Overscore

Quentin Stievenart

Université Libre de Bruxelles

April 26, 2013

Outline

- Project Goals
- Musical Notation Reminder
- Implementation
- Problems and Improvements
- Conclusion

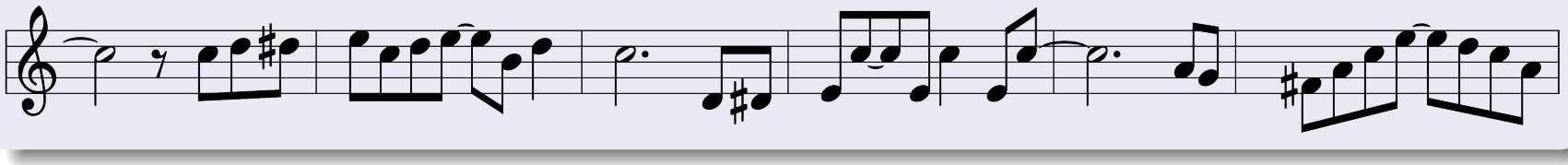
Project Goals

- Offer an alternative to the classical musical notation
  - Using Overtone[1]
- Build an OCR-like system for musical scores
  - OMR: *Optical Music Recognition*

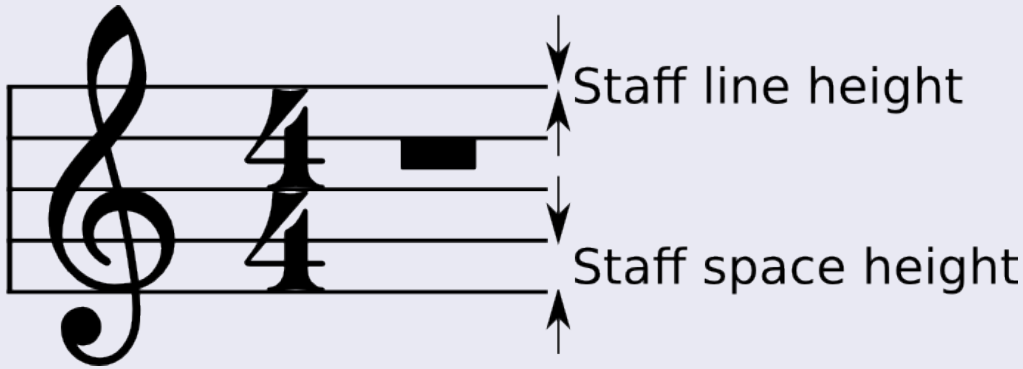
[1] `http://overtone.github.io/`

Musical Notation Reminder

Staff



Reference Lengths



Implementation

- **Input**: scanned musical score (`.png`)
- **OMR System**: image → MusicXML
  - Preprocessing
  - Staff Line Processing
  - Symbol Recognition
  - Musical Semantics
- **MusicXML Converter**: MusicXML → Overtone notation
- **Musical Notation**: plays file
- Each step is clearly delimited, and can be replaced with another implementation in another language

OMR System – Preprocessing

- **Input**: scanned, greyscale or color image
- **Output**: binary (black and white) image.
- Convert color image to greyscale
- Binarize greyscale image
- Compute reference lengths

Usage

```
$ lein2 run preprocessing \
            example/input/furelise.png \
            /tmp/furelise.png \
            /tmp/refs.txt
```

OMR System – Staff Line Processing

- **Input**: binary image
- **Output**: binary images and staves positions
- Done in two steps:
  1. **Identify** the positions of the staff lines
  2. **Remove** the staff lines

Usage

```
$ lein2 run staffline \
            example/preprocessed/furelise.png
```

OMR System – Symbol Recognition

- **Input**: binary image
- **Output**: list of segments with their class and position within the staff
- Done in two steps:
  1. **Segmentize** the image
     - Isolate each symbols in small segments
  2. **Classify** each segment

Usage

```
$ lein2 run segmentation \
            example/staffline/furelise-0.png \
            example/preprocessed/refs.txt \
            /tmp/segs.txt
$ lein2 run classification training-set \
            example/staffline/furelise-0.png \
            example/segmentation/segs.txt \
            /tmp/classes.txt
```

OMR System – Musical Semantics

- **Input**: segments with their class and position
- **Output**: MusicXML
- Group symbols together
- Interpret their meaning
  - eg. note pitch frome note head position, accidentals, ...

Usage

```
$ lein2 run semantics \
            example/classification/classes.txt \
            example/preprocessed/refs.txt \
            example/staffline/furelise-0.txt \
            /tmp/furelise-0.xml
```

Intermediate File Format

- **MusicXML**: widely used and supported, allows to represent complex scores without loss of information
- Parsed and converted into the notation using a set of simple rules

Usage

```
$ lein2 run generate \
            example/semantics/furelise-0.xml \
            /tmp/furelise-0.clj \
            furelise
```

Musical Notation

- Build on top of Overtone as a set of functions and macros

Usage

```
$ lein2 run play \
            example/generate/furelise-0.clj \
            furelise/furelise
```

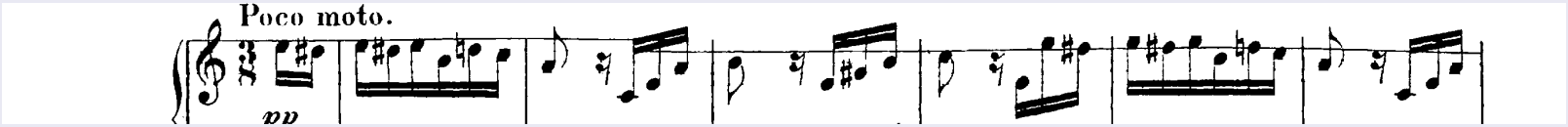
Outline

- OMR System
  - Preprocessing
  - Staff Line Processing
  - Symbol Recognition
  - Musical Semantics
- Intermediate File Format
- Musical Notation

Problems and Improvements – Preprocessing

- Most public domain scores are already binarized
- Could be improved with *deskewing* and *dewarping* algorithms, to avoid problems during staff line removal

Warped staff line



Problems and Improvements – Staff Line Processing

- Minor problems introduced when removing staff lines, but not critical

Problems and Improvements – Symbol Segmentation

- Not really documented in the litterature
- Far from perfect
- Highly dependent on the reference lengths (that can be tweaked)
- Problems to correctly segmentize both big and small segments

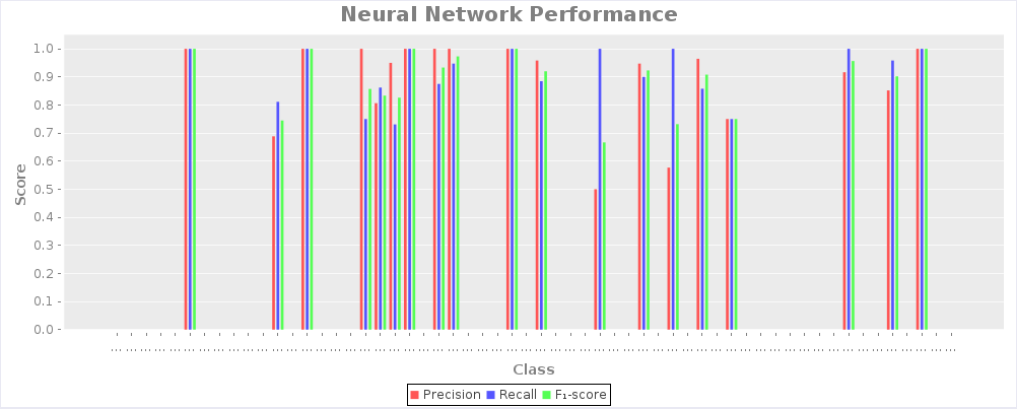
Incorrect Big Segments

Incorrect Small Segments

Problems and Improvements – Symbol Classification

- First implementation: kNN "by hand"
  - Way too slow
- Second implementation: neural network with *Encog*
  - Fast (except training time)
  - Not really accurate (correctness of 65% on example staff)
- Third implementation: kNN with *OpenCV*
  - Fast, no training time needed
  - Good accuracy (95% on example staff)
- Training set
  - Only two public training set availables:
    - OpenOMR: staff lines not removed, only 727 examples
    - Audiveris: stored as XML instead of images, 4159 examples
  - Used Audiveris' training set:
    - XML files converted to images (`lein2 run convert audiveris-training-set training-set`)
    - Symbols are not represented equally: note heads and beams covers 65% of the training set

Neural Network Performance



kNN Performance

Problems and Improvements – Musical Notation

- Miss some musical constructs
- Enough to support what is needed by the OMR system (and more)
- Could easily be extended to support more musical constructs

Conclusion

- OMR System: complete, but far from perfect
- Musical notation: not complete, but sufficient enough for what is supported by the OMR system
- Each part of the system can easily be replaced
- Available on GitHub:
  `https://github.com/acieroid/overscore`

Bibliography I