INTERNATIONAL AUDIO LABORATORIES ERLANGEN

AUDIO LABS

Lecture

Music Processing

Audio Features

Meinard Müller

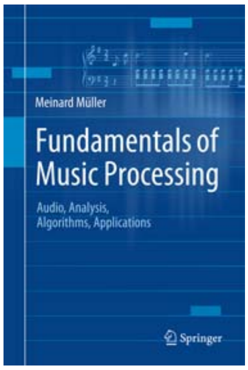
International Audio Laboratories Erlangen

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

Chapter	Music Processing Scenario
1	Music Representations
2	Fourier Analysis of Signals
3	Music Synchronization
4	Music Structure Analysis
5	Chord Recognition
6	Tempo and Beat Tracking
7	Content-Based Audio Retrieval
8	Musically Informed Audio Decomposition

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

Chapter	Music Processing Scenario
1	Music Representations
2	Fourier Analysis of Signals
3	Music Synchronization
4	Music Structure Analysis
5	Chord Recognition
6	Tempo and Beat Tracking
7	Content-Based Audio Retrieval
8	Musically Informed Audio Decomposition

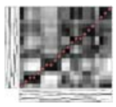
Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
483 p., 249 illus., hardcover
ISBN: 978-3-319-21944-8
Springer, 2015

Accompanying website:
www.music-processing.de

Chapter 2: Fourier Analysis of Signals

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

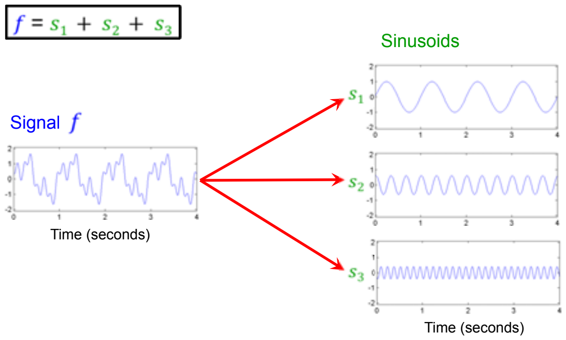


As a first music processing task, we study in Chapter 3 the problem of music synchronization. The objective is to temporally align compatible representations of the same piece of music. Considering this scenario, we explain the need for musically informed audio features. In particular, we introduce the 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 sinusoids (elementary signals).

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



Fourier Transform

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

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

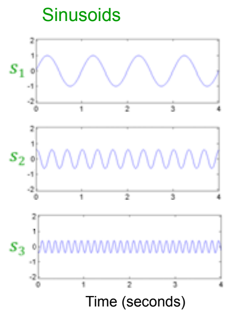
ω = frequency
A = amplitue
φ = phase

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.

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

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

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

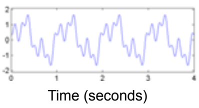


Fourier Transform

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

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

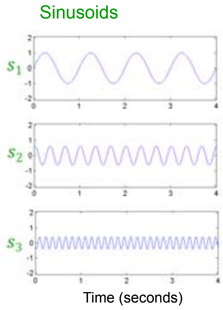
Signal f



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

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

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

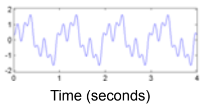


Fourier Transform

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

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

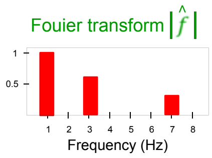
Signal f



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

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

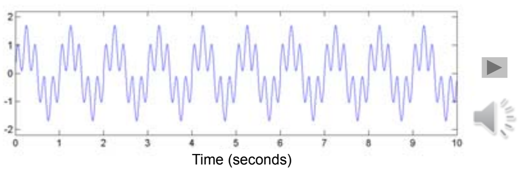
$A_3 = 0.4$, $\omega_3 = 7$, $\varphi_3 = 0.4$



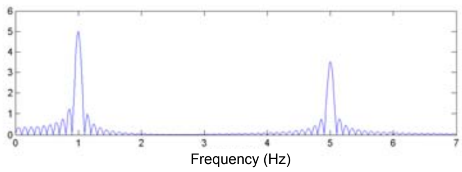
Fourier Transform

Example: Superposition of two sinusoids

f



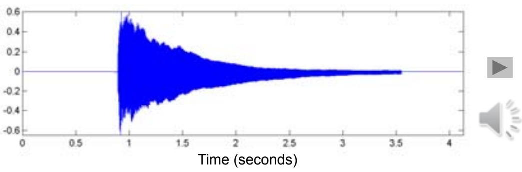
$|\hat{f}|$



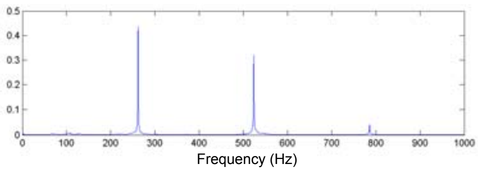
Fourier Transform

Example: C4 played by piano

f

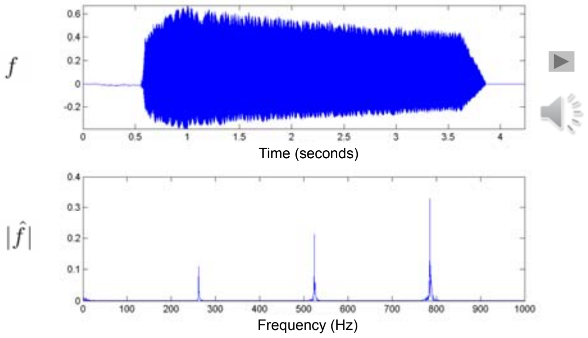


$|\hat{f}|$



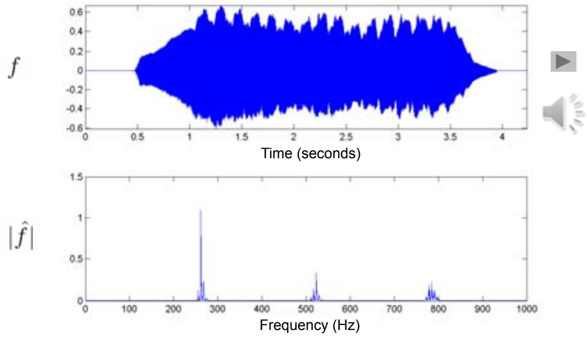
Fourier Transform

Example: C4 played by trumpet



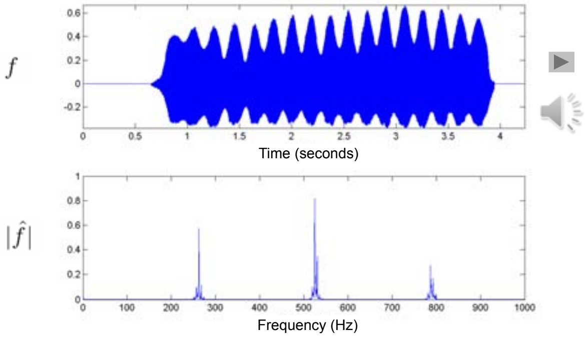
Fourier Transform

Example: C4 played by violine



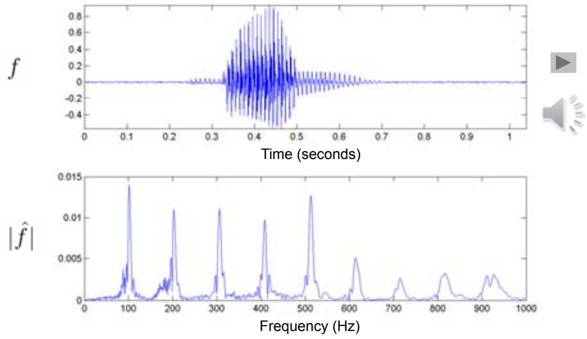
Fourier Transform

Example: C4 played by flute



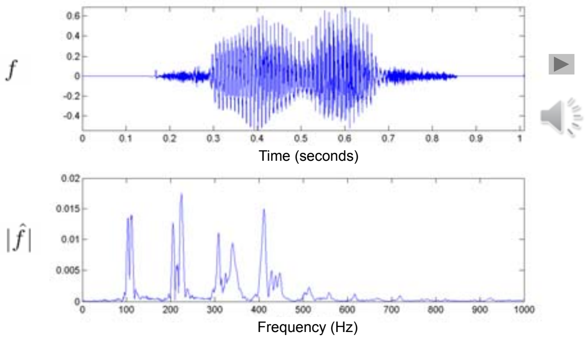
Fourier Transform

Example: Speech “Bonn”



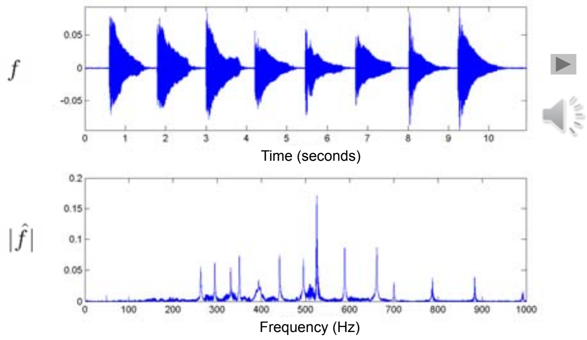
Fourier Transform

Example: Speech “Zürich”



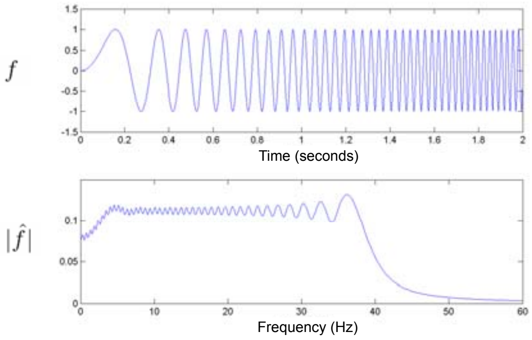
Fourier Transform

Example: C-major scale (piano)



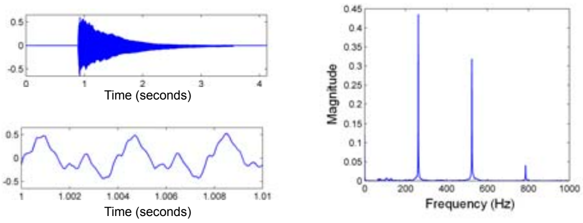
Fourier Transform

Example: Chirp signal



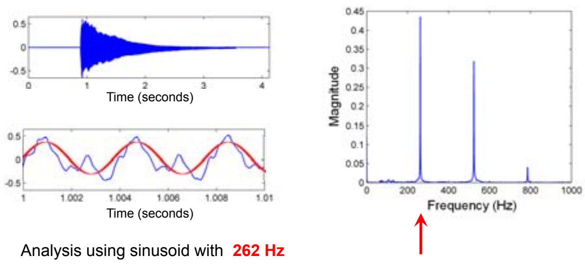
Fourier Transform

Example: Piano tone (C4, 261.6 Hz)



Fourier Transform

Example: Piano tone (C4, 261.6 Hz)



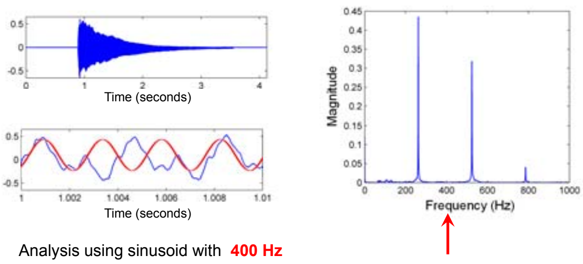
Analysis using sinusoid with **262 Hz**

→ high correlation

→ large Fourier coefficient

Fourier Transform

Example: Piano tone (C4, 261.6 Hz)



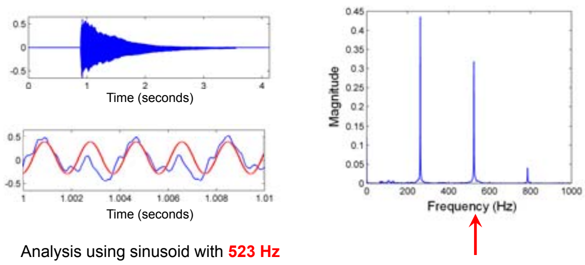
Analysis using sinusoid with **400 Hz**

→ low correlation

→ small Fourier coefficient

Fourier Transform

Example: Piano tone (C4, 261.6 Hz)



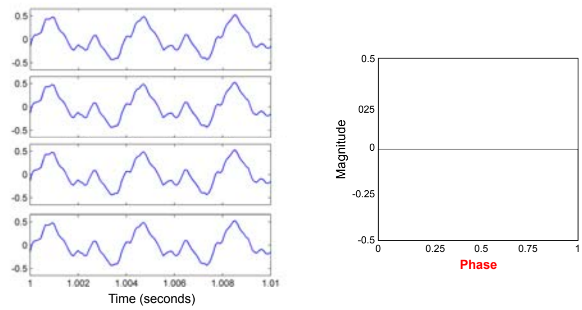
Analysis using sinusoid with **523 Hz**

→ high correlation

→ large Fourier coefficient

Fourier Transform

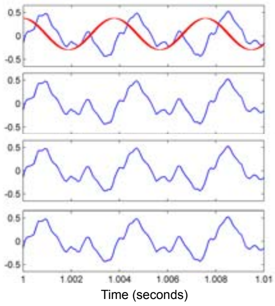
Role of phase

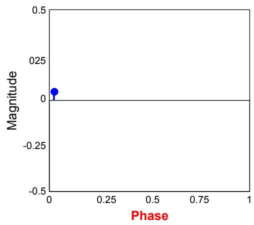


Fourier Transform

Role of phase

Analysis with sinusoid having frequency 262 Hz and phase $\varphi = 0.05$

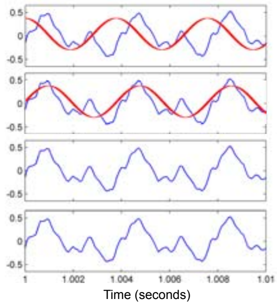


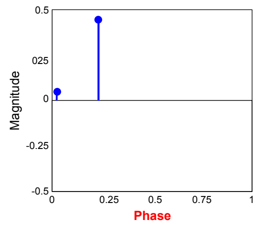


Fourier Transform

Role of phase

Analysis with sinusoid having frequency 262 Hz and phase $\varphi = 0.24$

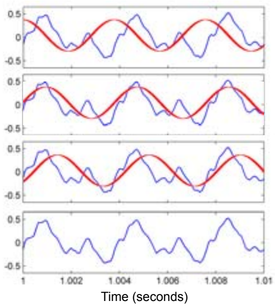


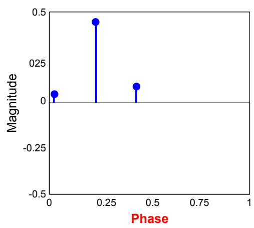


Fourier Transform

Role of phase

Analysis with sinusoid having frequency 262 Hz and phase $\varphi = 0.45$

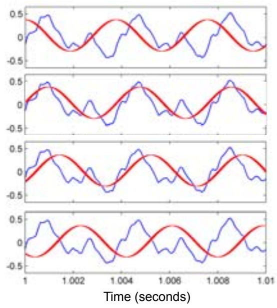


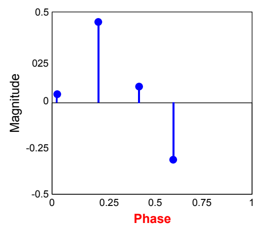


Fourier Transform

Role of phase

Analysis with sinusoid having frequency 262 Hz and phase $\varphi = 0.6$





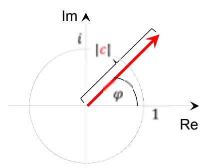
Fourier Transform

Each sinusoid 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 = amplitude
φ = phase

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



Complex formulation of sinusoids:

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

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

Fourier Transform

Signal $\quad f\colon \mathbb{R} \to \mathbb{R}$

Fourier representation $\quad f(t) = \int_{\omega \in \mathbb{R}} c_\omega \exp(2\pi i \omega t) d\omega$

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

Fourier Transform

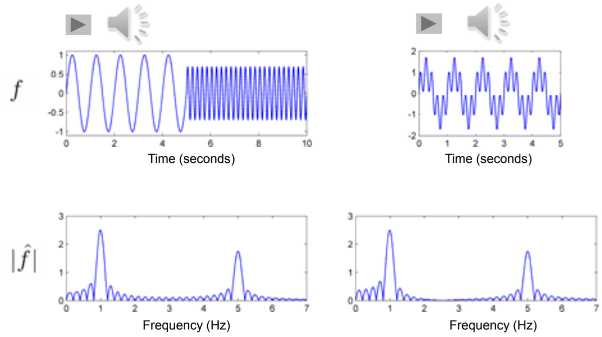
Signal $f: \mathbb{R} \to \mathbb{R}$

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

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

- 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):

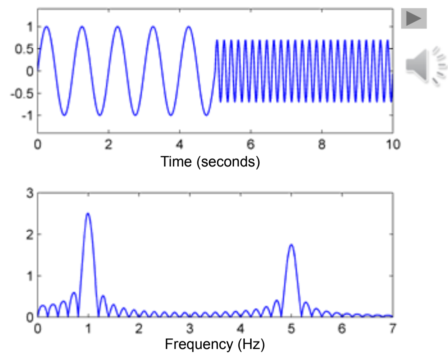
- Consider only a small section of the signal for the spectral analysis

  → recovery of time information

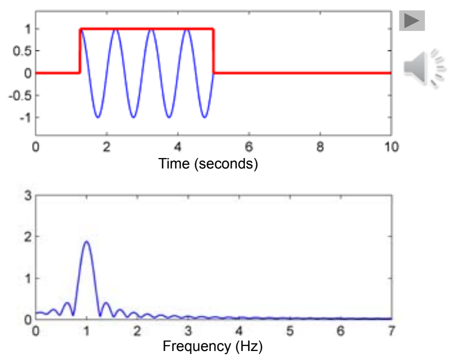
- Short Time Fourier Transform (STFT)

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

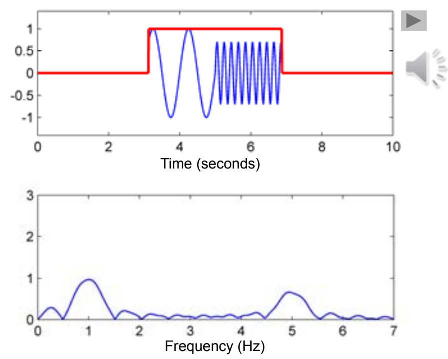
Short Time Fourier Transform



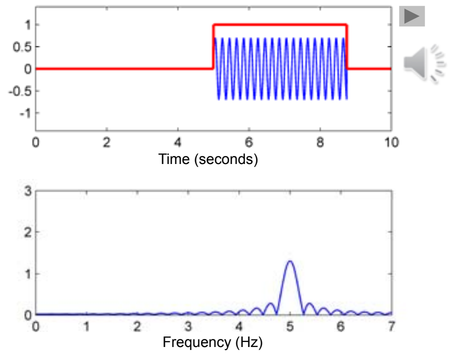
Short Time Fourier Transform



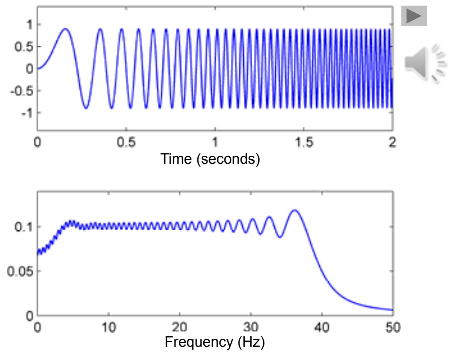
Short Time Fourier Transform



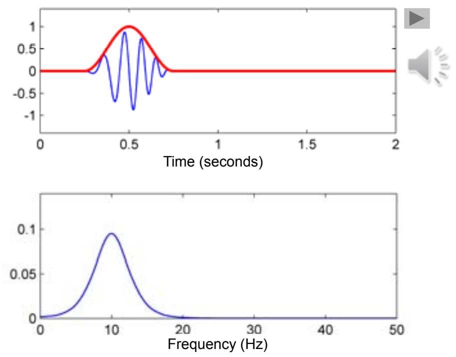
Short Time Fourier Transform



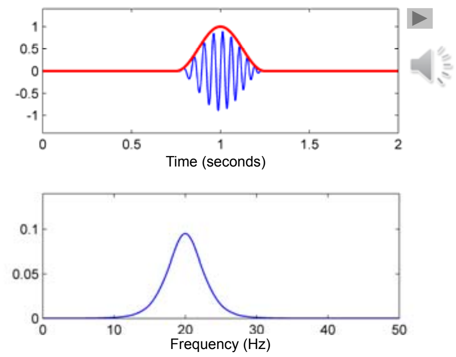
Short Time Fourier Transform



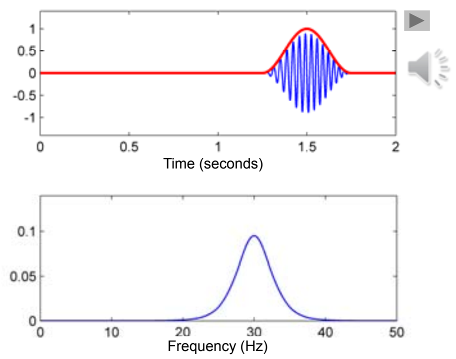
Short Time Fourier Transform



Short Time Fourier Transform

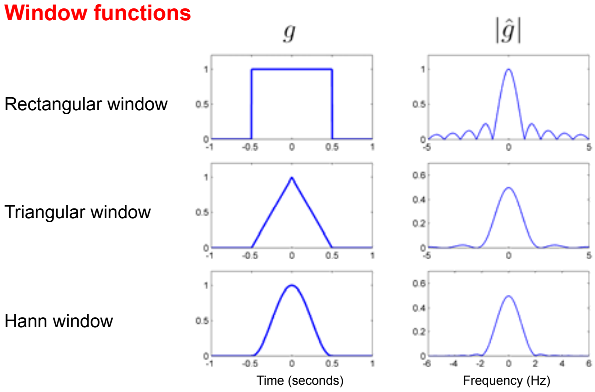


Short Time Fourier Transform



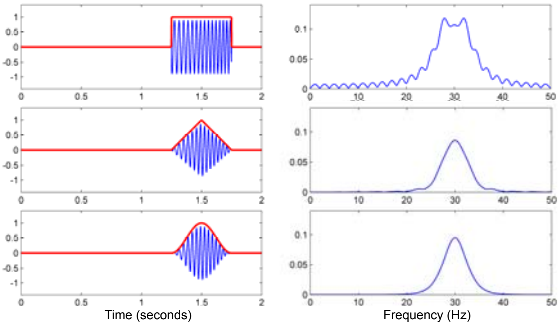
Short Time Fourier Transform

Window functions



Short Time Fourier Transform

Window functions



→ Trade off between smoothing and "ringing"

Short Time Fourier Transform

Definition

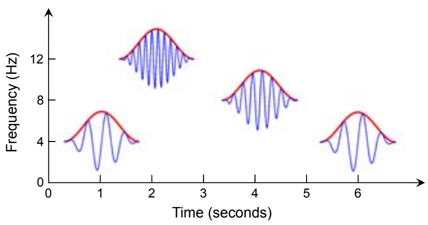
- Signal $f:\mathbb{R}\to\mathbb{R}$
- Window function $g:\mathbb{R}\to\mathbb{R}$ $\quad (g\in L^2(\mathbb{R}), \|g\|_2\neq 0)$
- STFT $\tilde{f}_g(t,\omega)=\int_{u\in\mathbb{R}} f(u)\overline{g}(u-t)\exp(-2\pi i\omega u)du = \langle f|g_{t,\omega}\rangle$

with $g_{t,\omega}(u) = \exp(2\pi i\omega(u-t))g(u-t)$ for $u\in\mathbb{R}$

Short Time Fourier Transform

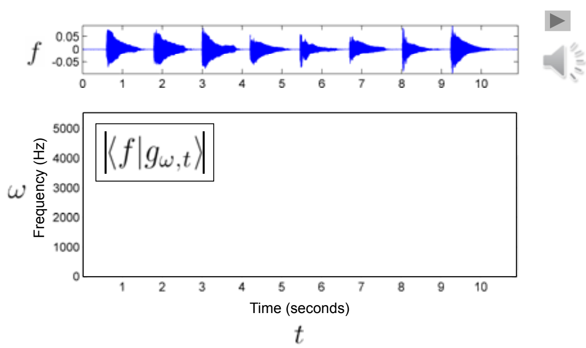
Intuition:

- $g_{t,\omega}$ is "musical note" of frequency ω centered at time t
- Inner product $\langle f|g_{t,\omega}\rangle$ measures the correlation between the musical note $g_{t,\omega}$ and the signal f



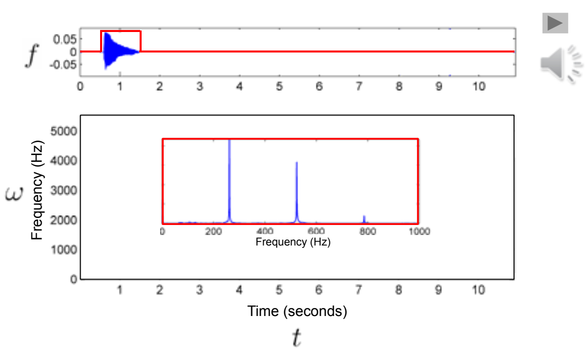
Time-Frequency Representation

Spectrogram



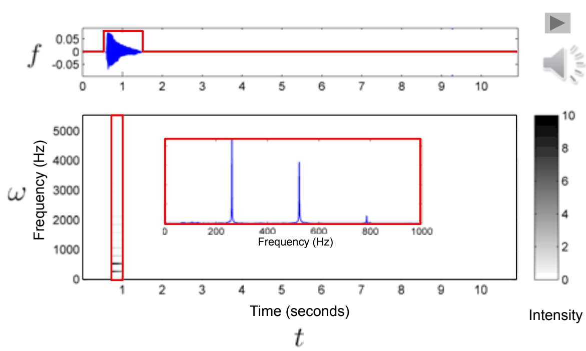
Time-Frequency Representation

Spectrogram



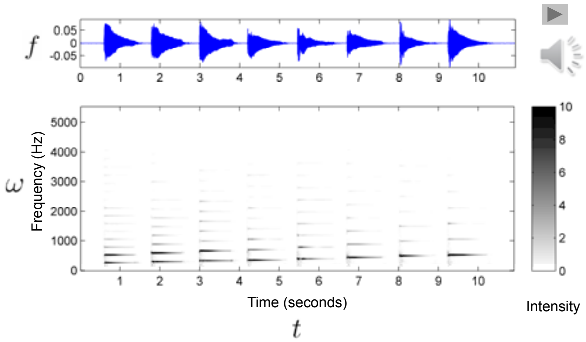
Time-Frequency Representation

Spectrogram



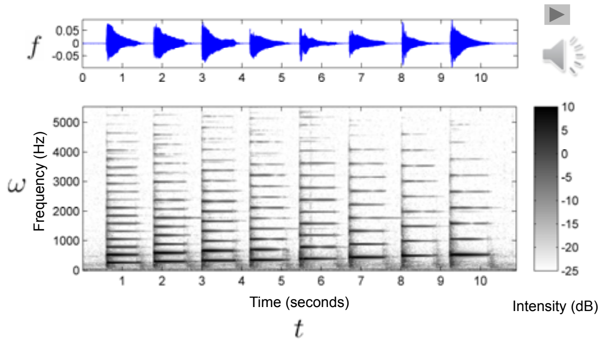
Time-Frequency Representation

Spectrogram



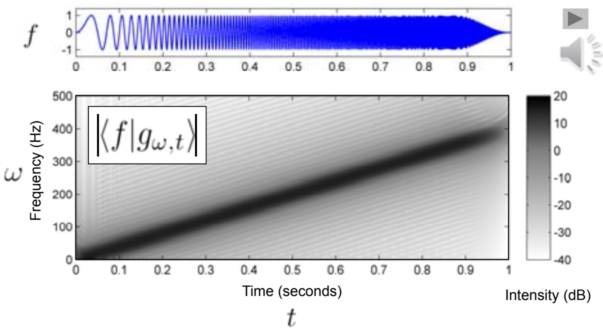
Time-Frequency Representation

Spectrogram



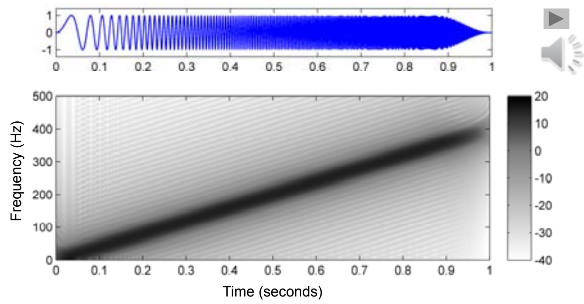
Time-Frequency Representation

Spectrogram



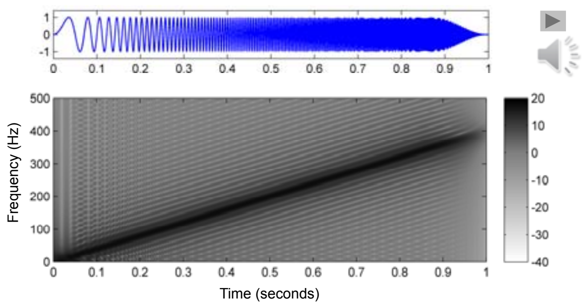
Time-Frequency Representation

Chirp signal and STFT with Hann window of length 50 ms



Time-Frequency Representation

Chirp signal and STFT with box window of length 50 ms



Time-Frequency Representation

Time-Frequency Localization

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

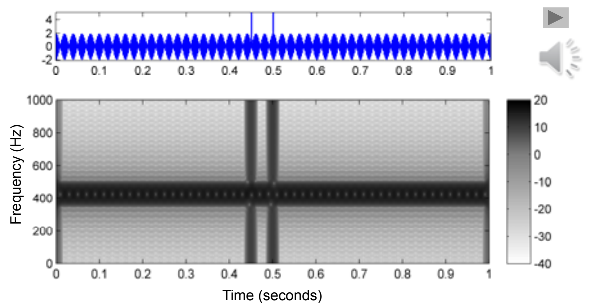
  Large window : poor time resolution
  good frequency resolution

  Small window : good time resolution
  poor frequency resolution

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

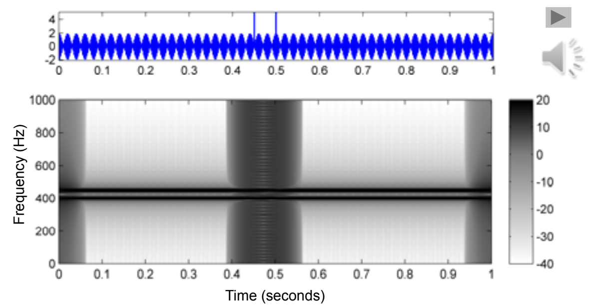
Time-Frequency Representation

Signal and STFT with Hann window of length 20 ms



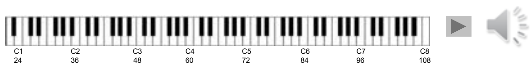
Time-Frequency Representation

Signal and STFT with Hann window of length 100 ms

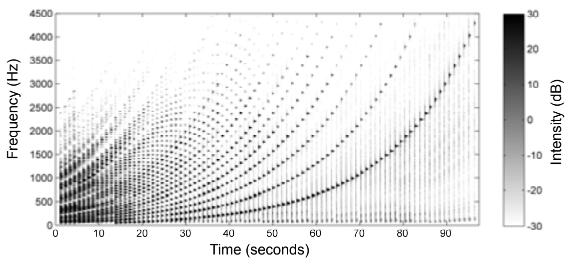


Audio Features

Example: Chromatic scale

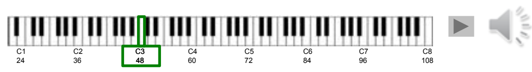


Spectrogram

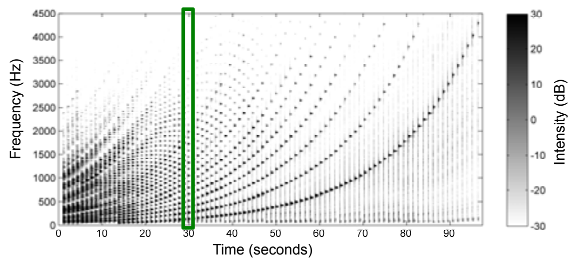


Audio Features

Example: Chromatic scale



Spectrogram



Audio Features

Model assumption: Equal-tempered scale

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

→ Logarithmic frequency distribution
Octave: doubling of frequency

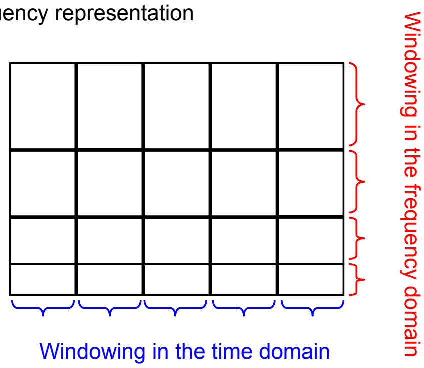
Audio Features

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.

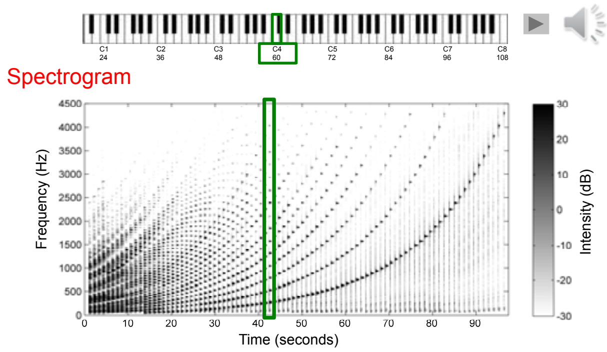
Audio Features

Time-frequency representation



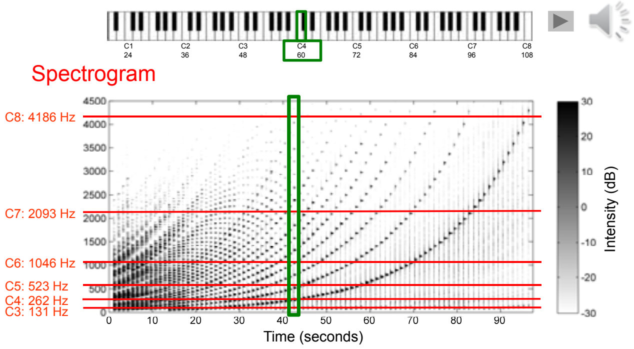
Audio Features

Example: Chromatic scale



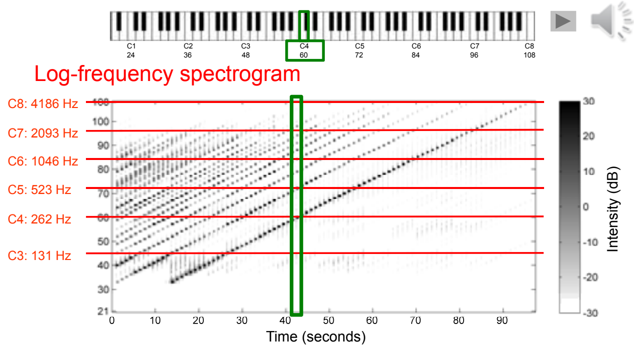
Audio Features

Example: Chromatic scale



Audio Features

Example: Chromatic scale



Audio Features

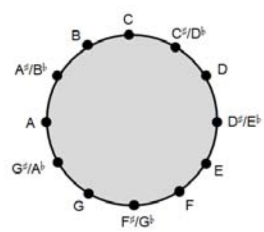
Frequency ranges for pitch-based log-frequency spectrogram

Note	MIDI pitch p	Center [Hz] frequency $F_{\text{pitch}}(p)$	Left [Hz] boundary $F_{\text{pitch}}(p-0.5)$	Right [Hz] boundary $F_{\text{pitch}}(p+0.5)$	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

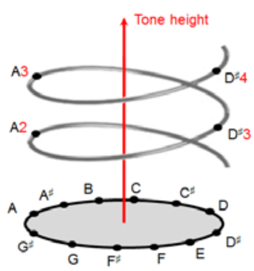
Audio Features

Chroma features

Chromatic circle



Shepard's helix of pitch



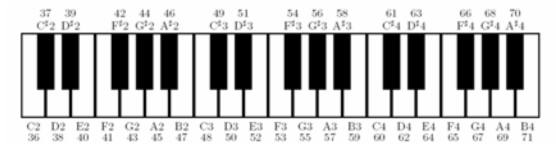
Audio Features

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.

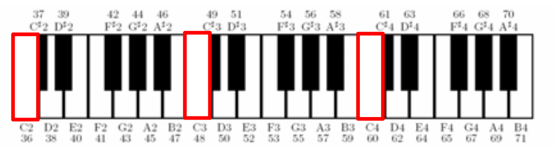
Audio Features

Chroma features



Audio Features

Chroma features

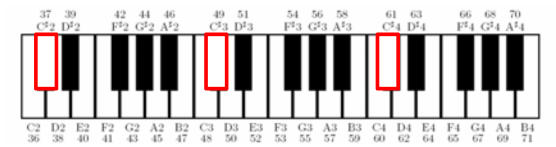


C2 C3 C4

Chroma C

Audio Features

Chroma features

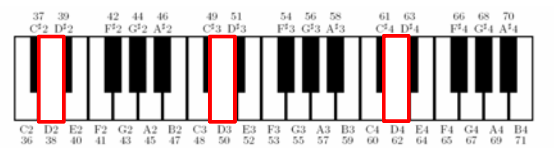


C#2 C#3 C#4

Chroma C#

Audio Features

Chroma features

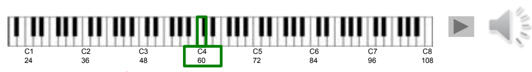


D2 D3 D4

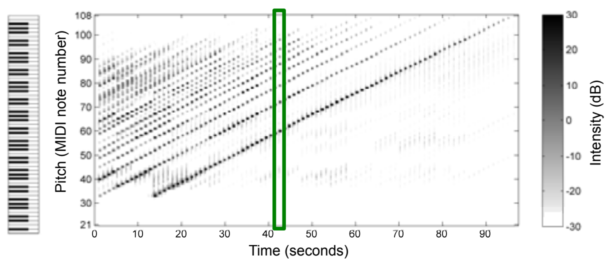
Chroma D

Audio Features

Example: Chromatic scale

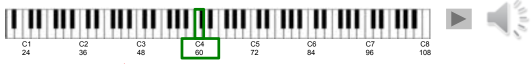


Log-frequency spectrogram

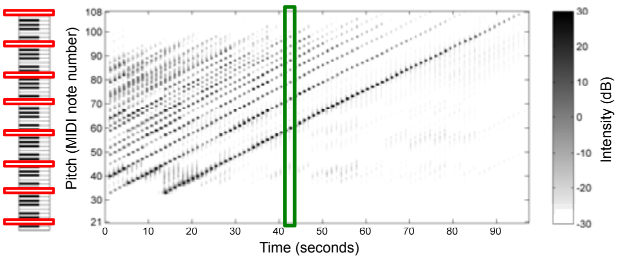


Audio Features

Example: Chromatic scale



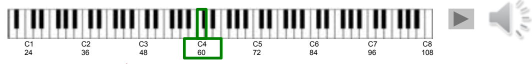
Log-frequency spectrogram



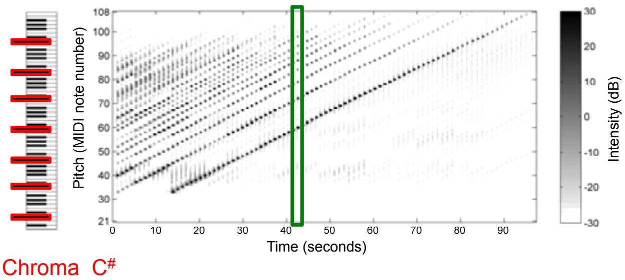
Chroma C

Audio Features

Example: Chromatic scale



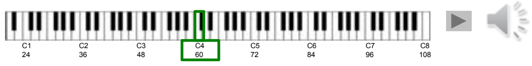
Log-frequency spectrogram



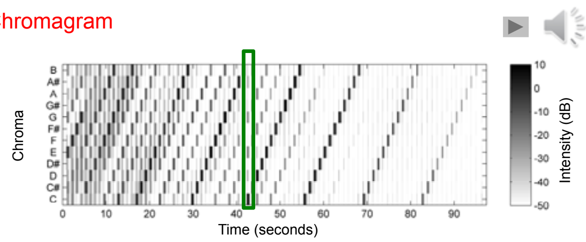
Chroma $C^{\#}$

Audio Features

Example: Chromatic scale

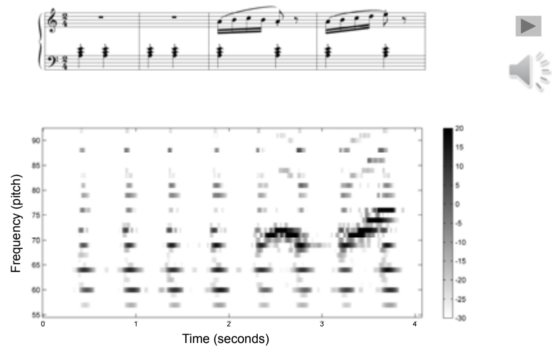


Chromagram



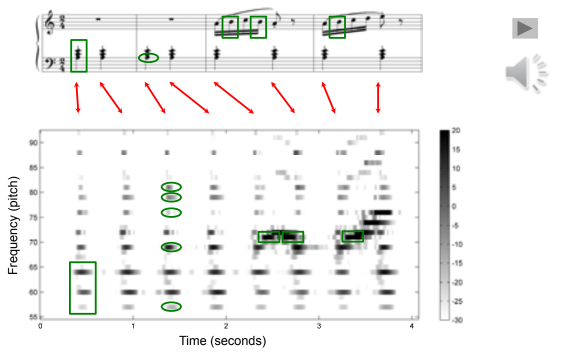
Audio Features

Chroma features



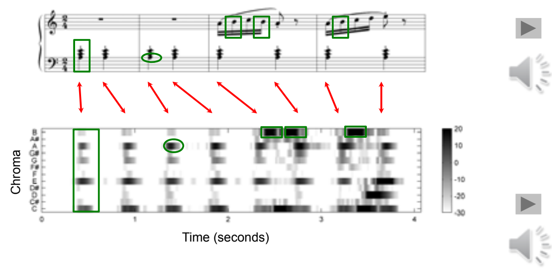
Audio Features

Chroma features



Audio Features

Chroma features



Audio Features

Chroma features

- Sequence of chroma vectors correlates to the harmonic progression
- Normalization $x \rightarrow x/\|x\|$ makes features invariant to changes in dynamics
- Further denoising and smoothing
- Taking logarithm before adding up pitch coefficients accounts for logarithmic sensation of intensity

Audio Features

Logarithmic compression

For a positive constant $\gamma \in \mathbb{R}_{>0}$
the **logarithmic compression**

$$\Gamma_\gamma : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0}$$

is defined by

$$\Gamma_\gamma(v) := \log(1+\gamma \cdot v)$$

A value $v \in \mathbb{R}_{>0}$ is replaced
by a compressed value $\Gamma_\gamma(v)$

Audio Features

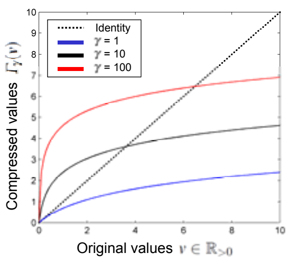
Logarithmic compression

For a positive constant $\gamma \in \mathbb{R}_{>0}$
the **logarithmic compression**

$$\Gamma_\gamma : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0}$$

is defined by

$$\Gamma_\gamma(v) := \log(1+\gamma \cdot v)$$

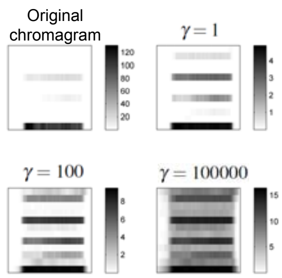


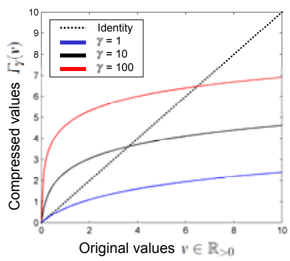
A value $v \in \mathbb{R}_{>0}$ is replaced
by a compressed value $\Gamma_\gamma(v)$

The higher $\gamma \in \mathbb{R}_{>0}$
the stronger the compression

Audio Features

Logarithmic compression





A value $v \in \mathbb{R}_{>0}$ is replaced
by a compressed value $\Gamma_\gamma(v)$

The higher $\gamma \in \mathbb{R}_{>0}$
the stronger the compression

Audio Features

Normalization

Replace a vector
by the normalized vector

$$x/\|x\|$$

using a suitable norm $\|\cdot\|$

Example:
Chroma vector $x \in \mathbb{R}^{12}$
Euclidean norm

$$\|x\| := \left(\sum_{i=0}^{11} |x(i)|^2\right)^{1/2}$$

Audio Features

Normalization

Replace a vector
by the normalized vector

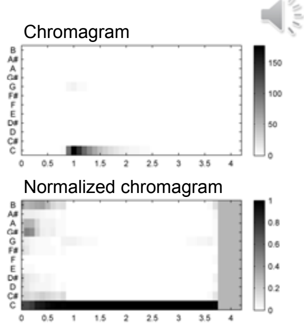
$$x/\|x\|$$

using a suitable norm $\|\cdot\|$

Example:
Chroma vector $x \in \mathbb{R}^{12}$
Euclidean norm

$$\|x\| := \left(\sum_{i=0}^{11} |x(i)|^2\right)^{1/2}$$

Example: C4 played by piano



Audio Features

Normalization

Replace a vector
by the normalized vector

$$x/\|x\|$$

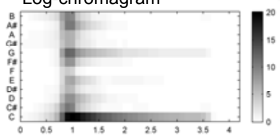
using a suitable norm $\|\cdot\|$

Example:
Chroma vector $x \in \mathbb{R}^{12}$
Euclidean norm

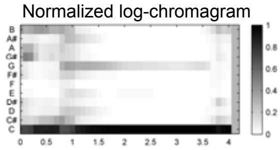
$$\|x\| := \left(\sum_{i=0}^{11} |x(i)|^2\right)^{1/2}$$

Example: C4 played by piano

Log-chromagram

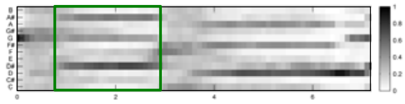


Normalized log-chromagram

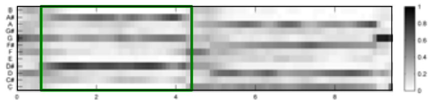


Audio Features

Chroma features (normalized)



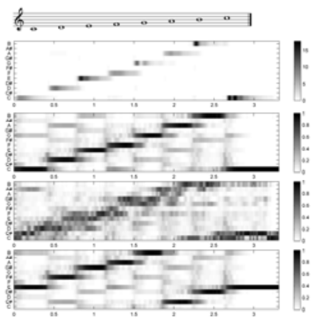
Karajan



Scherbakov

Audio Features

Chroma features



Chromagram

Chromagram after logarithmic compression and normalization

Chromagram based on a piano tuned 40 cents upwards

Chromagram after applying a cyclic shift of four semitones upwards

Audio Features

- There are many ways to implement chroma features
- Properties may differ significantly
- Appropriateness depends on respective application
- Chroma Toolbox (MATLAB)
  https://www.audiolabs-erlangen.de/resources/MIR/chromatoolbox
- LibROSA (Python)
  https://librosa.github.io/librosa/
- Feature learning: “Deep Chroma”
  [Korzeniowski/Widmer, ISMIR 2016]

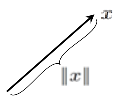
Additional Material

Inner Product

$$\langle x|y\rangle := \sum_{n=0}^{N-1} x(n)\overline{y(n)} \quad \text{for} \quad x, y \in \mathbb{C}^N$$

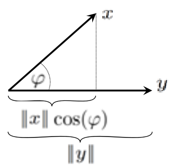
Length of a vector

$$\|x\| := \sqrt{\langle x|x\rangle}$$



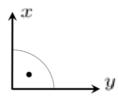
Angle between two vectors

$$\cos(\varphi) = \frac{|\langle x|y\rangle|}{\|x\|\cdot\|y\|}$$



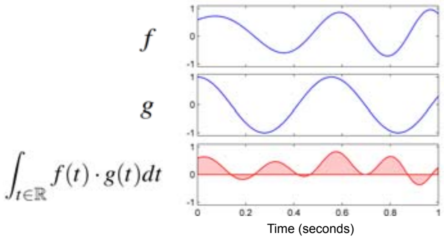
Orthogonality of two vectors

$$\langle x|y\rangle = 0$$



Inner Product

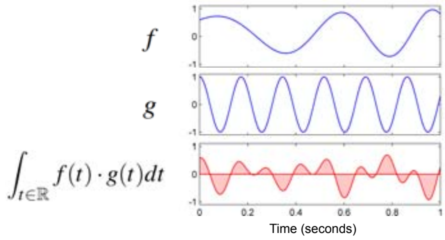
Measuring the similarity of two functions



→ Area mostly positive and large
→ Integral large
→ Similarity high

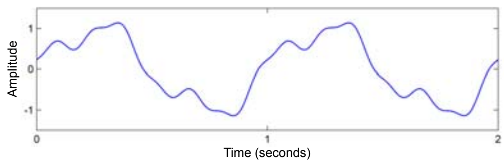
Inner Product

Measuring the similarity of two functions



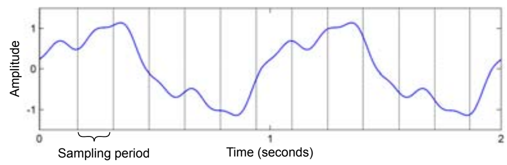
→ Area positive and negative
→ Integral small
→ Similarity low

Discretization



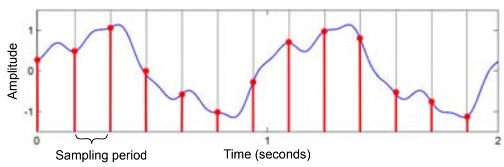
Discretization

Sampling



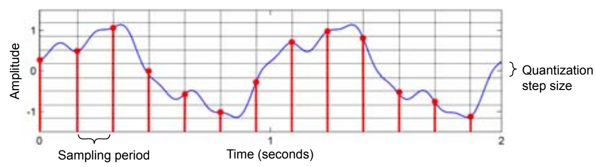
Discretization

Sampling



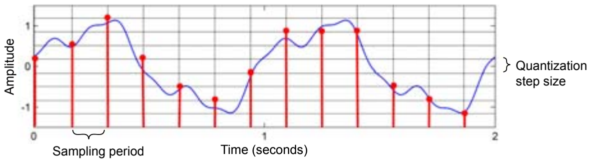
Discretization

Quantization



Discretization

Quantization



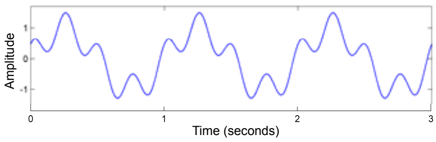
Discretization

Sampling

$f : \mathbb{R} \to \mathbb{R}$	CT-signal
$T > 0$	Sampling period
$x(n) := f(n \cdot T)$	Equidistant sampling, $n \in \mathbb{Z}$
$x : \mathbb{Z} \to \mathbb{R}$	DT-signal
$x(n)$	Sample taken at time $t = n \cdot T$
$F_s := 1/T$	Sampling rate

Discretization

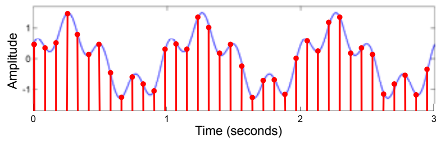
Aliasing



Original signal

Discretization

Aliasing

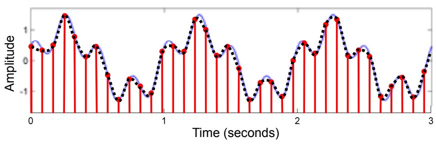


Original signal

Sampled signal using a sampling rate of 12 Hz

Discretization

Aliasing



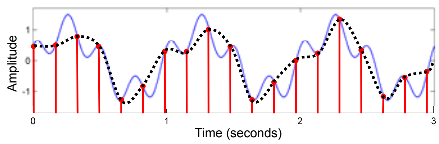
Original signal

Sampled signal using a sampling rate of **12 Hz**

Reconstructed signal

Discretization

Aliasing



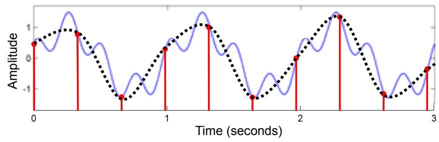
Original signal

Sampled signal using a sampling rate of **6 Hz**

Reconstructed signal

Discretization

Aliasing



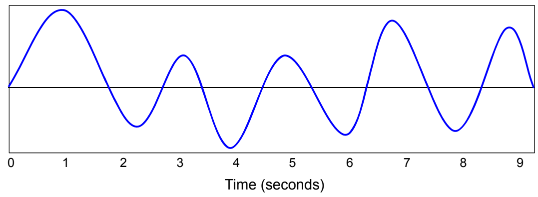
Original signal

Sampled signal using a sampling rate of **3 Hz**

Reconstructed signal

Discretization

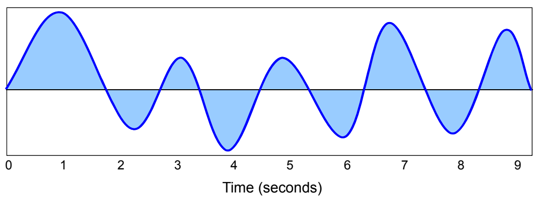
Integrals and Riemann sums



CT-signal f

Discretization

Integrals and Riemann sums

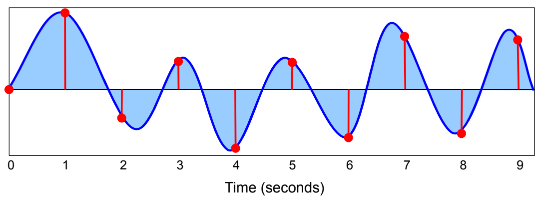


CT-signal f

Integral (total area) $\int_{t\in\mathbb{R}} f(t)\; dt$

Discretization

Integrals and Riemann sums



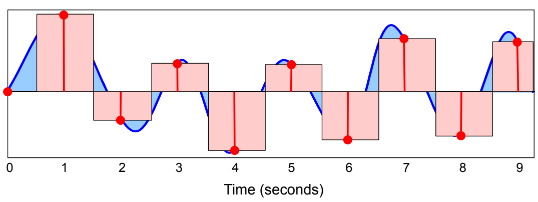
CT-signal f

Integral (total area) $\int_{t\in\mathbb{R}} f(t)\; dt$

DT-signals (obtained by 1-sampling) x

Discretization

Integrals and Riemann sums



CT-signal f

Integral (total area) $\int_{t\in\mathbb{R}} f(t)\; dt \approx \sum_{n\in\mathbb{Z}} x(n)$

DT-signals (obtained by 1-sampling) x

Riemann sum (total area) $\rightarrow$ **Approximation of integral**

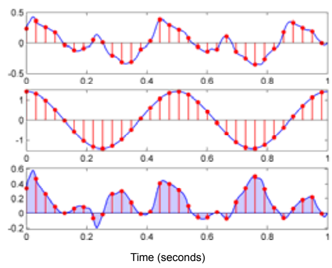
Discretization

Integrals and Riemann sums

First CT-signal and DT-signal

Second CT-signal and DT-signal

Product of CT-signals and DT-signals



f x

g y

$f \cdot g$ $x \cdot y$

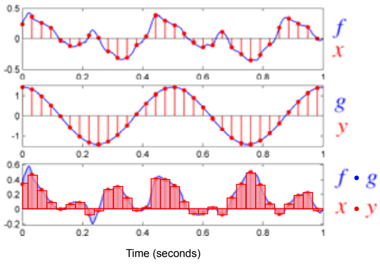
Discretization

Integrals and Riemann sums

First CT-signal
and DT-signal

Second CT-signal
and DT-signal

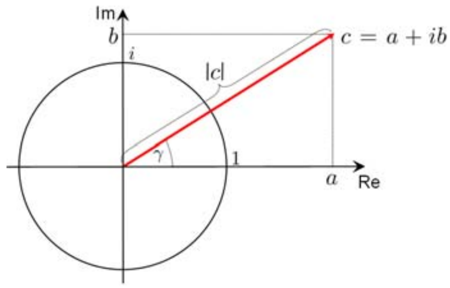
Product of CT-signals
and DT-signals



Integral $\approx$ **Riemann sum** $\quad \int_{t\in\mathbb{R}} f(t)\overline{g(t)}dt \approx \sum_{n\in\mathbb{Z}} x(n)\overline{y(n)}$

Exponential Function

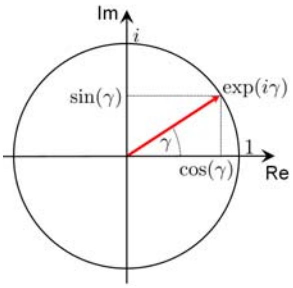
Polar coordinate representation of a complex number



Exponential Function

Real and imaginary part (Euler's formula)

$\exp(i\gamma) = \cos(\gamma) + i\sin(\gamma)$



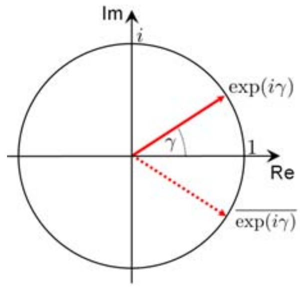
$|\exp(i\gamma)| = 1$

$\exp(i\gamma) = \exp(i(\gamma + 2\pi))$

Exponential Function

Complex conjugate number

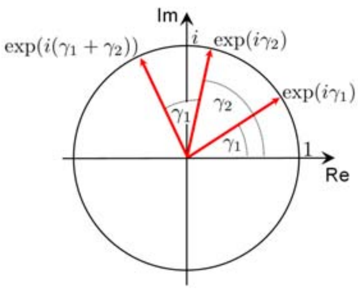
$\overline{\exp(i\gamma)} = \exp(-i\gamma)$



Exponential Function

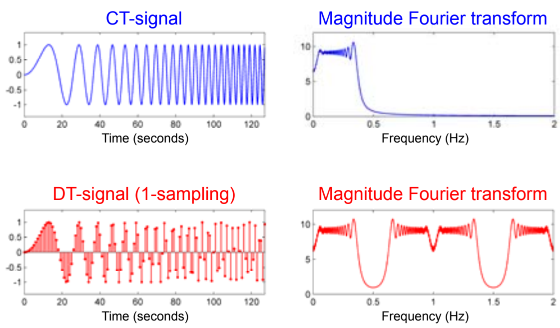
Additivity property

$\exp(i(\gamma_1 + \gamma_2)) = \exp(i\gamma_1)\exp(i\gamma_2)$



Fourier Transform

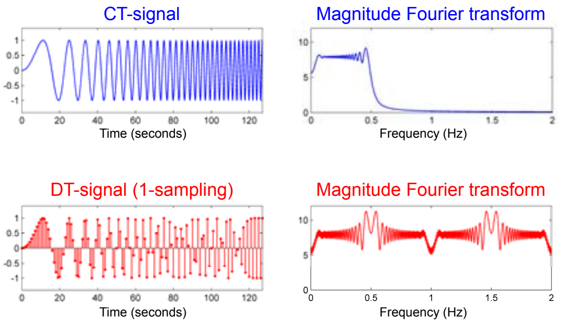
Chirp signal with λ = *0.003* $\quad f(t) := \begin{cases} \sin(\lambda \cdot \pi t^2), & \text{for } t \geq 0 \\ 0, & \text{for } t < 0 \end{cases}$



Fourier Transform

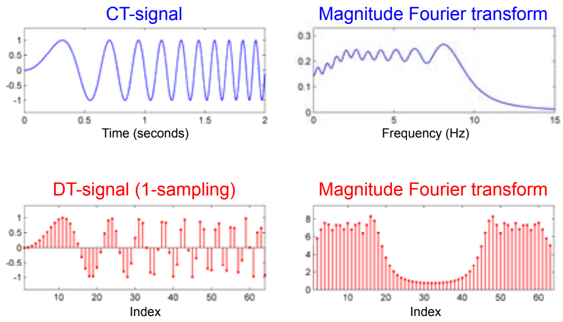
Chirp signal with λ = *0.004*

$$f(t) := \begin{cases} \sin(\lambda \cdot \pi t^2), & \text{for } t \geq 0 \\ 0, & \text{for } t < 0 \end{cases}$$



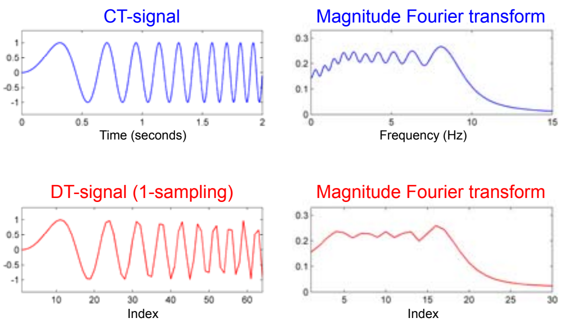
Fourier Transform

DFT approximation of Fourier transform



Fourier Transform

DFT approximation of Fourier transform



Fourier Transform

Discrete STFT

$$\mathcal{X}(m,k) := \sum_{n=0}^{N-1} x(n+mH)w(n)\exp(-2\pi ikn/N)$$

$x:\mathbb{Z}\to\mathbb{R}$	DT-signal
$w:[0:N-1]\to\mathbb{R}$	Window function of length $N\in\mathbb{N}$
$H\in\mathbb{N}$	Hop size
$K=N/2$	Index corresponding to Nyquist frequency
$\mathcal{X}(m,k)$	Fourier coefficient for frequency index $k\in[0:K]$ and time frame $m\in\mathbb{Z}$

Fourier Transform

Discrete STFT

$$\mathcal{X}(m,k) := \sum_{n=0}^{N-1} x(n+mH)w(n)\exp(-2\pi ikn/N)$$

Physical time position associated with $\mathcal{X}(m,k)$:

$$T_{\mathrm{coef}}(m) := \frac{m\cdot H}{F_s} \quad \text{(seconds)}$$

H = Hop size

F_s = Sampling rate

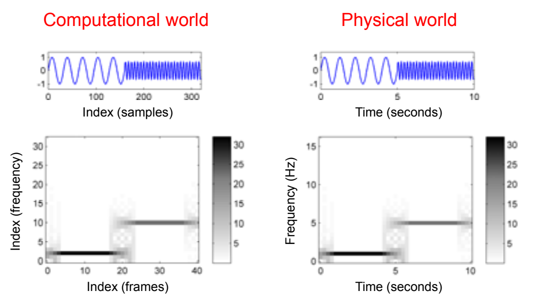
Physical frequency associated with $\mathcal{X}(m,k)$:

$$F_{\mathrm{coef}}(k) := \frac{k\cdot F_s}{N} \quad \text{(Hertz)}$$

Fourier Transform

Discrete STFT

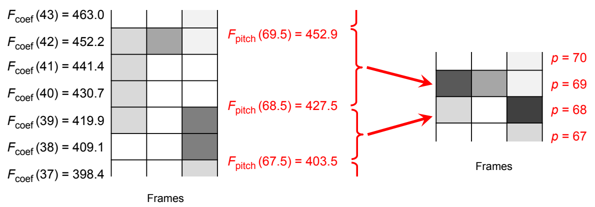
Parameters
N = 64
H = 8
F_s = 32 Hz



Log-Frequency Spectrogram

Pooling procedure for discrete STFT

Parameters
N = 4096
H = 2048
F_s = 44100 Hz



Fast Fourier Transform

Algorithm: FFT

Input: The length $N = 2^L$ with N being a power of two
The vector $(x(0),\dots,x(N-1))^\top \in \mathbb{C}^N$

Output: The vector $(X(0),\dots,X(N-1))^\top = \mathrm{DFT}_N \cdot (x(0),\dots,x(N-1))^\top$

Procedure: Let $(X(0),\dots,X(N-1)) = \mathrm{FFT}(N, x(0),\dots,x(N-1))$ denote the general form of the FFT algorithm.
If $N = 1$ then
$X(0) = x(0)$.
Otherwise compute recursively:
$(A(0),\dots,A(N/2-1)) = \mathrm{FFT}(N/2, x(0), x(2), x(4),\dots,x(N-2))$,
$(B(0),\dots,B(N/2-1)) = \mathrm{FFT}(N/2, x(1), x(3), x(5),\dots,x(N-1))$,
$C(k) = \omega_N^k \cdot B(k)$ for $k \in [0:N/2-1]$,
$X(k) = A(k) + C(k)$ for $k \in [0:N/2-1]$,
$X(N/2+k) = A(k) - C(k)$ for $k \in [0:N/2-1]$.

Signal Spaces and Fourier Transforms

Signal space	$L^2(\mathbb{R})$	$L^2([0,1))$	$\ell^2(\mathbb{Z})$
Inner product	$\langle f\|g\rangle = \int_{t\in\mathbb{R}} f(t)\overline{g(t)}dt$	$\langle f\|g\rangle = \int_{t\in[0,1)} f(t)\overline{g(t)}dt$	$\langle x\|y\rangle = \sum_{n\in\mathbb{Z}} x(n)\overline{y(n)}$
Norm	$\\|f\\|_2 = \sqrt{\langle f\|f\rangle}$	$\\|f\\|_2 = \sqrt{\langle f\|f\rangle}$	$\\|x\\|_2 = \sqrt{\langle x\|x\rangle}$
Definition	$L^2(\mathbb{R}) :=$ $\{f:\mathbb{R}\to\mathbb{C} \mid \\|f\\|_2 < \infty\}$	$L^2([0,1)) :=$ $\{f:[0,1)\to\mathbb{C} \mid \\|f\\|_2 < \infty\}$	$\ell^2(\mathbb{Z}) :=$ $\{f:\mathbb{Z}\to\mathbb{C} \mid \\|x\\|_2 < \infty\}$
Elementary frequency function	$\mathbb{R}\to\mathbb{C}$ $t \mapsto \exp(2\pi i\omega t)$	$[0,1)\to\mathbb{C}$ $t \mapsto \exp(2\pi ikt)$	$\mathbb{Z}\to\mathbb{C}$ $n \mapsto \exp(2\pi i\omega n)$
Frequency parameter	$\omega\in\mathbb{R}$	$k\in\mathbb{Z}$	$\omega\in[0,1)$
Fourier representation	$f(t) =$ $\int_{\omega\in\mathbb{R}} c_\omega \exp(2\pi i\omega t)d\omega$	$f(t) =$ $\sum_{k\in\mathbb{Z}} c_k \exp(2\pi ikt)$	$x(n) =$ $\int_{\omega\in[0,1)} c_\omega \exp(2\pi i\omega n)d\omega$
Fourier transform	$\hat{f}:\mathbb{R}\to\mathbb{C}$ $\hat{f}(\omega) = c_\omega =$ $\int_{t\in\mathbb{R}} f(t)\exp(-2\pi i\omega t)dt$	$\hat{f}:\mathbb{Z}\to\mathbb{C}$ $\hat{f}(k) = c_k =$ $\int_{t\in[0,1)} f(t)\exp(-2\pi ikt)dt$	$\hat{x}:[0,1)\to\mathbb{C}$ $\hat{x}(\omega) = c_\omega =$ $\sum_{n\in\mathbb{Z}} x(n)\exp(-2\pi i\omega n)$