JOHANNES KEPLER
UNIVERSITY LINZ

SPECIAL TOPICS

Audio and Music Processing - Exercise Track
344.032
KV, 2h, SS2020

Jan Schlüter
Institute of Computational Perception

PRELIMINARIES

- exercises can be done in **pairs** or **alone**
- we set up a forum in KUSSS to help finding group partners
- a small Java framework will be provided, you can implement your algorithms in any language though
- using ready-made onset detectors, beat trackers or tempo estimators from an MIR framework **does not count** of course! you have to implement your approaches yourself!

GETTING A NAME

- write an email to **jan.schlueter@jku.at**
- use this **email header prefix** “**[SPEZKAP20]**”
- list the group members (1 or 2) with **name and email address**
- you will be assigned a **group name**

GOALS OF THE EXERCISE

- implement **at least two onset detection** methods
- implement **at least one tempo extraction** method
- implement **at least one beat detection** method
- your methods **may share** any pre-/post-processing or peak-picking methods
- you will have to **decide** on a **final combination** of onset detection / tempo extraction / beat detection methods that produce your **final predictions**
- you may of course combine several methods that do the same thing (“ensemble”)
- your tempo extraction may also be based on other features besides the detected onsets

DATA DISTRIBUTION

- both the framework and data for training or development of your algorithms is available at:
  `http://teacap.cp.jku.at/files/spezkap_amp/2020`,
  user `spezkap_amp`,
  password `2020onsets`

TRAINING DATA

- you will find a training dataset consisting of:
  - 127 excerpts
  - excerpts are between 3 seconds and 2 minutes long
  - excerpts span across various genres
  - you can assume (nearly!) constant tempo over the excerpt
  - it might improve your results, if you allow for slight tempo variations
- for each excerpt you are also provided with 3 additional text files
  - WAVNAME.onsets.gt is a list of all **onset times**
  - WAVNAME.beats.gt is a list of all **beat times**
  - WAVNAME.tempo.gt contains multiple **tempo annotations**

A NOTE ON TEMPO ANNOTATIONS

- WAVNAME.tempo.gt needs a few explanations
- humans rarely agree on the exact tempo of a piece of music
- that is why there are often two tempi and a weighting
- $60 \;\; 120 \;\; 0.8$ means that 80% of the people said that the tempo is 60 [bpm], and 20% of the people say it is actually 120 [bpm]
- $60 \;\; 120 \;\; 0.1$ means that 10% of the people chose the first, and 90% the second tempo (the slower one always comes first)
- 60 means that there is a single annotation only (for 60 [bpm])
- the **evaluation** will **only** use the **tempo** that received the **majority vote**

EXTRA TRAINING DATA

- you will also find two additional training datasets
- one of 151 excerpts annotated with onsets only
- one of 696 excerpts annotated with beats and tempo only
- you may use these as additional material for training or validating your algorithms, or ignore them

TEST DATA

- in the end, you will have to run your algorithm on unseen **test data**
- for this, we provide 50 excerpts **without annotations**
- to be clear: this is not meant for you to validate your algorithms (you will need to reserve training data for this), but for the final submission

THE JAVA FRAMEWORK (1)

- the Java framework is available at the same download link as the data
- the framework provides simple ways to:
  - □ read in audio files in the WAV format
  - □ compute the STFT (magnitude, phase, unwrapped phase)
  - □ evaluate predictions, given a ground truth
- you are **free to change everything**

THE JAVA FRAMEWORK (2)

- the framework takes 5 parameters:
  - -i input directory
  - -n processor name
  - -p predict onsets, beats and tempo
  - -e evaluate all predictions that have a groundtruth
  - -s summarize all evaluations
- the **output** is written to the **input directory** as well:
  - WAVNAME.onsets.pr (estimated onset time)
  - WAVNAME.beats.pr (estimated beat positions)
  - WAVNAME.tempo.pr (estimated tempo)
  - summary.[onsets|beats|tempo].ev.txt files

THE JAVA FRAMEWORK (3)

- everything lives in the `at.jku.cp.spezi` package
- example code is in `at.jku.cp.spezi.example.TooSimple`
- you can put your code in `at.jku.cp.spezi.<shortname>.<ShortName>`
- if you got assigned the short name “`Alpha`” then you put your code in `at.jku.cp.spezi.alpha.Alpha`
- you could structure your code **beneath** this package
- but you **do not have to** …

THE JAVA FRAMEWORK (4)

- you might want to use `maven`, there is a `pom.xml` provided
- you build your code with `mvn package`
- your `jar` file will be in the `target` folder
- it'll be named `Spezi-1.0.jar`
- call it with `java -jar target/Spezi-1.0.jar` for instructions
- you can generate an Eclipse project via `mvn eclipse:eclipse`

VISUALIZING

- to see what your algorithms are doing, or to inspect the groundtruth, you can load the onset and the beat data into Sonic Visualizer
- load the WAV file first (File → Open)
- import the annotation (File → Import Annotation Layer)

EVALUATION (1)

- the framework already includes an evaluation procedure
- onset detection:
  - every onset estimate which is within ± 50 [ms] of an actual onset is counted as a true positive (TP)
  - for each actual onset only one onset estimate is allowed, others are counted as false positives (FP)
  - every onset estimate outside of the window is counted as a false positive too
  - every onset for which there is no predicted onset in the window, is counted as a false negative (FN)
  - **F-measure** is the important measure here

EVALUATION (2)

- beat detection:
  - □ the same as for onset detection, but with a bigger window of ± 70 [ms]
- tempo estimation:
  - □ if the estimate is within $\pm 4\%$ of the actual tempo, this is counted as correct

SUBMISSION

- **deadline** is on **June 17**
- submission happens via **email** to **jan.schlueter@jku.at**
- **please** use the subject header prefix “**[SPEZKAP20]**”
- use one of `[zip|tar|gzip|7z]`
- the following **must** be present in the archive:
  - □ **only** the **source** of your program
  - □ instructions on how to compile/run it, if you do not use the provided framework
  - □ a **folder** named `predictions`, containing the **final predictions** you made on the test set
  - □ exactly **1** slide, for a **2** minute presentation of your efforts **in PDF format**
  - □ a 2–3 page **description** of your approach, and the experiments you did (also in PDF format)

CONTENTS OF THE DESCRIPTION

- **which** methods you chose and **why**
  (bonus points, if you include **quantitative** evaluation)
- describe which kind of experiments you ran, different things you tried, funky ideas you had, . . .
- it is more important for us to see that you experimented with different methods, tried to understand the difficulties, than the actual performance of the final system
- curiosity and creativity will be rewarded

SUBMISSION DISCUSSION

- on **June 24** there will be a **presentation** session as well as a **discussion**
- **attendance** is **compulsory**!
- we will talk about:
  - □ the different approaches you implemented
  - □ problems and pleasant surprises with your implementations
  - □ evaluation of your systems on an independent test set
  - □ experiences with the exercise track
  - □ the lecture in general
  - □ beer and everything