INTERNATIONAL AUDIO LABORATORIES ERLANGEN

AUDIO LABS

Lecture

Music Processing

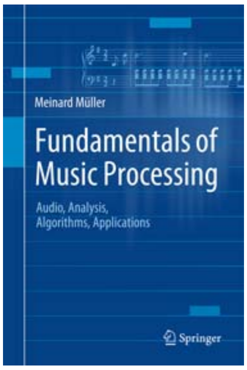
Audio Features

Meinard Müller
International Audio Laboratories Erlangen
meinard.mueller@audiolabs-erlangen.de

FAU FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Fraunhofer IIS

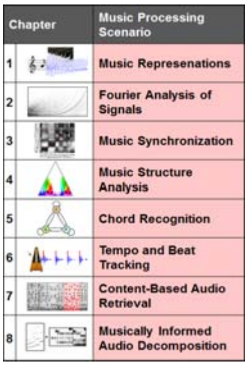
Book: Fundamentals of Music Processing



Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
483 p., 249 illus., hardcover
ISBN: 978-3-319-21944-8
Springer, 2015

Accompanying website:
www.music-processing.de

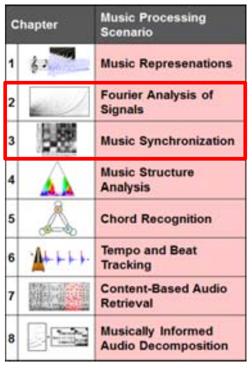
Book: Fundamentals of Music Processing



Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
483 p., 249 illus., hardcover
ISBN: 978-3-319-21944-8
Springer, 2015

Accompanying website:
www.music-processing.de

Book: Fundamentals of Music Processing



Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
483 p., 249 illus., hardcover
ISBN: 978-3-319-21944-8
Springer, 2015

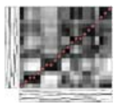
Accompanying website:
www.music-processing.de

Chapter 2: Fourier Analysis of Signals

2.1 The Fourier Transform in a Nutshell
2.2 Signals and Signal Spaces
2.3 Fourier Transform
2.4 Discrete Fourier Transform (DFT)
2.5 Short-Time Fourier Transform (STFT)
2.6 Further Notes

Important technical terminology is covered in Chapter 2. In particular, we approach the Fourier transform—which is perhaps the most fundamental tool in signal processing—from various perspectives. For the reader who is more interested in the musical aspects of the book, Section 2.1 provides a summary of the most important facts on the Fourier transform. In particular, the notion of a spectrogram, which yields a time–frequency representation of an audio signal, is introduced. The remainder of the chapter treats the Fourier transform in greater mathematical depth and also includes the fast Fourier transform (FFT)—an algorithm of great beauty and high practical relevance.

Chapter 3: Music Synchronization



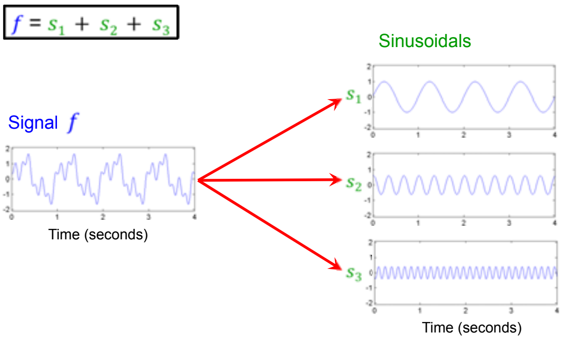
3.1 Audio Features
3.2 Dynamic Time Warping
3.3 Applications
3.4 Further Notes

As a first music processing task, we study in Chapter 3 the problem of music synchronization. The objective is to temporally align compatible representations of the same piece of music. Considering this scenario, we explain the need for musically informed audio features. In particular, we introduce the concept of chroma-based music features, which capture properties that are related to harmony and melody. Furthermore, we study an alignment technique known as dynamic time warping (DTW), a concept that is applicable for the analysis of general time series. For its efficient computation, we discuss an algorithm based on dynamic programming—a widely used method for solving a complex problem by breaking it down into a collection of simpler subproblems.

Fourier Transform

Idea: Decompose a given signal into a superposition of sinusoidals (elementary signals).

$$\boxed{f = s_1 + s_2 + s_3}$$



Fourier Transform

Each sinusoidal has a physical meaning and can be described by three parameters:

$$\boxed{s_{(A,\omega,\varphi)}(t) = A \cdot \sin(2\pi(\omega t - \varphi))}$$

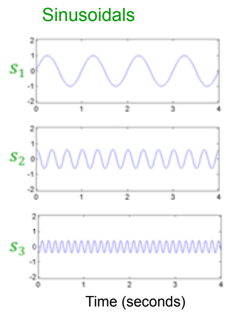
ω = frequency
A = amplitue
φ = phase

Interpretation:
The amplitude A reflects the intensity at which the sinusoidal of frequency ω appears in f.
The phase φ reflects how the sinusoidal has to be shifted to best correlate with f.

$A_1 = 1$, $\omega_1 = 1$, $\varphi_1 = 0$

$A_2 = 0.6$, $\omega_2 = 3$, $\varphi_2 = -0.2$

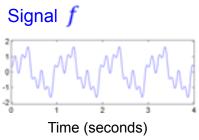
$A_3 = 0.4$, $\omega_3 = 7$, $\varphi_3 = 0.4$



Fourier Transform

Each sinusoidal has a physical meaning and can be described by three parameters:

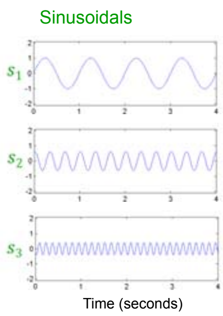
$$\boxed{f = s_1 + s_2 + s_3}$$



$A_1 = 1$, $\omega_1 = 1$, $\varphi_1 = 0$

$A_2 = 0.6$, $\omega_2 = 3$, $\varphi_2 = -0.2$

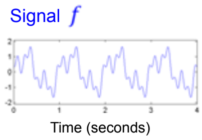
$A_3 = 0.4$, $\omega_3 = 7$, $\varphi_3 = 0.4$



Fourier Transform

Each sinusoidal has a physical meaning and can be described by three parameters:

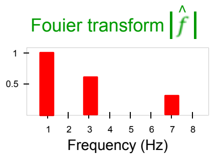
$$\boxed{f = s_1 + s_2 + s_3}$$



$A_1 = 1$, $\omega_1 = 1$, $\varphi_1 = 0$

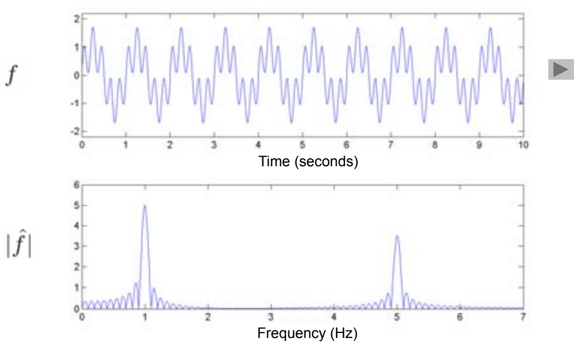
$A_2 = 0.6$, $\omega_2 = 3$, $\varphi_2 = -0.2$

$A_3 = 0.4$, $\omega_3 = 7$, $\varphi_3 = 0.4$



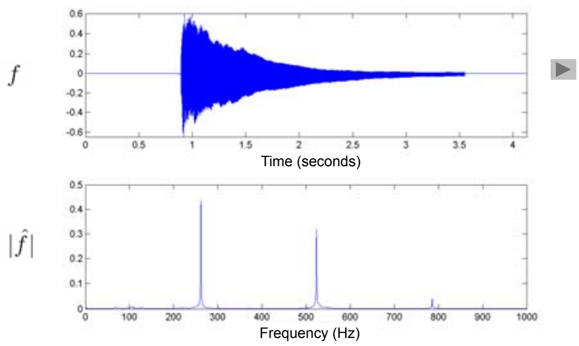
Fourier Transform

Example: Superposition of two sinusoidals



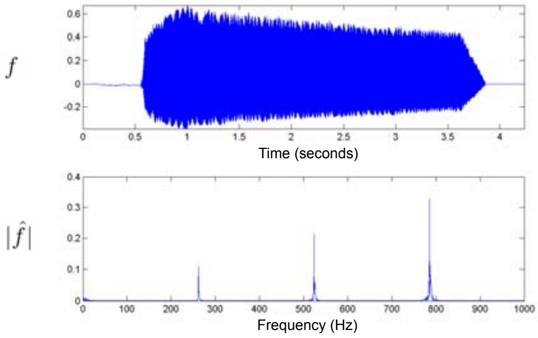
Fourier Transform

Example: C4 played by piano



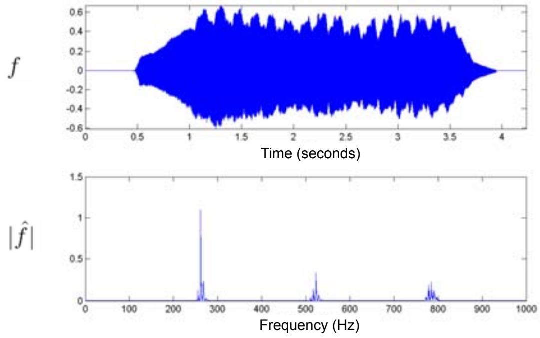
Fourier Transform

Example: C4 played by trumpet



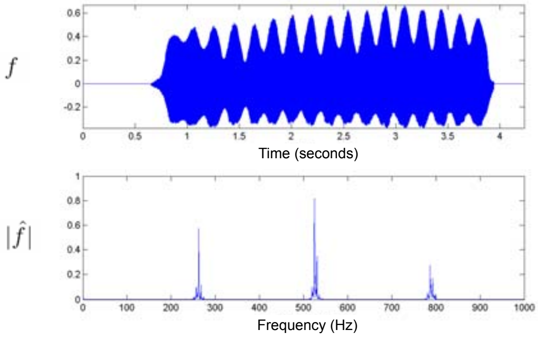
Fourier Transform

Example: C4 played by violine



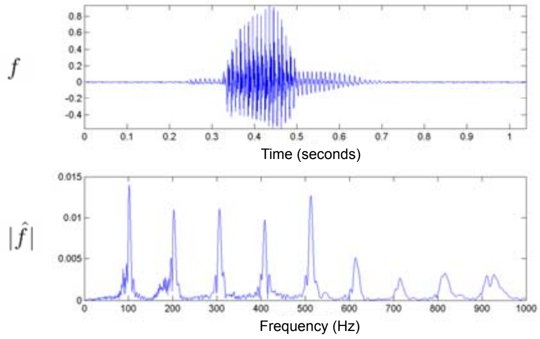
Fourier Transform

Example: C4 played by flute



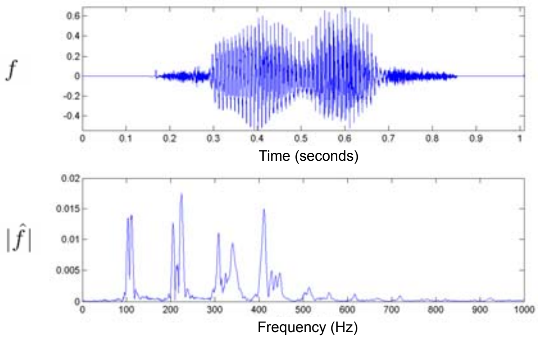
Fourier Transform

Example: Speech “Bonn”



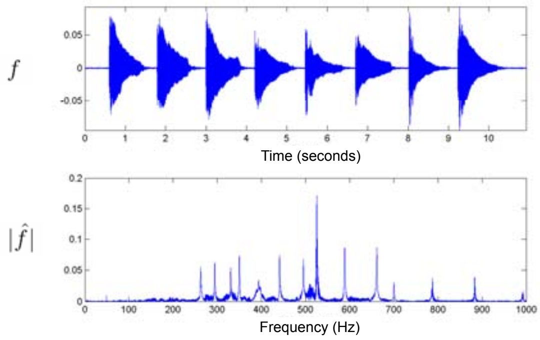
Fourier Transform

Example: Speech “Zürich”



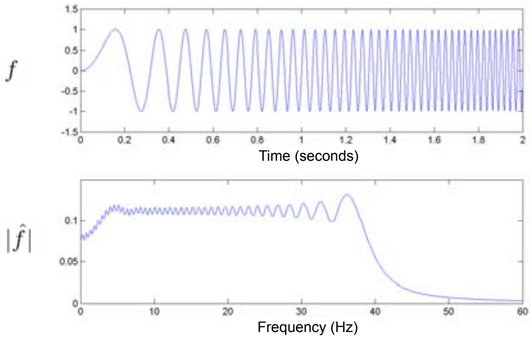
Fourier Transform

Example: C-major scale (piano)



Fourier Transform

Example: Chirp signal



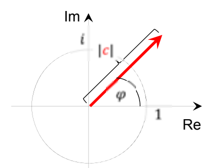
Fourier Transform

Each sinusoidal has a physical meaning and can be described by three parameters:

$$\boxed{s_{(A,\omega,\varphi)}(t) = A \cdot \sin(2\pi(\omega t - \varphi))}$$

ω = frequency
A = amplitue
φ = phase

Polar coordinates:
$c = |c| \cdot \exp(2\pi i \varphi)$



Complex formulation of sinusoidals:

$$\boxed{e_{(c,\omega)}(t) = c \cdot \exp(2\pi i \omega t) = c \cdot (\cos(2\pi\omega t) + i \cdot \sin(2\pi\omega t))}$$

ω = frequency
A = amplitue = $|c|$
φ = phase = $\arg(c)$

Fourier Transform

Signal $f : \mathbb{R} \to \mathbb{R}$

Fourier representation $f(t) = \int_{\omega \in \mathbb{R}} c_\omega e^{2\pi i \omega t} d\omega \ , \ c_\omega = \hat{f}(\omega)$

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

Fourier Transform

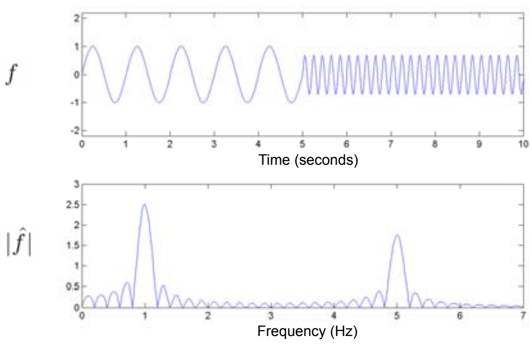
Signal $f : \mathbb{R} \to \mathbb{R}$

Fourier representation $f(t) = \int_{\omega \in \mathbb{R}} c_\omega e^{2\pi i \omega t} d\omega \ , \ c_\omega = \hat{f}(\omega)$

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

- Tells which frequencies occur, but does not tell when the frequencies occur.
- Frequency information is averaged over the entire time interval.
- Time information is hidden in the phase

Fourier Transform



Short Time Fourier Transform

Idea (Dennis Gabor, 1946):

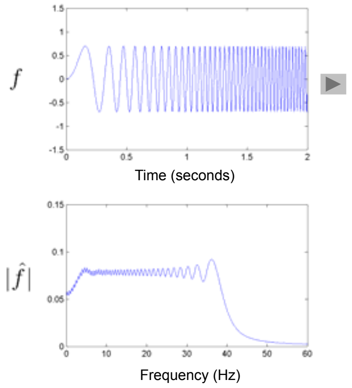
- Consider only a small section of the signal for the spectral analysis

  $\rightarrow$ recovery of time information

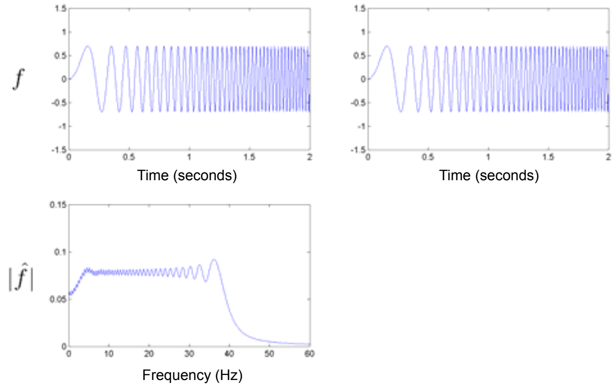
- Short Time Fourier Transform (STFT)

- Section is determined by pointwise multiplication of the signal with a localizing window function

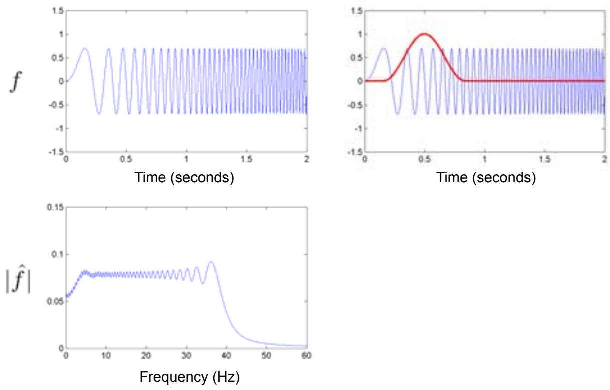
Short Time Fourier Transform



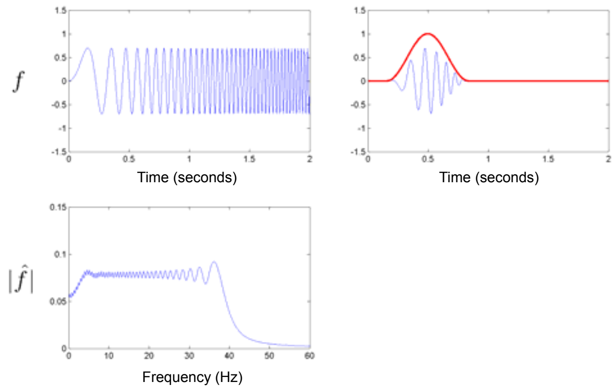
Short Time Fourier Transform



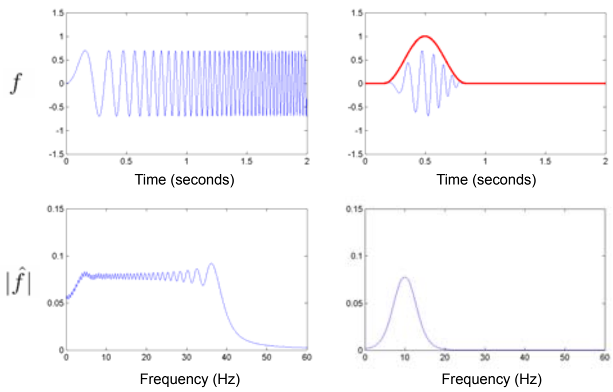
Short Time Fourier Transform



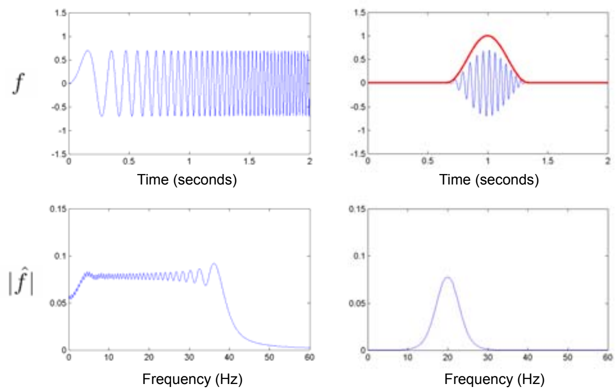
Short Time Fourier Transform



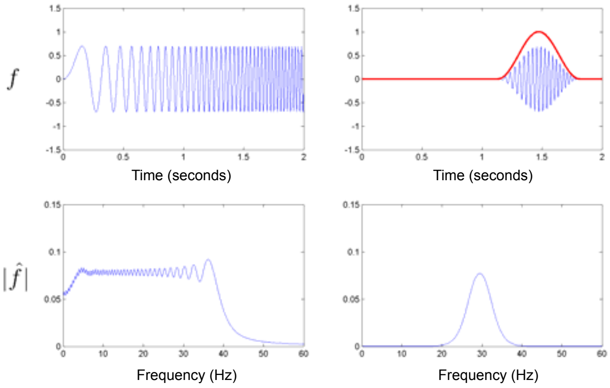
Short Time Fourier Transform



Short Time Fourier Transform



Short Time Fourier Transform



Short Time Fourier Transform

Definition

- Signal $f : \mathbb{R} \to \mathbb{R}$
- Window function $g : \mathbb{R} \to \mathbb{R}$ $\quad (g \in L^2(\mathbb{R}) \,, \|g\| = 1\ )$
- STFT $\tilde{f}(\omega, t) := \int_{\mathbb{R}} f(u)\bar{g}(u-t)e^{-2\pi i \omega u} du = \langle f | g_{\omega,t} \rangle$

with $\quad g_{\omega,t}(u) := e^{2\pi i \omega u} g(u-t), \quad u \in \mathbb{R}$

Short Time Fourier Transform

Intuition:

- $g_{\omega,t}$ is "musical note" of frequency ω, which oscillates within the translated window $u \to g(u-t)$

Short Time Fourier Transform

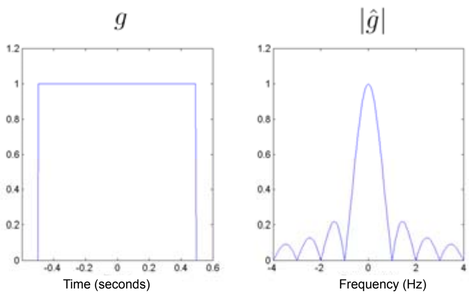
Intuition:

- $g_{\omega,t}$ is "musical note" of frequency ω, which oscillates within the translated window $u \to g(u-t)$

- Inner product $\langle f | g_{\omega,t} \rangle$ measures the correlation between the musical note $g_{\omega,t}$ and the signal f.

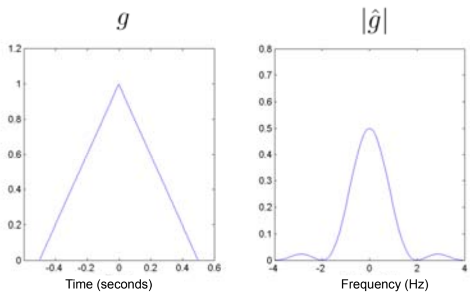
Window Function

Box window



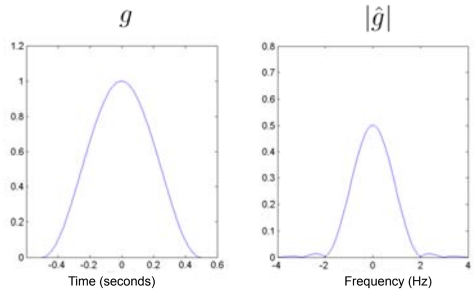
Window Function

Triangle window

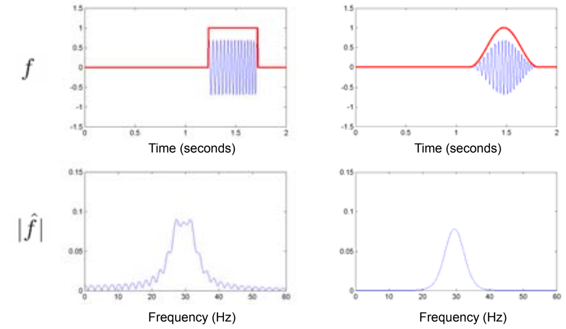


Window Function

Hann window

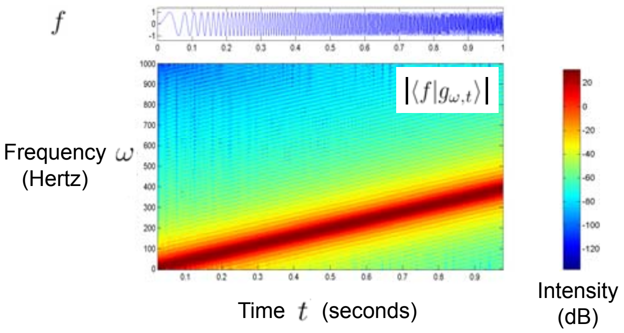


Window Function

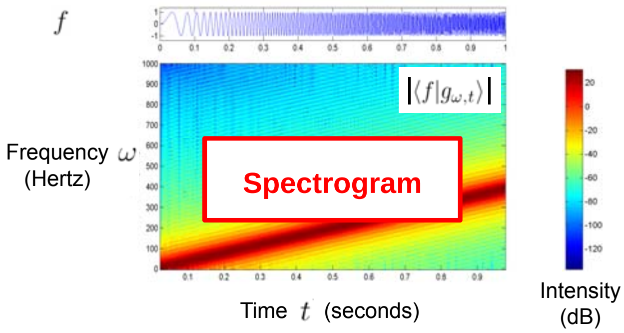


Trade off between smoothing and „ringing“

Time-Frequency Representation

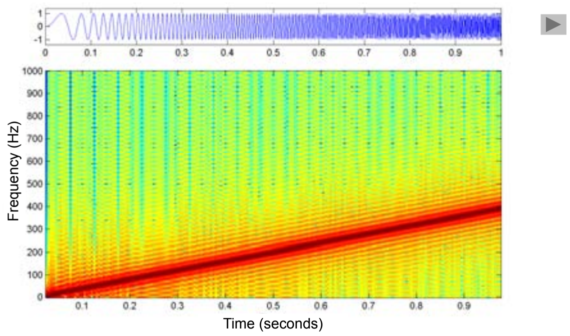


Time-Frequency Representation



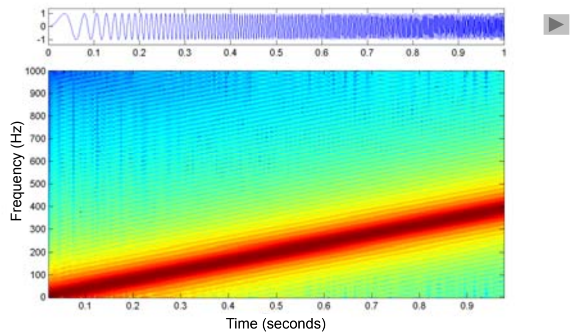
Time-Frequency Representation

Chirp signal and STFT with box window of length 0.05



Time-Frequency Representation

Chirp signal and STFT with Hann window of length 0.05



Time-Frequency Localization

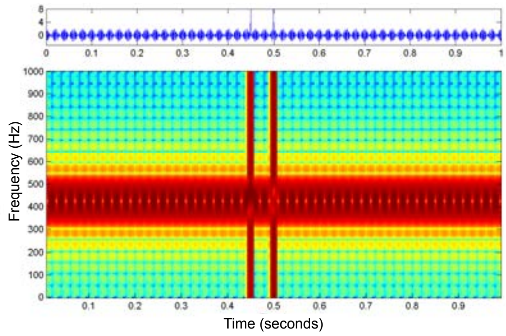
- Size of window constitutes a trade-off between time resolution and frequency resolution:

  Large window : poor time resolution
  good frequency resolution
  Small window : good time resolution
  poor frequency resolution

- Heisenberg Uncertainty Principle: there is no window function that localizes in time and frequency with arbitrary position.

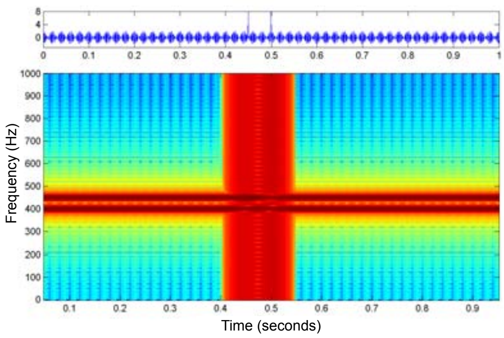
Short Time Fourier Transform

Signal and STFT with Hann window of length 0.02



Short Time Fourier Transform

Signal and STFT with Hann window of length 0.1



MATLAB

- MATLAB function SPECTROGRAM
- *N* = window length (in samples)
- *M* = overlap (usually $N/2$)
- Compute DFT_N for every windowed section
- Keep lower $N/2$ Fourier coefficients

→ Sequence of spectral vectors
(for each window a vector of dimension $N/2$)

Example

Let x be a discrete time signal $x(n) = f(Tn)$
Sampling rate: $1/T = 22050$ Hz
Window length: $N = 4096$
Overlap: $N/2 = 2048$
Hopsize: window length − overlap

Let
$$\begin{aligned} v_0 &:= (x(0),\ x(1),\ \ldots,\ x(4095)) \\ v_1 &:= (x(2048),\ \ldots,\ x(6143)) \\ v_2 &:= (x(4096),\ \ldots,\ x(8191)) \\ &\vdots \end{aligned}$$

v_m corresponds to window $[m \cdot 2048 : m \cdot 2048 + 4095]$

Example

Time resolution:

$$\frac{\text{hopsize}}{\text{sampling rate}} = \frac{4096 - 2048}{22050} = 0.093 = 93\ ms$$

Frequency resolution:

$$v = v_0\ ,\ \hat{v} := \mathrm{DFT}_N(v)$$

$$\hat{v}(k) \approx \frac{1}{T} \cdot \hat{f}\left(\frac{k}{N} \cdot \frac{1}{T}\right)$$

$$\omega = \frac{k}{N} \cdot \frac{1}{T} = k \cdot \frac{22050}{4096} = k \cdot 5.38 \text{ Hz}$$

Pitch Features

Model assumption: Equal-tempered scale

- MIDI pitches: $p \in [1:128]$
- Piano notes: $p = 21\ (\mathrm{A0})$ to $p = 108\ (\mathrm{C8})$
- Concert pitch: $p = 69\ (\mathrm{A4})$
- Center frequency: $f_{\mathrm{MIDI}}(p) = 2^{\frac{p-69}{12}} \cdot 440$ Hz

→ Logarithmic frequency distribution
Octave: doubling of frequency

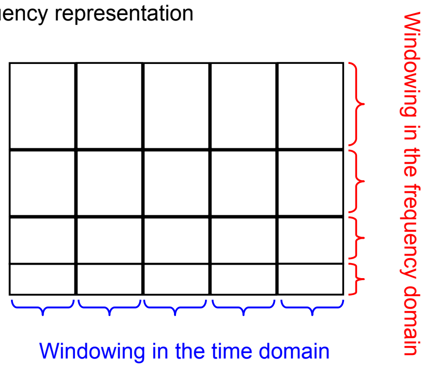
Pitch Features

Idea: Binning of Fourier coefficients

Divide up the fequency axis into
logarithmically spaced „pitch regions"
and combine spectral coefficients
of each region to a single pitch coefficient.

Pitch Features

Time-frequency representation



Pitch Features

Details:

- Let $\hat{v}$ be a spectral vector obtained from a spectrogram w.r.t. a sampling rate $1/T$ and a window length *N*. The spectral coefficient $\hat{v}(k)$ corresponds to the frequency

$$f_{\mathrm{coeff}}(k) := \frac{k}{N} \cdot \frac{1}{T}$$

- Let
$S(p) := \{k : f_{\mathrm{MIDI}}(p-0.5) \leq f_{\mathrm{coeff}}(k) < f_{\mathrm{MIDI}}(p+0.5)\}$
be the set of coefficients assigned to a pitch $p \in [1:128]$
Then the pitch coefficient $P(p)$ is defined as

$$P(p) := \sum_{k \in S(p)} |\hat{v}(k)|^2$$

Pitch Features

Example: A4, *p* = 69

- Center frequency: $f(p=69) = 2^{\frac{0}{12}} \cdot 440 = 440\ Hz$
- Lower bound: $f(p=68.5) = 2^{\frac{-0.5}{12}} \cdot 440 = 427.5\ Hz$
- Upper bound: $f(p=69.5) = 2^{\frac{0.5}{12}} \cdot 440 = 452.9\ Hz$
- STFT with $N = 4096$, $1/T = 22050$

$$\begin{array}{lcl} & \vdots & \\ f(k=79) & = & 425.3\ Hz \\ f(k=80) & = & 430.7\ Hz \\ f(k=81) & = & 436.0\ Hz \\ f(k=82) & = & 441.4\ Hz \\ f(k=83) & = & 446.8\ Hz \\ f(k=84) & = & 452.2\ Hz \\ f(k=85) & = & 457.6\ Hz \\ & \vdots & \end{array}$$

Pitch Features

Example: A4, *p* = 69

- Center frequency: $f(p=69) = 2^{\frac{0}{12}} \cdot 440 = 440\ Hz$
- Lower bound: $f(p=68.5) = 2^{\frac{-0.5}{12}} \cdot 440 = 427.5\ Hz$
- Upper bound: $f(p=69.5) = 2^{\frac{0.5}{12}} \cdot 440 = 452.9\ Hz$
- STFT with $N = 4096$, $1/T = 22050$

$$\begin{array}{lcl} & \vdots & \\ f(k=79) & = & 425.3\ Hz \end{array}$$

$$\left.\boxed{\begin{array}{lcl} f(k=80) & = & 430.7\ Hz \\ f(k=81) & = & 436.0\ Hz \\ f(k=82) & = & 441.4\ Hz \\ f(k=83) & = & 446.8\ Hz \\ f(k=84) & = & 452.2\ Hz \end{array}}\ \right\} S(p=69)$$

$$\begin{array}{lcl} f(k=85) & = & 457.6\ Hz \\ & \vdots & \end{array}$$

$$P(p=69) = \sum_{k=80}^{84} |\hat{v}(k)|^2$$

Pitch Features

Note	MIDI pitch	Center [Hz] frequency	Left [Hz] boundary	Right [Hz] boundary	Width [Hz]
A3	57	220.0	213.7	226.4	12.7
A#3	58	233.1	226.4	239.9	13.5
B3	59	246.9	239.9	254.2	14.3
C4	60	261.6	254.2	269.3	15.1
C#4	61	277.2	269.3	285.3	16.0
D4	62	293.7	285.3	302.3	17.0
D#4	63	311.1	302.3	320.2	18.0
E4	64	329.6	320.2	339.3	19.0
F4	65	349.2	339.3	359.5	20.2
F#4	66	370.0	359.5	380.8	21.4
G4	67	392.0	380.8	403.5	22.6
G#4	68	415.3	403.5	427.5	24.0
A4	69	440.0	427.5	452.9	25.4

Pitch Features

Note:

- $P \in \mathbb{R}^{128}$
- For some pitches, *S(p)* may be empty. This particularly holds for low notes corresponding to narrow frequency bands.

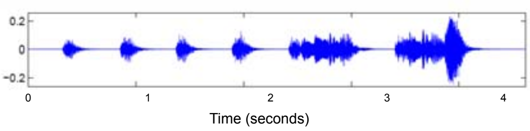
→ Linear frequency sampling is problematic!

Solution:

Multi-resolution spectrograms or multirate filterbanks

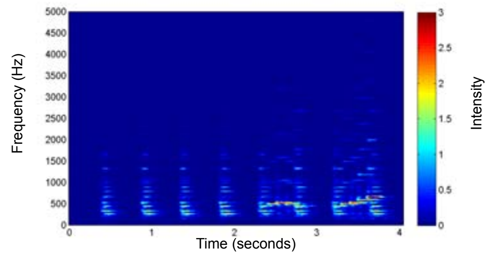
Pitch Features

Example: Friedrich Burgmüller, Op. 100, No. 2



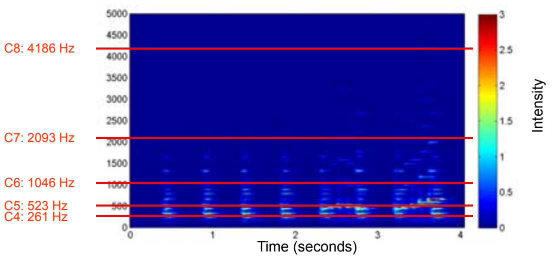
Pitch Features

Spectrogram



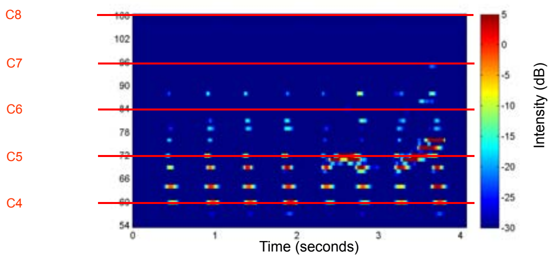
Pitch Features

Spectrogram



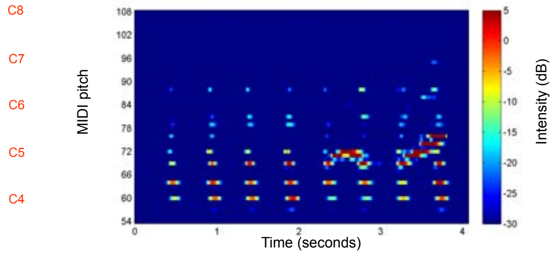
Pitch Features

Pitch representation



Pitch Features

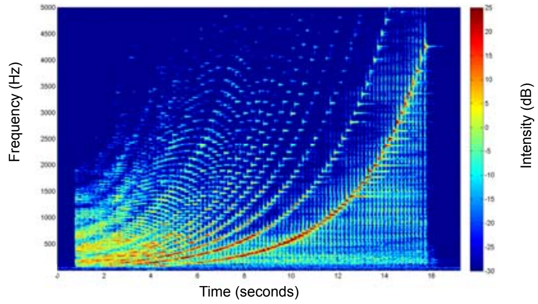
Pitch representation



Pitch Features

Example: Chromatic scale

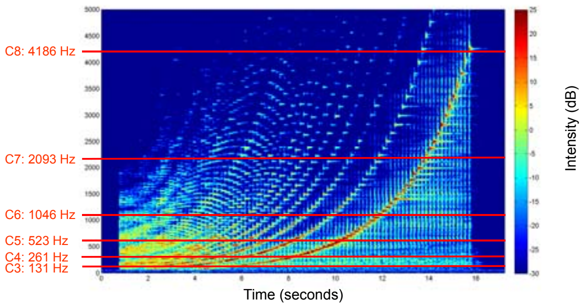
Spectrogram



Pitch Features

Example: Chromatic scale

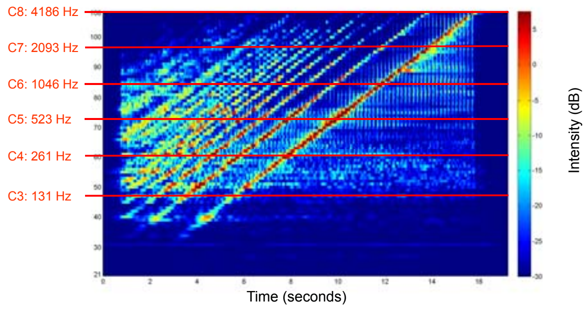
Spectrogram



Pitch Features

Example: Chromatic scale

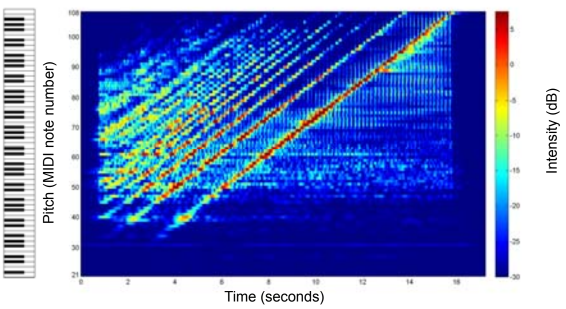
Log-frequency spectrogram



Pitch Features

Example: Chromatic scale

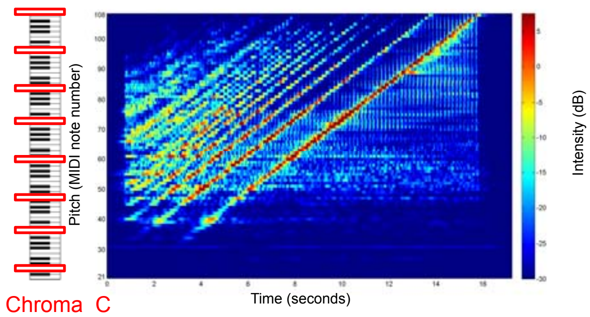
Log-frequency spectrogram



Pitch Features

Example: Chromatic scale

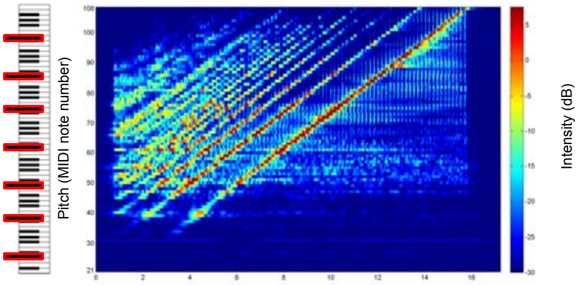
Log-frequency spectrogram



Pitch Features

Example: Chromatic scale

Log-frequency spectrogram

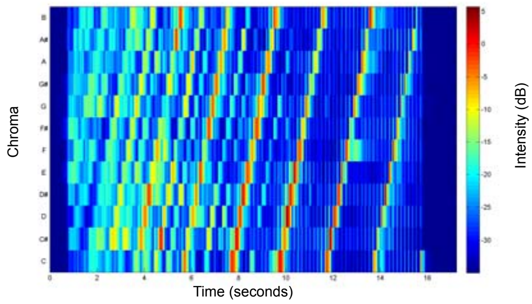


Chroma C#

Chroma Features

Example: Chromatic scale

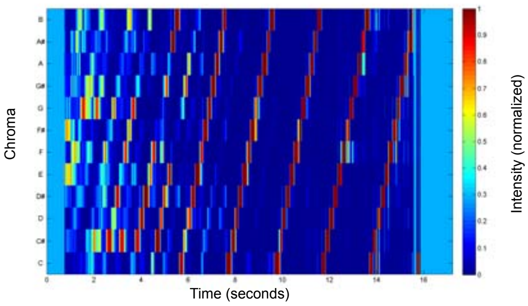
Chroma representation



Chroma Features

Example: Chromatic scale

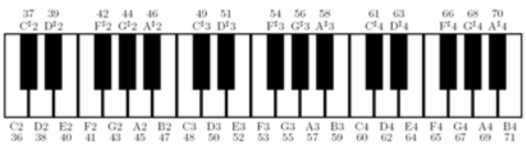
Chroma representation (normalized, Euclidean)



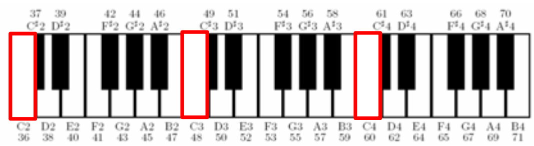
Chroma Features

- Human perception of pitch is periodic in the sense that two pitches are perceived as similar in color if they differ by an octave.
- Seperation of pitch into two components: tone height (octave number) and chroma.
- Chroma : 12 traditional pitch classes of the equal-tempered scale. For example:
  Chroma C $\widehat{=} \{\ldots , \mathrm{C0} , \mathrm{C1} , \mathrm{C2} , \mathrm{C3} , \ldots\}$
- Computation: pitch features → chroma features
  Add up all pitches belonging to the same class
- Result: 12-dimensional chroma vector.

Chroma Features



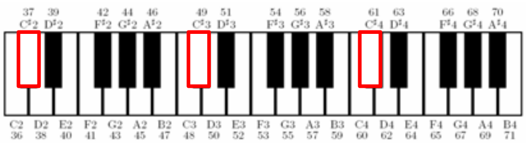
Chroma Features



C2 C3 C4

Chroma C

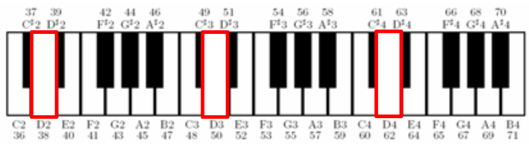
Chroma Features



C#2 C#3 C#4

Chroma C#

Chroma Features

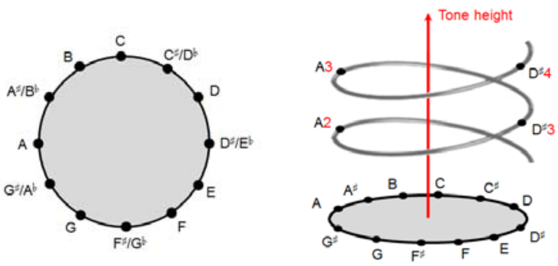


D2 D3 D4

Chroma D

Chroma Features

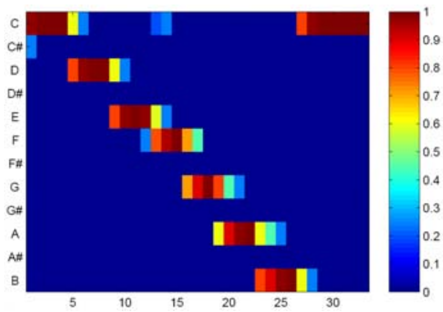
Chromatic circle Shepard's helix of pitch perception



Meinard Müller: Fundamentals of Music Processing
Chapter 1: Music Representations, Fig. 1.3

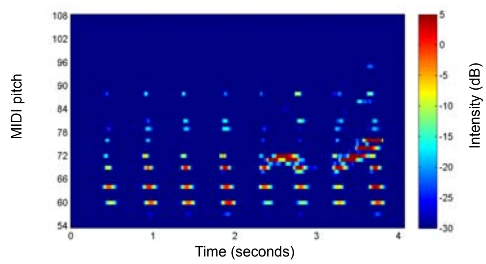
Chroma Features

Example: C-Major Scale



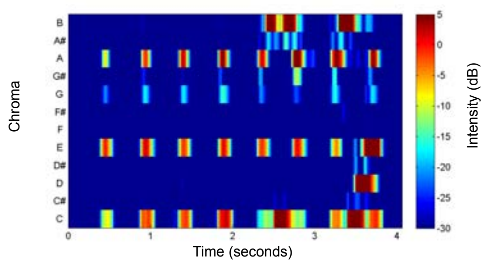
Chroma Features

Pitch representation



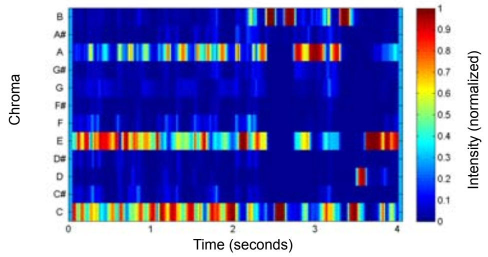
Chroma Features

Chroma representation



Chroma Features

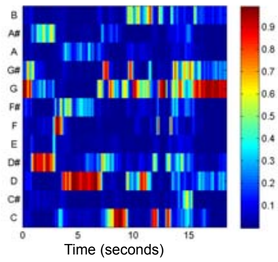
Chroma representation (normalized)



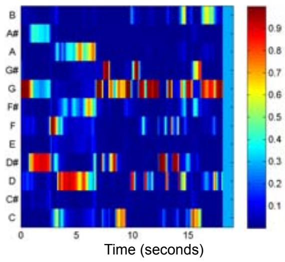
Chroma Features

Example: Beethoven's Fifth
Chroma representation (normalized, 10 Hz)

Karajan



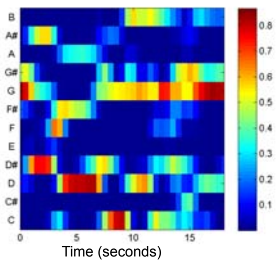
Scherbakov



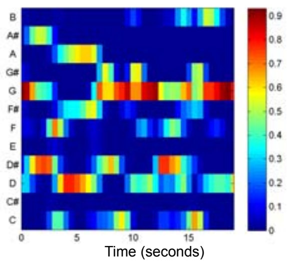
Chroma Features

Example: Beethoven's Fifth
Chroma representation (normalized, 2 Hz)
Smoothing (2 seconds) + downsampling (factor 5)

Karajan



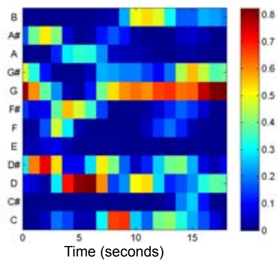
Scherbakov



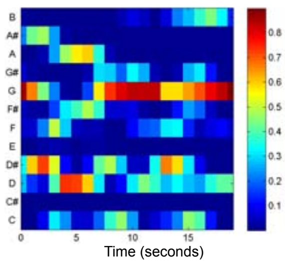
Chroma Features

Example: Beethoven's Fifth
Chroma representation (normalized, 1 Hz)
Smoothing (4 seconds) + downsampling (factor 10)

Karajan



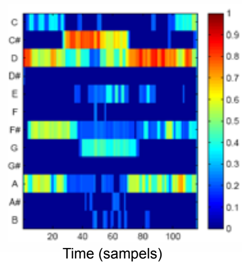
Scherbakov



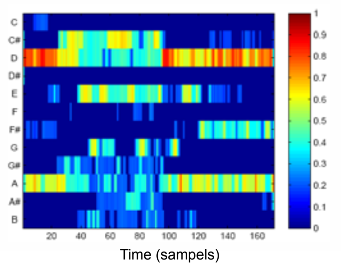
Chroma Features

Example: Bach Toccata

Koopman



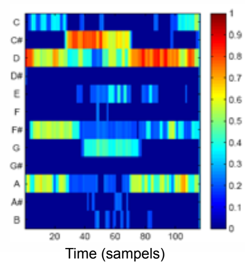
Ruebsam



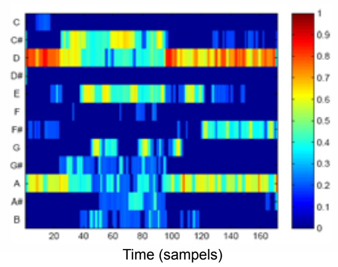
Chroma Features

Example: Bach Toccata

Koopman



Ruebsam

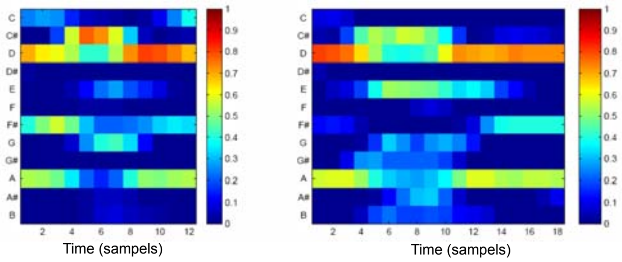


Feature resolution: 10 Hz

Chroma Features

Example: Bach Toccata

Koopman   Ruebsam

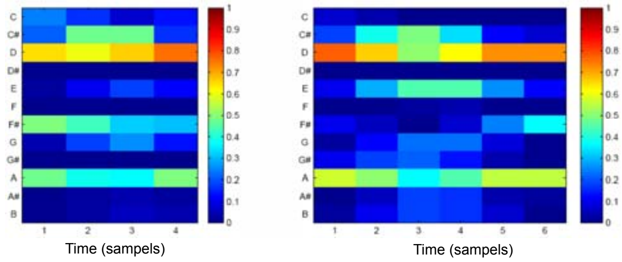


Feature resolution: 1 Hz

Chroma Features

Example: Bach Toccata

Koopman   Ruebsam



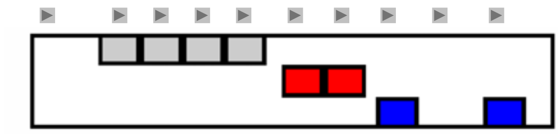
Feature resolution: 0.33 Hz

Chroma Features

- Sequence of chroma vectors correlates to the harmonic progression
- Normalization $v \rightarrow \frac{v}{\|v\|}$ makes features invariant to changes in dynamics
- Further quantization and smoothing: CENS features
- Taking logarithm before adding up pitch coefficients accounts for logarithmic sensation of intensity

Chroma Features

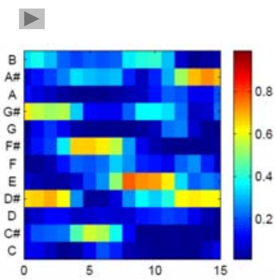
Example: Zager & Evans “In The Year 2525”



How to deal with transpositions?

Chroma Features

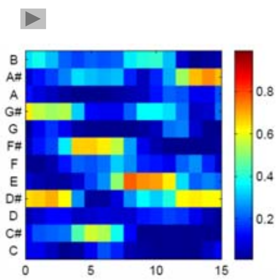
Example: Zager & Evans “In The Year 2525”



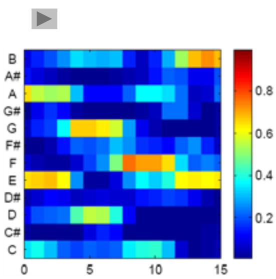
Original: $(v^1, \ldots, v^N)$

Chroma Features

Example: Zager & Evans “In The Year 2525”



Original: $(v^1, \ldots, v^N)$



Shifted: $(\sigma(v^1), \ldots, \sigma(v^N))$

Audio Features

- There are many ways to implement chroma features
- Properties may differ significantly
- Appropriateness depends on respective application

- http://www.mpi-inf.mpg.de/resources/MIR/chromatoolbox/
- MATLAB implementations for various chroma variants