AUTOMATIC MUSIC SCORE RECOGNITION

Chao-Wei Su (*蘇兆為*) *Chiou-Shann Fuh* (*傅楸善*)

Dept. of Computer Science and Information Engineering,
Taiwan University, Taiwan

ABSTRACT

Music industry becomes more and more popular. There are many music books we can buy in the store or find on the internet. If we can scan these music books and convert it to music file, we can do some further processes easily. This paper proposed some useful methods to detect the position of music score and music note, and used these information to create the music file.

Keywords *Music Score, Recognition*

1. INTRODUCTION

Music plays an important role in multimedia industry, and there are so many subjects in music analysis and retrieval, like music composition, melody detection, etc. In recent years, using machine learning method to improve MIR performance became more and more popular. However, there is a question: how do we collect the data? Before we can appropriately extract the information in music signal, other methods for collecting data are needed. One of these method is converting the music books into digital file. This paper used projection to detect music score lines and note stems, and used connected component to identify the note position. Finally, we create the music file to show the recognition result.

2. METHOD

2.1. RGB to binary

First, the music score is only black and white image. Thus, if our input images contain RGB channels, convert it to gray level image, then use threshold to binarize.

$$\mathrm{f(r,g,b)} = \begin{cases} 0, if \dfrac{r+g+b}{3} < 0.5 \\ 1, otherwise \end{cases} \quad (1)$$

2.2. Score line detection

Score lines play an important role in locating the music notes, so we need to find it out. Hough transform is a powerful way to do straight line detection, and it might be useful in this work. But if our input images are good enough, the score line should be horizontal, and we do not need such powerful method. We can simply use orthographic projection to detect the score lines. First, project the image to *y*-axis. We can do this simply by counting the number of pixels every row, then draw the histogram of result.

$$\mathrm{h(Image,y)} = \sum_{n=1}^{width} Image(n,y) \quad (2)$$

It should be like this:

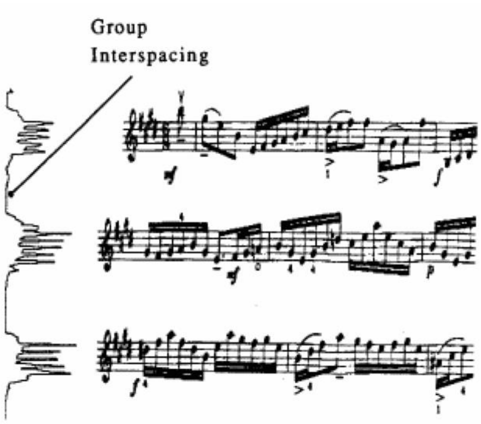

Fig. 1: Result of orthographic projection.

We can easily find that the peaks of histogram occur in the score lines position, so theoretically we can just find the five peaks per group, then we are done. The problem is, assume we can find the group position without confusing with other group, what is the threshold that is used to distinguish whether it is peak? If the threshold is not chosen properly, we might detect more than or less than five lines in the group. The second problem is that because the score lines are not always one pixel width, the top 5 peaks are not always the different lines. To fix these problems, we need to exclude the peaks that are too close to the others or too weak, then pick the top 5 peaks. If we cannot find out enough lines, we need to use the lines that have been found to calculate the space between two lines, and recover the missing lines.

Fig. 2: Result of line detection.

2.3. Score line removal

After we detect the score lines, we almost can begin to detect the music notes. But before that, we need to remove the score lines from the image first. We can assume the line width is about 2 or 3 pixels, and when we remove the lines, we should not destroy the music notes so that the note detection can be easier.

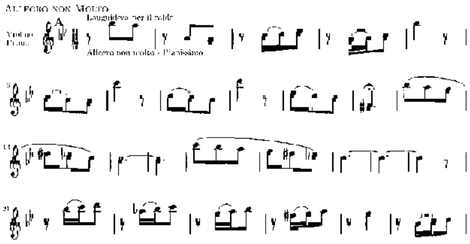

Fig. 3: Result of score line removal.

2.4. Note stem detection

Now we can start to detect the notes position, and the first step is to locate the "note stem", which is the vertical line next to the note. Using the similar tricks, we projected every group of score lines to *x*-axis, and found those value are high enough. We also need to remove some lines to prevent they are too close to each other. Then we found the vertical lines that are "continuous enough", means they do not have to be continuous at every point, as long as the gap between two segments is small enough.

Fig. 4: Result of stem detection.

2.5. Note connection detection

Now we have the note stem, and we should be ready to detect the note position. The idea is simple. We check the four corners of every stems, and find if there are some large groups of black pixels. Unfortunately, sometimes there is a connection between two notes, and it cause the error of note detection.

Fig. 5: Note and its stem.

So we need to identify which one is the real note. We can find the connected component in the region encompassed by two stems. If there are two connected components, no connection between these two notes. If there is only one connected component, it means these two notes are connected.

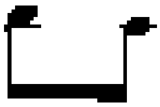

Fig. 6: One connected component.

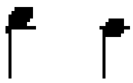

Fig. 7: Two connected components.

2.6. Note detection

As I mentioned before, we check if there are groups of black pixels on the four corner of note stem. But this time, we exclude the one that is identified as the connection between two notes.

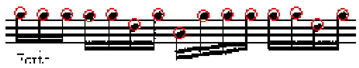

Fig. 8: Result of note detection.

2.7. Create music file

After we detected the score line and notes, we can use the information to create the music file. The bottom line of the score has the semitone 64. We can calculate the distance between the note position and this line, and divide it by the half of distance between two score line. Then we get the pitch difference between this note and pitch 64. Add this difference to semitone 64, then we get the final semitone of this note.

3. RESULT

We tested this system on several music sheets, and the following are some of the results.

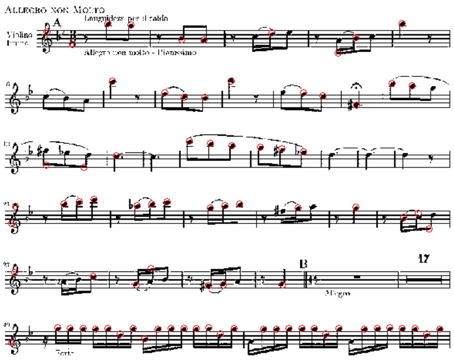

Fig. 9: Song "Estate / Summer".

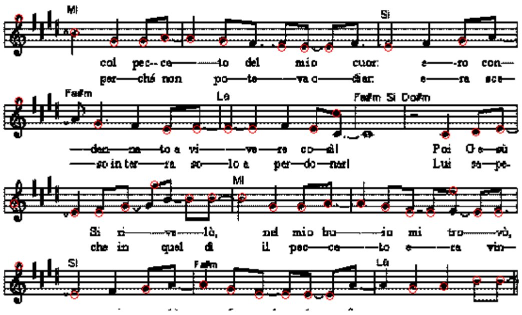

Fig. 10: Song "Lui mori!".

We can find that there are still some errors occur – including confusing stem with note, missing detection, and false note position. Another problem of this method is hard to find the hollow notes. We can only find the solid notes now, and I made this as part of our future works.

4. CONCLUSION

In this work, we proposed a simple method to recognize the note position in the music books. We can use this information to create a music file, or as the training data of MIR. There are some of defects need to be improved, and we hope this system can become more precise in the next step.

5. REFERENCES

[1] Z.W. Cai, and Y.W. Jiang, "Automatic Recognition of Printed Music Score," 2004.