Optical Music Recognition Without Staffline Removal

Tim Kang
University of California, San Diego
9500 Gilman Dr, La Jolla CA
tikang@ucsd.edu

Michael Perry
University of California, San Diego
9500 Gilman Dr, La Jolla CA
m2perry@ucsd.edu

Abstract

Optical music recognition algorithms have traditionally included removal of stafflines as a first step. In this project we have devised a way to perform OMR without staffline removal. The steps that are used involve template matching, creation of ROC curves to find the best threshold, thresholding to obtain coordinates of matched symbols, and finally pitch determination from the obtained points.

1 Introduction

While the optical character recognition of symbols from the Latin alphabet is a much-researched and widely developed field, the optical character recognition of musical symbols (sometimes referred to as OMR, optical musical recognition) is highly underdeveloped. There is an ongoing effort to adapt new and existing techniques to the optical character recognition of music symbols. This project aims to use techniques such as template matching and ROC curves to achieve an accurate OMR result without the removal of staves. We will discuss the tools that were used in this project, in addition to the process that we followed.

An overview of this projects steps proceeds as follows:

1. **Preprocessing**: Obtain image of score, invert image of score, convert inverted picture to Float image in Gamera
2. **Template Matching**: Convolve inverted float image with template.
3. **Receiver Operating Characteristic curves**: Use ROC curves to determine best threshold for convolution result and get locations of template
4. **Pitch Determination**: Use locations of templates combined with locations of staff lines to determine locations of pitches

2 Definitions

Music symbol

A music symbol is a token in the language of musical notation. It can represent a note, rest, modulation, a key signature, or any atomized element to convey musical information. In the context of this paper, a music symbol corresponds to a category of template, and each music symbol has its own ROC curve.

Stafflines

Notes and most other music symbol rest on stafflines, or simply known as staves. Staves are primarily used to determine pitch. Notes can rest above or below the staff, or when within the staff, on a line or in between lines. In the context of this paper, stafflines are necessary for the human viewer to determine pitch but are a hindrance to computers, as their visual presence are unneeded for pitch detection, and in fact makes template matching a slightly more difficult task.

3 Tools

Gamera

An open source framework, Gamera, was used throughout the course of this project. Gamera is an extensible document analysis framework for Python geared towards musical documents, and has a suite of plugins that allow for numerous filtering, image manipulation, and analysis techniques, even learning and classification [1]. It also comes packaged with a powerful, intuitive GUI, making rapid prototyping possible.

MuseScore

In addition to Gamera, the open source scorewriter MuseScore was also used, mostly to generate templates and obtain ground truth results. MuseScore is similar to commercial scorewriters like Finale or Sibelius, but with a more difficult user interface and fewer features. For the purposes of this project, however, a scorewriter of minimal features was adequate. Quick, simple score generation was desired, and MuseScore was more than adequate for this task.

4 Processes

All images shown below are generated from a toy example of a section of a score generated by MuseScore. Subsequent tests were done on other simple scores such as Twinkle, Twinkle Little Star, but for simplicity, this toy example was chosen:

4.1 Preprocessing

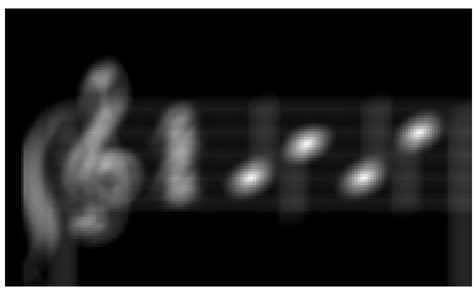
The preprocessing step is unique in the fact that the step of staff removal is not done first. In most OMR routines, staff removal is the first step taken after the binarization and rectification of the scanned image. All the classification algorithms are later performed on the image of the score with the removed staves.

The act of not removing the staves sets this project apart from other OMR efforts. Staff removal was not performed in this routine for many reasons. One reason is the imperfect nature of most staff removal algorithms. Gamera has a toolkit geared specifically for staff removal and detection called MusicStaves, which includes implementations of the most common staff removal algorithms discussed in [2], [3] and [5]. The state of the art in staff removal provides high-quality results, though not perfect. An imperfect staff removal, is often as good as no stave removal at all, as it leaves residues of the staves behind which hinders the classification algorithms.

In addition, the matching and classification algorithm used template matching, works in a way such that having staff lines or not would have a minimal effect on most of the musical symbols. This is because the symbols are either always in the same place relative to the staff line (treble clef, quarter rest) or are in a few limited possible places. For example, a quarter note can be either on the staff line or in between a staff line. Using this approach, the staff lines would be included in some of the templates and it would just require marginally more templates than template matching with staff lines removed.

4.2 Template matching

A convolution function built into Gamera was used to perform template matching. For the initial tests, we used template images generated from manually cropping scores produced by the open source scorewriter MuseScore. These templates are saved at a relatively high resolution and are rescaled to fit the size of the image that the convolution is being run on each time a convolution is run. Early tests using quarter note templates to obtain ground truth with simple scores yielded accurate results:



The template matching returns a value matrix with dimensions equal to the size of the image. The intensity of the values in the matrix corresponds to the likelihood that the object being matched exists at that particular location. The matrix of values originally has values ranging from 0 to over 2 million. A normalized matrix with a smaller range of values [0, 255] was then produced.

4.3 Receiver Operating Characteristic curves

ROC curves are used to determine the best threshold point to obtain the last possible amount of false positives and false negatives. A false negative is defined as an actual coordinate of the symbol in question that is rejected by the threshold and a false positive is defined as a coordinate that is matched due to the threshold while there should not be a match (either it matched a different symbol or simply other noise from the score). Raising the threshold will decrease the amount of false positives but increase the amount of false negatives while lowering the threshold will do the opposite.

A training set is created. Examples chosen for the training set should be similar to the examples that will be tested later in the testing set. The better the quality of examples chosen for this training set, the better the quality of the results will be. For each musical symbol, a text file of true location coordinates are parsed and compared with the convolution results. For each comparison, sensitivity and specificity are calculated using a threshold range from 0 to 255 and an ROC curve is built from this data. When a new piece of data is acquired, first the preprocessing and the template matching is run. A quick analysis of the ROC curve is done in order to find the optimal thresholding point. Then, the data is possibly added to training and the ROC curve is rebuilt.

4.4 Pitch determination

This process lays coordinates of all detected points over the stafflines. The set of detected points is represented by an array of x and y coordinate pairs. The entirety of the stafflines is represented by a two-dimensional array of elements (cluster of stafflines) x (individual staffline). The pitch of the detected points are determined by its offset from the highest possible y-coordinate of the staffline cluster, with an error of $2 \times stafflinewidth$ (which is normally 1-2 pixels thick). The reason why the offset if from the highest possible point and not the topmost staff is due to exceptionally high pitch notes that are located above the visible stafflines.

5 Conclusion

Several conclusions can be drawn from the results obtained from this experiments. It appears that classification of the music symbols without prior removal of staves is a viable solution, at least for music scores of low complexity. In fact, staffline removal is undesired if the symbol detection algorithm of choice is template matching. Furthermore, a combination of template matching and thresholding using a threshold value obtained from ROC curves is a viable way to obtain results.

As time was limited, there was only enough time to test and generate ROC curves for a small set of scores. In the future, testing with a larger data set of varying complexity and resolution would be useful in helping us generate more comprehensive statistics.

References

[1] K. MacMillan, M. Droettboom, I. Fujinaga (2001): Gamera: A Python-based toolkit for structured document recognition. International Python Conference, pp. 25-40, 2001

[2] Dalitz, C., Droettboom, M., Czerwinski, B., Fujigana, I.: Staff removal toolkit for gamera (20052007). http://music-staves. sourceforge.net

[3] Dalitz, C., Droettboom, M., Czerwinski, B., Fujigana, I.: A com- parative study of staff removal algorithms. IEEE Trans. Pattern Anal. Mach. Intell. 30, 753766 (2008)

[4] Fujinaga,I.:Staff detection and removal.In:George,S.(ed.)Visual Perception of Music Notation: On-Line and Off-Line Recogni- tion, pp. 139. Idea Group Inc, Hershey (2004)

[5] Capela, A., Rebelo, A., Cardoso, J.S., Guedes, C.: Staff line detection and removal with stable paths. In: Proceedings of the International Conference on Signal Processing and Multimedia Applications (SIGMAP 2008), pp. 263270 (2008). http://www.inescporto.pt/jsc/publications/ conferences/ 2008ACapelaSIGMAP.pdf