M2 ATIAM

Music Structure Discovery and Audio Summary Generation

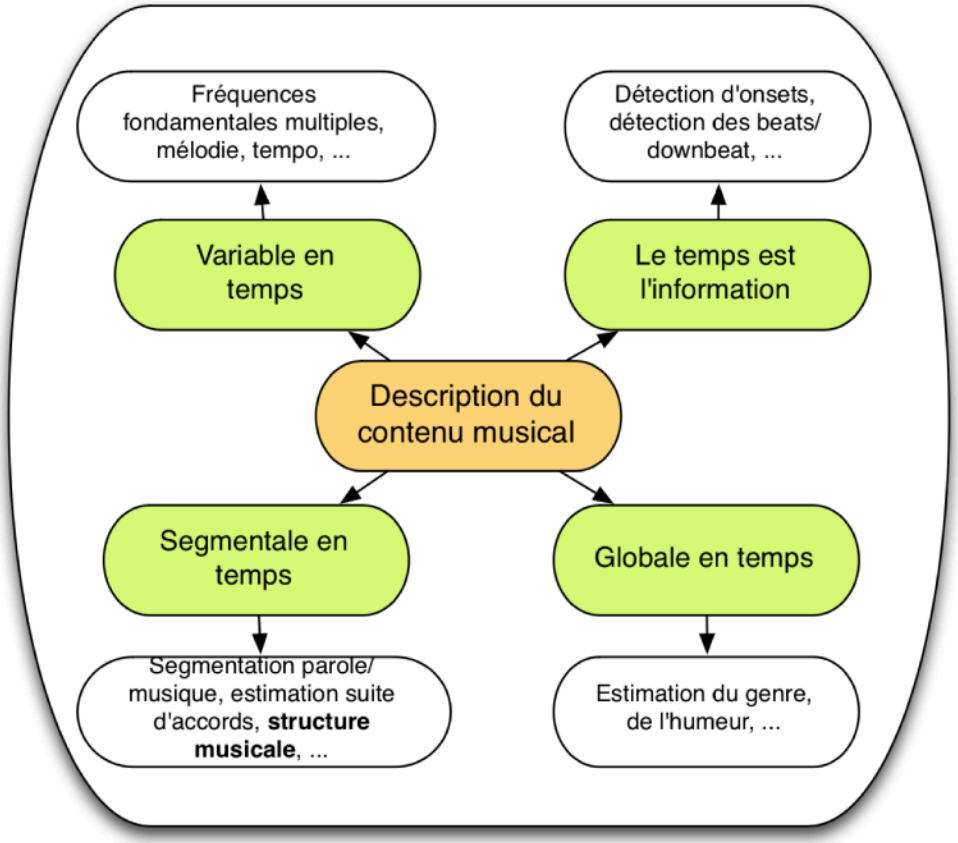
Geoffroy Peeters

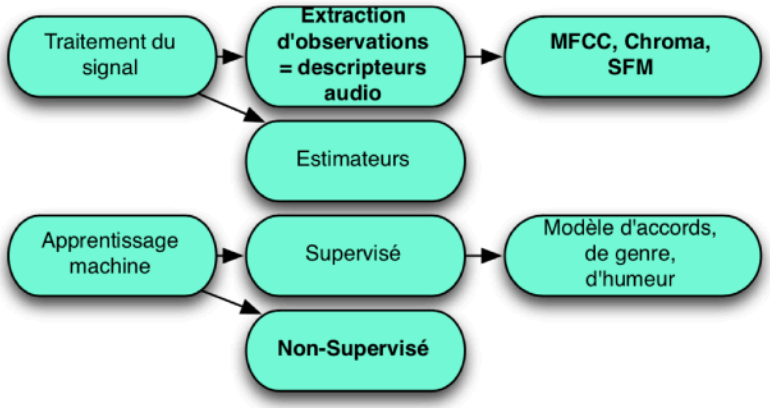
contact: geoffroy.peeters@telecom-paris.fr
Télécom-Paris, IP-Paris, France

Music Structure Discovery (MSD) - Audio Summary

Music Structure Discovery (MSD) - Audio Summary

Different type of description of musical content





Estimation of *the* Musical Structure → of *a* Musical Structure

– **Goal**
- Estimate a structure of a music track
- Automatically generate an **audio summary** which is representative of the content of the music track

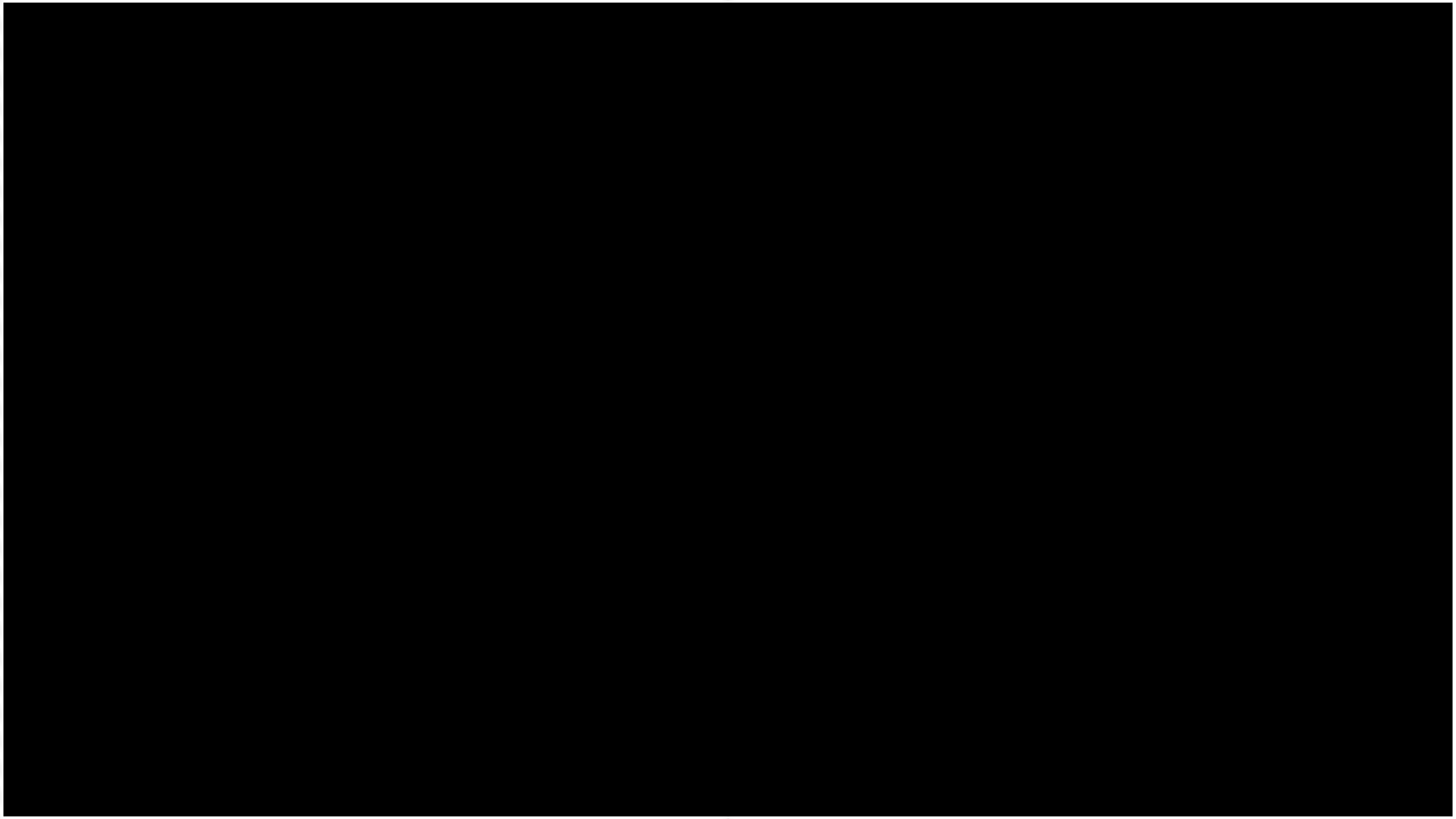
– **Applications**
- Interactive listing:
  – interactive music player
- Fast preview of music track
- **Audio and video examples**

– **Systems**
- Audio features extraction
- Structure visualisation
- Structure estimation
  – Traditional approaches: **unsupervised** learning
  – Recent approaches: **supervised** learning

source : Quaero project, MSSE-Orange interface

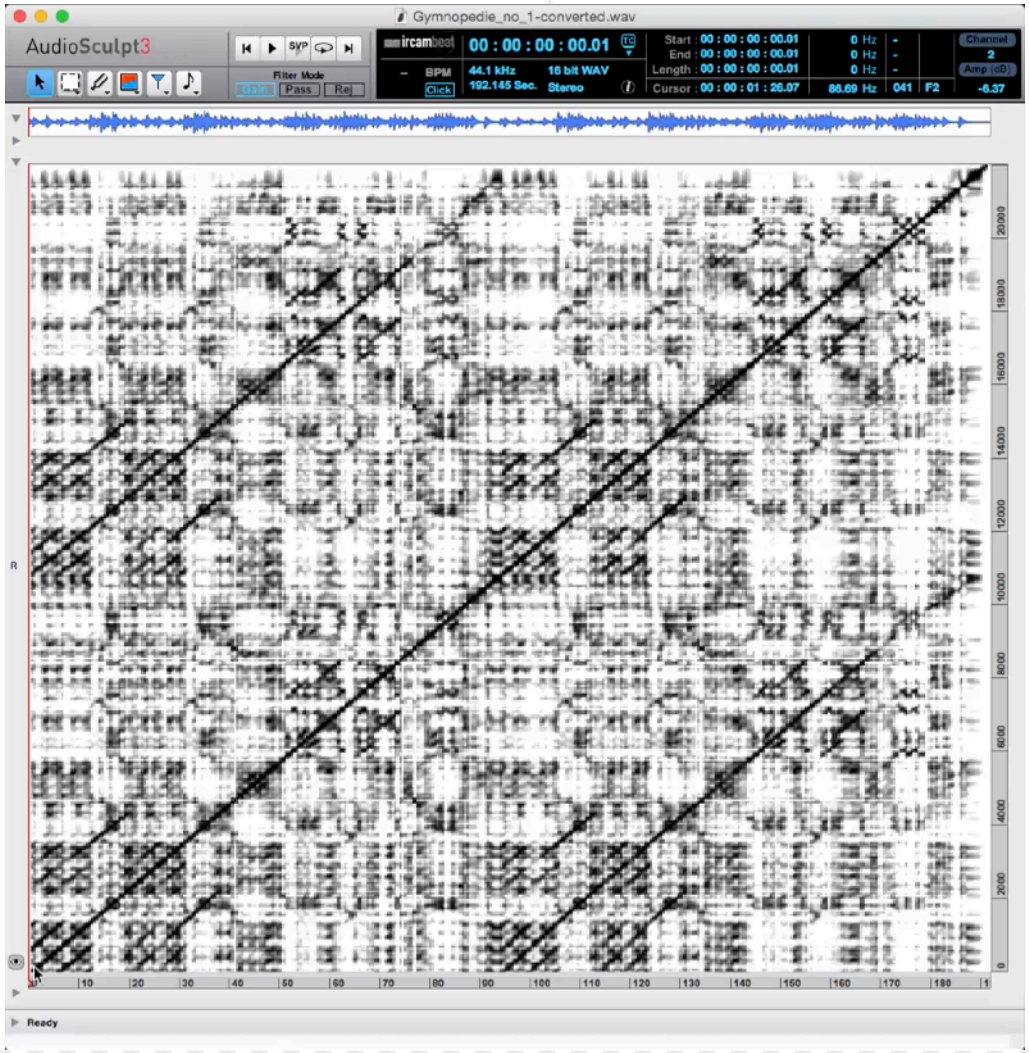
Music Structure Discovery (MSD) - Audio Summary

Example: Self-Similarity-Matrix



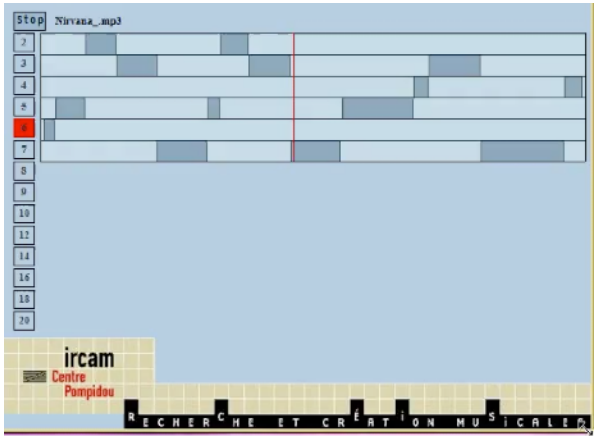
Music Structure Discovery (MSD) - Audio Summary

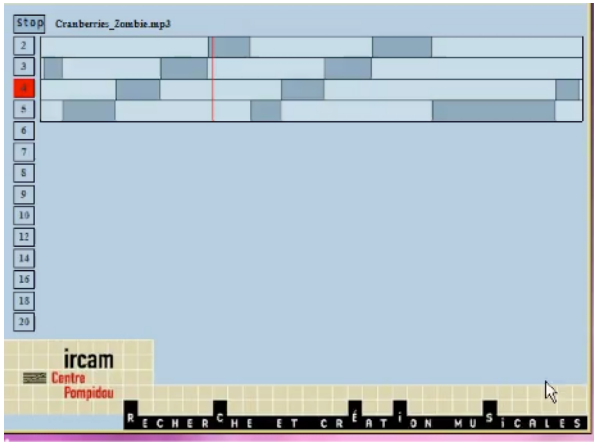
Example: Self-Similarity-Matrix

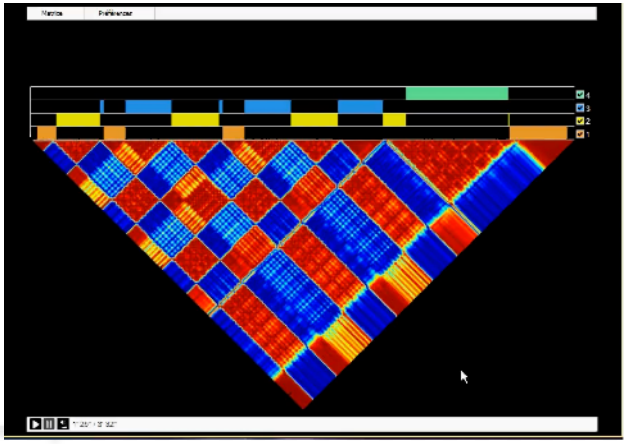


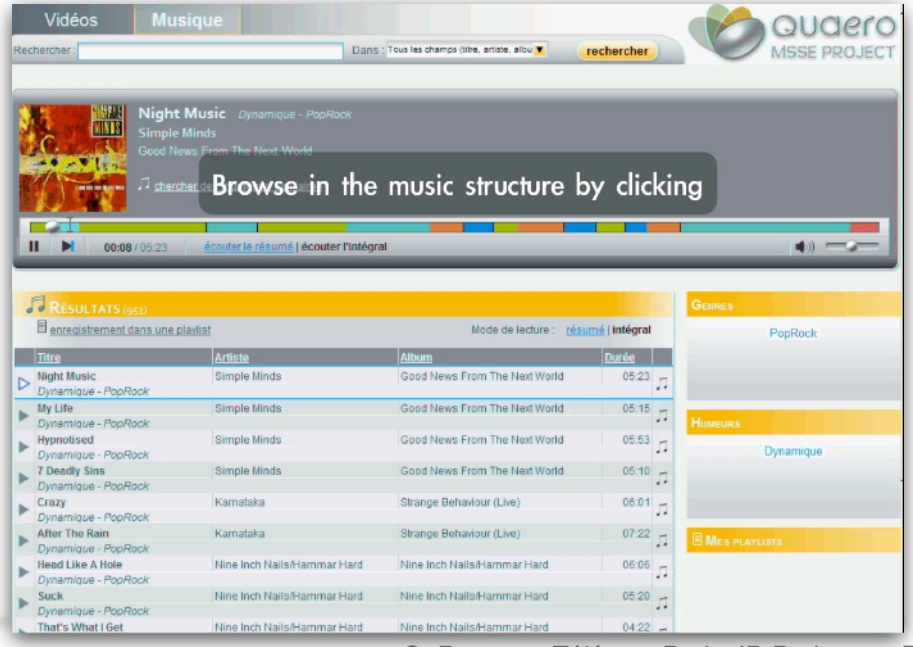
Music Structure Discovery (MSD) - Audio Summary

Example: Navigating into the structure









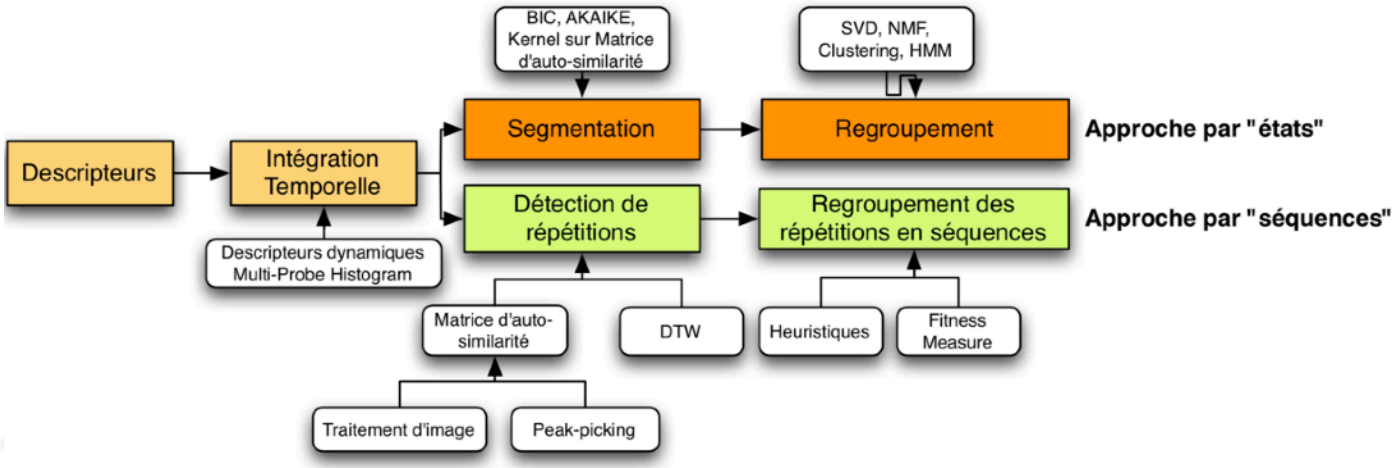
Music Structure Discovery (MSD) - Audio Summary

Example: Audio Summary

Systems for Music Structure Estimation

- 1) Extraction of meaningful observation from the audio signal
  - **Audio features**: allows to highlight different facets of the content (timbre, harmonicity, noise, ...)

- 2) Analyze the observations to estimate a structure
  - **State** approach
    - **segmenting** temporal stream of observation
    - **grouping** repeated homogeneous segments
  - **Sequence** approach
    - **detecting** non-homogeneous **repetitions**
    - **grouping** repeated segments into sequences



Music Structure Discovery (MSD) - Audio Summary Systems

Music Structure Discovery (MSD) - Audio Summary Systems

Brief overview of systems evolution

- *as usual, the **first systems** define the task, the performance measures, and provide a first test-set; **later systems** deals with scalability issues and create large test-set; **current systems** use this large dataset to train systems using deep-learning*

- (1) Self-Similarity-Matrix
  - **1999** → J. Foote. Visualizing music and audio using self-similarity. In Proc. of ACM Multimedia, 1999
- (2) Kernel-based segmentation
  - **2000** → J. Foote. Automatic audio segmentation using a measure of audio novelty. In Proc. of IEEE ICME, 2000
- (3) SSM-based audio summary generation
  - **2002** → M. Cooper and J. Foote. Automatic music summarization via similarity analysis. In Proc. of ISMIR, 2002.
- (4) Structure-based audio summary generation
  - **2002** → G. Peeters, A. Laburthe, and X. Rodet. Toward automatic music audio summary generation from signal analysis. In Proc. of ISMIR, 2002
- (5) DTW approach
  - Serra/Mueller/Groshe, Bisot/Peeters
- (6) Deep learning approaches
  - **2017** → A. Cohen-Hadria and G. Peeters. Music structure boundaries estimation using multiple self-similarity matrices as input depth of convolutional neural networks. In AES Conference on Semantic Audio, 2017.

Traditional machine-learning approach
(1) Audio features

Audio features

- Audio features ?
  - A numerical value extracted/estimated from the audio signal which the aim of highlighting a specific content of the audio signal
  - Why not using directly the waveform ? the STFT ?
    - much too high dimensional, much too complex to interpret

- Constraint
  - interpretability
  - low-dimensional
  - same number of dimension for all the data

- Computation ?
  - Mathematical formula
  - Estimation

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

Audio features

- Various **forms**:
  - **scalar**: spectral centroid, spectral spread, fundamental frequency, spectral roll-off, spectral flux, zero-crossing rate, RMS, ...
  - **vector**: Mel Frequency Cepstral Coefficients, coefficients LPC, coefficients PLP, ...

- Various **time validity**:
  - represent one **frame** of the audio signal → "instantaneous" feature
  - represent the content of a **set of local frames** →texture windows
  - represent **globally** the audio signal

- Highlight different facets of the audio **content**:
  - **timbre** content: Mel Frequency Cepstral Coefficients, LPC coefficients, PLP coefficients, ...
  - **harmonic** content: Pitch Class Profiles/ Chroma, ...
  - **noise** content: Spectral Flatness Measure, ...
  - **rhythmic** content: ...

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

Audio features examples

Zero-crossing rate (zcr)

- Measures the number of times the audio waveform cross the zero-axis
  - $$zcr = \frac{1}{N}\sum_{n=1}^{N} |sign(x_n) - sign(x_{n-1})|$$
- Usage: allows to distinguish
  - harmonic sounds → low zcr
  - noise sounds→ high zcr

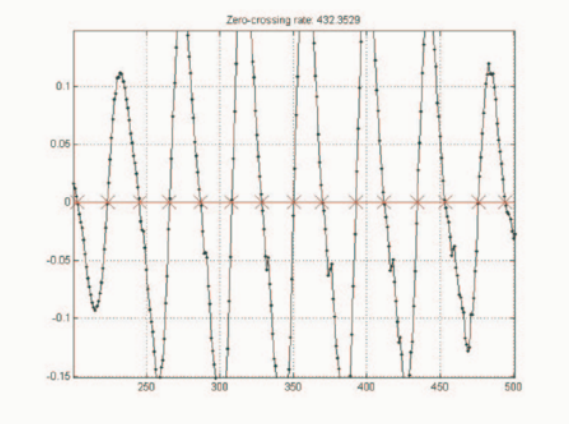


Figure 12 Zero-crossing rate (=432) during voiced speech region

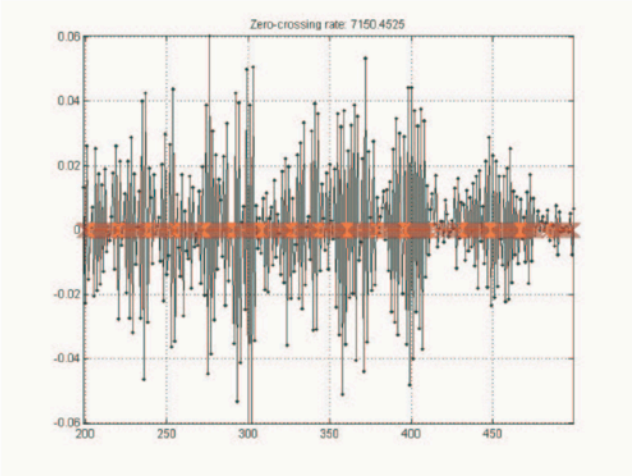


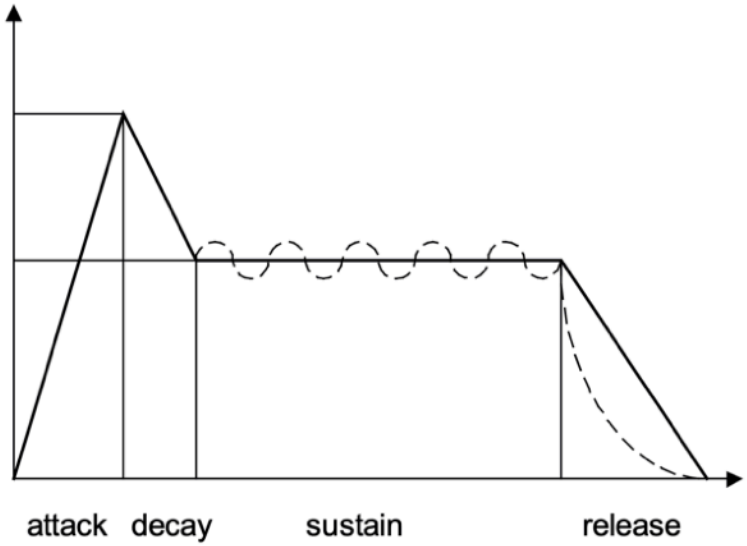
Figure 13 Zero-crossing rate (=7150) during unvoiced speech region

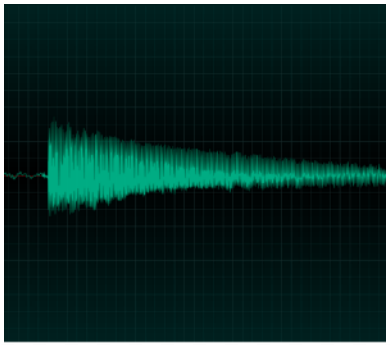
Audio features examples

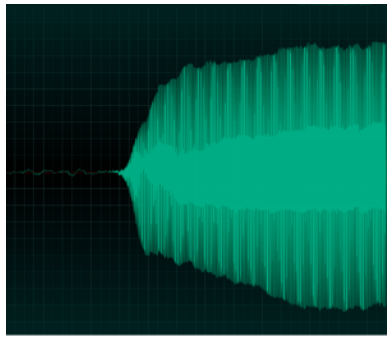
ADSR (Attack, Decay, Sustain, Release) temporal enveloppe

- Model of the temporal evolution (enveloppe) of the energy of a musical note

- Usage: allows to distinguish
  - fast attacks (percussive sounds) /slow attacks
  - fast decrease(non-sustained sounds) / slow decrease (sustained sounds)







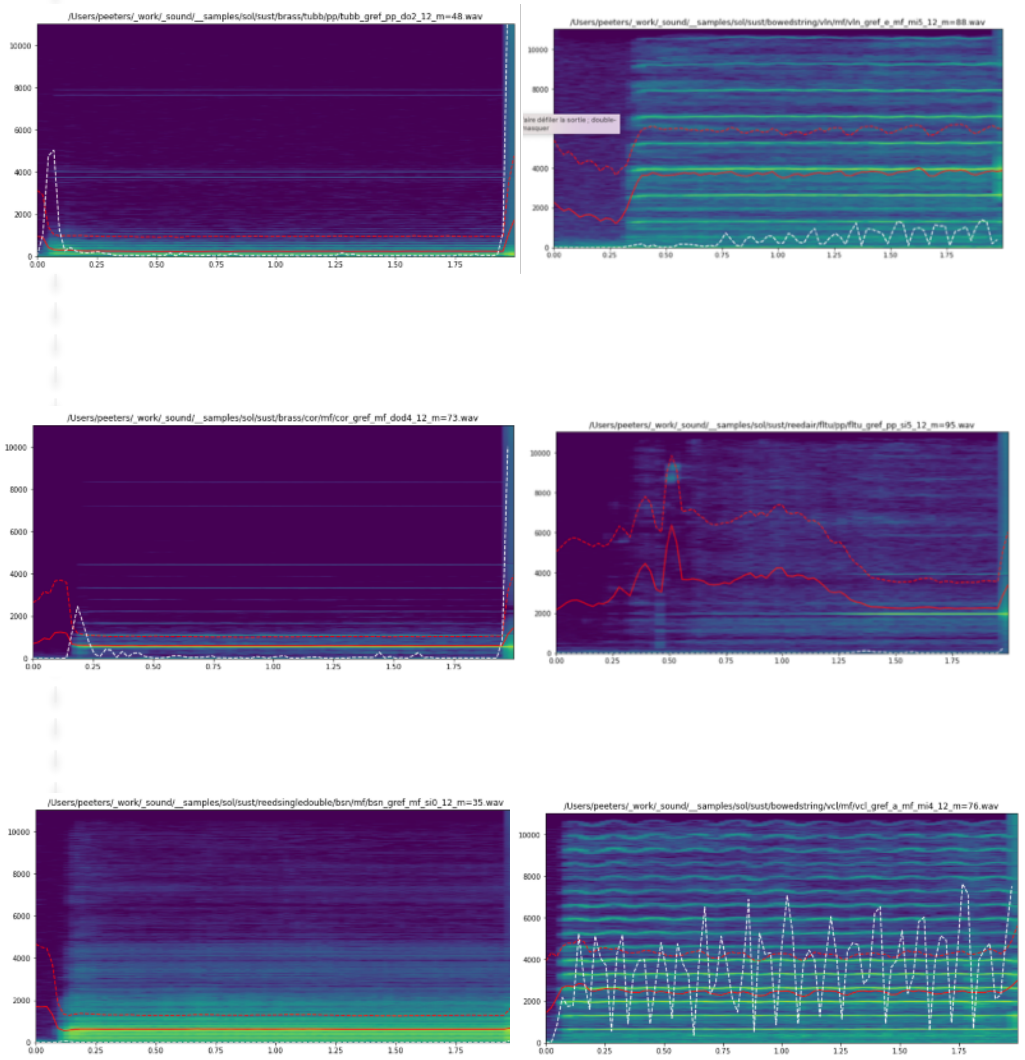
Audio features examples

Spectral shape description

- **Spectral centroid**
  - $$cs = \frac{\sum_k f_k A_k}{\sum_k A_k}$$
  - allows to distinguish between "dull" and "bright" sounds
- **Spectral spread**
  - $$es = \sqrt{\frac{\sum_k (f_k - cs)^2 A_k}{\sum_k A_k}}$$
  - allows to distinguish between "poor" and "rich" sounds
- **Spectral flux**
  - Measure the temporal variation of the spectrum

$$fs = \sum_k (A_k(t) - A_k(t-1))^2$$

  - allows to distinguish between "poor" and "rich" sounds



Mel Frequency Cepstral Coefficients (MFCCs)

Complex cepstrum

– **Goal**

- describe the shape of the spectrum (the timbre) of a signal using a reduced set of coefficients

– **Complex cepstrum** $c(\tau)$

$$c(\tau) = TF^{-1}\left[\log(X(\omega))\right]$$
$$= \frac{1}{2\pi}\int_{\omega} \log[X(\omega)] \cdot e^{j\omega\tau} d\omega$$

- τ is named "**quefrency**" (=frequency in reverse order)
- $x(t) \xrightarrow{TF} X(\omega) \xrightarrow{\log} \log(X(\omega)) \xrightarrow{TF^{-1}} c(\tau)$

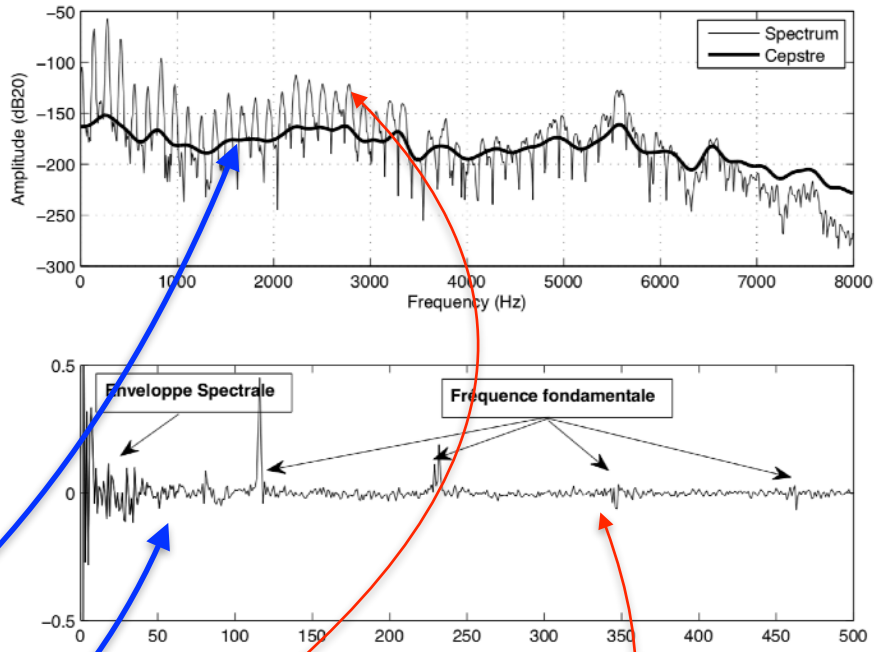
Audio features examples
Mel Frequency Cepstral Coefficients (2)

Source/filter model

- Source $e(t)$: periodic signal
- Filter $g(t)$: resonant/ anti-resonant filter

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

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



$$\xrightarrow{\log} \log(X(\omega)) = \underbrace{\log[G(\omega)]}_{\text{slow variations over } \omega} + \underbrace{\log[E(\omega)]}_{\text{fast variations over } \omega}$$

$$\xrightarrow{TF^{-1}} TF^{-1}\left[\log(X(\omega))\right] = \underbrace{TF^{-1}\left[\log[G(\omega)]\right]}_{\text{energy at quefrency } \tau<<} + \underbrace{TF^{-1}\left[\log[E(\omega)]\right]}_{\text{energy at quefrency } \tau>>}$$

Real cepstrum

– **Real ?** = cepstrum computed on the real part of the log-spectrum

$$X(\omega) = A(\omega) \cdot e^{j\phi(\omega)}$$
$$\log[X(\omega)] = \log[A(\omega)] + j\phi(\omega)$$
$$\Re\{\log[X(\omega)]\} = \log[A(\omega)]$$

$$\text{real cepstrum} = TF^{-1}\left[\Re\{\log[X(\omega)]\}\right]$$
$$= TF^{-1}\left[\log[A(\omega)]\right]$$
$$c(\tau) = \frac{1}{2\pi}\int_{\omega} \log[A(\omega)] \cdot e^{j\omega\tau} d\omega$$

– The amplitude spectrum $A(\omega)$ is real and symmetric
  - its Fourier Transform reduces to the real part
    – reduces to the projection of $\log[A(\omega)]$ on a set of cosinus → Discrete Cosine Transform (DCT)

Audio features examples
Mel Frequency Cepstral Coefficients (4)

Mel Frequency Cepstral Coefficients (MFCCs)

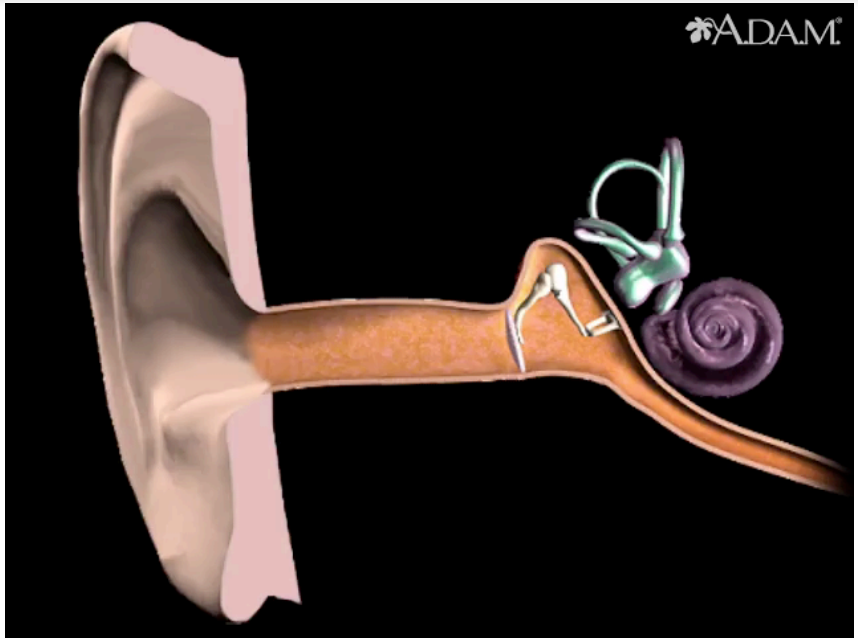
- MFCC ? = real cepstrum computed on the power spectrum $|X(\omega)|^2$ converted to the Mel scale (a perceptual scale)

- **Why perceptual scales ?**
  - Fourier Transform
    - decomposition on a set of sinusoidal components which frequencies are linearly spaced (f_k = 10Hz, 20Hz, 30Hz, ... Hz)
  - Human hearing:
    - decomposition on a set of filters which frequencies are logarithmically spaced (10, 20, 40, 80, ... Hz).
    - highest resolution in low-frequencies, lowest resolution in high frequencies
    - in speech, formants/resonances are closer together in low frequencies
  - MFCCs allows a more compact representation than the real cepstrum
- **How ?**
  - Use of perceptual scales: Mel-scale, Bark-scale, ERB-filters, Gamma-tone filters
- **Usage ?**
  - MFCCs are the most used features in audio: speech, music, environmental sounds recognition, ...

Audio features examples
Mel Frequency Cepstral Coefficients (5)

Human hearing

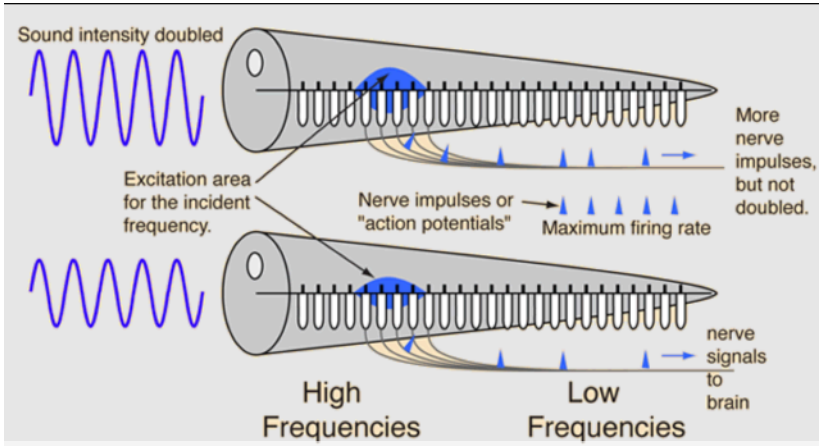
- Cochlea



https://medlineplus.gov/ency/anatomyvideos/000063.htm

- Critical bands
  - perception of two tones at f_1 and f_2
  - perception of a beating-tone at $\frac{f_1+f_2}{2}$

$$\cos f_1 + \cos f_2 = 2\cos\frac{f_1-f_2}{2}\cos\frac{f_1+f_2}{2}$$



source: http://hyperphysics.phy-astr.gsu.edu/hbase/Sound/loud.html

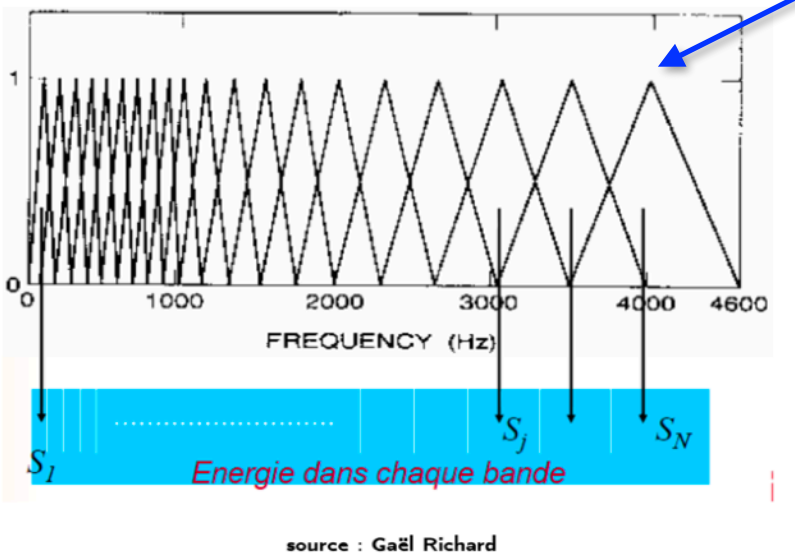
Audio features examples
Mel Frequency Cepstral Coefficients (6)

Mel scale ?

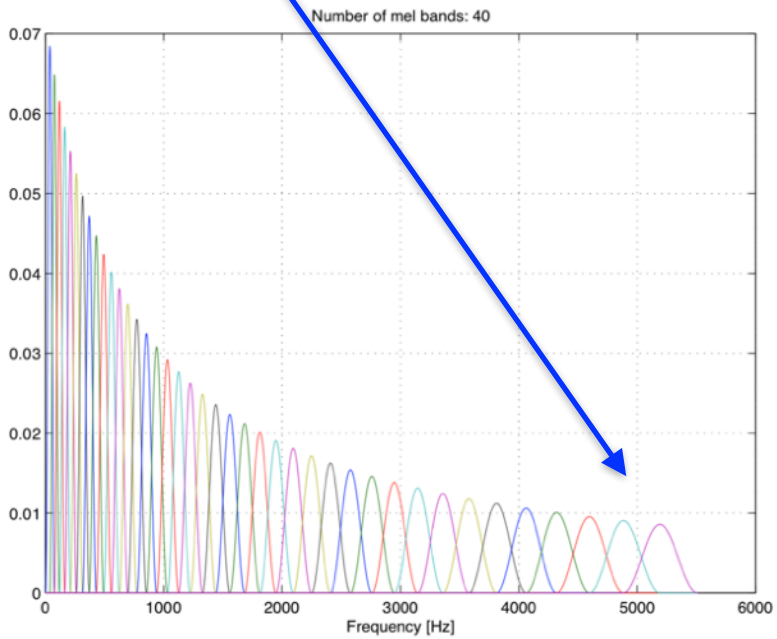
$$mel(f) = \frac{1000}{\ln 2} \ln\left(1 + \frac{f}{1000}\right)$$

- Remark: variations of the constant exist

different shapes for the filter: triangular, hanning, tanh



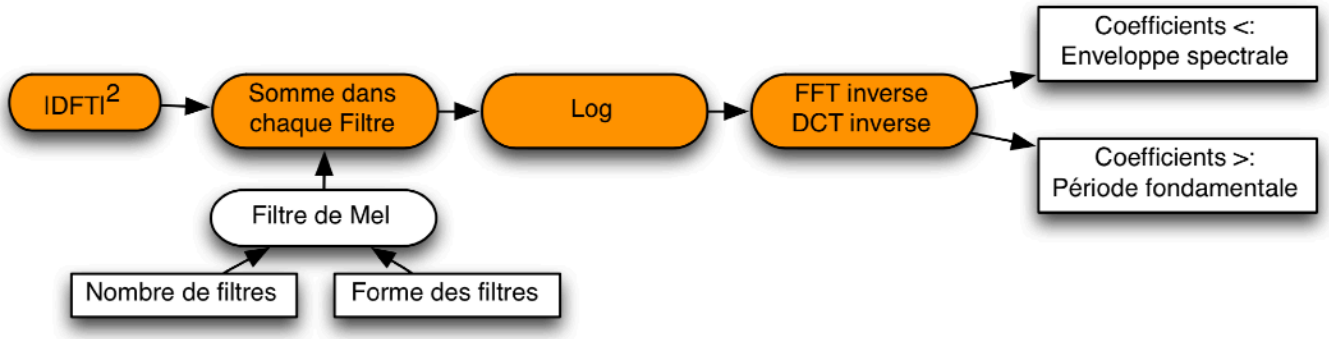
source : Gaël Richard



Fant, Gunnar. (1968). *Analysis and synthesis of speech processes.* In B. Malmberg (Ed.), *Manual of phonetics* (pp. 173-177). Amsterdam: North-Holland.

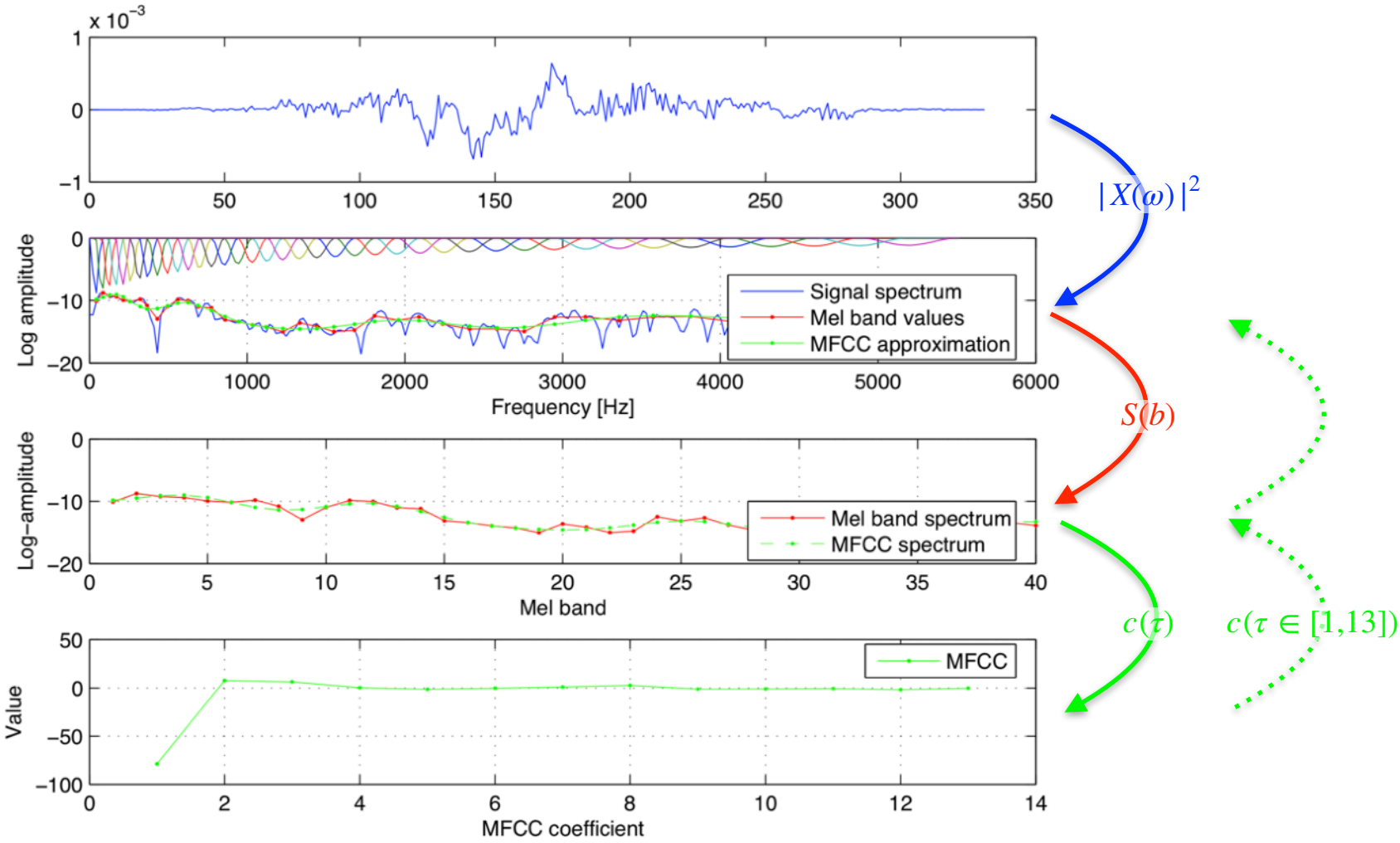
Computation steps for MFCCs

- Compute the power spectrum: $|X(\omega)|^2$
- Compute the Mel filters: $H_b(\omega)$ with $b \in [1,B]$
  - choice of the number of filters B: 40
  - choice of the shape of each filter: triangular, hanning, tanh, ...
- Convert the power spectrum to Mel bands: $S(b) = \sum_{\omega} |X(\omega)|^2 \cdot H_b(\omega)$
- Convert to logarithmic scale: $\log(S(b))$
- Compute the IFFT (or the IDCT): $c(\tau)$
- Select the first coefficients, close to 0 (usually the first 13 coefficients)
  - coefficients close to zero represent the decomposition of the Mel bands content on a set of cosinus with slow variations



Audio features examples
Mel Frequency Cepstral Coefficients (8)

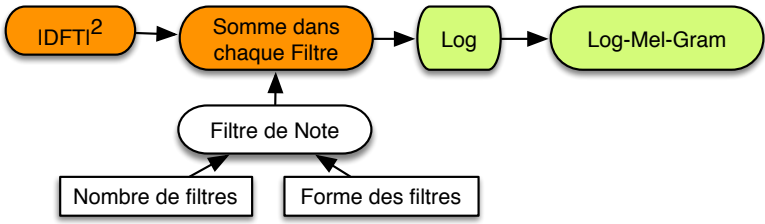
Example of the computation of MFCCs

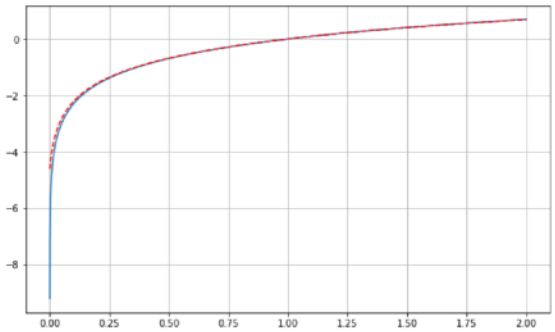


Audio features examples
Mel Frequency Cepstral Coefficients (9)

Variation for the case of DNN inputs: the Log-Mel-gram

- In the **Cepstrum**
  - the DCT is used to separate the contribution of the source and the filter
- In the **Mel Frequency Cepstral Coefficients**
  - the Mel-bands already mostly represent the filter contribution
  - the DCT is mostly used to de-correlated the dimensions
    - (latter used in GMM:diagonal covariance matrix Σ)
- **Log-Mel-Gram**
  - When using DNN, we don't need such a de-correlation of the inputs
  - we then bypass the DCT of the MFCC → Log-Mel-Gram
  - Tricks: to avoid singularity of $\log(x)$ → replace log by $\log(1+\gamma x)$ with $\gamma = 100$

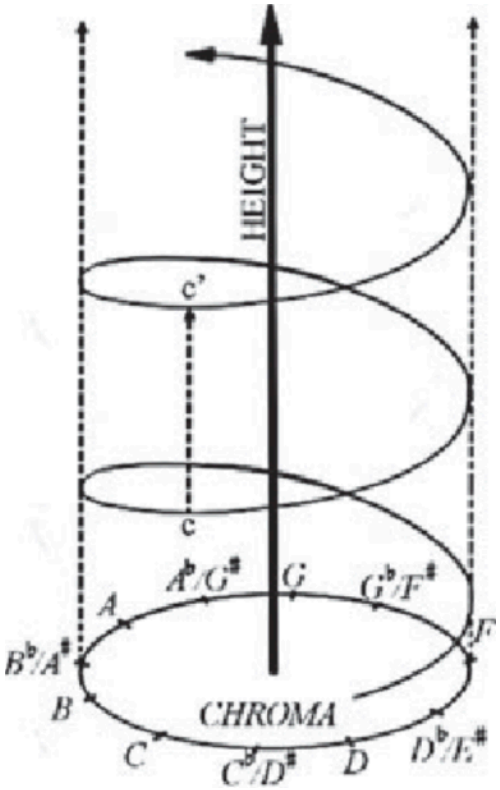




Chroma/ Pitch Class Profile (PCP)

Chroma/ Pitch Class Profile (PCP)

- **Helical model of pitch [Roger Shepard, 1964]**
  - represents the pitch of a note p as a two-dimensional structure:
  - $p = c + o \cdot 12$
    - chroma c (pitch-class)
    - tonal height o (octave number)

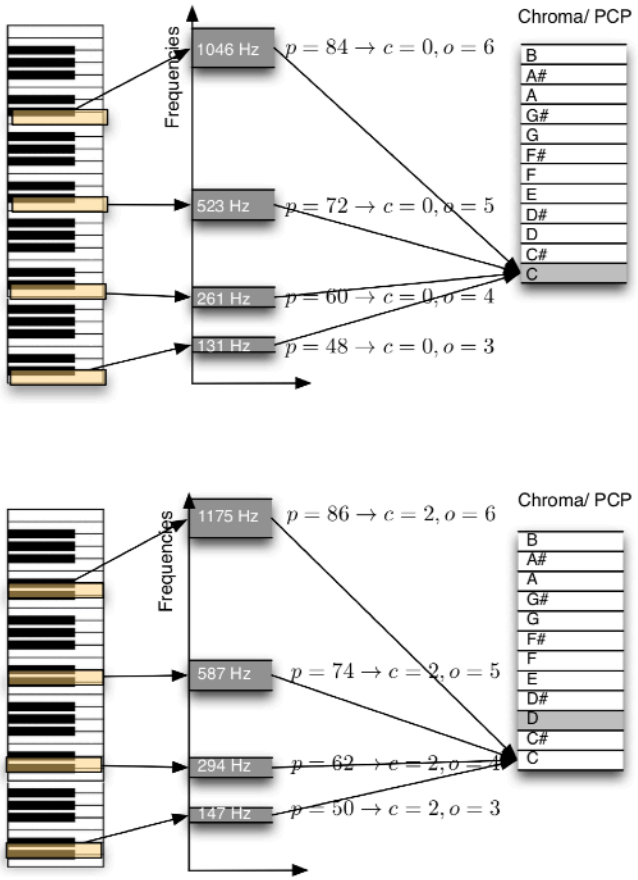


- **Definition: Chroma - Pitch Class Profile (PCP):**
  - represents the harmonic content of the spectrum at time n, $X(k,n)$, as a vector:
    - $C(c,n)$ $c \in [0,12[$

- **Usage:**
  - key estimation,
  - chord estimation,
  - cover detection

Chroma/ Pitch Class Profile (PCP)

- **Chroma computation** $C(c,n)$
  - We sum up the values of the spectrum $X(k,n)$ for all f_k which correspond to a given c
  - Relationship between the frequencies f_k of the DFT and the pitches p (semi-tone pitches in MIDI-scale)
    - $p(f_k) = 12\log_2\left(\frac{f_k}{440}\right) + 69,\ p \in \mathbb{R}^+$
    - $f(p) = 440 \cdot 2^{\frac{p-69}{12}}$



T. Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. In Proc. of ICMC (International Computer Music Conference), pages 464–467, Bejing, China, 1999.

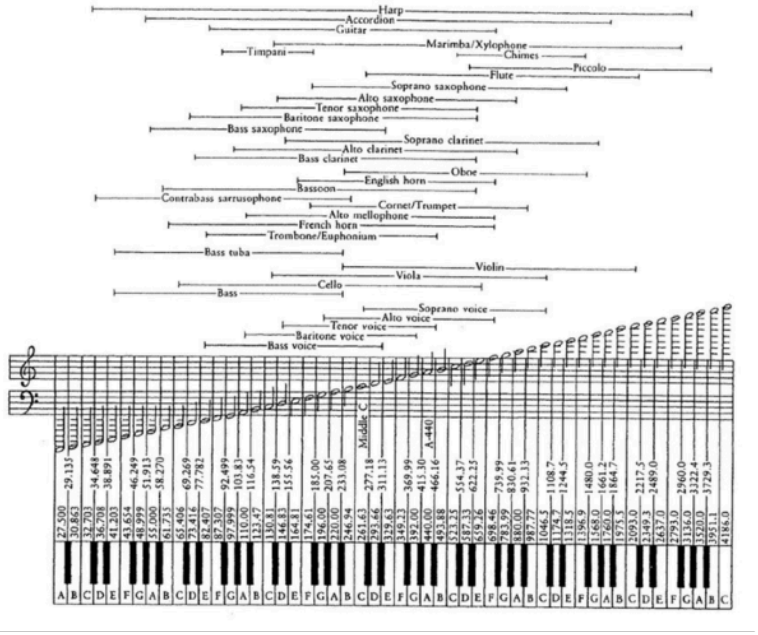
G. H. Wakefield. Mathematical representation of joint time-chroma distributions. In Proc. of SPIE conference on Advanced Signal Processing Algorithms, Architecture and Implementations, pages 637– 645, Denver, Colorado, USA, 1999.

Chroma/ Pitch Class Profile (PCP)

- **Spectral resolution ?**
  - Should allow separating adjacent musical notes
  - We define the width (at -6 dB) as $Bw = \frac{Cw}{L_{sec}}$
  - If $f_{\min}$ (the lowest frequency we consider in the spectrum) is 50 Hz
    - We need to separate $G\#1$ (51.91Hz) from $A1$ (55Hz)

      $$\rightarrow L_{sec} = \frac{Cw}{Bw} = \frac{2.35}{3.0869Hz} = 0.7613s$$
  - If $f_{\min}$ is 100 Hz
    - We need to separate $G\#2$ (103.82Hz) from $A2$ (110Hz)

      $$\rightarrow L_{sec} = \frac{Cw}{Bw} = \frac{2.35}{6.1738Hz} = 0.3806s$$



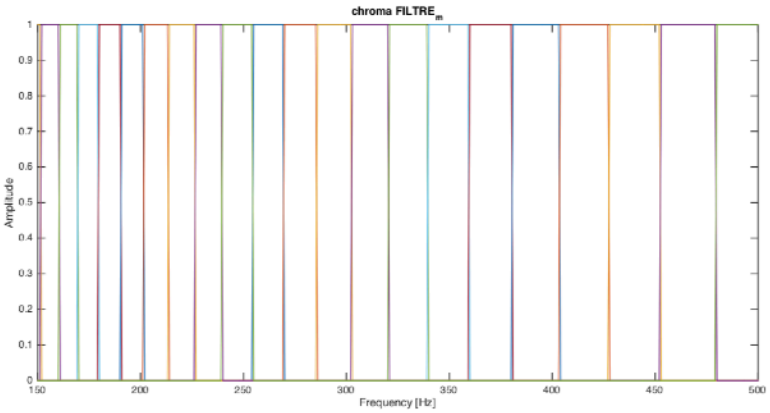
- Two possibilities:
  - Choice L_{sec} as a function of $f_{\min}$
  - Choose $f_{\min}$ as a function of L_{sec}

Chroma/ Pitch Class Profile (PCP)

- **Chroma computation** $C(c, n)$
  - We sum up the values of the spectrum $X(k, n)$ for all f_k which correspond to a given c
  - 1) Hard-mapping
  - 2) Soft-mapping

Chroma/ Pitch Class Profile (PCP)

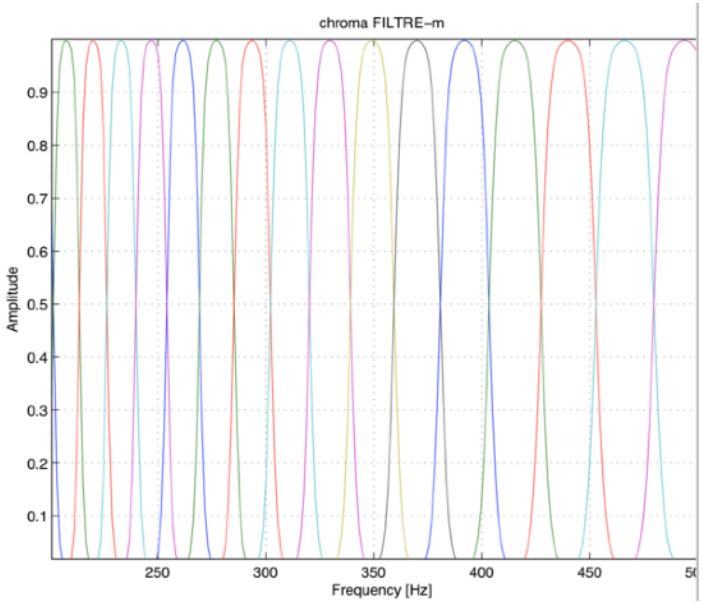
- **1) Hard-mapping ?**
  - A frequency f_k of the DFT only contributes to the closest pitch
  - Example:
    - the energy at $f_k = 452\ Hz\ (p(f_k) = 69.4658)$ only contributes to the pith p=69 (c=10)
    - while $f_k = 453\ Hz\ (p(f_k) = 69.5041)$ to p=70 (c=11).
  - Creation of bank of filters $H_{p'}$ centered on the semi-tone pitches $p' \in \{43,44,\ldots,95\}$:



Chroma/ Pitch Class Profile (PCP)

- **2) Soft-mapping ?**

  - A frequency f_k of the DFT contribues to different chromas with a weight inversely proportional to the distance between $p(f_k)$ and the p the closest
  - Example:
    - the energy at $f_k = 452\ Hz\ (p(f_k) = 69.4658)$ contributes nearly equally to p=69 (c=10) and p=70 (c=11).
  - Creation of bank of filters $H_{p'}$ centered on the semi-tone pitches $p' \in \{43,44,\ldots,95\}$:
    - Each filter is defined by the function
      - $H_{p'}(f_k) = \frac{1}{2}\tanh\left(\pi(1-2x)\right) + \frac{1}{2}$
    - where
      - x = relative distance between the center of the filter p' and the frequencies of the DFT $p(f_k)$
      - $x = R\left|p' - p(f_k)\right|$
    - The filters are evenly distributed and symetrical on the logarithmic scale of semi-tone pitches, non-zero between $p'-1$ and $p'+1$ with a maximum value at p'

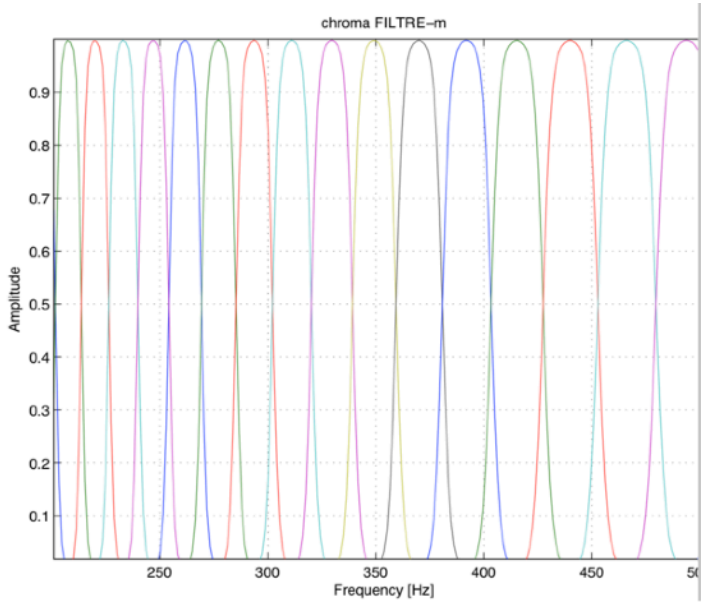


Chroma/ Pitch Class Profile (PCP)

- **2) Soft-mapping (cont.)**
  - The value of the semi-tone pitch spectrum $N(n')$ is given by multiplying the values of the DFT $A(f_k)$ with the bank of filters $H_{n'}$:

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

  - The mapping between semi-tone pitches n and the pitch-classes (chroma) c is defined by:
    - $c(p) = mod\,(p,12)$
  - The value of the chroma is obtained by summing up the values of equivalent semi-tone pitches
    - $C(c) = \sum_{p' \text{ tel que } c(p')=l} P(n') \quad c \in [0,12[$



Chroma/ Pitch Class Profile (PCP)

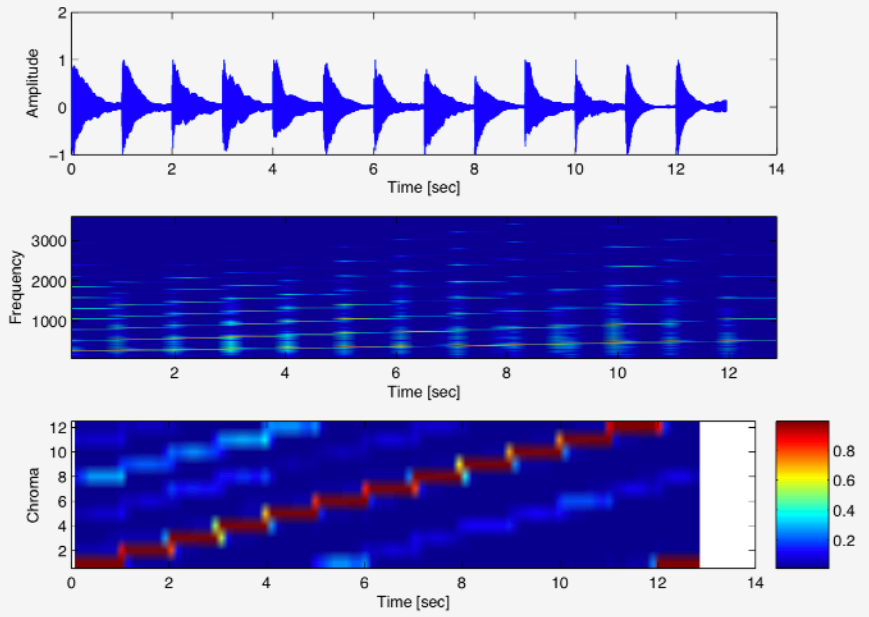
- **Limitations of Chromas - Pitch Class Profile (PCP)**
  - Presence of the upper harmonics of each note
    - In practice, for a given note C we don't have $[1,0,0,0,0,0,0,0,0,0,0,0]$
    - but rather $[a_1 + a_2 + a_4,0,0,0,a_5,0,0,a_4,0,0,0,0]$
    - Influence of the spectral envelope

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)
	...	...	...	...

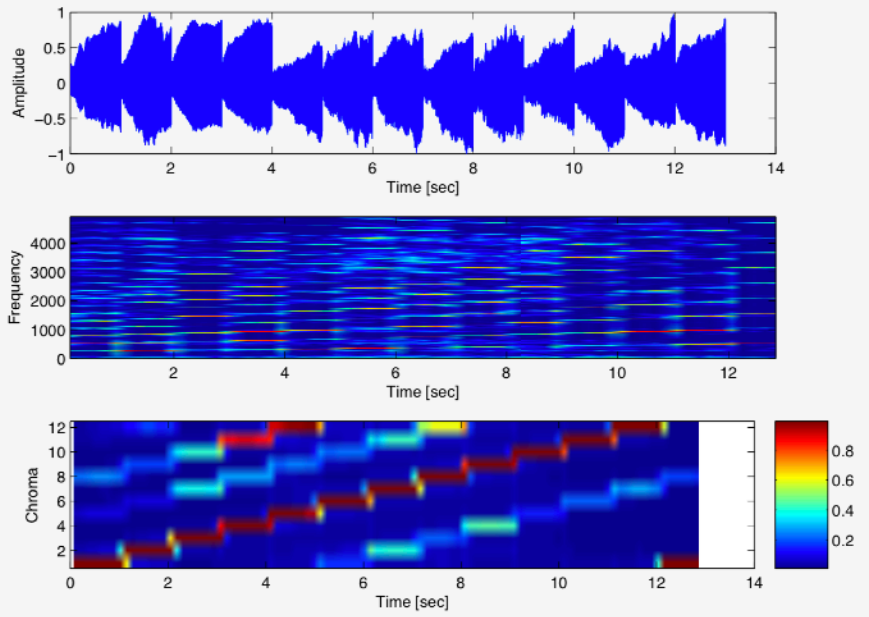
Chroma/ Pitch Class Profile (PCP)

- **Limitations of Chromas - Pitch Class Profile (PCP)**

Exemple piano

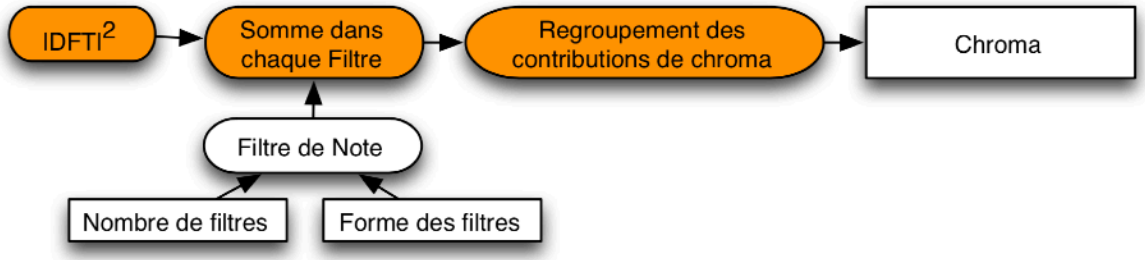


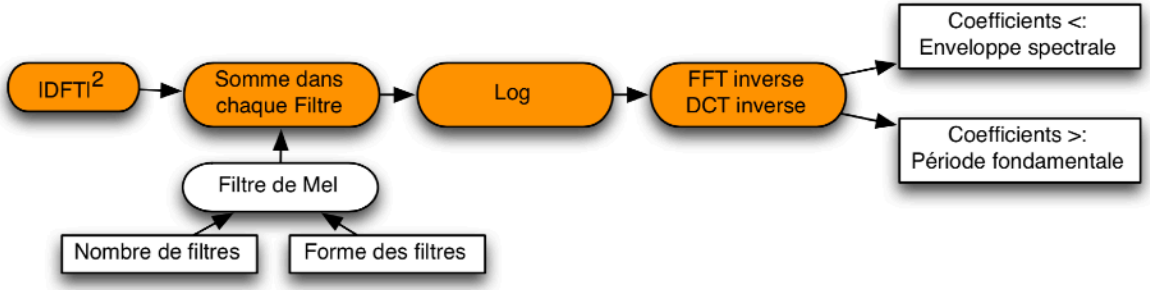
Exemple violon

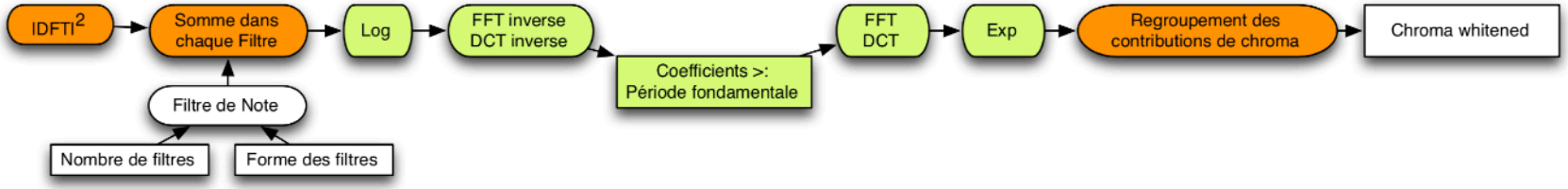


Chroma/ Pitch Class Profile (PCP)

- **Variation of the computation: whitening**



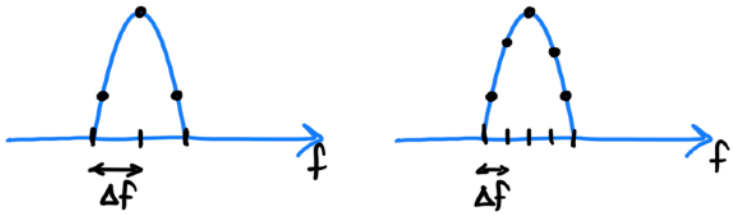




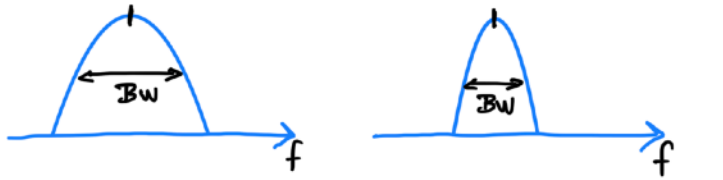
Transformée à Q constant

Constant-Q-Transform (CQT)

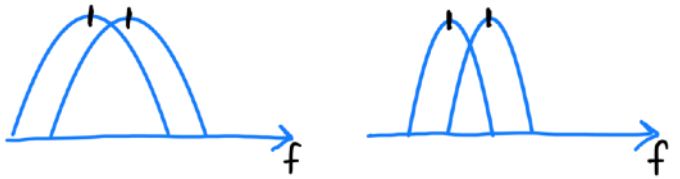
- Discrete Fourier Transform (DFT)
  - _ Definition : Spectral **precision** : $\Delta f = \frac{sr}{N}$
    - it is the step-size at which the Fourier spectrum is sampled
    - it depends on the size of the DFT: N
    - we can improve the precision by increasing N



  - _ Definition : Spectral **resolution** : $B_w = \frac{C_w}{L}$
    - it describes the ability to discriminate (separate in the spectrum) two adjacent simultaneous frequencies



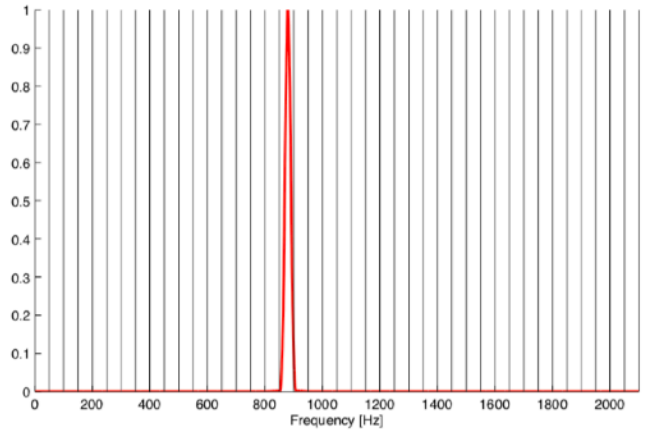
  - – Warning :
    - even if we increase N (zero-padding) while keeping L constant will not improve the resolution !



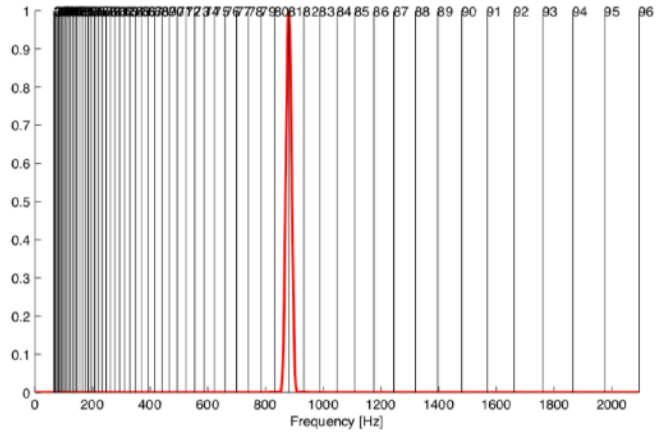
- In a DFT:
  - – Spectral precision and resolution are constant over frequencies

Constant-Q-Transform (CQT)

- In musical audio
  - the frequencies of the pitches are logarithmically spaced $f_k = f_0 \cdot 2^{\frac{k}{12}}$
    - if we choose A-4 = la-3 = 440 Hz as the reference
    - to go from midi-pitches m_k to frequencies f_k:
      - $f_k = 440 \cdot 2^{\frac{m_k - 69}{12}}$
    - to go from frequencies f_k to midi-pitches m_k:
      - $m_k = 12 \log_2 \frac{f_k}{440} + 69$
  - pitch frequencies are
    - close together in low frequencies,
    - distant in high frequencies
  - The **spectral resolution** of the DFT
    - is not sufficient (to separate adjacent notes) in low frequencies
    - is too large for high frequencies



Espacement linéaire de la DFT



Espacement logarithmique des hauteurs de notes

Constant-Q-Transform (CQT)

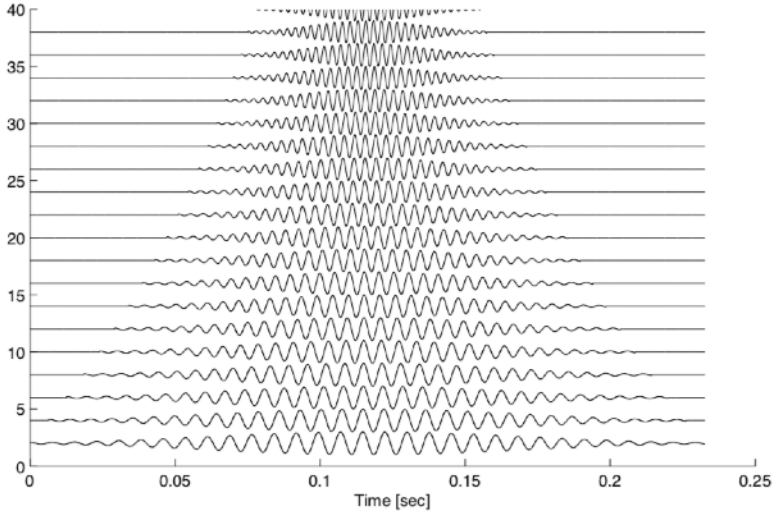
- Solution ?
  - Change the spectral resolution B_w depending on the frequency f_k being considered

- How ?
  - By changing the window length L for eadh frequency f_k
  - The factor $Q = \frac{f_k}{f_{k+1} - f_k}$ should remains constant in frequency
    - $Q = \frac{f_k}{Bw} = \frac{f_k}{Cw/L} = \frac{f_k \cdot L}{Cw}$
  - We choose a different L for each frequency f_k
    - $L_k = \frac{Q \cdot Cw}{f_k}$


An efficient algorithm for the calculation of a constant Q transform

Judith C. Brown

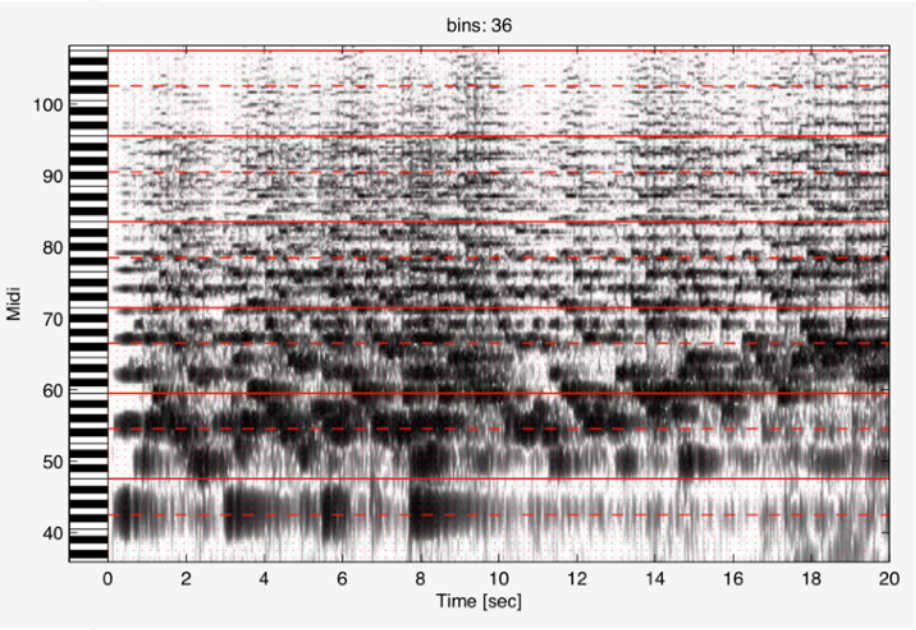
Miller S. Puckette



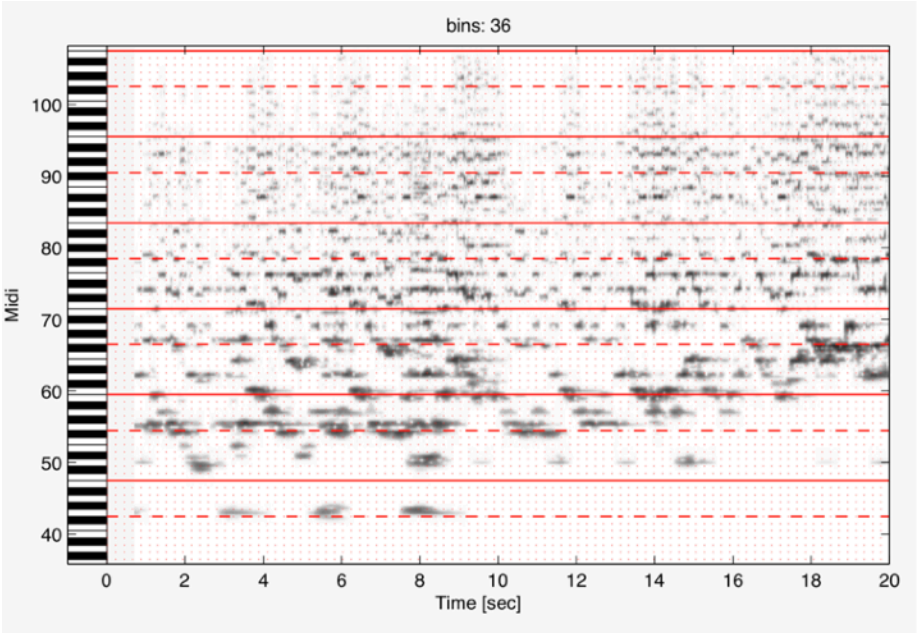
[J. Brown and M. Puckette. An efficient algorithm for the calculation of a constant q transform. JASA, 1992.]

Constant-Q-Transform (CQT)

Example (using DFT)

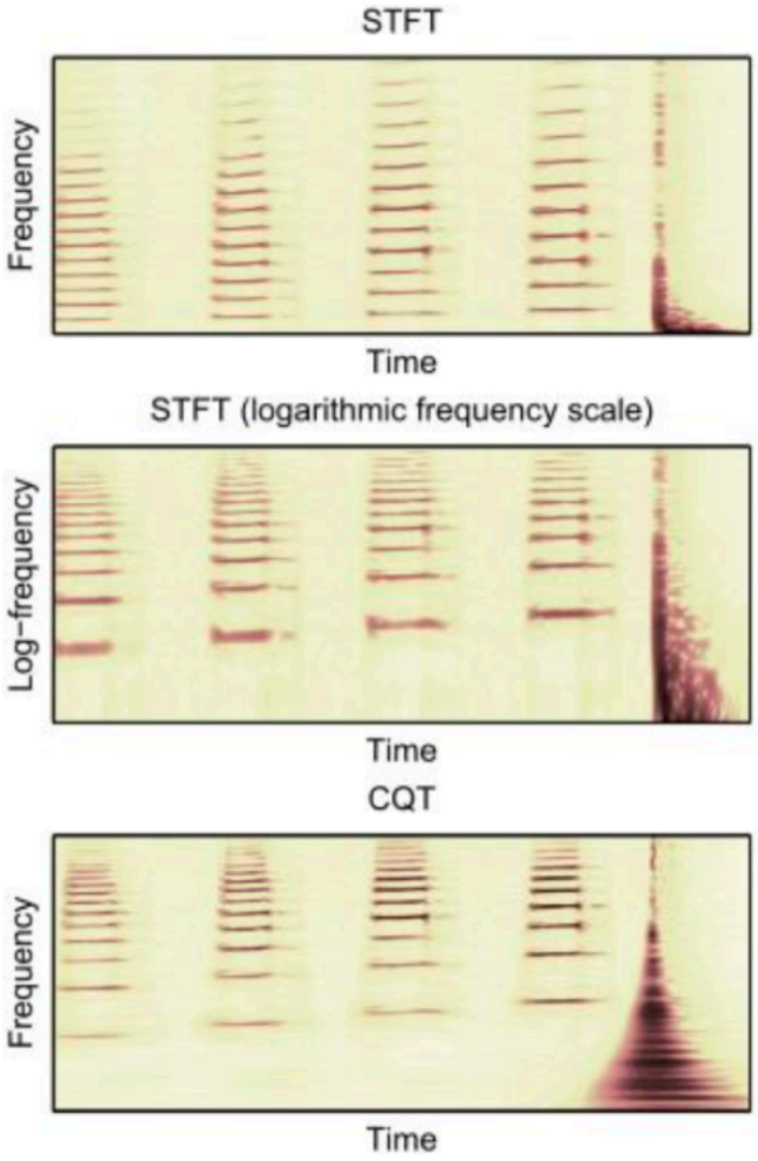


Example (using the CQT)



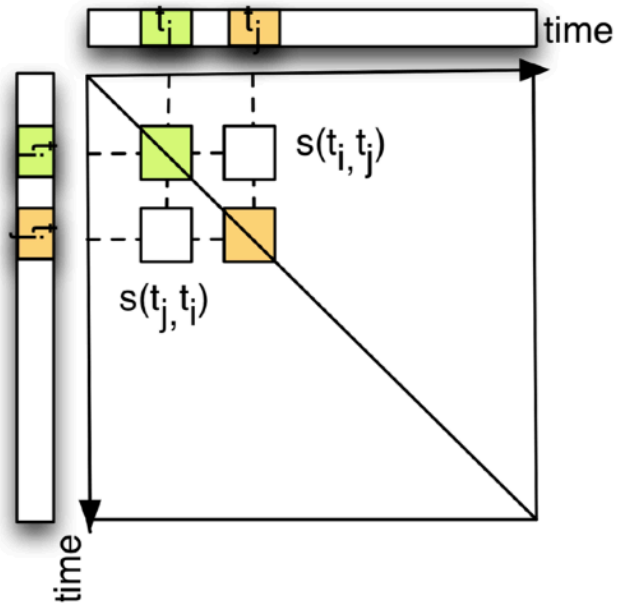
Constant-Q-Transform (CQT)

- In the Constant-Q-Transform (CQT):
  - A pitch difference corresponds to a translation over the (log) frequency axis



(1) Self-Similarity-Matrix (time, time)

- Visual representation of the temporal structure of a music track
- Indicates the similarity between two times t_i et t_j
- Similarity is computed by using the observations extracted from the signal around time frames i and j: $\mathbf{d}^{<i>}$ and $\mathbf{d}^{<j>}$
  - $s(t_i, t_j) = s(\mathbf{d}^{<i>}, \mathbf{d}^{<j>})$
- Self-Similarity-Matrix= values $s(t_i, t_j)$ represented as a matrix
  - $\mathbf{S}_{ij} = s(t_i, t_j) \quad \forall i, j$



- **How to read ? Interpretation ?**
  - High value in $\mathbf{S}_{ij}$ = high similarity between times t_i and t_j
  - If $t_i \simeq t_{i+1} \simeq t_{i+2}$, we observe an **homogeneous block** in $\mathbf{S}$
  - If the **sequence of times** $\{t_i, t_{i+1}, t_{i+2}, \ldots\}$ is similar to the sequence of times $\{t_j, t_{j+1}, t_{j+2}, \ldots\}$, we observe a lower/upper diagonal (symmetry) in $\mathbf{S}$

Music Structure Discovery (MSD) - Audio Summary Systems

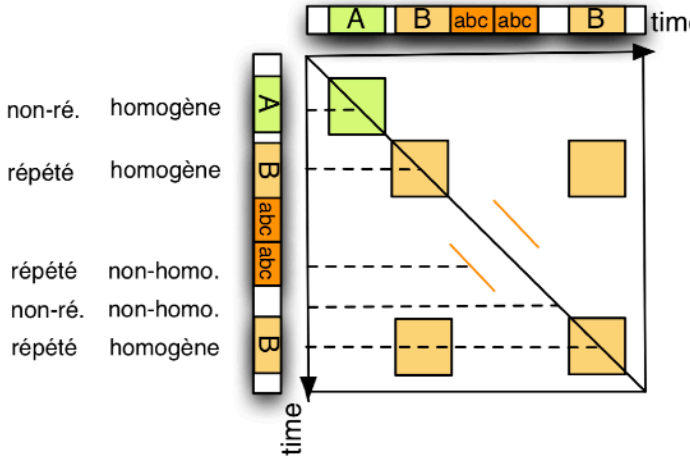
Homogeneity

– **Assumption:**
  - the music track is made of a succession of **homogeneous** $t_i \simeq t_{i+1} \simeq t_{i+2}, \ldots$ and non-homogeneous time segments

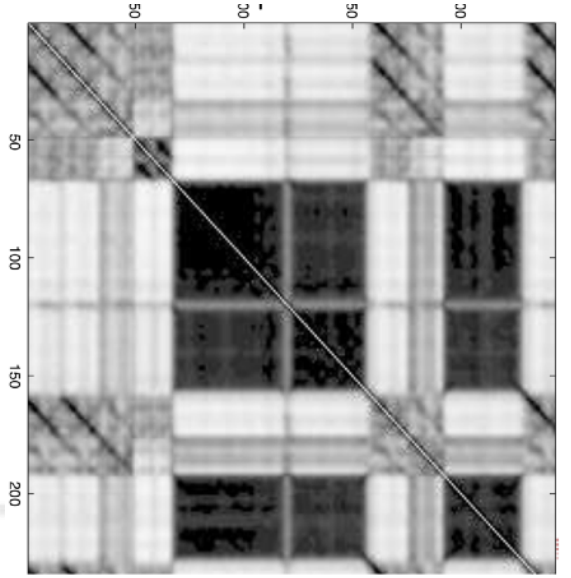
– **Homogeneous ?**
  - which contains a similar information according to the observation criteria
  - "A"' and "B"' in the Figure

– Example:
  - music accompaniment during a verse or a chorus

– Method:
  - **"state" approach**



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



Music Structure Discovery (MSD) - Audio Summary Systems

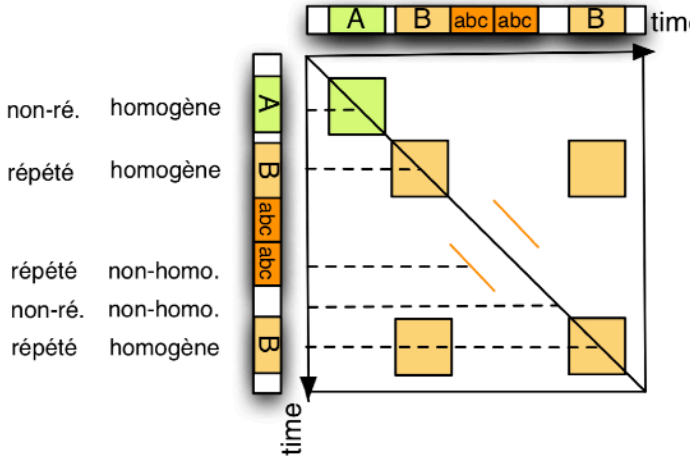
Repetitions

– **Assumption:**

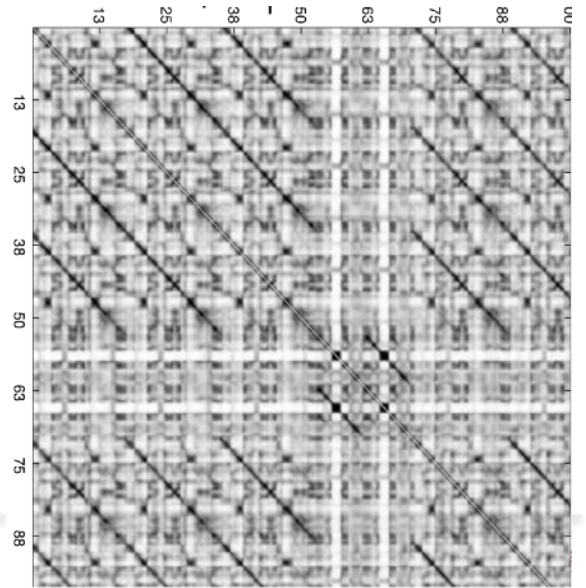
- the music track contains le morceau renferme des **répétitions** temporelles repetitions

– **Repetition ?**

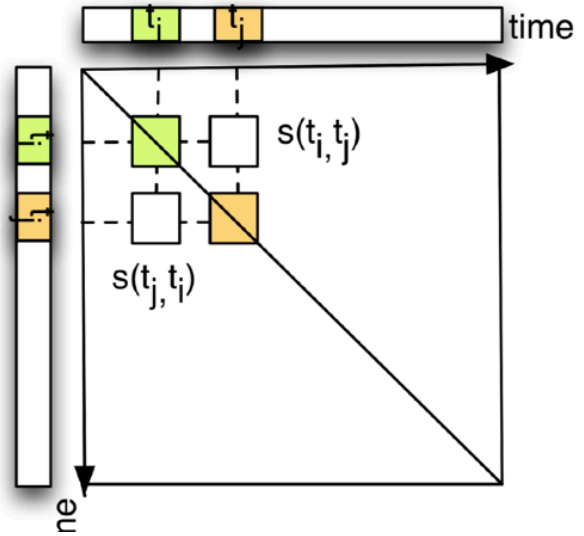
- can correspond to the repetition of **homogeneous** segments
  - $\{t_j, t_{j+1}, t_{j+2}\} \simeq \{t_i, t_{i+1}, t_{i+2}\}$ and $t_i \simeq t_{i+1} \simeq t_{i+2}$
  - "B" in the Figure
  - Method: **"state" approach**
- can correspond to the repetition of **non-homogeneous** segments
  - $\{t_j, t_{j+1}, t_{j+2}\} \simeq \{t_i, t_{i+1}, t_{i+2}\}$ but $t_i \neq t_{i+1} \neq t_{i+2}$
  - sequence "abc" in the Figure
  - Method: **"sequence" approach**



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



(1) Self-Similarity-Matrix (time, time)



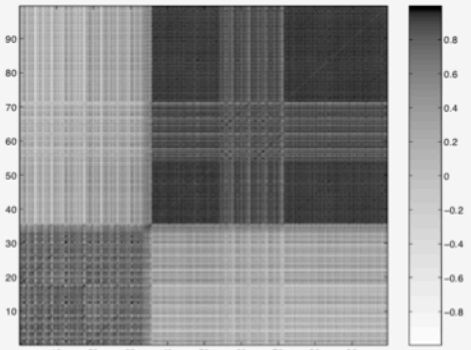
- Similarity between time t_i and t_j
- Similarity between the signal observations at time frame i and j:
  - $\mathbf{S}_{ij} = s(\mathbf{d}^{<i>}, \mathbf{d}^{<j>})$
- Audio features (multi-dimensional)
  - $\mathbf{d} = \{d_k\}\ k \in K$

– Choice of a distance

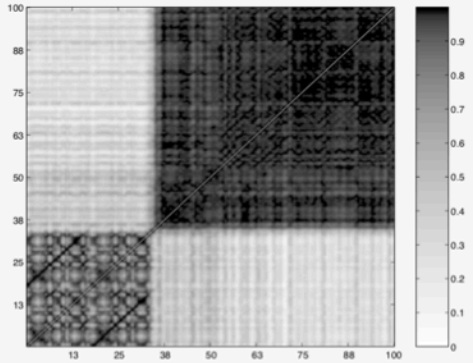
- Euclidean distance: $\sqrt{\sum_k (d_k^{<i>} - d_k^{<j>})^2}$
- Correlation: $\sum_k (d_k^{<i>} \cdot d_k^{<j>})$
- Cosine distance: $\frac{\sum_k (d_k^{<i>} \cdot d_k^{<j>})}{\sqrt{\sum_k (d_k^{<i>})^2}\sqrt{\sum_k (d_k^{<j>})^2}}$
- Pearson correlation: $\frac{\sum_k (d_k^i - \mu^i) \cdot (d_k^j - \mu^j)}{\sqrt{\sum_k (d_k^i - \mu^i)^2}\sqrt{\sum_k (d_k^j - \mu^j)^2}}$

Music Structure Discovery (MSD) - Audio Summary Systems

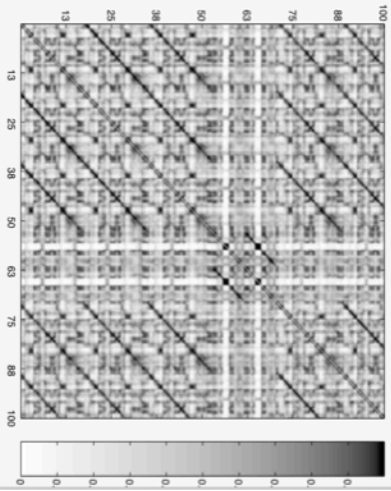
MFCC



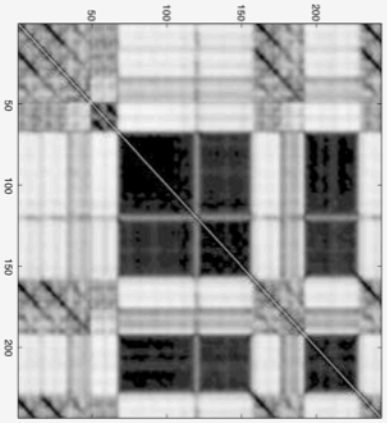
Modulation spectrum 1



Chroma



Modulation spectrum 2

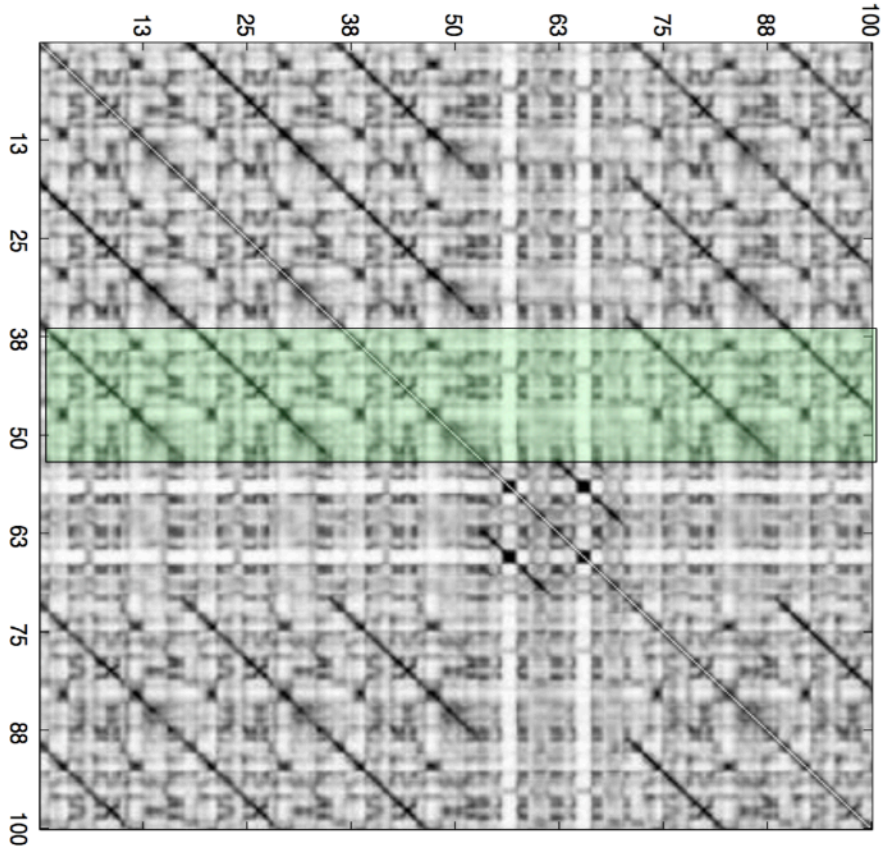


Geoffroy Peeters - LTCI, Télécom Paris, IP Paris - 42

Music Structure Discovery (MSD) - Audio Summary Systems

(3) SSM-based audio summary generation

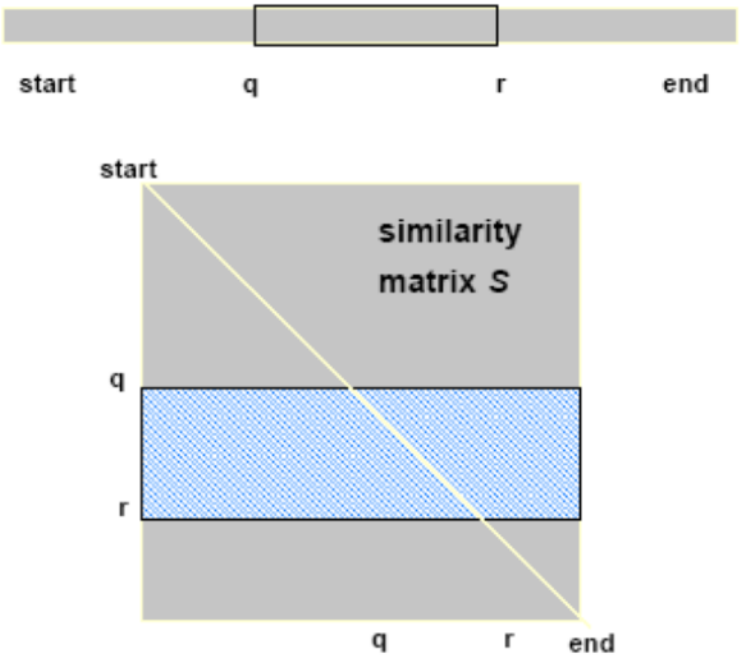
- Method:**"summary score"**
- Search for the continuous time segment which best represents the content of the music track according to a similarity criteria
- → generation of music "preview"



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

(3) SSM-based audio summary generation (cont.)

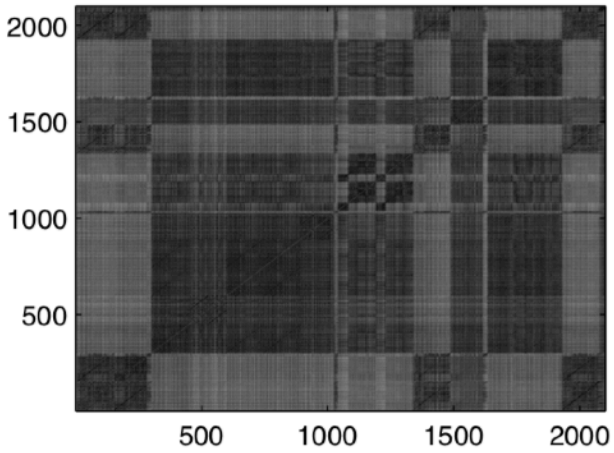
- Search for the segment which starts at q of duration $L = r - q$ which best explains the observed repetitions
- Average similarity between time q with all times of the music track
  - $\frac{1}{N}\sum_{n=1}^{N} \mathbf{S}_{q,n}$
- Average similarity between **segment** $[q, r]$ (of duration $L = r - q$) with all times of the music track
  - $R_{q,L} = \frac{1}{LN}\sum_{m=q}^{r}\sum_{n=1}^{N} \mathbf{S}_{m,n}$
- For a given L, we look for q^* which maximizes $R_{q,L}$
  - $q_L^* = \text{argmax}_{1 \le i \le N-L} R_{q,L}$
- Improvement: to favor the detection of summary at the start of the music track,
  - we add a weighting $w(n)$ function decreasing over time
  - $R_{q,L} = \frac{1}{LN}\sum_{m=q}^{r}\sum_{n=1}^{N} w(n) \cdot S_{m,n}$

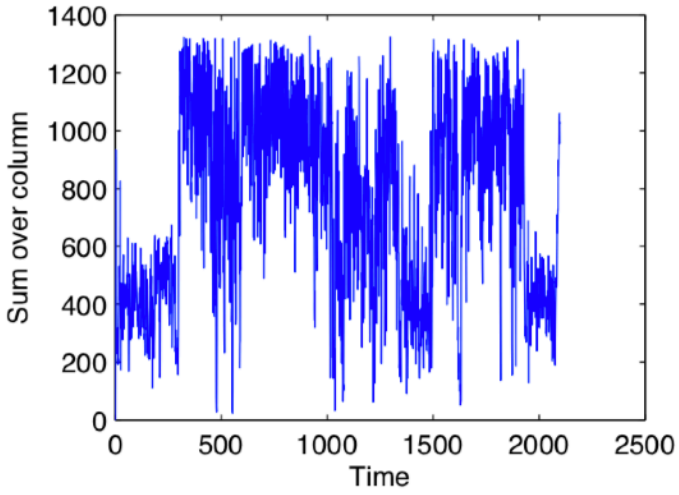


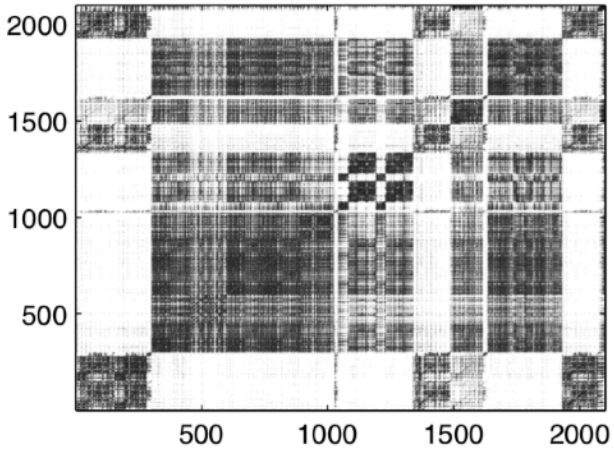
source : [Cooper and Foote, 2002, ISMIR]

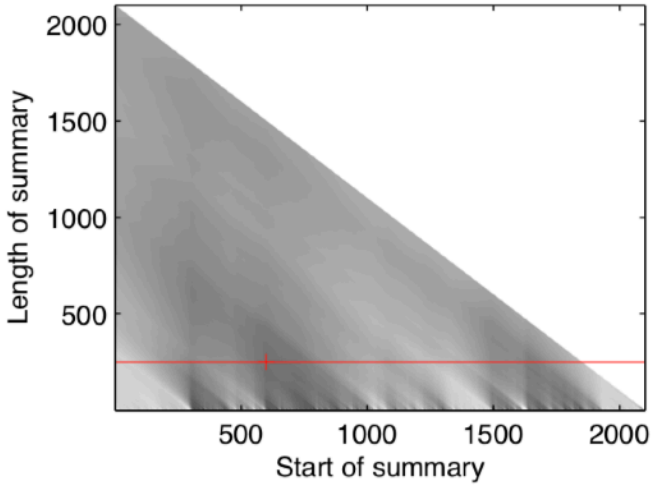
Music Structure Discovery (MSD) - Audio Summary Systems

(3) SSM-based audio summary generation (cont.)



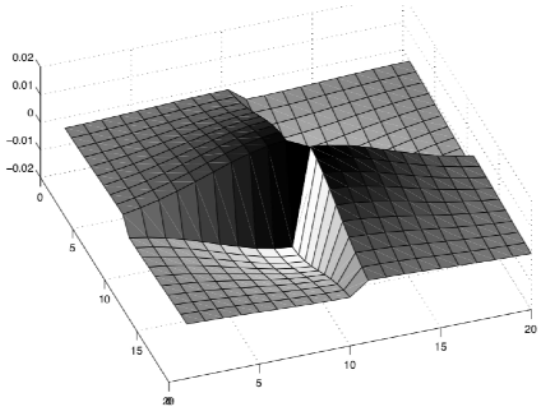




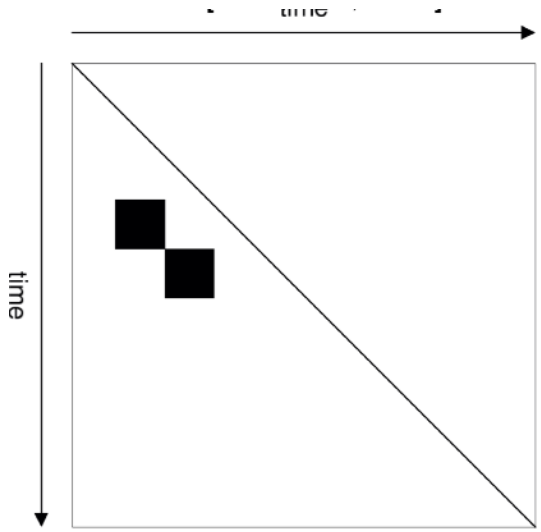


Temporal segmentation: Kernel-based approach

- Convolution of a Self-Similarity-Matrix $\mathbf{S}$ with a check-board ("damier") kernel
  - takes into account
    - intra-segment similarity (homogeneity) of the concent
    - dis-similarity between the content of the left and right segments
- Robust approach
- **"checker-board" kernel**
  - $\mathbf{C} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
  - The value at (t, t) on the main diagonal of the filtered matrix represent the similarity/ dis-similarity between the left segment $[t - \Delta, t]$ and right segment $[t, t + \Delta]$

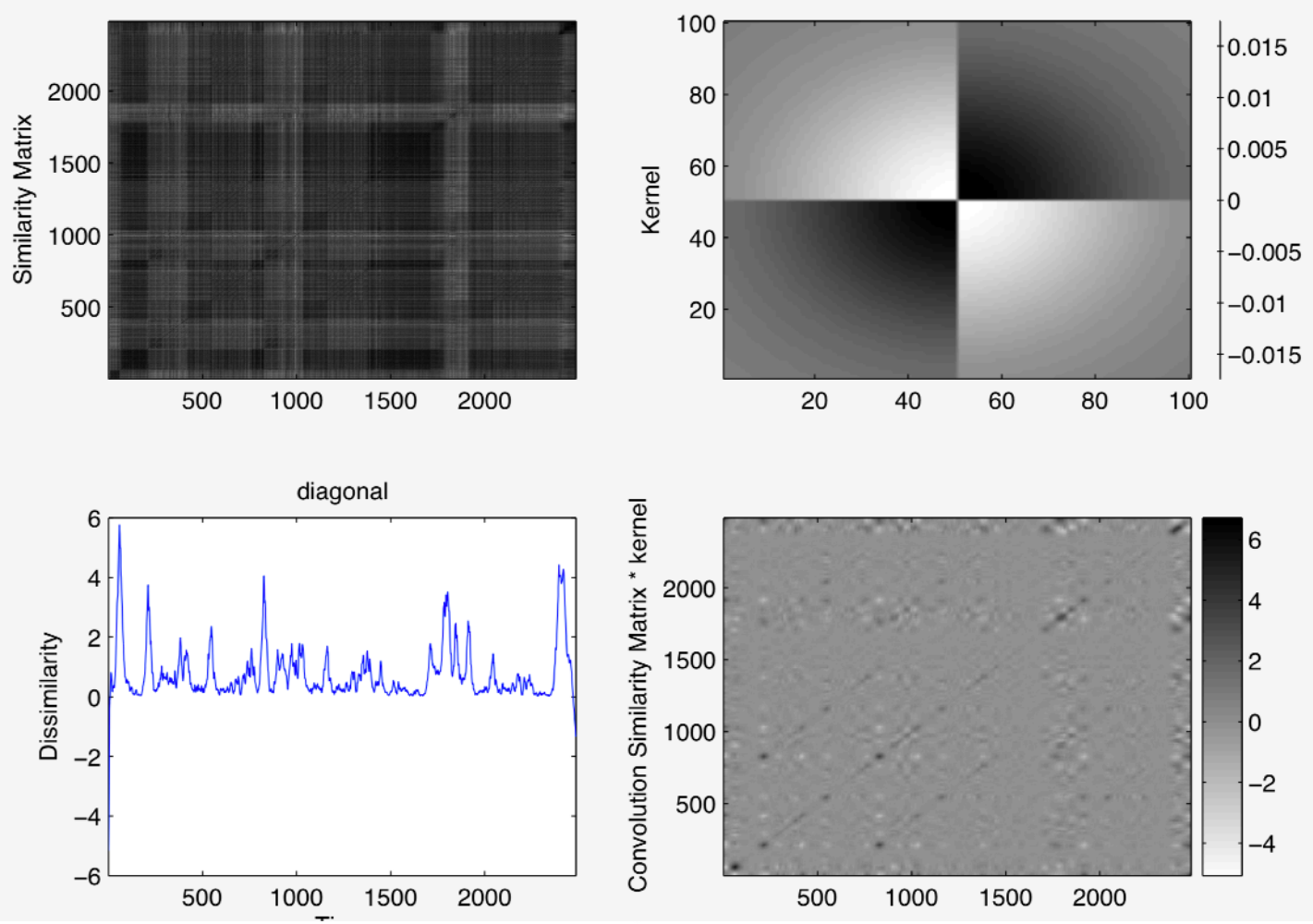


source : [Foote, 2000, ICME]



[J. Foote. Automatic audio segmentation using a measure of audio novelty. In Proc. of IEEE ICME, 2000.]

Temporal segmentation: Kernel-based approach (cont.)



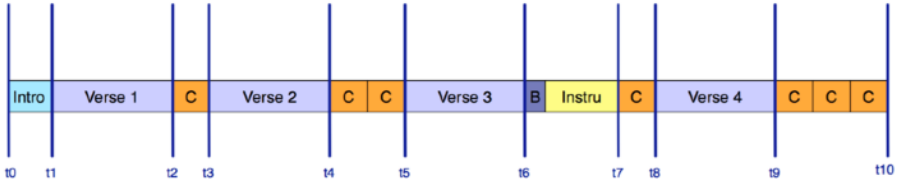
[J. Foote. Automatic audio segmentation using a measure of audio novelty. In Proc. of IEEE ICME, 2000.]

Music Structure Discovery (MSD) - Audio Summary Systems

(6) Deep-learning approaches

– **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

– **Method:**
  - 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, 2017.]

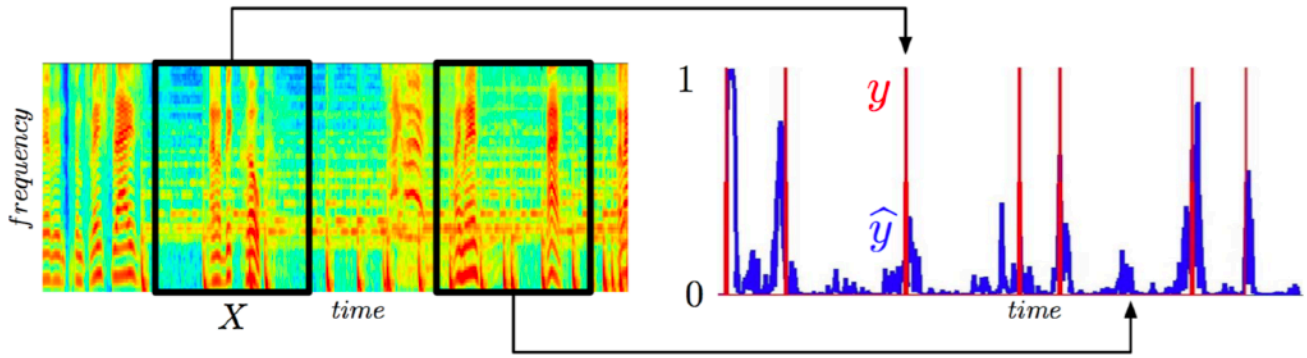
(6) Deep-learning approaches (cont.)

– **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.



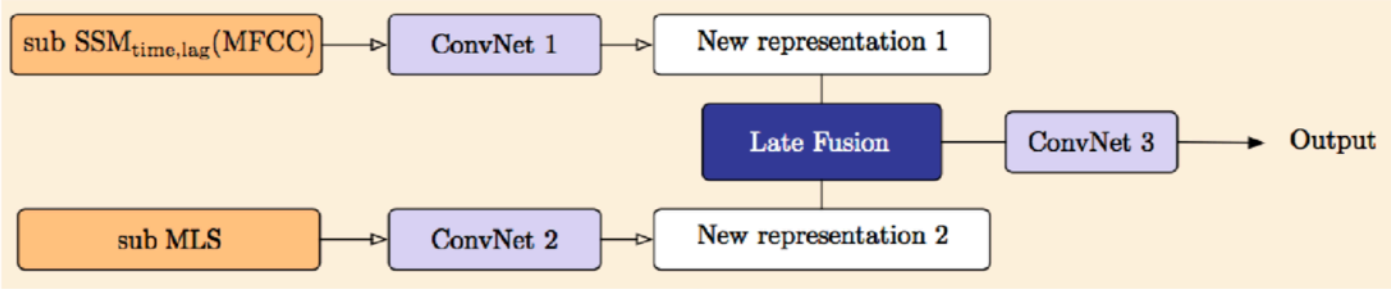
(6) Deep-learning approaches (cont.)

– **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 + $SSM_{time,lag}$:

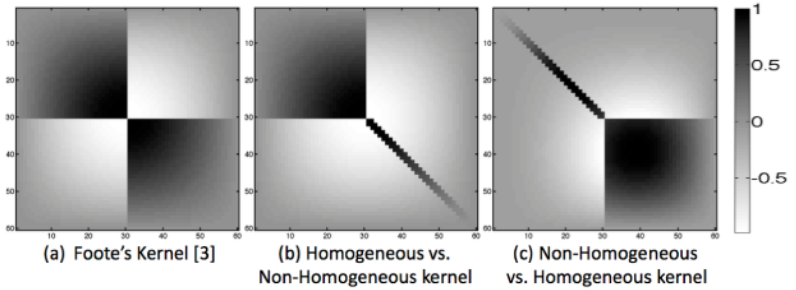
    Fusion of the two representations in a convolutional layer: Late Fusion.

    It is their best working model



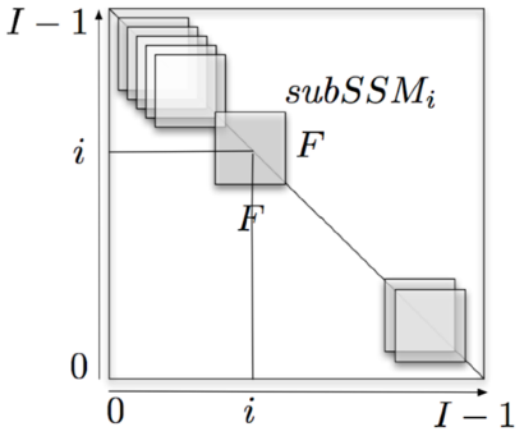
(6) Deep-learning approaches (cont.)

- **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 $SSM_{time,lag}$



Kaiser and Peeters multi-kernels for SSM segmentation

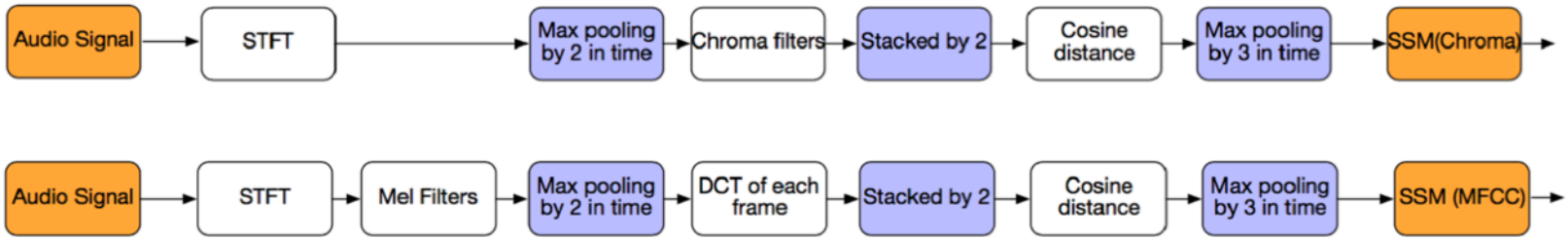
  - Use **square-sub-matrices** centered on the main diagonal of a Self-Similarity-Matrix time-time as input



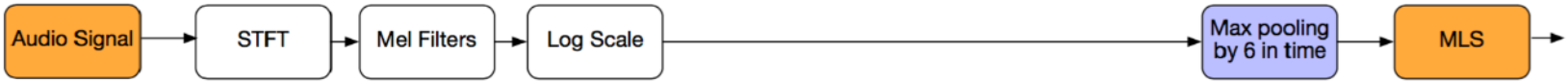
(6) Deep-learning approaches (cont.)

– **Input Representation : Our proposal**

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

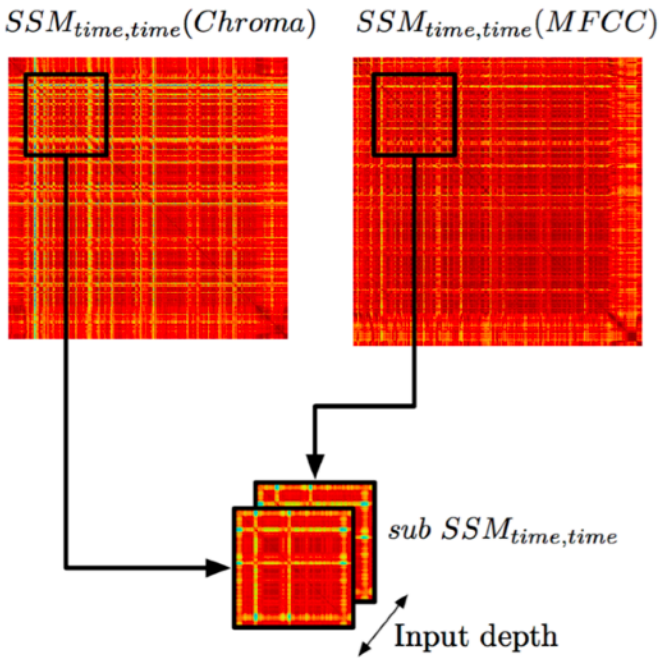


- Also use **Mel-Log-Spectrum (MLS)** as in [Ullrich et al., 2014] and [Grill and Schluter, 2015]



(6) Deep-learning approaches (cont.)

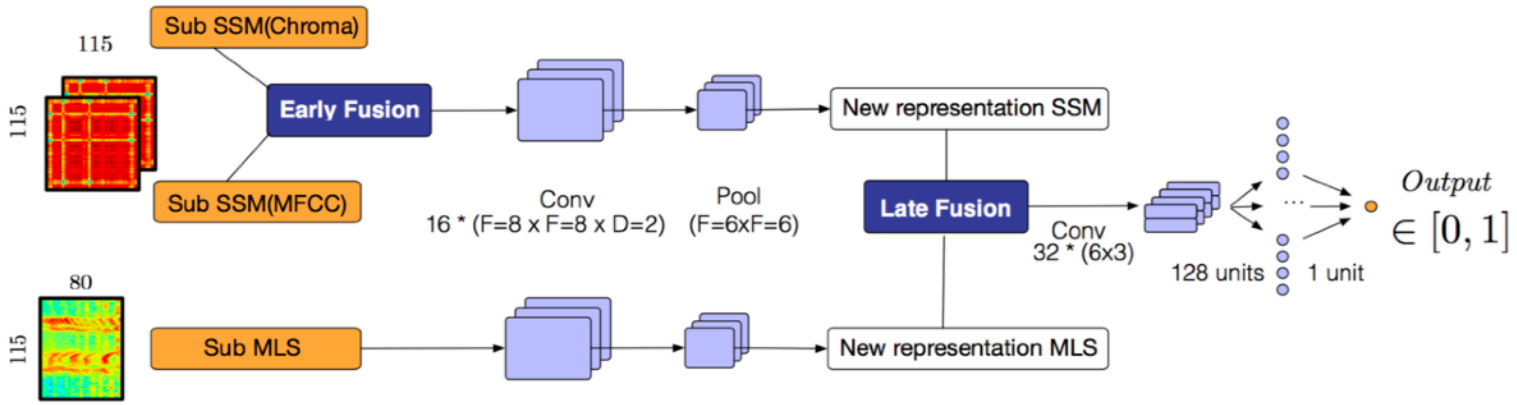
- **Input Representation : Our proposal**
  - **Depth**
    - Originally used to represent Red, Blue and Green (RGB) components of the input image.
  - Combine the SSM using the depth of the input layer of the ConvNet
    - Provide several points of view on the audio signal, according to different musical 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



(6) Deep-learning approaches (cont.)

– **Input Representation : Our proposal**

- **Late Fusion** between MLS and the SSM-depth



(6) Deep-learning approaches (cont.)

- **Evaluation**
  - **Dataset**
    - SALAMI: 1048 tracks, various music genres annotated at different scales and two 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 do not use peak-picking !
  - **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 $y = 1$

(6) Deep-learning approaches (cont.)

- **Systems compared;**
  - (1) MLS + $SSM_{time,time}$ (MFCC)
  - (2) MLS + $SSM_{time,time}$ (Chroma)
  - MLS + Depth-$SSM_{time,time}$
    - (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

- **Results:**
  - 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					

Music Structure Discovery (MSD) - Audio Summary Evaluation

Music Structure Discovery (MSD) - Audio Summary Evaluation

Task definition

- https://www.music-ir.org/mirex/wiki/2012:Structural_Segmentation

- Given an audio track
  → estimate the set of temporal (start:end) segments and associated structure label

```
0.0     Silence
0.464399092     Intro
14.379863945    no_function
23.986213151    no_function
33.622494331    Verse
42.956916099    no_function
49.681020408    Transition
67.005941043    Pre-Chorus
76.881292517    Chorus
86.425396825    no_function
98.689433106    Verse
108.166303854   no_function
115.474489795   Transition
129.466938775   Chorus
137.682789115   no_function
160.601927437   no_function
167.620181405   Pre-Chorus
177.151723356   Chorus
194.691836734   no_function
242.415328798   Outro
250.54893424    Fade-out
263.205419501   Silence
264.885215419   End
```

Music Structure Discovery (MSD) - Audio Summary Evaluation

Datasets

- https://www.audiocontentanalysis.org/data-sets/
- QMUL Isophonics (Beatles, Carole King, Queen, Michael Jackson)
  - http://isophonics.net/datasets
- AIST RWC (Real World Computing)
  - INRIA annotations: http://musicdata.gforge.inria.fr/structureAnnotation.html
- INRIA
  - Eurovision, Quaero: http://musicdata.gforge.inria.fr/structureAnnotation.html
- SALAMI (Structural Analysis of Large Amounts of Music Information)
  - https://ddmal.music.mcgill.ca/research/SALAMI/
- Harmonix-Set
  - https://github.com/urinieto/harmonixset
- ...

Music Structure Discovery (MSD) - Audio Summary Evaluation

Performance measures

– **Two criteria to evaluate**

- **(1)** get the correct segment boundaries (independently of the labels)
- **(2)** get the correct label at each time → **labels are relatives !!!**

H. Lukashevich. Toward quantitative measures of evaluating song segmentation. In Proc. of ISMIR (International Society for Music Information Retrieval), Philadelphia, PA, USA, 2008.

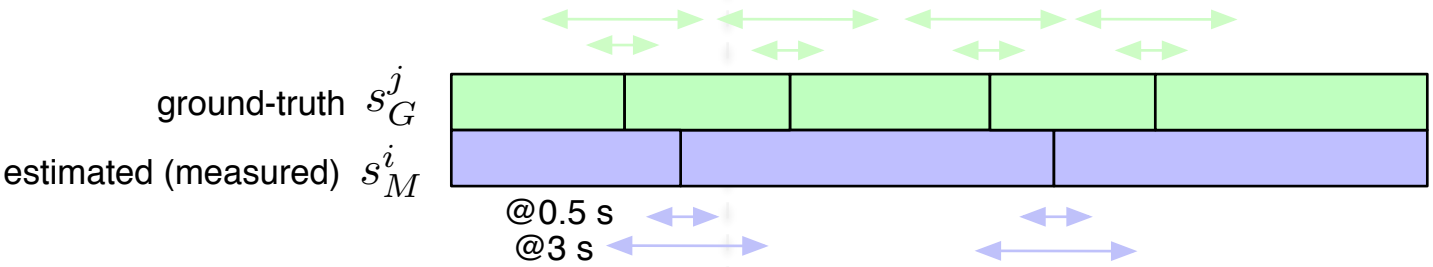
Music Structure Discovery (MSD) - Audio Summary Evaluation

(1) get the correct segment boundaries (independently of the labels)

- S_G^j: ground-truth segments
- S_M^i: estimated (measured) segments

- Segment boundary recovery
  - Recall, Precision, F-measure @0.5 s, @3 s

- Median distance from
  - an annotated segment boundary S_G^j to the closest found boundary S_M^i
  - a found segment boundary S_M^i to the closest annotated boundary S_G^j



Music Structure Discovery (MSD) - Audio Summary Evaluation

Example of results

Summary

Legend

Submission code	Submission name	Abstract PDF	Contributors
KSP1	ircamstructure_va2	PDF	Florian Kaiser, Thomas Sikora, Geoffroy Peeters
KSP2	ircamstructure_vt1	PDF	Florian Kaiser, Thomas Sikora, Geoffroy Peeters
KSP3	ircamstructure_mph1	PDF	Florian Kaiser, Thomas Sikora, Geoffroy Peeters
MHRAF1	Simbals_Structure	PDF	Benjamin Martin, Pierre Hanna, Matthias Robine, Julien Allali, Pascal Ferraro
OYZS1	OYZS	PDF	Nobutaka Ono, Shinya Yaku, Yuko Zou, Shigeki Sagayama
SBV1	Music Structure Inference System	PDF	Gabriel Sargent, Frédéric Bimbot, Emmanuel Vincent
SMGA1	SMGA1	PDF	Joan Serrà, Meinard Müller, Peter Grosche, Josep Lluis Arcos
SMGA2	SMGA2	PDF	Joan Serrà, Meinard Müller, Peter Grosche, Josep Lluis Arcos
SP1	ircamstructure_va1	PDF	Florian Kaiser, Thomas Sikora, Geoffroy Peeters

Summary Results [top]

Algorithm	Normalised conditional entropy based over-segmentation score	Normalised conditional entropy based under-segmentation score	Frame pair clustering F-measure	Frame pair clustering precision rate	Frame pair clustering recall rate	Random clustering index	Segment boundary recovery evaluation measure @ 0.5sec	Segment boundary recovery precision rate @ 0.5sec	Segment boundary recovery recall rate @ 0.5sec	Segment boundary recovery evaluation measure @ 3sec	Segment boundary recovery precision rate @ 3sec	Segment boundary recovery recall rate @ 3sec	Median distance from an annotated segment boundary to the closest found boundary	Median distance from a found segment boundary to the closest annotated one
MHRAF1	0.6319	0.5227	0.5722	0.5640	0.6723	0.6432	0.1879	0.1944	0.1992	0.4229	0.4446	0.4402	6.2389	5.4963
SMGA1	0.6231	0.6759	0.5809	0.6762	0.5826	0.6999	0.1924	0.1563	0.2816	0.4920	0.4040	0.7028	1.7681	6.7133
OYZS1	0.6215	0.5456	0.5006	0.5817	0.5954	0.5954	0.2874	0.4580	0.2527	0.4368	0.6409	0.3970	20.4822	3.7538
SP1	0.5899	0.5062	0.5543	0.5490	0.6395	0.6385	0.2789	0.2237	0.4371	0.4906	0.3924	0.7676	1.2193	7.3354
SMGA2	0.5542	0.7400	0.5282	0.7285	0.4712	0.6985	0.1782	0.1460	0.2572	0.4789	0.3959	0.6779	1.9191	6.5524
KSP3	0.5526	0.6052	0.5309	0.6120	0.5261	0.6663	0.2789	0.2237	0.4371	0.4906	0.3924	0.7676	1.2193	7.3354
KSP1	0.5251	0.6789	0.5019	0.6653	0.4464	0.6755	0.2787	0.2234	0.4368	0.4902	0.3921	0.7671	1.2171	7.3263
KSP2	0.5229	0.5026	0.5283	0.5503	0.5792	0.6353	0.2860	0.2291	0.4486	0.4899	0.3915	0.7676	1.1987	7.3180
SBV1	0.4773	0.6481	0.4596	0.6271	0.4250	0.6469	0.1566	0.1359	0.2095	0.4344	0.3781	0.5744	2.6786	8.1783