Music Score Reader
Based on Morphological Image Processing

Dingyi Li, Bing Han and Jia Ji
Department of Electrical Engineering
Stanford University
Stanford, US

Abstract - A music score reader which can "read" scanned music score and play the music out was developed in this project using Morphological Image Processing techniques. Staff lines were detected by adding up the pixel values along every row after tilting the image to horizontal orientation; note pitch were got by detecting note heads using sequence of erosion, dilation, open and close operations; note duration were calculated by detecting hook, beam and augmentation dot around the note head.

Keywords - music score; staff line; note; open; dilation

I. Introduction

Music Score Reader is a system that can "read" scanned music score and play it out. A Music Score Reader on computer/mobile platform is a useful tool for many people: a soloist could have the computer play an accompaniment for rehearsal; a music editor could make corrections to an old edition using a music notation program [1]. Other than those professional people in music, there is a large population of music lovers who have strong interest in singing/playing but can not read music score which is the most common way of recording music. In this project we are targeted to help those people who don't know how to read music score to learn music easily with the tool we built.

In this paper, we presented a music score reading system, which use image processing techniques to extract music information from the picture of a music score, and play out the music. To do this, we first binarized the image so that we can do Morphological Image Processing on the image in the following steps. Secondly, we tilted the image such that the staff lines are horizontal, and identified the locations of the staff lines. Following that, we detected objects of the score including the note head, hook, beam and augmentation dot. With those information, we can then calculate the pitch and duration of the notes. Finally, we translated those recorded information into playable audio data so that the music can be played out.

This paper is organized as follows, Section II shows the detailed algorithms of each step; Section III presents the experiments results and made analysis on them; Section IV lists future work that are expected to do in the future; Section V makes a conclusion of the paper.

II. Algorithms

A. Binarization

The input image in our data set is JPEG format. The color information is not important in the following part, so we firstly converted it to a gray level image. Then we reduced the gray level image to a binary image to differentiate the foreground and background. We use the popular method, Otsu's method, to perform the binarization. The algorithm assumes that the image to be thresholded contains two classes of pixels or bi-modal histogram (e.g. foreground and background) then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal [2]. With a proper threshold, we obtained a binary image containing all the useful information, such as notes and staff lines, as its foreground.

B. Untilt Image

Ideally, the input image downloaded from the internet has perfectly horizontal staff (a set of five horizontal lines and four spaces that each represents a different musical pitch) [3]. However in practical, people may take a picture of a piece of music score in which the staff has a small angle from the horizontal line. Thus, untilting the image is the first processing step for the binary image.

Firstly, we filled all the holes in the binary image. This made staffs become solid rectangles. Then we calculated the orientation (an angle value) of each rectangle and took an average of them as the orientation of the whole image. Based on this, we rotated the whole image with an angle of the orientation value and obtain an untilted image.

C. Staff Detection

To finally play out music, we need the information of pitch and duration of each note. For pitch calculation, we need to know the relative position of the head of a note compared with the location of its corresponding staff. Therefore, the location of each staff in the music score is the next step after we have an untilted image. Staff detection is indispensable since it also helps remove the lines in staffs in the following step.

To detect staffs, we sum up all the binary values for each row and get a histogram that shows how many non-zero values in each row of the binary image. Obviously, the rows containing the lines in staffs correspond to remarkably high peaks. Detecting the position of peaks in the histogram told us the location (row index) of the lines in staffs. With the location of five lines in each staff, we also obtain the space width in the staff by dividing the location difference of the upmost and downmost line by 4. In addition, by detecting the width of peaks in the histogram we stored the maximum width of a line in a staff. The values of width and space are useful to construct a structuring element for staff removal and to calculate the pitch of a note, respectively.

D. Staff Removal

As mentioned in previous part, we removed lines in staffs in the music score using the information of location and width of the lines. This step is the foundation of detecting head, hook, beam, and augmentation dot.

Firstly we construct a structuring element which is a vertical bar with 1 pixel horizontally and (width+1) pixels vertically, where the "width" is the maximum width of a line in a staff. Then we perform "open" operation on the whole image with the structuring element to remove all the lines in staffs.

E. Note Head Detection

Detecting note head is the key step of the algorithm. Because if we know where the note heads are, we can easily compare their locations and the staff line locations to get the pitches. We can also look into the areas around note heads to see if there are note hooks, beams or augmentation dots, which are essential for the note duration calculation.

To detect note heads, we perform a sequence of open operations. First we open the image with a horizontal structuring element, and as a result, all thin vertical lines in the image will be removed. Next we open the image with a disk structuring element, so that small irregular shapes are removed from the image. After this step most non-head region will be removed, but there are still some regions that have similar shapes to note head remains. We can further eliminate those regions by checking the area, axis ratio and solidity.

TABLE 1. Frequencies for Different Keys

Key Name	Frequency (Hz)	Key Name	Frequency (Hz)
A3	220	D5	587
B3	247	E5	659
C4	262	F5	698
D4	294	G5	784
E4	330	A5	880
F4	349	B5	988
G4	392	C6	1047
A4	440	D6	1175
B4	494	E6	1319
C5	523	F6	1397

Figure 1. Each of the staffs has seven notes and one rest

F. Hook/Beam/Augmentation Dot/Empty Head Detection

To detect note hook/beam, we need to look into the regions around the note heads. The first step is to identify the right region of interest for each head, depending on whether the note stick points upwards or downwards. Next we perform an open operation with horizontal structuring element to remove the note stick. If the note doesn't contain a hook/beam, nothing should be left in the region of interest after this step. Otherwise, we can find out the number of hooks/beams by examining the thickness of the hook/beam.

To detect the augmentation dot, we look at the region after the heads. Nothing should be there if there's not augmentation dot.

An empty head has twice the duration as a solid head, so it's very important that we detect if a note head is empty or not. Fortunately we can use Euler number to detect empty heads. By definition Euler number is number of objects minus the total number of holes in a binary image. So if a head is solid the Euler number is 1, and if a head is empty the Euler number is 0.

G. Pitch/Duration Calculation

This is the final step before playing out the music finally. To calculate the pitch of a note, we compared the relative position of the head of the note compared with the location of its corresponding staff. Figure 1 shows the relative position for seven notes, Doe, Ray, Mi, Fa, So, La and Tee.

As for duration calculation, the standard formula is below: For an empty head without stick, the duration is:

$$D = 16 \tag{1}$$

Otherwise,

$$D = 8 \times 0.5^{1-Empty} \times 0.5^{Num\ of\ Hooks/Beams} \times 1.5^{Aug\ Dot} \tag{2}$$

where "Empty" is a Boolean value that indicates if a head's empty.

After this step we know both the pitch and duration for each note. We are ready to play out the music.

H. Play the music

Each pitch key is associated with a unique frequency. Table 1 shows the detailed association relationship [4]. If we know the pitch and duration, we can generate a sinusoidal wave that

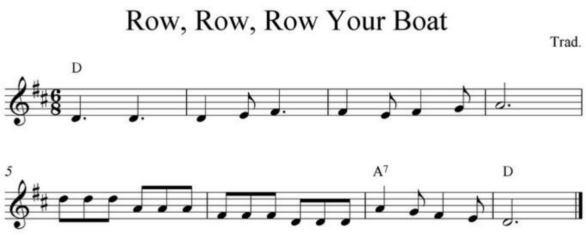

Figure 2. Sample Music Score

represents the note. In order to play the music, we concatenate the sinusoidal waves for each note together, and output the sound of the resulted wave.

III. EXPERIMEMNTS

Figure 2 shows a sheet of sample music score to which we applied our algorithms. Figure 3 shows the result of binarization. Figure 4 shows the result of staff detection. Figure 5 shows the result of note head detection. Our algorithms perform well on this sheet of music. All note head were detected and their pitches and durations were calculated correctly.

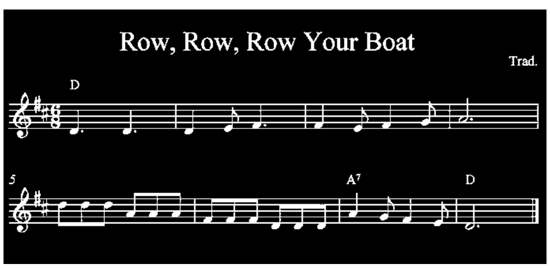

Figure 3. Result of Binarization

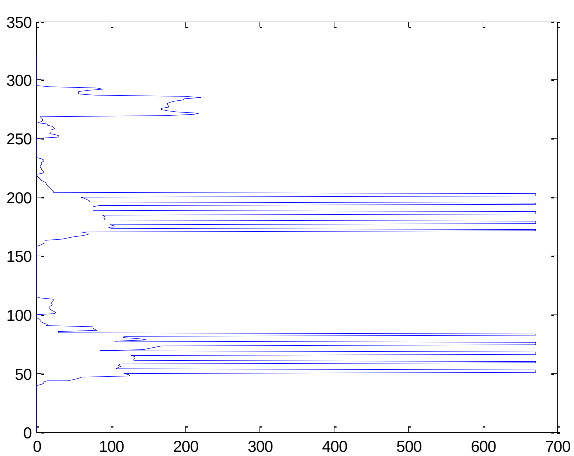

Figure 4. Result of Staff Detection

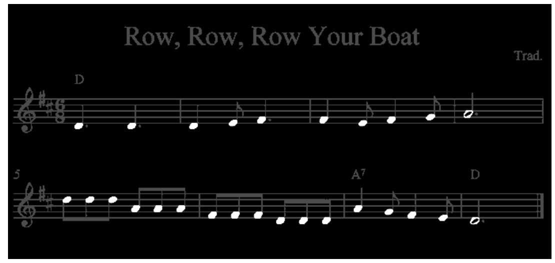

Figure 5. Result of Note Head Detection

We also performed our algorithm on some more complicated music scores, like the one in figure 6(a). We discovered that when the score is drawn standardly, which means the notes are well separated from each other and marked clearly, our score reader can read in the music with high accuracy. But when the notes on score are not well separated or not as clear, as the one in figure 6(b), the detection accuracy decrease very fast.

We tested our algorithm on 7 nice music score like the one in figure 6(a). The note head detection rate, which is defined as number of detected head divided by total number of note heads, is 98.11%. We also tested our algorithm on non-standard music scores like the one in figure 6(b), and the note detection rate is only 5.8%. We think that the reason for this is that when we develop our algorithms, we made assumptions on what a music score would look like. Those assumptions are reasonable for many music scores since most music scores are generated with some standards. But apparently, the music score in figure 6(b) is not what one would expect in most cases. We would like to fix this problem in the future.

IV. FUTURE WORK

As a further extension of this project, we would like to explore the following areas: 1) Improve the performance for non-standard scores; 2) Recognize other elements in the music score like clef, pause note, sharp and flat; 3) Support chord which is set of two or more notes that is played simultaneously; 4) Move to more platforms such as mobile devices or web.

Figure 6. (a) A clean standard music score (on the left)

(b) A non-standard music score (on the right)

V. CONCLUSION

In this project, we developed a music score reader which can "read" picture of music score, process it and play the music out. We binarized and untilted the image; identified the location of staff lines, used Morphological Image Processing techniques to recognize the pitch and duration of music notes; played music out with sinusoidal waves that have corresponding frequencies. For clean standard score, we achieved an accuracy of 98.11% for note head detection, whereas for nonstandard music score, the accuracy is only 5.8%.

REFERENCES

[1] Bainbridge, D., Bell, T. The Challenge of Optical Music Recognition. In Computers and the Humanities, 2001, Vol. 35, Number 2, pp. 95-121

[2] Nobuyuki Otsu. "A threshold selection method from gray-level histograms". IEEE Trans. Sys., Man., Cyber. 9 (1): 62–66.

[3] "Staff (music)", Wikipedia, http://en.wikipedia.org/wiki/Staff_(music)

[4] "Piano key frequencies", Wikipedia, http://en.wikipedia.org/wiki/Piano_key_frequencies

APPENDIX

Individual work break down:

Dingyi Li	Staff line detection, node head/beam detection, augmentation dot detection, empty head detection, testing and debugging
Bing Han	Binarization, untilt images, play music, android phone programming, testing and debugging
Jia Ji	android phone programming, node head/beam detection, testing and debugging