- Distance cosinusoidale : $\frac{\sum_k (d_k^i \cdot d_k^j)}{\sqrt{\sum_k (d_k^i)^2}\sqrt{\sum_k (d_k^j)^2}}$
- Correlation Pearson : $\frac{\sum_k (d_k^i - \mu^i) \cdot (d_k^j - \mu^j)}{\sqrt{\sum_k (d_k^i - \mu^i)^2}\sqrt{\sum_k (d_k^j - \mu^j)^2}}$
- ...

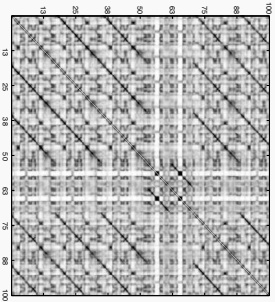


Matrice d'auto-similarité/distance (temps,temps)

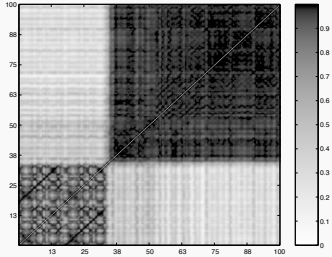
MFCC



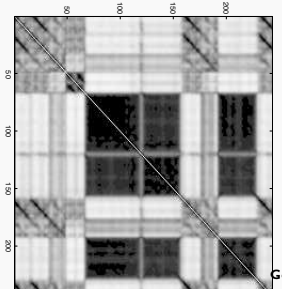
Chroma



Modulation spectrum 1

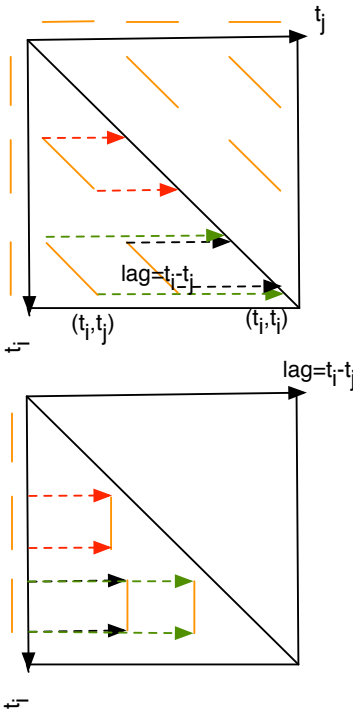


Modulation spectrum 2



Matrice d'auto-similarité/distance (temps,lag)

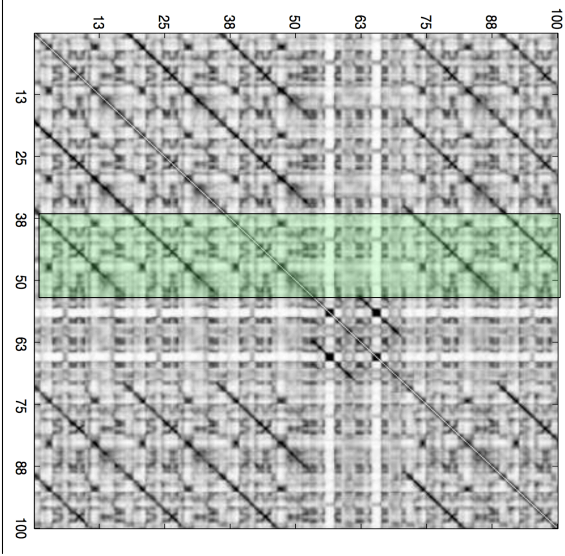
- Une valeur élevée dans $S(t_i, t_j)$ représente une similarité importante entre les instants t_i et t_j.
- Si une séquence de temps $t_i, t_{i+1}, t_{i+2}, ...$ est similaire à une séquence de temps $t_j, t_{j+1}, t_{j+2}, ...$ nous observons une diagonale supérieure/ inférieure dans S.
- Lag = distance entre la répétition (démarrant en t_i) et la séquence originale (démarrant en t_j)
  - cette distance est donnée par la projection de t_i sur la diagonale principale de la matrice : $t_i - t_j$
  - souvent constante
- Matrice de lag :
  $L = L(t_i, lag_{ij}) = S(t_i, t_i - t_j)$
  - les diagonales dans une matrice (temps,temps)
  - deviennent des lignes verticales dans une matrice (temps,lag)



Génération de résumé audio par méthode du "summary score"

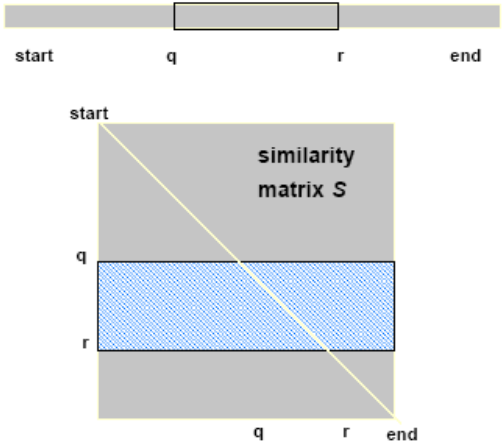
[M. Cooper and J. Foote. Automatic music summarization via similarity analysis. In Proc. of ISMIR, Paris, France, 2002.]

- Recherche du segment temporel continu représentant au mieux le contenu d'un morceau de musique selon un critère de similarité → création de "previews" musicaux



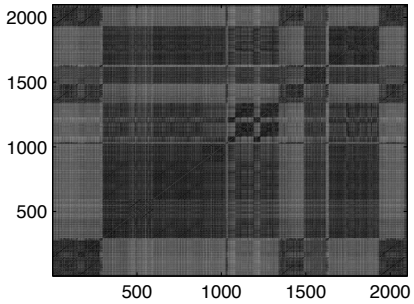
Génération de résumé audio par méthode du "summary score"

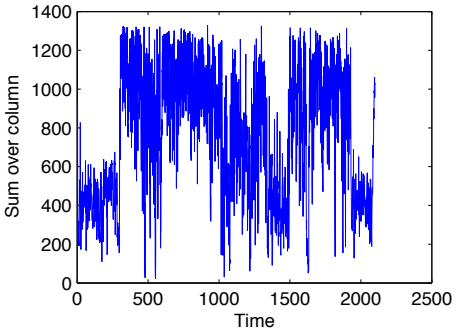
- Recherche du segment démarrant en q de durée $L = r - q$ expliquant le maximum de répétitions
- Similarité moyenne de l'instant q avec tous les temps du morceau
  - $\frac{1}{N} \sum_{n=1}^{N} S(q, n)$
- Similarité moyenne du segment $[q, r]$ (de longueur $L = r - q$) avec tous les temps du morceau
  - $s(q, L) = \frac{1}{LN} \sum_{m=q}^{r} \sum_{n=1}^{N} S(m, n)$
- Pour un L donné, nous cherchons q maximisant $s(q, L)$
  - $q_L = \text{argmax}_{1 \leqslant i \leqslant N-L} s(i, L)$
- Variante : pour favoriser la détection de résumés en début de morceau,
  - ajout d'une pondération $w(n)$ fonction décroissante du temps
  - $s(q, L) = \frac{1}{LN} \sum_{m=q}^{r} \sum_{n=1}^{N} w(n) S(m, n)$

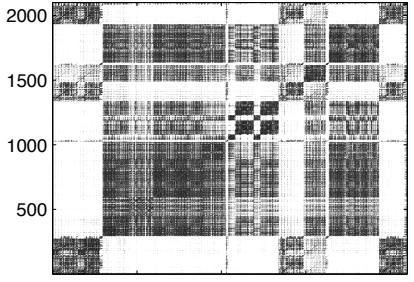


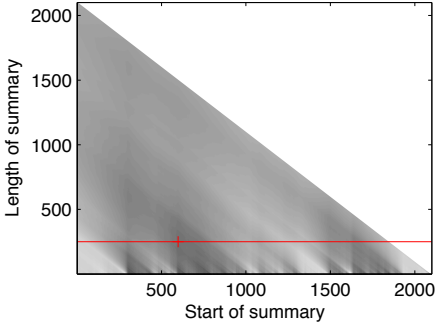
source : [Cooper and Foote, 2002, ISMIR]

Génération de résumé audio par méthode du “summary score"









4- Segmentation temporelle d'un flux de descripteurs

Segmentation trame-à-trame

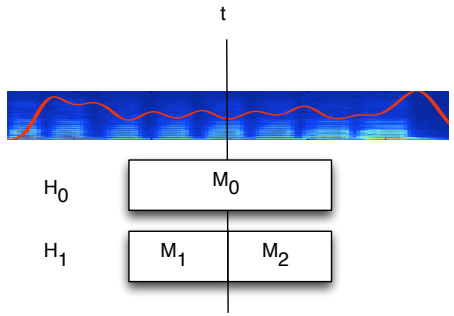
- Variation trame-à-trame de $\underline{d}^t$

Critère BIC (Bayes Information Criteria)

- Pour chaque temps t (potentiellement instant de rupture) on compare deux hypothèses
  - H0 : le signal obéit au même modèle probabiliste de part et d'autre de t, modèle noté $M_0(\mu_0, \Sigma_0)$
  - H1 : il y a un changement de modèle en t, deux modèles différents $M_1(\mu_1, \Sigma_1)$ et $M_2(\mu_2, \Sigma_2)$
- Critère Delta BIC

$$\Delta BIC = R(t) - \lambda P$$

$$R(t) = \frac{1}{2}(N \log(|\Sigma_0|) - t \log(|\Sigma_1|) - (N-t) \log(|\Sigma_2|)$$



- si $\Delta BIC > 0$, H1 est vérifiée
- paramètres :
  - P : proportionnel à la différence des nombres de paramètres estimés pour chaque hypothèse
  - λ : facteur de pénalité choisi tel que $\Delta BIC > 0$ si H1 est vérifiée

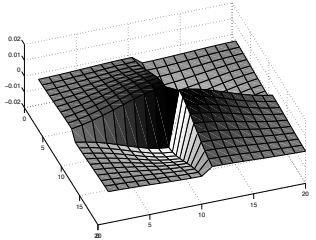
Convolution de la matrice d'auto-similarité par un noyau en damier

[J. Foote. Automatic audio segmentation using a measure of audio novelty. In Proc. of IEEE ICME, New York City, NY, USA, 2000.]

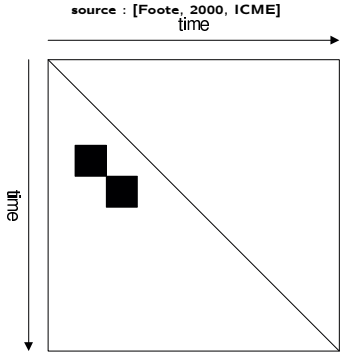
- Méthode "Novelty Curve" [Foote, 2000, ICME]
- Approche plus robuste
- Convolution de la matrice de similarité $\underline{\underline{S}}$ par un noyau prenant en compte
  - la similarité inter-segment (homogénéité) et
  - la dis-similarité entre **segments** gauches et droites
  - "checker-board" kernel :

$$C = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

  - La valeur de la diagonale de la matrice "filtrée" mesure la similarité/ dis-similarité des **régions** gauches et droites

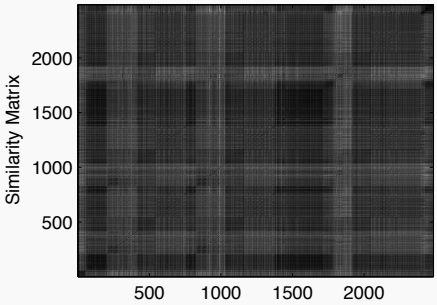


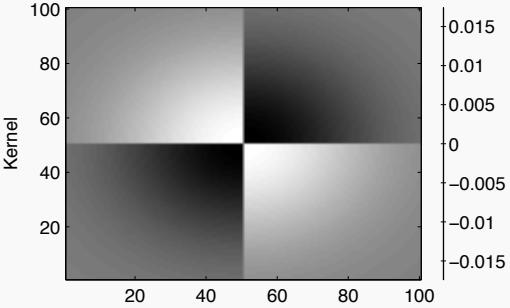
source : [Foote, 2000, ICME]

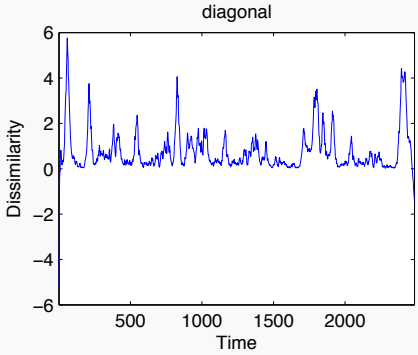


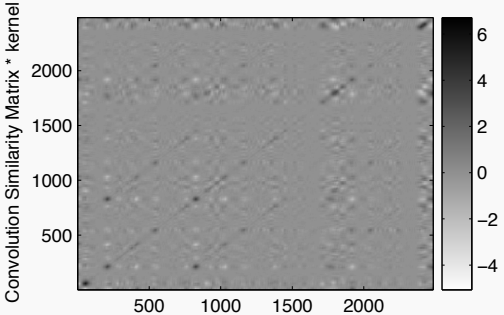
Convolution de la matrice d'auto-similarité par un noyau en damier

Exemple





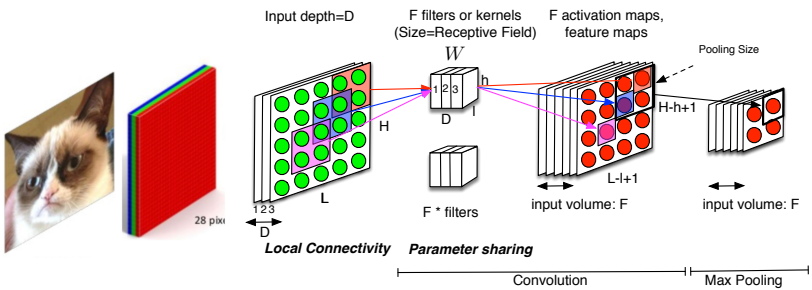




5- Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNN or ConvNet)

- **Goal ?** design Neural Networks specifically adapted to Computer Vision
  - Image = high-dimensional
    - $\Rightarrow$ 150x150 pixels $\Rightarrow n^{[0]} = 22.500$ (or 3 x 22.050 for RGB)
  - If $n^{[1]} = 512$ hidden units
    - $\Rightarrow \underline{\underline{W}}^{[0]}$ is (3 x 22.050, 512) = 33.868.800 weights to be learnt
- **How ?**
  - Reduce the number of weights to be learnt
  - Bring (use) invariance by translation : convolution
- **In practice ?**
  - Exploit the 2D topology :
    - local connectivity, parameter sharing, pooling



Local Connectivity, Parameter Sharing

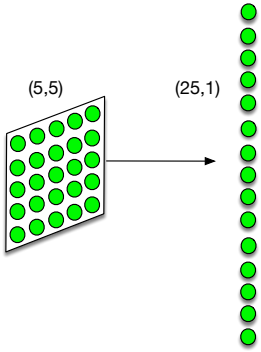
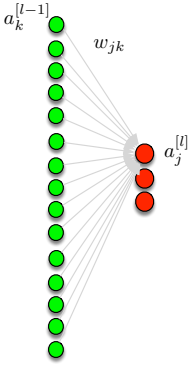
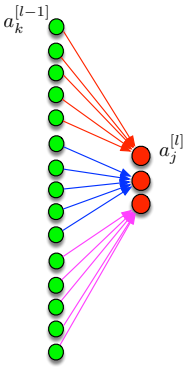


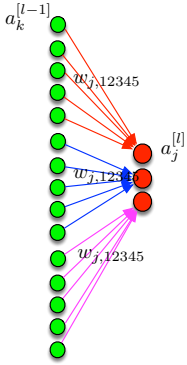
Image processing using MLP



MLP = Fully Connected
Each neuron at layer [l] is connected to all neurons at layer [l-1]



CNN = Local connectivity
Each neuron at layer [l] is only connected to a local part of the neurons at layer [l-1]



CNN = Parameter sharing
Weights to connect [l-1] to [l] are shared among the neurons of layer [l]

Convolution and Depth

For MLP

$$z_j^{[l]} = \sum_k w_{jk}^{[l]} a_k^{[l-1]} + b_j^{[l]}$$

$$a_j^{[l]} = \sigma(z_j^{[l]})$$

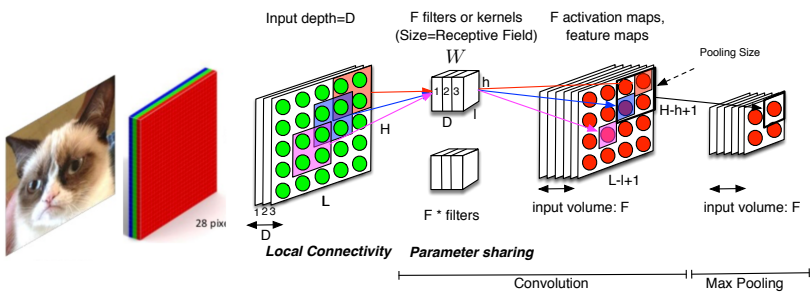
$$\delta_j^{[l]} = \frac{\partial C}{\partial z_j^{[l]}}$$

For CNN

- For one filter

$$z_{x,y}^{[l+1]} = w^{[l+1]} \circledast \sigma(z_{x,y}^{[l]}) + b_{x,y}^{[l+1]}$$
$$= \sum_{a,b} w_{a,b}^{[l+1]} \sigma(z_{x-a,y-b}^{[l]}) + b_{x,y}^{[l+1]}$$

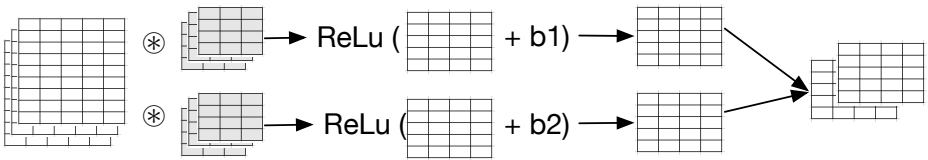
- The convolution is done over x and y (not over the volume)



Notations

- Notations (for Layer l) :
  - $f^{[l]}$: filter size
  - $p^{[l]}$: padding size
  - $s^{[l]}$: stride
- Inputs : $\underbrace{a^{[l-1]}}_{(n_H^{[l-1]}, n_W^{[l-1]}, n_C^{[l-1]})}$
  - for images : $n_C^{[0]} = 3$ (RBG channels)

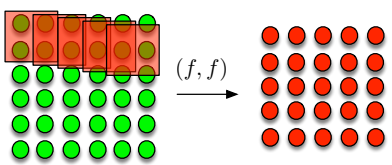
- Parameters : the set of $n_C^{[l]}$ filters or kernels $\underbrace{\underline{\underline{W^{[l]}}}}_{(f^{[l]}, f^{[l]}, n_C^{[l-1]})}$
- Output : $\underbrace{a^{[l]}}_{(n_H^{[l]}, n_W^{[l]}, n_C^{[l]})}$
  - where $n_C^{[l]}$ = the number of filters $\underline{\underline{W}}$ at layer l
  - $n_H^{[l]} = \lfloor \frac{n_H^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \rfloor$



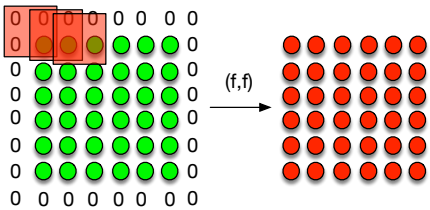
Stride and Padding

Stride s=1
Padding p=0 (valid)

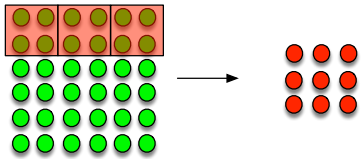
(n, n) $\quad (\frac{n+2p-f}{s} + 1, \frac{n+2p-f}{s} + 1)$



Stride s=1
Padding p=1(same)

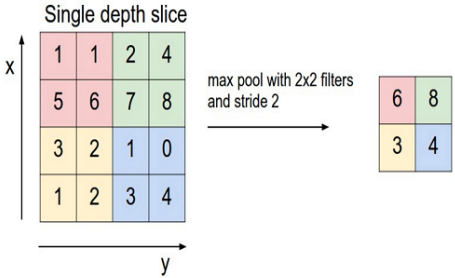


Stride s=2
Padding p=0 (valid)

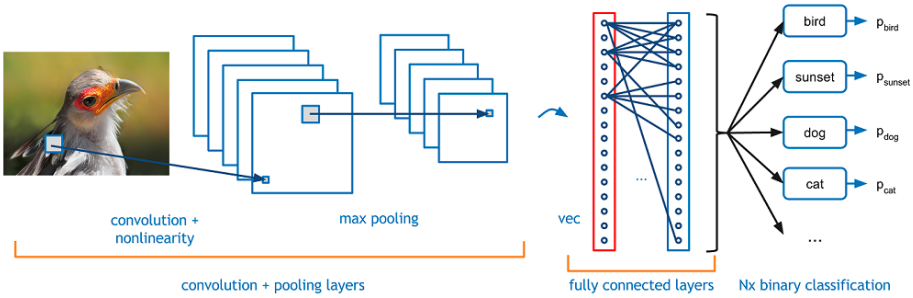


Pooling

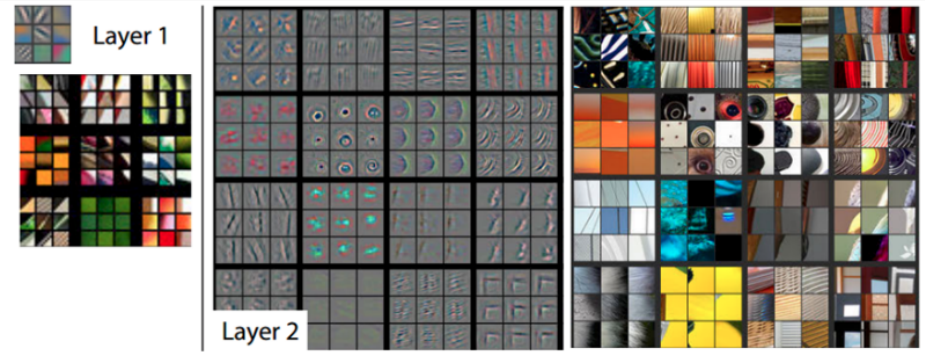
- Pooling (or sub-sampling) ?
  - keeping the maximum or the average of the values over a region
    - the region is defined by a size f
    - the way we move from region to region is defined by a stride s
    - usually $s = f$



The whole network= CNN + FC + SoftMax

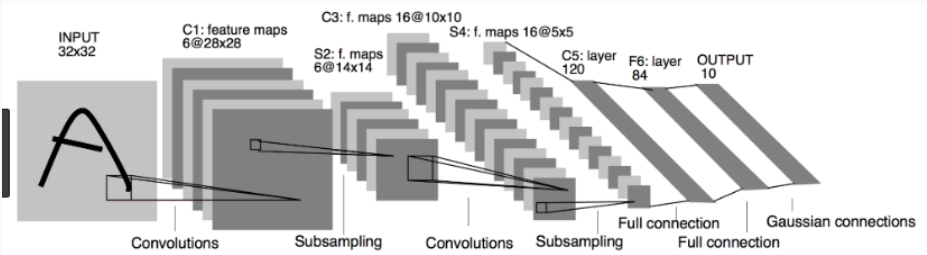


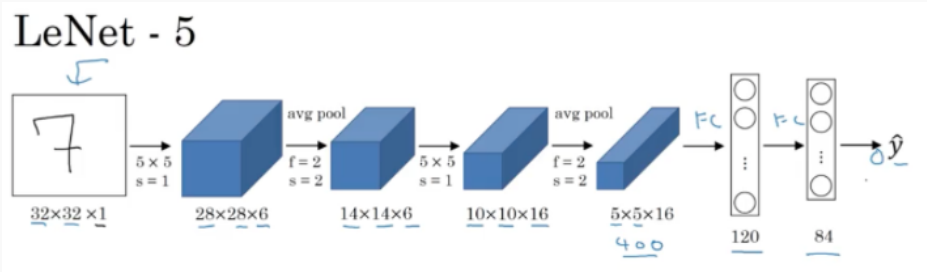
Using a DeconvNet



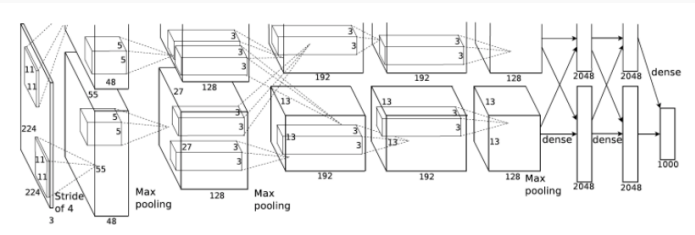
Visualizations of Layer 1 and 2. Each layer illustrates 2 pictures, one which shows the filters themselves and one that shows what part of the image are most strongly activated by the given filter. For example, in the space labled Layer 2, we have representations of the 16 different filters (on the left)

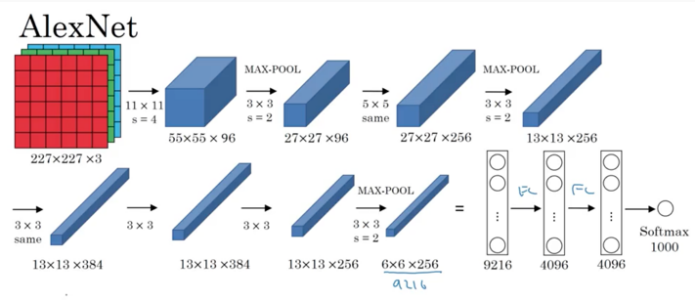
LeNet-5 [LeCun et al. 1998]



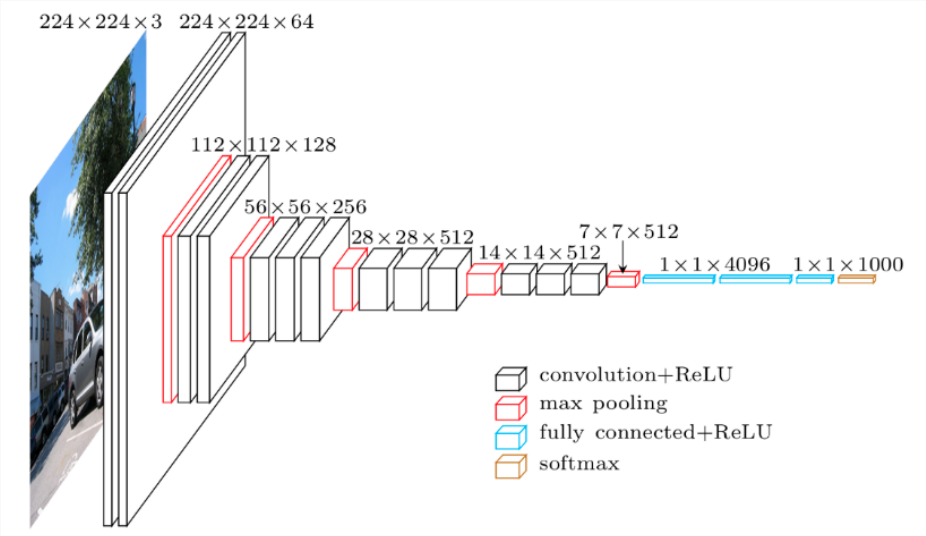


AlexNet [Alex Krizhevsky et al. 2012 ImageNet]





VGG16 [Simonyan, 2015 Very Deep]

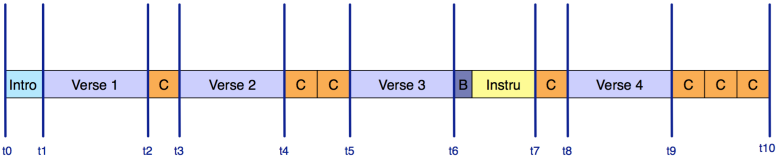


6- Structure estimation using the depth of Convolutional Neural Networks

Structure estimation using the depth of Convolutional Neural Networks

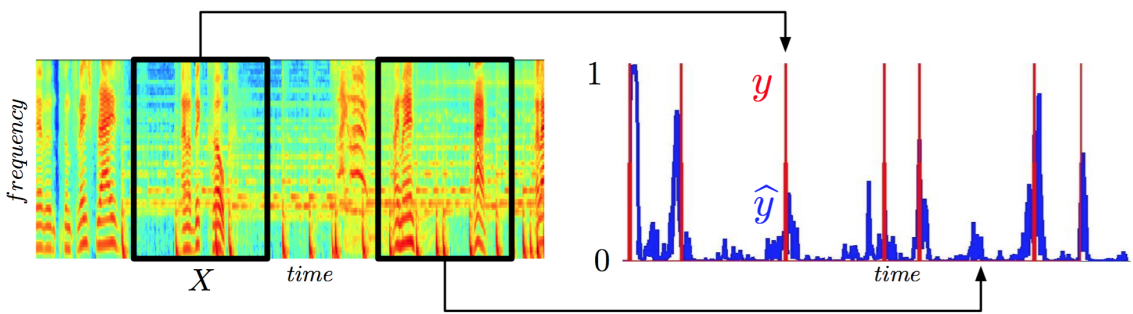
[Alice Cohen-Hadria and Geoffroy Peeters. Music structure boundaries estimation using multiple self-similarity matrices as input depth of convolutional neural networks. In AES Conference on Semantic Audio, Erlangen, Germany, June, 22–24, 2017.]

- Objective :
  - Estimate automatically the temporal structure of a music track by analyzing the characteristics of its audio signal over time.
  - Temporal structure : a succession of segments.



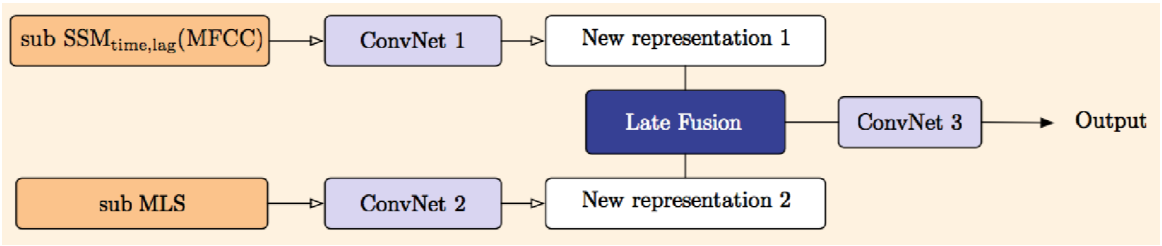
Using ConvNet for music boundary estimation

- Previous works :
  - [Grill and Schluter, 2015], [Ullrich et al., 2014] .
- Train step :
  - With a 2D representation of audio (X) and the boundaries associated $y \in \{0, 1\}$.
- Test step :
  - Output of the network :
    - For the center frame of the image excerpt :
      - probability that this frame is a boundary.
- To choose the actual boundaries :
  - peak picking algorithm proposed on this activation curve.



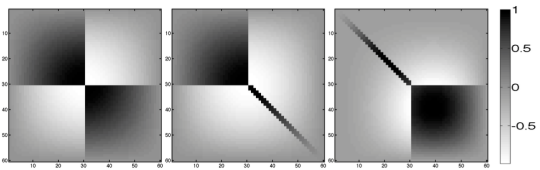
Input Representation : Previously proposed

- [Grill and Schluter, 2015] and [Ullrich et al., 2014] use as input :
  - MLS : Mel filtered Log-scale Spectrogram
  - $SSM_{time,lag}$: Self Similarity Matrix expressed in time-lag of MFCC features (SSM in lag instead of in time)
- Combining the difference representations
  - MLS + $SSM_{time,lag}$:
    - Fusion of the two representations in a convolutional layer : Late Fusion.
    - It is their best working model



Input Representation : Our proposal

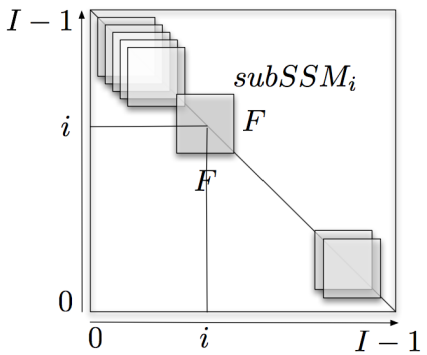
- Instead of $SSM_{time,lag}$ use $SSM_{time,time}$
  - We will use ConvNet to improve over Foote checkerboard kernels
  - Already used by [Foote, 2000] or by [Kaiser and Peeters, 2013]
  - Provides sharper edges at the beginning and ending of segments than $SSM_{time,lag}$



(a) Foote's Kernel [3] (b) Homogeneous vs. Non-Homogeneous kernel (c) Non-Homogeneous vs. Homogeneous kernel

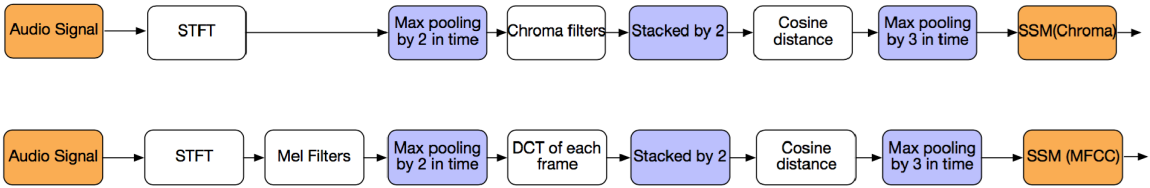
Kaiser and Peeters multi-kernels for SSM segmentation

- Use square-sub-matrices centered on the main diagonal of a Self-Similarity-Matrix time-time as input

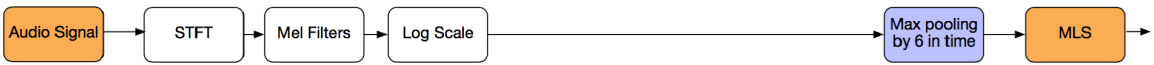


Input Representation : Our proposal

- Use several SSMs that highlight the content according to various viewpoints (timbre and harmony)

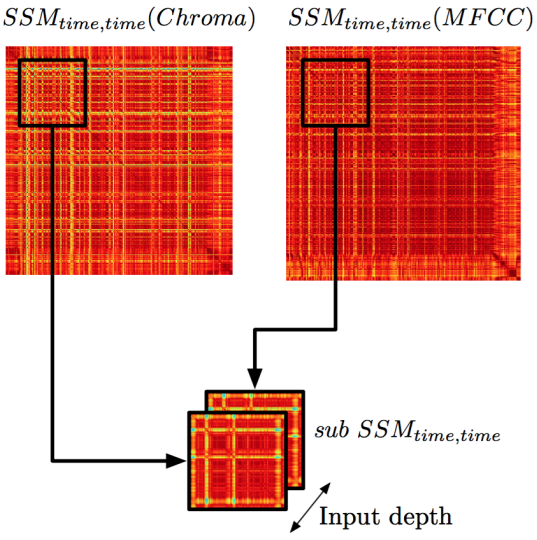


- Also use Mel-Log-Spectrum (MLS) as in [Ullrich et al., 2014] and [Grill and Schluter, 2015]



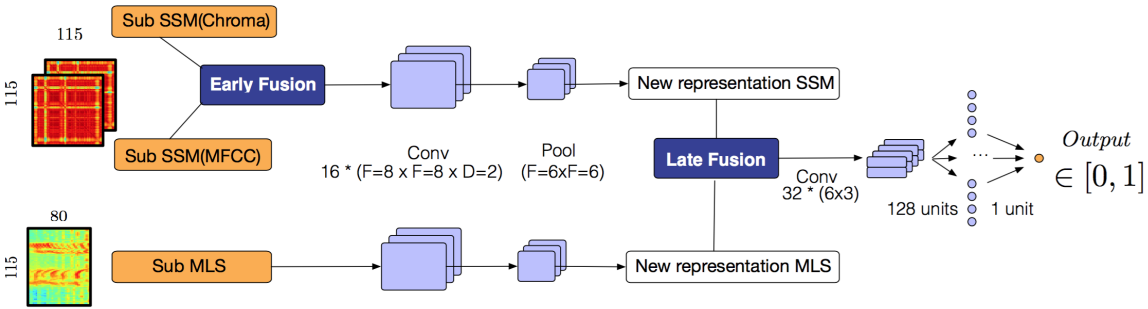
Input Representation : Our proposal

- Depth
  - Originally used to represent Red, Blue and Green (RGB) components of the input image.
- Combine the SSM using the depth of the input layer of the ConvNet
  - Provide several points of view on the audio signal, according to different musical descriptors (MFCC and Chroma).
  - Provides an early fusion of these point of view as input of the network.
  - Helpful to estimate different types of boundaries



Input Representation : Our proposal

- Late Fusion between MLS and the SSM-depth



Evaluation

- Dataset
  - SALAMI : 1048 tracks, various music genres (pop, classical, jazz or world music) annotated in structure at different temporal scales and by two different annotators
  - only have access to 732
  - [Grill and Schluter, 2015] used a second private dataset for training their system
  - Split : 400 training, 100 validation and 232 testing (artist filtering)
- Evaluation measures
  - F-measure, Precision, Recall at ± 0.5 s. and ± 3 s.
  - Area Under the ROC Curve (AUC) : True Positive (TP) rate and the False Positive (FP) rate
    - does not use any peak-picking algorithm
- Training
  - loss : binary-cross entropy
  - gradient update : AdaMax
  - mini-batch of 128 inputs
  - bagging over 5 networks
  - number of epochs : when the error on the validation set stop decreasing
  - dealing with class unbalancing
    - duplicate frames with $y = 1$ during training to deal with unbalancing
    - temporal smoothing of frames with $y = 1$

Evaluation

- Systems compared :
- ① MLS + $SSM_{time,time}$ (MFCC)
  - ② MLS + $SSM_{time,time}$ (Chroma)
  - MLS + Dept$SSM_{time,time}$
    - ③ With peak picking
    - ③' With a threshold on the output curve
  - MLS + $SSM_{time,lag}$ (MFCC) [Grill and Schluter, 2015]
    - ④ reimplemented
    - ⑤ published

- Didn't reach state-of-the-art results
  - Maybe because of the size of the dataset
- Using the self-similarity matrix expressed in time ① rather than in lag ④ provides an improvement at ± 0.5 s and ± 3 s.
- Using the depth of the input layer to combine the two $SSM_{time,time}$ ③ allows us to increase the F-measure at ± 0.5 s. and ± 3 s.
- Replacing the peak-picking algorithm ③ by a direct threshold on the network output ③' decreases the results

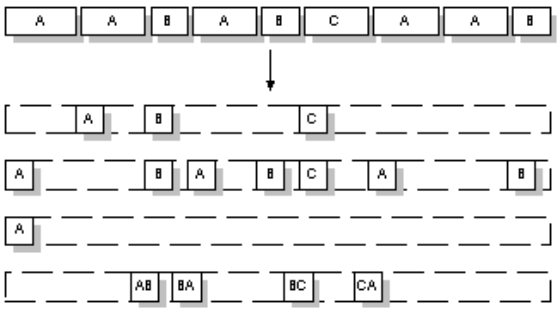
Model	± 0.5 s. tolerance				± 3 s. tolerance			
	F-m. (std)	Prec.	Rec.	AUC	F-m. (std)	Prec.	Rec.	AUC
① MLS + $subSSM^{mfcc}$	0.273 (0.132)	0.279	0.30	**0.810**	0.551 (0.158)	0.563	0.602	**0.946**
② MLS + $subSSM^{chroma}$	0.270 (0.135)	0.43	0.215	0.800	0.540 (0.153)	0.604	0.555	0.922
③ MLS + Depth($subSSM^{mfcc}$,$subSSM^{chroma}$)	**0.291** (0.120)	0.470	0.225	0.792	**0.629** (0.164)	0.755	0.624	0.930
③' MLS + Depth($subSSM^{mfcc}$,$subSSM^{chroma}$)	0.211 (0.08)	0.128	0.699	0.792	0.618 (0.156)	0.502	0.878	0.930
④[19] re-implemented: MLS+SSLM(MFCC)	0.246 (0.112)	0.291	0.239	0.774	0.580 (0.150)	0.666	0.568	0.927
⑤[19] published: MLS+SSLM(MFCC)	0.523	0.646	0.484					

7- Génération de résumé audio par estimation de structure

Génération de résumé audio par estimation de structure

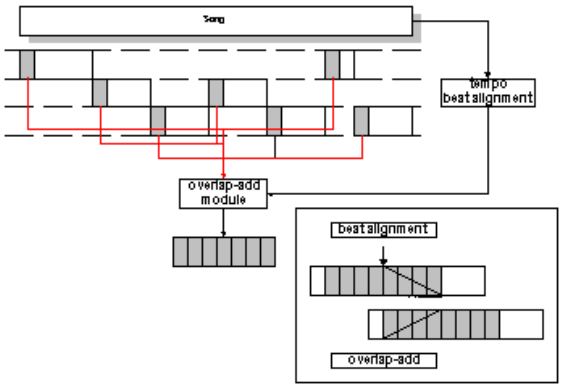
[G. Peeters, A. Laburthe, and X. Rodet. Toward automatic music audio summary generation from signal analysis. In Proc. of ISMIR, Paris, France, 2002.]Peeters, Laburthe and Rodet 2002, ISMIR]

- Stratégie proposée
  - choisir des extraits audio spécifiques en fonction du contenu dérivé de l'approche par séquence/ par état
- Construction du résumé
  - Le signal est représenté comme une succession de séquences/ états A A B A B C A A B
  - Quels séquences/ états pour le résumé ?
    - un exemple unique de chaque séquence/ état
    - reproduire la succession temporelle des séquences/ états
    - la séquence/ état le plus important (en terme de nombre de répétition, en terme d'extension temporelle)
    - exemple audio des transitions entre états



Génération de résumé audio par estimation de structure

- Construction du signal audio :
  - Extraits courts de signal audio correspondant aux séquences/ états choisis
  - Doit fournir une construction “cohérente” et “intelligente”
  - Continuité de l’information : Addition/ Recouvrement (Overlap-add), respect du tempo/ beat, taille des segments = kx4 or kx3 bars, synchronisation aux positions des beats

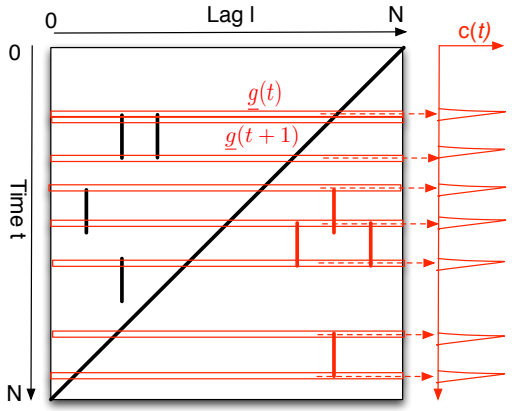


8- Estimation d'une structure musicale - approche par "séquence"

Segmentation : méthode des "Structural features"

[J. Serra, M. Muller, P. Grosche, and J. L. Arcos. Unsupervised detection of music boundaries by time series structure features. In Proc. of AAAI Conference on Artificial Intelligence, 2012.]

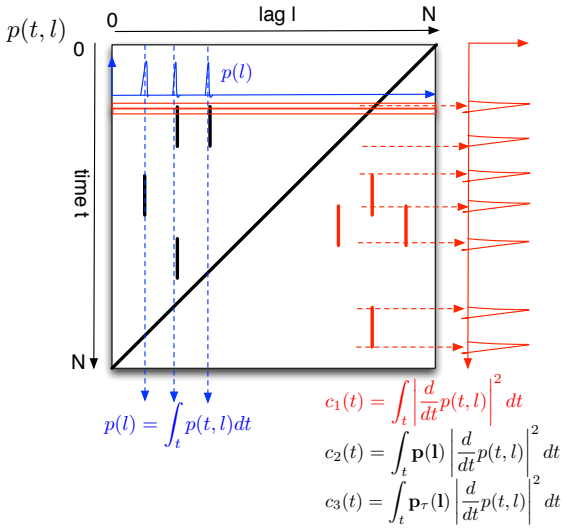
- Calcul de la matrice d'auto-similarité en (temps, lag)Lag-matrix
- On considère chaque ligne (les lags pour un temps donné) comme une "structural feature" $\underline{g}^t$
- On calcul la différence trame à trame de $\underline{g}^t$: $||\underline{g}^{t+1} - \underline{g}^t||^2$



Segmentation : méthode des "Structural features" avec probabilité a-priori

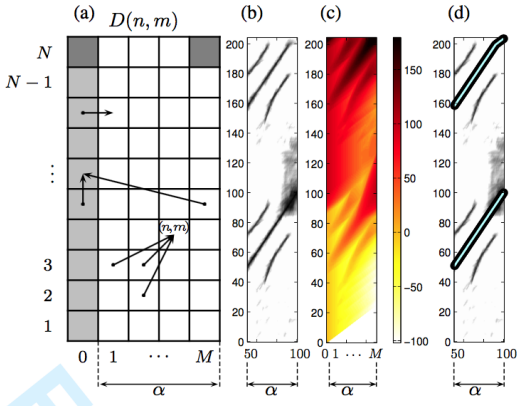
[G. Peeters and V. Bisot. Improving music structure segmentation using lag-priors. In Proc. of ISMIR, Taipei, Taiwan, 2014.]

- Pondération des “structural feature" par la probabilité a prior d’observer une répétition à un lag donné
- Calcul de cette probabilité par la méthode de Goto 2003
- On calcul la différence trame à trame



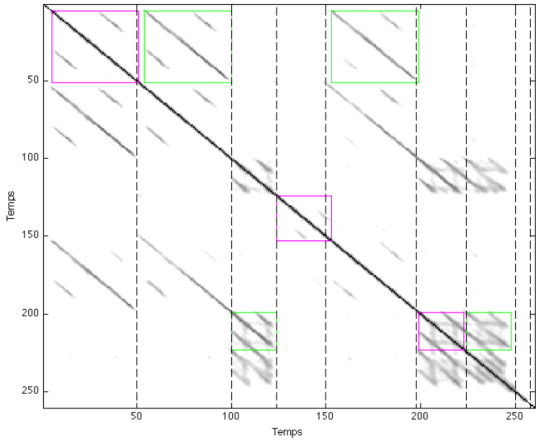
Regroupement par Dynamic Time Warping

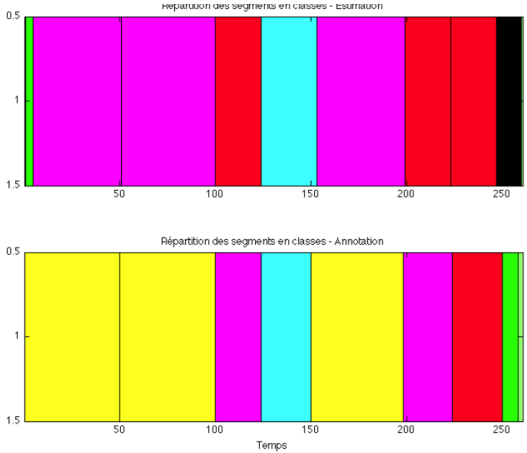
- Pour chaque segment détecté, on cherche quelles séquences temporelles sont expliquées
- Utilisation d'une version modifiée du Dynamic Time Warping



source : Mueller

Regroupement par Dynamic Time Warping





source : Bisot

Questions ?