

MUSIC INFORMATION RETRIVAL IN MATLAB

A toolbox for musical feature extraction from audio

By Olivier Lartillot and Petri Toiviainen
University of Jyväskylä, Finland

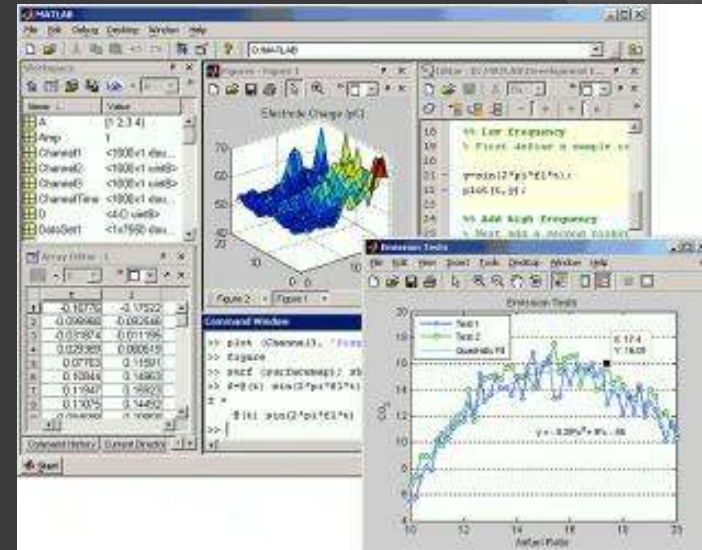
Outline

- ⦿ Motivation
- ⦿ Feature Overview
- ⦿ Example of Use (Rhythmic pulsation)
- ⦿ Toolbox Design
- ⦿ Related Work
- ⦿ Relation to Project
- ⦿ Article Critics

Motivation

● Matlab

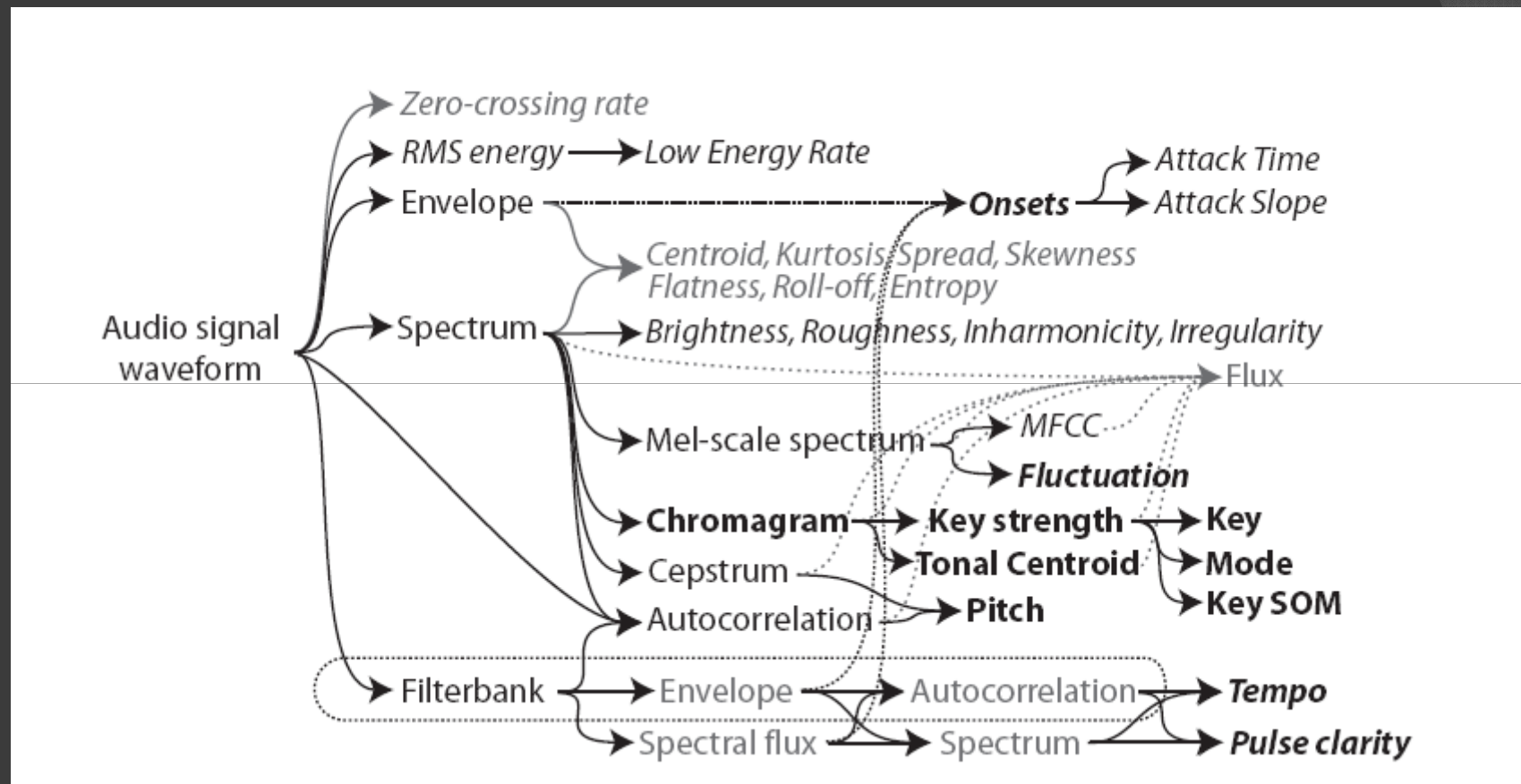
- Signal Processing Tool
- Visualisation Tool
- Math Framework
- Partial MIR frameworks exists:
 - Auditory Toolbox, NetLab, SOMtoolbox
- Lets combine to a **complete** framework



Motivation Continued

- ⦿ High interdependability between computation blocks (building blocks)
- ⦿ Prevent recomputations
- ⦿ Aim on ease of use
 - Overloaded methods for all kinds of inputs
 - High focus on syntax

Feature Overview



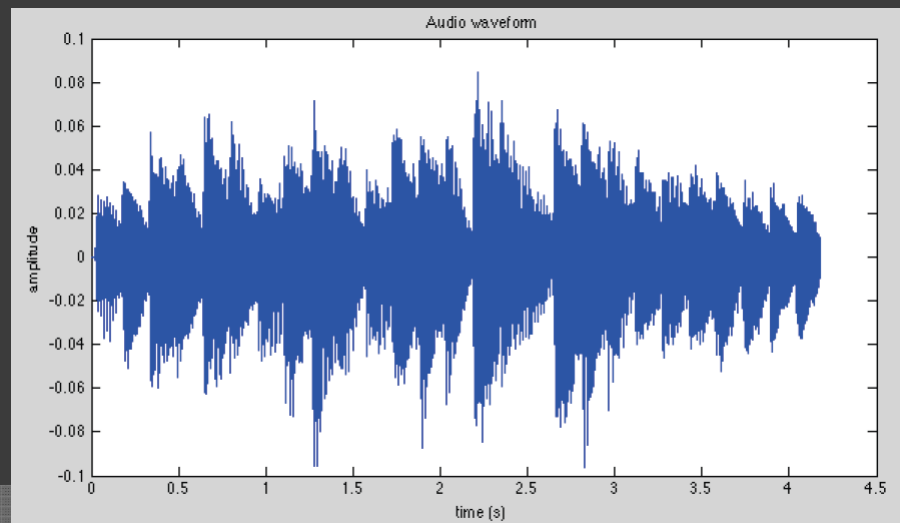
- **Pitch and tonality**
- ***Rhythm***
- ***Timbre and Dynamics***

Example - Rhythmic analysis

- ⦿ Aim to determine the 'tempo' of a song
- ⦿ Illustrate the syntactic use of MIRtoolbox
- ⦿ Explain central functions of MIRtoolbox

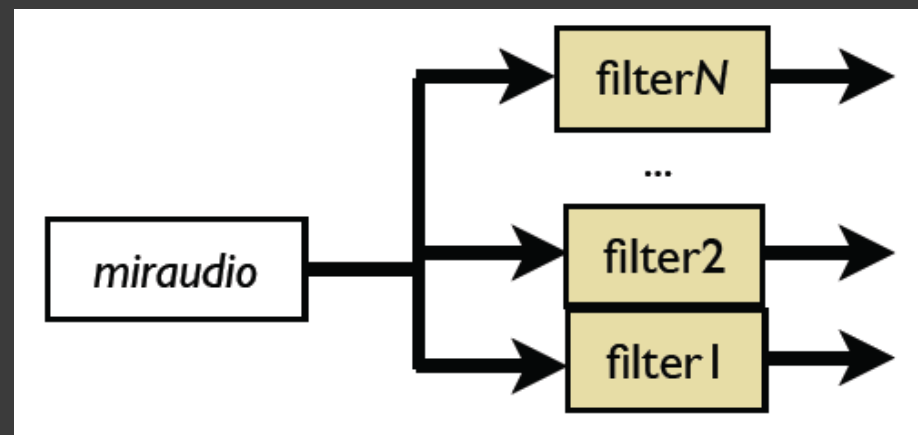
Example – miraudio()

- Preload of WAV and AU files
- Enables various operations on the input
 - *miraudio(..., 'Center')*
 - *miraudio(..., 'Sampling', r)*
 - *miraudio(..., 'Extract', t1, t2)*
 - *miraudio(..., 'Trim')*



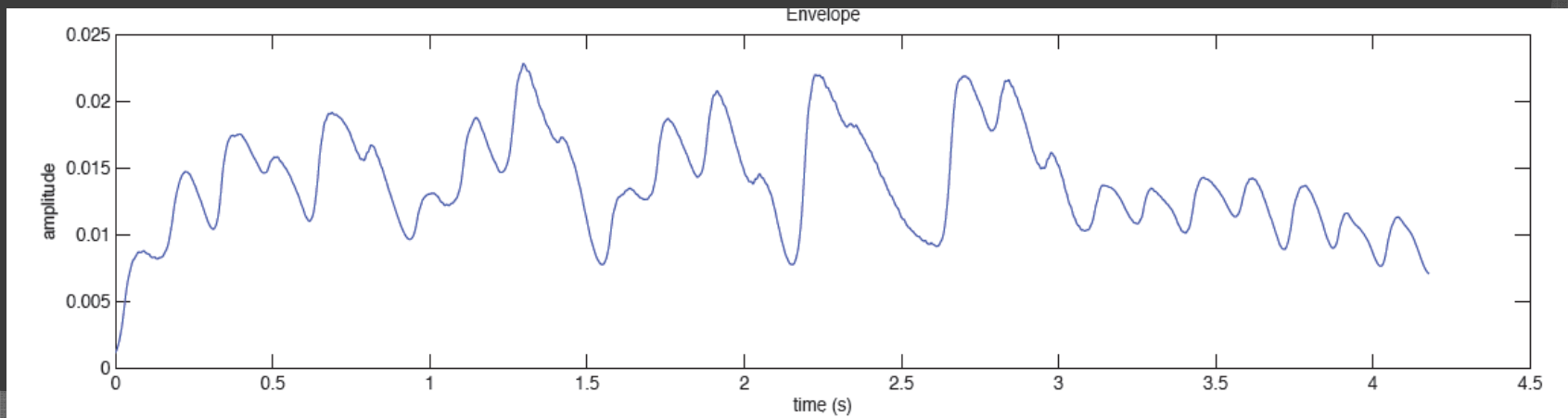
Example – mirfilterbank ()

- Filters the input using specified filter(s)
- Decomposes an input into channels
 - Audio input decomposed into frequencies
 - `mirfilterbank(a,20)`



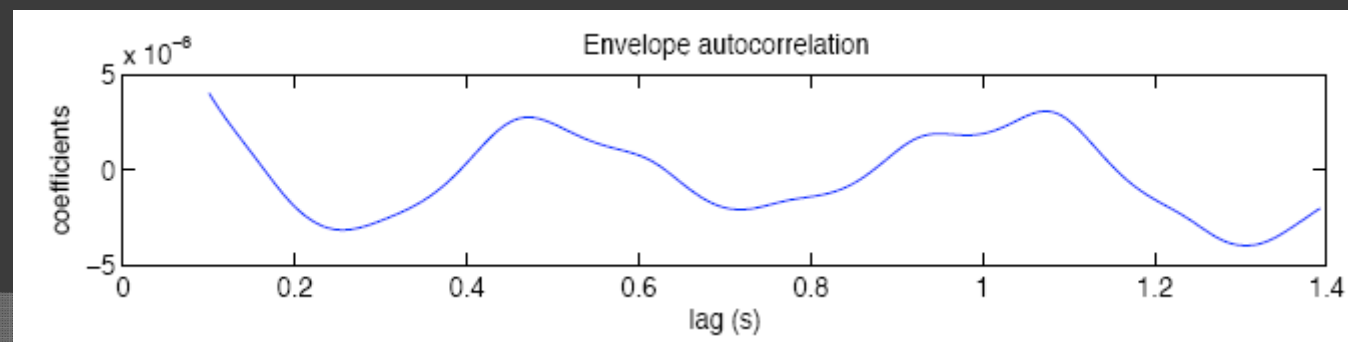
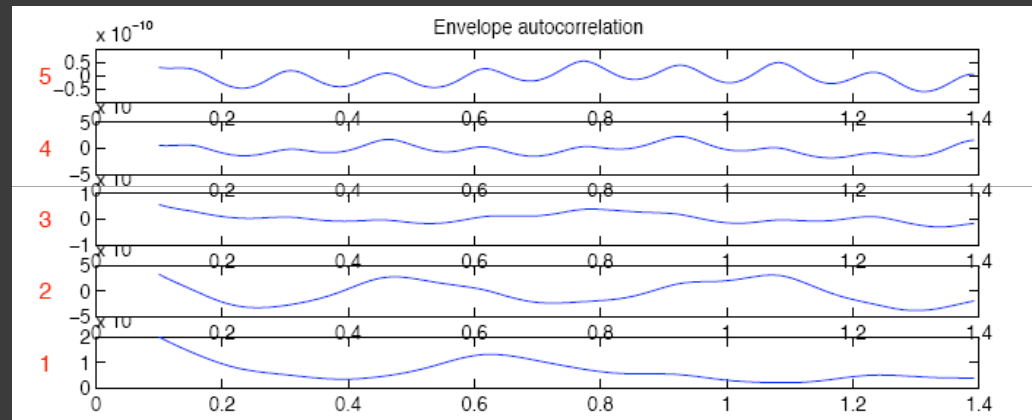
Example – mirenvelope()

- Formed curve by the global outer shape of a signal
 - Negative parts of signal reflected as positive
 - Too frequent oscillations gets removed
- Useful when showing signal evolution



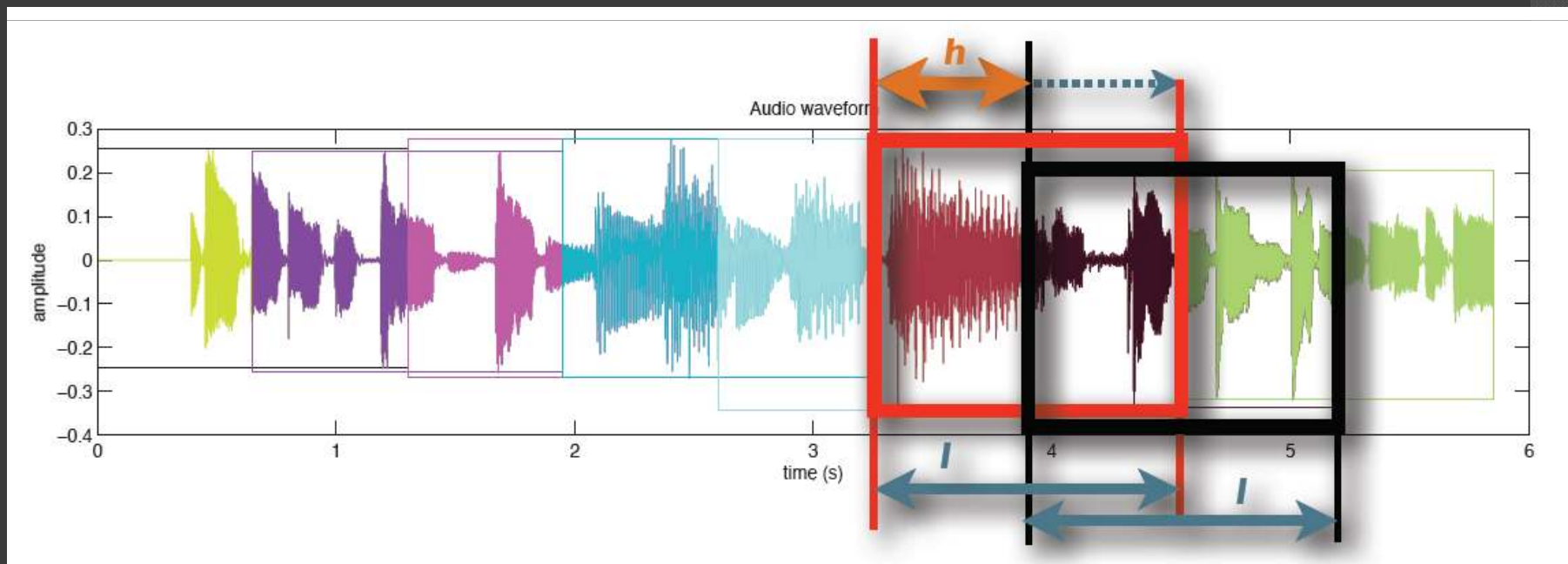
Example – mirsum()

- Composes channels into one global signal



Example – mirframe()

- Audio have a tendency to be dynamic
- `mirframe()` divides the signal into frames of time

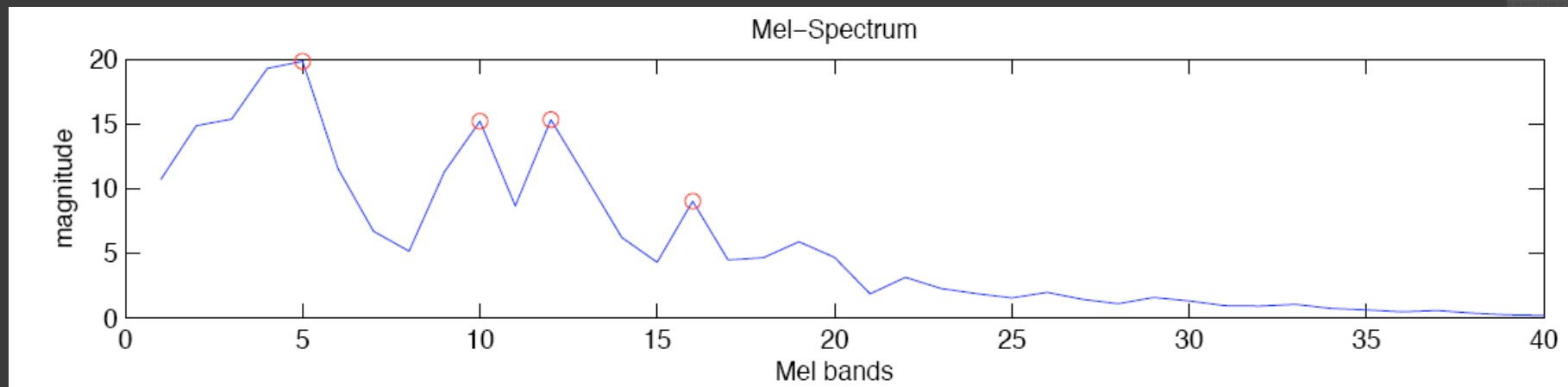


Example – mirautocor()

- Performing **autocorrelation** yields information about repeating events
- Multiplication between the signal and a shifted version of itself
- Results in a graph illustrating peaking patterns

Example – mirpeaks()

- Picks important local maxima



Example – mirtempo() & mirhisto()

- mirtempo() calculates Beats Per Minute (BPM)
- mirhisto() calculates a histogram

Example – Rhythm Sum-Up

1. Preload the audiofile into memory
2. Decomposes the signal into channels of frequency
3. Extract the "envelope" of the channels, differentiate halfwave to find "peaks" in each channel
4. Sum the envelopes together
5. Since a piece of music varies in pulsation, we decompose into frames of static length
6. Find repeating events (pulses) through autocorrelation
7. The peaks from the periodigram gets selected
8. Convert it into BPM (Beats per minute)
9. Frames may vary in BPM, get histogram

```
a = miraudio(myAudio.waw)
fb = mirfilterbank(a, 20)

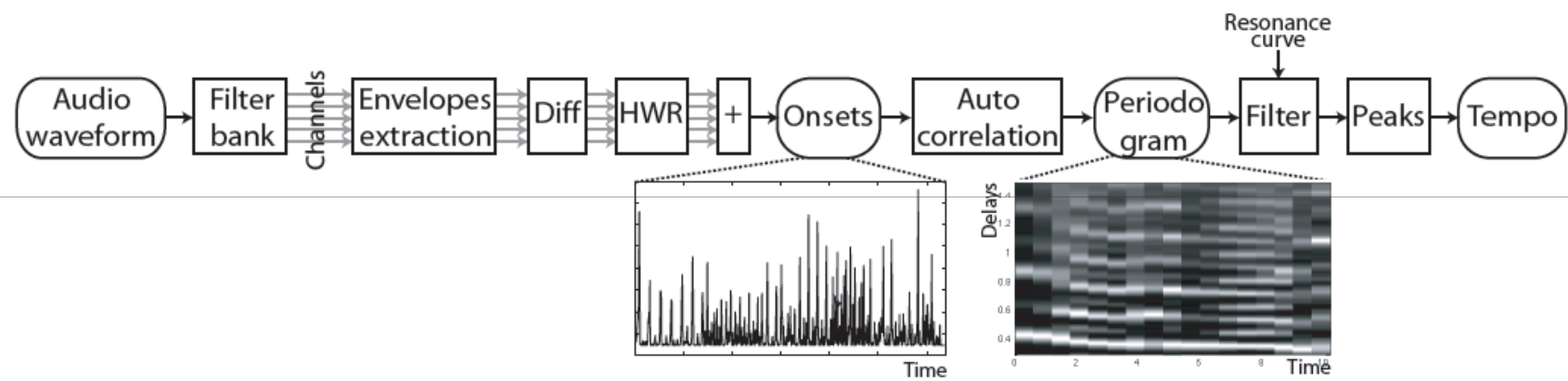
e = mirenvelope(fb, 'Diff',
                'HalfWave', 'Center')

s = mirsum(e, 'Center')
fr = mirframe(s, 3, .1)

ac = mirautocor(fr, 'Resonance')

p = mirpeaks(ac, 'Total', 1)

t = mirtempo(p)
h = mirhisto(t)
```



Example - Rhythmic analysis, simplified:

mirtempo(a, 'Frame')

Data:

- Visualization of all objects is possible.
- Default display type in Matlab is a graphical representation.
- List of data is accessible through `mirgetdata`.

Related Works

- ◎ Marsyas
 - Scripting language
 - Example: 20 lines in marsyas = 4 lines in MIRtoolbox
- ◎ jAudio
 - Framework in java

Relation to Project

- ⦿ Our project
 - String representation of a song
- ⦿ Relation
 - MIRtoolbox offers tools for tonality feature extraction
 - String representation = tonality

Article Critics

⦿ Negatives

- Short article
- Too compressed
 - Unclear article from chapter 3 and on
- Lack of technical explanation (example)

⦿ Positives

- User-Guide adds support to the article and especially the example
- Easy to read until chapter 3
- In general well written language



Memory Optimization

- ◎ Reflexive data representation is needed
 - Folders as input, could cause full memory
 - Loads one file of time
 - Even long audiofiles gets splitted
- ◎ Developed memory management functions
 - Assures safety at custom development

Adaptive Syntax:

- ◎ Adaptive inputs – heavy overloading
 - `mirspectrum('myfile')` (single file)
 - `mirspectrum('folder')` (batch)
 - `a = miraudio('..', '..', '..')` (preloading)
 - `s = mirspectrum(a)`
 - `mirautocor(a) * mirautocor(s)` (adaptive product)