Music Information Retrieval

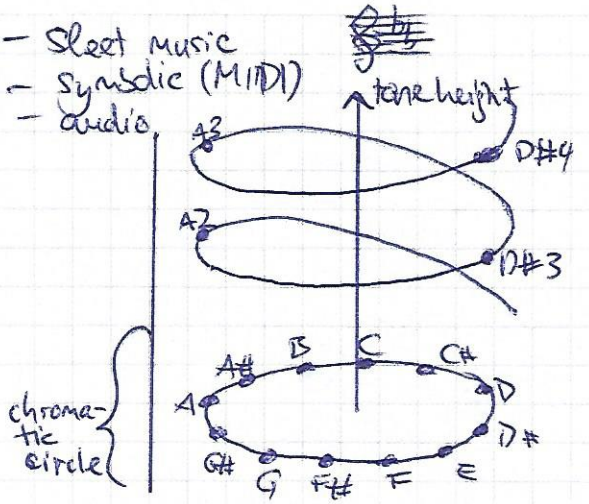
Exam preparation

Chapter 1. Music Representations

- Sheet music
- Symbolic (MIDI)
- Audio

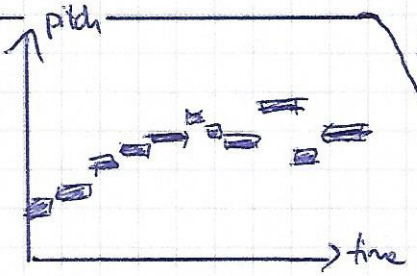
1.1. Sheet Music Representations

- pitches map into a pitch class
- an octave doubles or halves the frequency
- here: twelve-tone equal-tempered scale
- Scientific Pitch Notation:
  A4 ≙ 440 Hz
- chroma = "color" of a pitch, pitch class

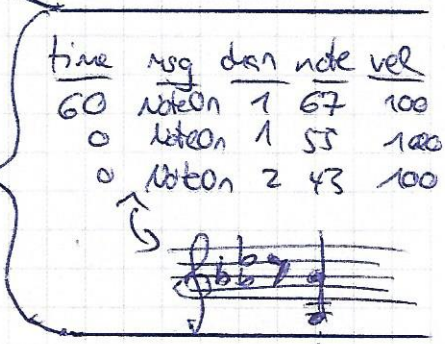


1.2. Symbolic Representations

- piano roll / player piano
  explicit musical meaning and can be passed to a computer



Shepard's helix of pitch (cyclic group $\mathbb{Z}/12\mathbb{Z}$)



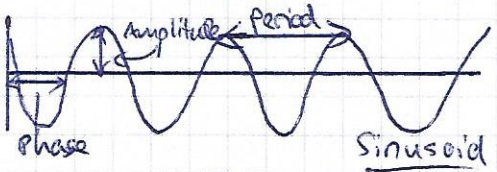
time	msg	dan	nde	vel
60	Note on	1	67	100
0	Note on	1	55	100
0	Note on	2	43	100

- MIDI = Musical Instrument Digital Interface
  ↝ message-based file format
- MusicXML / MuseScore / ... ↝ score representation

1.3. Audio Representation

music as acoustic waves (waveforms)

- period = cycle length
- frequency ≙ $\frac{1}{\text{period}}$ (Hz)
- amplitude = peak mean deviation
- phase = x s.t. $\sin(x) = 0$



- audible frequency range: 20 Hz – 20 kHz
- equal-tempered scale: $\boxed{F(p) = 2^{\frac{p-69}{12}} \cdot 440\ \text{Hz}}$

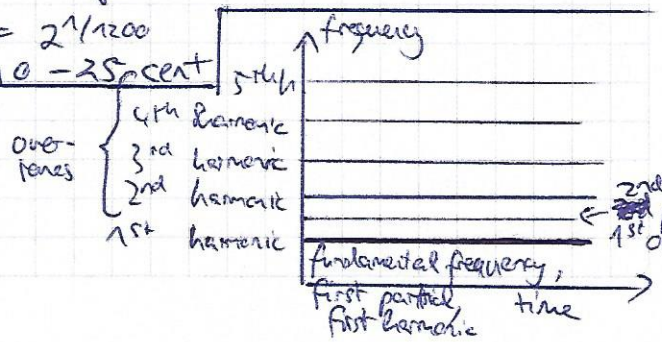
in particular $F(p+12) = 2F(p)$

$$\frac{F(p+1)}{F(p)} = 2^{1/12} = \sqrt[12]{2} \approx 1.059 = 100 \text{ cent}$$

i.e. $1 \text{ cent} = 2^{1/1200}$

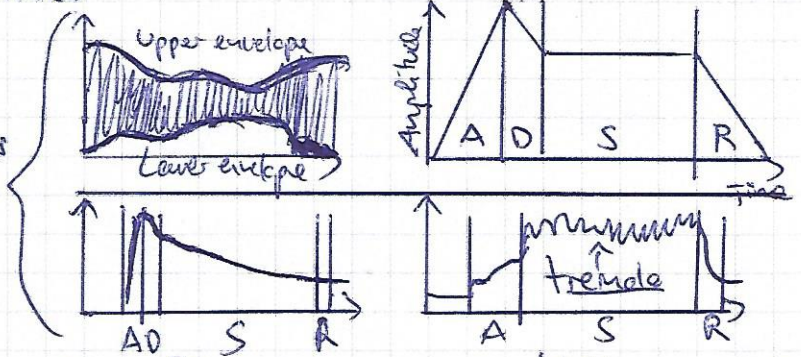
↝ just noticeable difference: 10 – 25 cent

- partials (overtones/harmonics) are closely related to timbre

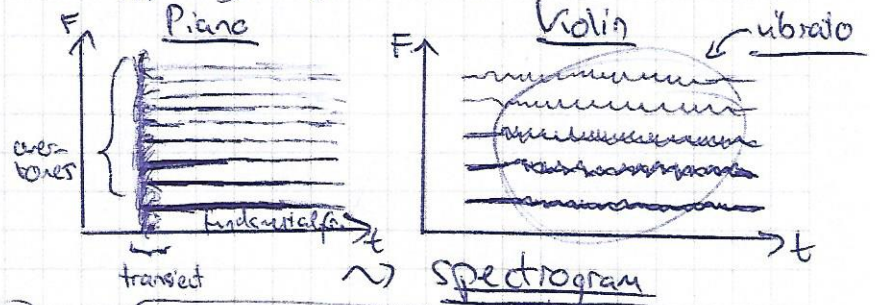


- pitch perception is logarithmic: $F(p) = 2^{\frac{p-69}{12}} \cdot 440\text{Hz}$
- loudness perception is logarithmic as well: $dB(I) = 10 \cdot \log_{10} \frac{I}{I_{TOH}}$
  $I_{TOH} = 10^{-12}\,W/m^2$. ← threshold of hearing
  $I_{TOP} = 10\,W/m^2$. ← threshold of pain
  ↝ $dB(2I) \approx dB(I) + 3$ ↝ double intensity ≙ $+3\,dB$
  ↝ phon = perceived loudness

- dynamics & timbre:
  ADSR model
  - **A**ttack – noise, transients
  - **D**ecay
  - **S**ustain
  - **R**elease

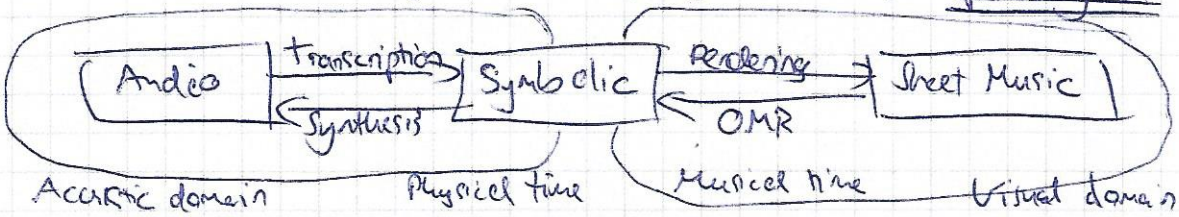


- variations in amplitude/frequency:
  - tremolo ≙ amplitude modulation
  - vibrato ≙ frequency modulation



↝ spectrogram

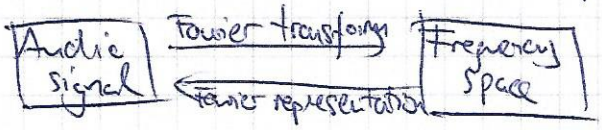
- OMR = Optical Music Recognition



Chapter 2. Fourier Analysis of Signals

sinusoid := $A \sin(2\pi(\omega t - \varphi))$ (amplitude, frequency, phase; alt. $\sqrt{2}\cos$)

idea: - compare signal with sinusoids of various frequencies $\omega \in \mathbb{R}$
- for each $\omega \in \mathbb{R}$, we get a magnitude coefficient $d_\omega \in \mathbb{R}_{\geq 0}$ and a phase coefficient $\varphi_\omega \in \mathbb{R}$



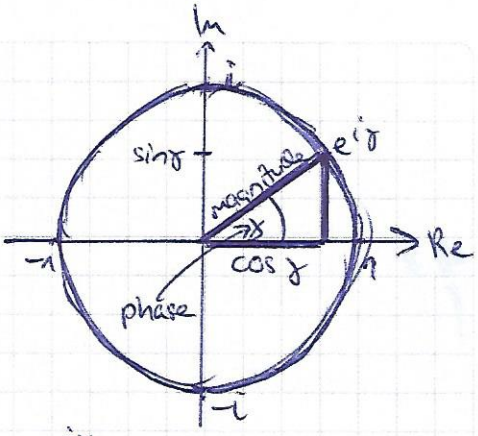
How to compare two signals f, g:
$\int_{t \in \mathbb{R}} f(t) \cdot g(t)\,dt$

"Fourier transform":
$d_\omega := \max_{\varphi \in [0,1)} \Phi_{\omega,\varphi}$
$\varphi_\omega := \operatorname{argmax}_{\varphi \in [0,1)} \Phi_{\omega,\varphi}$
where $\Phi_{\omega,\varphi} := \int_{t \in \mathbb{R}} f(t)\sqrt{2}\cos(2\pi(\omega t - \varphi))\,dt$

↑ comparison of f with a sinusoid

This can be simplified by introducing complex numbers:

$$c_\omega := \underbrace{\frac{d_\omega}{\sqrt{2}}}_{\text{magnitude} \hat{=} \text{norm of } c_\omega} \cdot e^{2\pi i(-\underbrace{\varphi_\omega}_{\text{phase} \hat{=} \text{angle } \gamma})} \in \mathbb{C}$$



$e^{i\gamma} := \cos\gamma + i \sin\gamma$

- Fourier transform of f: $\hat{f}(\omega) := c_\omega$

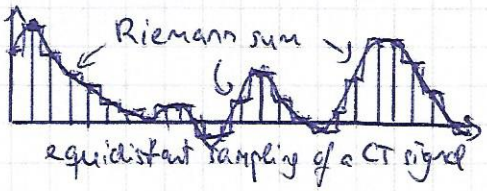
A main result in Fourier analysis is that

$$\hat{f}(\omega) = \int_{t \in \mathbb{R}} f(t) \cdot e^{-2\pi i \omega t} \, dt$$

(a weighted superposition then yields the Fourier representation)

$\rightsquigarrow$ continuous-time (CT) signal

approximated as



equidistant sampling of a CT signal

Discrete Fourier Transform (DFT)

- $F_S := 1/T$ (Hz)

sampling rate, sampling period

- Nyquist theorem: To reconstruct frequency Ω, the sampling rate must be at least $F_S \geq 2 \cdot \Omega$.

$\Rightarrow$ let $\Omega := F_S/2$, then Ω is the Nyquist frequency and the signal is Ω-limited

$\rightarrow$ often, $F_S = 44{,}1$ kHz as $\Omega = 22$ kHz > audible frequency range

- to calculate the Fourier transform, we take a finite amount of samples N and a finite amount of frequency bins M.
- set $N = M$ to use the Fast Fourier Transform algorithm.

$\rightsquigarrow$ $$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-2\pi i k n / N}$$ and $$F(k) := \frac{k}{N \cdot T} = \frac{k \cdot F_S}{N}$$

$\rightarrow$ discrete-time (DT) signal $\rightarrow$ there are as many bins as samples (N)

only consider the lower bins! $\Rightarrow$ $F(N) = F_S$, $F(N/2) = F_S/2 = \Omega$

Example: $F_S = 44{,}1$ kHz, then we have 22050 relevant bins (each 1 Hz wide)

Suppose N is a power of two.
In that case, the FFT algorithm can compute the DFT in $O(N \log_2 N)$ instead of $O(N^2)$.
$\Rightarrow$ Always sample with powers of two (window size)!

Short-Time Fourier Transform

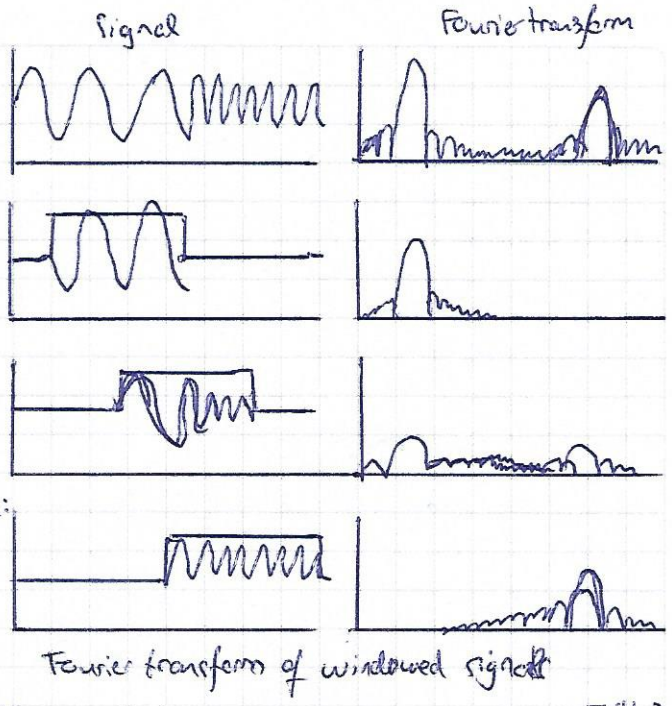
Problem: DFT hides time information completely
Idea: consider small "windows"

Let $N \in \mathbb{N}$ be the window size,
$H \in \mathbb{N}$ the hop size and
$w : [0, N-1] \to \mathbb{R}$ a window function

$\rightsquigarrow$ The discrete STFT of the signal x with the k^{th} Fourier coefficient at m^{th} time step:

$$\mathcal{X}(m,k) := \sum_{n=0}^{N-1} x(n+mH) \cdot w(n) \cdot e^{\frac{-2\pi i k n}{N}}$$

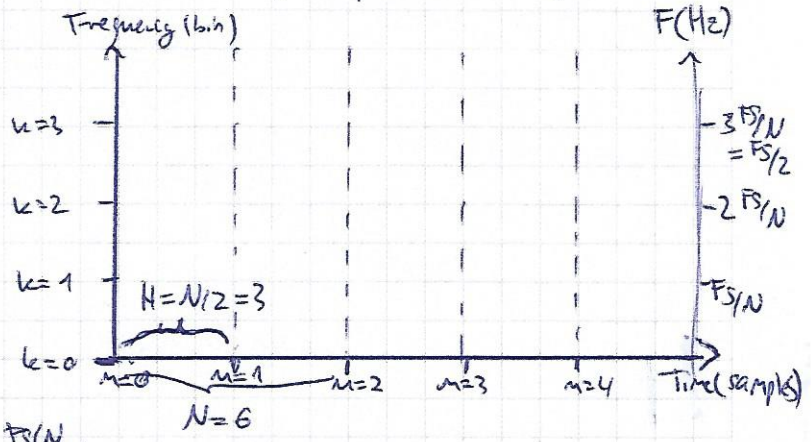
with $m \in \mathbb{Z}$, $k \in [0, N/2]$.



The meaning of m and k is as follows:

$$T(m) := \frac{m \cdot H}{F_S} \qquad F(k) := \frac{k \cdot F_S}{N}$$

Let F_S be the sample rate and N the window size.
Then there are $N/2$ frequency bins, each F_S/N Hz apart.
(From 0 to equally spaced up to the Nyquist frequency.)



Tradeoff:
good time resolution ($\leftarrow N$) = bad frequency resolution and vice-versa. (independent of H!)
- window size N proportional to the number of bins $N/2$
- window size N antiproportional to width of bins $F_S/N = \frac{2}{\#\text{bins}} = \frac{F_S/2}{N/2}$ ($\#$bins for 2 Hz!)

Example: $N = 4096$ $H = 2048$ $F_S = 44.1$ kHz
$\#$bins: 2048 (2049) — bad resolution for lower pitches!
width of bins: $\approx 10{,}7$ Hz
Suppose there are 40 frames.
Then the signal is $T(40) = 40 \cdot \frac{2048}{44.1} \approx 1{,}86$ seconds long.

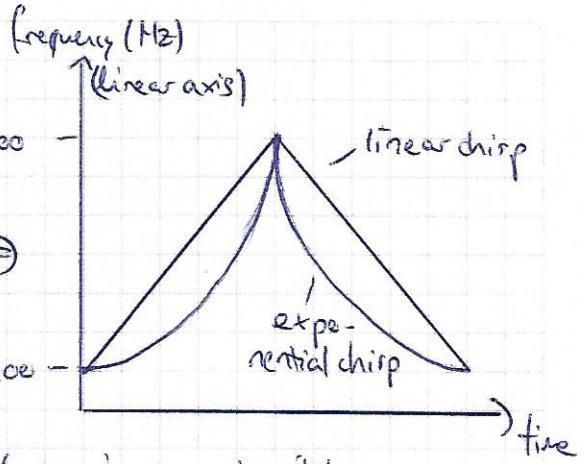
Chapter 3. Music Synchronisation

↳ a procedure which, for a given position in one representation of a piece of music, determines the corresponding position in another representation (or interpretation).

Two steps: (mid-level)
- Feature representation (e.g. Chroma, MFCC, ...)
- Dynamic time warping (DTW) to align sequences

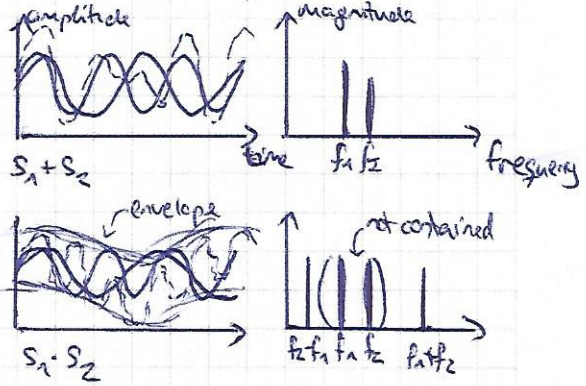
Exercise 1/2

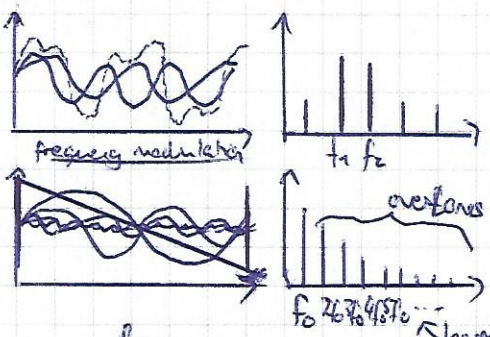
- a chirp is a signal continuously increasing in frequency.
  - linear: the function that describes the frequency is linear: $(f_0 + k/2 \cdot t) \cdot t$ ⟹
  - exponential: the derivative is exponential



The exponential chirp starts off slower, but gets much faster towards the end. Because of the logarithmic perception of pitch, an exponential chirp may be perceived as a more consistent/even increase in pitch.

- there are several ways to combine signals:
  Let s_1, s_2 be pure sinusoids with frequency f_1, f_2 respectively. The signal
  - $s_1 + s_2$ is the superposition of both pitches (polyphony) (contains f_1, f_2)
  - $s_1 \cdot s_2$ performs ring modulation ($f_1 - f_2$, $f_1 + f_2$), which changes the timbre
  - $\sin(f_1 \cdot 2\pi t + \frac{A}{f_2} s_2(t))$ performs a frequency modulation of f_2 into f_1



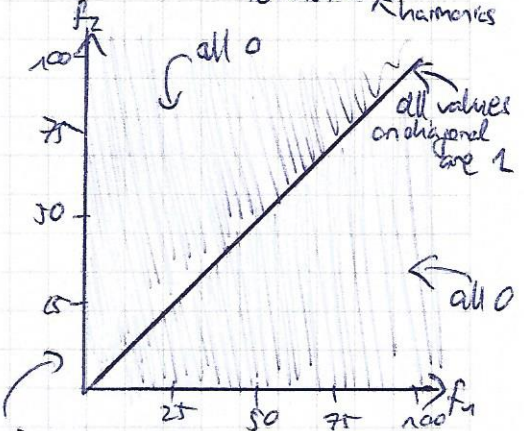


- more complex combinations are possible, like a sawtooth wave: (fundamental frequency f_0)

$$s(x) = \sum_{i=1}^{\infty} \frac{1}{i} \cdot \sin 2\pi i f_0 x$$

- Why can every real-valued function be decomposed into Fourier space?
  Because Fourier space has all sinusoids as an orthonormal basis. In other words,

$B = \{ \sin(2\pi\omega t) \mid \omega \in \mathbb{R} \}$ is an orthonormal basis for the ∞-dimensional Fourierspace, that is:
  - B spans the space of real-valued functions
  - B is minimal in doing so
  - two sinusoids $s_1 \neq s_2$ are orthogonal:
    $\langle s_1, s_2 \rangle = \sum_t s_1(t) \cdot s_2(t) = 0$
  - all sinusoids are normalized, i.e., $\|s\| = 1$.

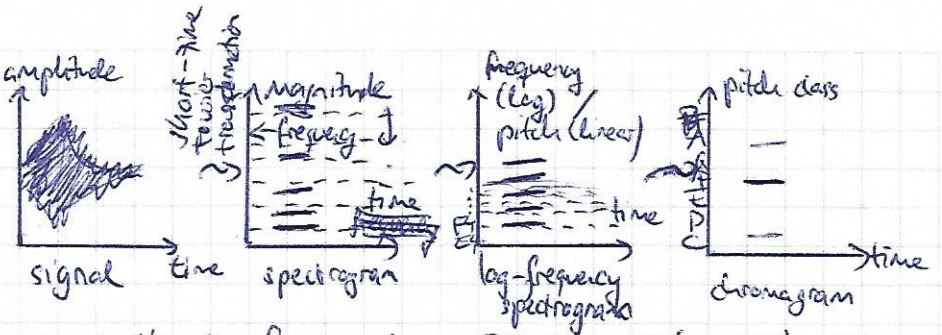


This can be visualized by plotting the inner product of different sinusoids.

⟹ The Fourier transform is just a change of basis in the space of real-valued functions.

3.1 Chromagrams

- calculate the STFT
- do a logarithmic compression to have an axis linear in pitch, this uses the bandwidth of a frequency bin: $BW(p) = F(p+0.5) - F(p-0.5)$ (cutoff frequencies) (pitch ± 50 cent)



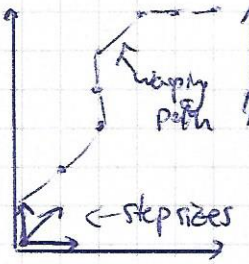
3.2 Dynamic Time Warping

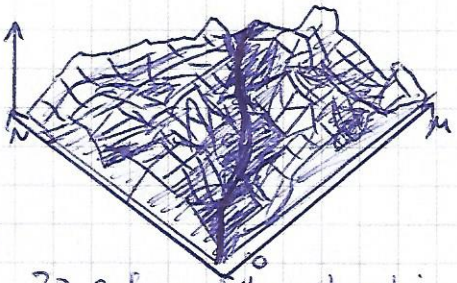
Idea: Compare two time-dependent sequences to find a good local alignment.

Given two sequences of feature vectors, calculate a cost matrix C that contains the cosine similarities between all feature vectors.

A warping path is a path from the bottom left to the top right with specific allowed step sizes.
- monotonicity: no going backwards
- boundary condition: start/end point

It accumulates the cost of all passed cells.
We aim to minimize the cost.
DTW distance := cost of optimal warping path





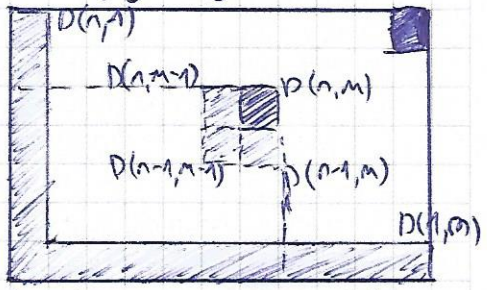
3D surface of the cost matrix with dynamic warping path

As the number of warping paths is exponential, we use dynamic programming instead ($O(NM)$).
We calculate the accumulated cost matrix D as follows:

DTW algorithm

$$D(n,1) = \sum_{k=1}^{n} C(k,1) \quad *$$
$$D(1,m) = \sum_{k=1}^{m} C(1,k) \quad *$$
$$D(n,m) = C(n,m) + \min \begin{cases} D(n-1,m-1) \\ D(n-1,m) \\ D(n,m-1) \end{cases}$$

→ depends on step sizes

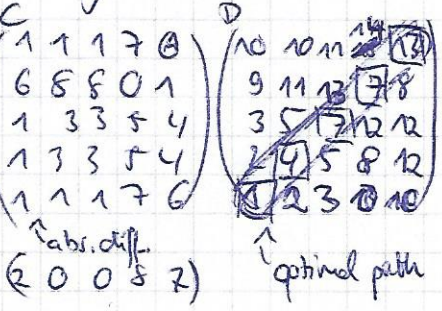


↳ does not guarantee an optimal path!

Each cell is the DTW distance for a subsequence of the given sequences.
To determine the optimal path, backtrack from the upper right corner by choosing the minimum of all possible origin cells.

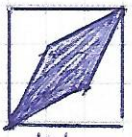
* alternatively, set $D(n,0) = D(0,m) = +\infty$

Step sizes can be changed to account for too large temporal deformations (e.g. →↑):



Variations of DTW: - step sizes to avoid temporal deformation
- local weights to favor diagonal movement
- global constraints to filter "bad" paths ⇒


Sakoe-Chiba band


Itakura parallelogram

Note that an optimal path does not necessarily exist (esp. for sequences of different length).

Multi-scale DTW by down-sampling

3.3 Applications

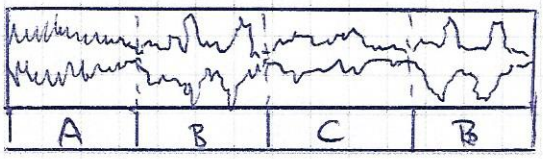
↗ robustness vs. expressiveness of features

Chromagram: - robust to variations in timbre and dynamics
- characterize melodic, harmonic and possibly temporal progression

- multimodal music navigation: Interpretation Switcher, Score Viewer
- performance analysis with tempo curves: find commonalities & characteristics of interpretations
  ↳ calculate DTW with reference version (annotated with BPM)
- online score following: automatic accompaniment

Chapter 4 Music Structure Analysis

4.1 General Principles



Goal: Divide a given music representation into temporal segments that correspond to musical parts. (motifs, phrases, sections, ...)
↳ methods are usually repetition-, novelty- or homogeneity-based.
↗ Waveform-Novelty
↗ Recurrence-Structure

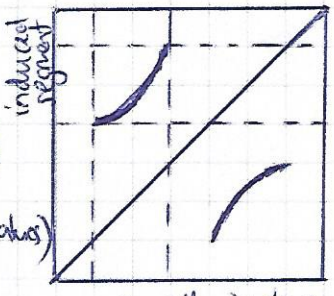
Different features are used in different contexts: eg., - Chroma - harmonics, but no timbre
- MFCC - timbre (envelope shape)
- Tempo - rhythm, but no pitch-features

Also, long window sizes in STFT can smooth out irrelevant variations.
↳ homogeneity-based segmentation

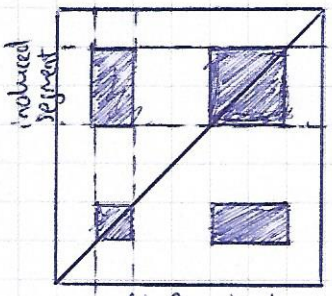
4.2 Self-Similarity Matrices

Idea: pairwise comparison of all feature vectors, yielding a SSM (self-sim. matrix).
(Comparison for example with cosine similarity)

Structures that appear are:
- blocks ≙ homogeneity (consistent features)
- paths ≙ repetition of subsequences



a path structure

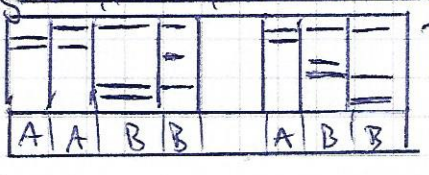


a block structure

Enhancement Strategies: - smoothing, downsampling
- path smoothing: averaging / low-pass filter in the direction of the main diagonal
- tempo-invariant smoothing, forward-backward smoothing → (assuming no tempo differences)
- transposition-invariant self-similarity matrix
- global/relative/local thresholding

$$S_{L,\Theta}(n,m) := \max_{\theta \in \Theta} \left[\frac{1}{L} \sum_{l=0}^{L-1} S(n+l_n^{\theta}, m+\lfloor l \cdot \theta \rfloor) \right]$$

$$S^{TI}(n,m) := \max_{i \in [0:M]} \left[S\left(\rho^i(\kappa_n), \kappa_m\right) \right]$$

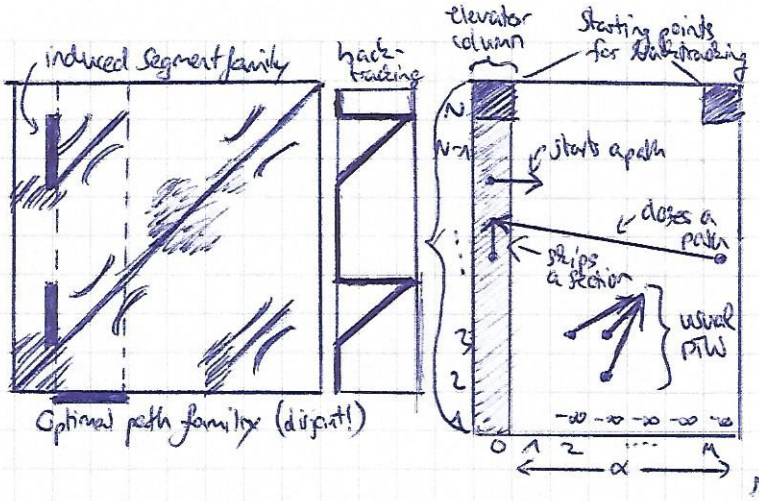


4.3 Audio Thumbnailing

Goal: obtain the most representative section of a recording (e.g., chorus).

Idea: Introduce fitness measure for a given segment that describes how well and how much a segment describes a recording; maximize the fitness.

induced segment family: nonoverlapping repetitions of a given segment



Try to align the segment with several induced segments with a modified DTW procedure. $\leq a(M,i)$

↳ differences: align the segment α multiple times, but completely (potentially not covering the whole recording and maximizing the path family score.

i.e. skip sections of X and go back to the beginning of α ('carriage return')

The accumulated cost matrix is defined by

$D(n,m) = S^{\alpha}(n,m) + \max\{P(i,j) | (i,j) \in \mathbb{C}(n,m)\}$ — step size condition as usual in DTW

$D(n,0) = \max\{D(n-1,0), D(n-1,M)\}$ — elevator column: skip segments, close up paths

$P(n,1) = D(n,0) + S^{\alpha}(n,1)$ — start a new path component coming from the elevator column

$D(1,m) = -\infty \quad (m \geq 2)$ — forces the first path to come from the elevator column

Eliminating the self-explanation, the normalized score describes how well this segment describes the segments. "precision" - favors short segments

The coverage describes how much of the recording it explains; this is the sum of all induced segments. "recall" - favors long segments

- short segments explain usually well, but not much
- long segments explain usually much, but not well

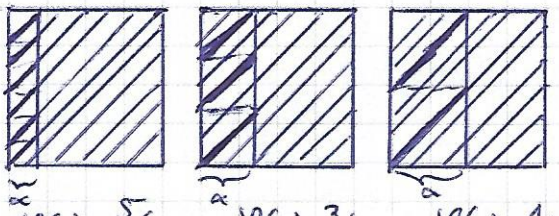
We unify this with a harmonic mean (F-measure):

$$\varphi(\alpha) := 2 \cdot \frac{\bar{\sigma}(\alpha) \cdot \bar{\rho}(\alpha)}{\bar{\sigma}(\alpha) + \bar{\rho}(\alpha)}$$

$\bar{\sigma}$: precision/score

$\bar{\rho}$: recall/coverage

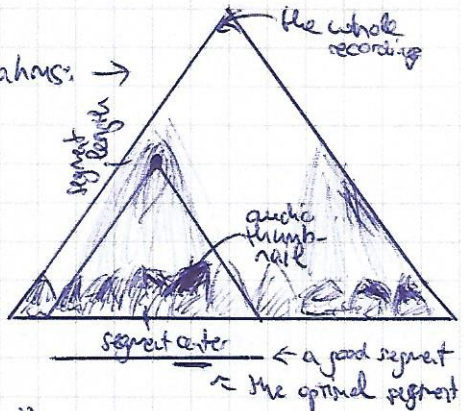
(favors shorter segments due to self-explanation)



$\varphi(\alpha) = 5/6$ $\varphi(\alpha) = 3/3$ $\varphi(\alpha) = 1/2$

Represent each segment by its length and center.

We can indicate the fitness of all segments in a scape plot. Brahms: →



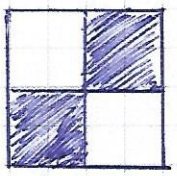
4.4 Novelty-based Segmentation

Goal: Locate musical changes in time (segment boundaries).

Novelty Detection: Slide a (Gaussian) checkerboard kernel across the diagonal of an SSM and calculate similarity.

This detects block-like structures.

→ novelty function (changes in timbre, pitch...)



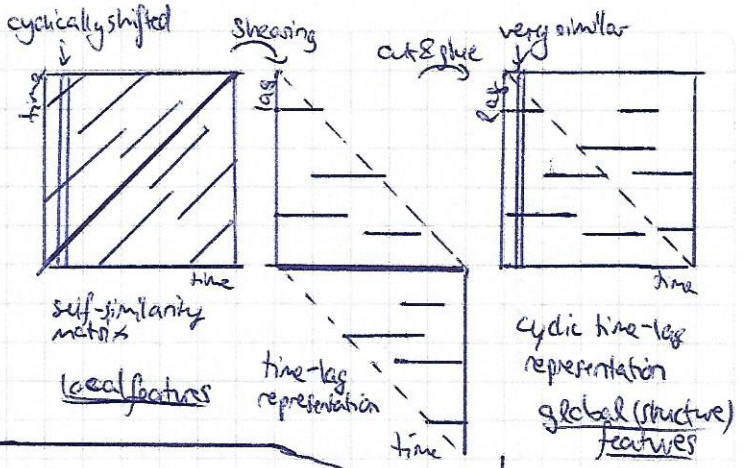
kernel (size matters!)

Structure Features (lag-based novelty)

combines local and global characteristics.
(frame-wise, entire frame)

↝ time-lag matrices
this transforms frame vectors so that structurally similar frames are no longer cyclically shifted, but almost identical.

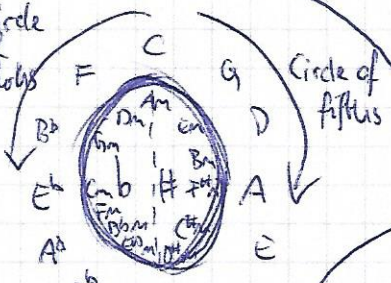
↝ new (high-level) feature representation (structure features)



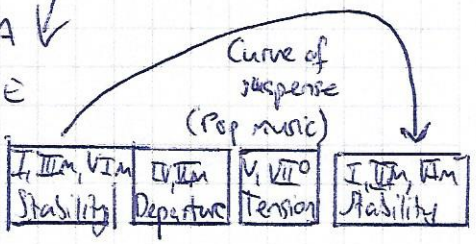
Chapter 5: Chord Recognition

5.1 Basic Theory of Harmony

Intervals, frequency ratios, chords, scales, circle of fifths / fourths



functional harmony theory models chord progressions and their likeliness.



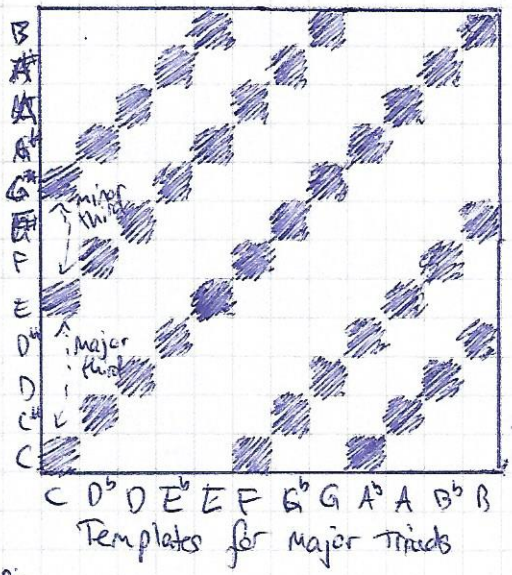
5.2 Template-based Chord Recognition

Based on chroma features, compare frames with given chord templates (cosine similarity) and find the best match. (Ignores functional harmony theory.)

$$\lambda_n := \operatorname{argmax}_{\lambda \in \Lambda} S(t_\lambda, x_n)$$

λ_n: chord of n^{th} frame; S: similarity; t_λ: template; x_n: frame

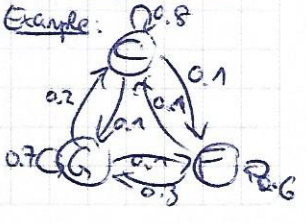
- However, there are multiple problems: some chords are very similar, no functional harmonies are regarded, there are many chord variations (e.g. Vierklang) (chord ambiguities)
  e.g.: $S(t_C, t_C) = S(t_G, t_{Em})$
- overtones & logarithmic pitches & major-minor-confusion
- tuning issues, broken chords, ...



5.3 HMM-Based Chord Recognition

takes into account the probability of harmonic progression using Hidden Markov Models.

We assume that $P(s_{n+1} = \alpha_j \mid s_n = \alpha_i, \ldots) = P(s_{n+1} = \alpha_j \mid s_n = \alpha_i)$. → "memoryless" (Markov property), Example:

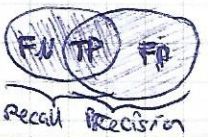


Markov chain:

$$\begin{cases} a_{ij} := P(s_{n+1} = \alpha_j \mid s_n = \alpha_i), \ \sum_{j=1}^{I} a_{ij} = 1 \\ c_i := P(s_1 = \alpha_i) \end{cases}$$

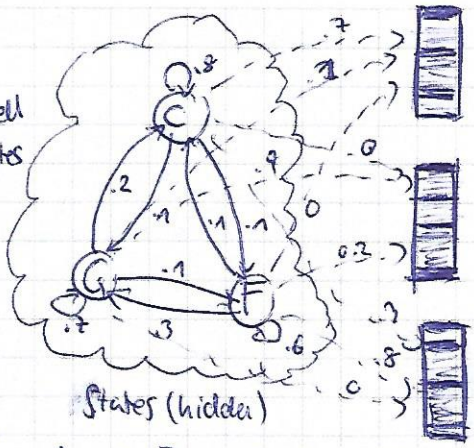
Evaluation:

$$F = \frac{2 \cdot P \cdot R}{P + R} \qquad P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN}$$



Hidden Markov Models

Goal: uncover relation between observed feature vectors and underlying chords. } Emission probabilities tell whether a chord generates a feature vector.



A discrete HMM is a tuple $\Theta := (\mathcal{A}, A, C, \mathcal{B}, B)$ with
- $\mathcal{A}$: set of states α_i
- A: state transition probabilities a_{ij}
- C: initial state probabilities c_i
- $\mathcal{B}$: set of observation symbols β_k
- B: emission probabilities b_{ik} (that state α_i emits β_k)

$A, C,$ and B are determined by experts or with learning procedures, $\mathcal{A}$ and $\mathcal{B}$ are fixed.

Problem: Uncovering the hidden state sequence
Given an observation sequence, we want to assign chord labels that best explain the observations.

Evaluation Problem: How likely is a given observation sequence $O = (o_1, \dots, o_N)$ for a given HMM Θ? (How well does it fit?)

$$P(O,S|\Theta) = c_{i_1} \cdot b_{i_1 k_1} \cdot a_{i_1 i_2} \cdot b_{i_2 k_2} \cdot \dots \cdot a_{i_{N-1} i_N} \cdot b_{i_N k_N}$$

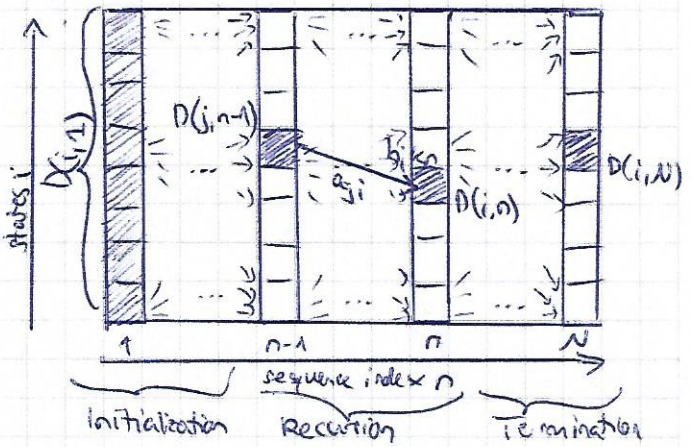
$$P(O|\Theta) = \sum_{S=(s_1,\dots,s_N)} P(O,S|\Theta)$$ – sum over all possible explaining state sequences (one of these is optimal!)

Uncovering Problem: Given Θ and O, what is the single state sequence $S = (s_1, \dots, s_N)$ that "best explains" O?

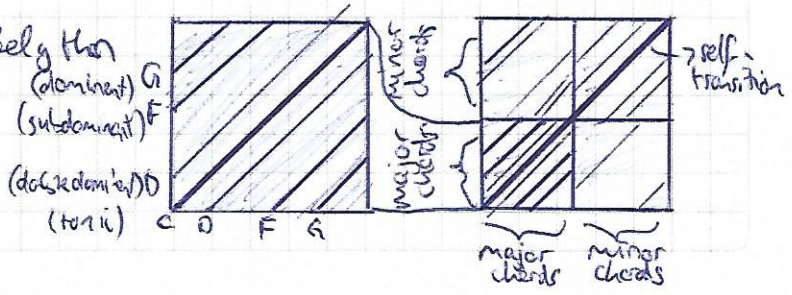
$$Prob^* = \max_{S=(s_1,\dots,s_N)} P(O,S|\Theta)$$
$$S^* = \operatorname*{argmax}_{S=(s_1,\dots,s_N)} P(O,S|\Theta)$$
} the state sequence that contributes most to $P(O|\Theta)$, the probability $Prob^*$ may be very small!

Viterbi algorithm: fill matrix similar to DTW

$$D(i,1) = c_i \cdot b_{i k_1}$$
$$D(i,n) = b_{i k_n} \cdot \max_{j \in [1:I]} \left(a_{ji} \cdot D(j, n-1) \right)$$
$$Prob^* = \max_{i \in [1:I]} D(i,N)$$



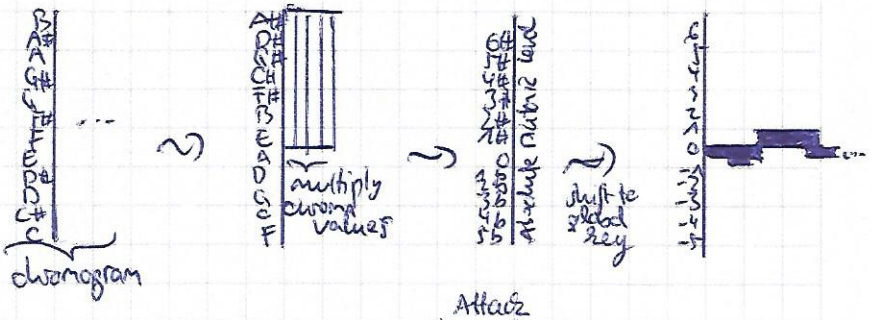
Application to Chord Recognition
$\mathcal{A} := \{C, C^\#, \dots, B, C_m, C^\#_m, \dots, B_m\}$
As for $\mathcal{B}$: Certain chord transitions are more likely than others (change from a tonic to a dominant is much more likely than transposing by one semitone).



In practice: Training – Baum-Welch algorithm,
Evaluation – Viterbi algorithm

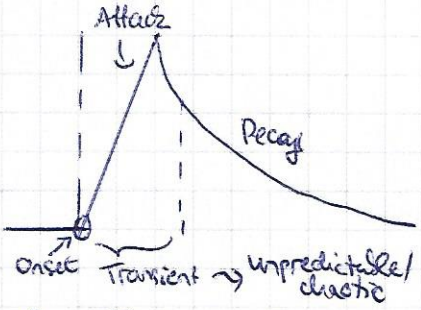
Local Key Detection

- detect modulations in key
- chroma smoothing
- reorder to perfect fifths (circle of fifths!)
- multiply chroma values
- shift to global key



Chapter 6. Tempo & Beat Tracking

Given an audio recording, determine the periodic sequence of beat positions ("tapping the foot when listening to music!").

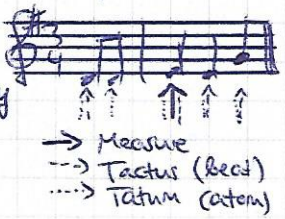


6.1 Onset Detection

onset = instant that marks the beginning of a transient
- ↳ percussive music: energy-based novelty
- ↳ non-percussive music: spectral-based approach

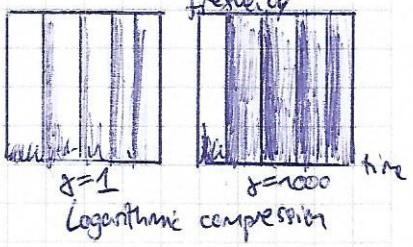
Steps:
- feature representation that reflects onsets better
- apply derivative operator to derive novelty function
- peak-picking algorithm

} see novelty in music structure analysis



Energy-based novelty:
- shift a bell-shaped function along the signal (local energy)
- take discrete derivative
- half-wave rectification (thresholding)
- (optional) logarithmic compression to account for logarithmic perception of loudness

Spectral-based novelty:
- logarithmic compression
- derivative, half-wave rectification
- accumulation into novelty function
- normalization (subtract local average)

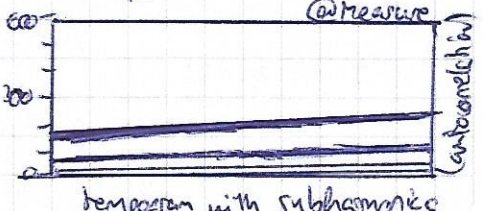


6.2 Tempo Analysis

extraction of tempo information is difficult due to tempo rubato (freely speeding up and down) and syncopation (stressing notes outside the grid of beat positions)
- beat: steady pulse that drives beat forward
- tempo: speed of the pulse

↝ strategy: analyze the novelty curve with respect to reoccurring or quasi-periodic patterns; avoid peak picking!

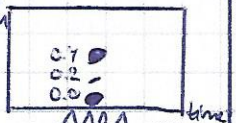
A tempogram is a time-tempo representation that encodes the local tempo of a music signal over time.

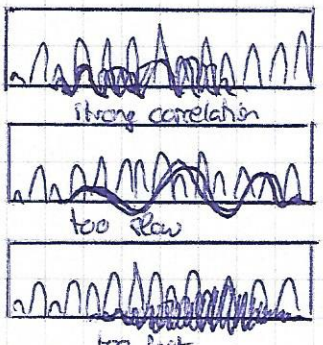


Fourier Tempogram = windowed sinusoidal →
- calculate a novelty curve as above (Δ)
- compute spectrogram (STFT) of Δ
- convert frequency axis into tempo

Autocorrelation Tempogram: ↳ sliding inner product
- compare Δ with time-lagged versions of itself
- convert lag-axis into tempo axis

Lag=0: Lag=0.2: Lag=0.4:

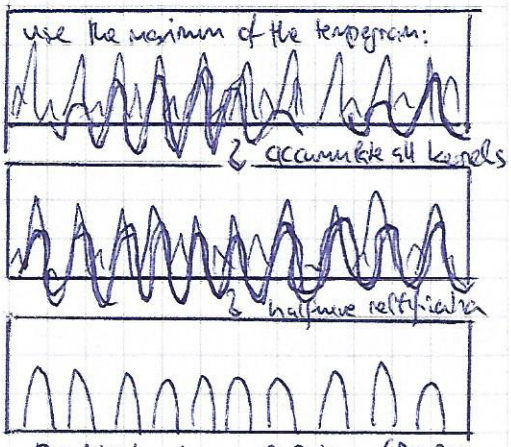




Comparison of windowed sinusoidals

Fourier ("fast")	vs.	Autocorrelation ("slow")
- reveals Δ's periodicities		- reveals Δ's self-similarities
- emphasizes harmonics		- emphasizes subharmonics
- @tatum/@tactus		- @tactus /@measure

use the maximum of the tempogram:



Predominant local Pulse (PLP)

6.3 Beat & Pulse Tracking

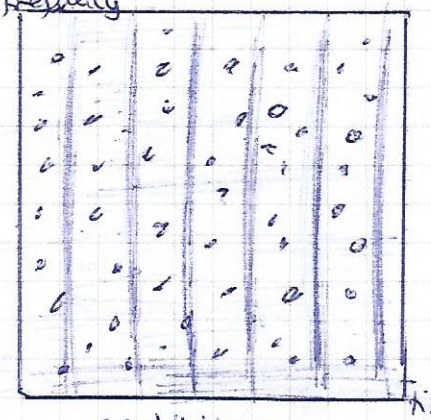
Goal: Given the tempo, find best sequence of beats.

PLP is a periodic enhancement of the novelty curve. The accumulator introduces error robustness and allows peak picking. The locality of the kernels allows for tempo variations.
Optionally, additional information (rough tempo range) can be used.
However, the predominant pulse level can change (maximum!).

↝ The detected beat positions can then be used as adaptive windows for improved feature extraction.

Chapter 7. Content-Based Audio Retrieval

Query-by-Example = { Audio identification / fingerprinting; Audio matching; Cover song identification }

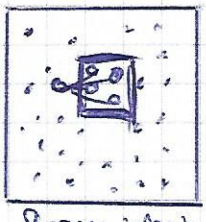


constellation map

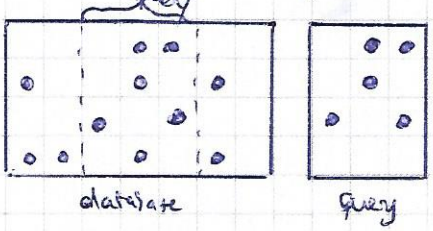
Audio identification / fingerprinting

Goal: Given a short audio query fragment, identify the original audio recording the query is taken from. (Shazam)

Idea: An audio fingerprint is a content-based compact signature that summarizes some specific audio content.
↝ constellation map of a spectrogram
↝ slide constellation map for query over the database
↝ requires clever indexing strategies
Idea: use pairs of peaks to increase hash specificity
- fix anchor point
- define target zone
- use pairs of peaks $(f_1, f_2, \Delta t)$



Shazam indexing

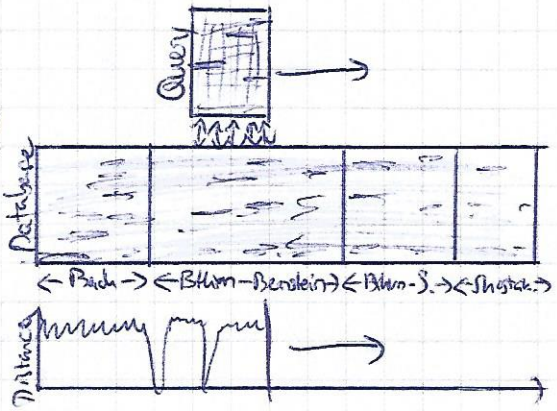


Audio matching

Goal: Given a short query audio fragment find all corresponding audio fragments of similar musical content (piece of music).

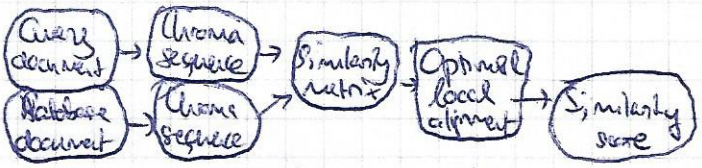
↳ use chroma features (allows to find different interpretations, instrumentations, rhythms, ...)
↳ smoothing, downsampling
↳ then use a matching procedure on the database
↳ for tempo differences, use local time warping or multiple scaling

- chroma: inv. to timbre, normalization: inv. to dynamics, smoothing: inv. to local time deviations

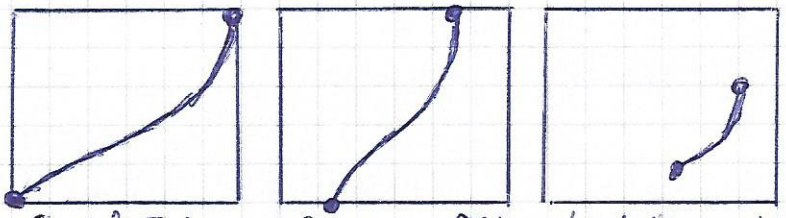


Cover song identification

Goal: Find all kinds of versions, interpretations or cover songs of a piece of music.
(potentially radically different interpretations!) ~ but usually at least one thing doesn't change.



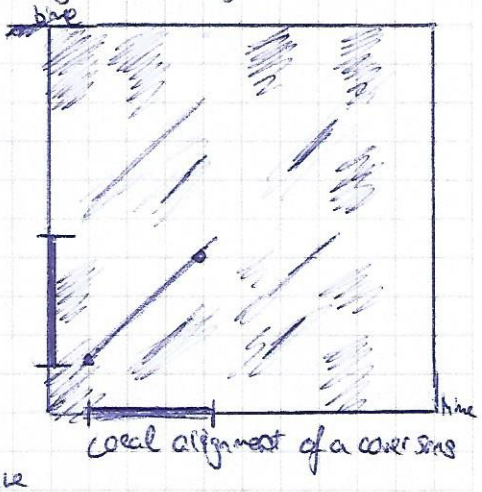
- calculate chroma sequences and a similarity matrix
- find a longest common subsequence with maximal score (local sequence alignment), e.g., Smith-Waterman algorithm used in bioinformatics for nucleotide/protein sequences



Classical DTW
compare whole recordings (audio identification)

Subsequence DTW
compare query with database (matching)

Local Alignment
compare most representative parts (cover song identification)

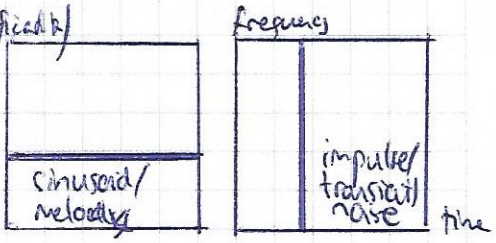


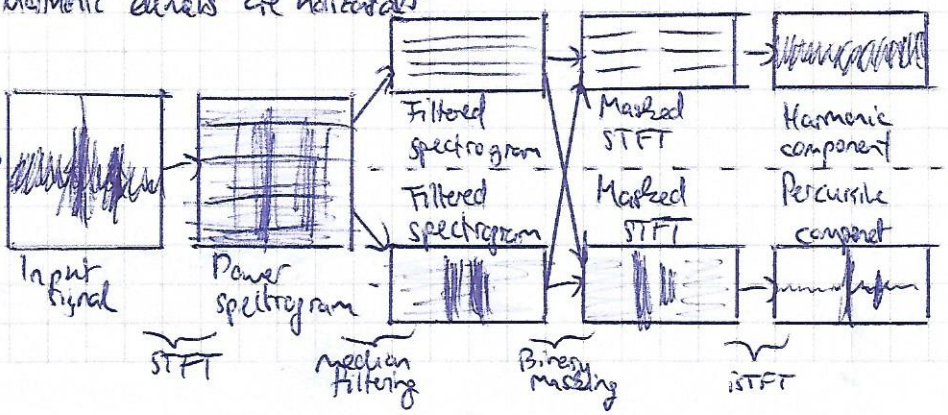
Chapter 8: Musically Informed Audio Decomposition

Source separation = decomposition of audio stream into different sources

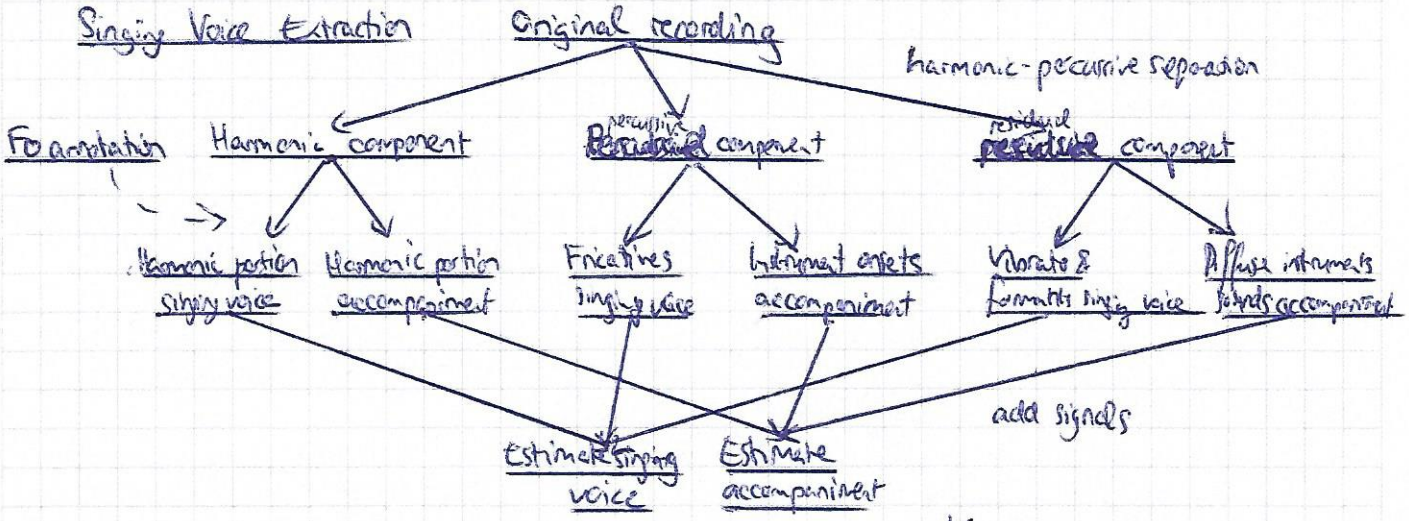
↳ harmonic-percussive separation
use the fact that percussive elements are vertical lines in a spectrogram and harmonic elements are horizontals

Harmonic: singing voice & accompaniment
Residual: vibrato/glissando noise
Percussive: drum hits, fricatives in singing voice



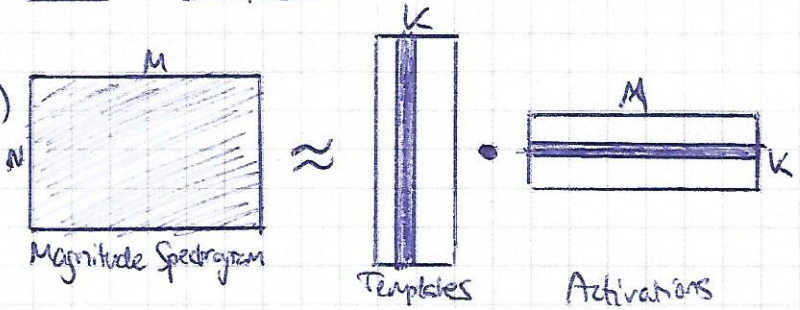


Singing Voice Extraction



Nonnegative Matrix Factorization

(source-informed audio decomposition)
factorization of a nonnegative matrix into two nonnegative matrices



- templates = pitch + timbre ("how does it sound")
- activations = onset time + duration ("when does it sound")

↝ Both matrices are initialized (e.g., randomly) and adapted via a learning procedure.
↳ or musically informed: prototypical pitches / overtones

Audio Mosaicing

Target Signal: Let it Be (Beatles)
Source Signal: Bees buzzing
Mosaic Signal: "Let It Bee"

} Idea: replace activation matrix with bee activations

BZZZZZ....

LE FIN □