ECE 532 - Project

Optical Music Recognition

Tianyang Chen

23369298

tianyangchen@email.arizona.edu

December 2016

1. Introduction

Optical music recognition is the application of optical character recognition to interpret sheet music or printed scores into editable or playable form. In this project, I developed a software that uses the scanned image of music staff as input, finally output a machine readable numbered musical notation and play the music by MATLAB.

The whole project includes the following files:

source code: omr.m
comp_label.m
playmusic.m
staff image: ttls.jpg (Twinkle Twinkle Little Star)
af.jpg (Adeste Flauto)
jb.jpg (Jingle Bells)
numbered musical notation: ttls.csv
af.csv
jb.csv

To run this program, simply type the following command line in MATLAB's command window:
omr(Input_image, Output_numbered_musical_notation);

For Example:
omr('ttls.png', 'ttls.xls');

The numbered musical notation was written in the following format:

1	2	0.25
1	2	0.25
5	2	0.25
5	2	0.25
6	2	0.25
6	2	0.25
5	2	0.5
4	2	0.25
4	2	0.25

Fig.1 example of numbered musical notation

The first column denotes the name of the note,
1-do 2-re 3-mi 4-fa 5-so 6-la 7-ti
The second column denotes musical scale,
0-bass clef 1-alto clef 2-treble clef
The third column denotes the type of the note,
1-whole note 1/2-half note 1/4-quarter note

2. Algorithm

This Algorithm can be divided into four steps. Firstly, locate and remove the staff line; secondly, music notes segmentation; thirdly, recognize the specific music note; finally, play the sound with corresponding frequency and write them to the file.

2.1 Locate and remove the staff line

To locate and remove the staff line, I used the method Horizontal Projection, which was put forward by Scott Sheridan and Susan E. George. The algorithm is easy, after binaryzation of the image, compute the number of black pixels in each row, obviously the staff line will be at the peak position.

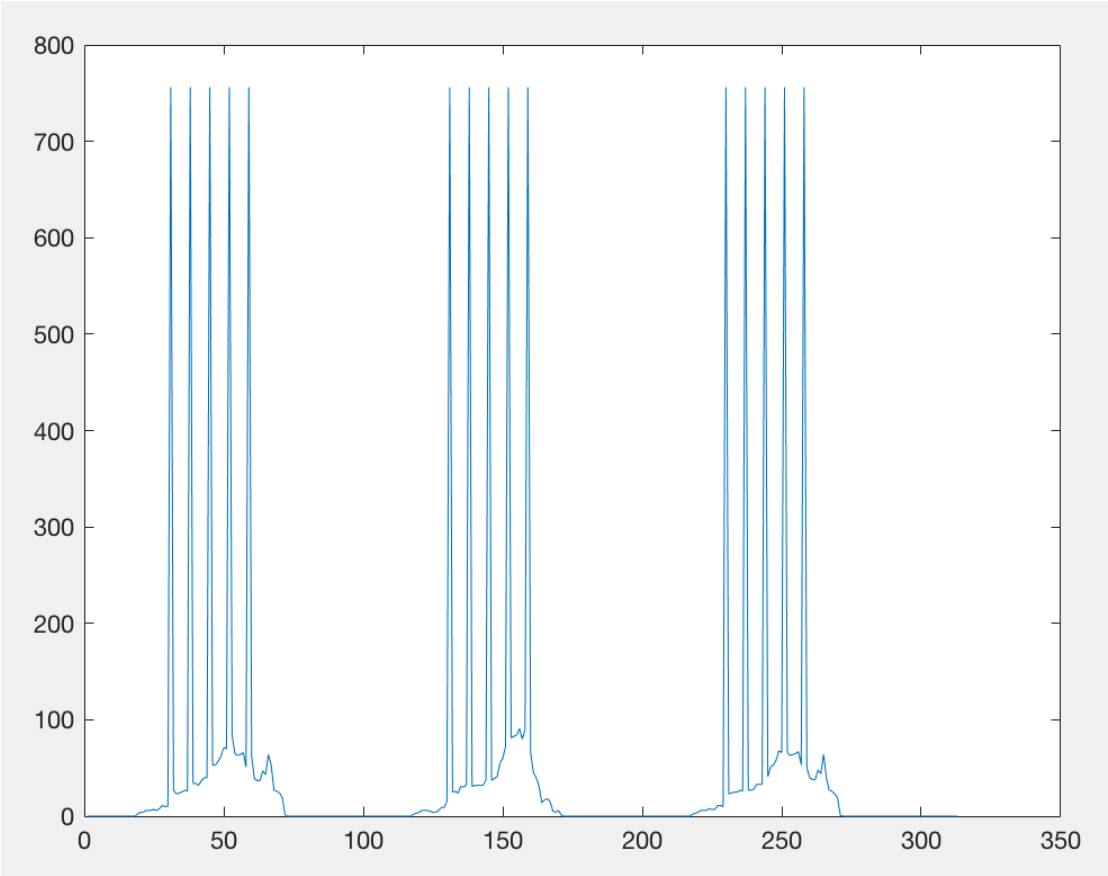

Fig.2 the horizontal projection graph of ttls.jpg

Here, I set the threshold to 0.5 times the maximum value and locate the specific row of each staff line.

After removing the staff line, the image of the notes are fractured, then I implemented an algorithm to fix these cracks. The method was easy, consider the pixels in each staff line, for each pixel, if the pixels above and below are both black, the middle pixel should also be black. In addition, during this process, I can also get the mean of the staff line

interval, which will be used to estimate the radius of the note head later. When I was working on the project, I also tried an image whose width of staff line is more than one pixel, this fix algorithm can't be implemented on this image. Therefore, one restriction of this program is that it can not be used on the image with more than one pixel width staff line.

2.2 Music notes segmentation

The only thing I'm interested in is the notes, the other symbols such as clefs, time signatures and bar lines should also be removed. First, I used the algorithm discussed in class to do the connected component labeling; second, I computed the bounding box of each connect component, By analysis the relationship between the height and width of each bounding box, I can judge which components should be removed. For instance, To remove the clefs, simply remove the box with height bigger than 4 times of the staff line interval; to remove the time signatures, just remove the box with about 2 times of the staff line interval as the height; to remove the bar lines, simply remove the box with very small width.

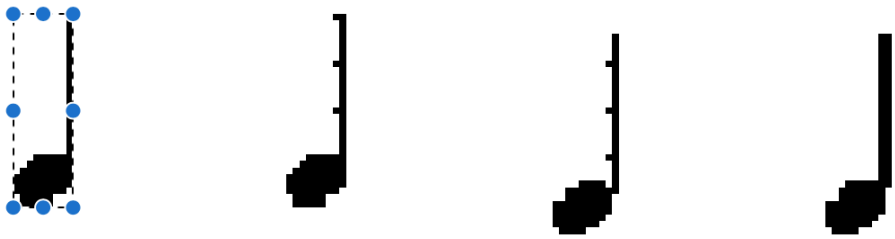

Fig.3 Bounding box

2.3 Music notes recognition

This process was finished in two steps, firstly, recognize the type of the note, secondly, recognize the position of the note. If the height of the bounding box is two times bigger than its width, which means the note has a tail, it could be either a half note or quarter note. If the height of the bounding box is similar to the staff line interval, it could only be a whole note. To distinguish half note from quarter note, I used the following algorithm,

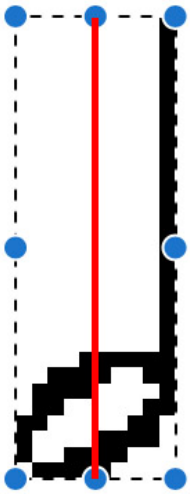

Fig.4 Distinguish half note from quarter note

I extracted a column of pixels at the middle of the bounding box exactly. Then computed the number of black pixels and the distance between the first black pixel and the last black pixel. If the distance is bigger, which means there must be some white pixels between the black pixels, and the note must be a half note.

To locate the note, subtract the height of bounding box by half the staff line interval, such that I can locate the note head in the bound box. Then I computed the distance between this head and the first staff line to determine the name of the note.

3. Performance Evaluation

There is already some analogous software in the market such as smartcore, I downloaded the evaluation version of this software to recognize the same music image as the project. And use accuracy to evaluate the performance.

$$Accuracy = \frac{correctly\ recognized\ notes}{total\ number\ of\ notes}$$

The results are shown below,

	Project	Smartscore
Twinkle Twinkle Little Star	100%	Failed
Adeste Flauto	90%	Failed
Jingle Bells	46%	64%

Table.1 Performance Evalution

I have no idea why Smartscore can't recognize Twinkle Twinkle Little Star and Adeste Flauto. It just reported 'failed to recognize'.

4. Discussion and Improvement

Although it looks that the project works better than Smartscore, the project also have a lot of restrictions.

(1) The project can only recognize whole note, half note and quarter note. If the music sheet is complex, it won't work.

(2) The project require high quality image. The width of the staff line have to be one pixel. And as you can see, the quality of 'jb.jpg' is not very high, as a result, after removing and fixing the staff line, there might be some cracks between the note head and note tail, such that one note are recognized as two components. Later, the tail will be removed by mistake.

(3) Not all the notes are tail on top and head on bottom. The project will incorrectly recognize the notes with heads on top.

To improve this project, in future work, I will try the following method,

(1) In order to locate the note head, I should first try Hough Transform to detect circles (note head) in each bounding box. Such that I could locate more complex notes, for example, sixteenth note.

(2) Try to combine template matching algorithm to improve accuracy.

(3) Find the algorithm to recognize other music symbols, such as rest symbol, key signature and appoggiatura.