Optical Music Recognition – Program Design

Quentin Stievenart

January 18, 2013

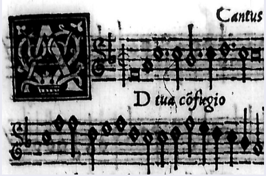
Outline

- Preprocessing
  - Binarization
  - Reference Lengths
  - Optional Techniques
- Staff Line Processing
  - Identification
  - Removal
- Symbol Recognition
  - Segmentation
  - Classification
- Musical Semantics
- Design choices

Preprocessing

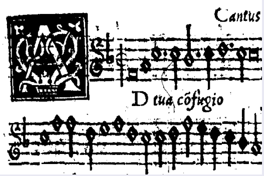
- **Input**: scanned, grey-scale or color image.
- **Output**: binary (black and white) image.

Input



(from [3])

Output



(from [3])

Preprocessing – Binarization

- First, convert the image in grey-scale using the following formula for each pixel[10]:

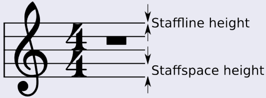
$$Grey = 0.3\ R + 0.59\ G + 0.11\ B$$

- **Otsu**'s method[13] is the most commonly and one of the fastest *global thresholding* method, according to [16].
- **Niblack**'s method[12] is the most commonly used *adaptive binarization* method, according to [16].
- The methods performing the best on degraded music sources are[3]:
  - **Brink and Pendock**, 1996[2].
  - **Pugin**, 2007, but the corresponding article couldn't be found.
  - **Gatos et al.**, 2004[8].

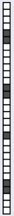
Preprocessing – Reference Lengths

- Also, extract reference lengths needed by the rest of the process (**staffspace height** and **staffline height**).
- Found using *Run-Length Encoding*[16][1]:
  - Encode each column of the binary image using RLE.
  - The most common black-run is the **staffline height**.
  - The most common white-run is the **staffspace height**.

Reference lengths



RLE for reference lengths



$$RLE = (4, 2, 6, 2, 5, 1, 6, 2, 3)$$

□ ■ □ ■ □ ■ □ ■ □

$\Longrightarrow$ Staffline height $= 2$
Staffspace height $= 6$

Preprocessing – Optional Techniques

- Noise reduction:
    - Eliminate isolated black pixels
    - Fill-in isolated white pixels
- Skew detection and correction[6]
    - The image can be skewed if the scan was not done perfectly horizontally.
- Less common[9]: enhancement, blurring, morphological operations.

Preprocessing – Choice

- To implement:
  1. **Otsu**'s method: simple and widely documented.
  2. The RLE-based algorithm to identify reference lengths.
- No other preprocessing.
- Should be enough for most (non-degraded) musical scores. If needed, a better binarization or other noise-reduction methods can be easily added afterwards.

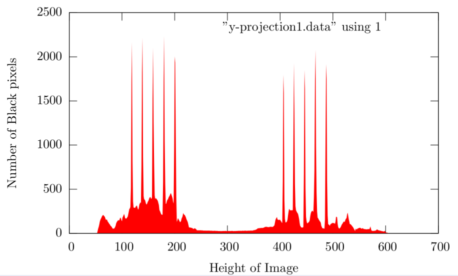
Staff Line Processing

- **Input**: binary image
- **Output**: binary image and staves positions
- **Identify** the positions of the staff
  - Needed to interpret the semantics of the symbols during the generation step.
- **Remove** the staff lines (optional)
  - Isolate the musical symbols (simplify segmentation step).
  - Reduces noise (simplify classification step).
  - Only a few OMR systems don't do this step.
- Challenging step if the staff lines are distorted.
  - We can improve results by adding a **skew correction** step in the preprocessing of the image.

Staff Line Identification

- Use vertical projections
- 5 maxima spaced of the staffspace height = a staff
- Store the position of each staff

Vertical projections



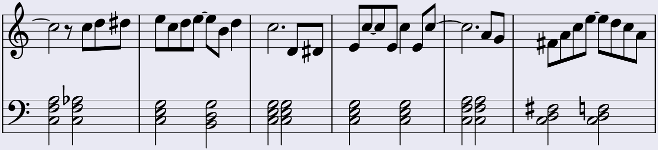
(from [6])

Staff Line Removal

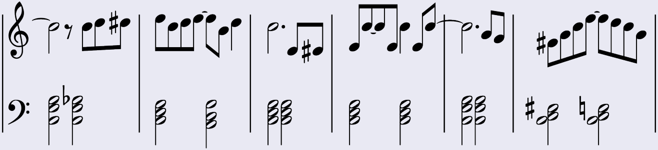
- Lots of algorithms:
  - Linetracking Runlength[14]
  - Linetracking Chord[11]
  - Carter[4]
  - Fujinaga[7]
  - Roach and Tatem[17]
  - Skeleton[5]
- No best algorithm according to [5], but for undeformed images, Roach and Tatem perform better than the others.

Staff Line Removal

Input



Output



Staff Line Processing – Choice

- To implement:
  1. Identification using vertical projections.
  2. Removal using **Roach and Tatem** algorithm (Open Source implementation of multiple staff line removal algorithms available[1]).

[1] http://music-staves.sf.net/

Symbol Recognition

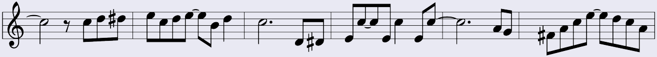
- **Input**: binary image (and staves positions)
- **Output**: list of musical symbols with their positions within the staff
- Done in two steps:
  1. Segmentation
  2. Classification

Symbol Recognition – Segmentation

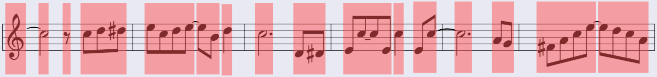
- **Input**: binary image
- **Output**: small binary images with their positions
- Not really described in the litterature, except for OpenOMR[6].
- OpenOMR's method:
  1. **Level 0 image segmentation** (L0) using horizontal projection: find groups of symbols.
  2. **Note head detection** using vertical projection: find which L0 segments contains note heads.
  3. **Symbol segmentation** (L1): separate symbols in L0 segments that contains note heads.
  4. **Note processing** (L2): seperate note heads from other symbols by doing:
     - Stem removal, using vertical projection
     - Staff line removal

Symbol Recognition – Segmentation (OpenOMR)

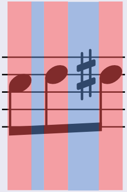
Input



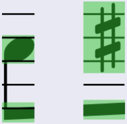
Level 0



Level 1



Level 2



Symbol Recognition – Classification

- **Input**: small binary image
- **Output**: corresponding musical symbol
- Uses well-known classification algorithms: **Neural Networks** (NN), **k-Nearest Neighbors** (kNN), **Support Vector Machines** (SVM), **Hidden Markov Models** (HMM).
- According to [15]:
  - The two best methods are SVM and kNN.
  - For handwritten music, SVM performs better than kNN.
  - For typesetted music, kNN performs better than SVM.
- Training sets:
  - OpenOMR: training set for methods without staffline removal, 727 files.
  - Audiveris: stored in `xml` files describing vertical runs representing the symbol, 4918 files.
  - No others publicly available training set found.

Symbol Recognition – Choice

- To implement:
  1. Segmentation in a similar way than OpenOMR[6].
  2. Classification using **kNN**, trained with Audiveris' training set.

Musical Semantics

- **Input**: musical symbols and their position relative to the staff
- **Output**: MusicXML
- No specific method
- Rules and heuristics
  - For example, for a **note**:
    - **Duration**: note head type? note flag type?
    - **Pitch**: note head vertical position? accidental?

Design Choices

- Select a set of scores for tests and evaluation of the system
- Each stage takes input from a file and outputs to a file
- Preprocessing:
  1. **Binarization**: Otsu's method
  2. **Reference lengths**: RLE-based algorithm
- Staff line processing:
  1. **Identification**: horizontal projections
  2. **Removal**: Roach and Tatem
- Symbol Recognition:
  1. **Segmentation**: horizontal/vertical projections
  2. **Classification**: k-Nearest-Neighbor with Audiveris training set
- Musical Semantics: rule-based

Bibliography I

H. Bitteur.
Audiveris handbook.
http://audiveris.kenai.com/docs/manual/handbook.html.

A. D. Brink and N. E. Pendock.
Minimum cross-entropy threshold selection.
Pattern Recognition, 29(1):179–188, 1996.

J. A. Burgoyne, L. Pugin, G. Eustace, and I. Fujinaga.
A comparative survey of image binarisation algorithms for optical recognition on degraded musical sources.
In *ISMIR*, pages 509–512, 2007.

N. P. Carter and R. A. Bacon.
Automatic recognition of printed music.
In H. S. Baird, H. Bunke, and K. Yamamoto, editors, *Structured Document Image Analysis*, pages 456–65. Springer-Verlag, Berlin, 1992.

C. Dalitz, M. Droettboom, B. Pranzas, and I. Fujinaga.
A comparative study of staff removal algorithms.
IEEE Trans. Pattern Anal. Mach. Intell., 30(5):753–766, May 2008.

Bibliography II

A. Desaedeleer.
Reading Sheet Music.
Master's thesis, Imperial College London, Technology and Medicine, Department of Computing, London, 2006.

I. Fujinaga.
Staff detection and removal.
In Susan George, editor, *Visual Perception of Music Notation: On-Line and Off-Line Recognition*, pages 1–39. Idea Group Inc., 2004.

B. Gatos, I. Pratikakis, and S. J. Perantonis.
An adaptive binarization technique for low quality historical documents.
In Simone Marinai and Andreas Dengel, editors, *Document Analysis Systems*, volume 3163 of *Lecture Notes in Computer Science*, pages 102–113. Springer, 2004.

R. Göcke.
Building a system for writer identification on handwritten music scores.
In *Proceedings of the IASTED international conference on signal processing, pattern recognition, and applications*, pages 205–255, Anaheim, 2003. Acta Press.

Bibliography III

J. He, Q. D. M. Do, A. C. Downton, and J. H. Kim.
A comparison of binarization methods for historical archive documents.
In *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, ICDAR '05, pages 538–542, Washington, DC, USA, 2005. IEEE Computer Society.

P. Martin and C. Bellissant.
Low-level analysis of music drawing images.
Proceedings of the International Conference on Document Analysis and Recognition, pages 417–25, 1991.

W. Niblack.
An introduction to digital image processing.
Strandberg Publishing Company, Birkeroed, Denmark, Denmark, 1985.

N. Otsu.
A Threshold Selection Method from Gray-level Histograms.
IEEE Transactions on Systems, Man and Cybernetics, 9(1):62–66, 1979.

R. J. Randriamahefa, J. P. Cocquerez, C. Fluhr, F. Pépin, and S. Philipp.
Printed music recognition.
Proceedings of the International Conference on Document Analysis and Recognition, pages 898–901, 1993.

Bibliography IV

A. Rebelo, G. Capela, and J. S. Cardoso.
Optical recognition of music symbols: A comparative study.
International Journal on Document Analysis and Recognition, 13:19–31, 2010.

A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marcal, C. Guedes, and J. S. Cardoso.
Optical music recognition - state-of-the-art and open issues.
International Journal of Multimedia Information Retrieval, 2012.

J. W. Roach and J. E. Tatem.
Using domain knowledge in low-level visual processing to interpret handwritten music: an experiment.
Pattern Recognition, 21(1):33–44, January 1988.