Lecture

Speech and Audio Signal Processing

Lecture 13: Music Signal Processing

Content

Three examples of music processing based on major features of music:

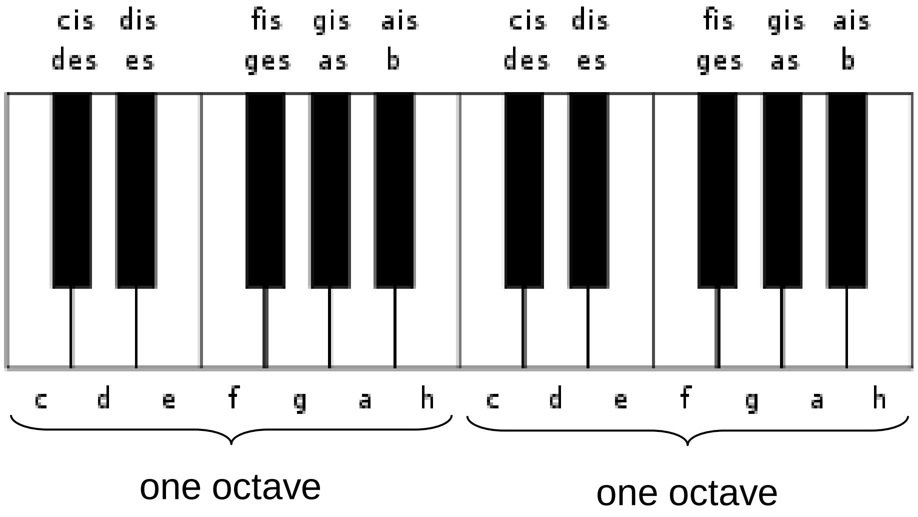
- Chroma based processing:
  - Tone pitch analysis and relation to chroma features.
  - Normalization and log chroma features.
  - Chromagrams.
  - Application: audio matching procedures: retrieval of audio queries.
    - Target: Insensitivity to dynamics, timbre, articulation, and tempo.

- Beat processing:
  - Beat detection and beat tracking including applications.
  - Calculation procedures.

- Shazam – App:
  - Recognition of music songs

Tone pitch

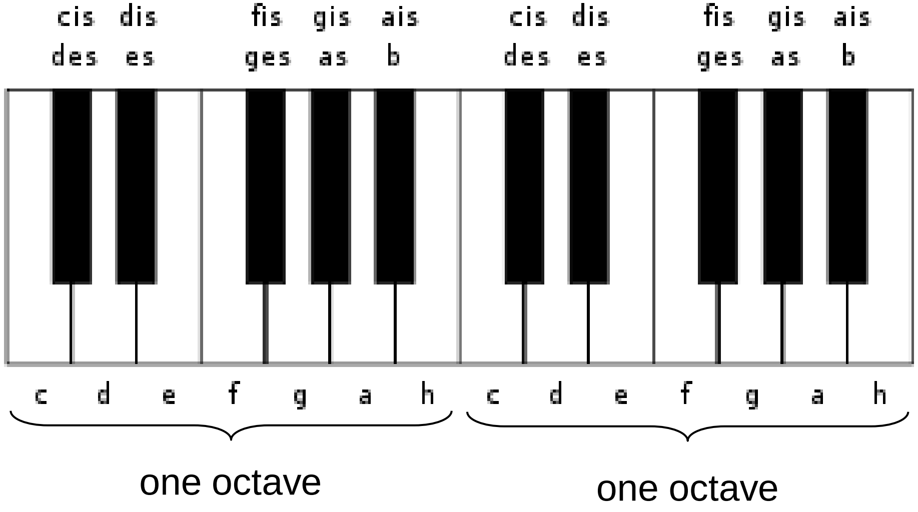
- Western tonal music is typically based on 12 pitch classes, known as “chroma”
- One octave contains these 12 pitch classes
- Consecutive octaves double or halve the frequencies of the respective pitches.



Tone pitch

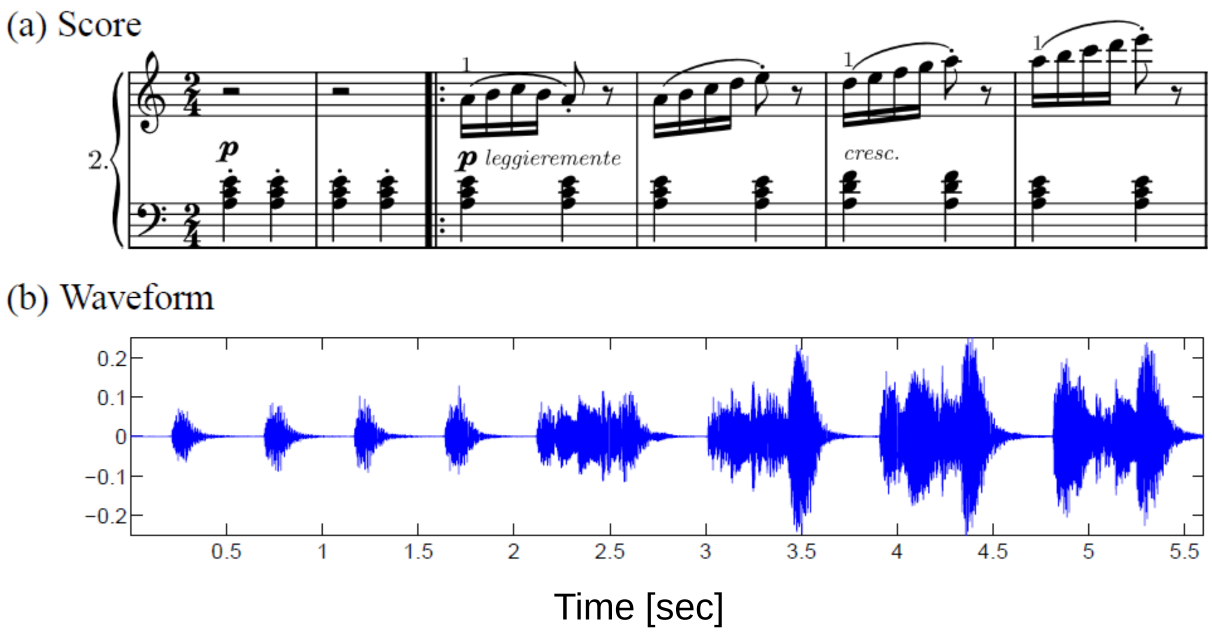
- A typical chroma feature extraction is based on 88 frequency bands.
- The center frequencies of these bands go from A0 to C8, i.e., going form the pitch A in octave 0 to C in octave 8 and covering seven full octaves.

- Known reference value: A4 => 440 Hz (C4 => 262 Hz)
- leads to A0 => 27.5 Hz (440 Hz / (2^4)).
- and C8 => 4192 Hz.



Music signal description

- Description by scope and corresponding waveform:


(a) Score

(b) Waveform

Extraction of chroma features: Overview

- The extraction procedure for chroma features consists of several steps:
  - 1) Decomposition of the audio signal into 88 pitch frequency bands corresponding to the pitches A0 to C8 (equal to the MIDI pitches p = 21 to 108).
  - 2) *Chroma Pitch* (CP) features are calculated by summing the energies of all pitch values corresponding to the same chroma. Example: For chroma C the energy values of the pitches C1, to C8 are summed.

    This leads to the 12-dimensional chroma vector:

$$\boldsymbol{x}_{\mathrm{CP}} = [\underset{\substack{\uparrow \\ \text{C}}}{x(1)},\, \underset{\substack{\uparrow \\ \text{Cis}}}{x(2)},\, \ldots,\, x(12)]^{\mathrm{T}}$$

Extraction of chroma features: Overview

- The extraction procedure for chroma features consists of several steps:
  - 3) Normalization by a vector norm, with either $p = 1$ or $p = 2$.

$$\|\boldsymbol{x}_{\mathrm{CP}}\|_p = \left(\sum_{i=1}^{12} |x(i)|^p \right)^{1/p}$$

  - 4) *Chroma-Log-Pitch* (CLP) features:
    Taking the log value of the normalized CP feature values.

Targeted application example

- **Automatic audio retrieval, i.e., retrieval of audio clips:**

Meaning:

Given a short query audio clip, automatically retrieve the corresponding excerpts, independent of the interpretation, e.g., by different conductors.

Also, the same audio clip may by present in the same piece of music several times, by several repetitions or different interpretations.

=> **Targets:**

Robustness with respect to:

- dynamics, timbre, articulation, and (local) tempo variations.

Typical query clip lengths of 10 to 30 sec.

Pitch representation

- 1) Decomposition of the audio signal into 88 pitch frequency bands

  - Apply 88 elliptic band pass filters (IIR filters) to filter out the relevant pitch signals.

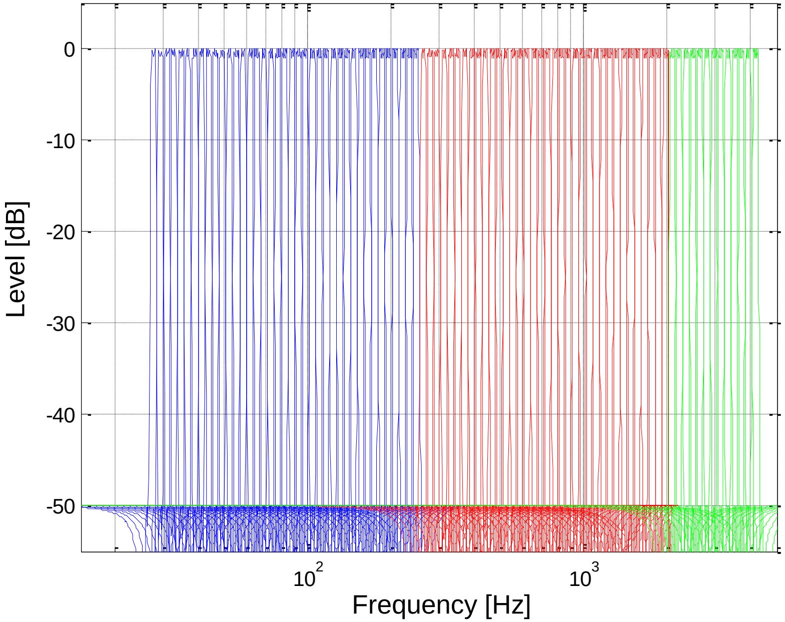
  - In order to cope with the small pass-band for low frequencies, low- and mid-range frequency components are first extracted and subsampled at lower frequencies.

  - Example setup:
    fs = 22.05 kHz;
    => low frequency components (up to 250 Hz) are subsampled
    by a factor of 25 => fs_low = 882 Hz
    => mid frequency components (up to 2 kHz) are subsampled
    by a factor of 5 => fs_mid = 4410 Hz.

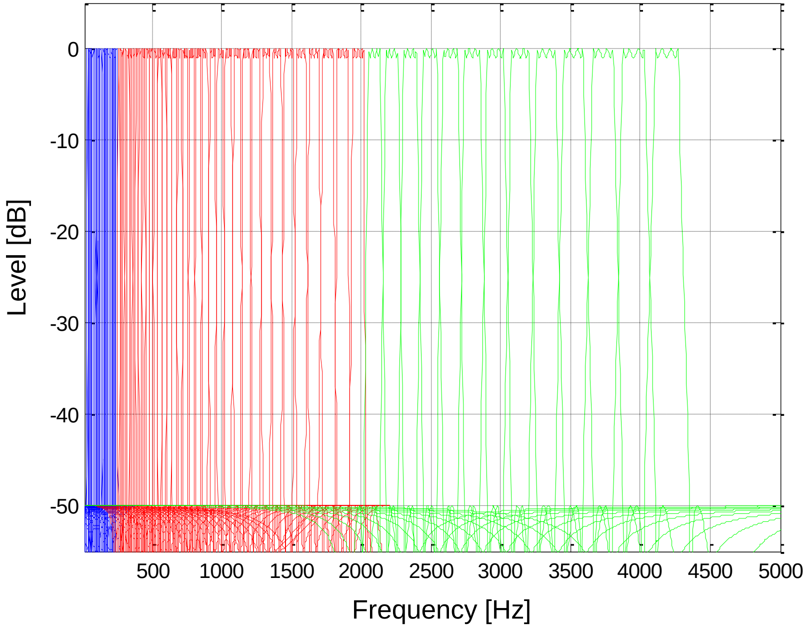
Pitch representation

- 1) Decomposition of the audio signal into 88 pitch frequency bands
  - Showing the frequency responses of the 88 band-pass filters:

Logarithmic frequency scale

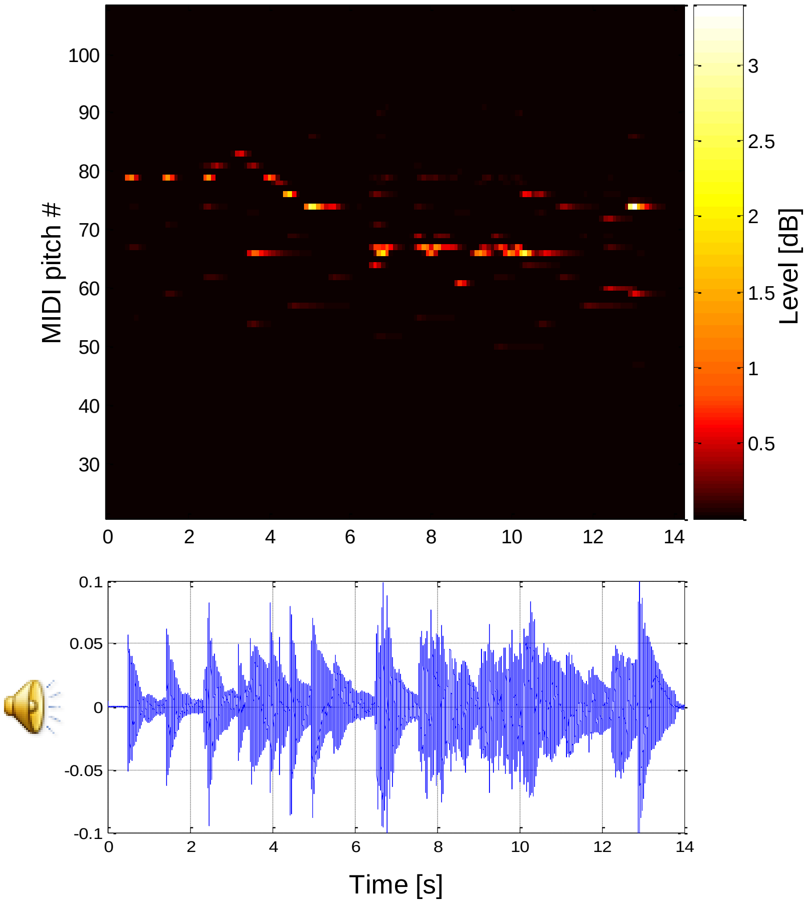


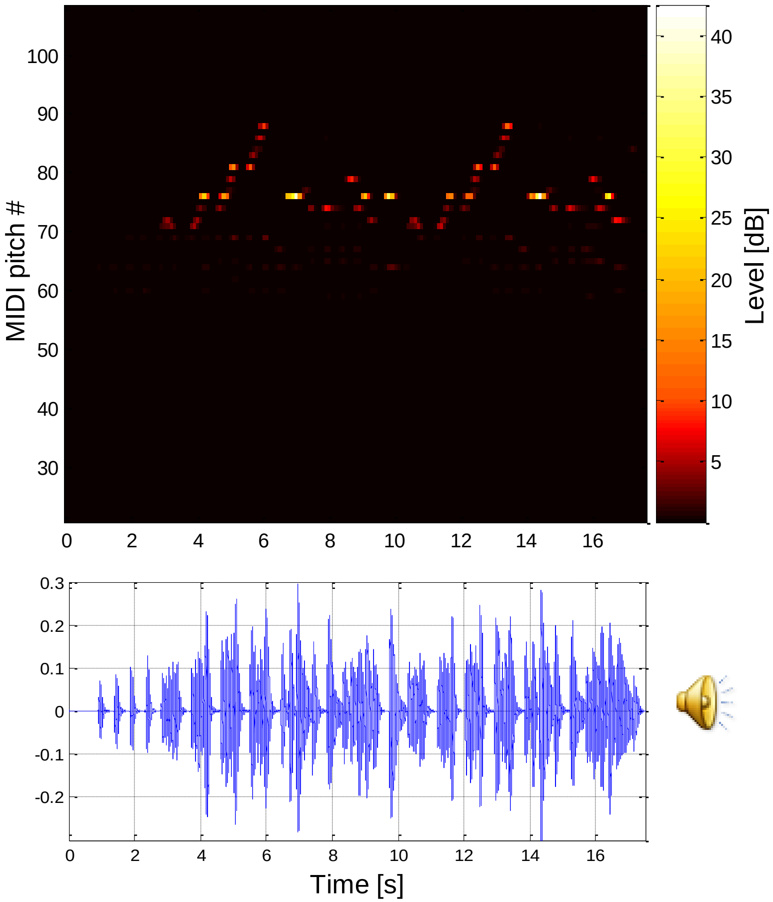
Linear frequency scale



Pitch representation

- Two examples for pitch energy analysis over time:



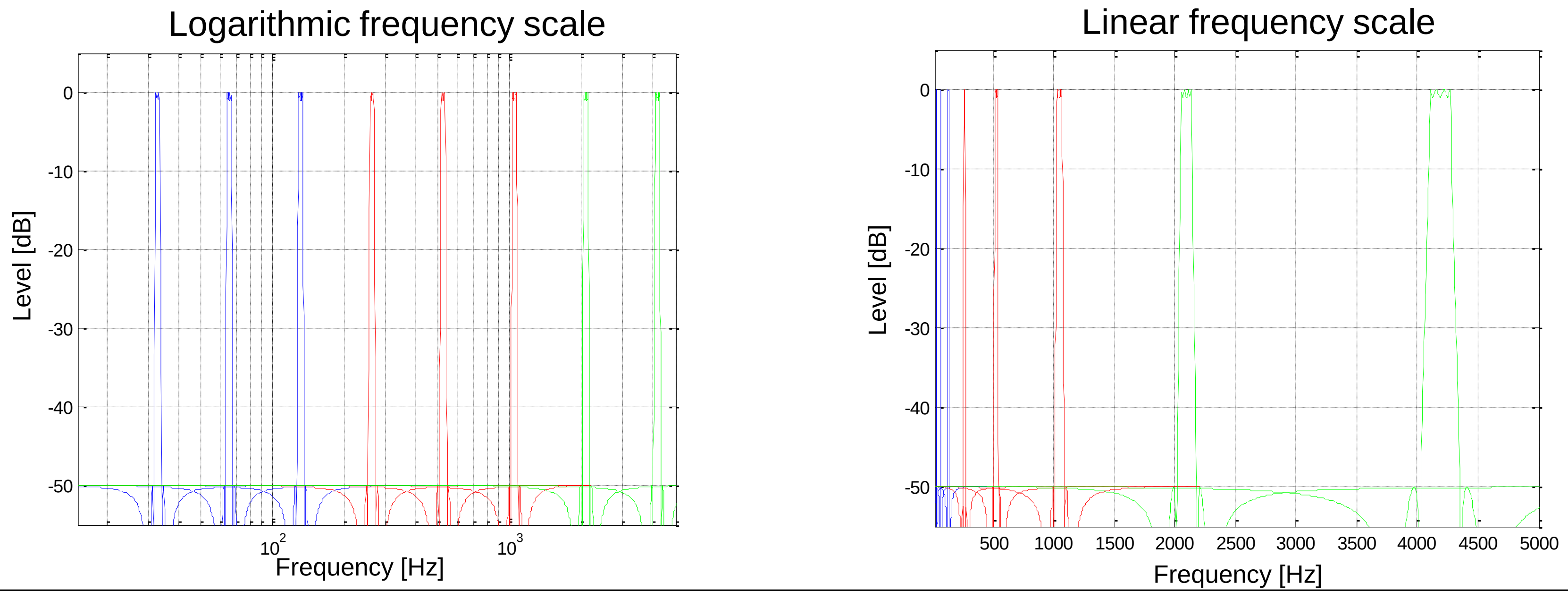


Frequency selective filtering

- The extraction procedure for chroma features consists of several steps:

  - 2) *Chroma Pitch* (CP) feature calculation
    => leading to a 12-dimensional chroma vector:
    => Summing the power of the corresponding pitch energy values:

  - Example for the filters for the chroma C:



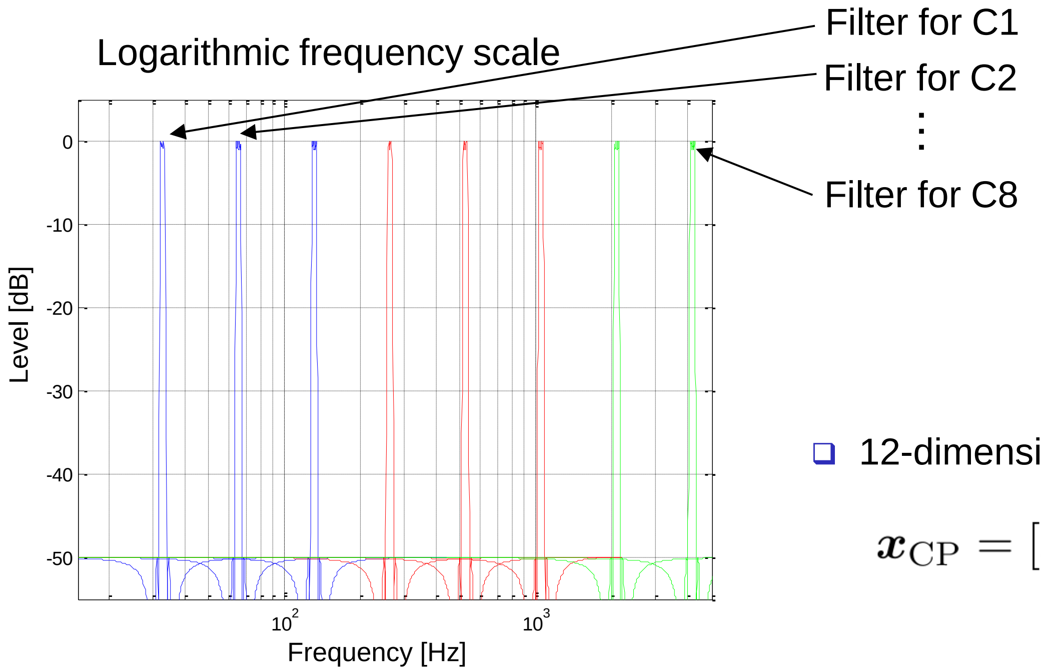
Logarithmic frequency scale

Linear frequency scale

Chroma pitch (CP) calculation

- The extraction procedure for chroma features consists of several steps:

  - 2) *Chroma Pitch* (CP) feature calculation
    => leading to a 12-dimensional chroma vector:
    => Summing the power of the corresponding pitch energy values:



Logarithmic frequency scale

Filter for C1

Filter for C2

⋮

Filter for C8

- Example for the chroma C: Summation of the energy of the eight filter outputs:

$$x(1) = \sum_{i=1}^{8} \sigma_{Ci}^2$$

- 12-dimensional chroma vector:

$$\boldsymbol{x}_{\mathrm{CP}} = [\underset{\uparrow\,\mathrm{C}}{x(1)},\ \underset{\uparrow\,\mathrm{Cis}}{x(2)},\ \ldots,\ x(12)]^{\mathrm{T}}$$

Chroma pitch (CP) and Chroma log pitch (CLP)

- 2) **Chroma pitch (CP) feature**: 12-dimensional chroma vector:

$$\boldsymbol{x}_{\mathrm{CP}} = [\underset{\uparrow\,\text{C}}{x(1)},\ \underset{\uparrow\,\text{Cis}}{x(2)},\ \ldots,\ x(12)]^{\mathrm{T}}$$

- 3) **Normalized chroma pitch feature**
  normalization by a vector norm, with either $p = 1$ or $p = 2$.

$$\boldsymbol{x}_{\mathrm{CP,\ norm}} = \boldsymbol{x}_{\mathrm{CP}} / \|\boldsymbol{x}_{\mathrm{CP}}\|_p \qquad \text{with: } \|\boldsymbol{x}_{\mathrm{CP}}\|_p = \left(\sum_{i=1}^{12} |x(i)|^p\right)^{1/p}$$

- 4) **Chroma log pitch (CLP) feature:**
  normalized pitch energy values before summation:

$$\boldsymbol{x}_{\mathrm{CLP}} = [x_{\mathrm{dB}}(1),\ x_{\mathrm{dB}}(2),\ \ldots,\ x_{\mathrm{dB}}(12)]^{\mathrm{T}}$$

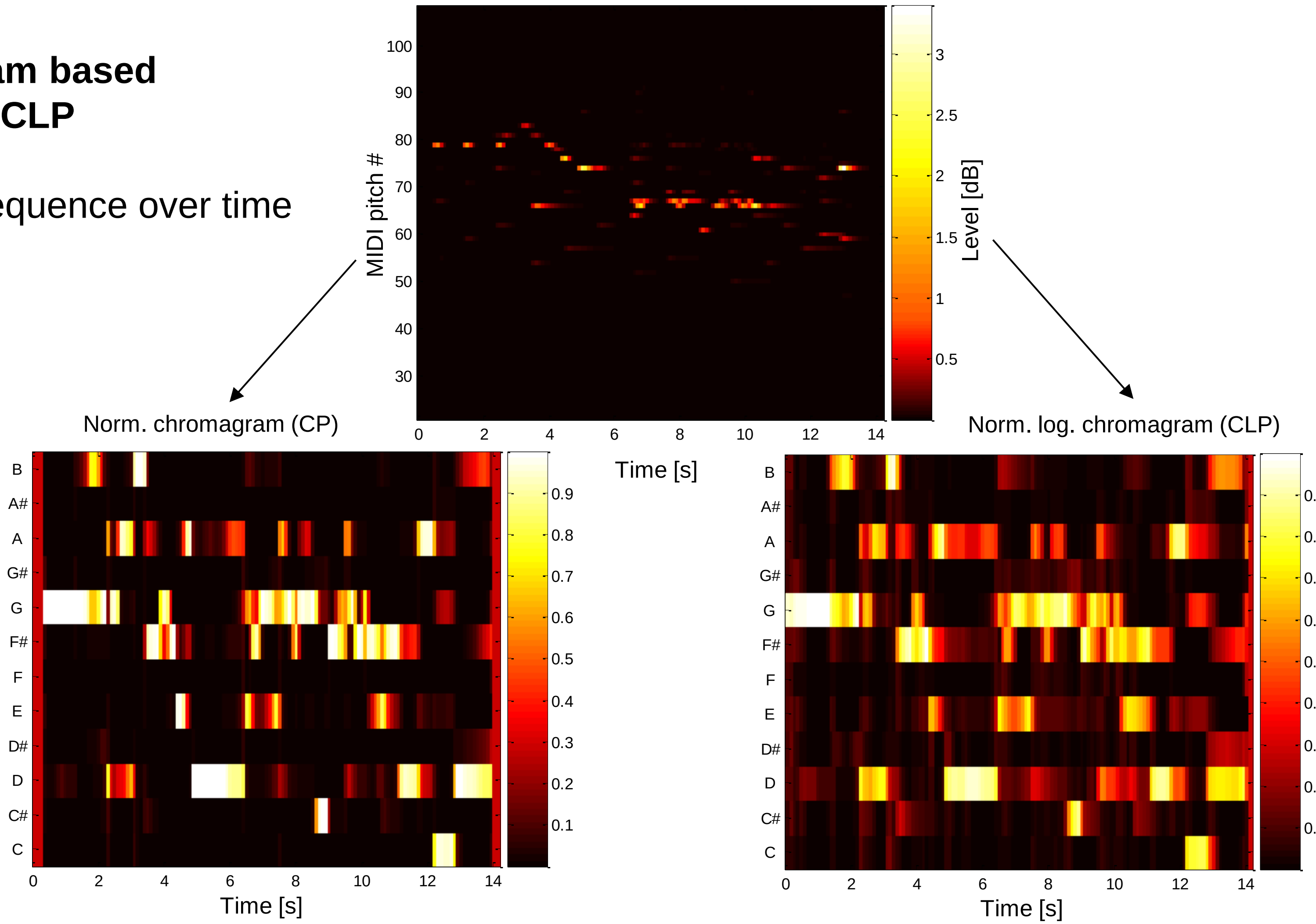
$$x_{\mathrm{dB}}(1) = \sum_{i=1}^{8} 10 * \log 10 \left(\frac{\sigma_{Ci}^2}{\sigma_{0\ \mathrm{dB}}^2}\right)$$

Example for the chroma C

Chroma pitch (CP) and Chroma log pitch (CLP)

- **Chromagram based on CP and CLP**

=> Vector sequence over time



CENS features (Chroma Energy Normalized Statistics)

- **Chroma Energy Normalized Statistics (CENS) feature**:
  Target: reduce sensitivity with respect to articulation and local tempo variation. => level quantization and time smoothing

=> 4-step logarithmic quantization of the normalized CP feature:

$$\boldsymbol{x}_{\text{CP, norm}} = \boldsymbol{x}_{\text{CP}} / \|\boldsymbol{x}_{\text{CP}}\|_p$$

- The choice of logarithmic values introduces a logarithmic suppression:

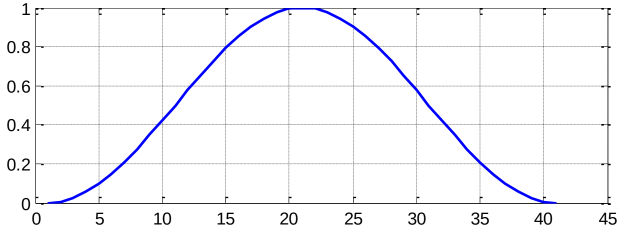
$$x_{\text{CENS}}(i) = \frac{1}{4}\begin{cases} 4 & : \text{ if } x_{\text{CP, norm}}(i) \geq 0.4 \\ 3 & : \text{ if } 0.2 \leq x_{\text{CP, norm}}(i) < 0.4 \\ 2 & : \text{ if } 0.1 \leq x_{\text{CP, norm}}(i) < 0.2 \\ 1 & : \text{ if } 0.05 \leq x_{\text{CP, norm}}(i) < 0.1 \\ 0 & : \text{ if else} \end{cases}$$

$$\boldsymbol{x}_{\text{CENS}} = [x_{\text{CENS}}(1),\, x_{\text{CENS}}(2),\, \ldots,\, x_{\text{CENS}}(12)]^{\text{T}}$$

CENS features (Chroma Energy Normalized Statistics)

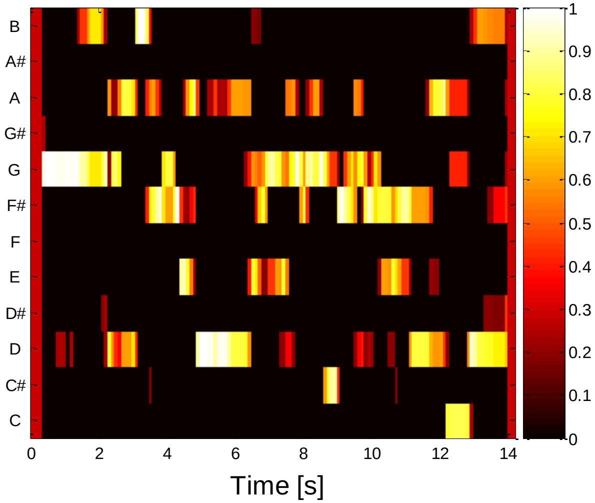
- Chroma Energy Normalized Statistics **(CENS) feature**:

- Smoothing over time e.g., by a convolution with a Hann window of length 41:

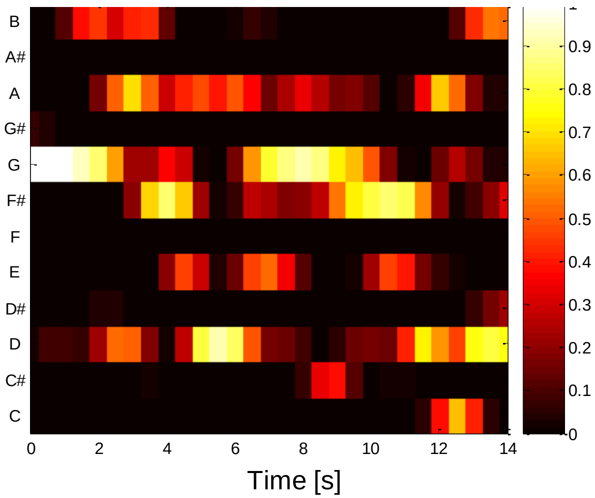


- => subsampling by a factor 10 is possible:

original CENS chromagram



smoothed and subsampled CENS chromagram
=> Including a normalization by the L2-norm.



CENS features (Chroma Energy Normalized Statistics)

- **CENS feature**s
  are appropriate for audio matching tasks, i.e., retrieve audio clips in pieces of music. => Robustness & Accuracy

- They exhibit the following properties:
  - Characterization of music accurately, independently of the specific interpretation.
  - Parameters such as „dynamics” and „timbre“ / „articulation“ are masked out by a large extend.
  - Reasons:
    - Normalization => **invariant to dynamics**
    - Chroma instead of pitch => takes the close octave relationship in melody and harmony into account. Additionally, **robustness to variations in timbre**.
  - Log. energy thresholds (quantization)
    => **insensitivity to noise components**
  - Hann windowing / smoothing => **insensitivity to local time variations**.

Audio matching procedure

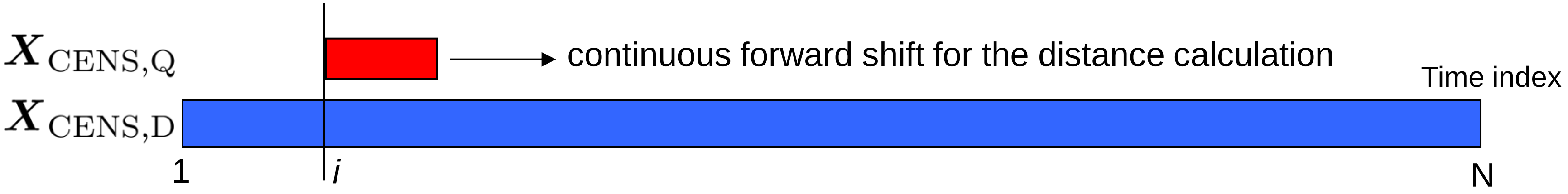
- **Setup:**
  - Assumptions:
    - Short audio clip (10-30 sec) => **Query Q**
    - Large Audio document (e.g., concatenation of several audio pieces) => **Document D**
  - Two feature sequences:

$$\boldsymbol{X}_{\mathrm{CENS,Q}} = [\boldsymbol{x}_{\mathrm{CENS,Q}}(1),\, \boldsymbol{x}_{\mathrm{CENS,Q}}(2),\, \ldots,\, \boldsymbol{x}_{\mathrm{CENS,Q}}(M)]$$

$$\boldsymbol{X}_{\mathrm{CENS,D}} = [\boldsymbol{x}_{\mathrm{CENS,D}}(1),\, \boldsymbol{x}_{\mathrm{CENS,D}}(2),\, \ldots,\, \boldsymbol{x}_{\mathrm{CENS,D}}(N)] \qquad \text{with: } N \gg M$$

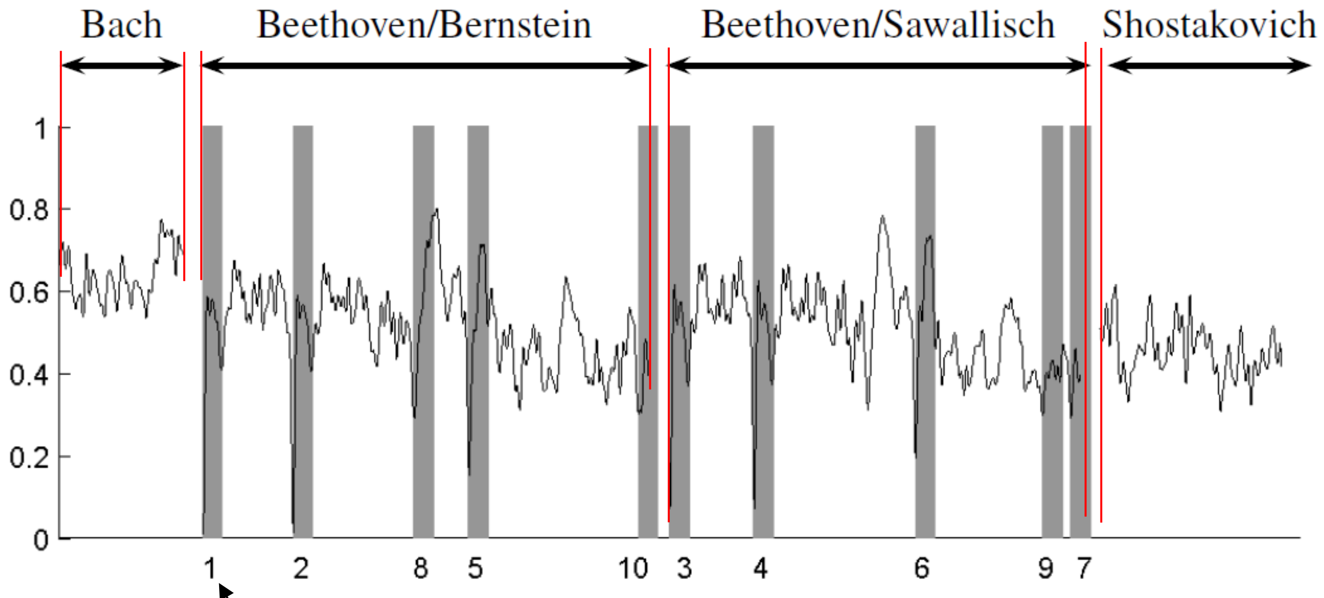
  - Distance for the time index *i* of the document D (products of identical vectors result in ‘1’):

$$\Delta^{(i)} = 1 - \frac{1}{M}\sum_{j=1}^{M} \boldsymbol{x}_{\mathrm{CENS,D}}(i+j-1)^{\mathrm{T}}\, \boldsymbol{x}_{\mathrm{CENS,Q}}(j)$$


$\boldsymbol{X}_{\mathrm{CENS,Q}}$
$\boldsymbol{X}_{\mathrm{CENS,D}}$
continuous forward shift for the distance calculation
Time index
1
i
N

Audio matching procedure

- **Example result:**
  - Concatenated pieces of four different pieces with the query occurring in two pieces several times



Number => according to the order of the minima

Global tempo variations

- In case of different interpretations (by different conductors) of the same piece of music
  => one has to account for tempo variations

- This can be done by varying the length of the smoothing window and the down-sampling factor (s. slide 17 => length *l* = 41, down-sampling, *ds* = 10)

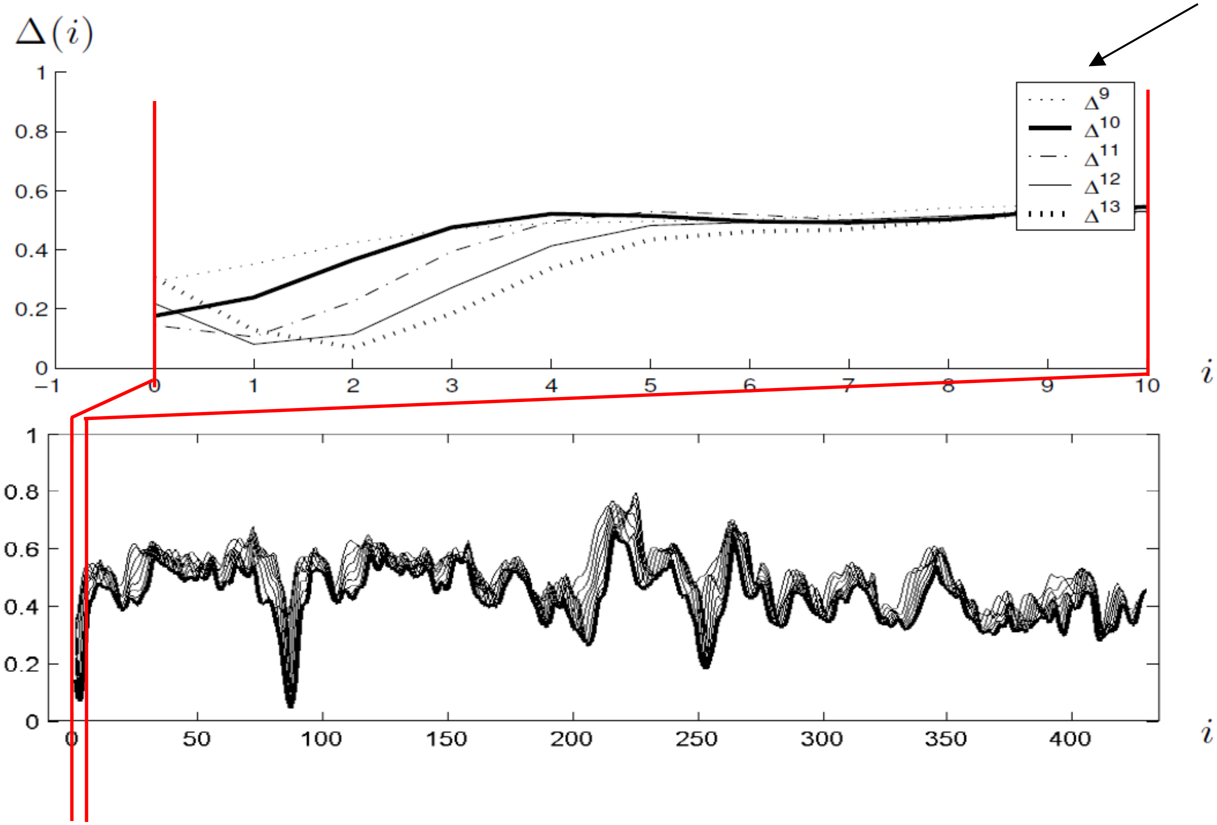
  => different tempo variations (*tv*) possible (values referring to query):

l	29	33	37	41	45	49	53	57
ds	7	8	9	10	11	12	13	14
tv	1.43	1.25	1.10	1.00	0.90	0.83	0.77	0.70

- => Comparison with 8 different queries

Global tempo variations

- Example for four different down-sampling values of the query 9,10,12, and 13 with the document being down-sampled with a value of 10:



Showing a zoom of the lower plot

First 10 distance values of below

Distance measures for differently sampled queries:

CRP (Chroma DCT-Reduced log Pitch) features

- Additional feature (CRP) based on cepstral processing:
  Motivation: further removal of the sensitivity with respect to timbre

se remover os componentes de low-quefrency, vc tira o "
"envelope" da musica e fica independente do instrumento

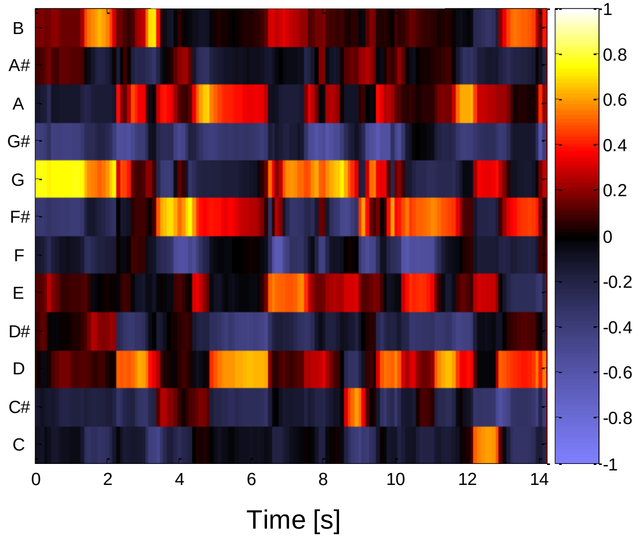
- Timbre is characterized by the spectral envelope.

- => Remember: in speech processing: first cepstral coefficients characterize the envelope
- => calculate the cepstral feature coefficients of music and remove the first coefficients

- This procedure is as follows:
  - Log. compression on the 88 pitch values
  - DCT => cepstral coefficients
  - Removal of the first cepstral coefficients
  - Inverse DCT
  - Projection on 12-dim chroma vector
  - Normalization with L2 norm

Spectral envelope removal

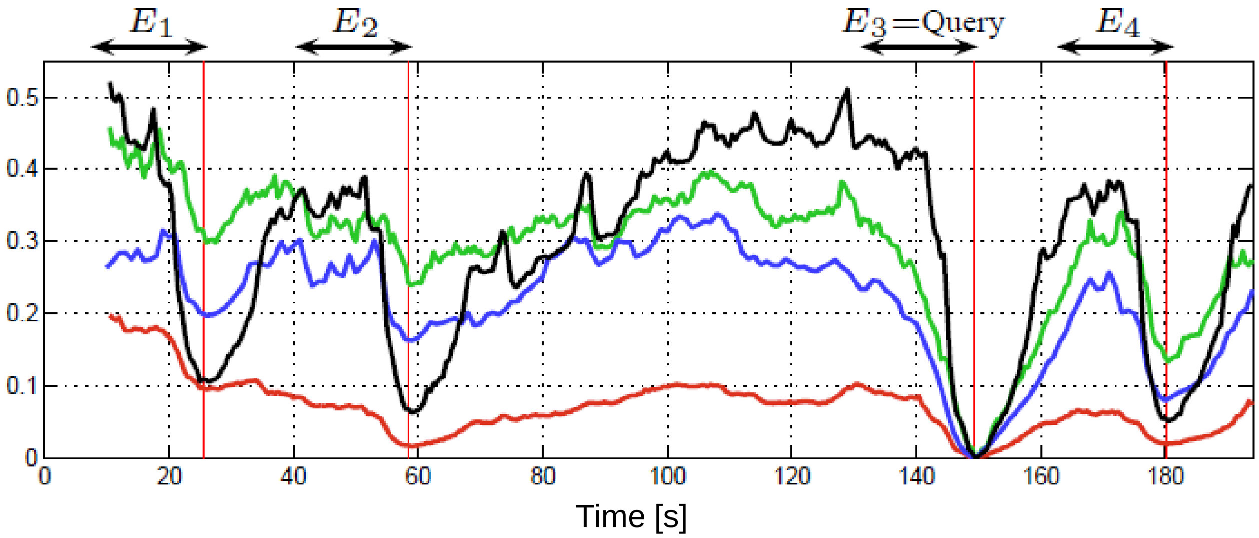
CRP chromagram



Comparison of several features

- **Feature comparison:**
- Document with four matches with four different instrumentation:
  => clarinet, strings, trombone, tutti (=> Klarinette, Streicher, Posaune, Orchester)

- => CRP feature has highest sensitivity toward the queries
  independent of the instrumentation:



Feature colors:

CP: green
CLP: red
CENS: blue
CRP: black

Beat tracking and detection

- **Target & applications:**
  - **Beat detection:**
    - Analyse an audio signal and detect a beat, e.g., harmonic onsets.
    - In case of a presence of a beat, this is an (one!) indicator for music. (Other indicators are harmonics, constants of a frequency over a longer time, etc.)
  - **Beat tracking:**
    - Here the assumption is that a music signal is analysed and a beat is present which should be tracked as best as possible.
  - **Indicators of beats:**
    - The main indicator used in the following is an onset detection based on a summation of raising signal power in frequency bands.

Beat tracking and detection

- **Several step procedure:**
  - 1) Frequency analysis (with STFT or filterbank processing)

$$X\left(e^{j\Omega_\mu},n\right)=\sum_{k=0}^{N-1}x(n-k)\,h_k\,e^{-j\frac{2\pi}{N}k\mu}$$

  - 2) Logarithm of the magnitude spectrum:

$$X_{\log}(e^{j\Omega_\mu},n)=\log\left(\left|X(e^{j\Omega_\mu},n)\right|+\epsilon\right)$$

  - 3) Calculation of a novelty function:

$$\Delta(n)=\sum_{\mu=0}^{M-1}\left|X_{\log}(e^{j\Omega_\mu},n)-X_{\log}(e^{j\Omega_\mu},n-1)\right|_{\geq 0}$$

with: $|x|_{\geq 0}=\begin{cases} x & : \quad \text{if } x\geq 0 \\ 0 & : \quad \text{else}\end{cases}$

=> Coherent summation of increasing signal slopes.

Beat tracking and detection

- **Several step procedure:**
  - 3) Novelty function:

$$\Delta(n) = \sum_{\mu=0}^{M-1} |X_{\log}(e^{j\Omega_\mu}, n) - X_{\log}(e^{j\Omega_\mu}, n-1)|_{\geq 0}$$

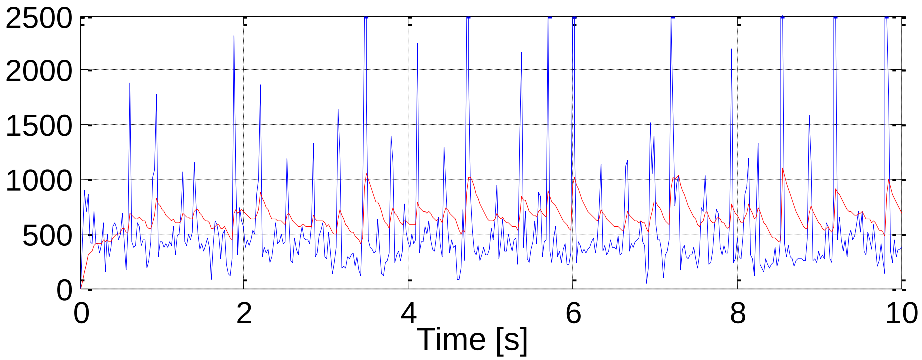
Taking the properties of the harmonic beats into account:
Periodic power increase for a wide frequency range.
=> coherent summation of all frequency components with a power increase.

  - 4) Detect peaks by subtracting the mean and taking only positive values:

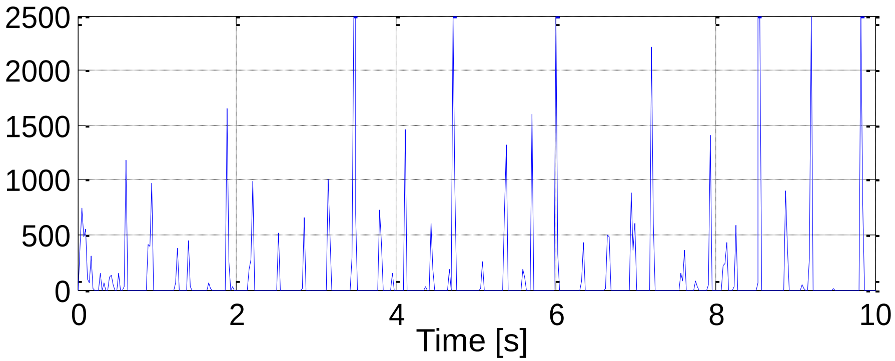
$$\Delta_{\text{sub}}(n) = |\Delta(n) - \overline{\Delta(n)}|_{\geq 0}$$

Sound example: 🔊

Blue: Novelty function:$\Delta(n)$;Red: short-term mean:$\overline{\Delta(n)}$



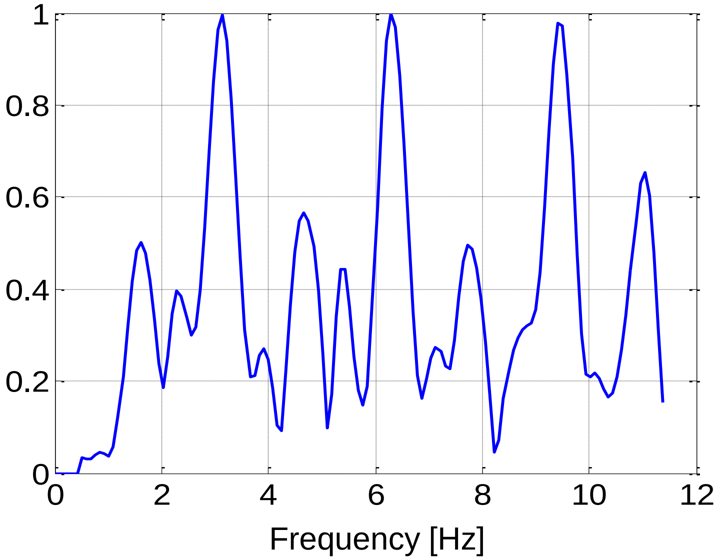
Novelty function – short-term mean: $\Delta_{\text{sub}}(n)$



Beat detection

- **Several step procedure:**
  - 5) Detection of periodicals of the novelty function: $\Delta_{\text{sub}}(n) = |\Delta(n) - \overline{\Delta(n)}|_{\geq 0}$
    - Frequency analysis of the novelty function:

$$\mathcal{F}_\Delta(e^{j\Omega_\mu}, n) = \sum_{k=0}^{N-1} \Delta_{\text{sub}}(n-k)\, w_{\text{hann}}(k)\, e^{-j\frac{2\pi}{N}k\nu}$$



First peak at 3.15 Hz => 190 Beats / min.

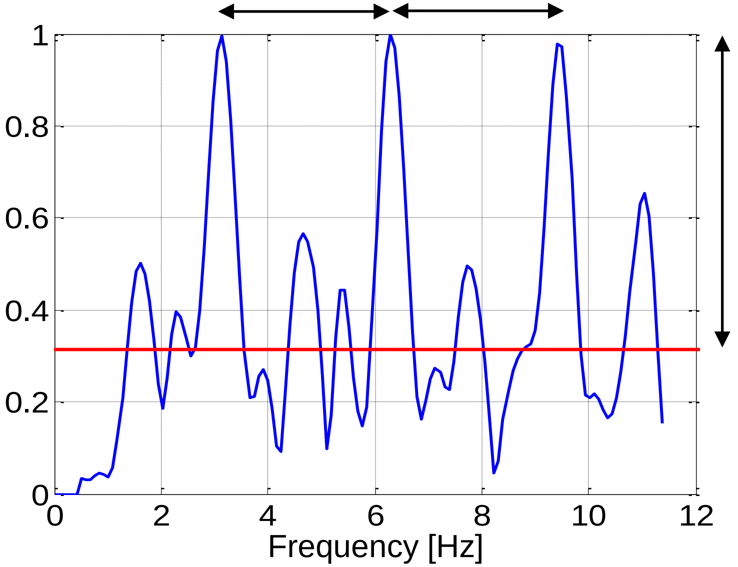
Other peaks are harmonics

Beat detection

- **Several step procedure:**
  - 6) Analysis of the spectrum of the novelty function:

    Detection based on the detected maxima of the spectrum:
    In case large maxima are observed in a periodic order

    => A beat is present.



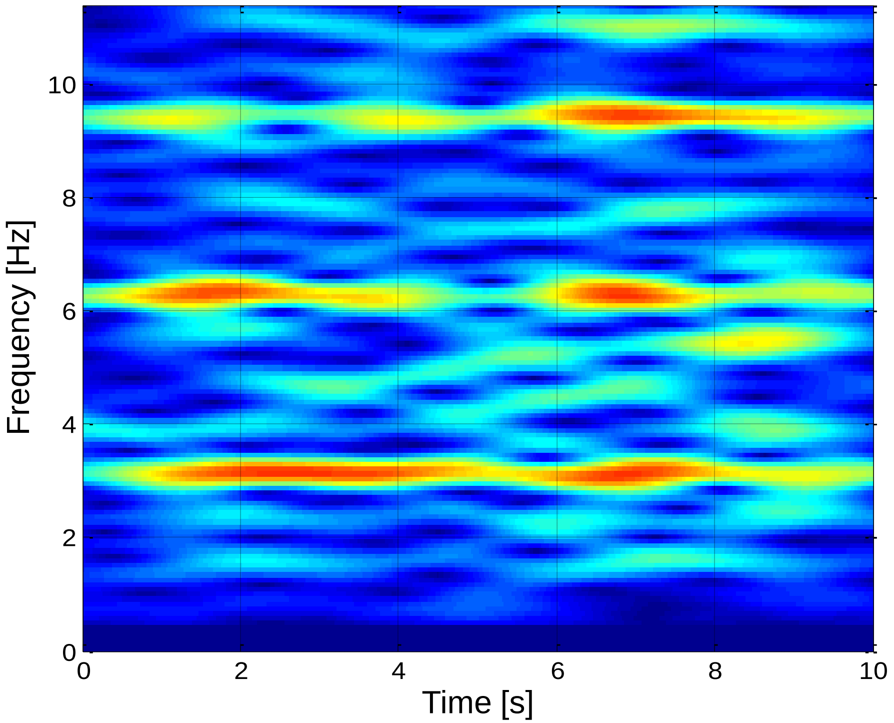
2) Periodic order of the peaks

1) High peak – mean ratio

Beat detection

- **Several step procedure:**
  - 6) Analysis of the spectrum of the novelty function: $\mathcal{F}_\Delta(e^{j\Omega_\mu}, n)$

    Plot $\mathcal{F}_\Delta(e^{j\Omega_\mu}, n)$ as spectrogram:



Beat tracking

- For the TRACKING: **Assumption:**
  - A beat, i.e., a music signal, is present.
  - In this application, the beat should be tracked as best as possible.

  - Based on the frequency analysis of the novelty function:

$$\mathcal{F}_\Delta(e^{j\Omega_\mu}, n) = \sum_{k=0}^{N-1} \Delta_{\text{sub}}(n-k)\, w_{\text{hann}}(k)\, e^{-j\frac{2\pi}{N}k\nu}$$

  - A kernel function is calculated:

$$\kappa_k(n) = w_{\text{hann}}(n-k)\, \cos(2\pi(\Omega_{\max,n}n - \phi_{\max,n}))$$

with: $$\Omega_{\max,n} = \underset{\Omega\in\Omega_\mu}{\arg\max} |\mathcal{F}_\Delta(e^{j\Omega_\mu}, n)|$$

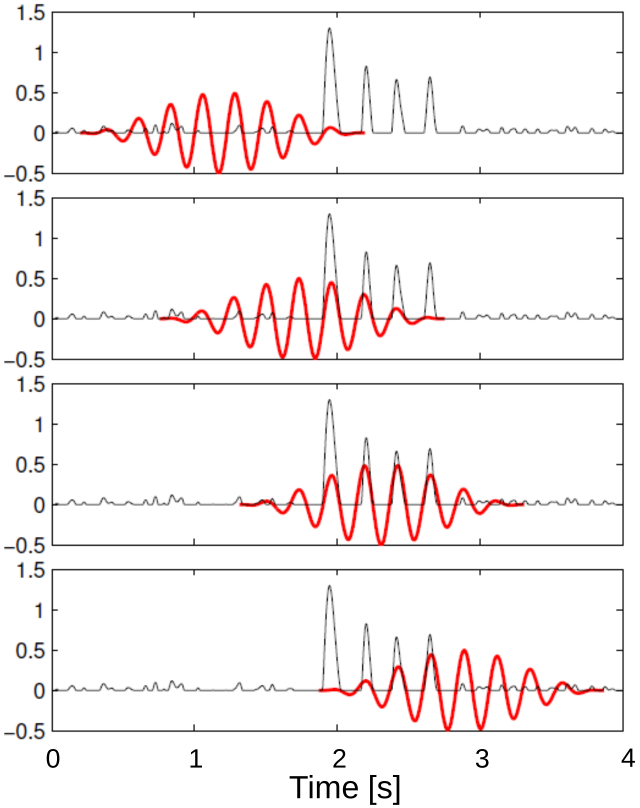
$$\phi_{\max,n} = \frac{1}{2\pi}\arccos\left(\frac{\text{Re}\{\mathcal{F}_\Delta(e^{j\Omega_{\max,n}}, n)\}}{|\mathcal{F}_\Delta(e^{j\Omega_{\max,n}}, n)|}\right)$$

Beat tracking

- A kernel function is calculated:

$$\kappa_k(n) = w_{\mathrm{hann}}(n-k)\,\cos(2\pi(\Omega_{\mathrm{max},n}n - \phi_{\mathrm{max},n}))$$

In this example kernels of length 2 sec are calculated every 0.5 sec.

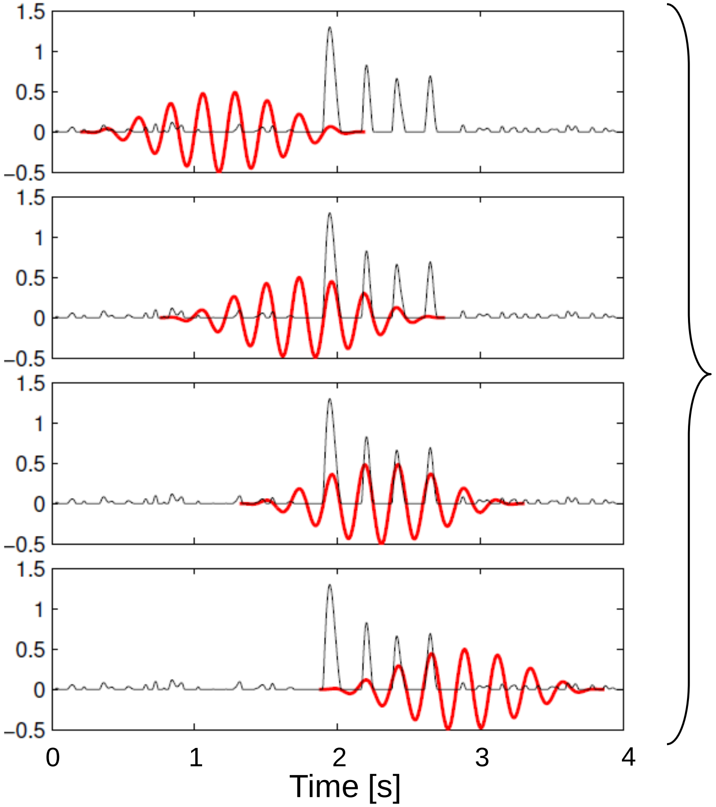


Blue: Known indicator function

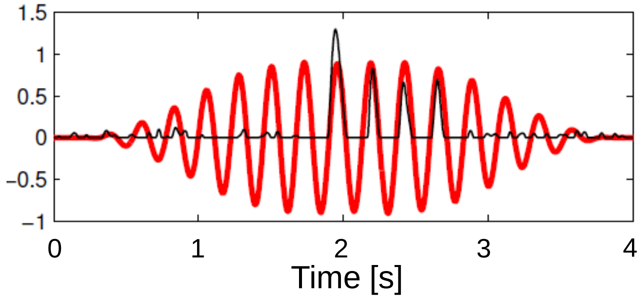
Red: Kernel functions

Beat tracking

- The kernel functions are summed and half-wave rectified in order to generate a so-called *Local Periodicity Curve (PLP):*
=> half-wave rectification of the summed kernels.



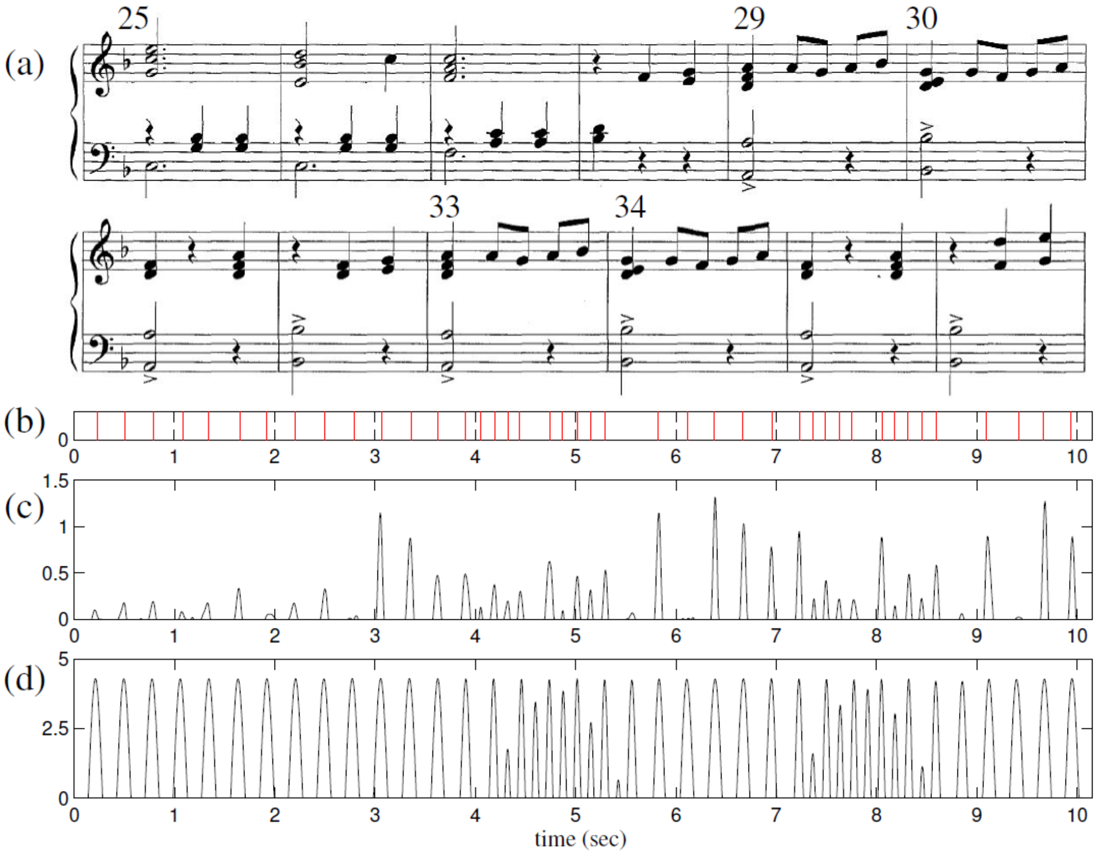
Summation:



Blue:	Known indicator function
Red:	Kernel functions

Beat tracking

- The *Predominant Local Periodicity Curve (PLP)* shows a **continuous periodicity indication**: Prominent extraction of a periodic beat indication



Known indicator function (novelty function)

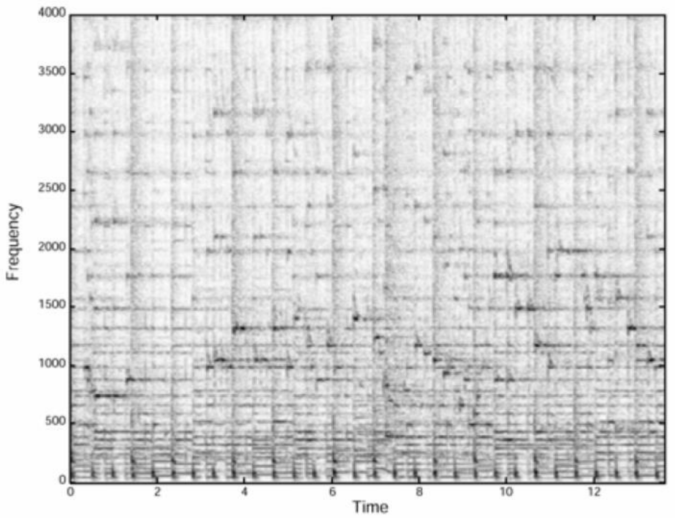
Predominant Local Periodicity Curve (PLP)

Shazam – App: Music recognition

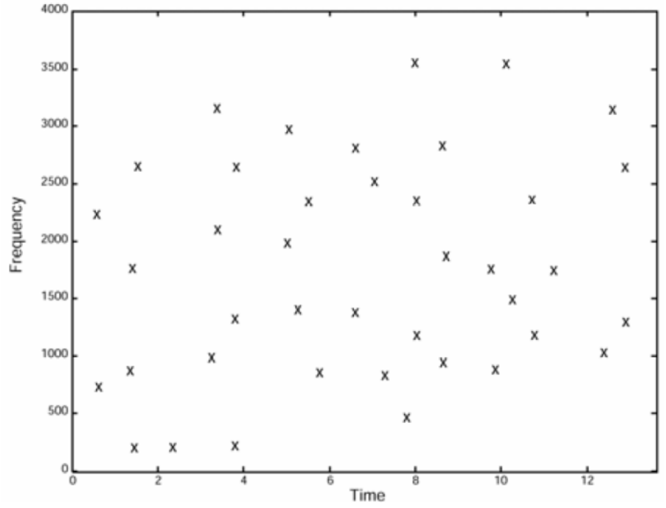
- **Target:**
  - Recognize a song based on a short sample 10-30 sec:
    - mixed with heavy noise including reverberations
    - low microphone quality
    - including codec compression disturbances
    - => quick decision, with a high reliability
  - Has to **work on original song**, not on songs played by others!

- **Approach:**
  - Generate fingerprints of a song
    - which are translation invariant,
    - show a high robustness,
    - need only low data rates for server exchange, and
    - show low computational complexity.

Constellation Maps



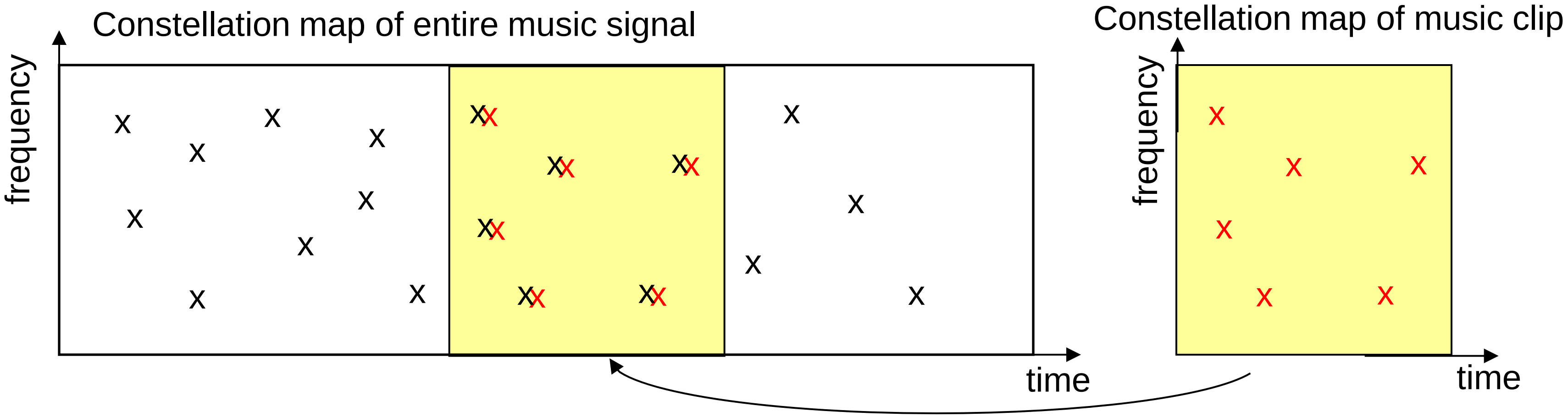
- **Spectrogram:**
  - Detect peaks in the spectrogram.
  - The peaks should have highest energy within a certain neighborhood.
  - The peaks should be sufficiently uniformly distributed.
  - The peaks show the advantage that they are typically robust with respect to noise.



- **Constellation Maps:**
  - Map of the peaks without amplitude information.
  - => reduction of data and insensitivity to power equalization

Recognition based on constellation maps

- **General procedure:**
  - Find the matching of the constellation maps of the reference song and the frame recorded by the search procedure.

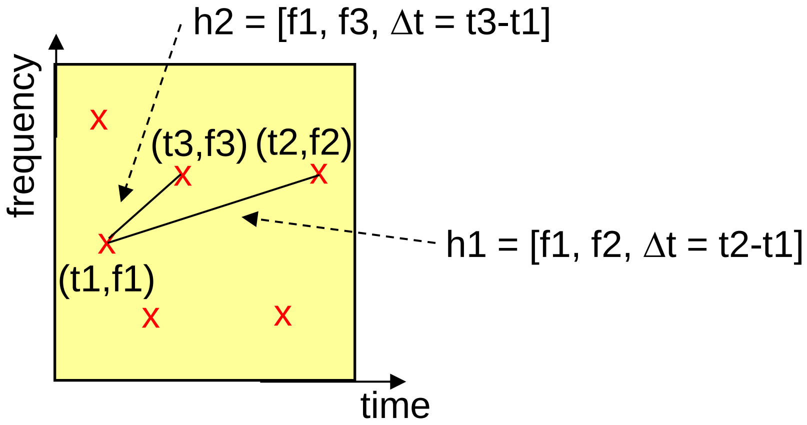

Constellation map of entire music signal
frequency
time
Constellation map of music clip
frequency
time

  - Procedure is robust against
    - added points, e.g., due to noise and
    - deleted points, e.g., due to compression
  - Disadvantage: Computationally very demanding, since matching has to be performed against a huge data base.

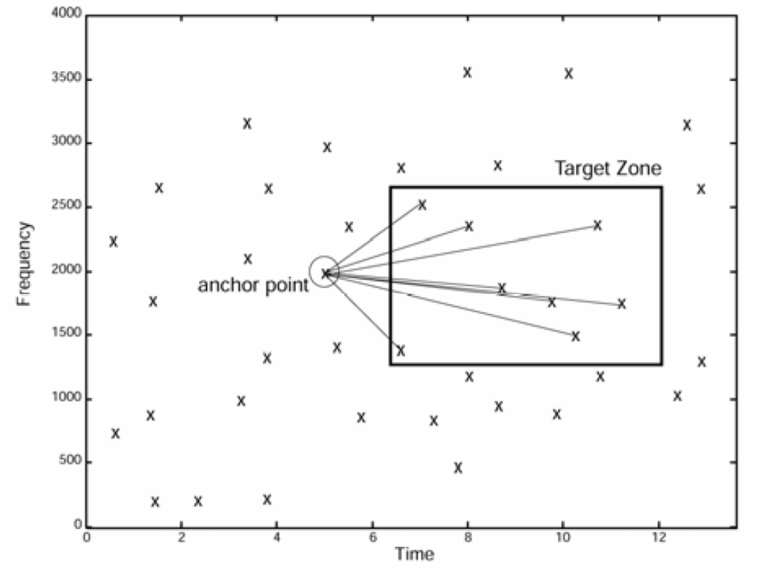
Matching based on fast combinatorial hashing

- **Hashing:**
  - Fast computational hashing is performed.
  - A hash is an identifier calculated between two constellation points.
  - A hash contains the three values: the two frequency values of the constellation points and the time difference: **h = [f1, f2, Δt]**
  - A hash is stored with the time information relative to the start of the clip:

    Hash:time => h:t1



Determination of hashes

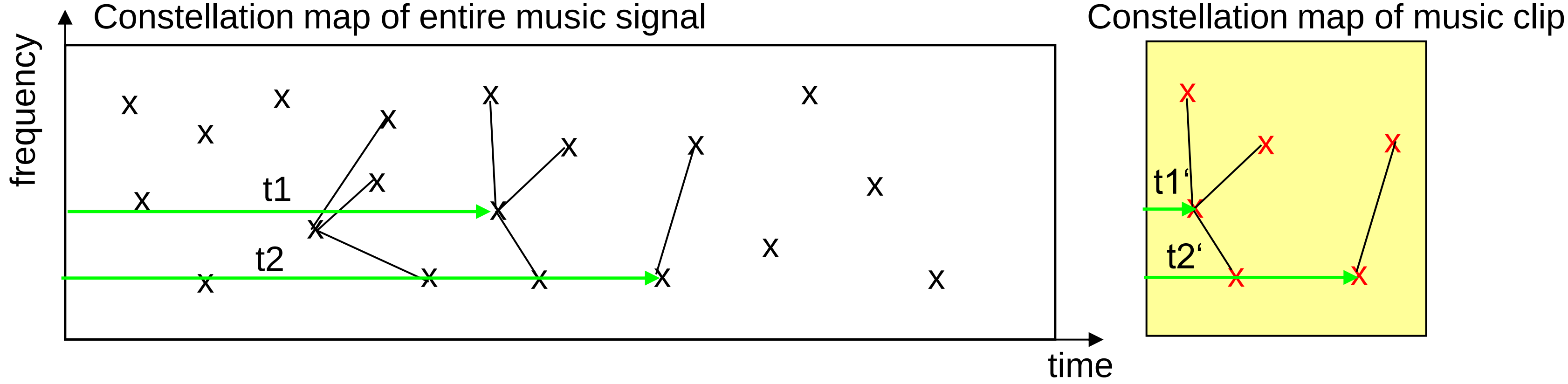


- **Hashes:**
  - Anchor points and target zones are defined
  - Hashes are calculated within the respective target zones.

- **Detection of clip in the music signal:**
  - Hashes of the entire music signal and the clip are compared.
  - The sample times of the corresponding hashes are compared.

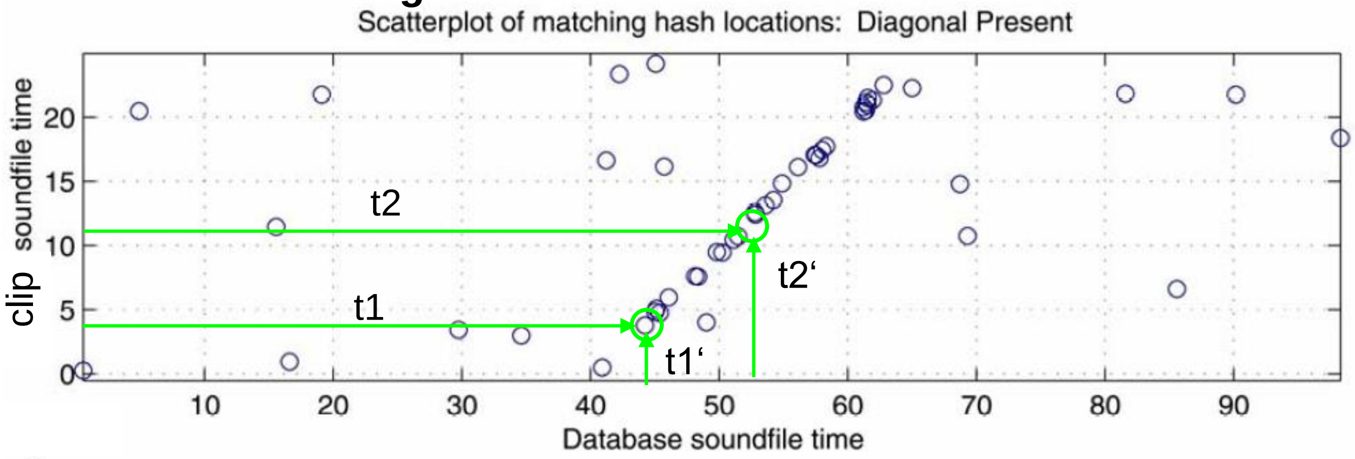
Hash matching

- **Comparison of the time values of corresponding hashes:**



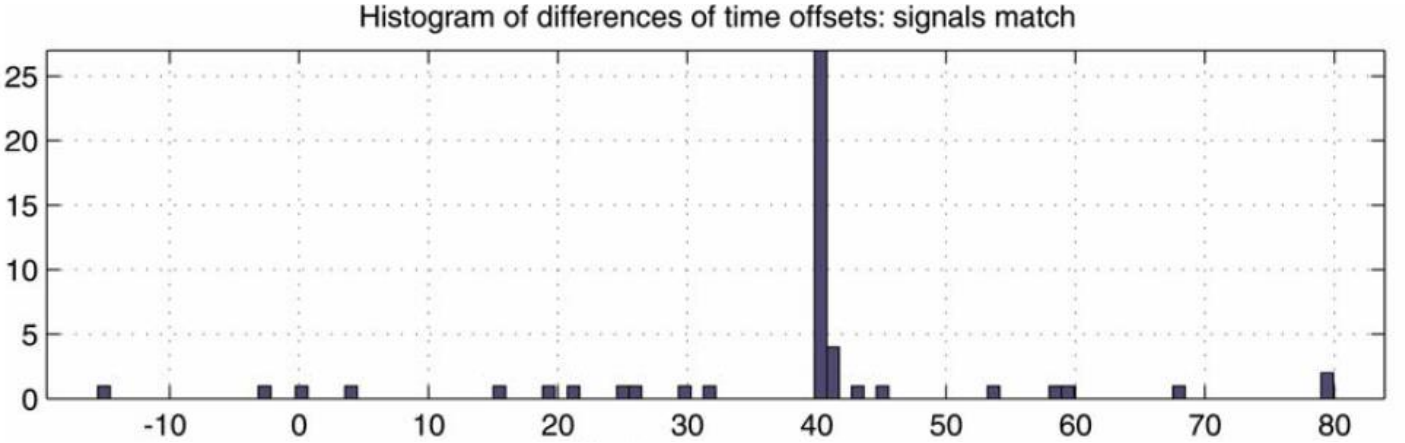
Constellation map of entire music signal
frequency
time
t1
t2
Constellation map of music clip
t1'
t2'

- **Plot time values of the clip with respect to the time values of the music soundfile => fitted diagonal**



Hash matching

- **Calculate the histograms of the difference of the offset times of matched hashes:**



Difference of the offset times of matched hashes: ti – ti‘

- **Detection of the histogram peak.**
- **Scoring:**
  - Score of a match: number of points in the histogram peak
  - Score level threshold: adjust according to desired false positive rate

Performance / Robustness

- **Robustness:**
  - Performs well in presence of noise (voices, traffic noise, even other music)
  - Performs well even if subject to non-linear distortion
  - Significant match for a corrupted 15 sec sample by only about 1–2% of the hashes
  - Scatterplot histogramming technique allows discontinuities
  - Immunity to network dropouts and masking by interference
  - => Good performance only for song versions contained in the data base!

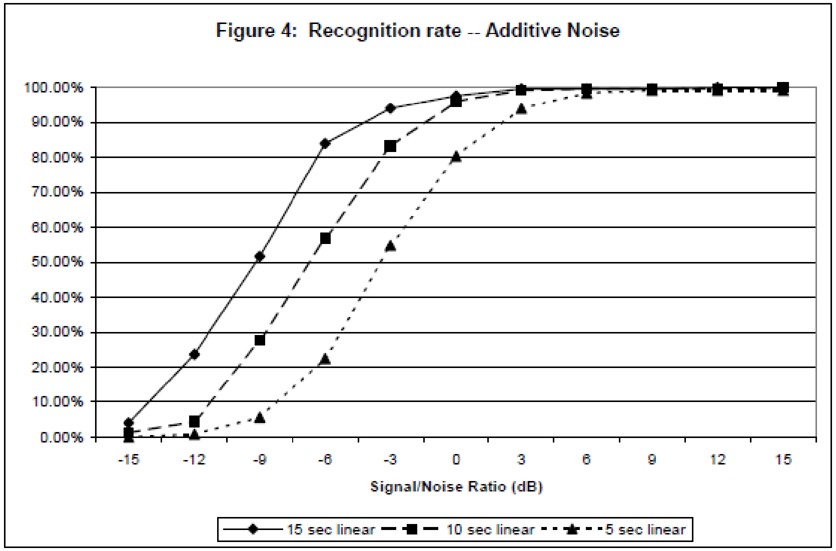


Figure 4: Recognition rate -- Additive Noise

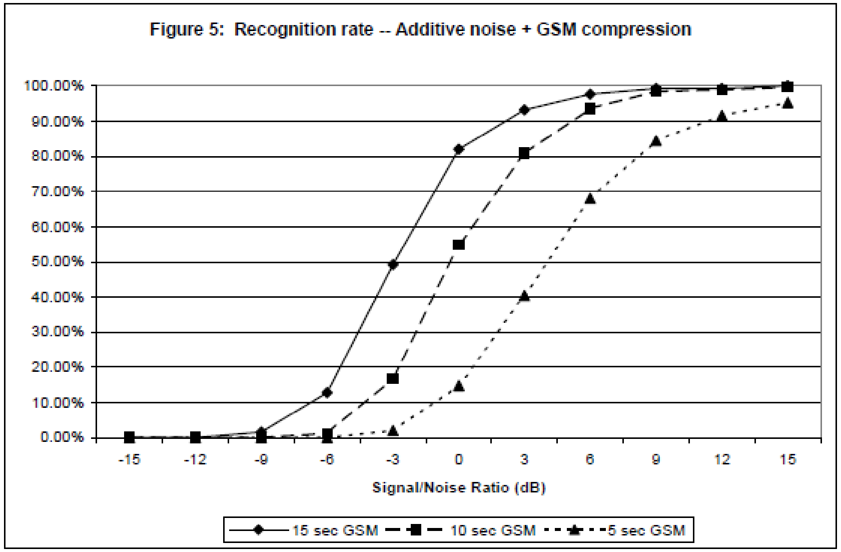


Figure 5: Recognition rate -- Additive noise + GSM compression

Summary

- Chorma based processing:
  - Tone pitch analysis and relation to chroma features.
  - Normalization and log chroma features.
  - Chromagrams.
  - Application: audio matching procedures: retrieval of audio queries.
    - Target: Insensitivity to dynamics, timbre, articulation, and tempo.

- Beat processing:
  - Beat detection and beat tracking including applications.
  - Calculation procedures.

- Shazam-App:
  - Procedure for detecting music signals

References

Chroma based processing:

[1] M. Müller: Chroma toolbox: *Matlab implementations for extracting variants of Chroma-Based Audio Features*, Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), pp. 215-220, 2011.

[2] M. Müller, F. Kurth, and M. Clausen: *Audio matching via chroma-based statistical features*, Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR), pp. 288-295, 2005.

Beat processing:

[3] P. Grosche and M. Müller: *A mid-level representation for capturing dominant tempo and pulse information in music recordings*, Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR), Kobe, Japan, pp. 189-194, 2009.

[4] M.E.P. Davies and M.D. Plumbley: Context-Dependent Beat Tracking of Musical Audio, IEEE Transactions on Audio, Speech, and Language Processing, vol.15, no.3, March 2007.

Shazam App:

[5] A. L. Wang: An industrial-strength audio search algorithm. In ISMIR 2003, 4th Symposium Conference on Music Information Retrieval, pages 7–13, 2003