Learning-By-Doing: Using the FMP Python Notebooks for Audio and Music Processing

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

Guest Lecture
Central Conservatory of Music (CCOM)
Beijing, March 2023

Meinard Müller: Research Group
Semantic Audio Processing
- Michael Krause
- Yigitcan Özer
- Simon Schwär
- Johannes Zeilier
- Peter Meier (external)
- Christof Weiß
- Sebastian Rosenzweig
- Frank Zalkow
- Christian Dittmar
- Jonathoren Driedger
- Thomas Prätzlich
- ...

International Audio Laboratories Erlangen
- Fraunhofer Institute for Integrated Circuits IIS
- Largest Fraunhofer institute with ≈1000 members
- Applied research for sensor, audio, and media technology
- Friedrich-Alexander Universität Erlangen-Nürnberg (FAU)
- One of Germany’s largest universities with ≈40,000 students
- Strong Technical Faculty

Audio
Audio Coding
Psychoacoustics
Music Processing
3D Audio
Internet of Things
Music Processing: A Multifaceted Research Area

- Music is a ubiquitous and vital part of our lives
- Digital music services: Spotify, Pandora, iTunes, ...
- Music yields intuitive entry point to support and motivate education in technical disciplines
- Music bridges the gap between engineering, computer science, mathematics, and the humanities

Fundamentals of Music Processing (FMP)

Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
Springer, 2015

Accompanying website: www.music-processing.de

2nd edition
Meinard Müller
Fundamentals of Music Processing
Using Python and Jupyter Notebooks
Springer, 2021

Accompanying website: www.music-processing.de

Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
Springer, 2015

Accompanying website: www.music-processing.de

2nd edition
Meinard Müller
Fundamentals of Music Processing
Using Python and Jupyter Notebooks
Springer, 2021

Accompanying website: www.music-processing.de
FMP Notebooks: Education & Research

FMP Notebooks

Python Notebooks for Fundamentals of Music Processing

The FMP notebooks offer a collection of educational material closely following the textbook Fundamentals of Music Processing (FMP). This is the starting website, which is opened when calling https://www.audiolabs-erlangen.de/FMP. Besides giving an overview, this website provides information on the license, the main contributors, and some links.

https://www.audiolabs-erlangen.de/FMP

- ... provide educational material for teaching and learning fundamentals of music processing.
- ... combine textbook-like explanations, technical concepts, mathematical details, Python code examples, illustrations, and sound examples.
- ... bridge the gap between theory and practice being based on interactive Jupyter notebook framework.
- ... are freely accessible under a Creative Commons license.

FMP Notebooks: Education & Research

FMP Notebooks

Structured in 10 parts

- Part B: Basic introductions to
  - Jupyter notebook framework
  - Python programming
  - Other technical concepts underlying these notebooks

Structured in 10 parts

- Part B: Basic introductions to
  - Jupyter notebook framework
  - Python programming
  - Other technical concepts underlying these notebooks

- Part 0: Starting notebook
FMP Notebooks
Structured in 10 parts

- Part B: Basic introductions to
  - Jupyter notebook framework
  - Python programming
  - Other technical concepts underlying these notebooks

- Part 0: Starting notebook

- Part 1 to Part 8: Different music processing scenarios

Part 6: Tempo and Beat Tracking

- When listening to a piece of music, we as humans are often able to tap along with the musical beat
- Automated beat tracking: Simulate this cognitive process by a computer

Example: Queen – Another One Bites The Dust

Time (seconds)
Tempo and Beat Tracking

Tasks
- Onset detection
- Beat tracking
- Tempo estimation

Onset Detection (Spectral Flux)

Audio recording

Tempo := 60 / period

Beats per minute (BPM)
Onset Detection (Spectral Flux)

Compressed spectrogram

Steps:
1. Spectrogram
2. Logarithmic compression

Onset Detection (Spectral Flux)

Spectral difference

Steps:
1. Spectrogram
2. Logarithmic compression
3. Differentiation & half wave rectification
4. Accumulation

Onset Detection (Spectral Flux)

Novelty curve

Steps:
1. Spectrogram
2. Logarithmic compression
3. Differentiation & half wave rectification
4. Accumulation
5. Normalization

Onset Detection (Spectral Flux)

Normalized novelty function

Steps:
1. Spectrogram
2. Logarithmic compression
3. Differentiation & half wave rectification
4. Accumulation
5. Normalization
Onset Detection (Spectral Flux)

Steps:
1. Spectrogram
2. Logarithmic compression
3. Differentiation & half wave rectification
4. Accumulation
5. Normalization

Normalized novelty function
Peak positions indicate beat candidates

Deep Learning
Steps:
1. Spectrogram
2. Logarithmic compression
3. Differentiation & half wave rectification
4. Accumulation
5. Normalization

Normalized novelty function
Peak positions indicate beat candidates

Local Pulse and Tempo Tracking

Fourier temogram (STFT of novelty function)

Optimizing local periodicity kernel

Local Pulse and Tempo Tracking

Fourier temogram (STFT of novelty function)

Optimizing local periodicity kernel

Onset Detection (Spectral Flux)
Local Pulse and Tempo Tracking

Fourier temogram (STFT of novelty function)

Optimizing local periodicity kernel

Accumulation of kernels

Halfwave rectification

Novelty Curve

Predominant Local Pulse (PLP)

FMP Notebooks

Structured in 10 parts

- Part B: Basic introductions to
  - Jupyter notebook framework
  - Python programming
  - Other technical concepts underlying these notebooks
- Part 0: Starting notebook
- Part 1 to Part 8: Different music processing scenarios

Part B: Basics

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Started</td>
<td>Explanation on how to install and use the FMP notebooks</td>
</tr>
<tr>
<td>Installation</td>
<td>Installation of Python using Conda</td>
</tr>
<tr>
<td>Jupyter Notebook</td>
<td>Usage of Jupyter notebook framework</td>
</tr>
<tr>
<td>Python Basics</td>
<td>Introduction of data types, control structures, and functions</td>
</tr>
<tr>
<td>Python Style Guide</td>
<td>Recommendations for programming style</td>
</tr>
<tr>
<td>Multimedia</td>
<td>Integration of multimedia objects into notebooks</td>
</tr>
<tr>
<td>Python Visualization</td>
<td>Generation of figures and images</td>
</tr>
<tr>
<td>Python Audio</td>
<td>Reading and writing audio files</td>
</tr>
<tr>
<td>Numba</td>
<td>Acceleration of Python functions via JIT compilation</td>
</tr>
<tr>
<td>Annotation Visualization</td>
<td>Visualization of annotations (single value, segments)</td>
</tr>
<tr>
<td>Sonification</td>
<td>Sonification methods (onsets, F0 trajectories, pitch, chroma)</td>
</tr>
<tr>
<td>Libs</td>
<td>Library of FMP-specific Python functions</td>
</tr>
<tr>
<td>MIR Resources</td>
<td>Links to resources that are useful for MIR</td>
</tr>
</tbody>
</table>
Part B: Basics
Annotation Visualization
Examples for visualizing annotations of time positions and segments.

Part 1: Music Representations
Symbolic Format: CSV
Visualization of a piano-roll representation (Fugue BWV 846 by Bach).

Part 1: Music Representations
Waves and Waveforms
Videos illustrating the concepts of transverse, longitudinal, and combined waves.

Part 2: Fourier Analysis of Signals
Discrete Fourier Transform (DFT)
The matrix $\text{DFT}_N$ and a visualization of its real and imaginary parts for the case $N = 32$.

Part 2: Fourier Analysis of Signals
STFT: Padding
Time-domain signal and magnitude Fourier transform.

Part 2: Fourier Analysis of Signals
Digital Signals: Quantization
Uniform and nonuniform quantization (based on $\mu$-law encoding) using $\mu = 6$ quantization levels.
Part 3: Music Synchronization
Transposition and Tuning

Tuning procedure using a comb-filter approach.

Music synchronization result obtained for two input chromagrams (obtained from two recordings of the beginning of Beethoven’s Fifth Symphony).

Part 4: Music Structure Analysis
SSM: Synthetic Generation

Structure annotation and different synthetically generated SSMs.

Part 5: Chord Recognition
Template-Based Chord Recognition

Chord recognition task illustrated by the first measures of the Beatles song “Let It Be.”

Part 6: Tempo and Beat Tracking
Novelty: Comparison of Approaches

Comparison of novelty detectors using a matrix-based visualization.

Prefiltering experiments for a template-based and an HMM-based chord recognizer applied to two different input chroma representations (STFT, CQT).

The evaluation is performed on the basis of four Beatles songs (LetItB, HereCo, ObLaDi, PennyL).
Part 6: Tempo and Beat Tracking
Cyclic Tempogram

Different tempogram representations of a click track with increasing tempo.

Part 6: Tempo and Beat Tracking
Adaptive Windowing

Example of adaptive windowing using a parameter $\alpha$ to control the neighborhood’s relative size to be excluded.

Part 7: Content-Based Audio Retrieval
Audio Identification

Evaluation measures that indicate the agreement between two constellation maps computed for an original version (Reference) and a noisy version (Estimation).

Part 7: Content-Based Audio Retrieval
Audio Matching

Transposition-invariant matching function illustrated by Zager and Evans’ song “In the Year 2525.”

Part 7: Content-Based Audio Retrieval
Evaluation Measures

Various evaluation metrics applied to a toy example.

Part 8: Audio Decomposition
Instantaneous Frequency Estimation

Interpretation of time–frequency bins of an STFT using (frame-dependent) instantaneous frequency values.
Part 8: Audio Decomposition

Fundamental Frequency Tracking

Salience representation with trajectories computed by
(a) a frame-wise approach,
(b) an approach using continuity constraints, and
(c) a score-informed approach.

Time (seconds)

Frequency (cents)

Time (seconds)

Time (seconds)

(a) (b) (c)

Nonnegative Matrix Factorization (NMF)

Matrix V and randomly initialized matrices W and H.

Matrix V and matrices W and H after training.

Error terms over iteration.

NMF procedure applied to a toy example.

https://www.audiolabs-erlangen.de/FMP

https://www.audiolabs-erlangen.de/FMP

https://www.audiolabs-erlangen.de/FMP

Basics + 8 Chapters

Basics + 8 Chapters

Tempo and Beat Tracking

Tempo and Beat Tracking

https://www.audiolabs-erlangen.de/FMP
Definition

We assume that we are given a discrete-time novelty function \( \Delta : \mathbb{Z} \to \mathbb{R} \) that indicates note onset candidates. The idea of Fourier analysis is to detect its peaks and valleys. A high correlation of \( \Delta \) with a windowed sine wave indicates a periodicity of the structure (given a suitable phase). This correlation (along with the phase) can be obtained by the short-time Fourier transform. To this end, we fit a window function \( w(t) \) of length \( N \), centered at \( n = 0 \) (e.g., a sampled Hann window). Then, for a free parameter \( \omega \in \mathbb{R} \) and time parameter \( n \in \mathbb{Z} \), the complex Fourier transform of \( \Delta \) is defined by

\[
\mathcal{F}(\Delta, \omega) := \sum_{n=-N/2}^{N/2-1} \Delta(n) w(n-m) \exp(-2\pi i \omega m).
\]
### Talk Outline

- Tempo and Beat Tracking
- Visualization
- Results
- Evaluation
- Sonification

### FMP Notebooks

- Teaching
- Understanding
- Programming
- Baselines
- Research
- Multimedia

### Resources (Group Meinard Müller)

- FMP Notebooks:
  - [https://www.audiolabs-erlangen.de/FMP](https://www.audiolabs-erlangen.de/FMP)
- libfmp:
  - [https://github.com/meinardmueller/libfmp](https://github.com/meinardmueller/libfmp)
- synctoolbox:
  - [https://github.com/meinardmueller/synctoolbox](https://github.com/meinardmueller/synctoolbox)
- libtsm:
  - [https://github.com/meinardmueller/libtsm](https://github.com/meinardmueller/libtsm)
- Preparation Course Python (PCP) Notebooks:
  - [https://www.audiolabs-erlangen.de/resources/MIR/PCP/PCP.html](https://www.audiolabs-erlangen.de/resources/MIR/PCP/PCP.html)
  - [https://github.com/meinardmueller/PCP](https://github.com/meinardmueller/PCP)

### Resources

- librosa:
  - [https://librosa.org/](https://librosa.org/)
- madmom:
  - [https://github.com/CPJKU/madmom](https://github.com/CPJKU/madmom)
- Essentia Python tutorial:
  - [https://essentia.upf.edu/essentia_python_tutorial.html](https://essentia.upf.edu/essentia_python_tutorial.html)
- mirdata:
  - [https://github.com/mir-dataset-loaders/mirdata](https://github.com/mir-dataset-loaders/mirdata)
- open-unmix:
  - [https://github.com/sigsep/open-unmix-pytorch](https://github.com/sigsep/open-unmix-pytorch)
- Open Source Tools & Data for Music Source Separation:
  - [https://source-separation.github.io/tutorial/landing.html](https://source-separation.github.io/tutorial/landing.html)