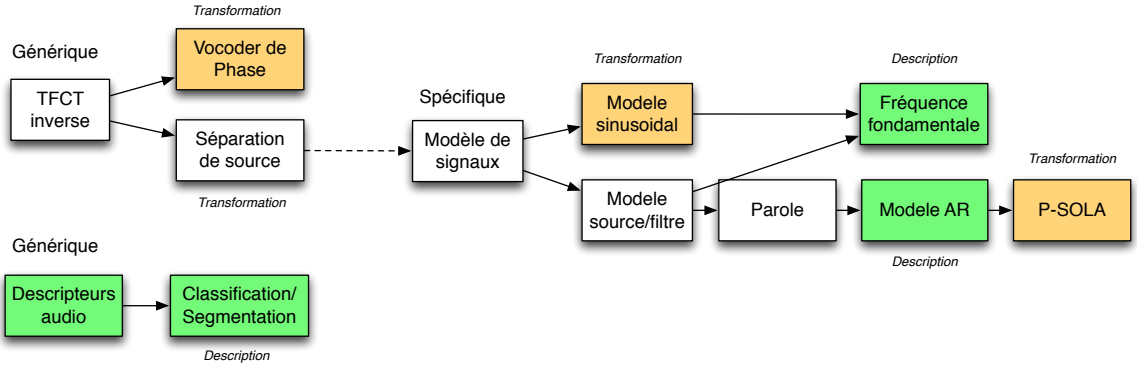
ENSEA 3ème SyM (2016-2017)
Traitement du signal audio musical
Partie 2: modèles de signaux, estimateurs, descripteurs audio et apprentissage machine

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

1- Introduction



Les modèles de signaux

Pourquoi des modèles de signaux ?

- On suppose que le signal a été produit par un certain **modèle**
- Permet de **réduire** le nombre de paramètres observés du signal
  - la TFCT contient beaucoup trop d'information
- Permet d'obtenir des paramètres plus facilement **interprétables** (indexation) et manipulables (transformation, synthèse)
  - la TFCT fournit des paramètres non directement exploitables
- Quels modèles ?
  - Modèle sinusoidal harmonique
  - Modèle source/ filtre
  - Modèle autorégressif

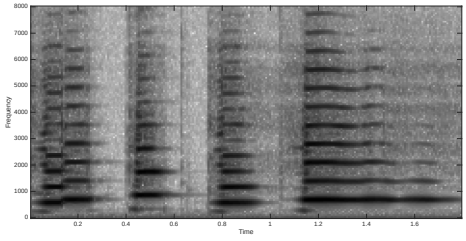
Le modèle sinusoidal

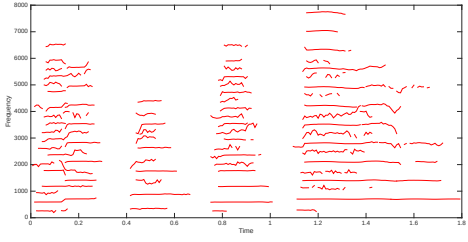
Le modèle sinusoidal

- Le signal $x(t)$ est représenté par une somme de sinusoides plus un bruit

$$x(t) = \sum_{h=1}^{H(t)} A_h(t)\cos(\phi_h(t)) + b(t)$$

  - les sinusoides sont en nombre limité
  - les paramètres des sinusoides sont lentement variables en temps





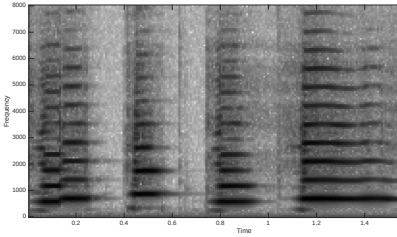
Le modèle sinusoidal

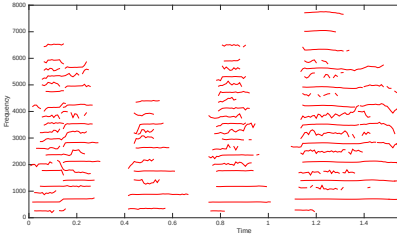
- Lentement variables en temps ?
  - $\rightarrow$ stationnarité locale
  - on peut approximer localement en temps $A_h(t)$ par sa valeur à une trame donnée $A_h(t_m)$
  - on peut approximer localement en temps l'évolution de la phase par $\phi(t) = \omega_h(t_m)(t - t_m) + \phi_h(t_m)$

$$x(t) = \sum_{h=1}^{H(t)} A_h(t) \cos(\phi_h(t)) + b(t)$$

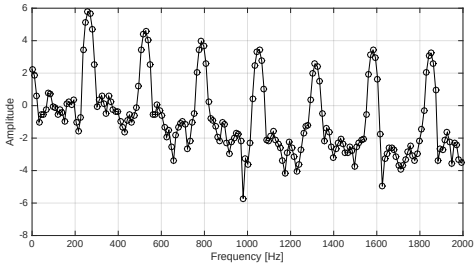
$$\hat{x}_m(t_m) = \sum_{h=1}^{H(t)} A_h(t_m) \cos(\omega_h(t_m)(t - t_m) + \phi_h(t_m)))$$

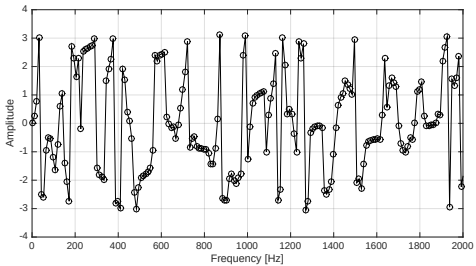
$$\hat{x}(t) = \sum_m \hat{x}_m(t) + b(t)$$





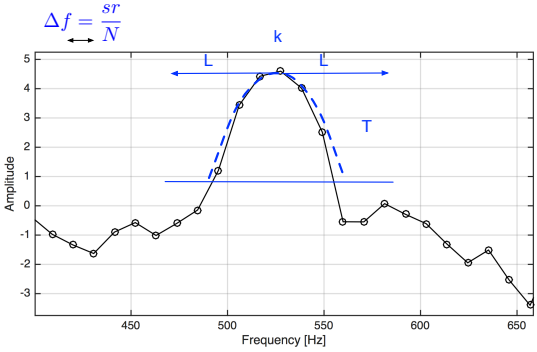
Comment trouver les $\omega_h(t_m)$ et $A_h(t_m)$?





Comment trouver les $\omega_h(t_m)$ et $A_h(t_m)$?

- 1. Détection des maxima locaux/ des piques de $|X(k,m)|$ (peak-picking)
  - k est un maximum local si
    - si il est supérieur à la valeur de ses voisins sur une longueur L à gauche et à droite
    - si il est supérieur à T fois la valeur de ces voisins
    - ...
  - Amélioration de la **précision** fréquentielle :
    - utilisation du zero-padding

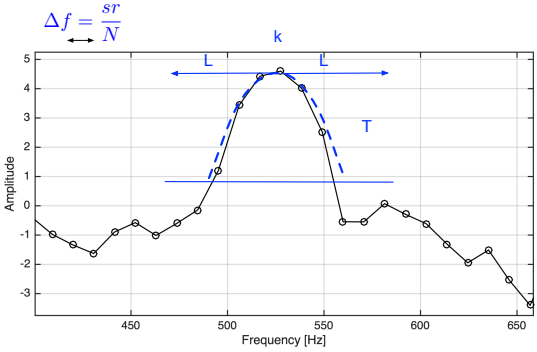


Comment trouver les $\omega_h(t_m)$ et $A_h(t_m)$?

- 2. **Interpolation** du spectre de puissance sur trois points :

$$\omega_h = \omega_k + \frac{\Delta}{2}\frac{P_{k-1} - P_{k+1}}{P_{k-1} - 2P_k + P_{k+1}}$$
$$P_h = P_k - \frac{1}{8}\frac{(P_{k-1} - P_{k+1})^2}{P_{k-1} - 2P_k + P_{k+1}} \quad (1)$$

  - avec $P_k = |X(k,m)|^2$
  - avec $\Delta = \omega_k - \omega_{k-1}$



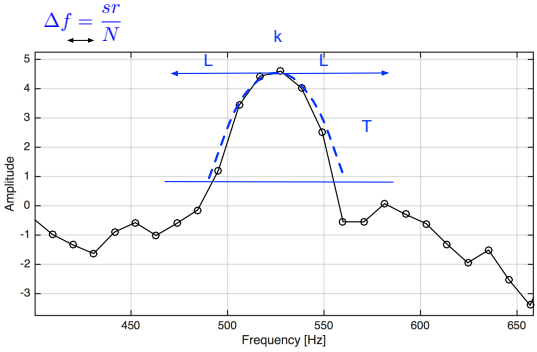
Comment trouver les $\omega_h(t_m)$ et $A_h(t_m)$?

- 3. **Régression** parabolique
  - on peut montrer que la forme du spectre (dans le cas d'une fenêtre d'analyse gaussienne de variance σ^2) est
  - $\log(|X(\omega)|) = \log(A_h) - \frac{\sigma^2}{2}(\omega - \omega_h)$

$$\omega_h = \omega_k + \frac{s}{\Delta}(\log(A_{k+1}) - \log(A_{k-1}))$$

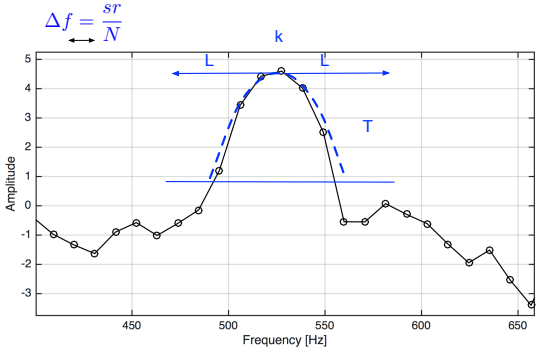
$$\log(A_h) = \frac{\Delta^2}{6s} + \frac{\omega_h^2}{4s} + \frac{\sum \log(A_i)}{3} \quad (2)$$

  - avec $\log(A_k) = \log(|X(k,m)|)$
  - avec $s = \frac{1}{2(2\pi\sigma)^2}$



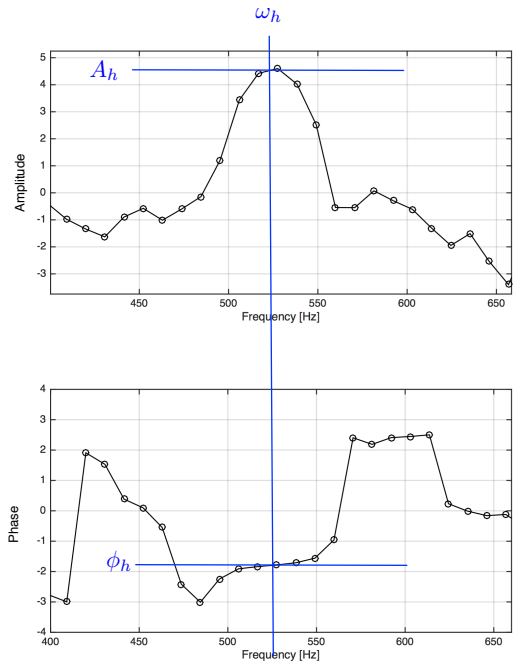
Comment trouver les $\omega_h(t_m)$ et $A_h(t_m)$?

- 4. Fréquence instantannée pour trouver plus précisément ω_h



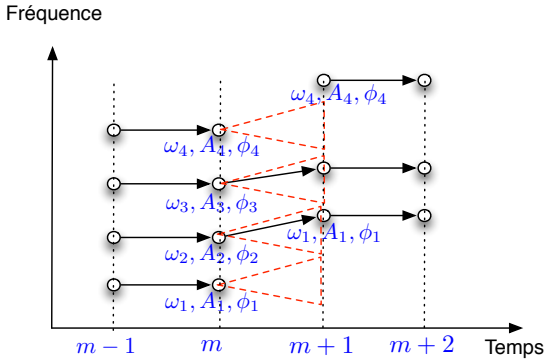
Comment trouver les $\phi_h(m)$?

- Prendre la phase correspondant à ω_h
- Le spectre de phase correspondant à une sinusoide est un plateau
- Attention :
  - la phase retournée par l'algorithme FFT est donnée par rapport au début de la fenêtre d'analyse
  - mais l'énergie de la fenêtre d'analyse est au milieu
  - il faut corriger le spectre de phase :
  - $x(t+t_0) \rightleftarrows X(f)\exp(j2\pi f t_0)$
    - avec $t_0 = L/2$



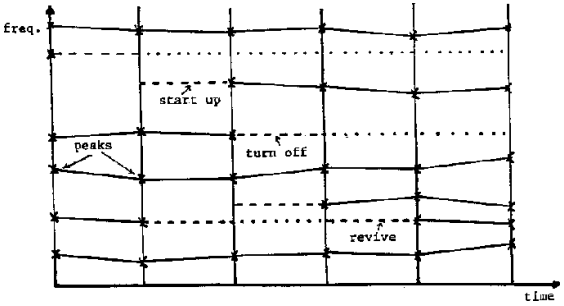
Création de trajectoire temporelle de sinusoide (partial tracking)

- A chaque trame d'analyse, on a estimé
  - un ensemble de H triplets de paramètres $\{\omega_h, A_h, \phi_h\}$
- Nous cherchons a créer des trajets continus (à travers le temps) de sinusoides
  - Nous devons connecter les $\{\omega_h, A_h, \phi_h\}$ à un instant m donné aux $\{\omega_h, A_h, \phi_h\}$ à l'instant suivant $m+1$
- Méthode du cône fréquentielle
  - on cherche à connecter le peak $\omega_h(m)$ à un peak $\omega_?(m+1)$
  - $w_?(m+1) \in [w_h - \Delta, \omega_h + \Delta]$
    - Δ représente la variation de fréquence acceptée entre deux trames
  - $A_?(m+1) \in [A_h - \Delta, A_h + \Delta]$
    - Δ représente la variation d'amplitude acceptée entre deux trames



Création de trajectoire temporelle de sinusoide (partial tracking)

- Il faut également gérer au cours du temps
  - les naissances de sinusoides
  - les morts de sinusoides

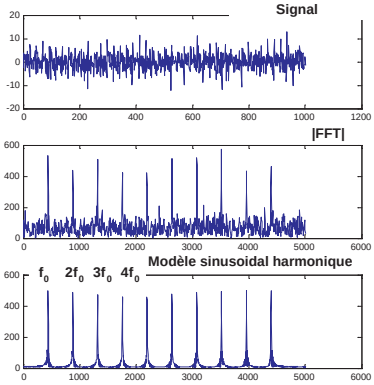


Le modèle sinusoidal harmonique

Le modèle sinusoidal harmonique

- Même chose que le modèle sinusoidal mais
  - $x(t)$ = source mono-phonique et harmonique
  - note de musique, parties voisées parole
- Conséquences :
  - les fréquences $f_h(t_m) = hf_0(t_m)$
    - Si on connait $f_0(t_m)$, on en déduit la position des sinusoides : $f_h(t_m) = hf_0(t_m)$,
    - Il ne reste plus qu'à affiner $f_h(t_m)$ et déterminer le $A_h(t_m)$ et le $\phi_h(t_m)$ correspondant
  - Nombre de sinusoides H = nombre d'harmoniques
    - $H(t)$ est constant au cours du temps
    - Plus besoin de créer les trajectoires

      - le ω_1 à m se connecte par définition au w_1 à $m+1$



Le modèle sinusoidal harmonique

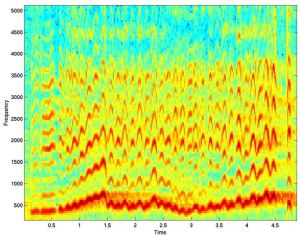
$$\hat{x}_m(t_m) = \sum_{h=1}^{H(t)} A_h(t_m)\cos(\omega_h(t_m)(t-t_m)+\phi_h(t_m)))$$

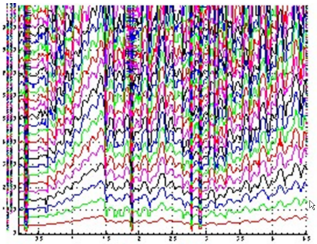
devient

$$\hat{x}_m(t) = \sum_{h=1}^{H} A_h(t_m)\cos(h\omega_0(t_m)(t-t_m)+\phi_h(t_m))) \quad (3)$$

$$\hat{x}(t) = \sum_m \hat{x}_m(t) + b(t)$$

- Méthode :
  - on commence par détecter la fréquence fondamentale au cours du temps $f_0(t_m)$
  - autour de chaque $h\omega_0(t_m)$ on cherche les valeurs précises de $\{\omega_h, A_h, \phi_h\}$
  - pas de création de trajets





Utilisation du modèle sinusoidal, sinusoidal harmonique ?

- Transformations de haute qualité
  - traitement séparé de la partie sinusoidale, de la partie bruitée (contrairement au vocodeur de phase)
- Traitements plus poussés
  - changer l'harmonicité du signal, modifier certaines harmoniques
- Compression
  - transmission uniquement de $f_0(t)$ et de l'enveloppe spectrale
    - valeurs de A_h sous forme compressé
- Indexation audio
  - extraire des descripteurs audio plus précis du signal

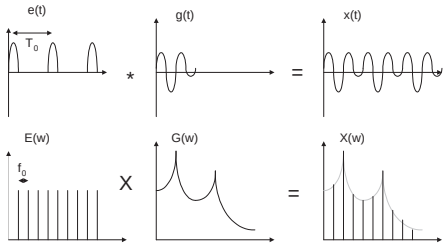
Différentes méthodes d'estimation

- de la fréquence fondamentale
  - auto-corrélation, AMDF, Yin, Cepstre, maximum de vraisemblance, ...
- de l'enveloppe spectrale
  - LPC, Cepstre, MFCC, ...

Le modèle source/ filtre

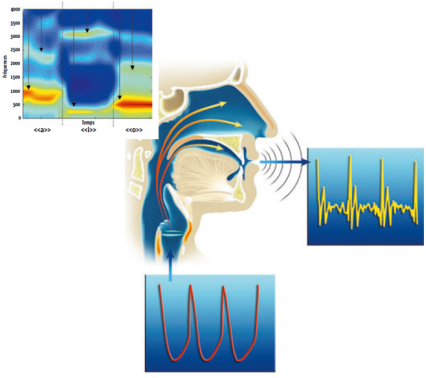
Le modèle source/ filtre

- Hypothèse :
  - le signal $x(t)$ est le résultat du passage d'une excitation (un pulse, une série de pulse) dans un filtre (résonnant)
    - Exemples : le signal de parole, certains instruments de musique (trompette)
- Modélisation temporelle :
  - un signal d'excitation $e(t)$ passe (convolution) à travers un filtre $g(t)$:
  - $x(t) = e(t) \circledast g(t)$
- Modélisation fréquentielle
  - la multiplication de la TF du signal d'excitation (source) par la TF du filtre.
  - $X(\omega) = E(\omega) \cdot G(\omega)$

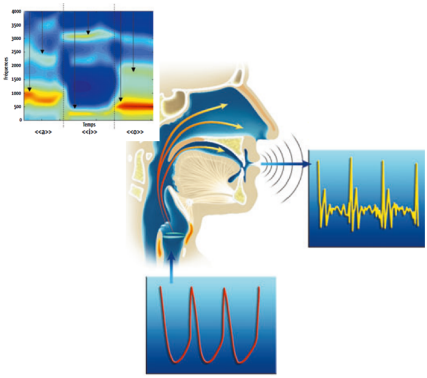


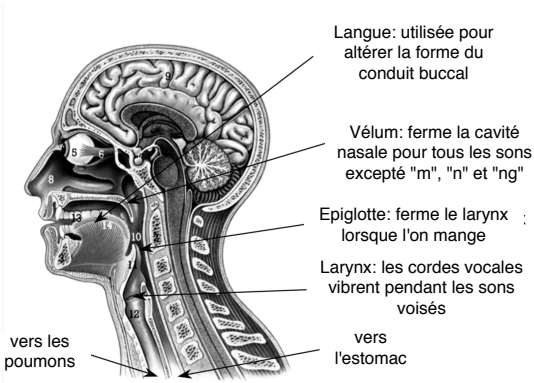
Production du signal vocal

- Le signal de parole (pour sa partie voisée) est créé par
  - les cordes vocales
    - une excitation régulière, périodique
  - le conduit bucco-nasal (bouches et nez)
    - filtrage résonant et anti-résonant

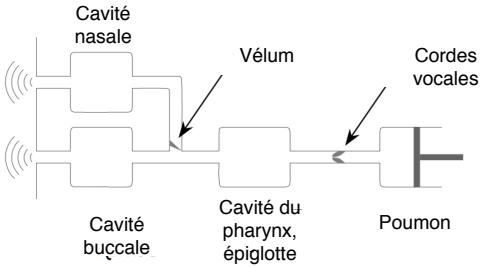


- Ouverture/fermeture périodique des cordes vocales
  - détermine la hauteur
  - hauteur de 100Hz ?
    - pulses d'air espacés de $T_0 = \frac{1}{f_0} = \frac{1}{100} = 10ms.$
  - appelé **signal d'excitation** (ou **source**), $e(t)$.
- Conduit bucco-nasal
  - créer les différentes voyelles pour une hauteur donnée
  - renforce (résonance) et retire (anti-résonances) certains fréquences.
    - filtre résonant (AR : Auto-Regressif)
    - filtre anti-résonant (MA : Moving Average)
    - Total = filtre dit "ARMA".

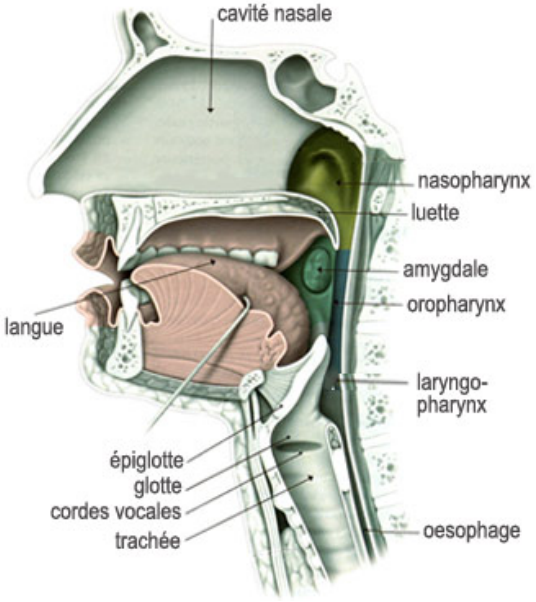




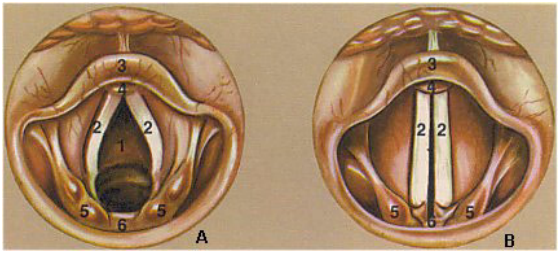
source : Mike Brookes



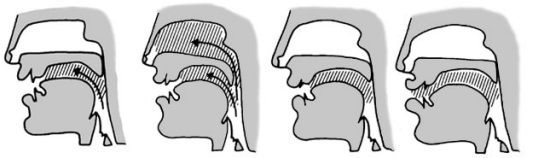
source : Mike Brookes



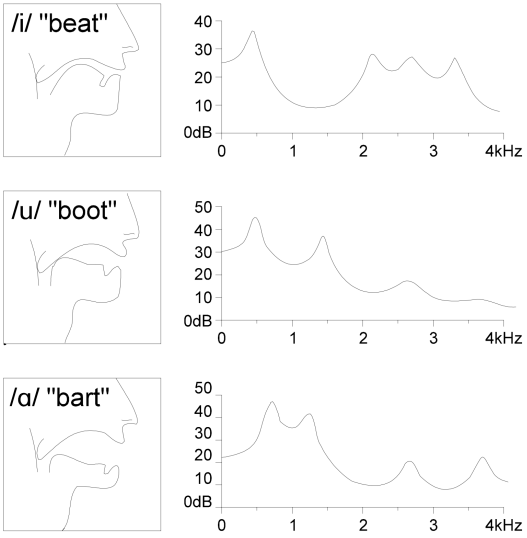
source : outilsrecherche.over-blog.com



source : outilsrecherche.over-blog.com

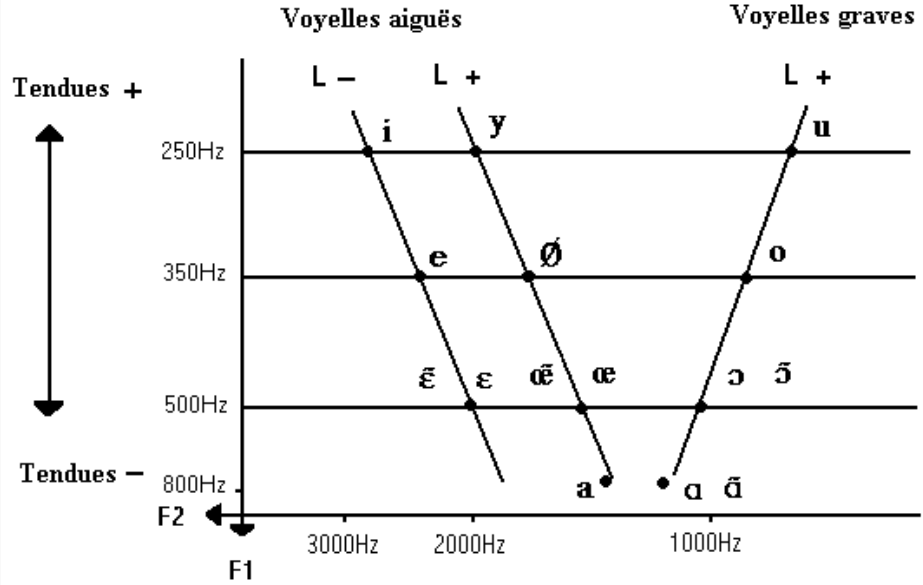


source : outilsrecherche.over-blog.com

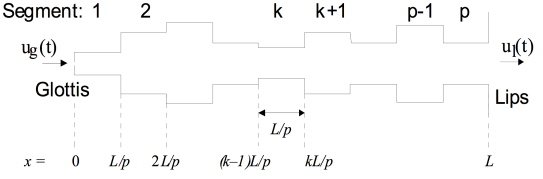


source : Mike Brookes

Fréquences des formants pour les différentes voyalles

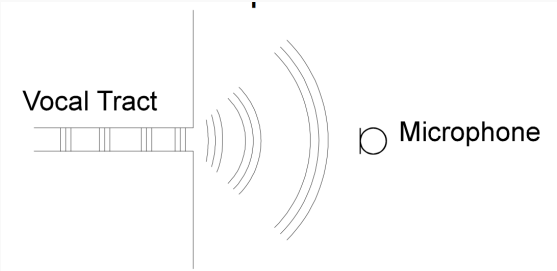


Représentation sous-forme de tube



source : Mike Brookes

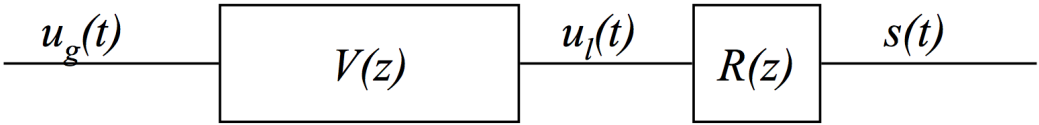
Radiation des lèvres



source : Mike Brookes

- Filtre passe-haut $R(z) = 1 - z^{-1}$

Système équivalent

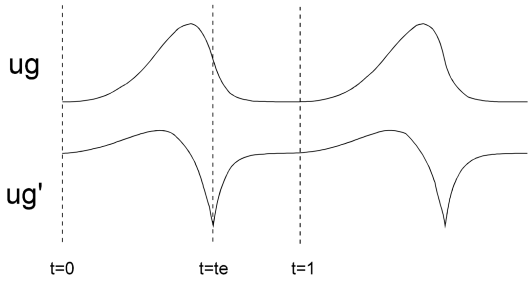


Modèle de forme-d'onde pour la glotte

- Modèle "LF" (Liljencrants et Fant)

$$u'_g(t) = e^{at}\sin(bt) \text{ pour } 0 \leqslant t < t_e$$
$$= c + de^{-ft}e^{at}\sin(bt) \text{ pour } t_e \leqslant t < 1$$

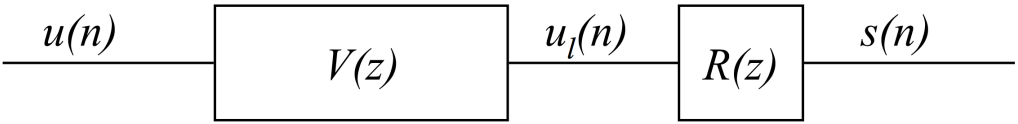
  - avec $u_g(0) = u_g(1) = 0$ et $u'_g(t)$ continu en t_e



source : Mike Brookes

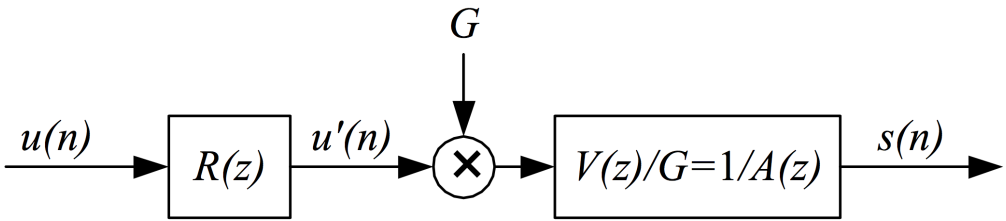
La prédiction linéraire, modèle auto-régressif (AR)

- Système équivalent de production vocale

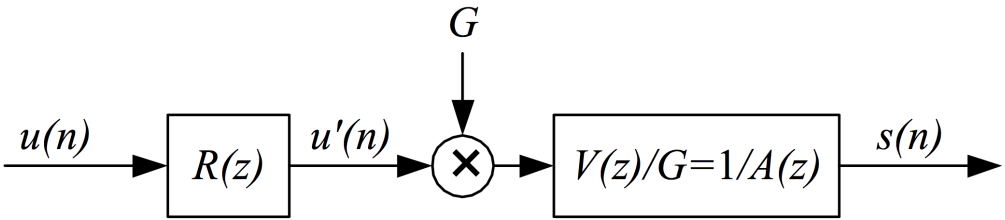


source : Mike Brookes

- Inversion de l'ordre de $V(z)$ et $R(z)$
  - puisque linéaire et
  - puisque $V(z)$ ne change pas significativement durant la réponse impulsionnelle de $R(z)$ et inversement



source : Mike Brookes



source : Mike Brookes

- $x(n) = Gu'(n) + \sum_{j=1}^{P} a_j x(n-j)$
  - Si le gain des résonances du conduit vocal est important, le second terme va dominer
- $x(n) \simeq \sum_{j=1}^{P} a_j x(n-j)$
  - La partie de droite est la prédiction de $x(n)$ comme combinaison linéaire des échantillons passés de la voix

Modèle auto-régressif : $x(n) \simeq \sum_{j=1}^{P} a_j x(n-j)$

- On définit l'erreur de prédiction à l'échantillon n comme

$$\begin{aligned} e(n) &= x(n) - \sum_{j=1}^{P} a_j x(n-j) \\ &= x(n) - a_1 x(n-1) - a_2 x(n-2) \ldots - a_P x(n-P) \end{aligned}$$

- En transformée en Z
  - $E(z) = X(z)A(z)$
- Etant donné une trame de signal de parole $\{F\}$, on cherche les valeurs a_i qui minimize
  - $Q_E = \sum_{n \in \{F\}} e^2(n)$
- Minimisation par rapport aux $a_i \rightarrow$ différenciation de Q_E par rapport aux a_i

$$\frac{\partial Q_E}{\partial a_i} = \sum_{n \in \{F\}} \frac{\partial (e^2(n))}{\partial a_i} = \sum_{n \in \{F\}} 2e(n) \frac{\partial e(n)}{\partial a_i} = - \sum_{n \in \{F\}} 2e(n) x(n-i)$$

- Les valeurs optimales de a_i doivent satisfaire les P équations :

$$\text{pour } i = 1 \dots P \quad \sum_{n \in \{F\}} e(n)x(n-i) = 0$$

$$\text{puisque } e(n) = x(n) - \sum_{j=1}^{P} a_j x(n-j)$$

$$\text{on a } \sum_{n \in \{F\}} \left(x(n)x(n-i) - \sum_{j=1}^{P} a_j x(n-j)x(n-i) \right) = 0$$

$$\sum_{j=1}^{P} a_j \sum_{n \in \{F\}} x(n-j)x(n-i) = \sum_{n \in \{F\}} x(n)x(n-i)$$

- Système de i équations à résoudre

$$\sum_{j=1}^{P} \phi_{ij} a_j = \phi_{i0}$$

  - avec $\phi_{ij} = \sum_{n \in \{F\}} x(n-i)x(n-j)$

- Système de i équations à résoudre

$$\sum_{j=1}^{P} \phi_{ij} a_j = \phi_{i0} \tag{4}$$

  - avec $\phi_{ij} = \sum_{n \in \{F\}} s(n-i)s(n-j)$

- Sous forme matricielle

$$\begin{aligned} \underline{\underline{\phi}}\,\underline{a} &= \underline{c} \\ \underline{a} &= \underline{\underline{\phi}}^{-1}\underline{c} \end{aligned} \tag{5}$$

  - avec $\underline{\underline{\phi}}$ matrice **symétrique** et définie **semi-positive**

- Rappel
  - Matrice **symétrique :**
    - $\phi_{ji} = \phi_{ij} \Leftrightarrow \underline{\underline{\phi}}^T = \underline{\underline{\phi}}$
  - Matrice **définie semi-positive** :
    - $\sum_{i,j} x_i \phi_{ij} x_j \geqslant 0 \Leftrightarrow \underline{x}^T \underline{\underline{\phi}}\,\underline{x} \geqslant 0$ pour tout x
  - Matrice de **Toeplitz** :
    - les diagonales sont constantes :
    - $\phi_{i+1,j+1} = \phi_{ij} = f(i-j)$

Solution 1 : prédiction linéaire par auto-corrélation

- on choisit $\{F\}$ comme l'interval infini
  - $\phi_{ij} = \sum_{n=-\infty}^{+\infty} x(n-i)x(n-j)$
- à cause de la symétrie et de l'interval infini on a
  - $\phi_{ij} = \phi_{|i-j|,0} = R_{|i-j|}$
  - avec R_k la séquence d'auto-corrélation du signal de parole
- Dans ce cas ϕ_{ij} est une matrice de Toeplitz (les diagonales sont constantes),
  - inversion en $O(p^2)$ au lieu de $O(p^3)$
- Les équations $\underline{\underline{\phi}}\underline{a} = \underline{c}$ sont appelées **équations de Yule-Walker**.
  - l'algorithme d'inversion correspondant est appelé **algorithme de Durbin-Levinson**

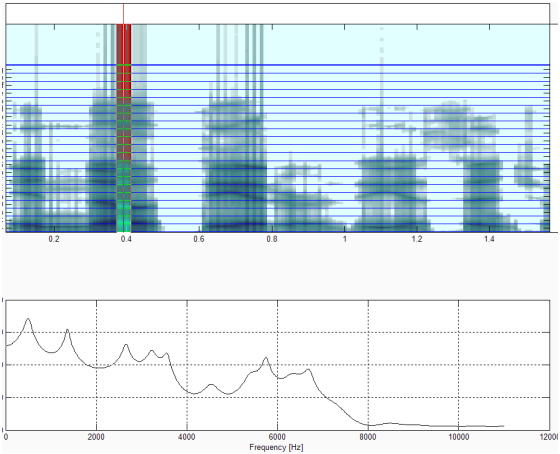
- Rappel
  - Matrice **symétrique :**
    - $\phi_{ji} = \phi_{ij} \Leftrightarrow \underline{\underline{\phi}}^T = \underline{\underline{\phi}}$
  - Matrice **définie semi-positive** :
    - $\sum_{i,j} x_i \phi_{ij} x_j \geqslant 0 \Leftrightarrow \underline{x}^T \underline{\underline{\phi}} \underline{x} \geqslant 0$ pour tout x
  - Matrice de **Toeplitz** :
    - les diagonales sont constantes :
    - $\phi_{i+1,j+1} = \phi_{ij} = f(i-j)$

Solution 2 : prédiction linéaire par covariance

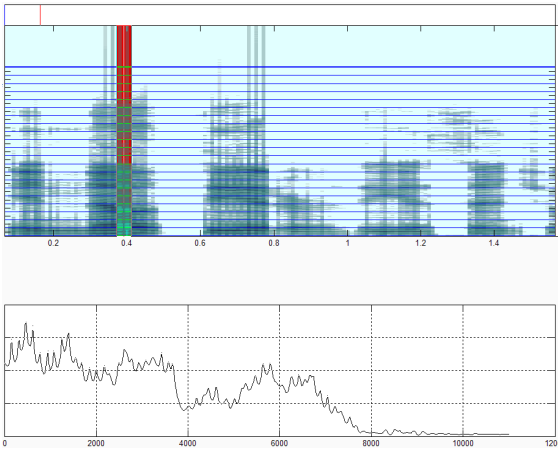
- on choisit $\{F\}$ comme un interval fini de parole $\{F\} = s(n) \quad 0 \leqslant n \leqslant (N-1)$
  - $\phi_{ij} = \sum_{n=0}^{N-1} x(n-i)x(n-j)$
- la matrice ϕ_{ij} est symétrique mais plus de Toeplitz
  - caclul plus lourd $O(p^3)$

- Rappel
  - Matrice **symétrique :**
    - $\phi_{ji} = \phi_{ij} \Leftrightarrow \underline{\underline{\phi}}^T = \underline{\underline{\phi}}$
  - Matrice **définie semi-positive** :
    - $\sum_{i,j} x_i \phi_{ij} x_j \geqslant 0 \Leftrightarrow \underline{x}^T \underline{\underline{\phi}} \underline{x} \geqslant 0$ pour tout x
  - Matrice de **Toeplitz** :
    - les diagonales sont constantes :
    - $\phi_{i+1,j+1} = \phi_{ij} = f(i-j)$

Choix du nombre de pôle P



$P = 40$



$P = 200$

Représentation des résonances

- Paramètres a_k du filtre auto-régressif

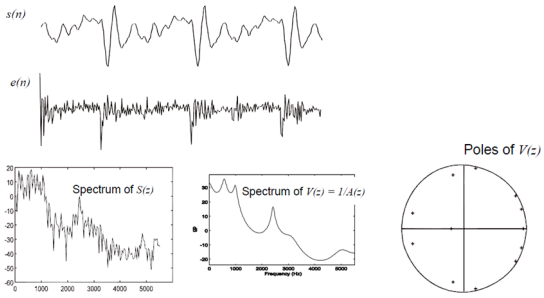
$$V(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \quad (6)$$

  - Réponse en fréquence très sensible aux petits changements des a_k
  - Interpolation difficile
- Factorisation des pôles de $V(z)$

$$1 - \sum_{k=1}^{p} a_k z^{-k} = \prod_{k=1}^{p} (1 - x_k z^{-1}) \quad (7)$$

- Coefficients de réflection du tube
- Log Area Ratio du tube équivalent

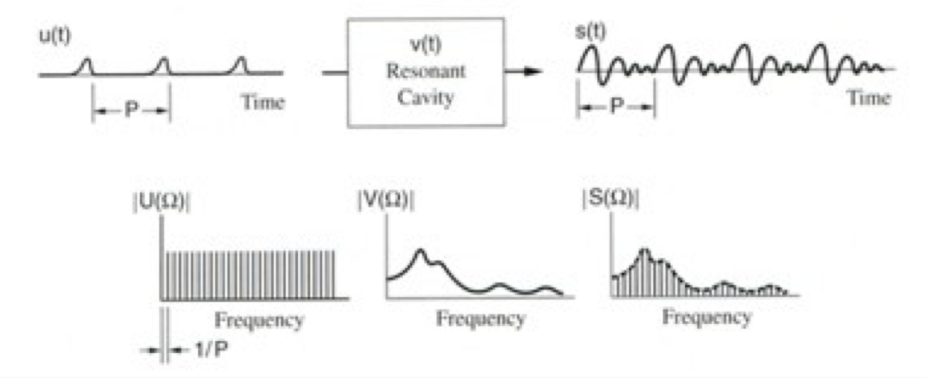
$$g_i = \log(\frac{A_{i+1}}{A_i}) = \log(\frac{1 + r_i}{1 - r_i}) \quad (8)$$



Transformation du signal par la méthode P-SOLA

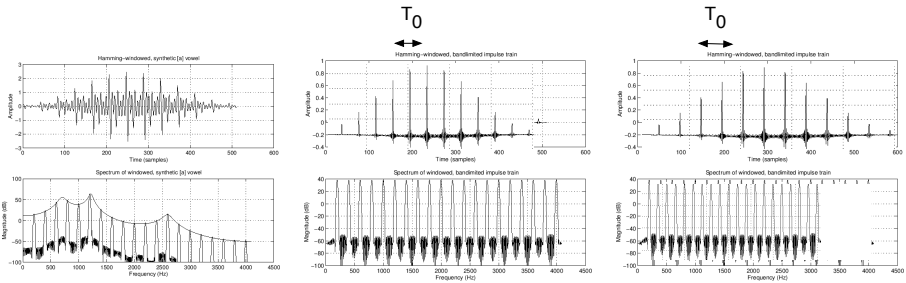
Transformation du signal par la méthode P-SOLA

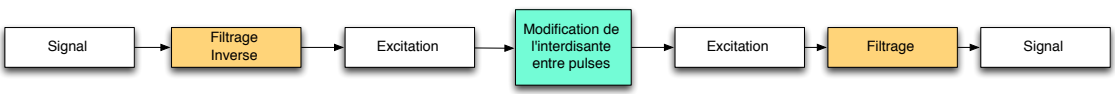
- Modèle source/filtre
  - un signal d'excitation $e(t)$ passe (convolution) à travers un filtre $g(t)$:
    - $x(t) = e(t) \circledast g(t)$
    - $e(t) = \sum_m \delta(t - mT_0)$
  - Modélisation fréquentielle
    - la multiplication de la TF du signal d'excitation (source) par la TF du filtre.
    - $X(\omega) = E(\omega) \cdot G(\omega)$
    - $X(\omega) = \frac{1}{T_0} G(h\omega_0)$



Transformation du signal par la méthode P-SOLA

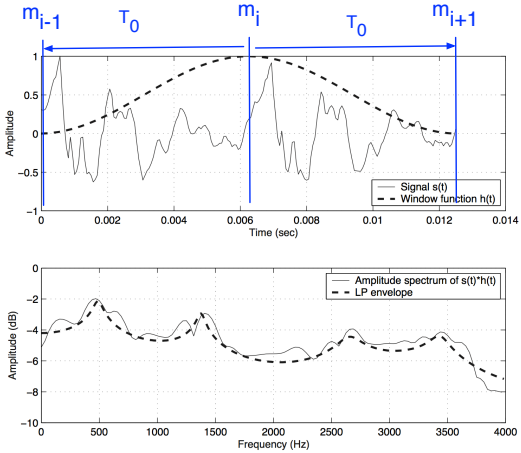
- LP-P-SOLA (Linear Predictive P-SOLA)
  - déconvolution du signal de parole $x(t)$ par le filtre de prédiction linéaire $g(t)$ estimé
  - modification de la position des pulses glottiques dans $e(t)$
  - convolution du signal résultant par le filtre de prédiction linéaire $g(t)$



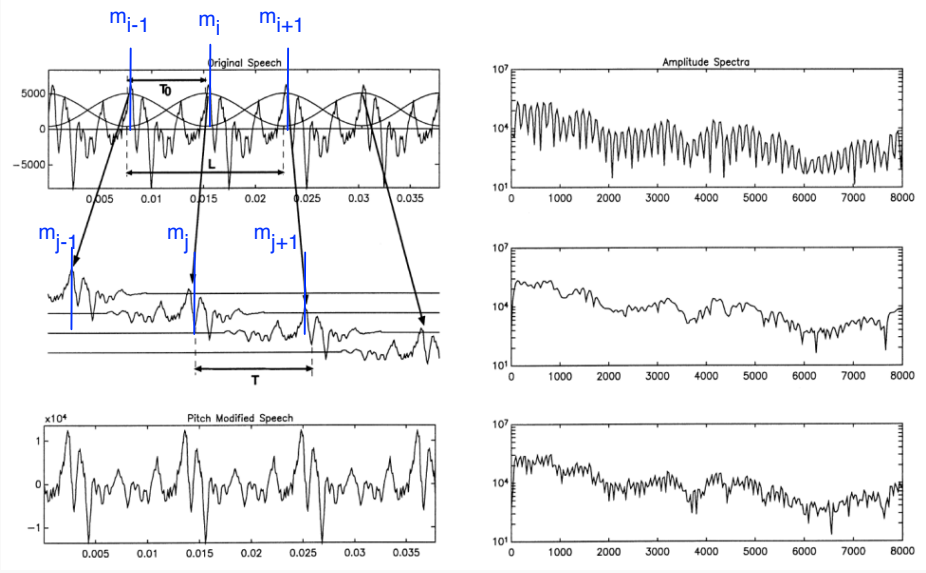


Transformation du signal par la méthode P-SOLA

- **Pitch-Synchronous** Overlap Add
- P-SOLA **Analyse**
  - **forme d'onde élémentaire**
  - fenêtrage du signal autour des instants de fermeture de glotte sur une longueur $[-T_0, T_0]$
  - $x_i(t) = x(t - m_i)w_i(t)$
    - m_i : instants d'analyse, position des instants de fermeture de glotte
    - tel que $m_{i+1} - m_i = T_0$
    - tel que $w_i(t) = w\left(\frac{t}{2T_0(m_i)}\right)$ fenêtre de taille égale à $2T_0$
  - on fait l'hypothèse que $x_i(t)$ approxime bien la R.I. du filtre de prédiction linéaire



P-SOLA Synthèse (abaissement de la hauteur : $T > T_0$)



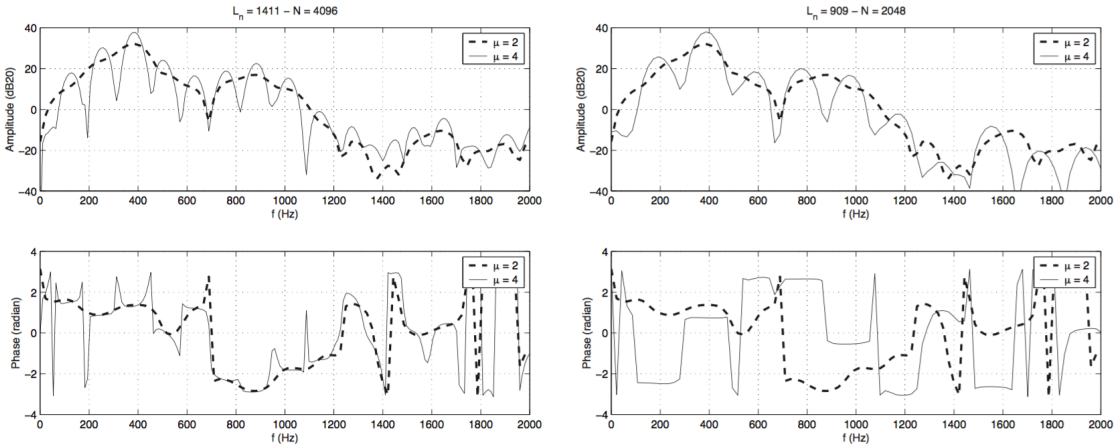
Transformation du signal par la méthode P-SOLA

- P-SOLA **Synthèse**
  - pour modifier la hauteur :
    - on modifie directement l'interdistance entre formes d'ondes élémentaires
    - $m_{j+1} - m_j = T$
  - pour allonger la durée :
    - on recopie plusieurs fois formes d'ondes élémentaires
  - pour raccourci la durée :
    - on retire certaines formes d'ondes élémentaires
- re-synthèse par addition/recouvrement dans le domaine temporelle
  - $\hat{x}(t) = \sum_j x_j(t + m_j)$
    - m_j instants de synthèse

- P-SOLA Pro :
  - synthèse extrêmement rapide, de très grandes qualité pour la parole
- P-SOLA Con :
  - limiter au traitement de signaux répondant au modèle source/filtre
  - approximation diffiçile pour les pitchs élevés
  - nécessite l'estimation de $f_0(t)$
  - nécessite une estimation des instants de fermeture de glotte t_a^i

Transformation du signal par la méthode P-SOLA

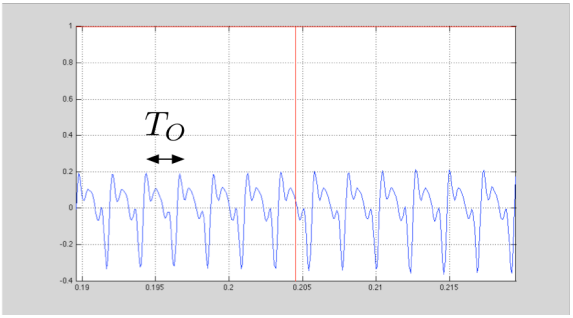
- Préservation de l'enveloppe spectrale

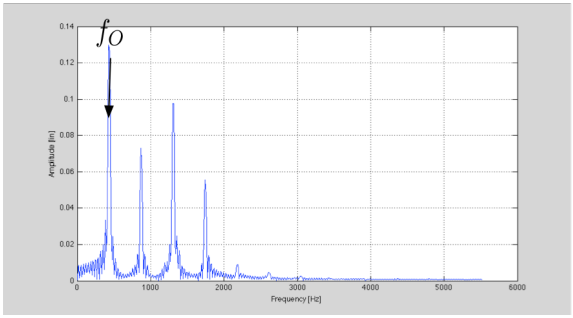


3- Utilisation de modèles de signal

Période fondamentale T_0 ou fréquence fondamentale f_0

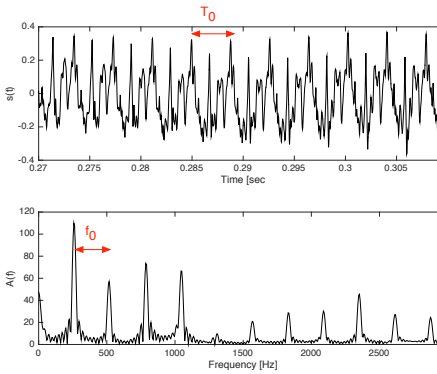
- f_0 : fréquence fondamentale en Hz
  - exemple La3/A4= 440Hz
- $T_0 = \frac{1}{f0}$: période fondamentale en secondes
  - exemple La3/A4 = 0.0023s.





$$x(n) = \sum_{h=1}^{H} 2A_h \cos(2\pi h f_0 n + \phi_h) + w(n)$$

- $f_0 = \frac{1}{T_0}$: fréquence/ période fondamentale
- H est le nombre total d'harmoniques
- A_h sont les amplitudes des harmoniques, $A_h \geqslant 0$
- ϕ_h sont les phases des harmoniques, $\phi_h \in [-\pi, \pi]$
- $w(n)$ est un bruit blanc centré de variance σ^2



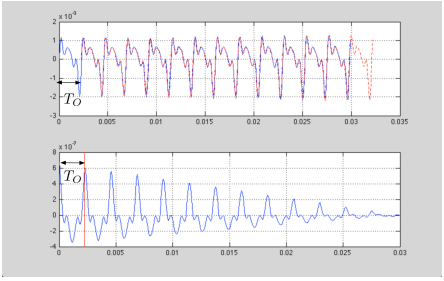
Auto-covariance

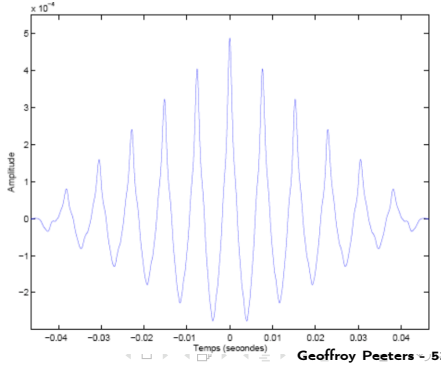
- $x(n)$ est un processeur SSL* centré d'auto-covariance
  - $(\star)$ SSL : stationnaire au sens large
    - $\mu_x(t) = \mu_x$ et $P(t, \tau) = P(t - \tau)$
- Auto-covariance :
  $r_x(m) = \sum_{h=1}^{H} [2A_h^2 \cos(2\pi h f_0 m)] + \sigma^2 \delta(m)$

Auto-corrélation biaisée

$$\hat{r}_x(m) = \frac{1}{N} \sum_{n=0}^{N-1-m} x(n)x(n+m) \text{ si } m \geqslant 0$$

- $E[\hat{r}_x(m)] = \frac{N-|m|}{N} r_x(m)$
- $|\hat{r}_x(m)| \leqslant \hat{r}_x(0)$

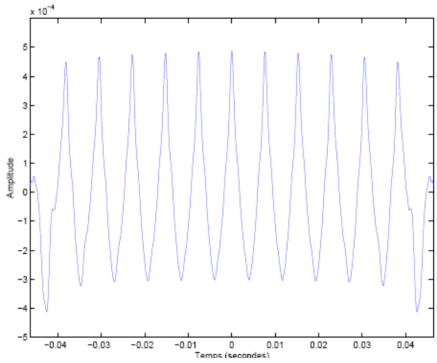




Auto-corrélation non-biaisée

$$\tilde{r}_x(m) = \frac{1}{N-m} \sum_{n=0}^{N-1-m} x(n)x(n+m) \text{ si } m \geqslant 0$$

- $E[\tilde{r}_x(m)] = r_x(m)$
- $Var[\tilde{r}_x(m)] = (\frac{N}{N-m})^2 Var[\hat{r}_x(m)]$
- $|\tilde{r}_x(m)| \leqslant \tilde{r}_x(0)$

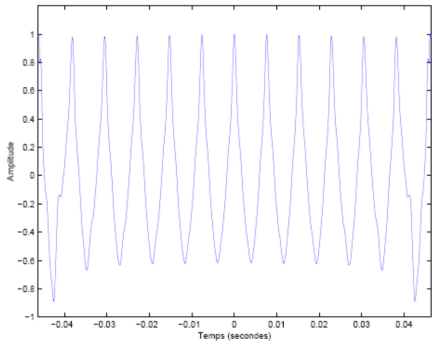


source : Richard, 2012

Auto-corrélation normalisée

$$\bar{r}_x(m) = \frac{\sum_{n=0}^{N-1-m} x(n)x(n+m)}{\sqrt{\sum_{n=0}^{N-1-m} x(n)^2}\sqrt{\sum_{n=0}^{N-1-m} x(n+m)^2}}$$

- $|\bar{r}_x(m)| \leqslant \bar{r}_x(0) = 1$
- $|\bar{r}_x(m)| = 1$ ssi les vecteurs sont colinéaires

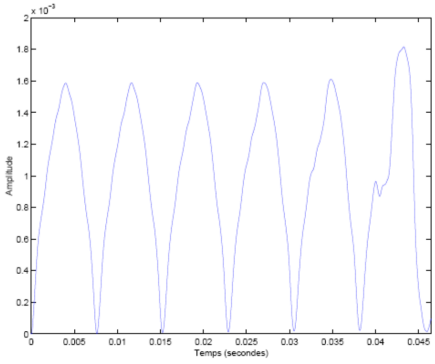


source : Richard, 2012

Average Square Difference Function (ASDF)

$$ASDF(m) = \frac{1}{N-m} \sum_{n=0}^{N-1-m} (x(n) - x(n+m))^2$$

- $ASDF(m) = 0$ ssi x est de période $T_0 = m$
- La période T_0 peut être estimée en recherchant le minimum de l'écart quadratique entre les signaux $x(n)$ et $x(n+m)$
- $E[ASDF(m)] = 2(r_x(0) - r_x(m))$

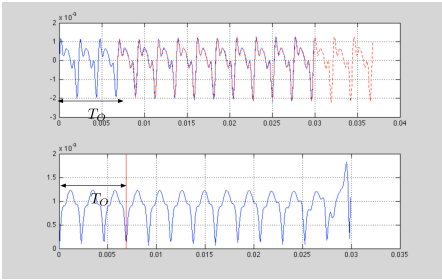


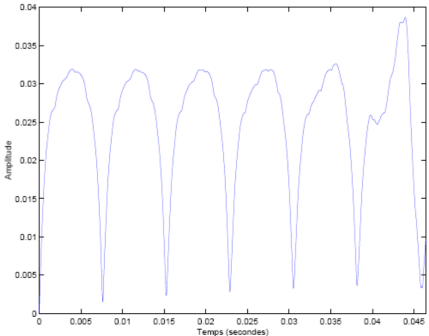
source : Richard, 2012

Average Magnitude Difference Function (AMDF)

$$AMDF(m) = \frac{1}{N-m} \sum_{n=0}^{N-1-m} |x(n) - x(n+m)|$$

- $AMDF(m) = 0$ ssi x est de période $T_0 = m$





Algorithme Yin

[A. de Cheveigné, H. Kawahara, YIN, a fundamental frequency estimator for speech and music, JASA, 2002]

- Point de départ : méthode de l'auto-corrélation
- Améliorations
  - 1) Utilisation de l'ASDF
  - 2) Normalisation
  - 3) Seuillage
  - 4) Interpolation
  - 5) Minimisation locale en temps

Version	Gross error (%)
Step 1	10.0
Step 2	1.95
Step 3	1.69
Step 4	0.78
Step 5	0.77
Step 6	0.50

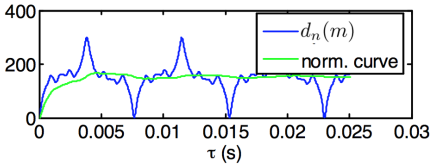
source : Richard, 2012

Algorithme Yin

- 1) Utilisation de l'ASDF
  - $d_t(\tau) = \sum_{j=t+1}^{t+W}(x_j - x_{j+\tau})^2$
  - lien avec l'auto-corrélation $d_t(\tau) = r_t(0) + r_{t+\tau}(0) - 2r_t(\tau)$
  - Gain : l'ASDF est beaucoup moins sensible aux variations des amplitudes relatives que l'ACF (qui est sensible, par exemple, à l'accentuation des partiels d'ordre pair)
- 2) Normalisation
  - Normalisation par la "moyenne cumulée"

$$\begin{aligned} d'_t(\tau) &= 1 && \text{si } \tau = 0 \\ &= \frac{d_t(\tau)}{\frac{1}{\tau}\sum_{j=1}^{\tau} d_t(j)} && \text{sinon} \end{aligned} \tag{9}$$

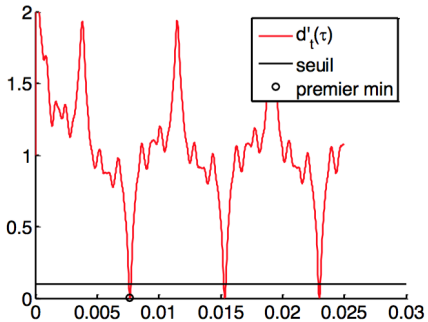
  - Gain : permet d'éviter les erreurs pour les F0 élevées (suppression du lobe en 0)

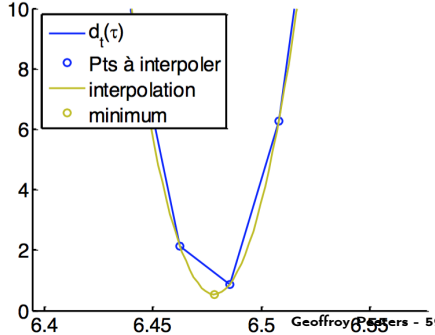


source : Richard, 2012

Algorithme Yin

- 3) Seuillage absolu
  - La plus petite période inférieure au seuil est choisie
  - Si aucune période n'est inférieure au seuil, alors le minimum global est choisi
- 4) Interpolation parabolique autour du minimum
  - Réalisée sur $d_t(\tau)$ (i.e avant normalisation)
  - Gain : meilleure précision sur la valeur de F_0
- 5) Minimisation locale en temps
  - si on note T_t la période estimée au temps t
  - pour un temps t,
    - on cherche pour $\theta \in [t - T_{\max}/2, t + T_{\max}/2]$ ($T_{\max}$ est la période la plus grande considérée, 25 ms)
    - le minima de $d_\theta(T_\theta)$
  - on réitère avec cette nouvelle estimation et intervalle de recherche de $\pm 20\%$
  - Gain : effet de lissage en cas de fluctuations de l'estimation
- Autres méthodes possibles pour le lissage : filtre médian, programmation dynamique





Algorithme Yin

Evaluation sur quatre bases de données de parole

- annotées automatiquement (par YIN, à partir du laryngographe) puis vérifiées et triées à la main

	Gross error (%)					
Method	DB1	DB2	DB3	DB4	Average	(low/high)
pda	10.3	19.0	17.3	27.0	**16.8**	(14.2/2.6)
fxac	13.3	16.8	17.1	16.3	**15.2**	(14.2/1.0)
fxcep	4.6	15.8	5.4	6.8	**6.0**	(5.0/1.0)
ac	2.7	9.2	3.0	10.3	**5.1**	(4.1/1.0)
cc	3.4	6.8	2.9	7.5	**4.5**	(3.4/1.1)
shs	7.8	12.8	8.2	10.2	**8.7**	(8.6/0.18)
acf	0.45	1.9	7.1	11.7	**5.0**	(0.23/4.8)
nacf	0.43	1.7	6.7	11.4	**4.8**	(0.16/4.7)
additive	2.4	3.6	3.9	3.4	**3.1**	(2.5/0.55)
TEMPO	1.0	3.2	8.7	2.6	**3.4**	(0.53/2.9)
YIN	0.30	1.4	2.0	1.3	**1.03**	(0.37/0.66)

source : Richard, 2012

Cepstre réel

- **Auto-correlation** du signal temporel $\hat{r}(\tau)$:

$$\hat{r}(\tau) = \int_t x^\star(t)x(t+\tau)dt$$

  - Sa Transformée de Fourier $\Gamma(\omega)$:

$$\Gamma(\omega) = \int_\tau \left(\int_t x^\star(t)x(t+\tau)dt \right) e^{-j\omega\tau} d\tau$$
$$\Gamma(\omega) = |X(j\omega)|^2$$

-

- Donc **Auto-correlation** du signal temporel :

$$\hat{r}(l) = \frac{1}{N-l} \sum_k |X(k)|^2 \cos\left(2\pi k \frac{l}{N}\right)$$

- **Cepstre réel** du signal temporel :

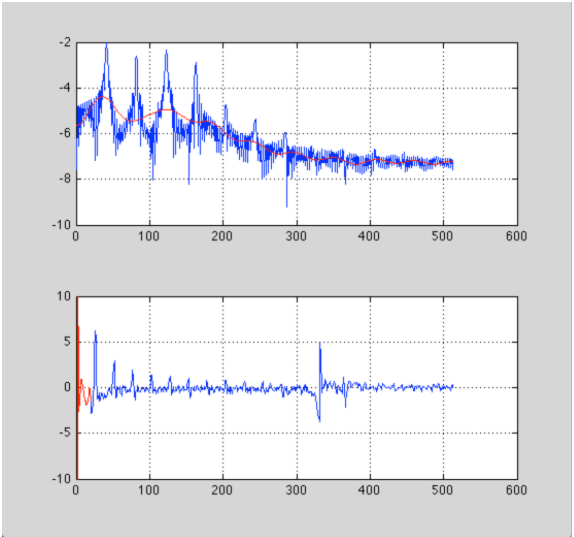
$$\hat{c}(l) = \frac{1}{N-l} \sum_k \log(|X(k)|) \cos\left(2\pi k \frac{l}{N}\right)$$

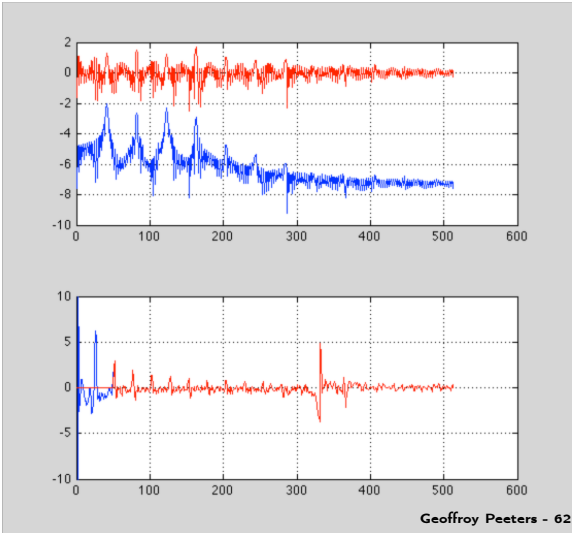
- Relation avec le modèle source/filtre :

$$x(t) = e(t) \circledast g(t)$$
$$X(\omega) = E(\omega) \cdot G(\omega)$$
$$\log(X(\omega)) = \log(E(\omega)) + \log(G(\omega))$$

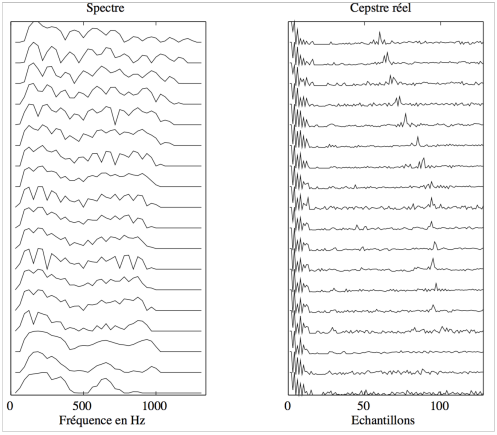
Cepstre réel

- Le cepstre permet de séparer
  - l'enveloppe spectrale
    - ce qui varie lentement
    - basse fréquence de la TF^{-1}
  - **la fréquence fondamentale**
    - ce qui varie rapidement
    - haute fréquence de la TF^{-1}

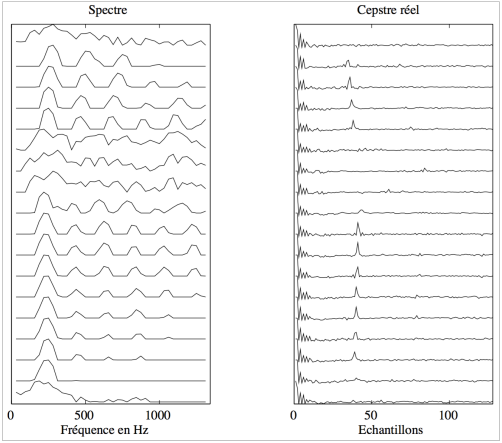




Cepstre réel



source : voix d'homme, Laroche, 1995



source : voix de femme, Laroche, 1995

Approche par le maximum de vraisemblance

- Modèle de signal : $x(n) = a(n) + w(n)$
  - a est un signal périodique de période T_0
  - w est un bruit blanc gaussien de variance σ^2
- vraisemblance des observations

$$p(x|T_0, a, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{N/2}} e^{-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(x(n)-a(n))^2} \tag{10}$$

- log-vraisemblance

$$L(T_0, a, \sigma^2) = -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(x(n) - a(n))^2 \tag{11}$$

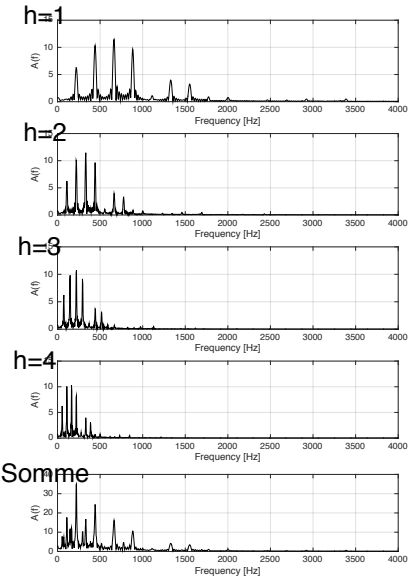
- Méthode :
  - maximiser successivement L par rapport à a, puis σ^2 et enfin T_0

Somme spectrale

- On peut montrer que la maximisation de L par rapport à $F_0 = \frac{m}{N}$ revient à maximiser la somme spectrale

$$S(e^{j2\pi\frac{m}{N}}) = \sum_{h=1}^{H} \hat{R}_x(e^{j2\pi\frac{m}{N}\cdot h})$$

$$S(\omega) = \sum_{h=1}^{H} |X(e^{j\omega\cdot h})|^2 \text{ pour } \omega < \frac{\pi}{H}$$

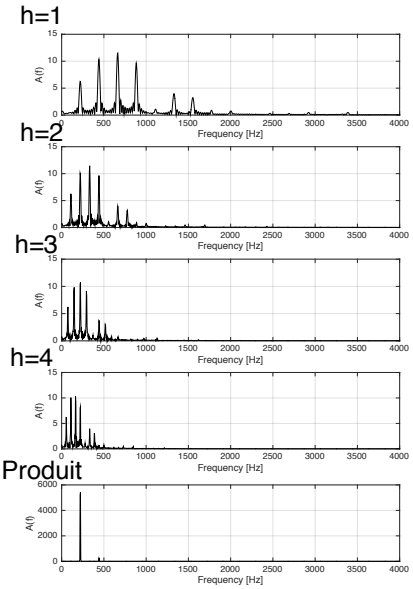


Produit spectral

- Par similitude avec la somme spectrale on peut définir le produit spectral (souvent plus robuste)

$$P(e^{j2\pi\frac{m}{N}}) = \prod_{h=1}^{H} \hat{R}_x(e^{j2\pi\frac{m}{N}\cdot h})$$

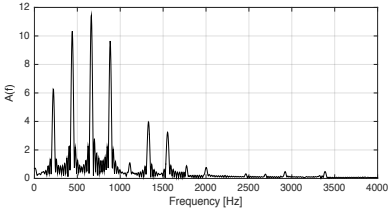
$$P(\omega) = \prod_{h=1}^{H} |X(e^{j\omega\cdot h})|^2 \text{ pour } \omega < \frac{\pi}{H}$$

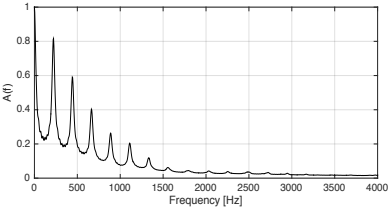


Auto-corrélation du spectre d'amplitude

- Mesure de la périodicité de l'espacement entre les harmoniques
  - ne fait pas l'hypothèse qu'il existe de l'énergie à la fréquence f_0

$$\hat{R}(k) = \frac{1}{N-k} \sum_{\kappa=0}^{N-k-1} |X(\kappa)||X(\kappa+k)| \tag{12}$$

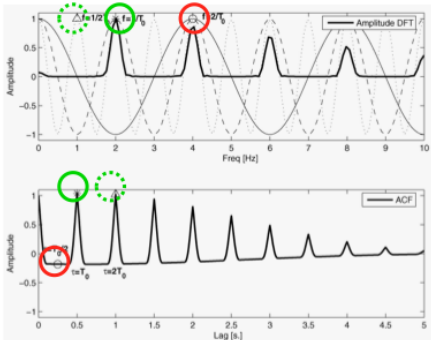




Combinaison de la DFT et de l'ACF

[Geoffroy Peeters, Music pitch representation by periodicity measures based on combined temporal and spectral representations, IEEE ICASSP, 2006]

- Méthodes temporelles $T(\tau_l)$
  - Auto-correlation du signal temporel
  - Cesptre réel du signal temporel
- Méthodes fréquentielles $S(f_k)$
  - Spectre d'amplitude (réassigné fréquentiellement)
  - Auto-correlation du spectra d'ampliture (réassigné fréquentiellement)
- Principe
  - Les erreurs pontielles d'octave sont dans des directions opposées
  - Combiner les deux représentations
- Méthode
  - Calculé les valeurs de la représentation temporelle aux fréquences f_k
    - interpolation) de $T(\tau_l)$ à f_k : $T(1/f_k)$
  - Calculé le produit :
    - $P(f_k) = S(f_k)T(1/f_k)$

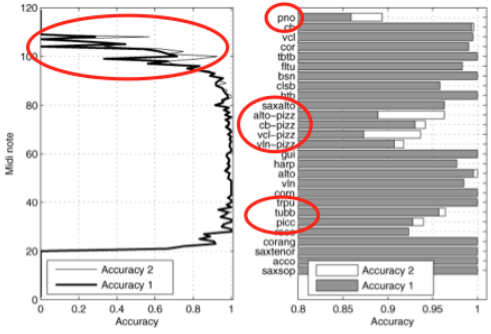


Combinaison de la DFT et de l'ACF

- Résultats

	accuracy 1	accuracy 2
DFT / ACF	81,6	91,7
DFT / CEP	91,4	95,8
ACFofDFT / ACF	95	96,1
ACFofDFT / CEP	**97**	**97,6**
ACFofREAS / CEP	97	97,3
Yin	94,9	95,5

- Résultats

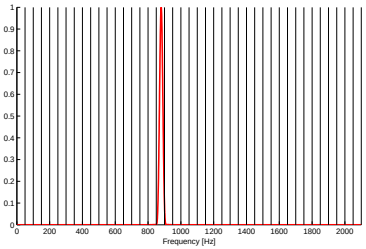


Transformée à Q-Constant (CQT)

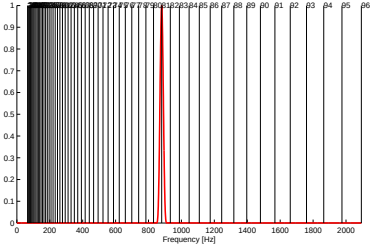
- La DFT
  - Définition : **La précision fréquentielle** : $\Delta f = \frac{sr}{N}$
    - c'est le pas d'échantillonnage du spectre
    - elle dépend de la taille de la DFT : N
    - on peut l'augmenter en augmentant N
  - Définition : **La résolution fréquentielle** : $Bw = \frac{Cw}{L}$
    - c'est le pouvoir de séparation entre deux fréquences présentes simultanément dans le spectre, le pouvoir de résoudre spectralement
  - Attention :
    - même si on augmente N (zero-padding) en gardant L constant on n'améliore pas la résolution !
- Dans la DFT, la précision et la résolution fréquentielle sont constantes à travers les fréquences

Transformée à Q-Constant (CQT)

- En audio musical
  - les fréquences sont logarithimiquement espacées
    - pour passer des fréquences aux hauteurs de notes : $m_k = 12 \cdot log_2 \frac{f_k}{440} + 69$
    - pour passer des hauteurs de notes aux fréquences : $f = 440 \cdot 2^{\frac{m-69}{12}}$
  - les hauteurs de notes sont plus rapprochées en basses fréquences, plus espacées en hautes fréquences
- La **résolution fréquentielle** de la DFT
  - n'est pas suffisante pour résoudre les hauteurs de notes adjacentes en basses fréquences,
  - est trop importante en hautes fréquences



Espacement linéaire de la DFT



Espacement logarithmique des hauteurs de notes

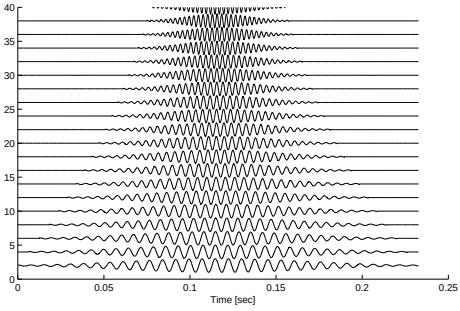
Transformée à Q-Constant

[J. Brown and M. Puckette. An efficient algorithm for the calculation of a constant q transform. JASA, 1992.]

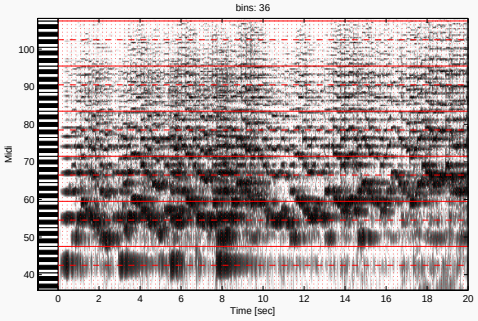
- Solution ?
  - Changer la **résolution fréquentielle** en fonction des fréquences considérées
- Comment ?
  - En changeant la longueur temporelle de la fenêtre pour chaque fréquence considérée
  - Le facteur $Q = \frac{f_k}{f_{k+1} - f_k}$ doit rester constant en fréquence

$$Q = \frac{f_k}{Bw} = \frac{f_k}{Cw/L} = \frac{f_k \cdot L}{Cw}$$

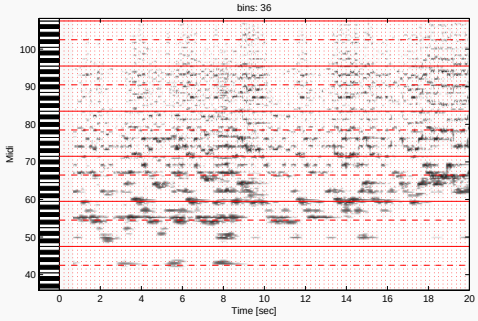
  - on choisit un L pour chaque fréquence f_k
    - $L_k = \frac{Q \cdot Cw}{f_k}$



Exemples (en utilisant la DFT)

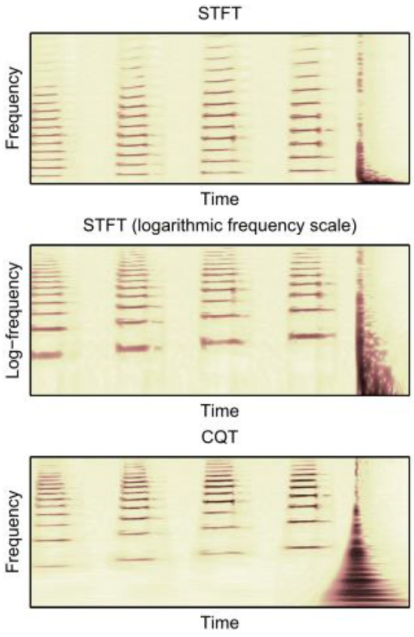


Exemples (en utilisant la CQT)



Transformée à Q-Constant (CQT)

- Sur une transformée à Q constant :
  - Une différence de pitch correspond à une translation sur l'axe des fréquences

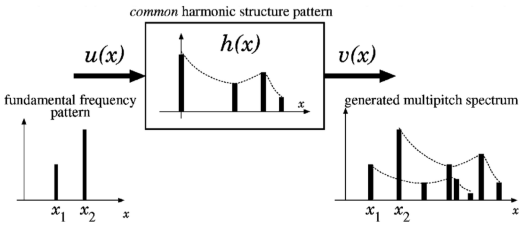


source : Richard, 2012

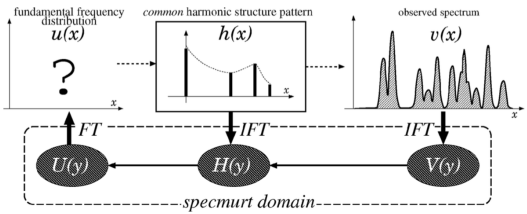
SpecMurt

[S. Saito, H. Kameoka, K. Takahashi, T. Nishimoto, and S. Sagayama. Specmurt analysis of polyphonic music signals. IEEE TASLP, 2008]

- En log-fréquence la transposition d'un son devient une translation sur l'axe
- Analyse Specmurt :
  - Le spectre est formé de la convolution des notes $u(x)$ et d'une structure harmonique $h(x)$
    - suppose une structure harmonique commune à toutes les notes
    - La structure harmonique est partagée par toutes les notes à la même trame mais pas nécessairement à des trames différentes (en contraste avec d'autres méthodes comme la NMF)
  - $v(x) = u(x) \star h(x)$
  - IFFT du power-spectrum en log-fréquences : $V(y) = U(y) \cdot H(y)$
  - estimation itérative de $u(x)$ et $h(x)$



source : Duan, Benetos, 2015



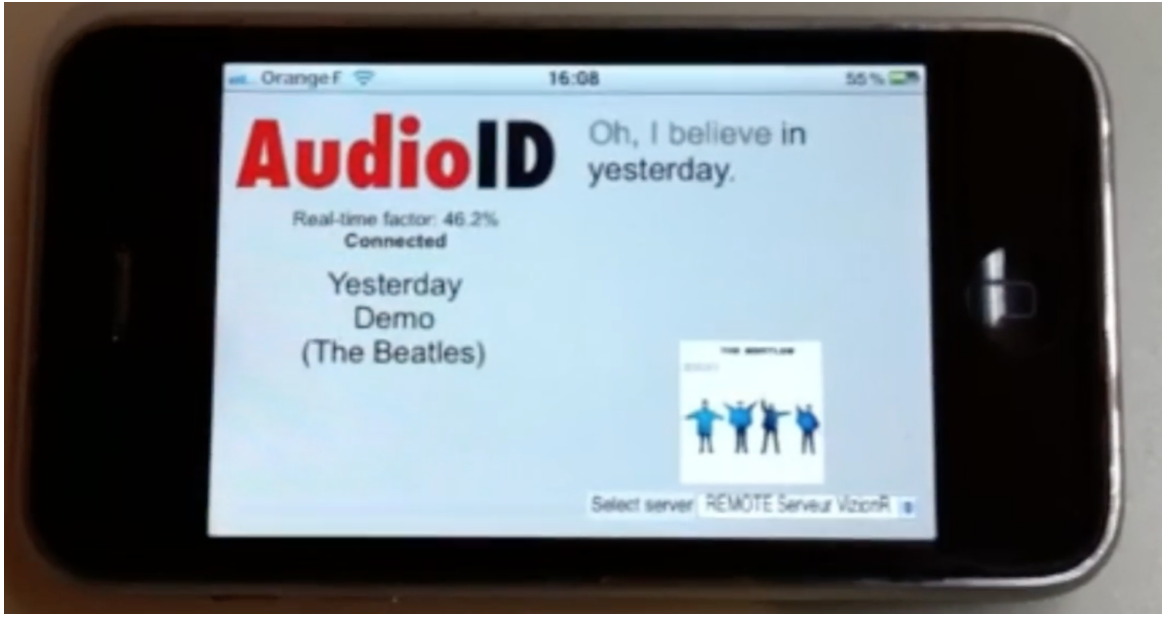
source : Duan, Benetos, 2015

Applications du traitement audio pour la description musicale

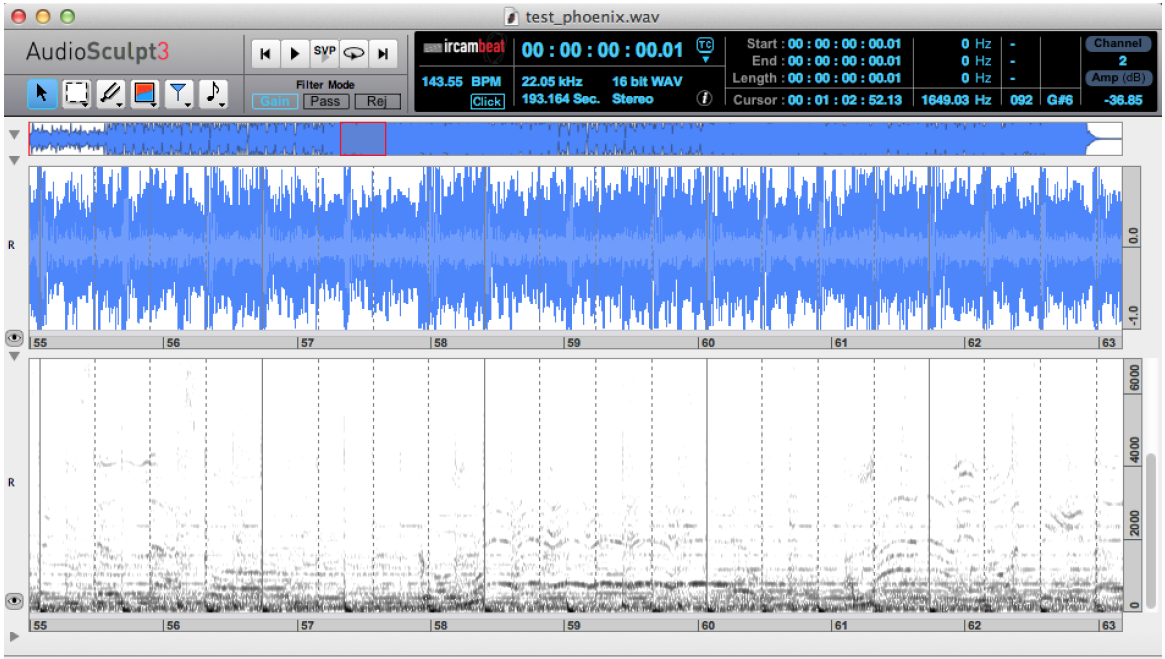

Google™
Audio BETA
Enter a keyword, record a query or drag an example clip.
Search Audio
Audio Preferences
Audio Help
Steve Jobs interview
7 min 14 sec
Speech
Metric - Raw Sugar
3 min 47 sec
Music - Indie Pop
Grenade explosion
23 sec
Sound effect

4- Applications du traitement audio pour la description musicale

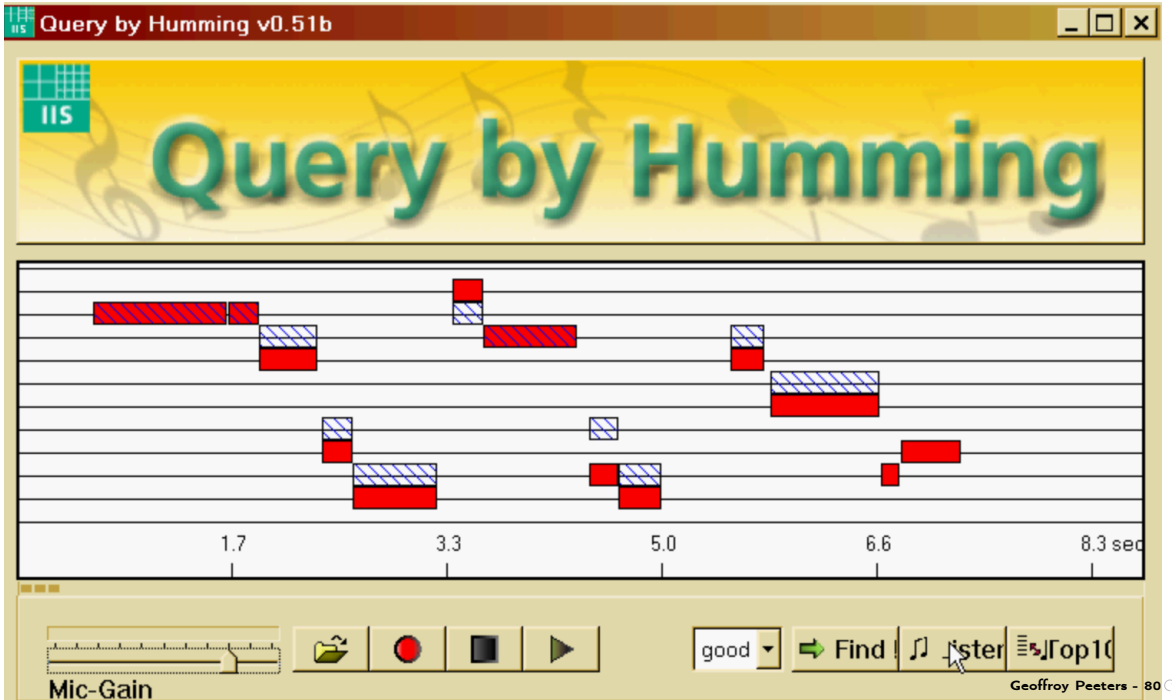
- Identification audio
  - recherche de doublons, gestion de copyright, attacher des méta données à une instance d'un morceau



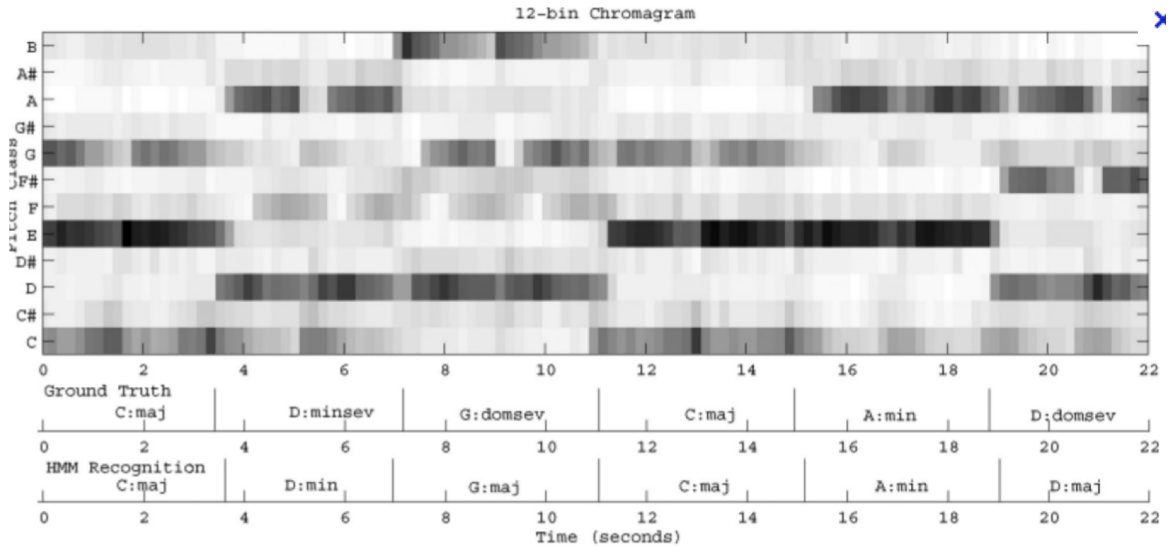
- Estimation du tempo, de la position des temps/ premier-temps
  - DJing, mainpulation du contenu (add swing ...)



- Nouveaux modes de recherche :
  - par chantonnement/ sifflement

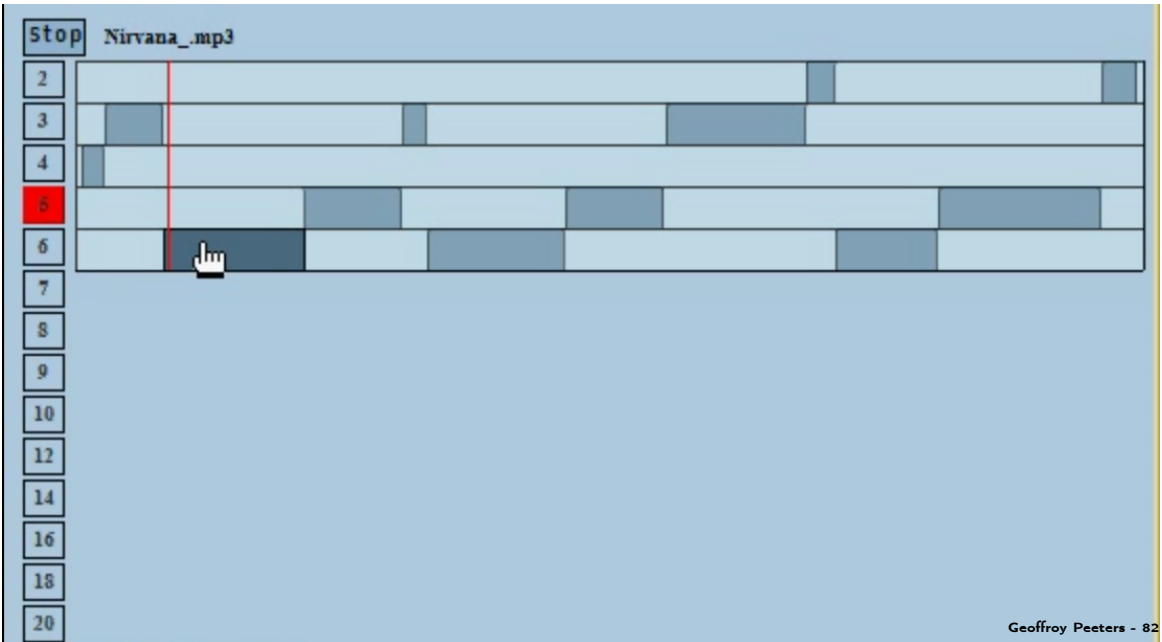


- Estimation des accords
  - obtenir des guitar-tab automatiquement



4- Applications du traitement audio pour la description musicale

- Navigation à l'intérieur d'un morceau de musique par couplet/refrain
  - Génération automatique de résumé audio
- Dé-linéarisation d'un flux audio :
  - segmentation de flux radio, télé et étiquetage des parties

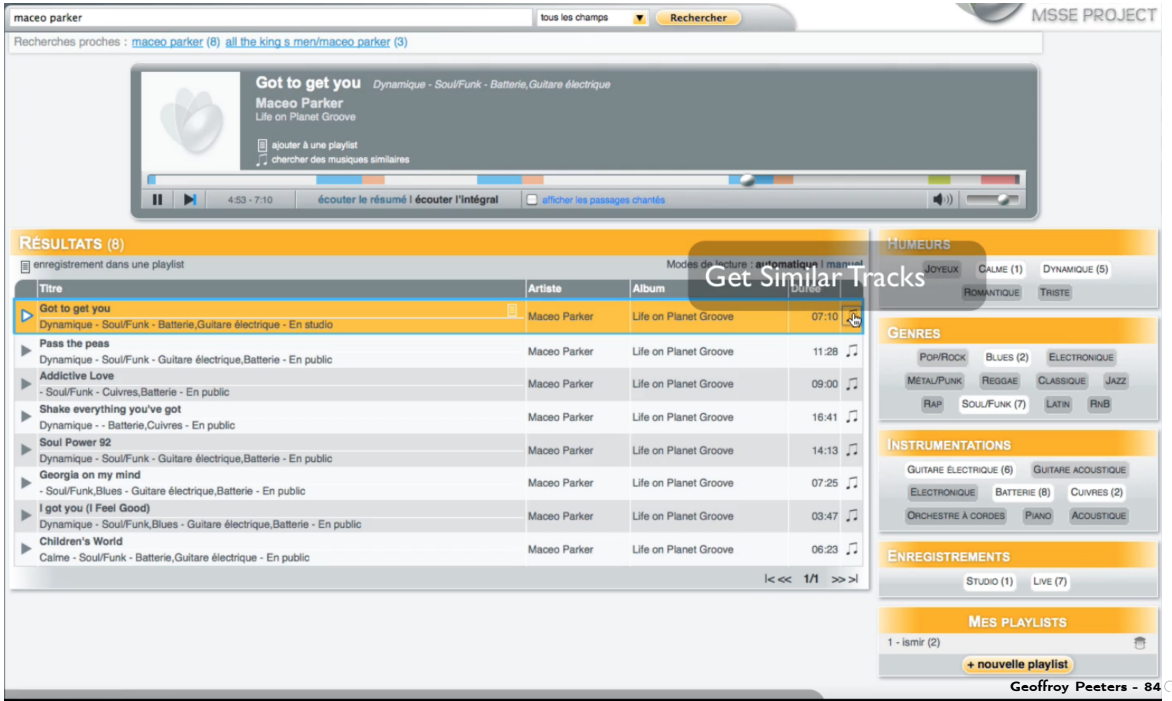


- Détection des cover, reprises ou ... des plagias

Titre	Artiste	Album	D.	Pop.
Let It Be	The Beatles Recovered Band	30 Beatles Top Hits	03:50	
Let It Be	The Hit Co., The Tribute Co.	A Tribute to the Beatles: The Lat...	03:42	
Let It Be	Labrinth	Let It Be	03:05	
Let It Be Me	Ray LaMontagne	Gossip in The Grain	04:41	
Let It Be – The Beatles Tribute	Let It Be	Let It Be – The Beatles Tribute	03:49	
Let It Be	Lois	Let It Be – The Voice 2	03:15	
Let It Be	The Yesteryears	A Tribute to #1 Beatles Hits – T...	03:48	
Let It Be	Aretha Franklin	This Girl's In Love With You	03:33	
Let It Be Sung	Jack Johnson, Matt Costa, Zach Gill,...	If I Had Eyes	04:09	
Let It Be	Vox Angeli	Gloria	03:26	
Let It Be	Paul McCartney	Good Evening New York City	03:54	
Hey Jude	Let It Be	Hey Jude	03:55	
Let It Be	Joan Baez	Greatest Hits And Others	03:51	

4- Applications du traitement audio pour la description musicale

- Recherche d'un contenu audio dans une base de données
  - autrement que par "artistes", "titres" (Google musical)

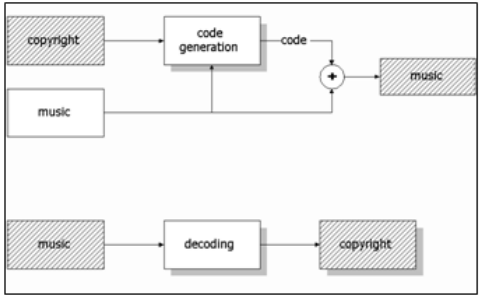


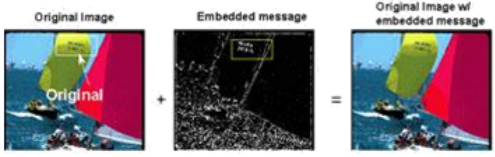
Identification audio

- Objectif :
  - Reconnaître un morceau diffusées sur radio, télé, Internet, bar, discothèque, ...
  - Identifier l'enregistrement (ISRC), pas l'oeuvre (ISWC)

Méthode du Watermarking

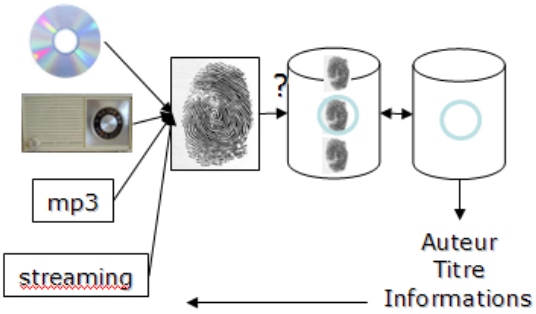
- Codage :
  - introduction d'un code identifiant robuste mais inaudible dans le signal sonore
- Décodage :
  - pour un nouveau signal : extraction du code (si il est présent) et recherche de ce code dans une base de données





Méthode du Fingerprint

- Shazam, Midomi, Philips, ...
- Codage :
  - prise d'empreinte du signal, stockage dans une base de données
- Décodage :
  - pour un nouveau signal, prise d'empreinte, comparaison avec les empreintes de la base de données
- Challenge :
  - déterminer un ensemble réduit de descripteurs audio extraits du signal sonore permettant d'identifier de manière unique un extrait musical

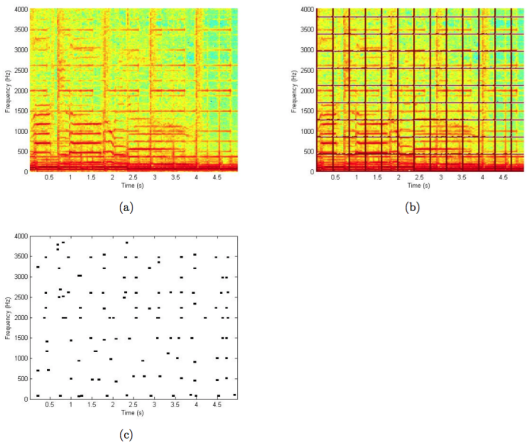


Algorithme de Fingerprint de Shazam

[A. L.-C. Wang. An industrial strength audio search algorithm. In Proc. of ISMIR, 2003.]

[S. Fenet. Empreintes Audio et Stratégies d'Indexation Associées pour l'Identification Audio à Grande Echelle. PhD thesis, Télécom Paris-Tech, 2013.]

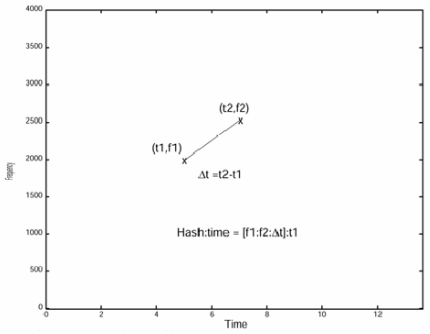
- Extraction de points saillants dans le plan temps/fréquence
  - Calcul du spectrogramme
    - fenêtre de Hamming, L=64 ms, S=32 ms
  - Dans chaque pavé du spectrogramme (Δt=0.4 s, Δf) :
    - détection du maximum → valeur = 1
  - = "constellation points"



source : Sebastien Fenet

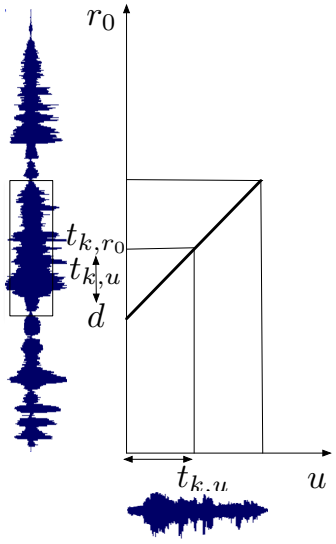
A) Partie stockage de signature

- Représentation des "constellation points" :
  - chaque point est pris comme un "anchor point" ayant une "target zone"
    - $[f_1, f_2, t_2 - t_1]$
    - + le temps de l'anchor t_1
- Méthode de "pruning" des points
  - on ne garde que les pairs de points pour lesquels
    - $f_2 - f_1 < \Delta f_{\max} = 350Hz$
    - $t_2 - t_1 < \Delta T_{\max} = 3s$
- Stockage des triplets
  - $[f_1, f_2, t_2 - t_1]$ stocké sur 32 bits



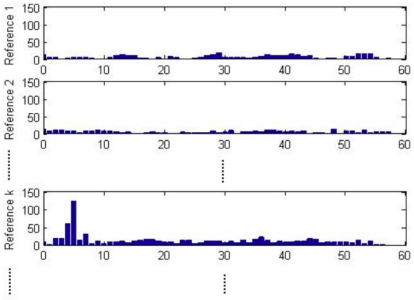
B) Partie matching de signature

- si le signal inconnu u est un extrait de r_0 démarrant au temps d
  - alors toutes les clefs apparaissant dans u doivent être trouvées dans r_0
  - une clef k de u au temps $t_{k,u}$ doit être trouvé dans r_0 au temps $t_{k,r_0} = d + t_{k,u}$
  - si on étudie l'ensemble des valeurs $\{t_{k,r_0} - t_{k,u}\}$ pour toutes les clefs k de u, on doit avoir un maximum d'accumulation en d

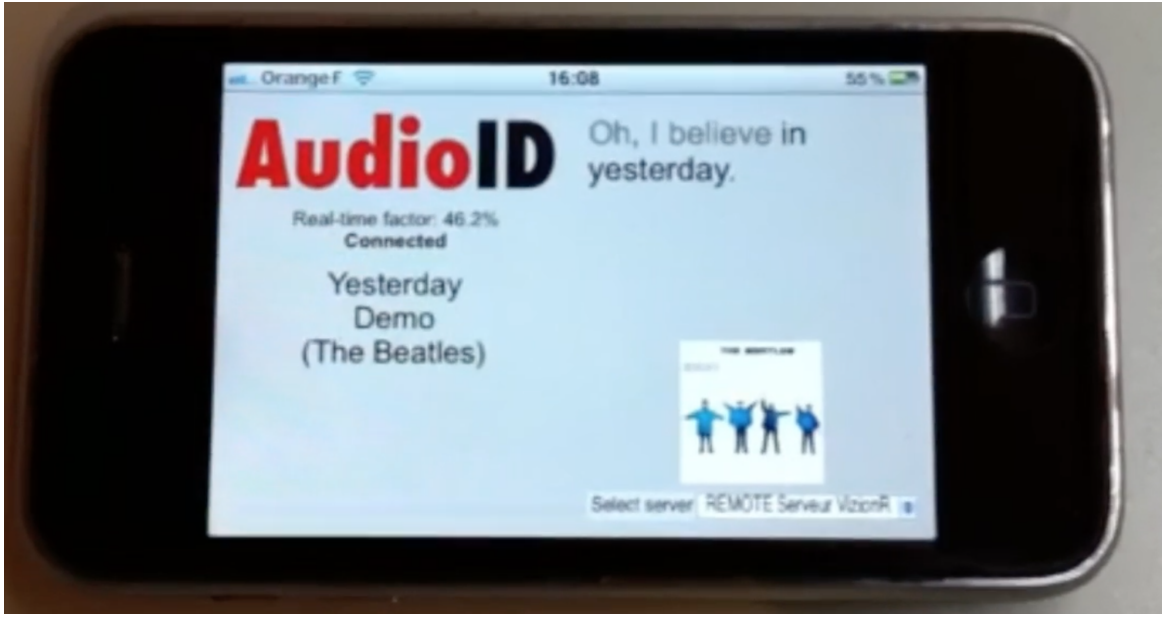


B) Partie matching de signature

- Méthode :
  - pour toutes les clefs k de u , pour chaque référence r_i, on stocke toutes les valeurs $\{t_{k,r_i} - t_{k,u}\}$ dans un histogramme
  - l'histogramme avec le plus grand maximum donne la référence du signal inconnu
  - la position du maximum dans cet histogram donne le point de démarrage d dans le signal de référence



source : Sebastien Fenet


Orange F
16:08
55 %
AudioID
Oh, I believe in yesterday.
Real-time factor: 46.2%
Connected
Yesterday
Demo
(The Beatles)
Select server
REMOTE Serveur VizionR

5- Descripteurs audio

Les 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.]

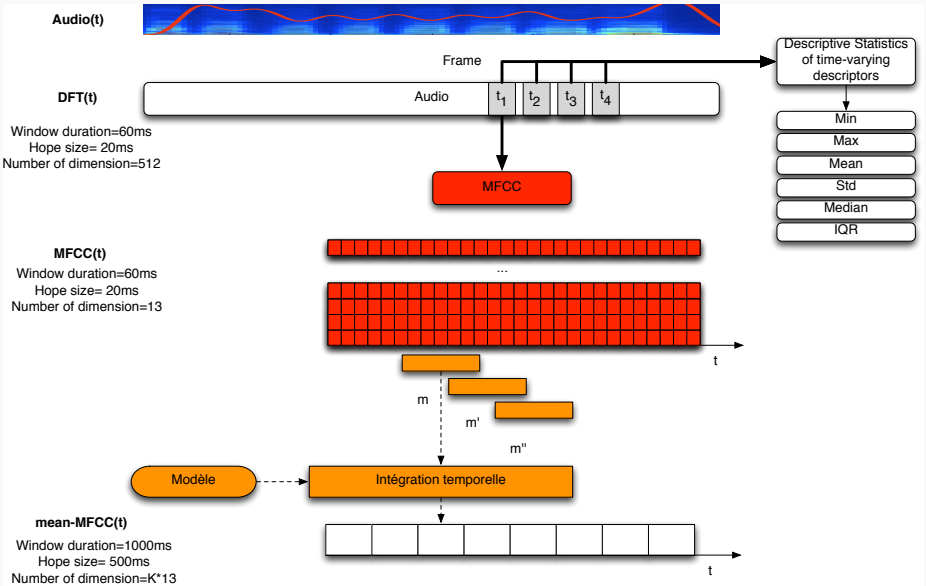
- Valeurs numériques extraites du signal audio dont le but est de représenter une propriété particulière de son contenu
  - Tout est dans la forme d'onde, dans la TFCT, diffiçile à lire, trop grande dimension
- Contrainte :
  - on veut le même nombre de dimensions pour toutes les données
- Extraction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

Les 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.]

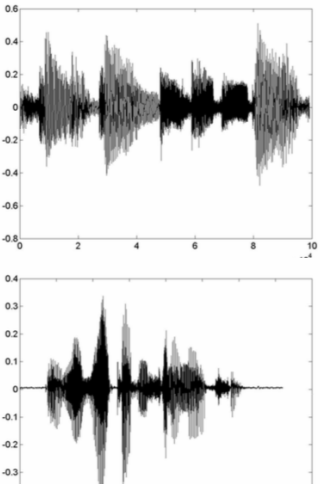
- Différentes **formes** :
  - **scalaire** : Centroide spectral, étendue spectrale, fréquence fondamentale, spectral roll-off, spectral flux, zero-crossing rate, RMS, ...
  - **vecteur** : Mel Frequency Cesptral Coefficients, coefficients LPC, coefficients PLP ...
- Différentes **temporalité** :
  - 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** : ...

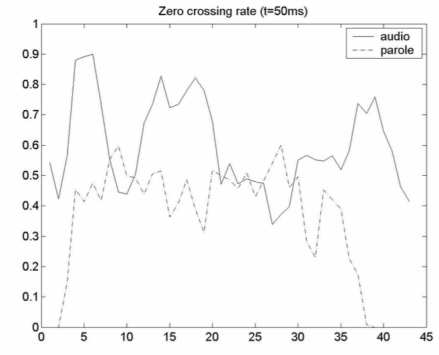
Les descripteurs audio



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

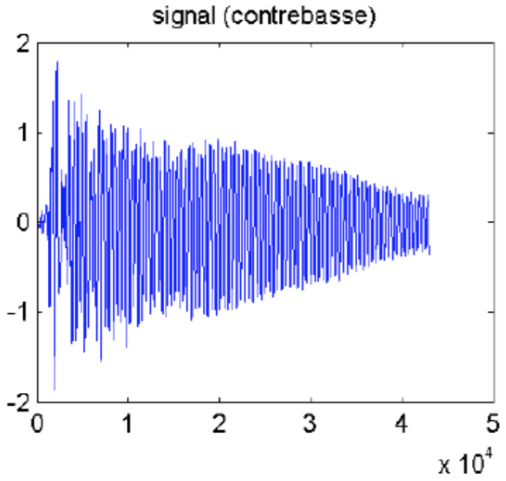


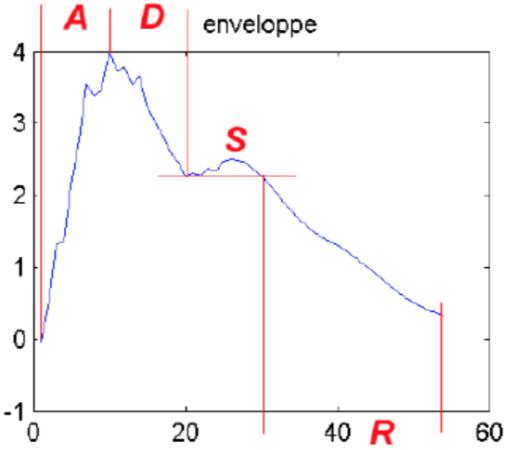


source : Gaël Richard

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)

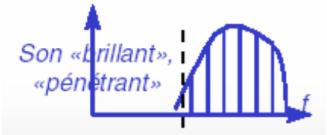




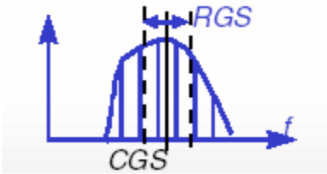
Description du spectre (barycentre, étendue spectral)

- **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

Objectif

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

Cepstre complexe

- **Cepstre complexe** $c(\tau)$:

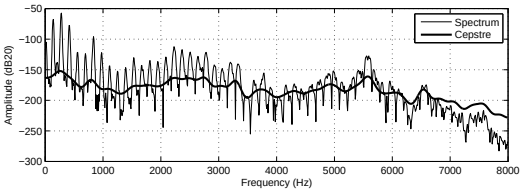
$$\begin{aligned} c(\tau) &= TF^{-1}\left[\log(X(\omega))\right] \\ &= \frac{1}{2\pi}\int_{\omega}\log(X(\omega))e^{j\omega\tau}d\omega \end{aligned} \tag{13}$$

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

Cepstre complexe

- Modèle source/ filtre :
  - Source : signal périodique
  - Filtre : résonant/ anti-résonant

$$x(t) = e(t) \circledast g(t)$$
$$\xrightarrow{TF} X(\omega) = E(\omega) \cdot G(\omega) \quad (14)$$



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

$$\xrightarrow{TF^{-1}} TF^{-1}\left[\log(X(\omega))\right] = \underbrace{TF^{-1}\left[\log(E(\omega))\right]}_{\text{énergie aux céfrences } \tau>>} + \underbrace{TF^{-1}\left[\log(G(\omega))\right]}_{\text{énergie aux céfrences } \tau<<} \quad (15)$$

Cepstre réel

- **Cepstre réel** :
  - Cepstre calculé sur la partie réelle du log-spectrum

$$\begin{aligned} X(\omega) &= A(\omega) \cdot e^{j\phi(\omega)} \\ \log(X(\omega)) &= \log(A(\omega)) + j\phi(\omega) \\ \Re(\log(X(\omega))) &= \log(A(\omega)) \end{aligned} \tag{16}$$

$$\begin{aligned} \text{cepstre réel} &= TF^{-1}\left[\Re(\log(X(\omega)))\right] \\ &= TF^{-1}\left[\log(A(\omega))\right] \\ c(\tau) &= \frac{1}{2\pi}\int_{\omega} \log(A(\omega)) e^{j\omega\tau} d\omega \end{aligned} \tag{17}$$

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

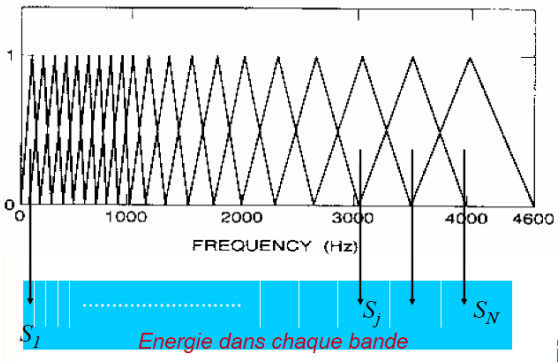
Mel Frequency Cepstral Coefficients (MFCCs)

- **Mel Frequency Cepstral Coefficient** :
  - Cepstre réel calculé sur un spectre d'énergie exprimé en convertissant l'énergie $|X(\omega)|^2$ 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, ...

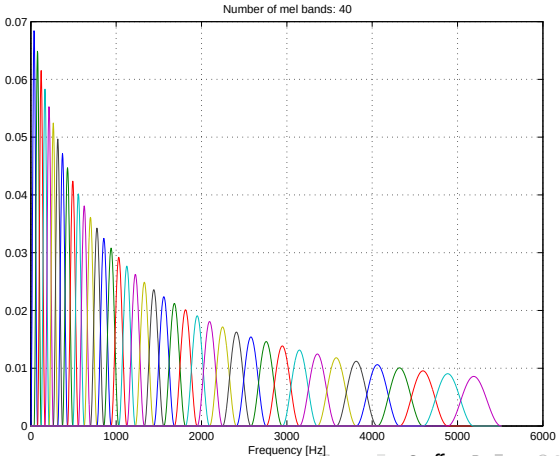
Mel Frequency Cepstral Coefficients (MFCCs)

- Echelle de Mel :

$$M = f \quad \text{pour} \quad f < 1000Hz$$
$$M = f_c \left(1 + \log_{10}\left(\frac{f}{f_c}\right)\right) \quad \text{pour} \quad f \geqslant 1000Hz \tag{18}$$

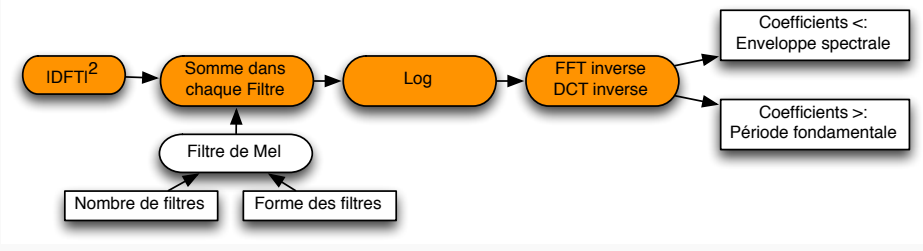


source : Gaël Richard

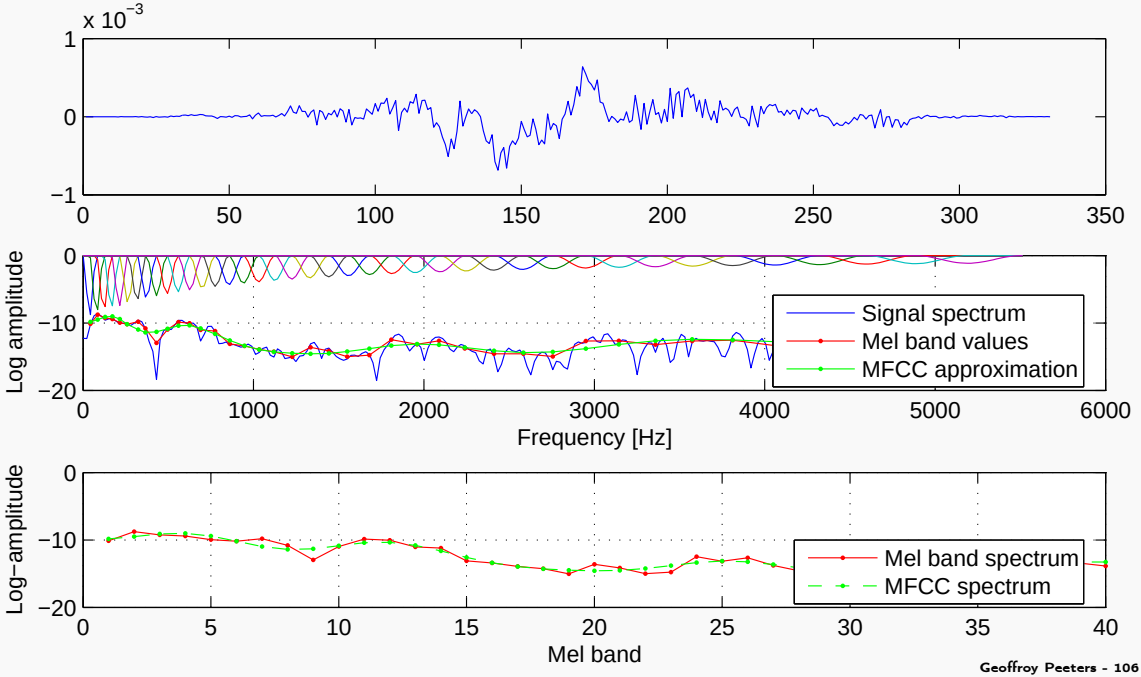


Mel Frequency Cepstral Coefficients (MFCCs)

- Calcul du spectre de puissance : $|X(\omega)|^2$
- Calcul des filtres de Mel : $H_b(\omega)$ 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 : $S(b) = \sum_\omega |X(\omega)|^2 \cdot H_b(\omega)$
- 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

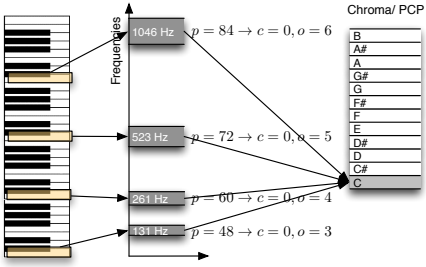


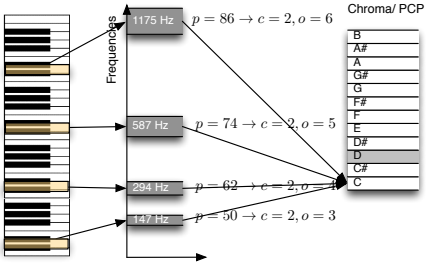
Exemple de calcul de MFCCs



Définition des Chroma - Pitch Class Profile (PCP)

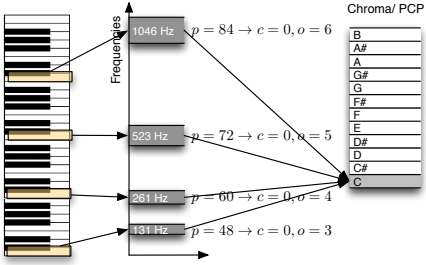
- **Objectif :**
  - le spectre à l'instant n : $X(k, n)$
  - représenter son contenu harmonique sous forme d'un vecteur : $C(c, n)$ $c \in [0, 12[$
- Utilisations :
  - reconnaissance de tonalité,
  - reconnaissance de suite d'accords,
  - détection de "cover versions"
- Shepard-1964 :
  - représenter 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),

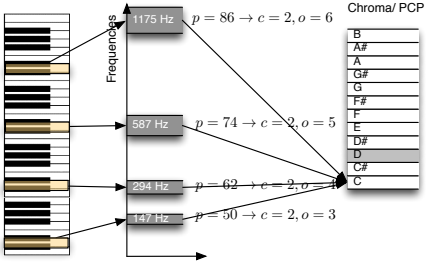




Calcul des Chromas - Pitch Class Profile (PCP)

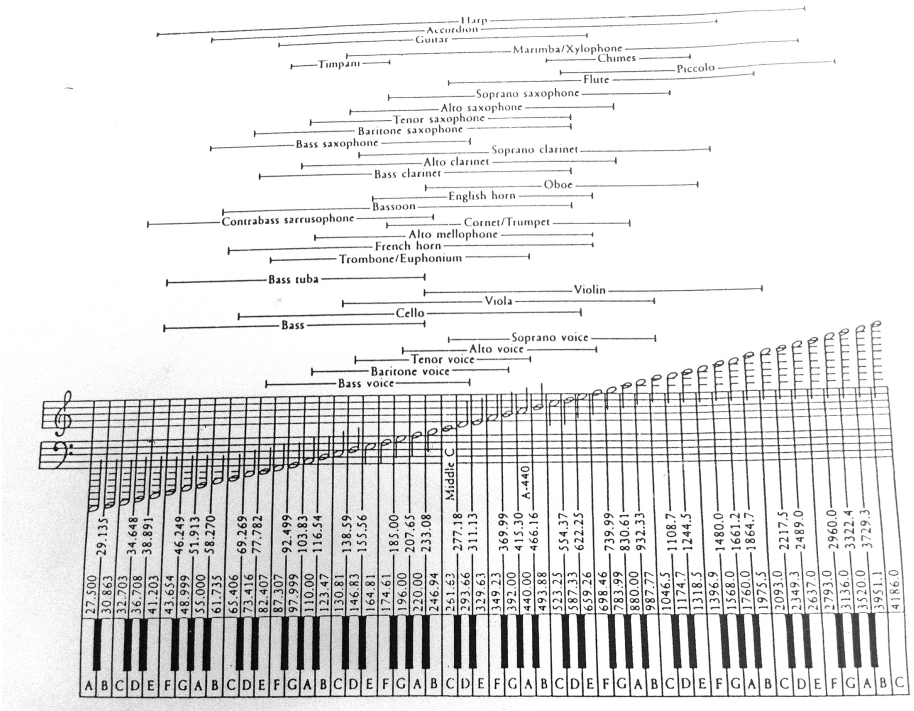
- Relation entre les fréquences f_k de la DFT et les hauteurs de note p (hauteurs de demi-tons en échelle de notes MIDI)
  - $p(f_k) = 12 \log_2 \left(\frac{f_k}{440} \right) + 69, \quad p \in \mathbb{R}^+$
  - $f(p) = 440 \cdot 2^{\frac{p-69}{12}}$
- Calcul des chromas $C(c, n)$
  - On additionne toutes 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}

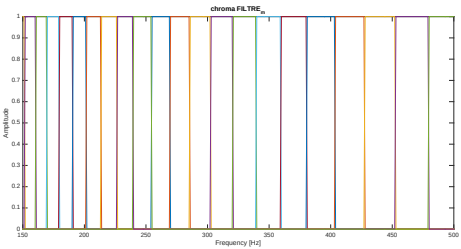

Harp
Accordion
Guitar
Marimba/Xylophone
Timpani
Chimes
Piccolo
Flute
Soprano saxophone
Alto saxophone
Tenor saxophone
Baritone saxophone
Bass saxophone
Soprano clarinet
Alto clarinet
Bass clarinet
Oboe
English horn
Bassoon
Contrabass sarrusophone
Cornet/Trumpet
Alto mellophone
French horn
Trombone/Euphonium
Bass tuba
Violin
Viola
Cello
Bass
Soprano voice
Alto voice
Tenor voice
Baritone voice
Bass voice
Middle C
A-440
27.500 29.135
30.863 32.703 34.648
36.708 38.891 41.203
43.654 46.249 48.999
51.913 55.000 58.270
61.735 65.406 69.269
73.416 77.782 82.407
87.307 92.499 97.999
103.83 110.00 116.54
123.47 130.81 138.59
146.83 155.56 164.81
174.61 185.00 196.00
207.65 220.00 233.08
246.94 261.63 277.18
293.66 311.13 329.63
349.23 369.99 392.00
415.30 440.00 466.16
493.88 523.25 554.37
587.33 622.25 659.26
698.46 739.99 783.99
830.61 880.00 932.33
987.77 1046.5 1108.7
1174.7 1244.5 1318.5
1396.9 1480.0 1568.0
1661.2 1760.0 1864.7
1975.5 2093.0 2217.5
2349.3 2489.0 2637.0
2793.0 2960.0 3136.0
3322.4 3520.0 3729.3
3951.1 4186.0
A B C D E F G A B C D E F G A B C D E F G A B C D E F G A B C D E F G A B C D E F G A B C D E F G A B C D E F G A B C

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

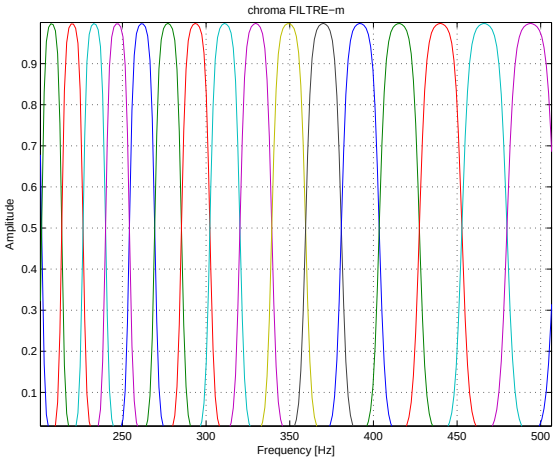
Hard-mapping

- 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

- 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 $p' \in [43, 44, \ldots, 95]$:

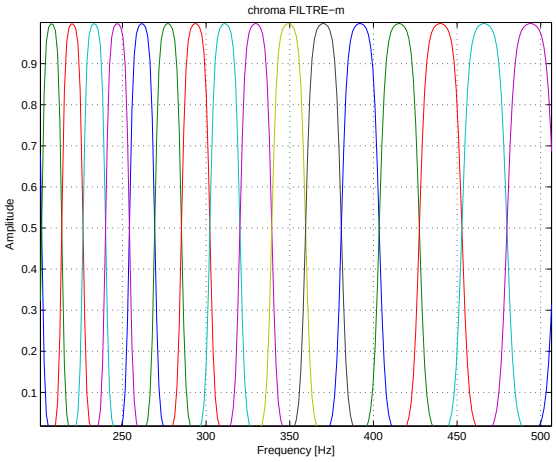


Soft-mapping

- 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'.



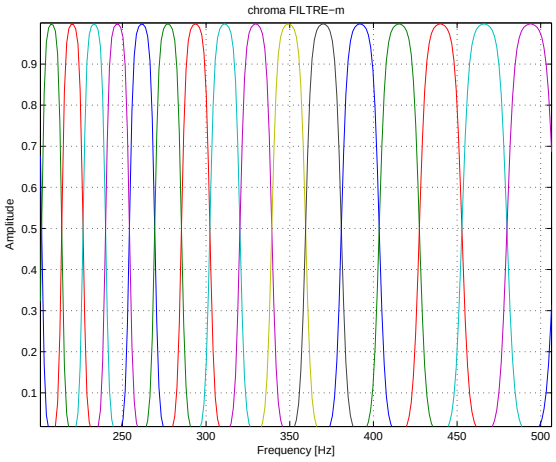
Calcul des Chromas - Pitch Class Profile (PCP)

- 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

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



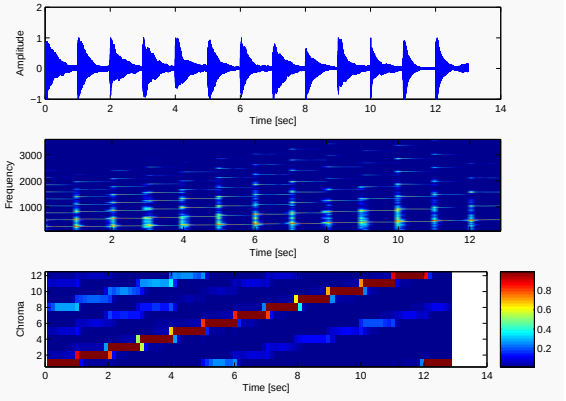
Limitations des Chromas - Pitch Class Profile (PCP)

- 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

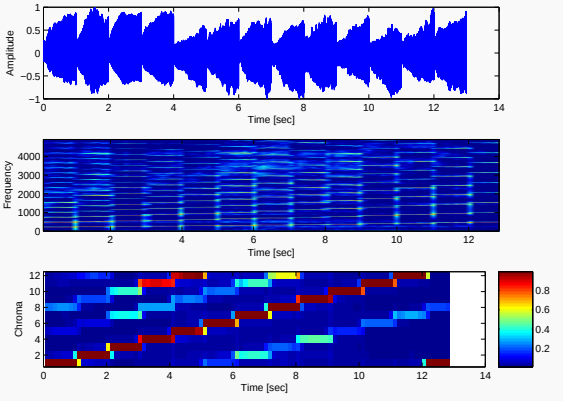
Pitch	Harmonic	Frequency f_μ	MIDI-scale m_μ	Chroma/PCP p
c3	f_0	130.81	48	1 (=c)
	$2f_0$	261.62	60	1 (=c)
	$3f_0$	392.43	67.01	8.01 ($\simeq$ g)
	$4f_0$	523.25	72	1 (=c)
	$5f_0$	654.06	75.86	4.86 ($\simeq$ e)
	...	...	...	...

Limitations des Chromas - Pitch Class Profile (PCP)

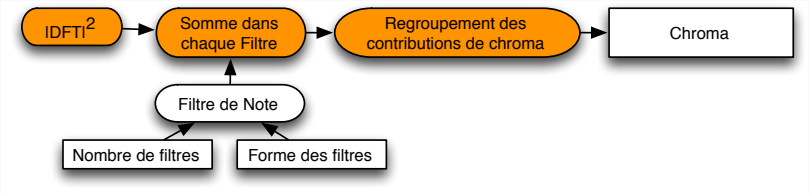
Exemple piano

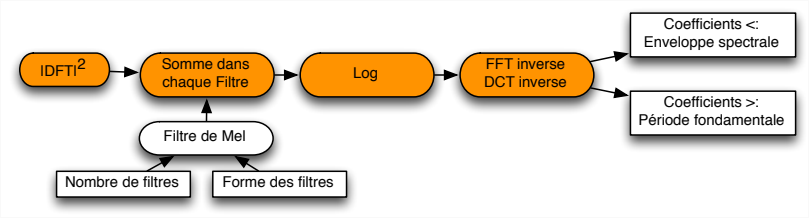


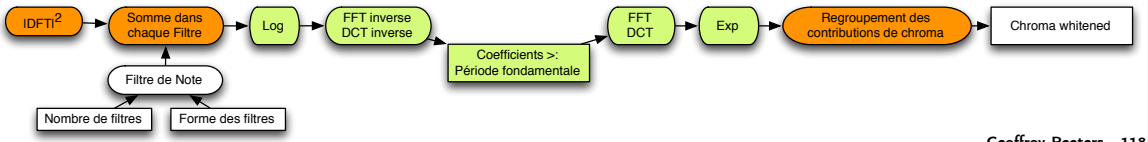
Exemple violon



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

$$SFM = \frac{\left(\prod_{k\in K} a(k)\right)^{1/K}}{\frac{1}{K}\sum_{k\in K} a(k)} \tag{19}$$

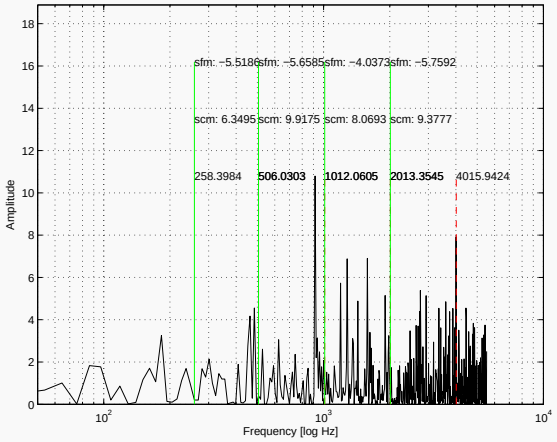
  - SFM$\simeq$0 pour signaux tonaux, SFM$\simeq$1 pour signaux bruités
  - Calcul effectué dans plusieurs bandes de fréquence :
    - $[250-500]$, $[500-1000]$, $[1000-2000]$, $[2000-4000]$ Hz (MPEG-7)

- **Mesure de tonalité** :

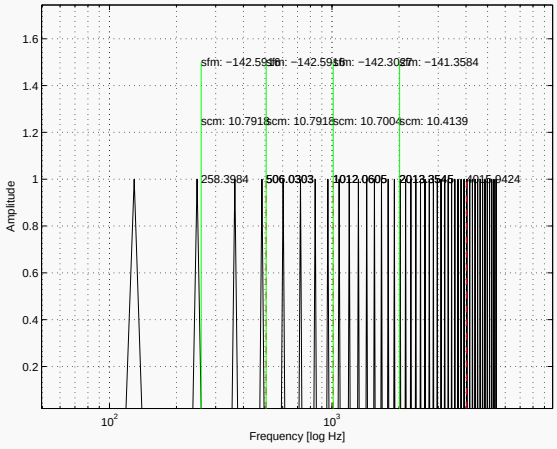
$$SFM_{dB} = 10\log_{10}(SFM) \qquad Tonality = \min\left(\frac{SFM_{dB}}{-60}, 1\right) \tag{20}$$

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

Exemple cas bruité

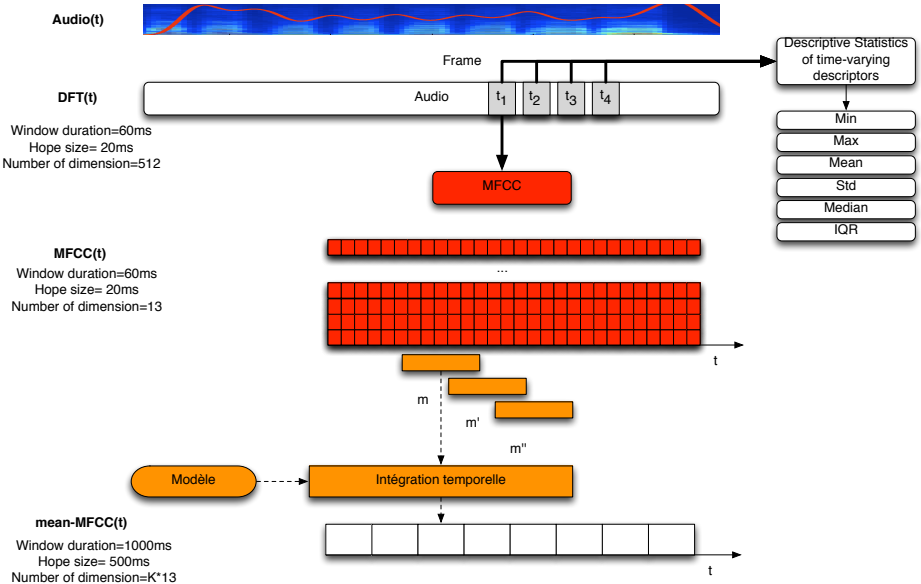


Exemple cas non-bruité



Intégration temporelle

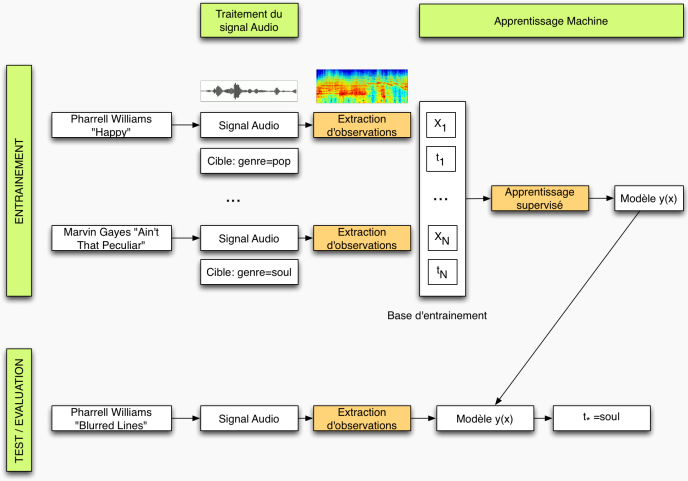
- **Objectifs :**
  - Représenter le comportement temporel des observations
    - Calcul des **dérivées et accélérations** des observations (Δ-MFCC, Δ-Δ-MFCC) $\rightarrow$ permet de représenter le comportement temporel du descripteur au cours du temps
  - Réduire la quantité de données à traiter
    - 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


Audio(t)
Frame
Descriptive Statistics of time-varying descriptors
DFT(t)
Audio
t1
t2
t3
t4
Min
Max
Mean
Std
Median
IQR
Window duration=60ms
Hope size= 20ms
Number of dimension=512
MFCC
MFCC(t)
Window duration=60ms
Hope size= 20ms
Number of dimension=13
...
t
m
m'
m''
Modèle
Intégration temporelle
mean-MFCC(t)
Window duration=1000ms
Hope size= 500ms
Number of dimension=K*13
t

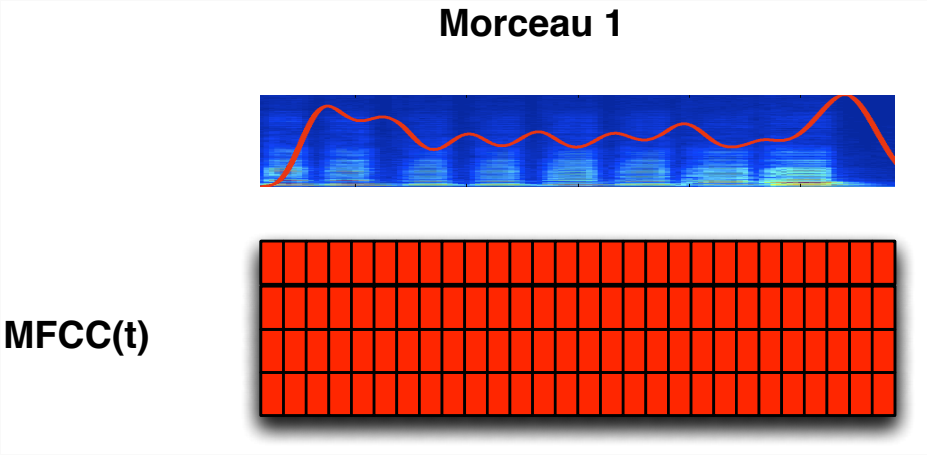
Classification Audio

Classification Audio

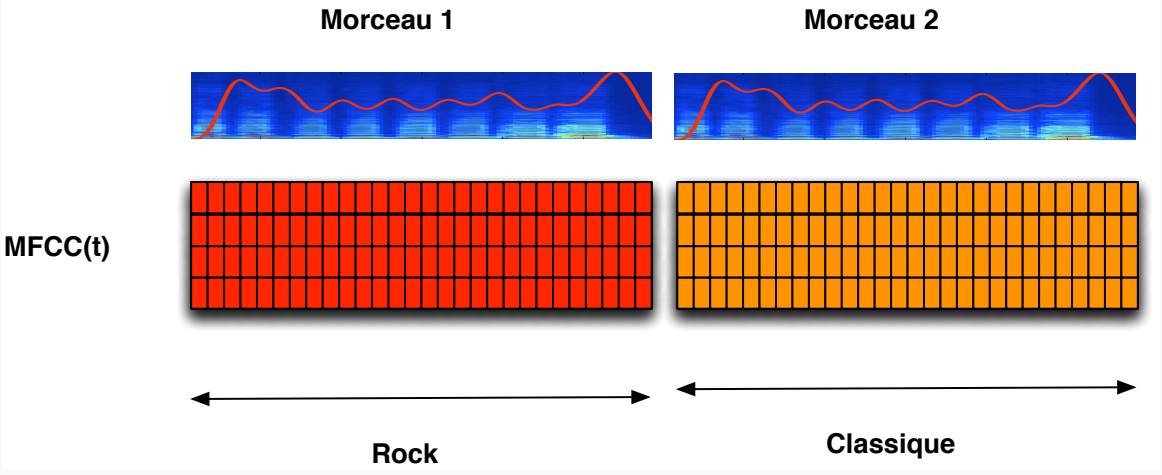
- Utilisation des descripteurs audio en entrée d'un algorithme d'**apprentissage supervisé**
- Exemples d'utilisation :
  - auto-tagging en genre, en mood (humeur), en instrumentation
  - segmentation d'un flux tempos en catégories paroles/musiques, musique instrumentale/chantée



Extraction des descripteurs sur un fichier audio

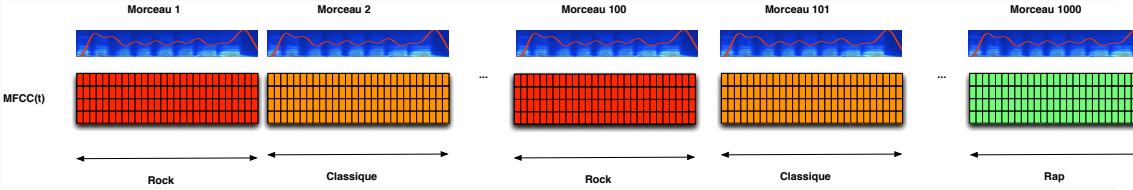


Extraction des descripteurs sur plusieurs fichier audio + assignation des labels de classes aux données

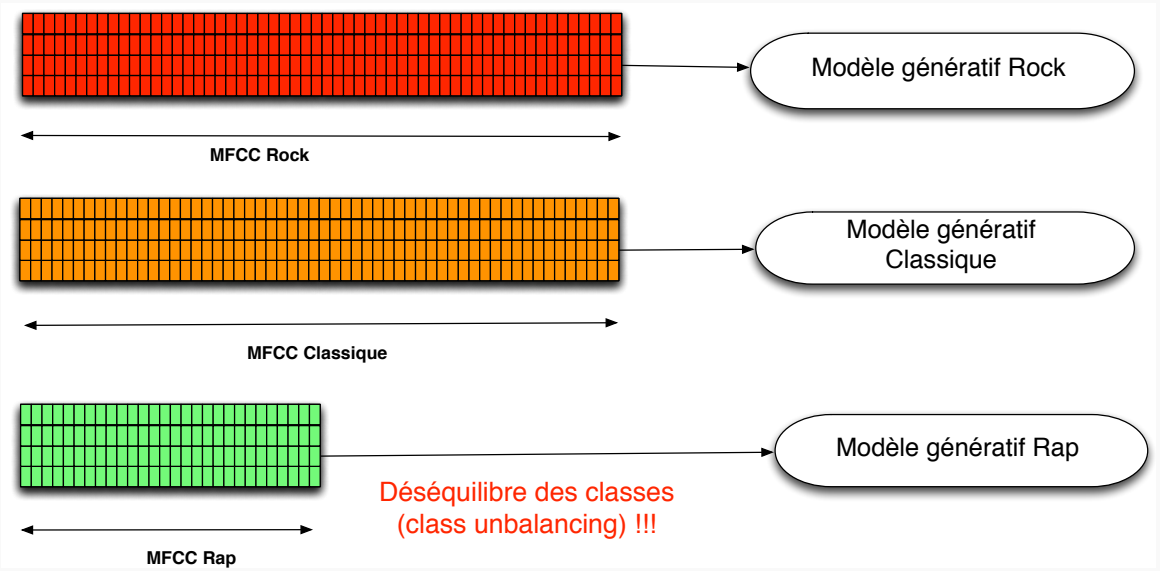


Extraction des descripteurs sur une collection de fichiers + assignation des labels de classes aux données

- La collection peut contenir plusieurs millions de fichiers audio
- Le nombre de labels de classes peut être très important (99 genre musicaux)



Apprentissage



Algorithmes d'apprentissage supervisé

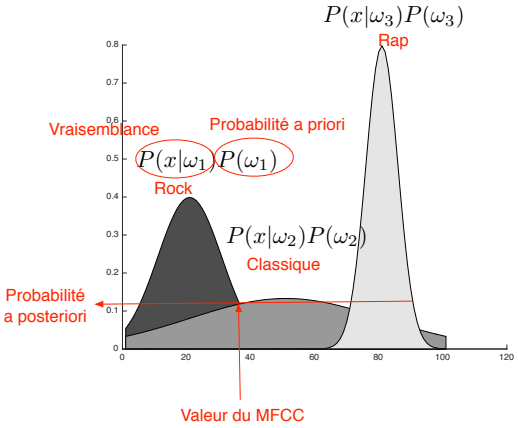
- Modèles génératifs : Gaussien, GMM
- Modèles discriminants : LDA, SVM
- Approche par exemplification : KNN

Modèle génératif gaussien

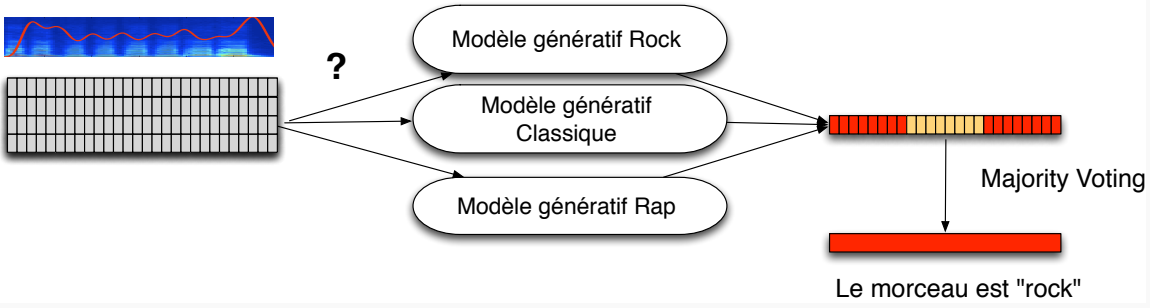
- On modélise chaque classe ω_i par une densité de probabilité gaussienne
  - $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}\Sigma^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\}$
- On applique la règle de décision Bayesienne

$$p(\omega = \omega_i|x) = p(\omega = \omega_j) \cdot \frac{p(x|\omega = \omega_i)}{p(x)}$$

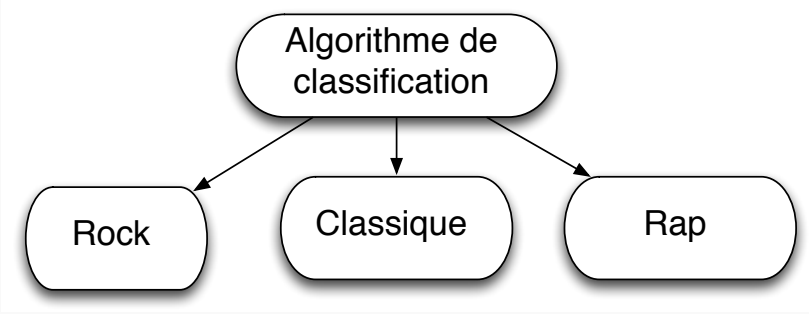
$$\text{posterior} = \text{prior} \cdot \frac{\text{vraisemblance}}{\text{evidence}}$$



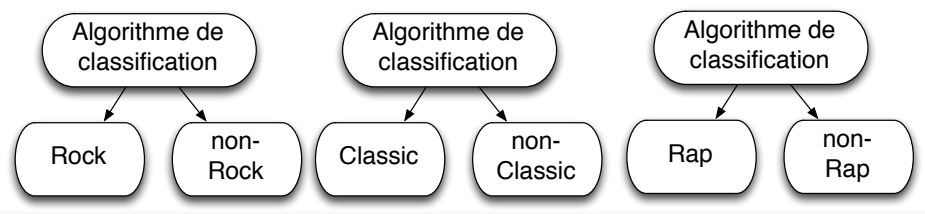
Estimation du label de classe d'un fichier audio inconnu



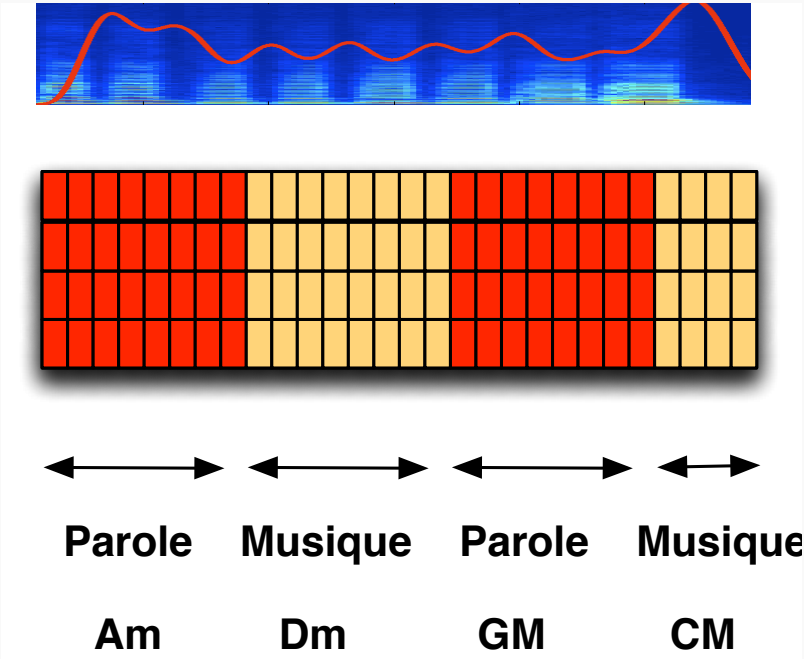
Classificateur multi-classes



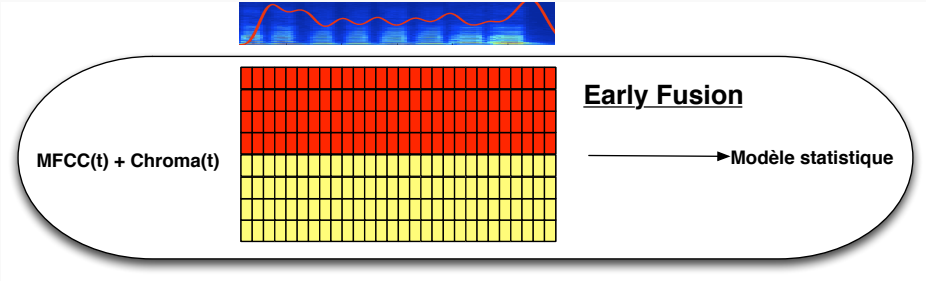
Classificateur binaire (One versus All)



Segmentation = classification local en temps



Early Fusion



Late Fusion