Image Processing

Partial Report

QUEIROZ, Breno Cunha
11218991

NÚÑEZ, Henrique Hiram Libutti
11275300

MOREIRA, Maria Eduarda Kawakami
11218751

July 2020

1 Introduction

This project aims to extract notes from a picture of a music sheet, and be capable of playing and saving the music. To solve this problem it is necessary to create a robust algorithm that is able to differentiate objects that are expected in the image from foreign objects, such as texts and strange images. It is also necessary to identify in which positions the lines are, since they vary from image to image and are extremely important for determining the musical note.

2 Input Images

We got the input images in the website https://musescore.com/ Here are two examples of music score:

Canon in D[1]

La Danse Macabre[2]

3 Image processing

3.1 Binarization

For binarizing the image we used Otsu's threshold, known as the most widely used method. It computes the optimal threshold by separating the intensities in two classes that minimizes intra-class variance and maximizes variance among classes.

3.2 Segmentation

After binarizing the image the only step missing to segment the image is to distinguish different notes, which can be done attributing a seed to each set of black pixels. To do this we are searching for black pixels on the image and attributing a seed to the set of pixels in which it is contained. Then the procedure for region segmentation by conquering is called.

3.2.1 Segmentation strategy

This procedure uses a conquering strategy, by evaluating the distance between the mean intensity value of region until now and the value of the pixels contained in a 4-neighbourhood of the current location. In case the calculated distance satisfies a given threshold, the procedure is then called recursively in one of the four directions mentioned earlier. With a then computed region, it is possible to extract useful information from the original music piece. We now must do processing for achieving two aspects: the tone and the timing.

3.3 Note recognition

To recognize the notes segmented we are comparing the set of pixels with each seed using a kernel of note recognition. If the set of pixels matches with a note, we calculate the x and y of the note's gravity center, the round mark that defines its tone.

3.3.1 Symbol classification

Since we have not extracted the actual notes yet, it is hard to define which methods will best serve our purposes. Probably some supervised machine learning technique, such as Support Vector Machine or Random Forests is the best solution.

3.3.2 Tone

To get the tone of each note, after getting the coordinates of the note's gravity center, we compare them with the coordinates of each line previously detected, and then cross this information with clef detection to find the actual tone.

3.4 Playing the piece

After getting all the tones and delay between each tone, we can start playing the music. To do this we are using mingus, which is a python library to play tones using songfonts. With this library we can download songfonts (sound from piano, guitar, violin, among others) from the internet and play the same music with different instruments.

References

[1] *Canon in D.*
`https://musescore.com/user/88585/scores/105013`

[2] *La Danse Macabre.*
`https://musescore.com/user/54180/scores/90702`