

Lecture

Music Processing

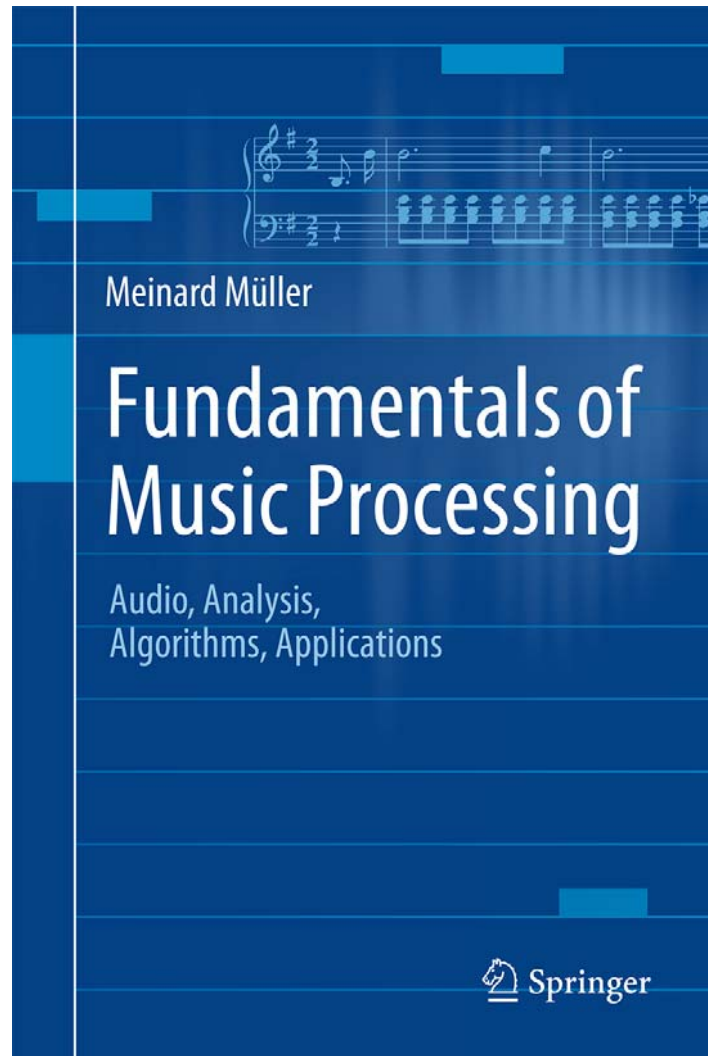
Audio Features

Meinard Müller

International Audio Laboratories Erlangen

meinard.mueller@audiolabs-erlangen.de

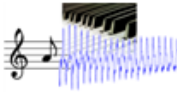
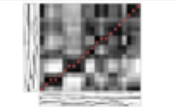
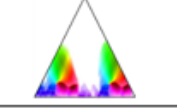
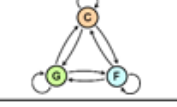
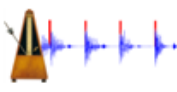
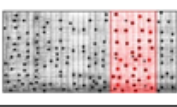
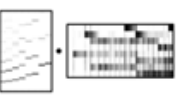
Book: Fundamentals of Music Processing



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

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

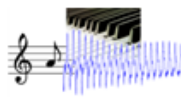
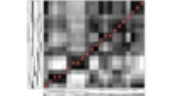
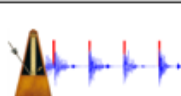
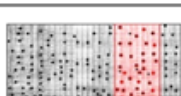
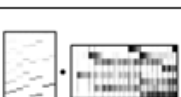
ISBN: 978-3-319-21944-8

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

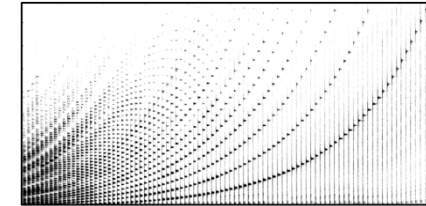
Springer, 2015

Accompanying website:

www.music-processing.de

Chapter 2: Fourier Analysis of Signals

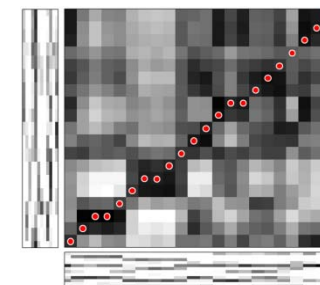
- 2.1 The Fourier Transform in a Nutshell
- 2.2 Signals and Signal Spaces
- 2.3 Fourier Transform
- 2.4 Discrete Fourier Transform (DFT)
- 2.5 Short-Time Fourier Transform (STFT)
- 2.6 Further Notes



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



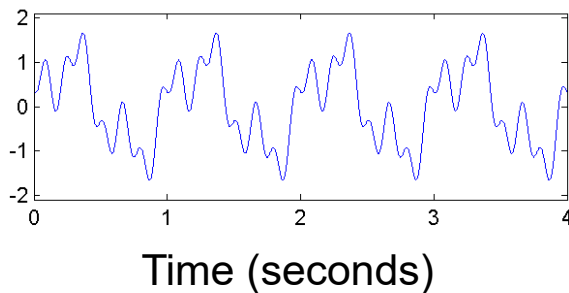
As a first music processing task, we study in Chapter 3 the problem of music synchronization. The objective is to temporally align compatible representations of the same piece of music. Considering this scenario, we explain the need for musically informed audio features. In particular, we introduce the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

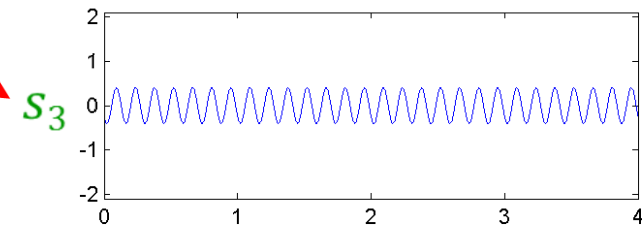
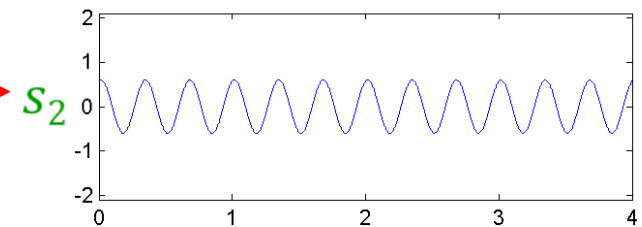
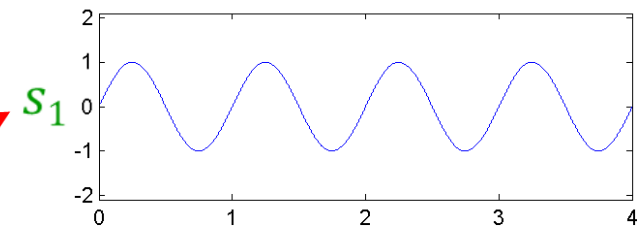
Idea: **Decompose** a given **signal** into a superposition of **sinusoids** (elementary signals).

$$f = s_1 + s_2 + s_3$$

Signal f



Sinusoids



Time (seconds)

Fourier Transform

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

$$s(A, \omega, \varphi)(t) = A \cdot \sin(2\pi(\omega t - \varphi))$$

ω = frequency

A = amplitude

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 .

$$A_1 = 1$$

$$\omega_1 = 1$$

$$\varphi_1 = 0$$

$$A_2 = 0.6$$

$$\omega_2 = 3$$

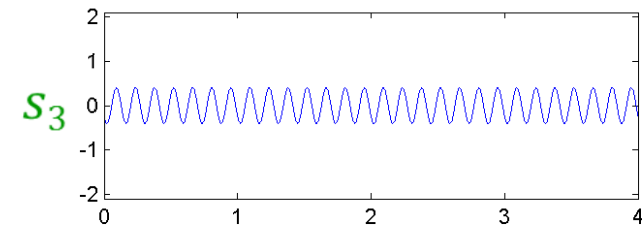
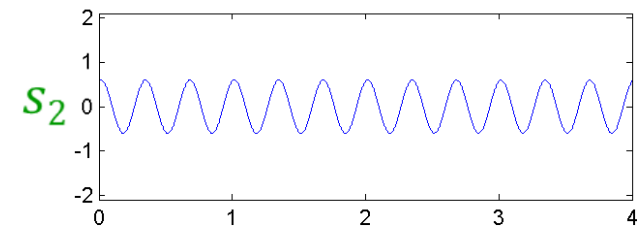
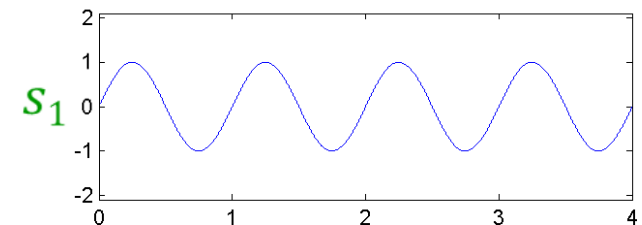
$$\varphi_2 = -0.2$$

$$A_3 = 0.4$$

$$\omega_3 = 7$$

$$\varphi_3 = 0.4$$

Sinusoids



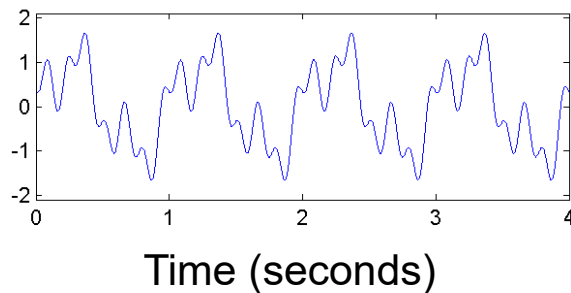
Time (seconds)

Fourier Transform

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

$$f = s_1 + s_2 + s_3$$

Signal f

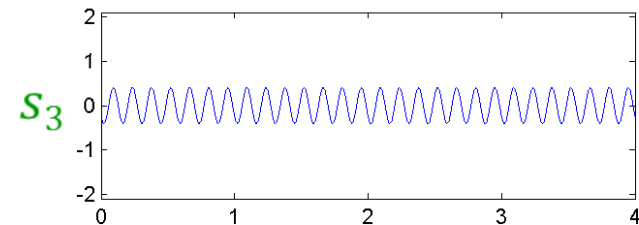
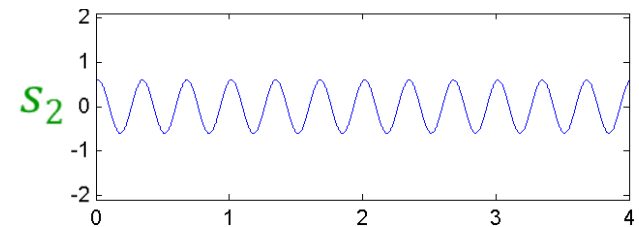
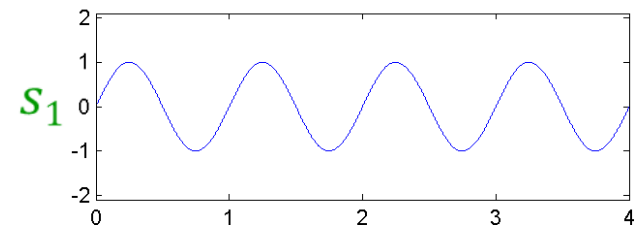


$$\begin{aligned} A_1 &= 1 \\ \omega_1 &= 1 \\ \varphi_1 &= 0 \end{aligned}$$

$$\begin{aligned} A_2 &= 0.6 \\ \omega_2 &= 3 \\ \varphi_2 &= -0.2 \end{aligned}$$

$$\begin{aligned} A_3 &= 0.4 \\ \omega_3 &= 7 \\ \varphi_3 &= 0.4 \end{aligned}$$

Sinusoids



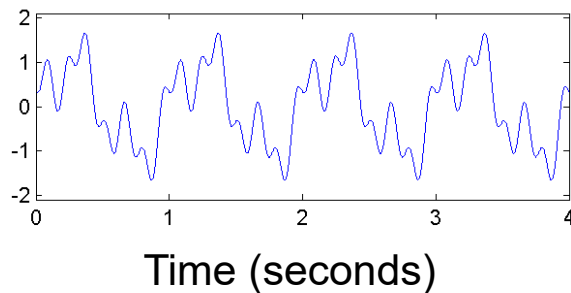
Time (seconds)

Fourier Transform

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

$$f = s_1 + s_2 + s_3$$

Signal f



$$A_1 = 1$$

$$\omega_1 = 1$$

$$\varphi_1 = 0$$

$$A_2 = 0.6$$

$$\omega_2 = 3$$

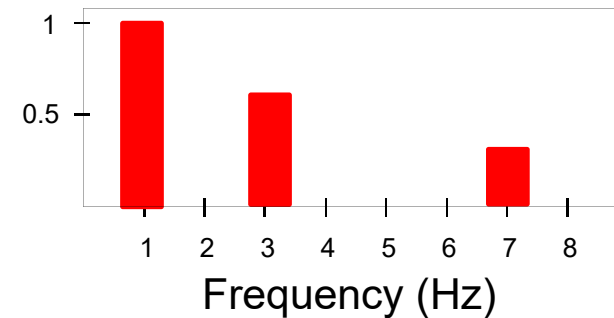
$$\varphi_2 = -0.2$$

$$A_3 = 0.4$$

$$\omega_3 = 7$$

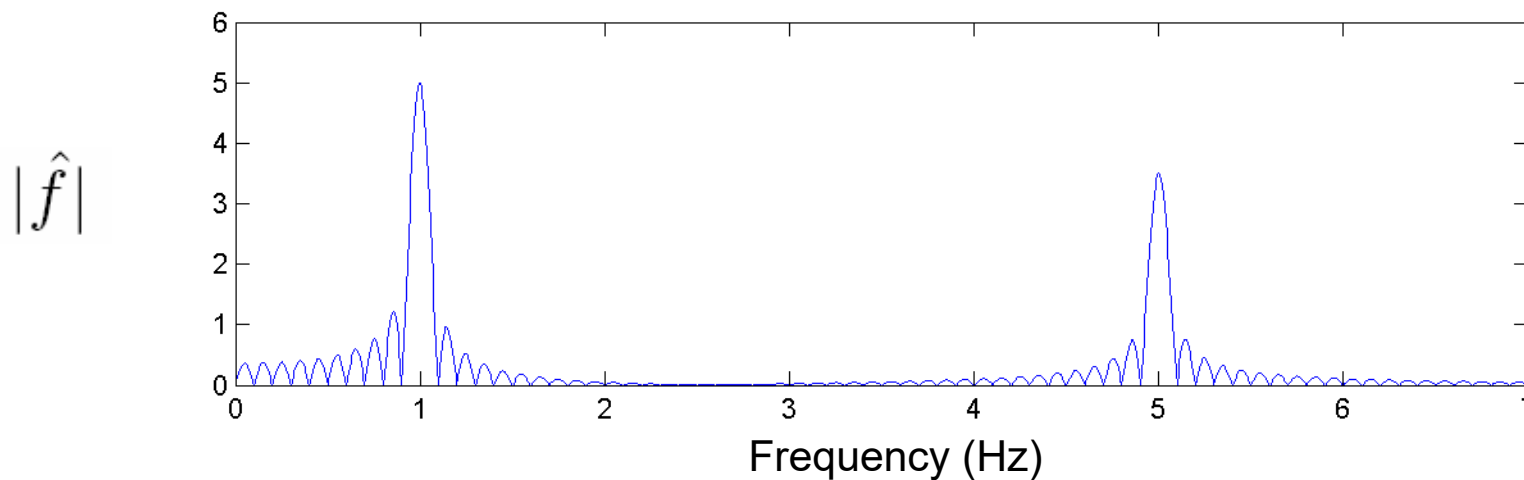
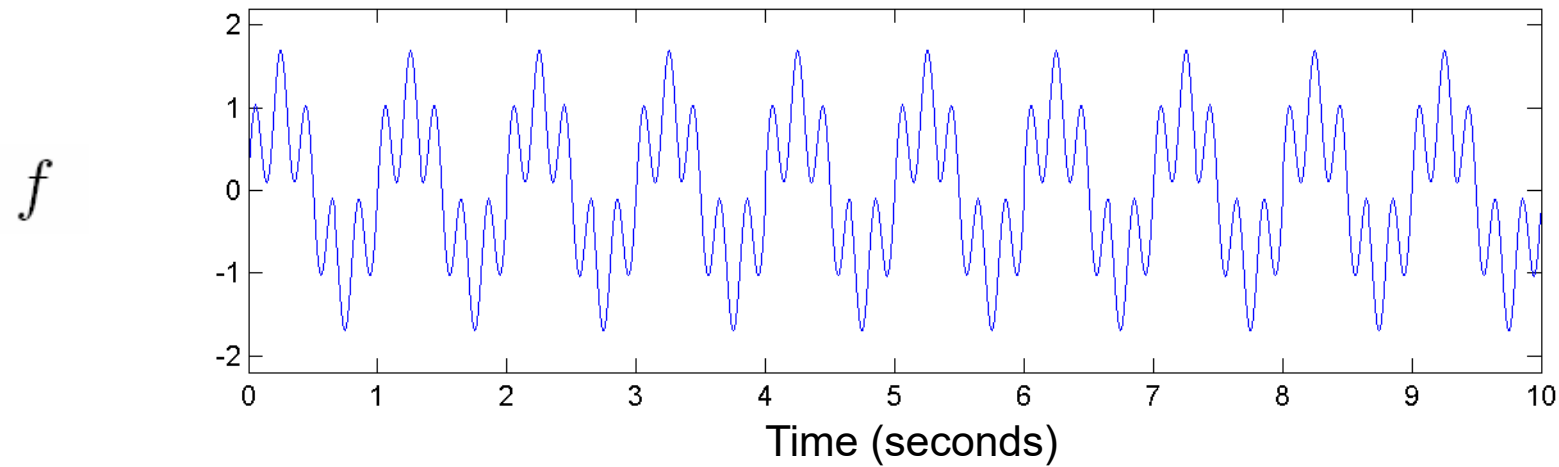
$$\varphi_3 = 0.4$$

Fourier transform $|\hat{f}|$



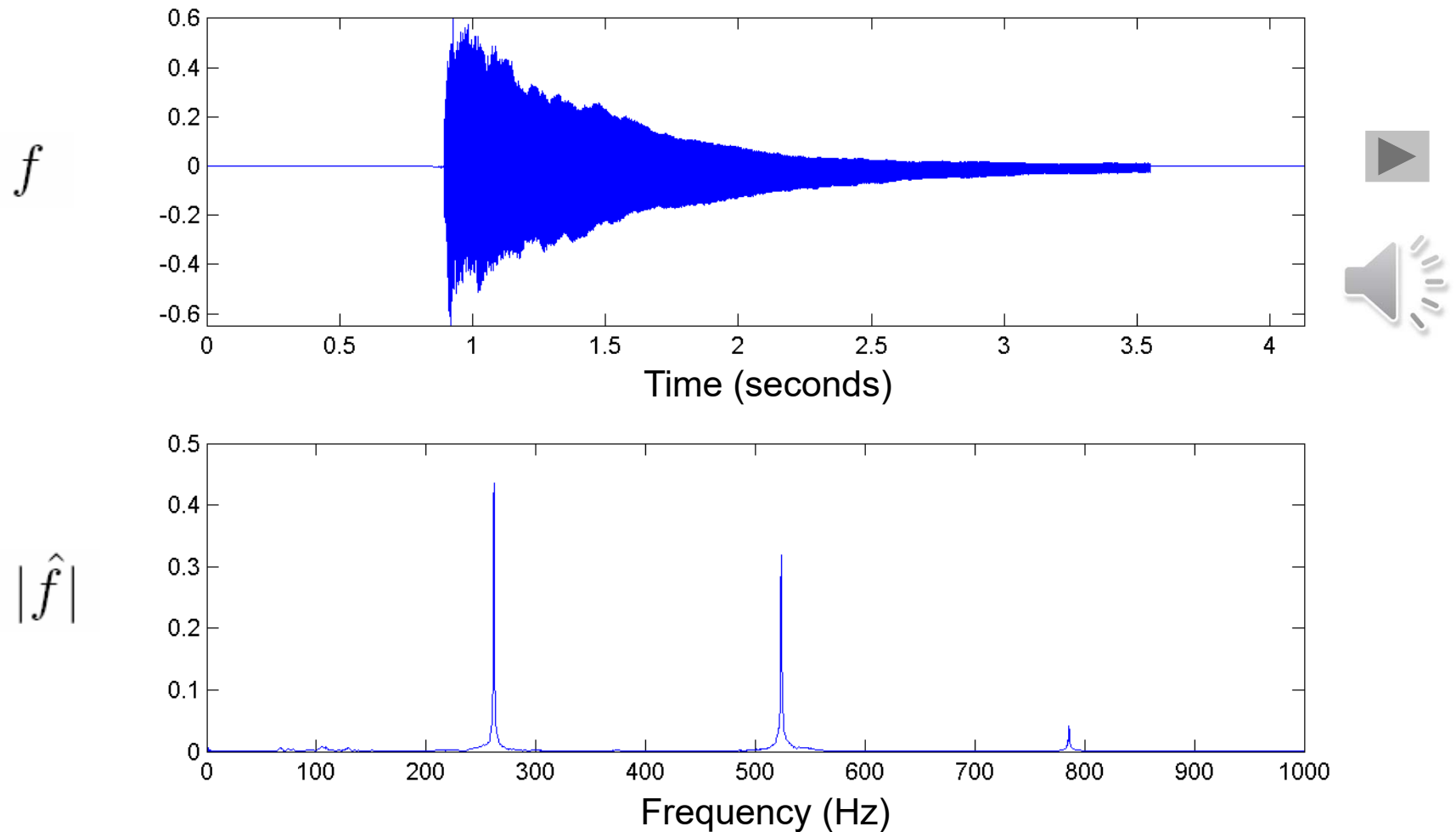
Fourier Transform

Example: Superposition of two sinusoids



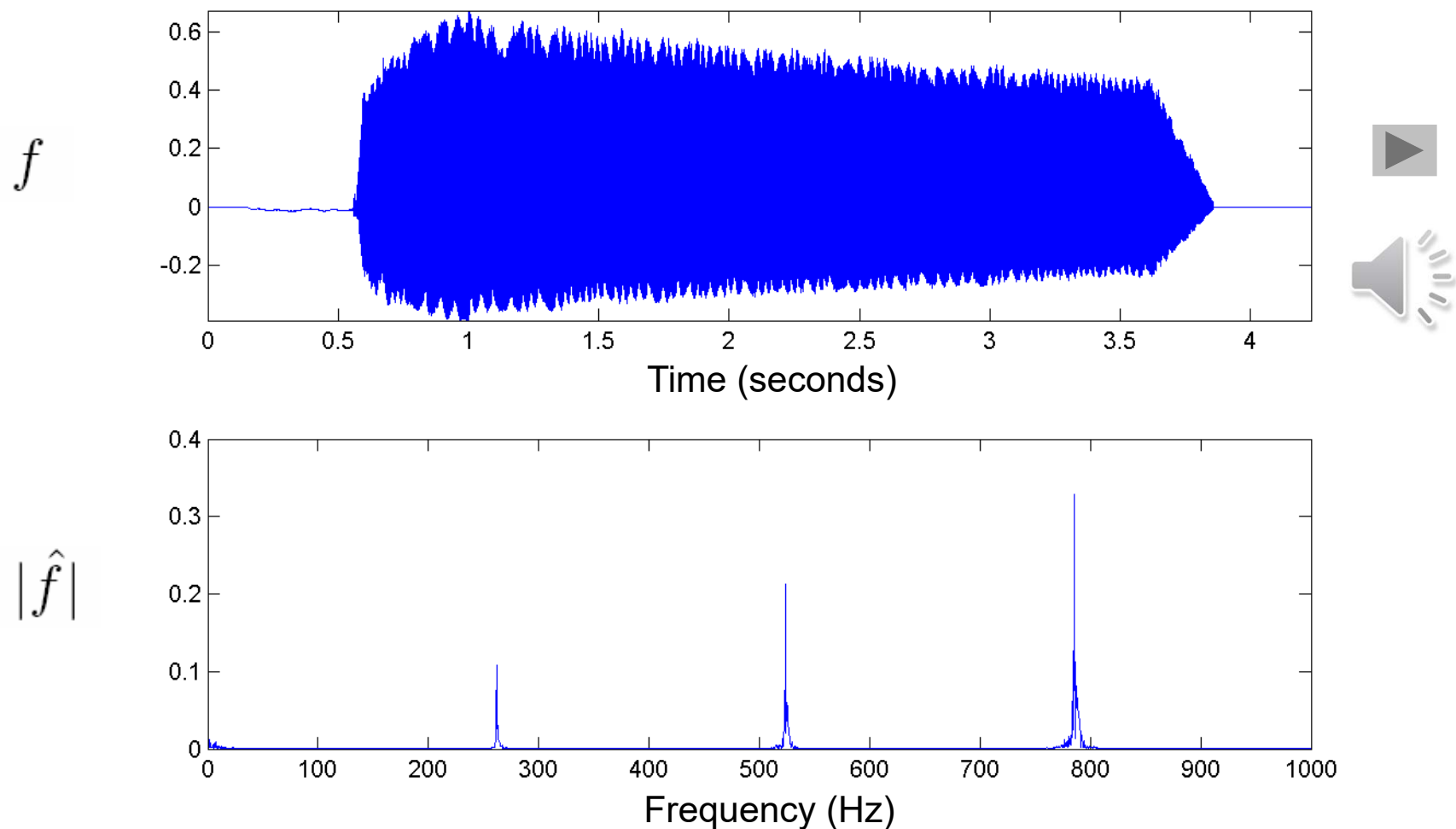
Fourier Transform

Example: C4 played by piano



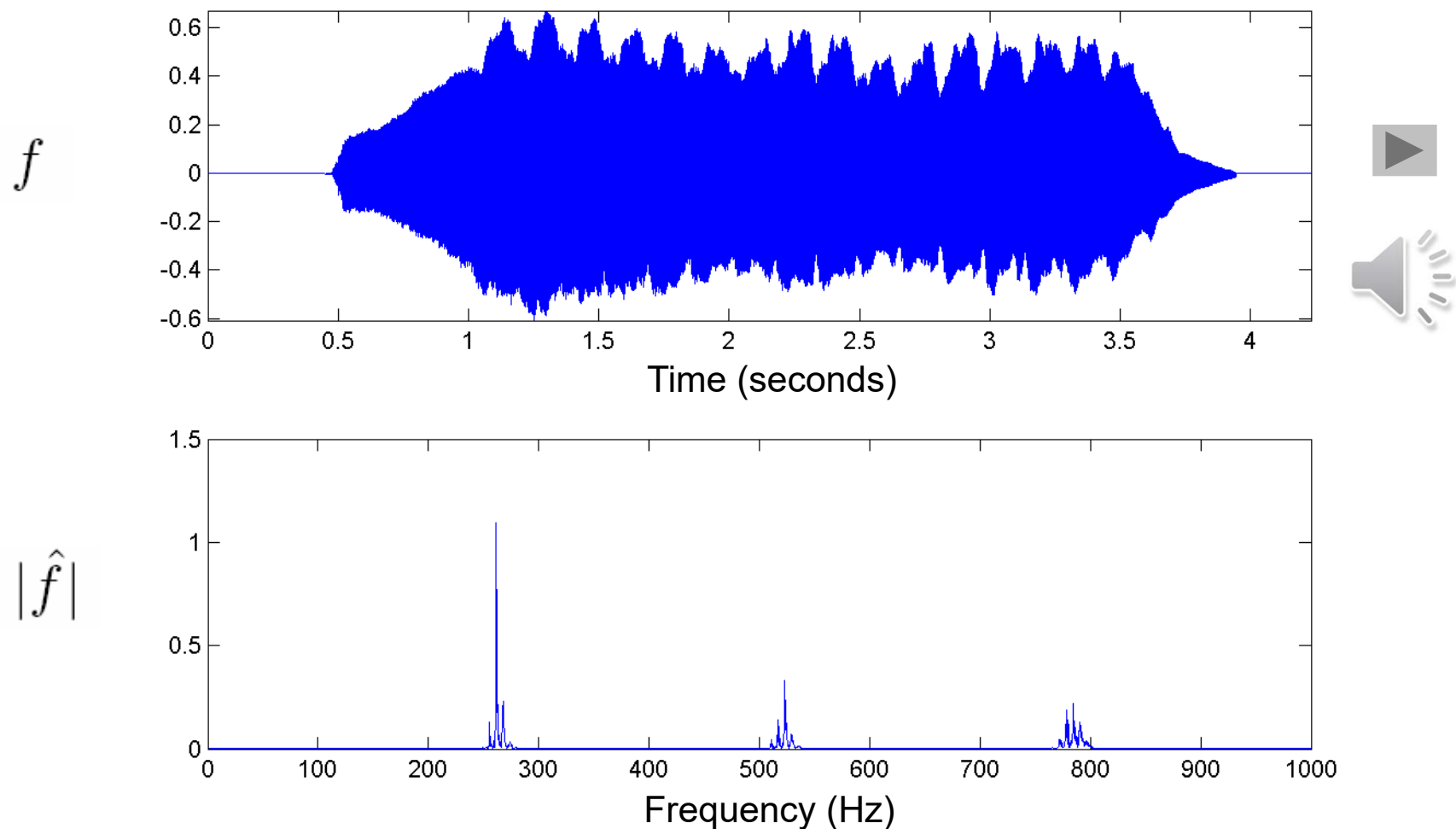
Fourier Transform

Example: C4 played by trumpet



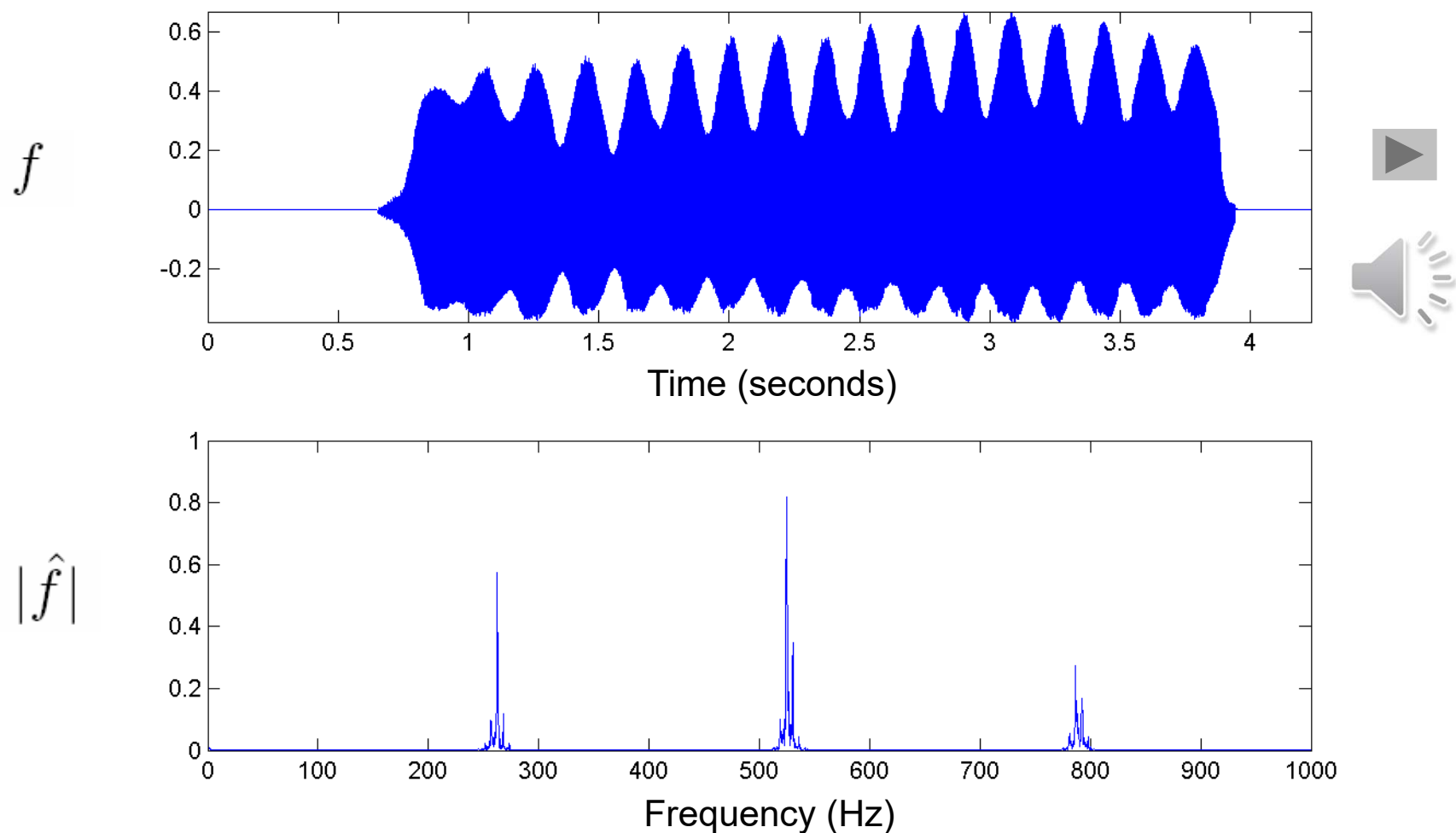
Fourier Transform

Example: C4 played by violin



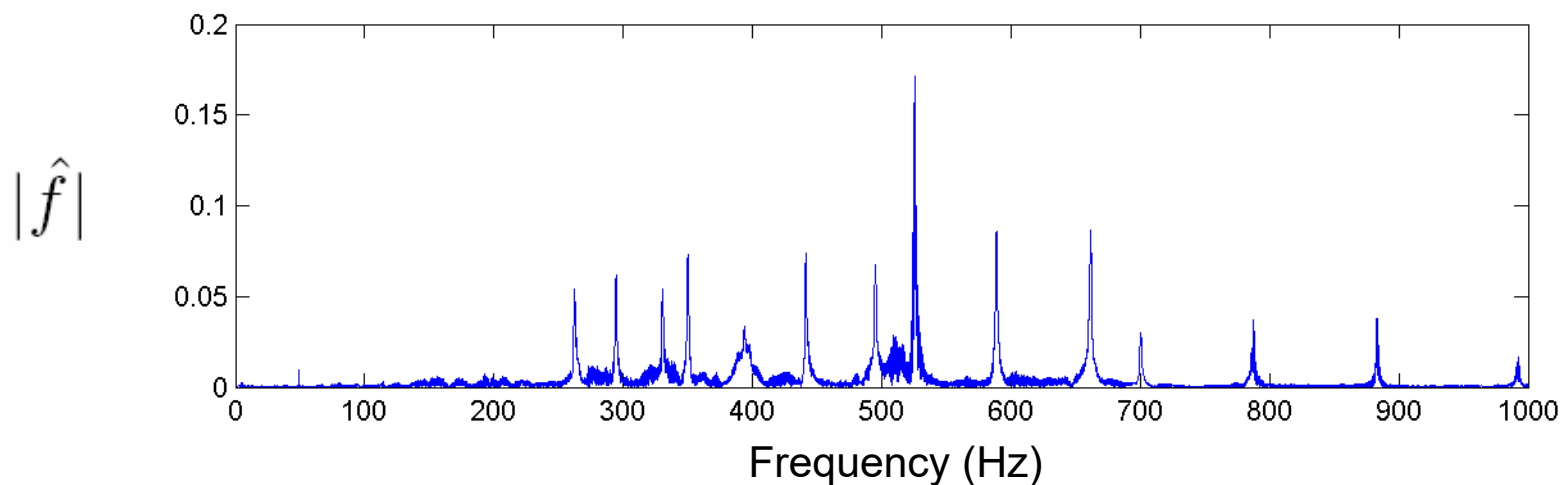
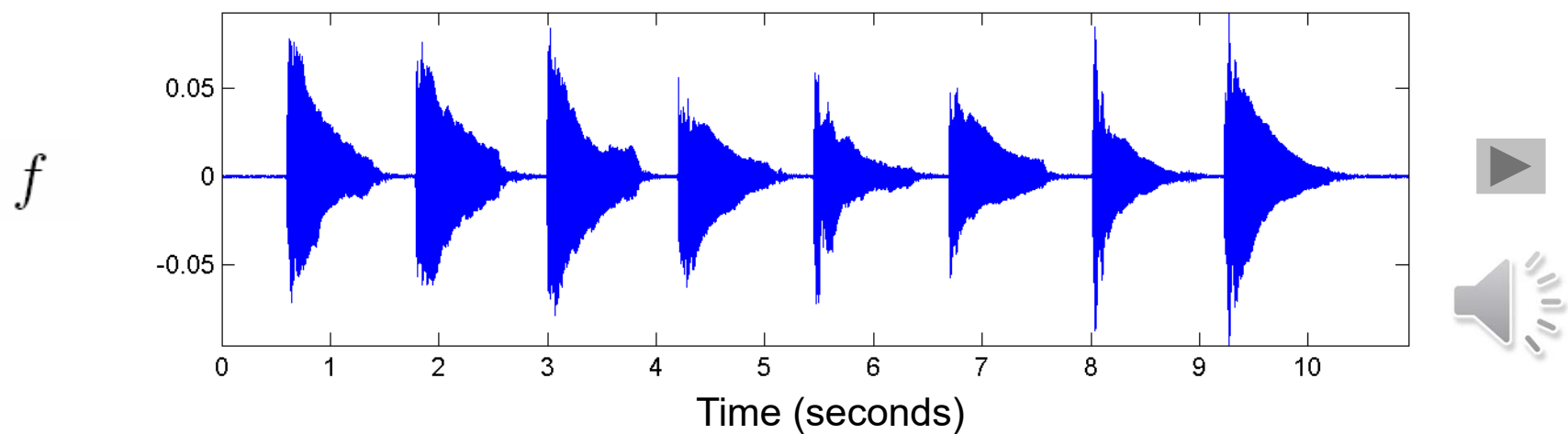
Fourier Transform

Example: C4 played by flute



Fourier Transform

Example: C-major scale (piano)



Fourier Transform

Signal

$$f: \mathbb{R} \rightarrow \mathbb{R}$$

Fourier representation

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

Fourier transform

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

Fourier Transform

Signal

$$f: \mathbb{R} \rightarrow \mathbb{R}$$

Fourier representation

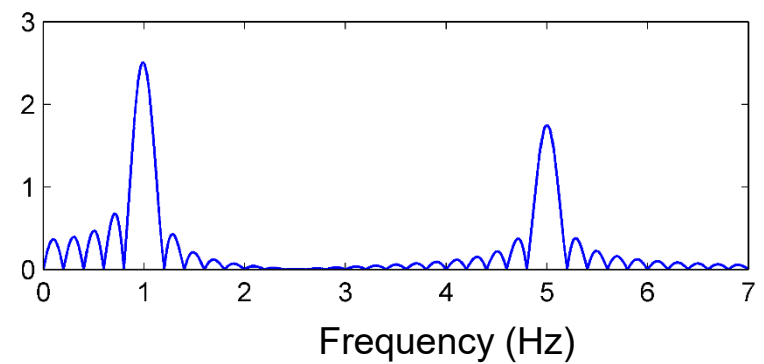
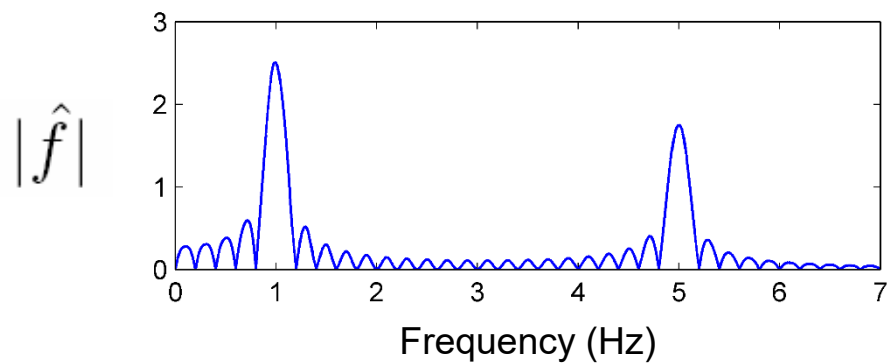
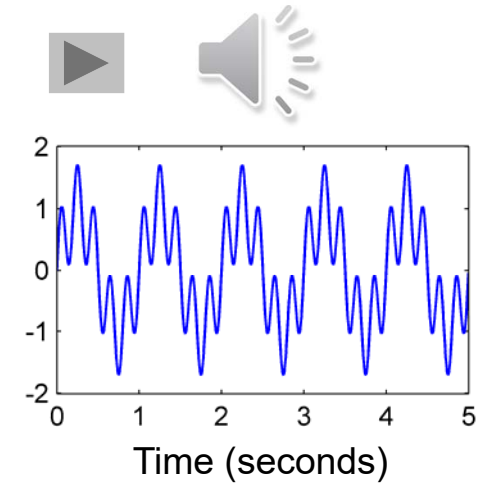
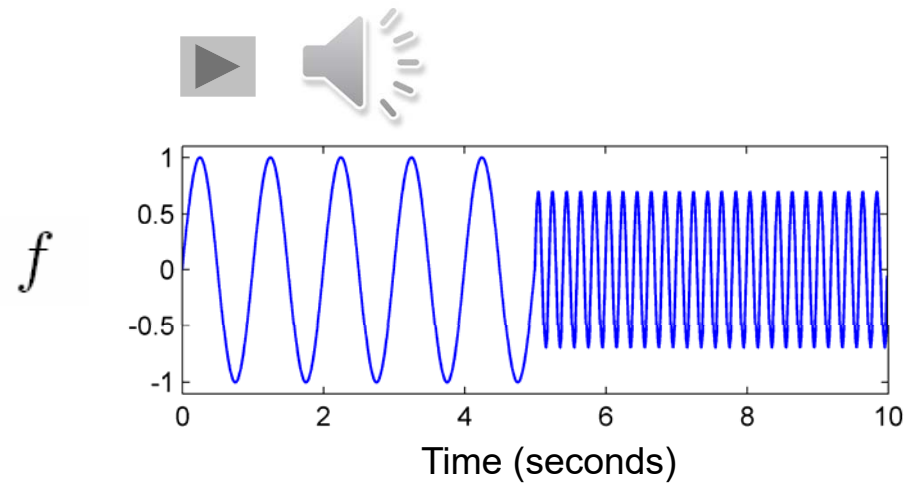
$$f(t) = \int_{\omega \in \mathbb{R}} c_{\omega} \exp(2\pi i \omega t) d\omega$$

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

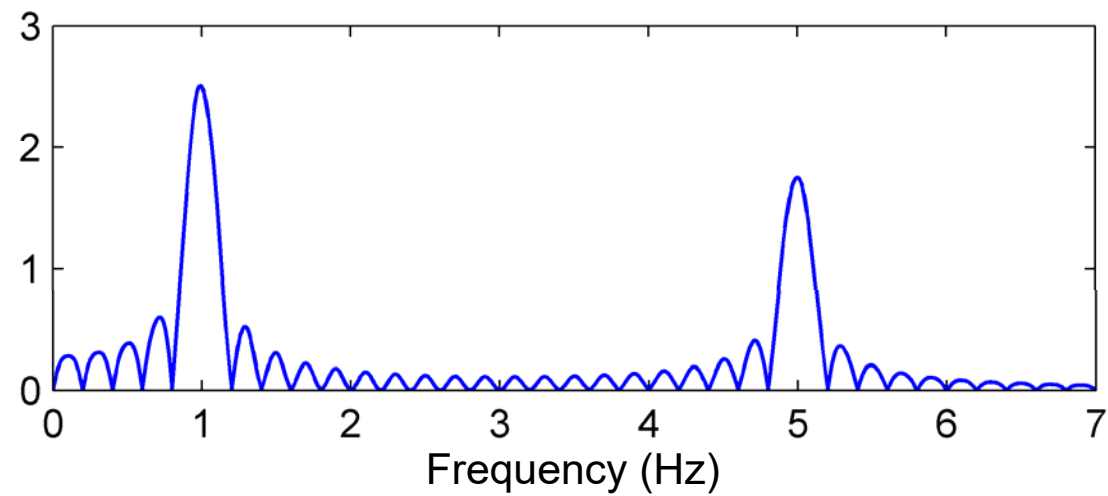
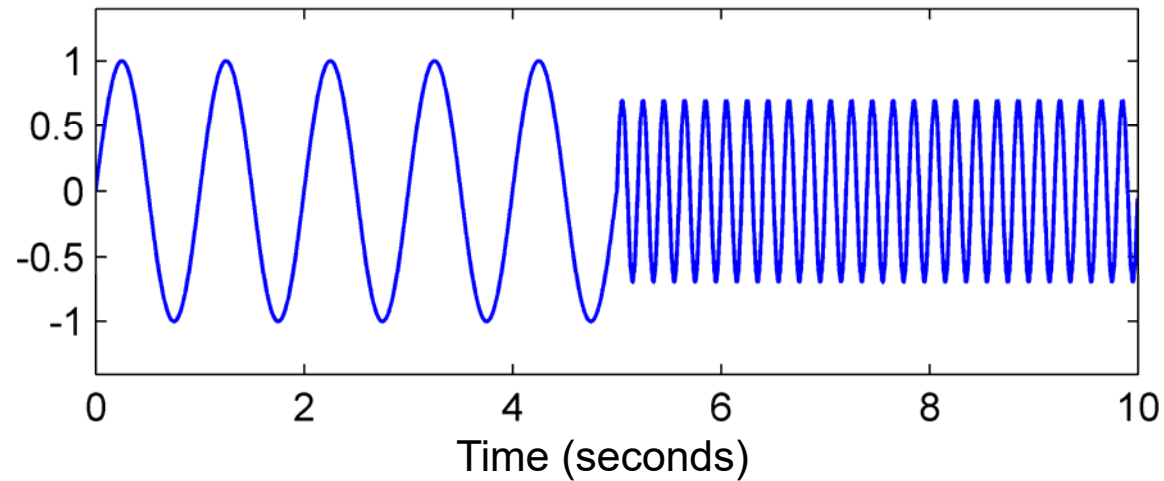


Short Time Fourier Transform

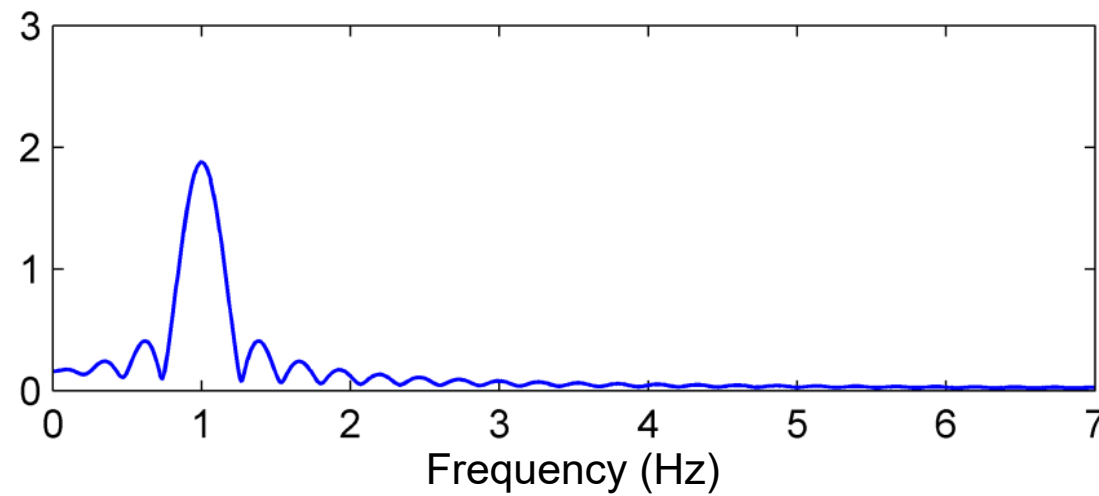
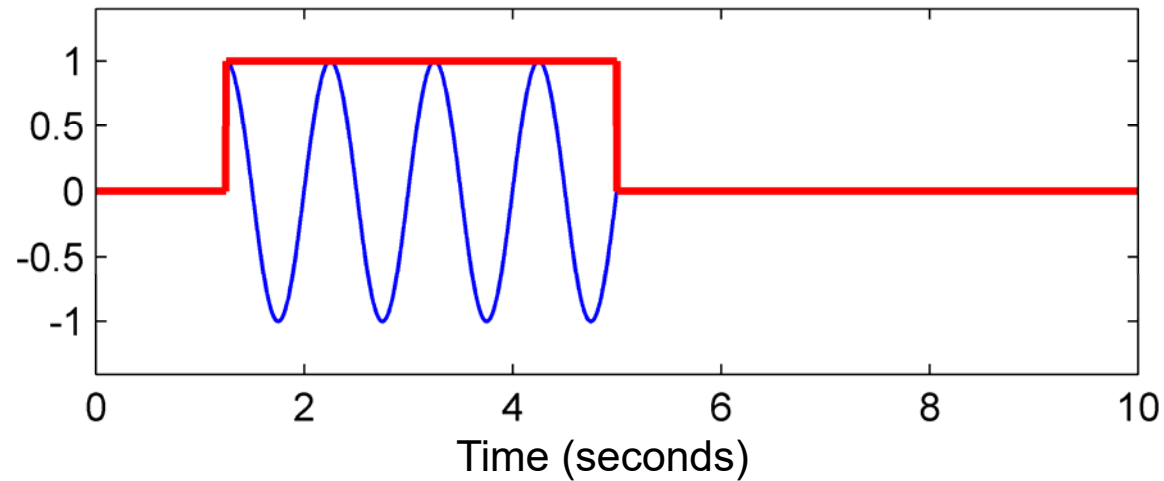
Idea (Dennis Gabor, 1946):

- 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**

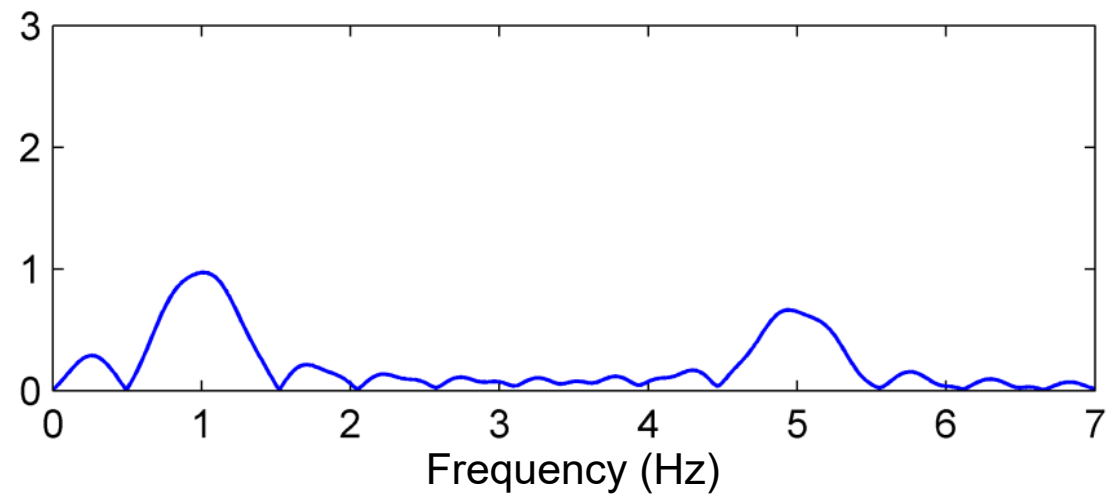
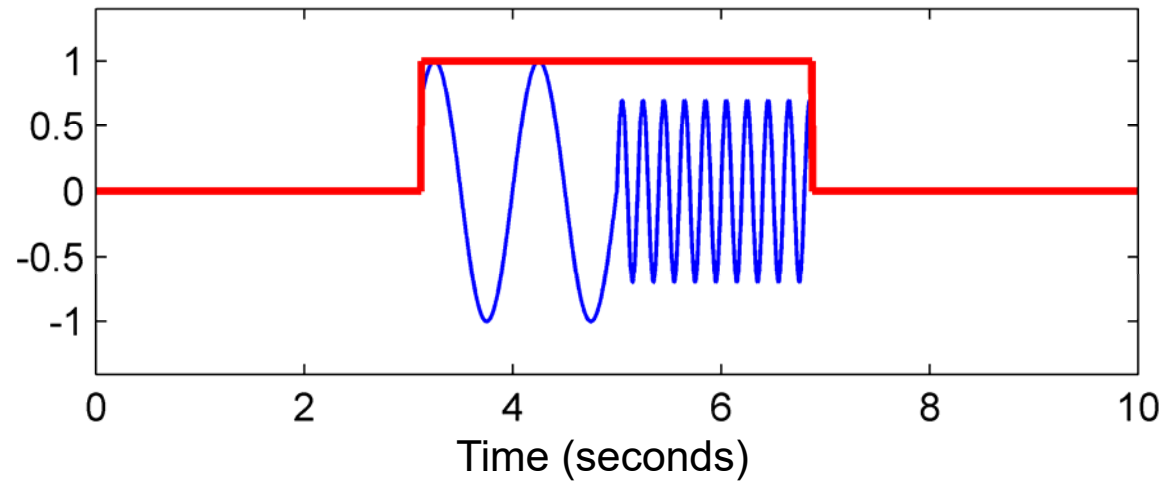
Short Time Fourier Transform



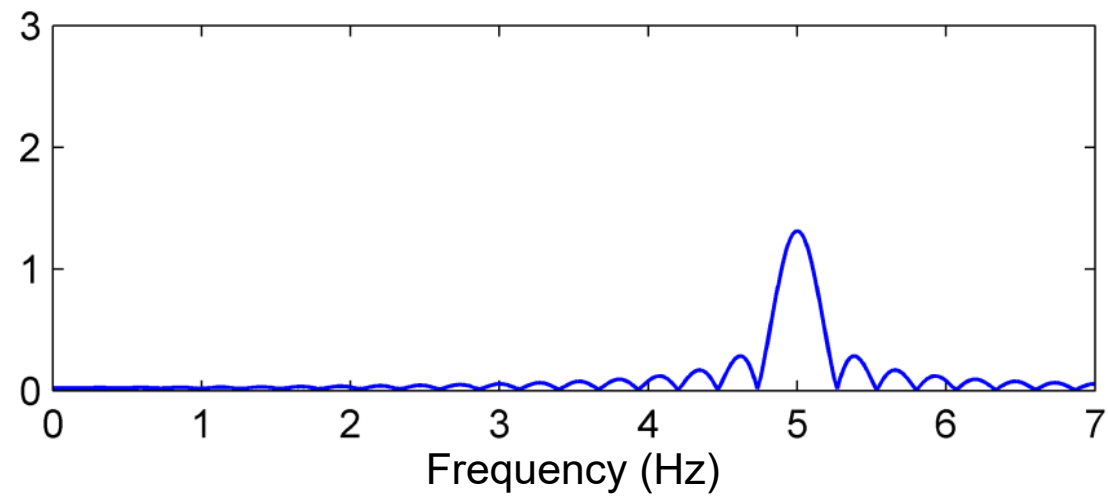
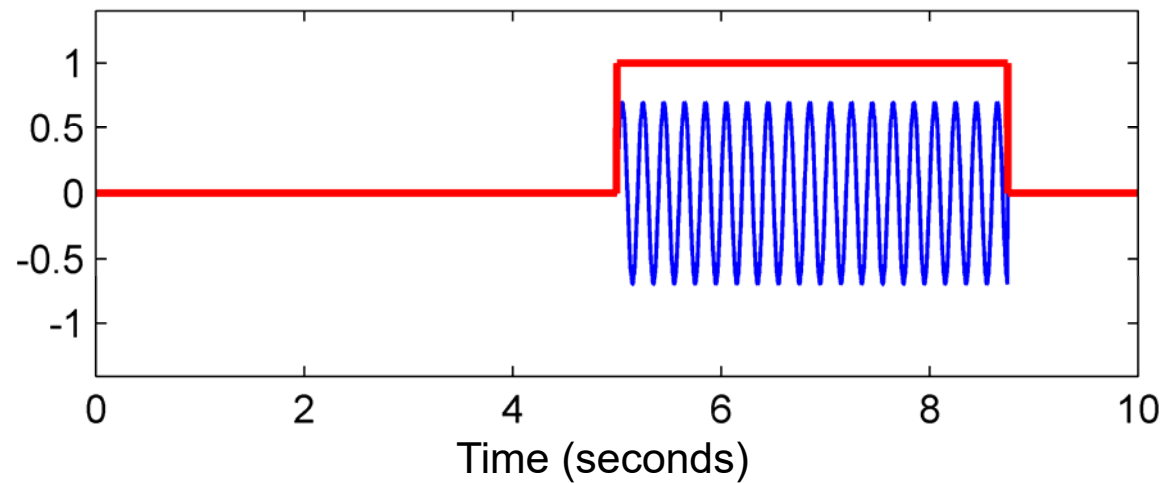
Short Time Fourier Transform



Short Time Fourier Transform



Short Time Fourier Transform



Short Time Fourier Transform

Definition

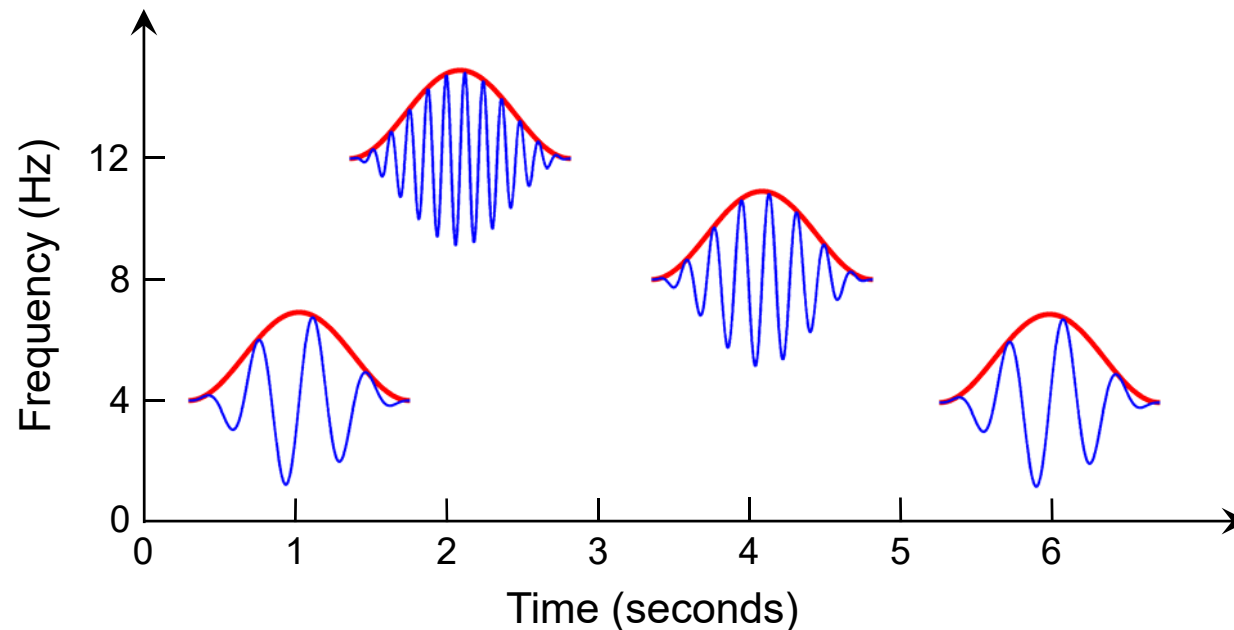
- Signal $f: \mathbb{R} \rightarrow \mathbb{R}$
- Window function $g: \mathbb{R} \rightarrow \mathbb{R} \quad (g \in L^2(\mathbb{R}), \|g\|_2 \neq 0)$
- STFT $\tilde{f}_g(t, \omega) = \int_{u \in \mathbb{R}} f(u) \bar{g}(u - t) \exp(-2\pi i \omega u) du = \langle f | g_{t, \omega} \rangle$

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

Short Time Fourier Transform

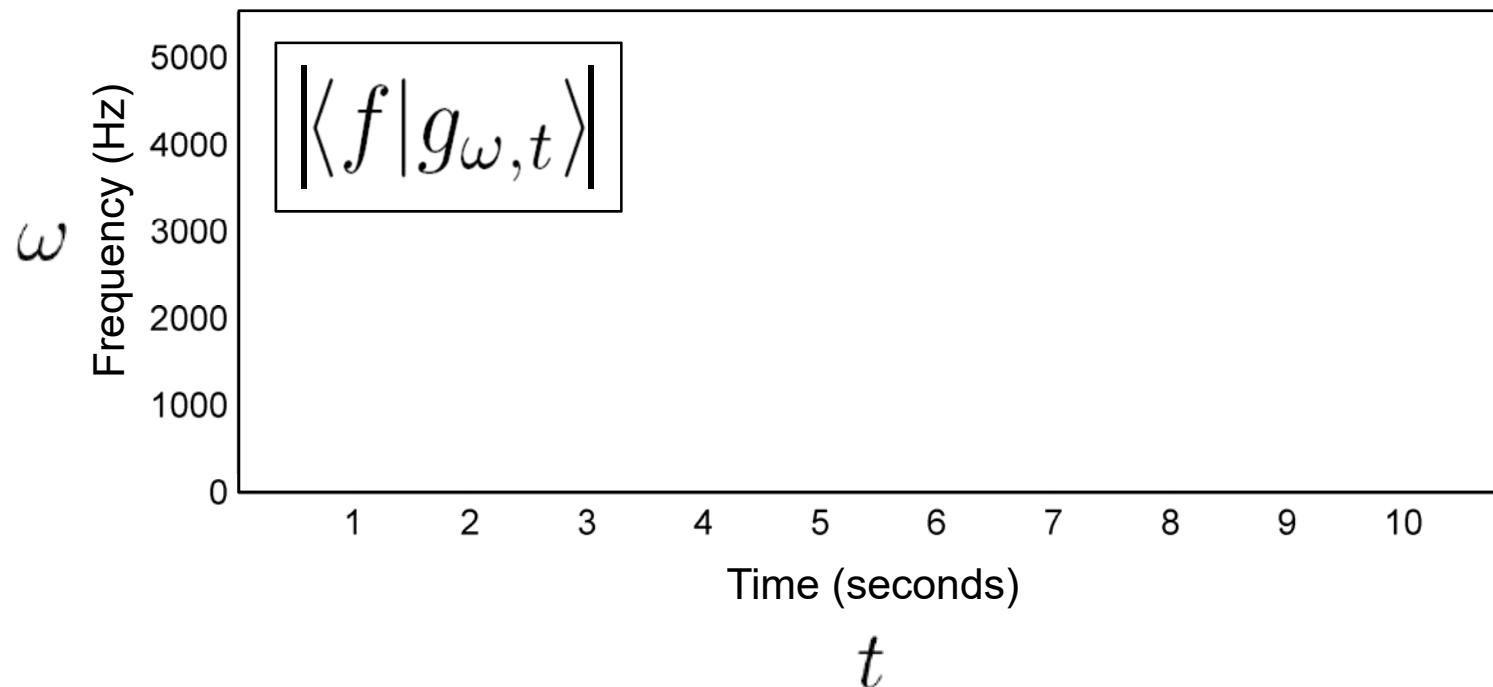
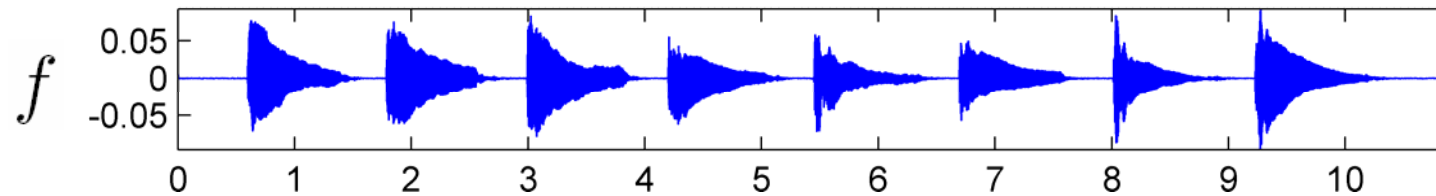
Intuition:

- $g_{t,\omega}$ 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



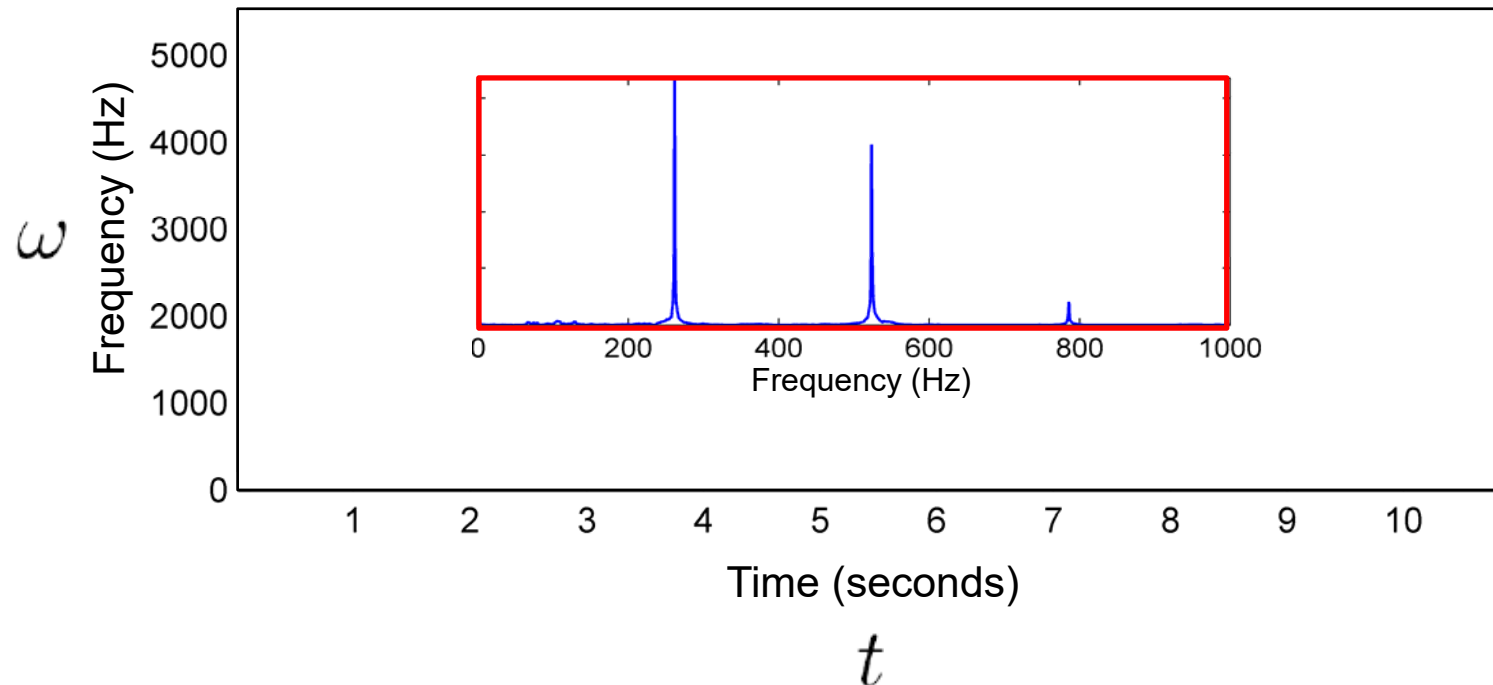
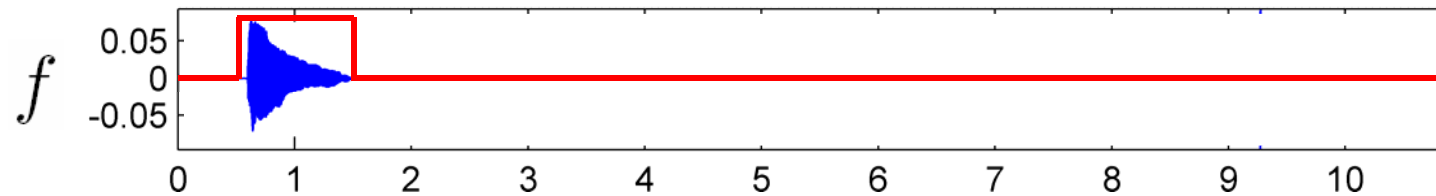
Time-Frequency Representation

Spectrogram



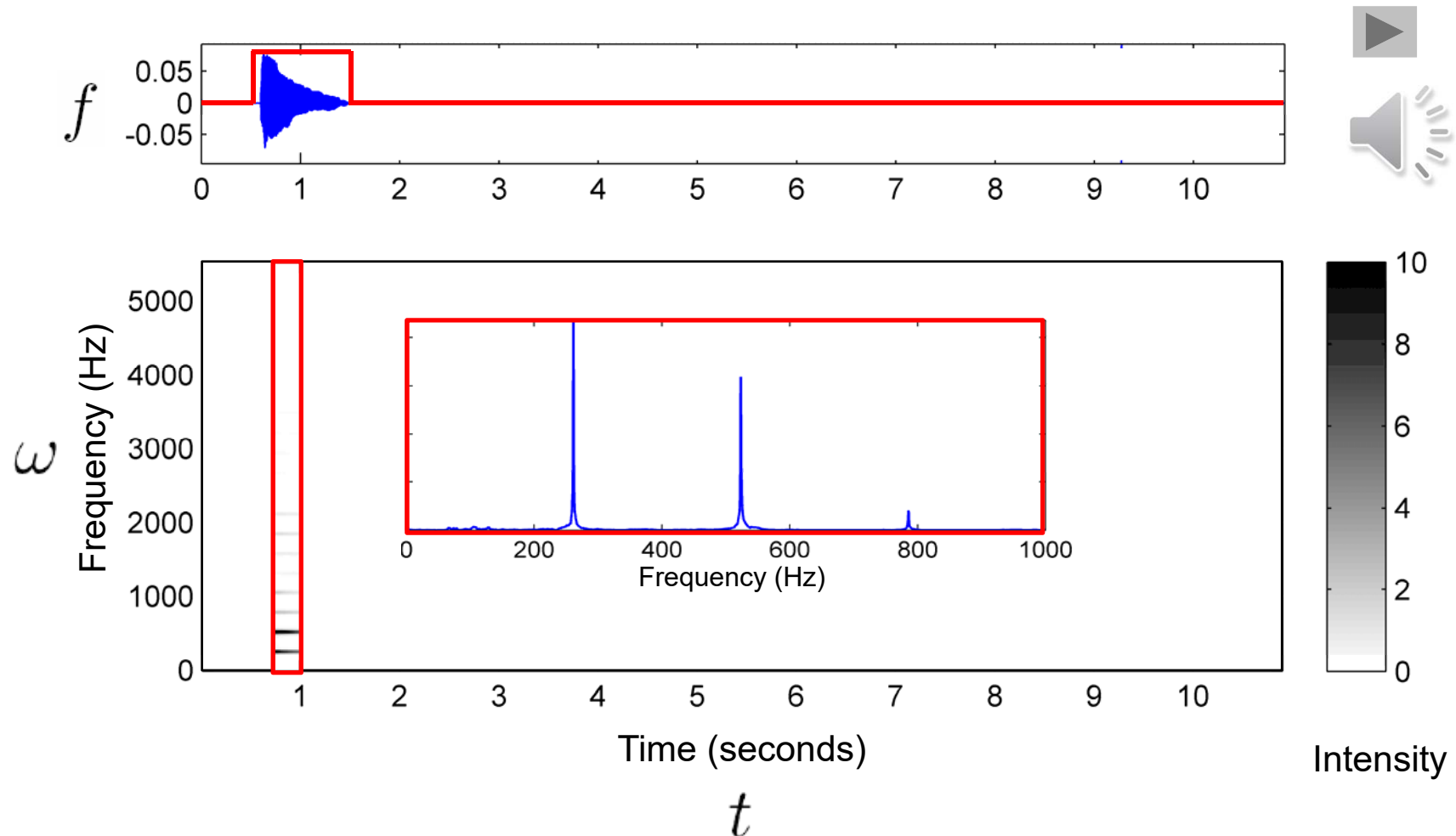
Time-Frequency Representation

Spectrogram



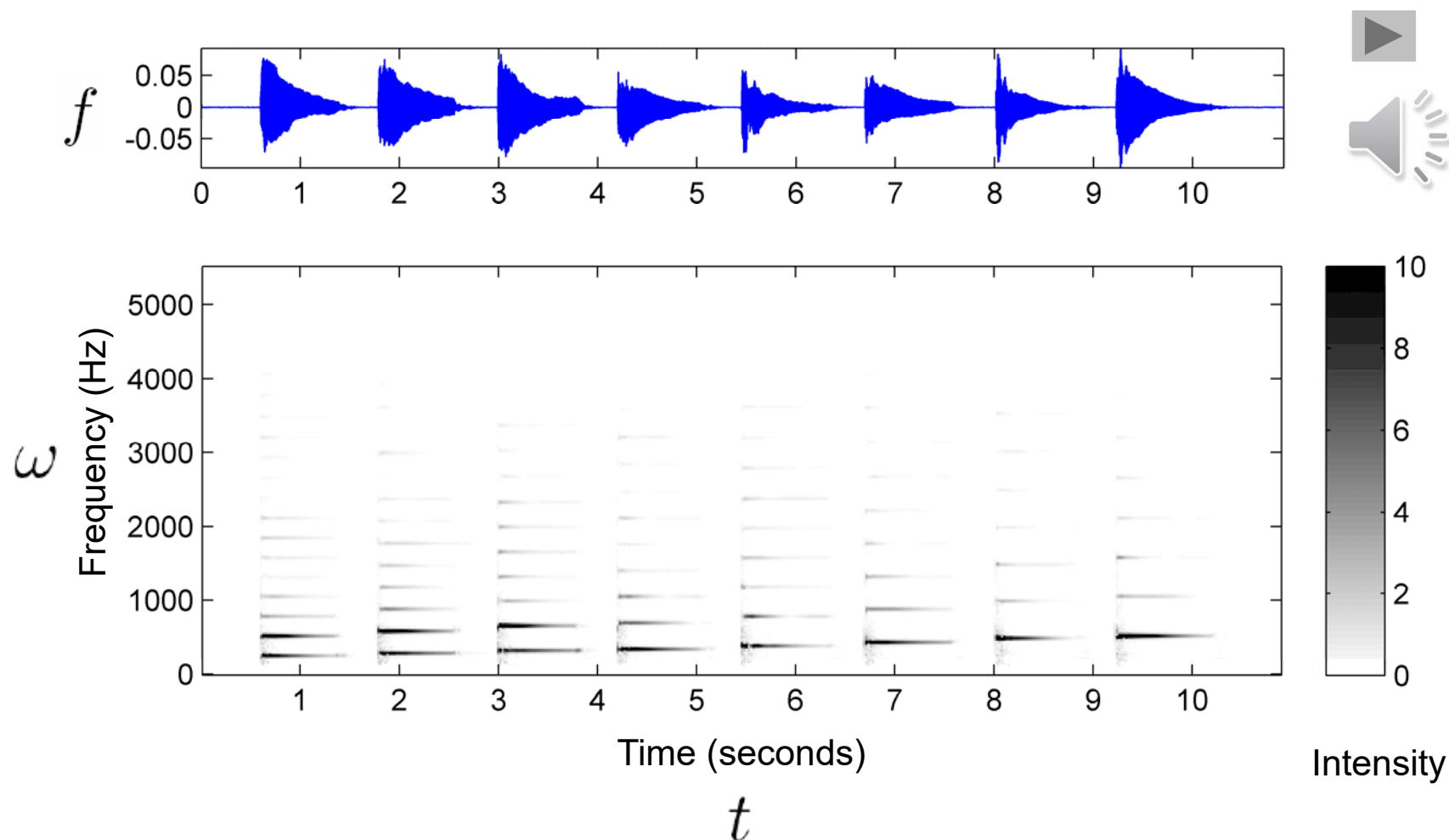
Time-Frequency Representation

Spectrogram



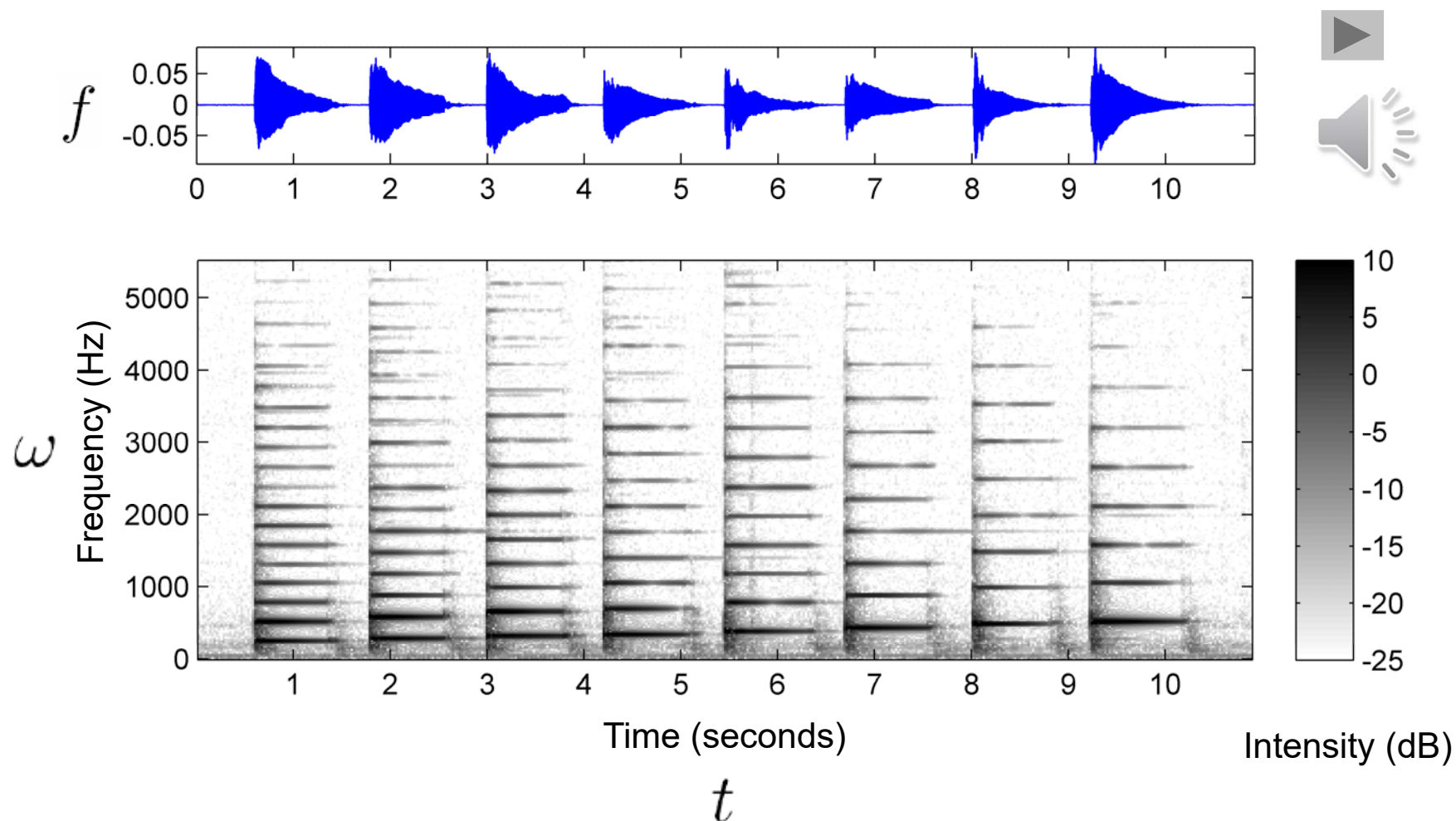
Time-Frequency Representation

Spectrogram



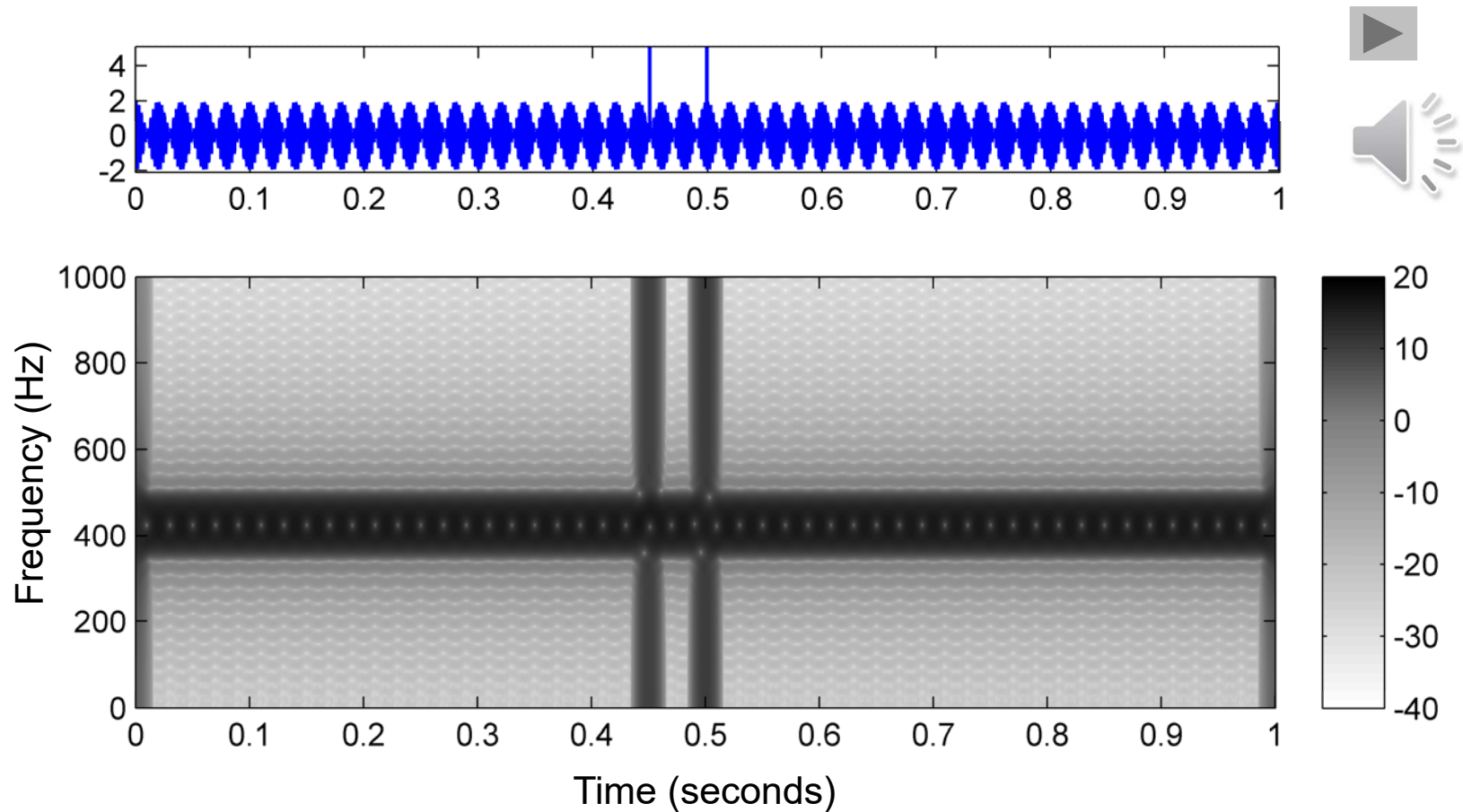
Time-Frequency Representation

Spectrogram



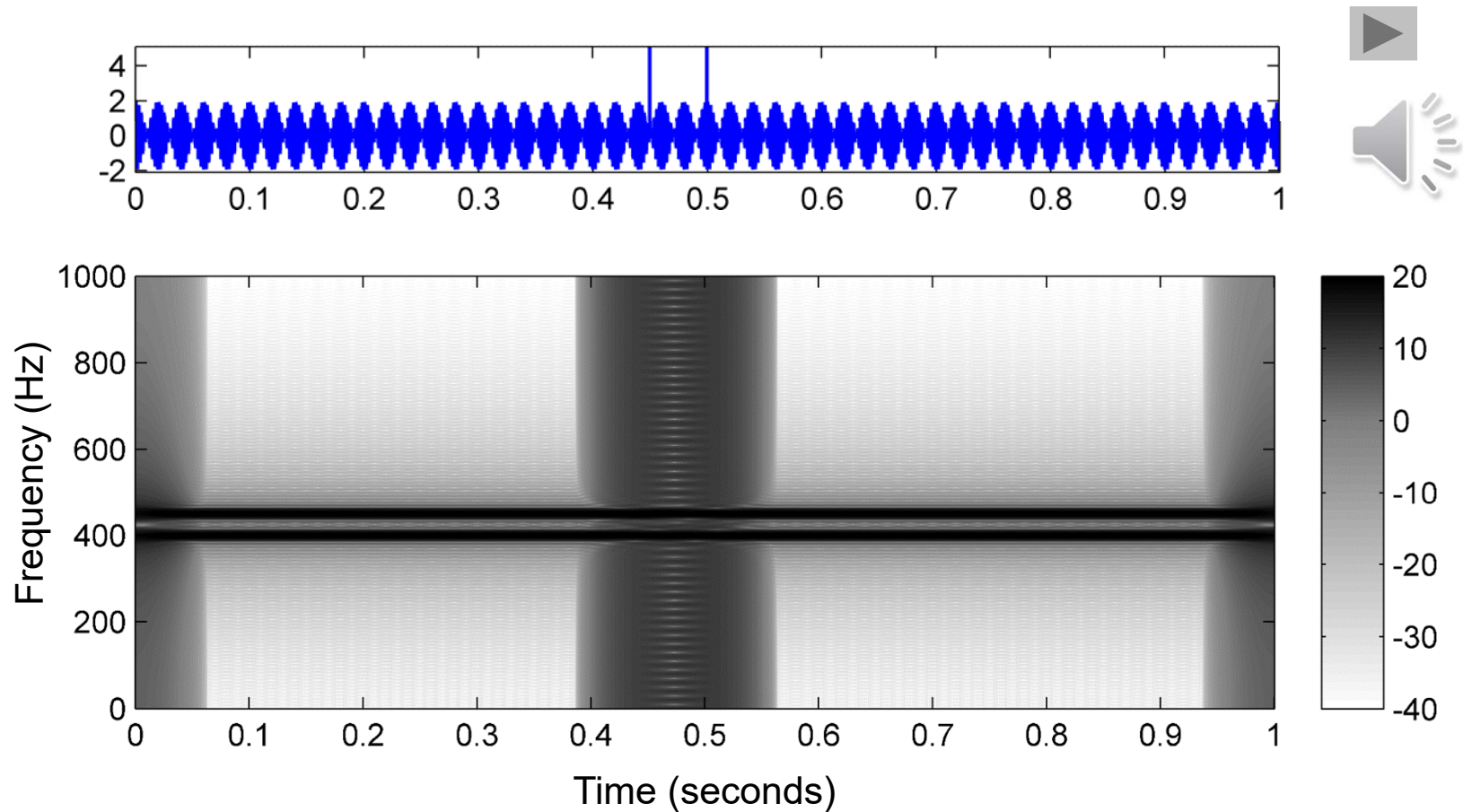
Time-Frequency Representation

Signal and STFT with Hann window of length 20 ms



Time-Frequency Representation

Signal and STFT with Hann window of length 100 ms



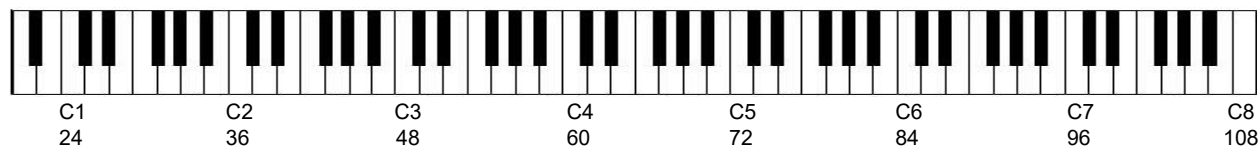
Time-Frequency Representation

Time-Frequency Localization

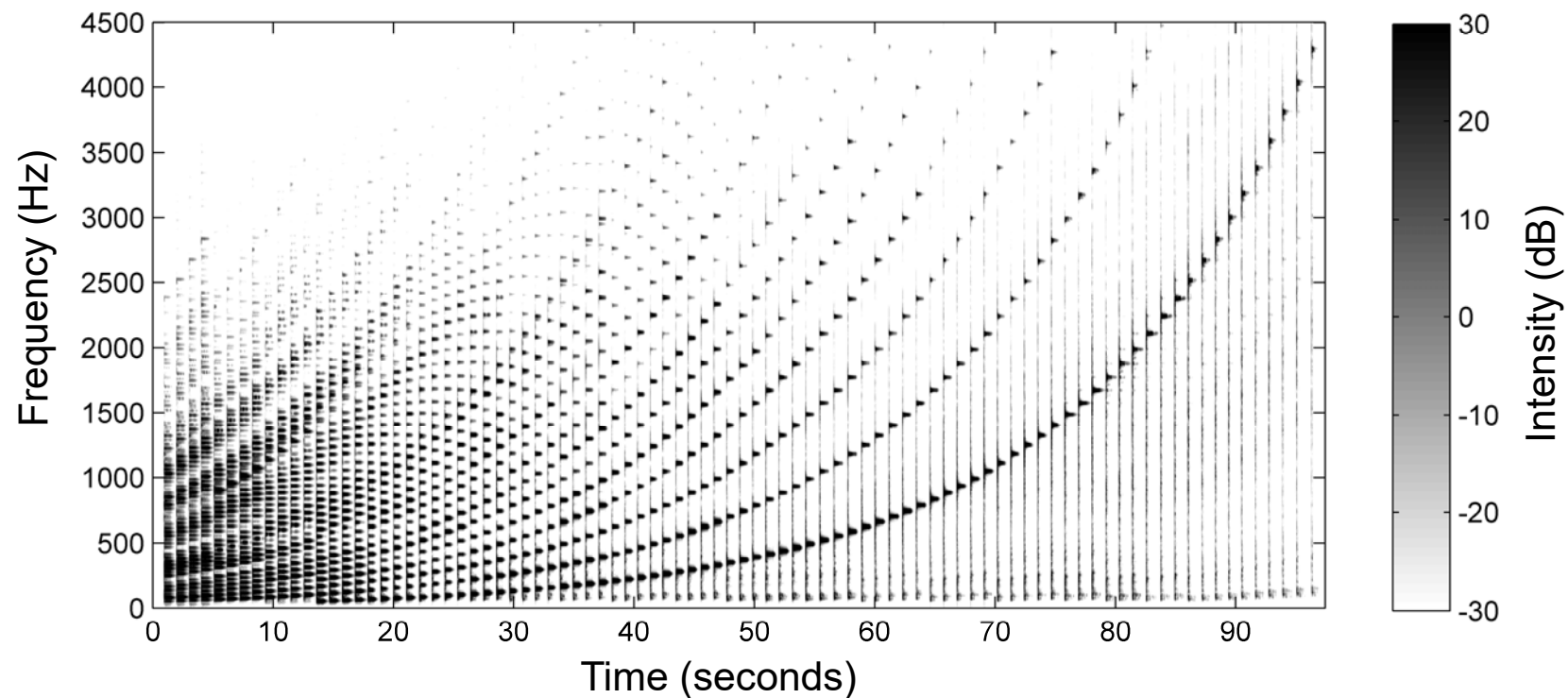
- Size of window constitutes a trade-off between time resolution and frequency resolution:
 - Large window** : poor time resolution
good frequency resolution
 - Small window** : good time resolution
poor frequency resolution
- **Heisenberg Uncertainty Principle**: there is no window function that localizes in time and frequency with arbitrary position.

Audio Features

Example: Chromatic scale

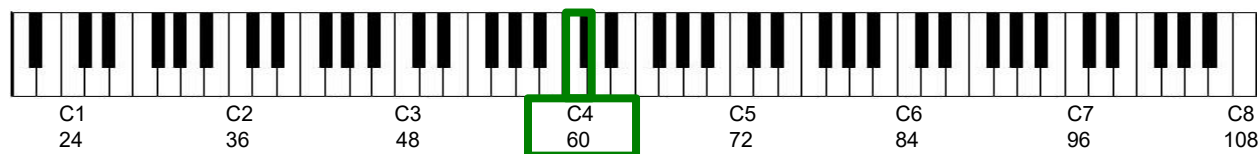


Spectrogram

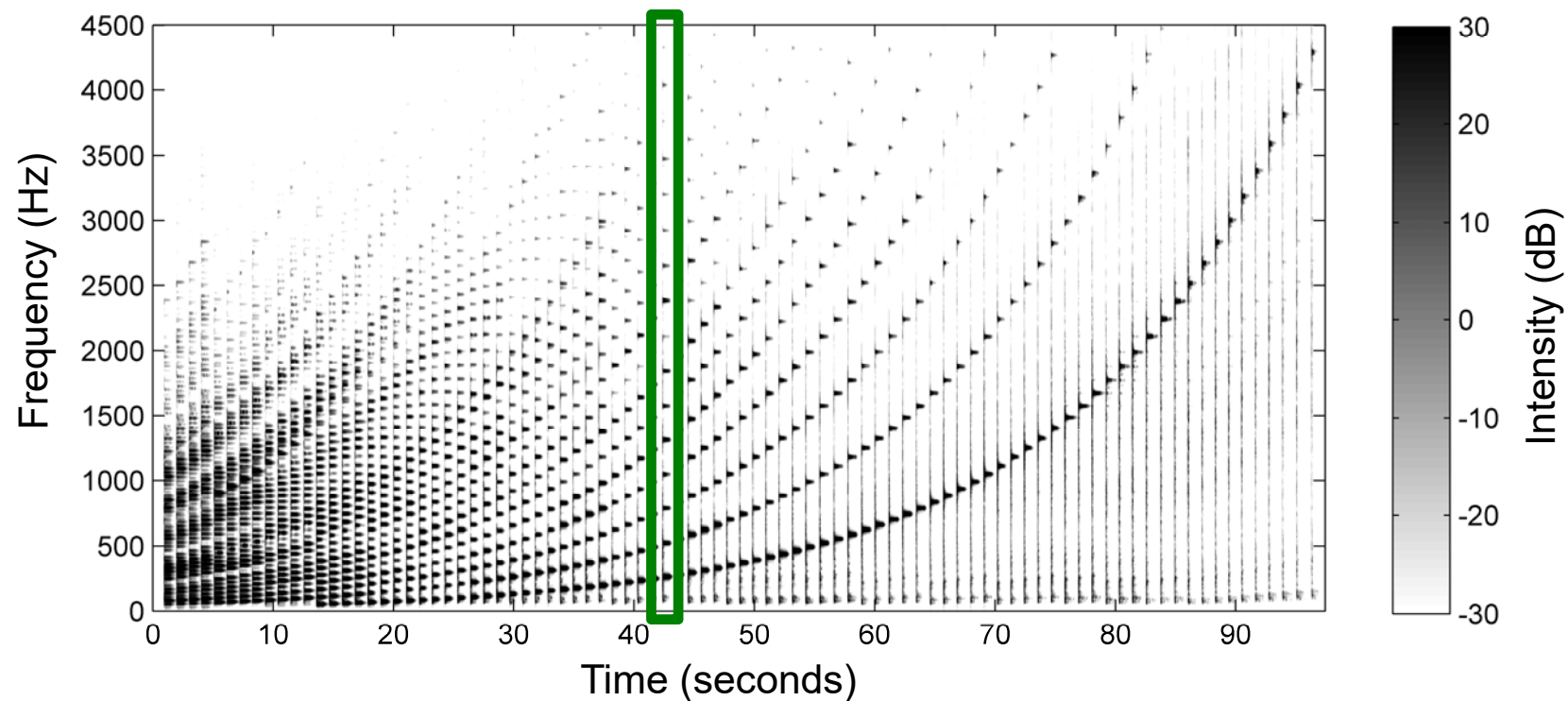


Audio Features

Example: Chromatic scale

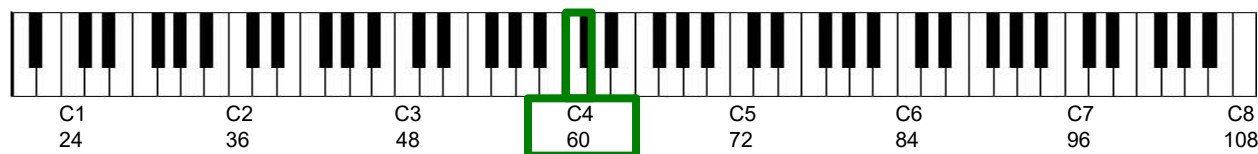


Spectrogram

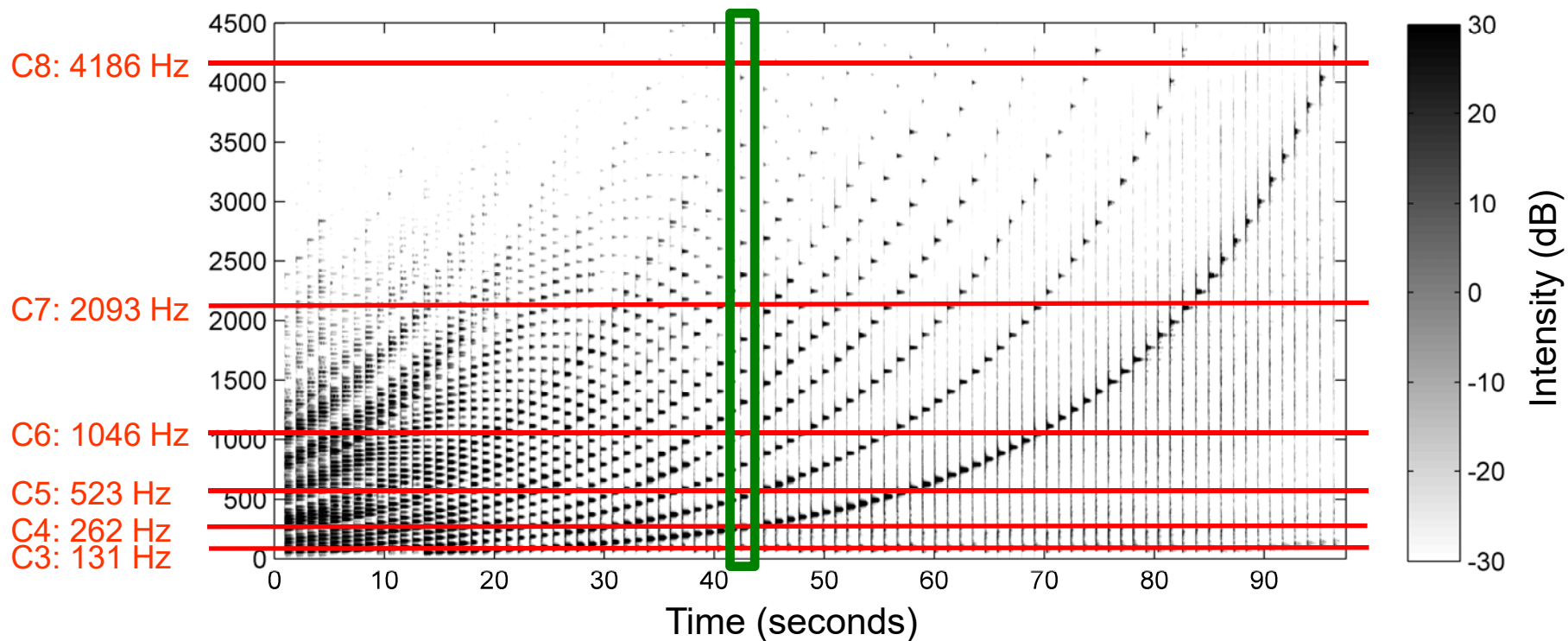


Audio Features

Example: Chromatic scale

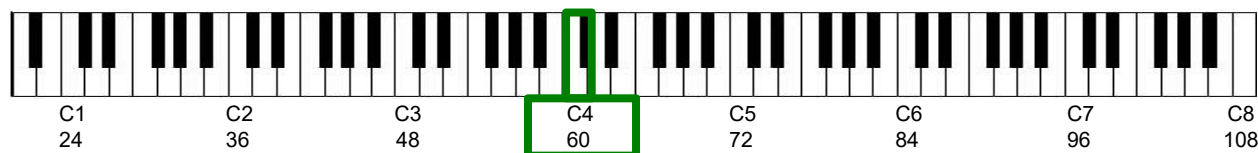


Spectrogram

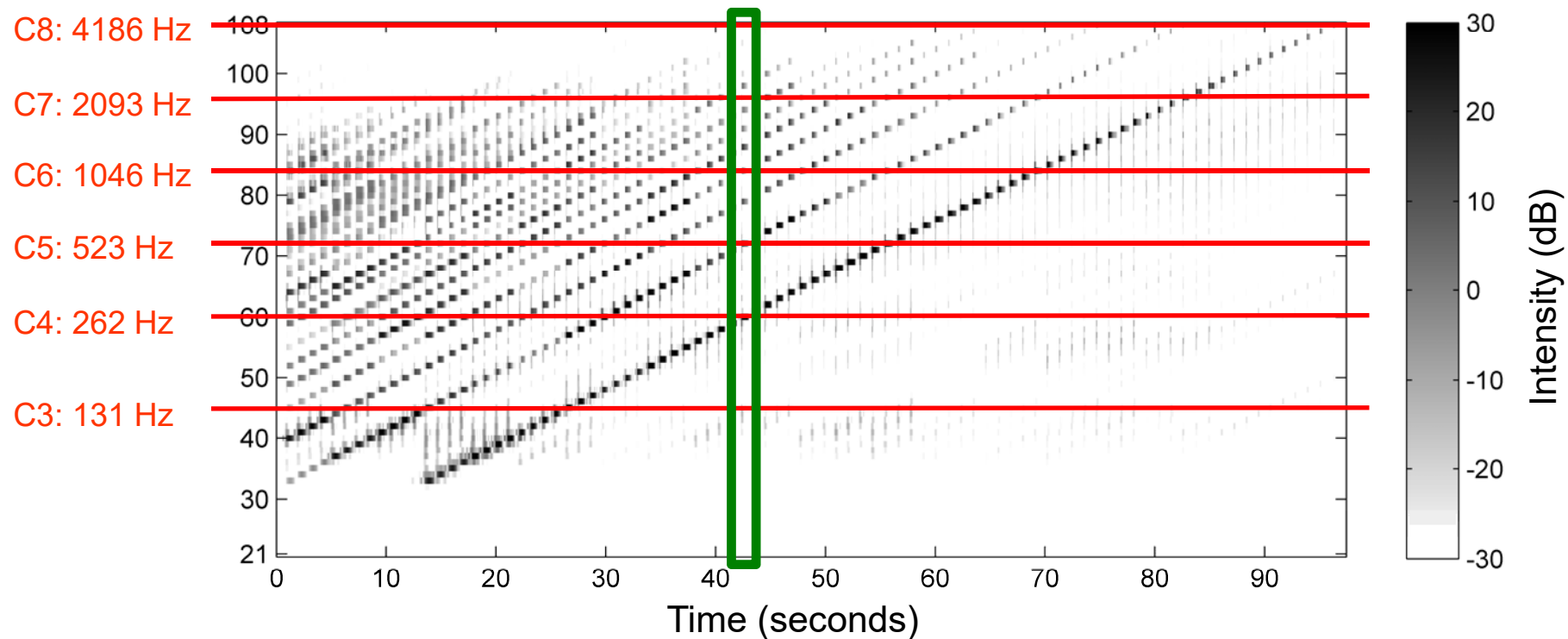


Audio Features

Example: Chromatic scale

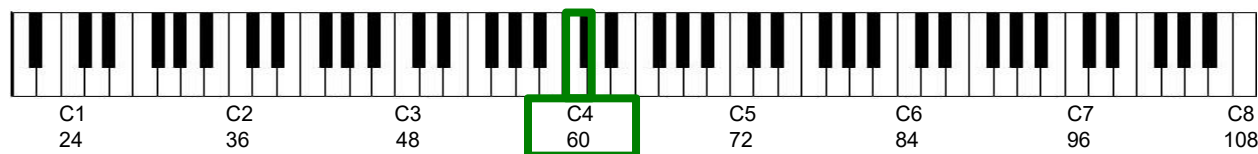


Log-frequency spectrogram

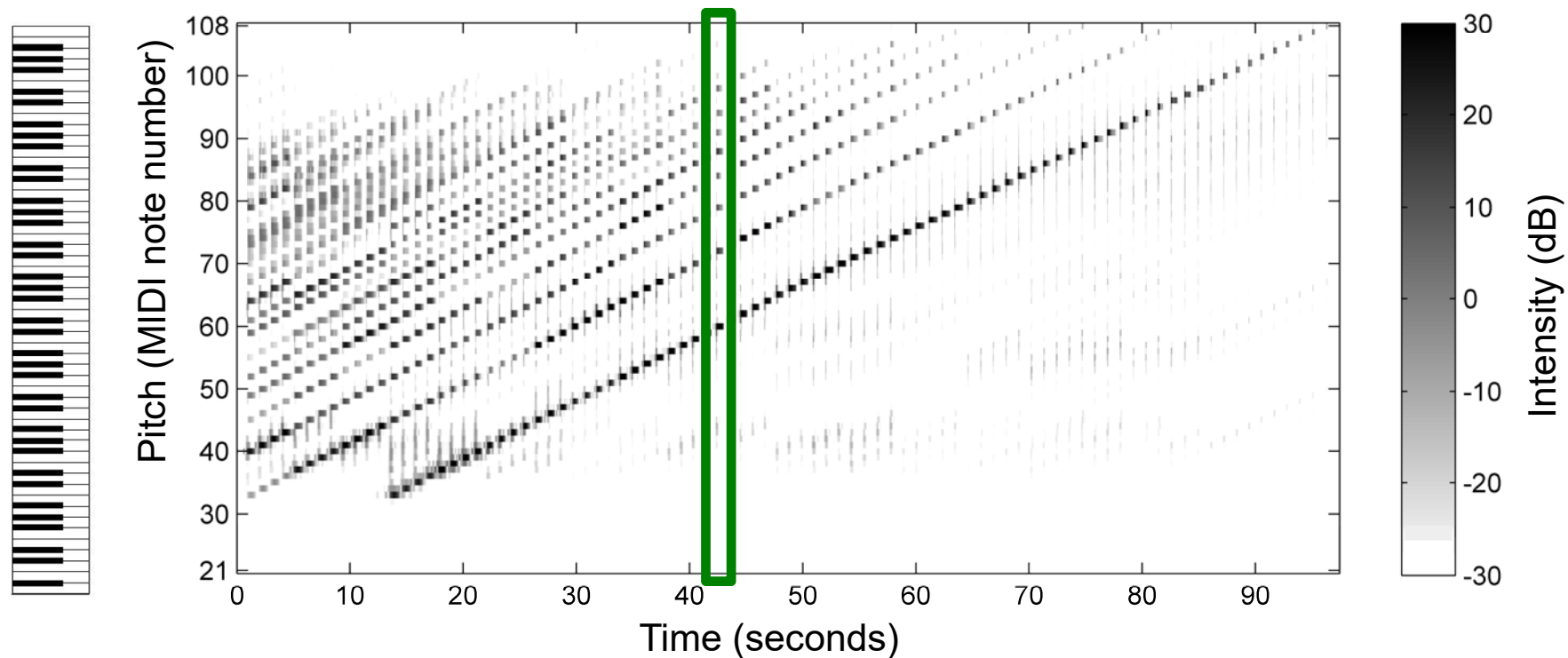


Audio Features

Example: Chromatic scale

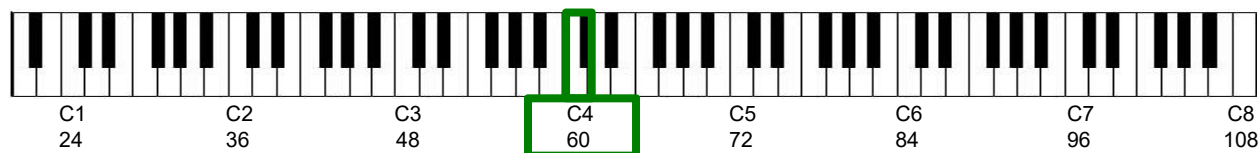


Log-frequency spectrogram

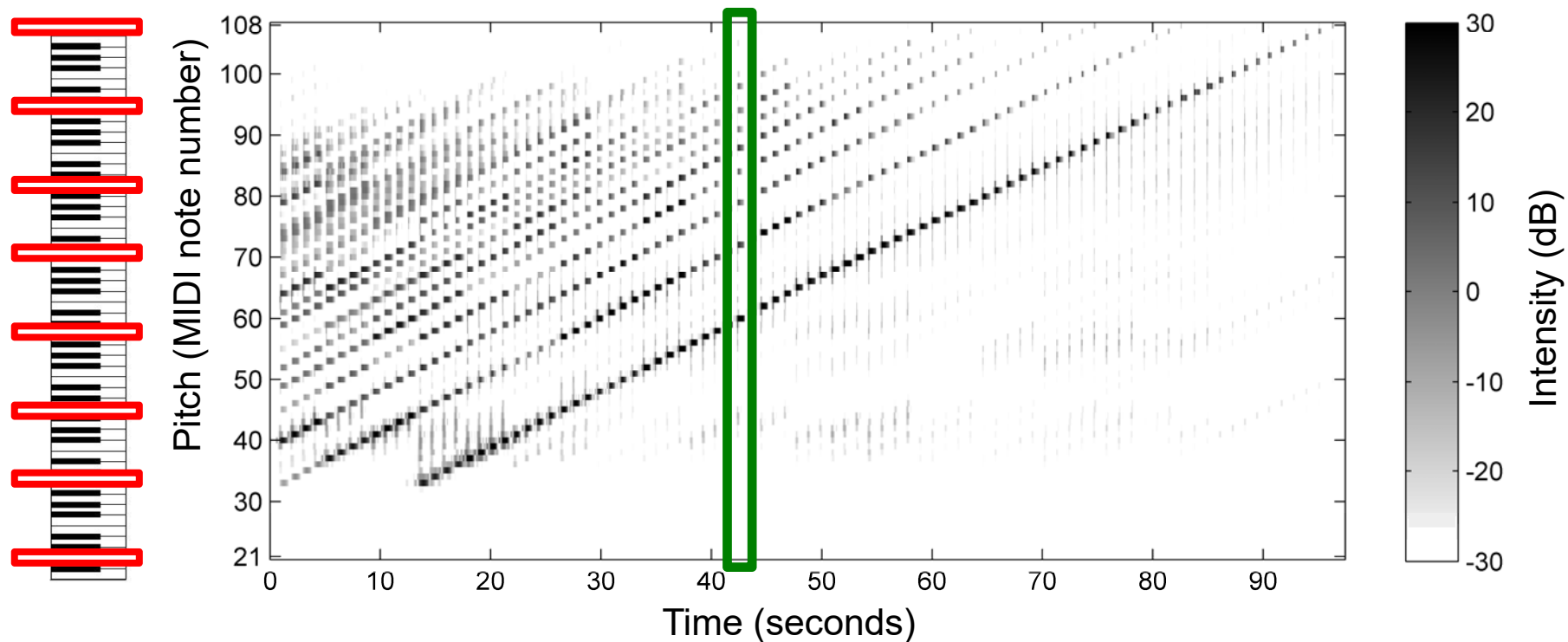


Audio Features

Example: Chromatic scale



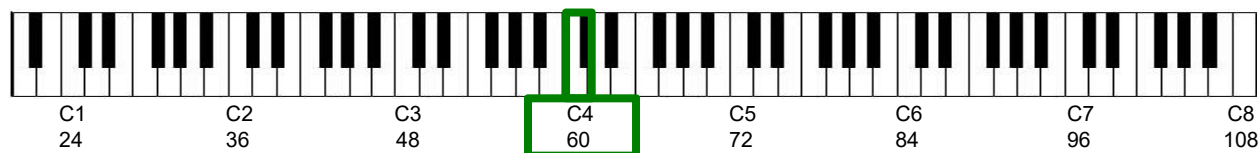
Log-frequency spectrogram



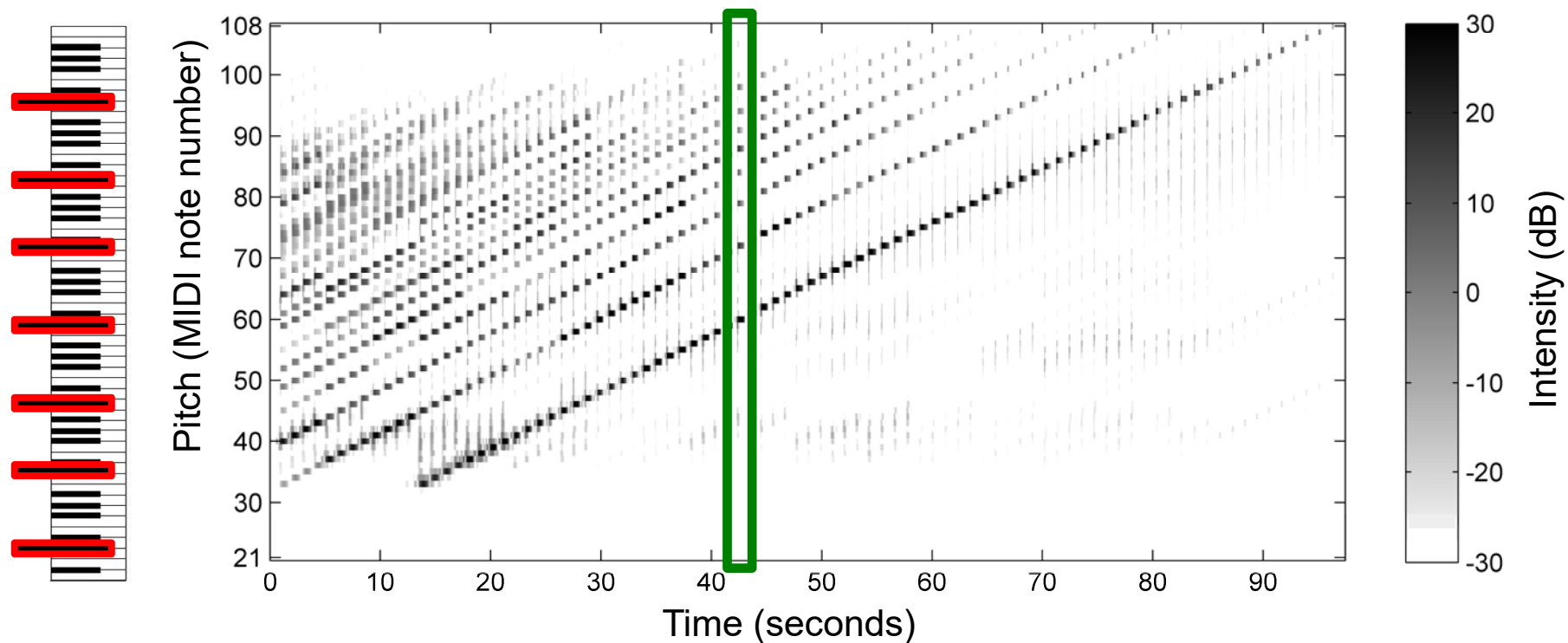
Chroma C

Audio Features

Example: Chromatic scale



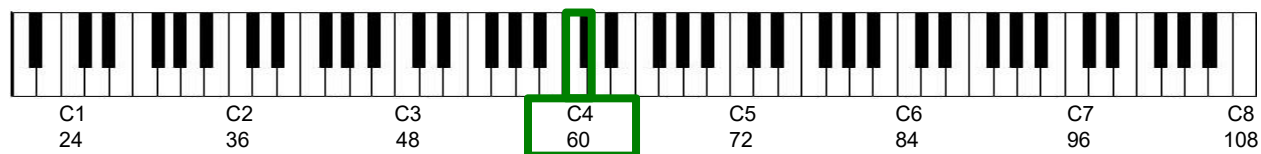
Log-frequency spectrogram



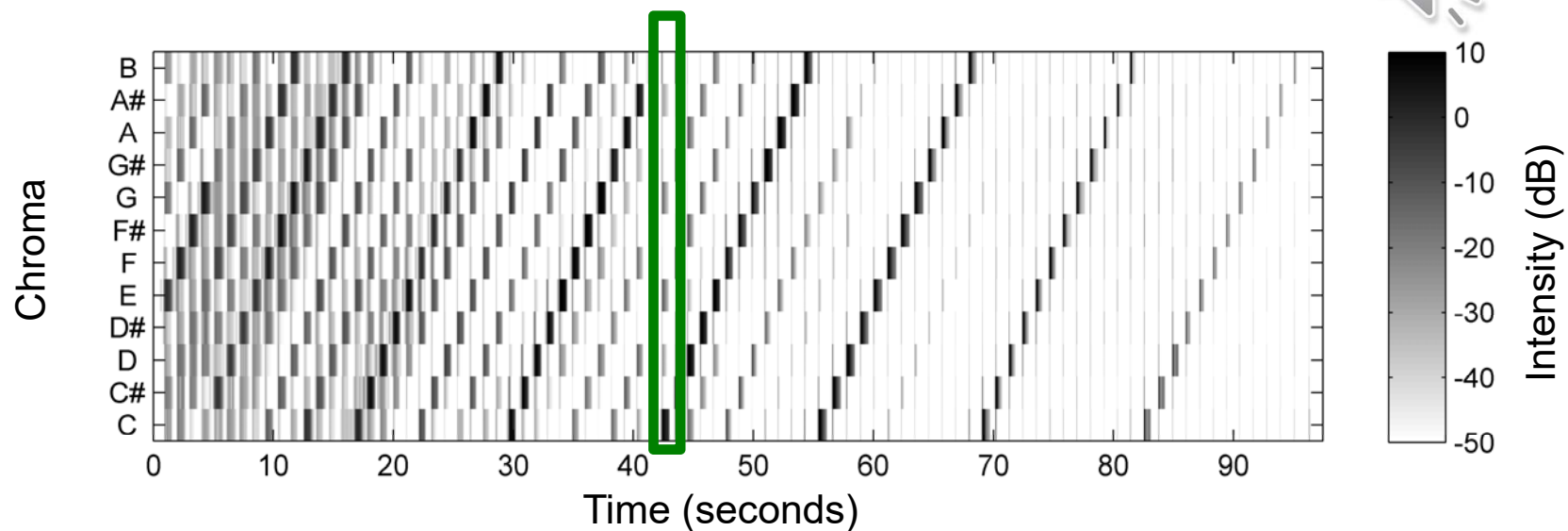
Chroma C[#]

Audio Features

Example: Chromatic scale



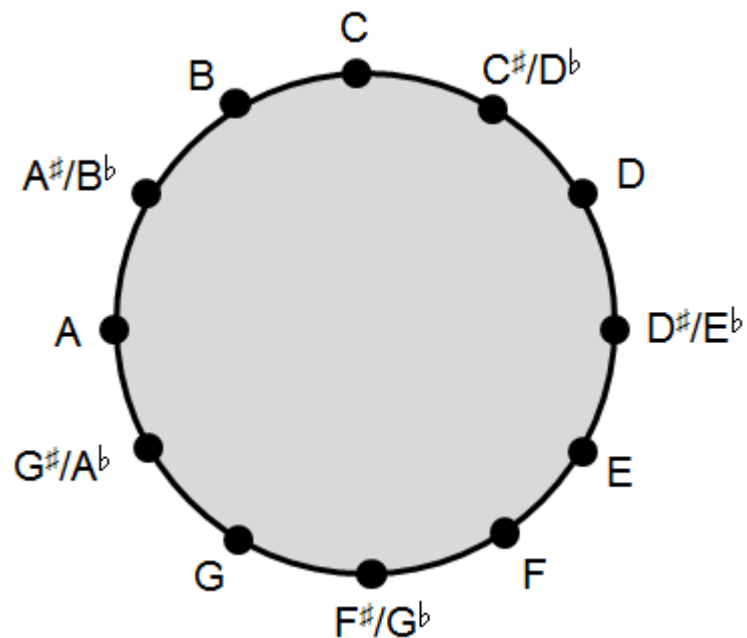
Chromagram



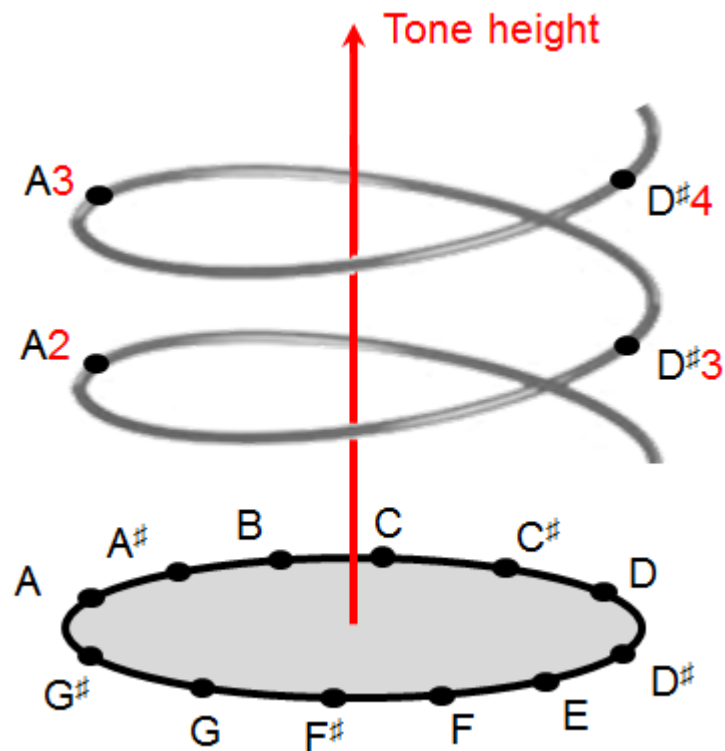
Audio Features

Chroma features

Chromatic circle

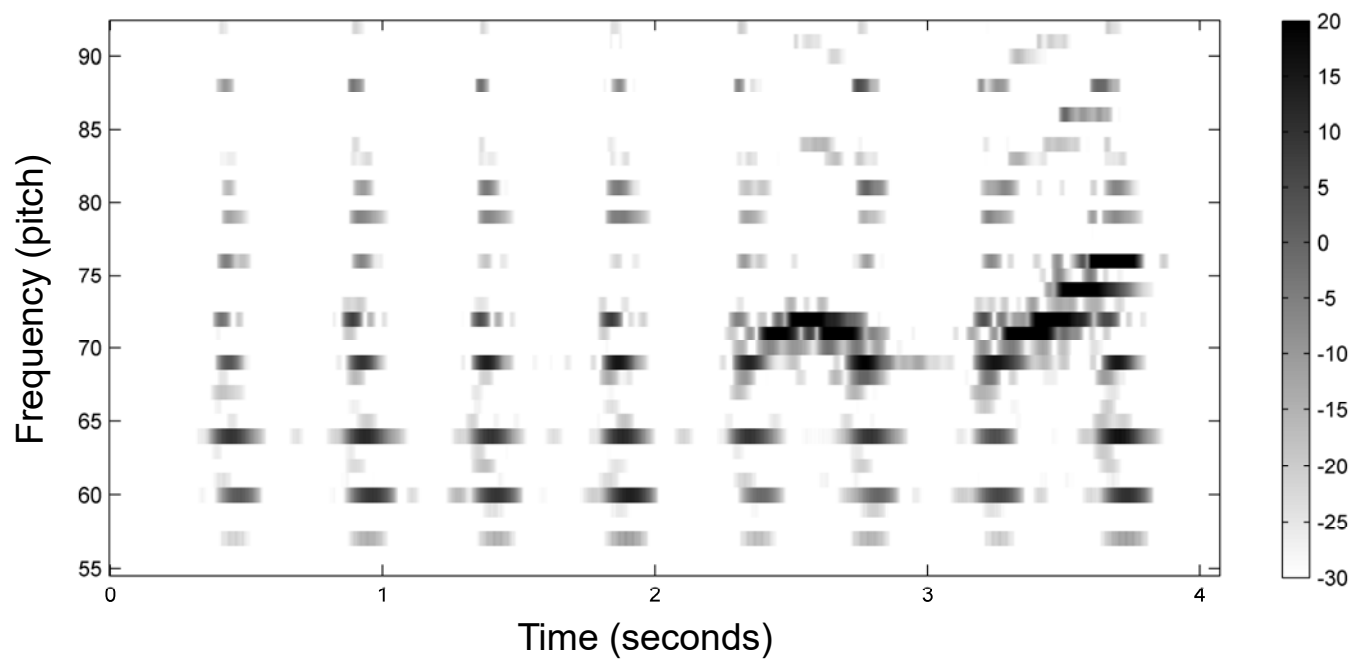


Shepard's helix of pitch



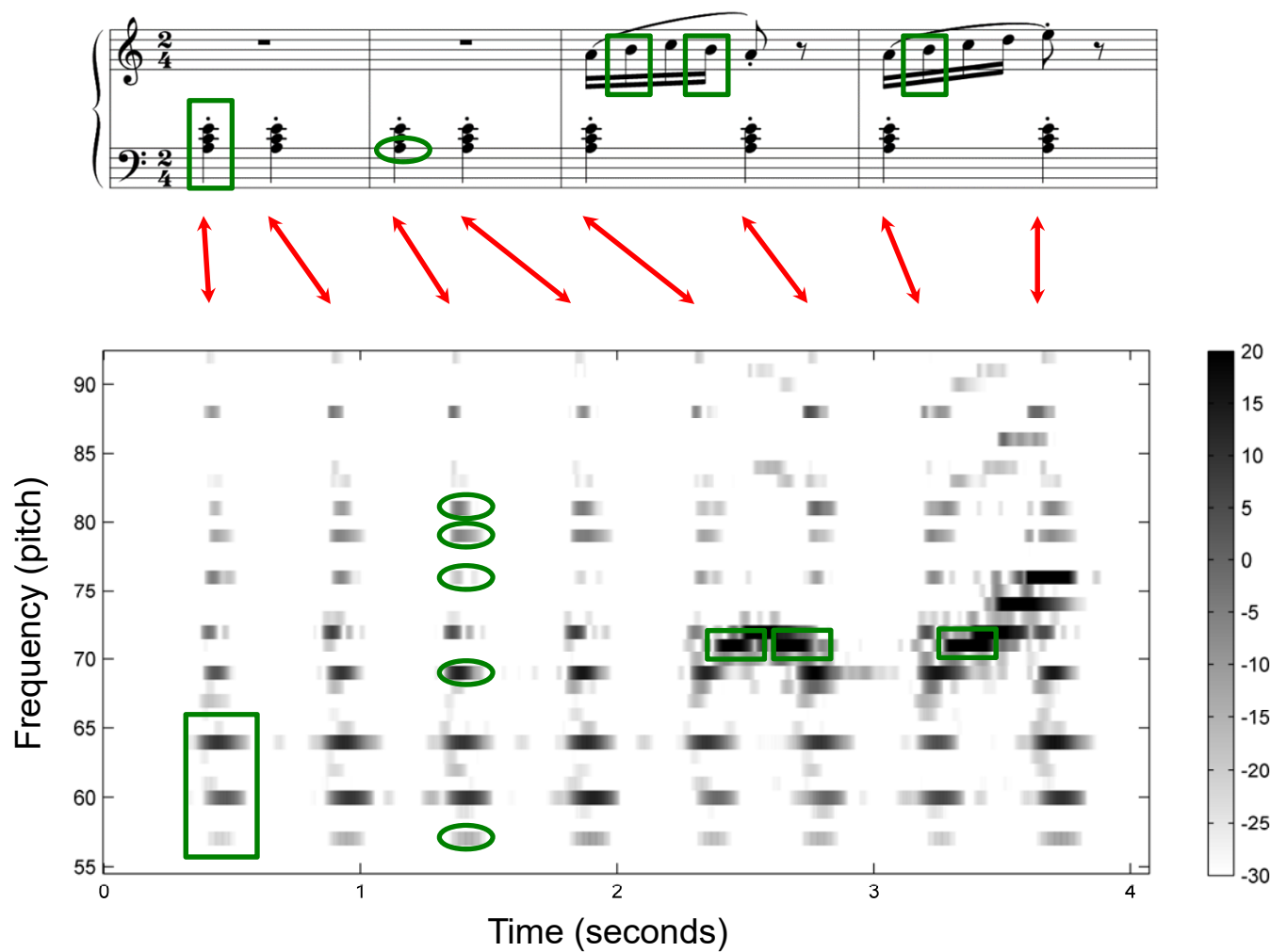
Audio Features

Chroma features



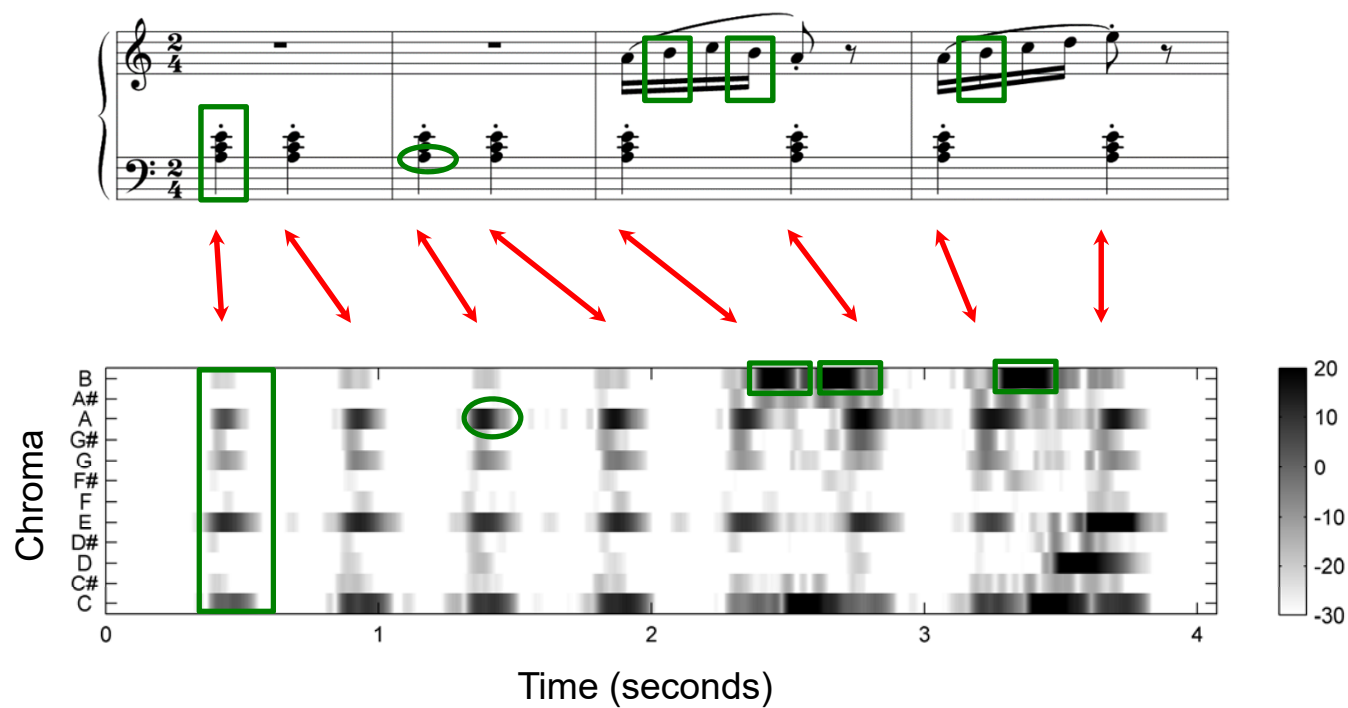
Audio Features

Chroma features



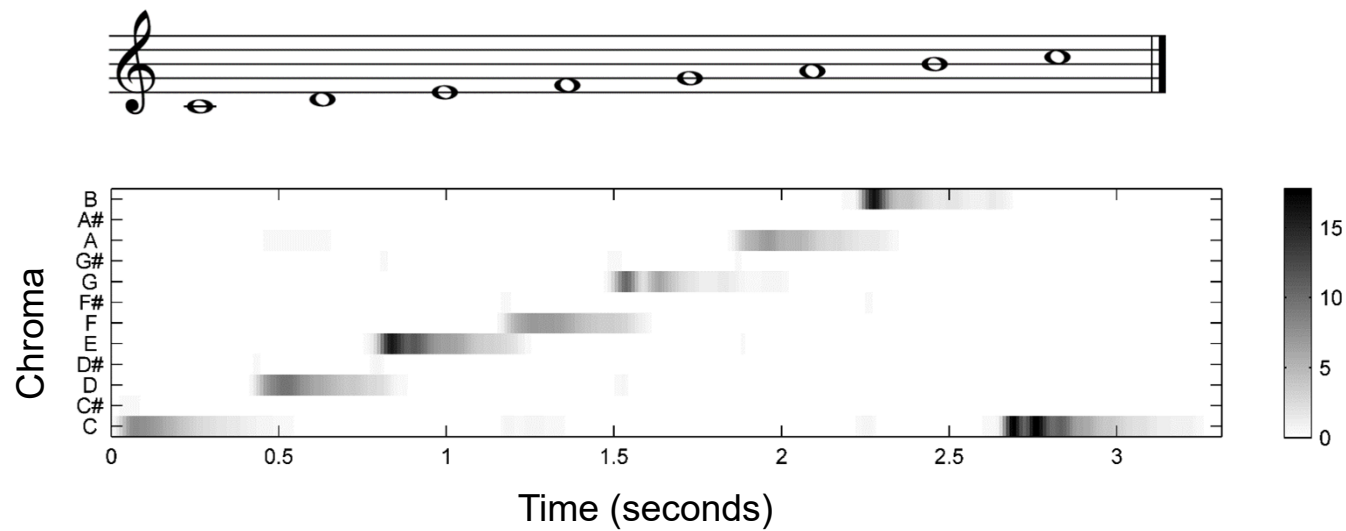
Audio Features

Chroma features



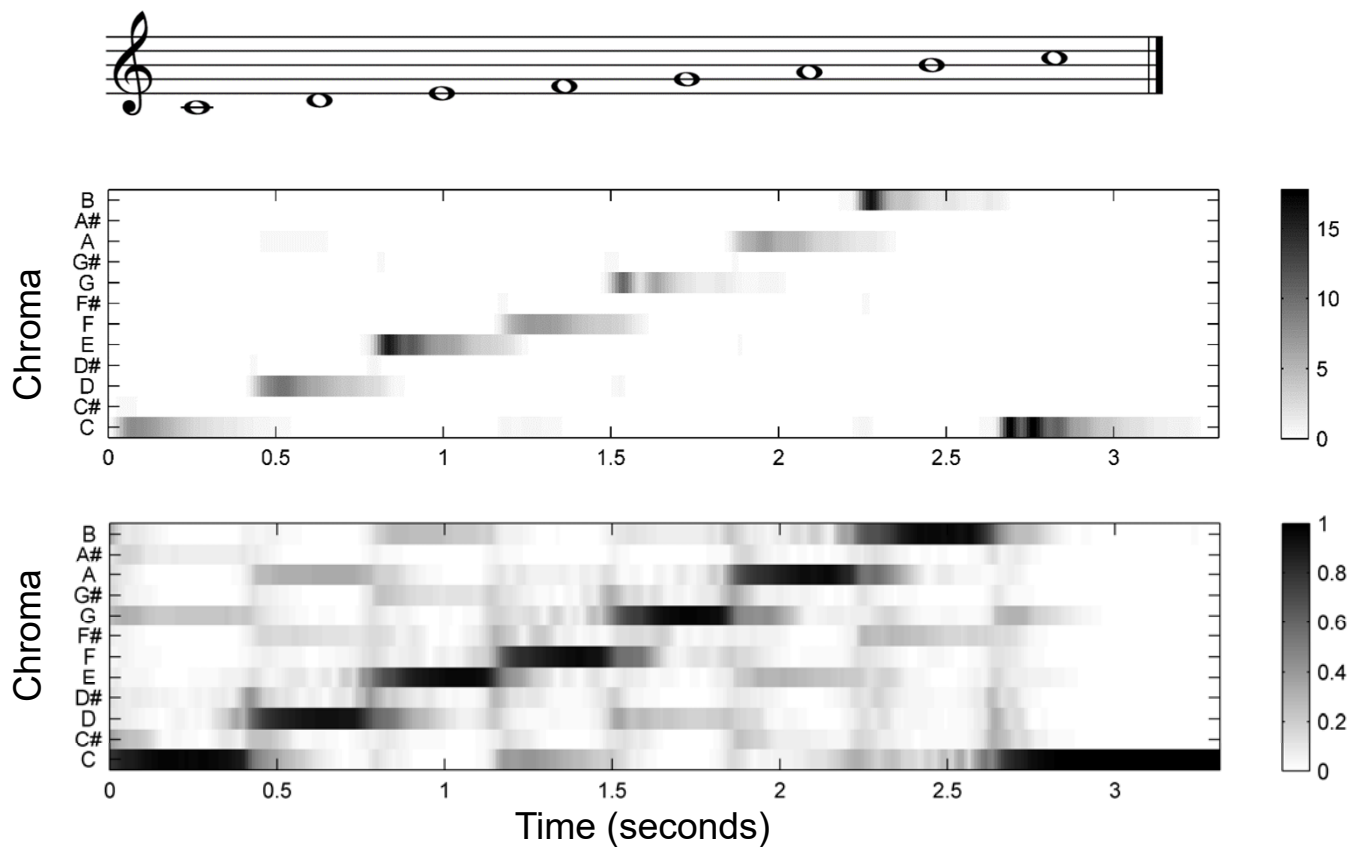
Audio Features

Chroma features



Audio Features

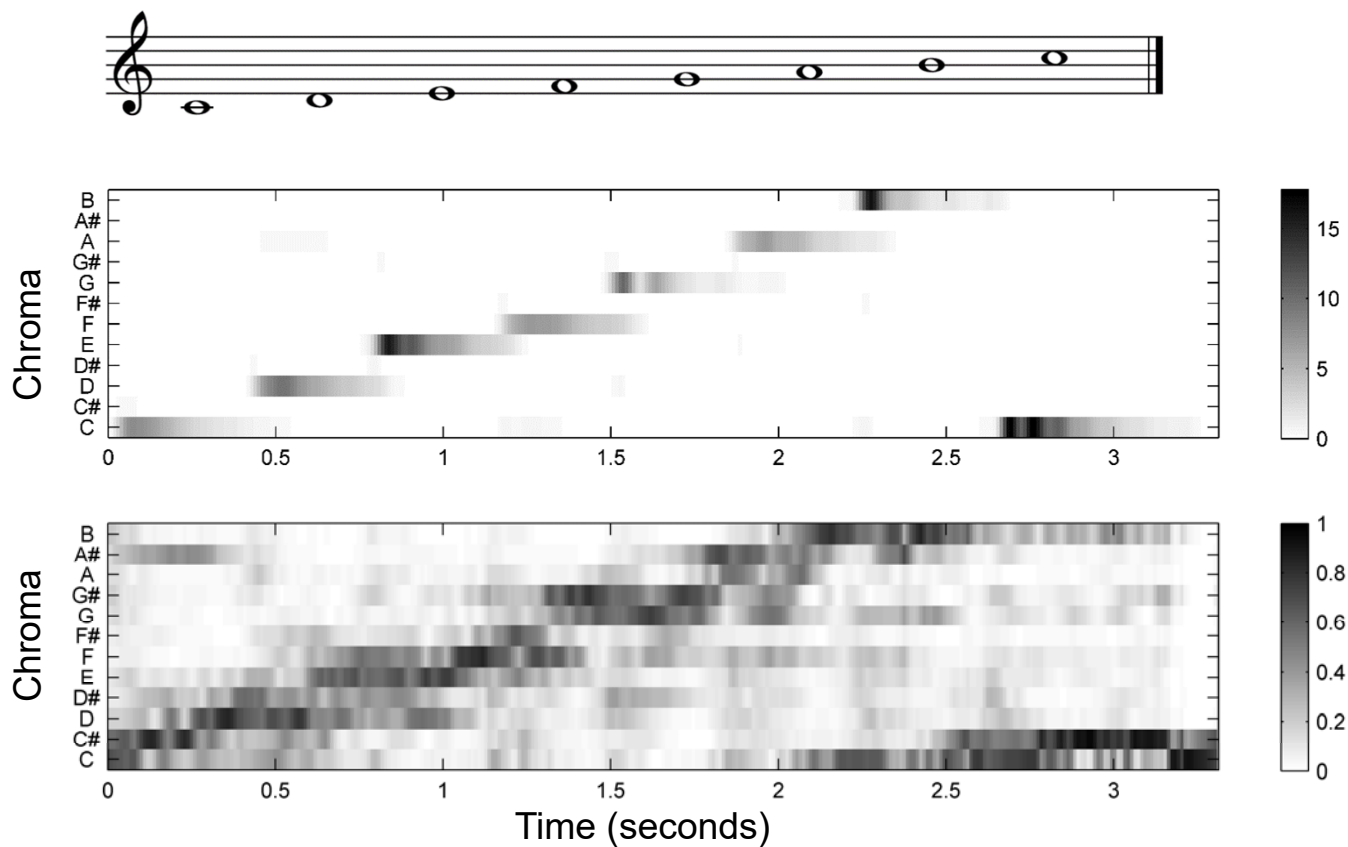
Chroma features



Effect of logarithmic compression and normalization

Audio Features

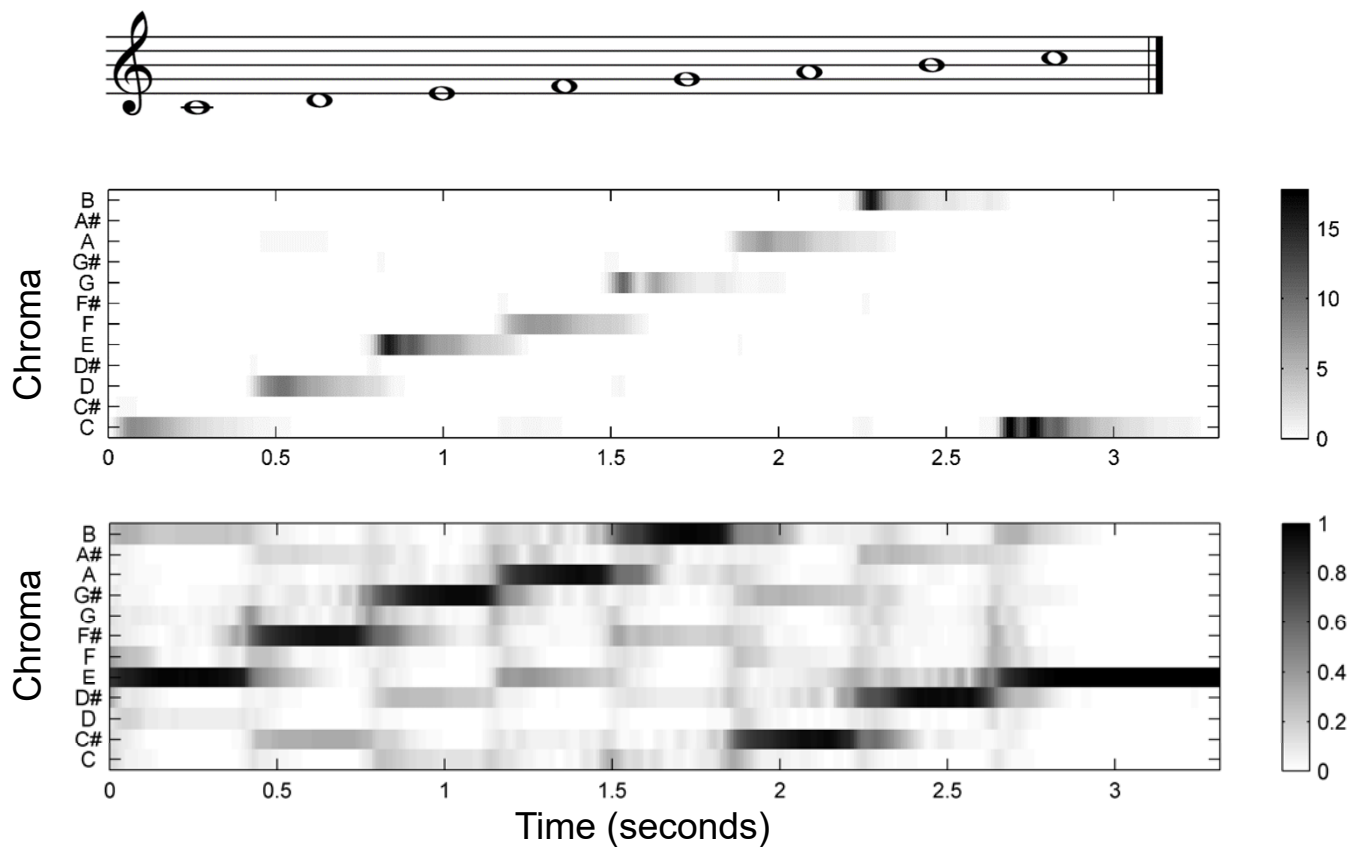
Chroma features



Piano tuned 40 cents upwards

Audio Features

Chroma features



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]