Music Score Recognition

Piotr Wilczyński

# Introduction

The purpose of the project is to develop a piece of software that converts a music score into an audio file. Sheet music is first scanned, then passed to the application and finally a MIDI file that plays the given music is generated.

# Methodology

To begin with, the scanned image is binarized so that it is only composed of black and white pixels. There are several ways to choose a threshold for this binarization, for example by taking the value of the pixel that is the most prevalent across the image.

Since scanned images are never perfectly straight, the image must be rotated slightly. The stave lines are a good indicator of the level of skewness. First, a histogram is obtained that shows the number of pixels in each row. Then, Stochastic Gradient Decent is used to find the angle at which the document was scanned. The histogram of the straightened image should clearly show the five stave lines by sudden increases and decreases in the numbers of pixels.

The next step is symbol recognition. The stave will potentially be removed, although some experiment will be done first to see which approach makes the detection easier. A set of music symbols will be provided for the application to train on. Connected Component Analysis will be used to extract separate characters from the image and then those characters will be compared to the training set. The note stems will also possibly be removed for easier pitch identification. An attempt will also be made to recognise certain accents and embellishment to make the audio file more faithful to the sheet music.

Having generated a skeleton of music notation, the application will pass it to an external tool generating MIDI files. The approach of composing an audio file directly might also be taken.

# Software

The application will be written in Java and throughout its execution, a user interface will be displayed showing the progress of the recognition. The user will also be able to select a scanned image from the disc through this interface.

# Timetable

The first working prototype of the project should be completed by the end of the Autumn Term. Then, the possibilities of improvement will be explored. Specifically, the rest of the project should focus on developing one or more components to provide higher accuracy and the ability to recognize more complex sheet music.

Autumn Term:  
Week 1 – background reading  
Week 2 – displaying the histogram on the interface  
Week 3 – straightening the image  
Week 4 – reading on Connected Component Analysis  
Week 5 – organising a training set of symbols  
Week 6 – first attempt to detect separate symbols  
Week 7 – debugging the detection algorithm  
Week 8 – duration identification  
Week 9 – stem removal and pitch identification  
Week 10 – generating a simple audio file  
Week 11 – additional testing, documentation and reading

Spring Term:  
Week 1 – exploring options for improvement, reading  
Week 2 – attempt to recognise certain accents  
Week 3 – debugging the accent detection algorithm  
Week 4 – chord recognition  
Week 5 – debugging the chord recognition algorithm  
Week 6 – testing and improving the chord recognition algorithm  
Week 7 – binarization  
Week 8 – looking into possibilities of better efficiency  
Week 9 – documentation and polishing up the code  
Week 10 – documentation and testing  
Week 11 – presentation

# Evaluation

The end result of the project will be tested on music scores of different complexity. The criteria used for evaluation will be the accuracy corresponding to certain complexity as well as efficiency in terms of runtime and the usability of the user interface.

# Previous research

Approaches that have already been researched will be taken into consideration, however existing papers indicate that the state-of-the-art accuracy is still far from perfect and there is a lot of scope for improvement. There are several directions for advancement: the correctness of the detected music notation, the complexity of the sheet music and the performance of the recogniser. There is also the concept of processing handwritten music, either on a printed stave or a stave also written by hand.

# Literature

These are some of the papers that the project will consult:

Homenda, W. (2005) *Optical Music Recognition: The Case Study of Pattern Recognition*. Warsaw: Faculty of Mathematics and Information Sci, Warsaw University of Technology.

Miyao, H. and Maruyama, M. (2006) *An online handwritten music symbol recognition system*. Springer-Verlag.

Rebelo A., Capela G. and Cardoso, J.S. (2009) *Optical recognition of music symbols*. Springer-Verlag.

Tardón, L.J., Sammartino S., Barbancho I., Gómez, V. and Oliver A. (2009) *Optical Music Recognition for Scores Written in White Mensural Notation*. Malaga, Departamento de Ingeniería de Comunicaciones, E.T.S. Ingeniería de Telecomunicación, Universidad de Málaga.

# Git

The project can be monitored in the following repository:

<https://git-teaching.cs.bham.ac.uk/mod-ug-proj-2017/pxw328.git>