Exercise Track

Audio and Music Processing

22. Juni 2013

Wolfgang Küllinger (0955711)
Fabian Jordan (0855941)

Inhaltsverzeichnis

1 Einleitung

2 Architecture

2.1 Class structure

The project was separated into several classes. The given framework simple calls methods which are defined in the package `at.jku.amp.lepatriinu`.

- `at.jku.amp.lepatriinu.Analyzer`
- `at.jku.amp.lepatriinu.BeatDetector`
  - `at.jku.amp.lepatriinu.AutoCorrelationBeatDetector`
- `at.jku.amp.lepatriinu.OnsetDetector`
  - `at.jku.amp.lepatriinu.GroundTruthPicker`
  - `at.jku.amp.lepatriinu.HighFrequencyOnsetDetector`
  - `at.jku.amp.lepatriinu.SimpleOnsetDetector`
  - `at.jku.amp.lepatriinu.SpectralFluxOnsetDetector`
- `at.jku.amp.lepatriinu.TempoExtractor`
  - `at.jku.amp.lepatriinu.InterOnsetTempoExtractor`

2.1.1 `Analyzer`

The *Analyzer* class defines all the constants that are needed for any algorithm in the project to succeed. In order to make central configurations possible, the constants are collected in this single class.

Furthermore it provides the central interface to the `at.cp.jku.teaching.amprocessing` project. It is initialized with a pre-processed (e.g. FFT) audio file of type `Audiofile`. The order of usage is important. In the first place, onset detection can be done. This information can directly be retrieved from the audio file. The found onsets are the base for tempo extraction. Therefore the onset list has to be provided as parameter of the tempo extraction function. Finally beat detection can be performed. In order to be able

to use the best algorithms both the onset list as well as the calculated tempo should be provided.

1. Onset detection: `onsets = performOnsetDetection()`
2. Tempo extraction: `tempo = performTempoExtraction(onsets)`
3. Beat Detection: `beats = performBeatDetection(onsets, tempo)`

Constants

In the following section the used constants are explained.

Constant	**Type**	**Default value**	**Description**
ONSET_DETECTOR	OnsetDetector	OnsetDetector.FLUX	specifies the used onset detection algor

Tabelle 2.1: Constants in `Analyzer`

2.2 Graphical user interface

3 Evaluierung