Automatic Music Score Recognition System Using Digital Image Processing

Yuan-Hsiang Chang, Zhong-Xian Peng, Li-Der Jeng.

Abstract**—**Music has always been an integral part of human's daily lives. But, for the most people, reading musical score and turning it into melody is not easy. This study aims to develop an *Automatic music score recognition system using digital image processing*, which can be used to read and analyze musical score images automatically. The technical approaches included: (1) staff region segmentation; (2) image preprocessing; (3) note recognition; and (4) accidental and rest recognition. Digital image processing techniques (e.g., horizontal /vertical projections, connected component labeling, morphological processing, template matching, etc.) were applied according to musical notes, accidents, and rests in staff notations. Preliminary results showed that our system could achieve detection and recognition rates of 96.3% and 91.7%, respectively. In conclusion, we presented an effective automated musical score recognition system that could be integrated in a system with a media player to play music/songs given input images of musical score. Ultimately, this system could also be incorporated in applications for mobile devices as a learning tool, such that a music player could learn to play music/songs.

Keywords**—**Connected component labeling, image processing, morphological processing, optical musical recognition.

I. Introduction

WITH the advance of image processing and computer vision techniques in recent years, the techniques have been integrated in human's daily lives. Typical image processing and computer vision applications include document processing, smartphone applications, video surveillance systems, multimedia systems, and/or video games, etc.

In image processing techniques, the Optical Character Recognition (OCR) is an important technique that has been widely used in handwriting inputs, license plate recognition, and augmented reality applications. The objective of the OCR technique is to allow the computer to analyze the text images, and then convert to texts (typically the ASCII codes) which computer can handle. For example, [1] proposed a method to calculate the appropriate threshold for converting gray-level images to binary images automatically. Casey and Lecolinet [2] proposed a character segmentation system based on connected component analysis and feature extraction, which were used to segment and recognize each character from document images. Liu et al. [3] proposed a handwritten character strings recognition system for address reading, in which characters were segmented using the connected component analysis. Each character was then recognized using a beam search algorithm and a character classifier.

In addition to the OCR technique, pattern recognition techniques are also drawing attention of many researchers. Patterns (or symbols) are commonly seen in documents and/or other scenarios in which text information is not used for the representation (such as music notes, traffic signs, gestures, etc.). However, recognition of such patterns (or symbols) may require expertise to achieve effective representation and/or communication (e.g., music notes in a music score, etc.).

Musical score is a form to record music by symbols which may include the pitch and tempo information about the music and/or songs. With the development of pattern recognition techniques, musical score recognition has also become a research topic lately. For example, [4] proposed a method to detect and remove staff lines using derivation and connected component analysis. Chen et al. [5] proposed a conventional architecture of Optical Music Recognition (OMR) using the staff-lines detection as the key stage. They explored two methods, namely the Hough transform and Mathematical Morphology, for detecting all staff-lines of an image. Dutta et al. [6] proposed a different method to detect and remove staff lines from musical documents. The methodology considered a staff line segment as a horizontal linkage of vertical black runs with uniform height. They also used the neighboring properties of a staff line segment to validate it as a true segment. Yoo et al. [7] proposed a system to recognize musical scores in low resolution images captured by the digital camera of a mobile phone. They presented a mask based approach to cope with incomplete information in the low resolution images. Toyama et al. [8] proposed a score recognition method which could be applicable to the complex music scores. Symbol candidates were detected by template matching, and then selected by considering the relative positions and mutual connections. Rossant and Bloch [9] proposed an optical music recognition system based on a fuzzy modeling of symbol classes and music writing rules. The objective was to disambiguate the recognition hypotheses output by the individual symbol analysis, followed by the fuzzy modeling to account for imprecision in symbol detection. Parker [10] implemented a complete optical music recognition system, called Lemon. Their system included the techniques, i.e., staff line detection, text segmentation, line detection, symbol recognition, note head recognition, and semantic interpretation.

Yuan-Hsiang Chang, Ph.D. is with the Information and Computer Engineering Department, Chung-Yuan Christian University, Chung Li, Taiwan, R.O.C. (phone: 886-3-265-4713; fax: 886-3-265-4799; e-mail: author@ boulder.nist.gov).

Zhong-Xian Peng, is a graduate student with the. Information and Computer Engineering Department, Chung-Yuan Christian University, Chung Li, Tawian, R.O.C

Li-Der Jeng is with the Electronic Engineering Department, Chung-Yuan Christian University, Chung Li, Taiwan

II. Method

In this study, we present an “Automatic Music Score Recognition System Using Digital Image Processing”, which was aimed to automatically recognize musical scores.

Several system hypotheses can be described as follows:

- The musical score is a printed document (i.e., black musical notes or symbols in white background).
- The musical score is a scanned document in an upright position, therefore no perspective distortions are observed.
- The image is with sufficient resolution and in good quality.

Fig. 1 shows the flow chart of our system. The processes include: *Image Analysis and Segmentation*, *Image Preprocessing*, *Note recognition*, and *Accidental and Rest Recognition*.

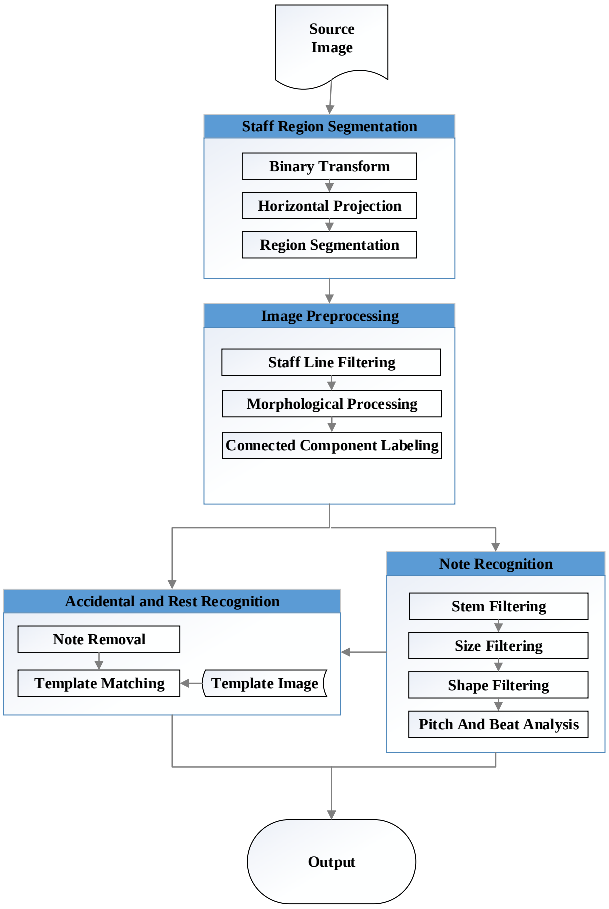


Fig. 1 System flow chart of the *Automatic Music Score Recognition System Using Digital Image Processing*

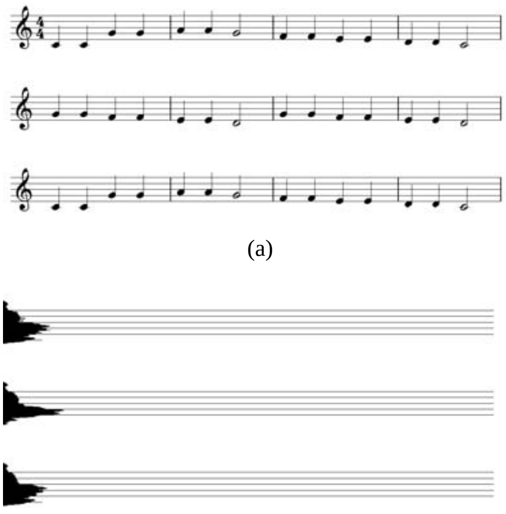
A. Staff Region Segmentation

A page of musical score consist of number of row stave, with a top-down order when playing. A staff consists of five staff lines, while notes are recorded on staff lines with respect to the height of each line to determine its pitch. Therefore, the first step of our system was to segment sub-regions for each staff. The processes included: *Binary Transform*, *Horizontal Projection*, and *Region Segmentation*.

Binary Transform was simply used to convert the input image to a binary image. In this study, the Otsu’s algorithm was used to find the optimal threshold T that minimizes the within-class variances. As a result, given an input image I and the threshold T, a binary image I_B can be acquired using:

$$I_B(x,y)=\begin{cases}255, & if \quad I(x,y)>T\\ 0, & otherwise\end{cases} \qquad (1)$$

Horizontal Projection: The image projection is a method for projecting source data to selected area to reduce the dimension of the source data, such that the data could be easily processed. Image projections can be implemented in either the horizontal or vertical directions, resulted in horizontal or vertical projections. Here, the *horizontal projection* was applied in our system to acquire the horizontally projection histogram as shown in Fig. 2. By projecting the musical score in Fig. 2 (a) horizontally, total number of pixels for each row could be determined in Fig. 2 (b). As shown, the five staff lines were associated with five obvious peak values in terms of number of pixels.


(a)

(b)

Fig. 2 Binary musical score image and the corresponding horizontal projects

Region Segmentation: The objective of the *region segmentation* was to identify and segment regions of stave from the original image such that each region of the staff could be processed independently.

Based on the result given in Fig. 2, the staff line height H_L, the staff line space S_L, and the distribution of staff lines, could be obtained. An example is shown in Fig. 3.

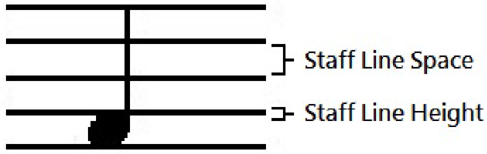


Fig. 3 An example of the staff line height H_L and the staff line space S_L. The height of the note head is approximately the same as the staff line space

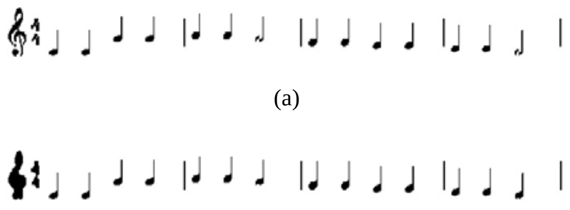
According to the horizontal projections, the histogram represented the number of pixels for each row in the binary image. Because each staff contained exactly five horizontal staff lines, the peaks of the horizontal projections at the location of staff line represented the locations of the five staff lines. Therefore, the five staff lines could be obtained by back-projections of the five peaks in the histogram, such that the resulting image contains only the staff lines without any note heads (or other symbols).

B. Image Preprocessing

In image preprocessing, our objective was to extract or isolate the musical symbols to be independent regions from the staff by removing the staff lines in the image. However, during staff line removal processes, several musical notes (or other symbols) may be damaged if they are associated with weak structures (shapes). Therefore, our system incorporated the image preprocessing processes to retain the musical notes (or symbols), while removing the staff lines for further processes.

***Staff Line Filtering*:** In the musical score, musical symbols are recorded on the staff. As a result, musical symbols are connected with the staff lines in images. In this step, our objective was to retain the music notes (or symbols) in images, while removing the staff lines. After acquiring the Staff line space and height by horizontal projections, our system removed all the black pixels at each rows of staff lines. Because this process may damage the structure (shapes) of musical notes (or symbols), additional criterion was included. The staff line height H_L was selected as the threshold and black pixels were removed only if the observed height was smaller than the staff line height. An example is shown in Fig. 4 (a).

Morphological Processing: Although the aforementioned process was able to remove staff lines effectively, the structure (shapes) of the musical notes (or symbols) could be affected. The process of *Morphological processing* was used to retain the complete structure (shapes) of each musical note (or symbol).



(a)

(b)

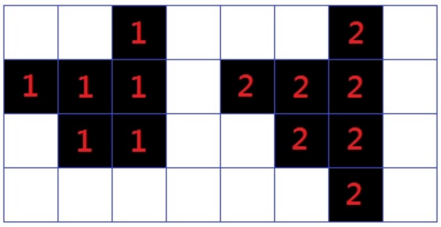
Fig. 4 An example of the image preprocessing: (a) result of staff line filtering; (b) result of morphological processing

Closing processing is a technology of morphological processing that can be used to link the small gaps of structure and to improve the connectivity for regions in images. The morphological closing is defined by:

$$I \bullet E = (I \oplus E)\Theta\ E \qquad (2)$$

where I is the input image and is E the structuring element. The image is first dilated ($\oplus$ is the dilation operation) and then eroded (Θ is the erosion operation) with the structuring element. In our system, the morphological closing was applied, an example is shown in Fig. 4 (b).

Connected Component Labeling: A region of connected pixels with an identical label was referred as a connected component. The objective of the connected component labeling was to assign a unique label to each connected component. An example is shown in Fig. 5.



		1				2	
1	1	1		2	2	2	
	1	1			2	2	
						2	

Fig. 5 An example of the connected component labeling is shown. After labeling, each connected component (region) is assigned a unique label

Given a binary images with *binary*-1s (black pixels) and *binary*-0s (white pixels), we applied the connected component labeling to extract each connected component. Therefore, each musical notes (or symbols) could be identified and labeled. The seed filling algorithm for the connected component labeling is given by:

1) Search each pixel in the image until the value of the current pixel *P* with the *binary*-1 value and has not been labeled yet;
2) Select *P* as the seed pixel and assign a label *L* to *P*, then check each pixel that is adjacent to the seed pixel;
3) If there is a pixel Q with the *binary*-1 value that is adjacent to the seed pixel, return to step 2 until there is no pixels with the *binary*-1 is found;
4) Update the label and return to step 1 until all pixels in the image have been checked.

C. Note Recognition

Once the connected regions for the musical notes (or symbols) were identified and labeled, the final step was to recognize them. In our system, the processes were divided into two major recognition phases: (1) Note recognition; and (2) Accidental and rest recognition.

In staff notations, a musical note can be split into three parts according to its structure: head, stem and tail, as shown in Fig. 6. The head (e.g., solid or not) is mainly used to determine the pitch by its position with respect to the staff lines; the tails is mainly used to determine the beat; the stem is used to connect both the head and the tail. Our system was designed to detect the note heads first, followed by the detection of stems and tails.

Stem Filtering: A stem is used to connect the head and the tail in a musical note. With the different position of a musical note, the stem is either extended upward or downward form the head. In addition, a whole note has no stems. Because of the variety in structures (shapes), recognition of musical note could be difficult. To simplify the task, the *Stem filtering* was designed to remove stems for the musical notes, while retaining the structure of the note heads.

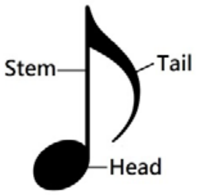


Fig. 6 An example of a typical musical note (i.e., eighth note): The structure consists of head, stem and tail

In this step, vertical projections were applied and the histogram was acquired to determine the approximated location of each musical note. During vertical projections, peaks in the histogram were related to the location of the stems, despite there are different types of musical notes (e.g., 4th or 8th notes, etc.). Therefore, stems of each musical notes could be filtered (removed).

Size Filtering: Although there are differences in representing musical notes by different publishers, the height of the note heads is generally the same as the staff line space. Therefore, the staff line space S_L was used as the threshold to determine if a connected region actually represents the head of a musical note.

Shape Filtering: The head of a musical note is generally an oval shape which is symmetrical with respect to its center. Here, we detected note heads by calculating the rate of symmetric for each connected component.

In a symmetric connected component *L*, for any point $P \in L$, there exists a point $P_S \in L$ such that $P - P_C = P_C - P_S$, where P_C is the center of the connected component *L*, as defined by:

$$\forall P \in L, \exists ! P_S : P - P_C = P_C - P_S \tag{3}$$

According to the equation, the rate of symmetry *R* can be obtained for each connected component, as given by (4)

$$R = \frac{Sum_S}{Sum_A} \tag{4}$$

where Sum_s is the total number of the symmetric pixels and Sum_A is the total number of all the pixels in the connected component. Using the rate of symmetry, we could therefore remove all the regions that were not likely to be the heads of musical notes. An example is shown in Fig. 7.

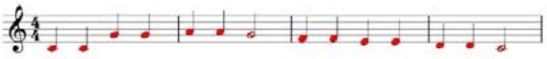


Fig. 7 An example of the note recognition, in which all the heads of the musical notes are marked

Pitch and Beat Analysis: The pitch of the musical note is defined by its position with respect to the staff lines. The difference of pitches among musical notes is based on the *scale* as a basic unit, and the distance of each scale is with half height of the staff line space. Hence, we defined the pitch of each note by calculating the distance between musical notes and the datum line. The datum line was defined as the position of the keynote, i.e., the position of middle *C*.

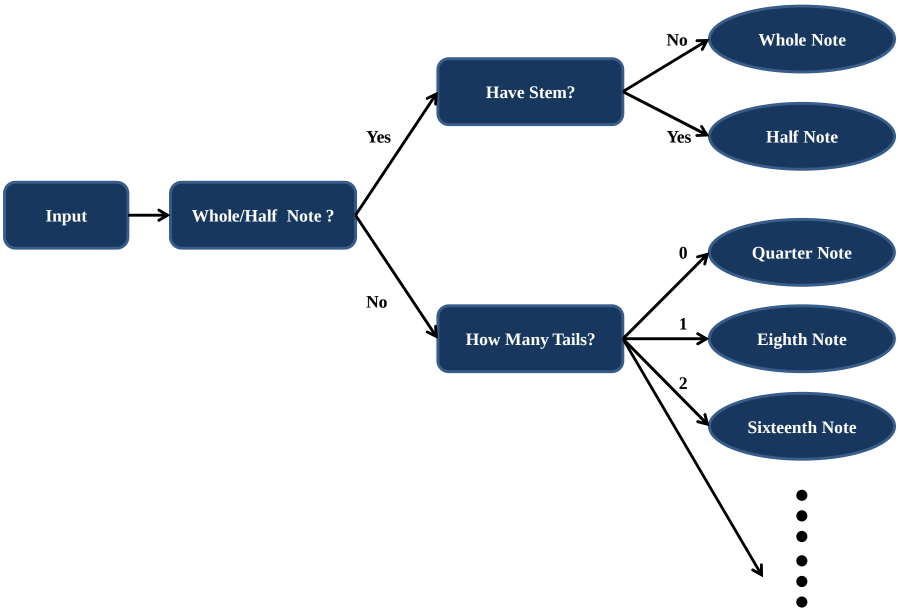


Fig. 8 The classifier for the beat of a musical note: Each musical note can thus be classified based on the properties if the stem/tails exist

The beat of a musical note represents the length of the note being played, and each note has its own beat. The beat of a musical note is represented by three parts: (1) The note is solid or not; (2) The note has a stem or not; and (3) How many tails do the note have. Based on these properties, our system incorporated a classifier for the beat of a musical note, as shown

Open Science Index, Computer and Information Engineering Vol:9, No:7, 2015 publications.waset.org/10005799/pdf

in Fig. 8. Furthermore, a dot is often used to adjust the beat for the musical notes. The dot is meant to increase the beat of the musical note by half of its original beat. For example, a note with two beats will become three beats. The symbol of a dot in a music score is a round black spot, and is generally marked at the right side of the note head. The size of dots is smaller than the half size of heads and is disconnected with other symbols. Based on the dot property as described, our system was designed to incorporate additional process for the process of recognizing the dot associated with a note head.

D. Accidental and Rest Recognition

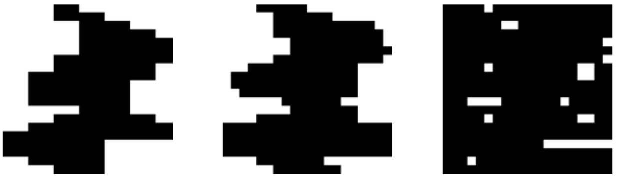
Accidentals are the musical symbols used to modify the pitch of a musical note. The most common accidentals can be described as follow: (1) the *sharp* is used to raise the pitch of a musical note by a semitone; (2) the *flat* is used to reduce the pitch of a musical notes by a semitone; and (3) the *natural* is used to recover the pitch of a musical note to its natural key. The accidentals are typically recorded in two ways: (1) marked at the start of a staff represent a key signature; and (2) marked at the left of a note to adjust the pitch of the musical note.

Rests are the musical symbols used to represent the pauses in music/song. Unlike musical notes, rests have no pitches so that the height of rests in a score is fixed. However, shapes of rests with different beats are relatively irregular than musical notes.

To identify the accidentals and/or rests, the technique of template matching is used in our system. The technique was used to compare an unknown symbol with respect to known template images (i.e., template images for possible accidentals and/or rests) in the database to recognize the symbol. In our system, once a region (sub-image) containing a symbol was detected, the region (sub-image) was then normalized to the same size with the template image and the logical XOR operation was applied:

$$I_{XOR}(x,y)=\begin{cases} 0, & if \quad I_T(x,y)=I_C(x,y) \\ 255, & \quad otherwise \end{cases} \tag{5}$$

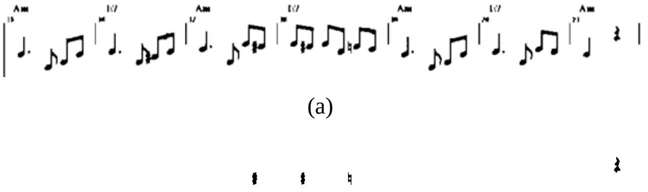
where I_T represent the template image, I_C represent the sub-image containing a symbol, and I_{XOR} represent the result image after the exclusive-OR operation. An example is shown in Fig. 9.



(a) (b) (c)

Fig. 9 An example of the template matching for the rest recognition: (a) sub-image of an unknown symbol; (b) template image of the *crotchet rest*; (c) resulting image after the exclusion-OR operation

Fig. 10 shows an example for the recognition of accidentals and rests in a musical score. Using the template matching, our system was able to identify the accidentals and rests. However, our system failed to detect the sharp symbol which is connected with the tail of a musical note.



(a)

(b)

Fig. 10 An example of the accidental and rest recognition: (a) input image; (b) result of the recognition. There is a sharp that is connected with the tail of a musical note, resulting in a recognition failure in our system

III. Results

In this section, we present the research environment and recognition results using our system in several musical score images.

A. Research Environment

The system development was based on a personal computer: Pentium(R) Dual-Core E5200 2.5GHz with 2GB memory, and Microsoft Windows 7 operating system. The system software was developed using the Microsoft Visual Studio C/C++ 2010 and the Intel Open Source Computer Vision Library (OpenCV) Version 2.4.8.

To evaluate our system performance, a set of digital images with musical scores of various complexities were collected from the Internet.

B. Result of Musical Score Recognition

Table I summarizes the results of musical symbol detection and recognition of our system using five images of musical scores. The detection rate was evaluated with the probability when musical notes or symbols (including accidentals and rests) were correctly detected. Then, based on the detected notes or symbols, the recognition rate was evaluated with the probability when the musical notes or symbols were correctly recognized.

TABLE I
The Results of Musical Symbol Detection and Recognition

	Detection		Recognition	
	Detected / Total	Rate	Recognized / Detected	Rate
Image 1	42 / 42	100%	42 / 42	100(%)
Image 2	62 / 62	100%	62 / 62	100(%)
Image 3	74(1) / 74	97.3%	64 / 74	86.4(%)
Image 4	85(4) / 88	87.5%	65 / 85	76.5(%)
Image 5	89(1) / 92	96.7%	85 / 89	95.5(%)
Total	382(6) / 388	96.3%	318 / 382	91.7(%)

Recognition results of our system are shown in the following. Fig. 11 shows the recognition results for the musical score *Twinkle, Twinkle, Little Star*. All the musical notes, including the pitches and beats, were successfully detected and recognized. The detected musical notes are marked accordingly.

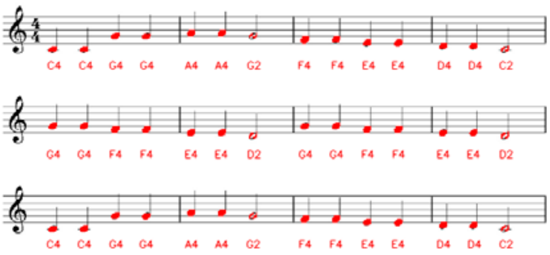

C4 C4 G4 G4 A4 A4 G2 F4 F4 E4 E4 D4 D4 C2
G4 G4 F4 F4 E4 E4 D2 G4 G4 F4 F4 E4 E4 D2
C4 C4 G4 G4 A4 A4 G2 F4 F4 E4 E4 D4 D4 C2

Fig. 11 Recognition results of the musical score *Twinkle, Twinkle, Little Star*

Fig. 12 shows the recognition results for the musical score *The Swallow*. Although the musical score was more complicated, all the musical notes, including the pitches and beats, were successfully detected and recognized.

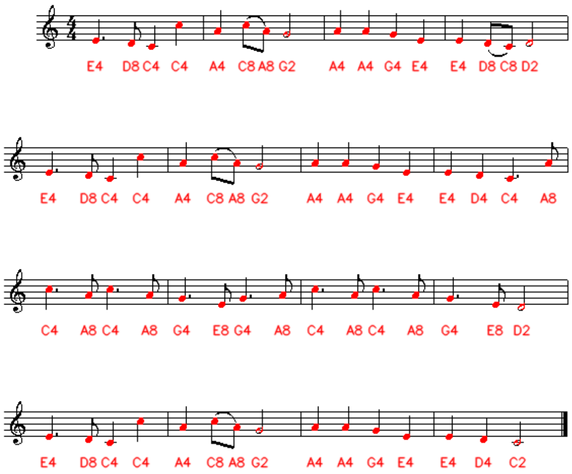

E4 D8 C4 C4 A4 C8 A8 G2 A4 A4 G4 E4 E4 D8 C8 D2
E4 D8 C4 C4 A4 C8 A8 G2 A4 A4 G4 E4 E4 D4 C4 A8
C4 A8 C4 A8 G4 E8 G4 A8 C4 A8 C4 A8 G4 E8 D2
E4 D8 C4 C4 A4 C8 A8 G2 A4 A4 G4 E4 E4 D4 C2

Fig. 12 Recognition results of the musical score *The Swallow*

Fig. 13 shows the recognition results for the musical score *I Will Sing You*. The original image was in JPEG compressed format. All the musical notes, accidentals, and rests were successfully detected and recognized, despite there was a few errors in the recognition of the pitches.

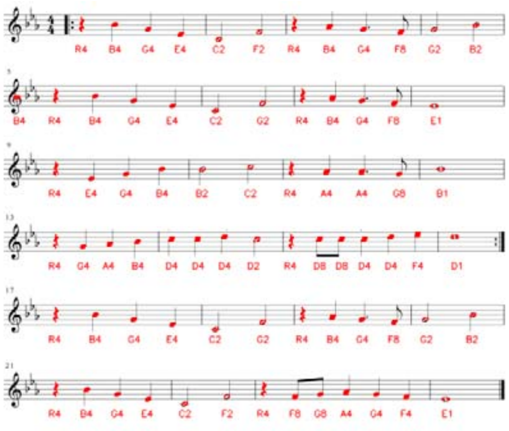

R4 B4 G4 E4 C2 F2 R4 B4 G4 F8 G2 B2
B4 R4 B4 G4 E4 C2 G2 R4 B4 G4 F8 E1
R4 E4 G4 B4 B2 C2 R4 A4 A4 G8 B1
R4 G4 A4 B4 D4 D4 D4 D2 R4 D8 D8 D4 D4 F4 D1
R4 B4 G4 E4 C2 G2 R4 B4 G4 F8 G2 B2
R4 B4 G4 E4 C2 F2 R4 F8 G8 A4 G4 F4 E1

Fig. 13 Recognition results of the musical score *I Will Sing You*

Fig. 14 shows the recognition results for the musical score *At This Moment*. The quality was relatively poor mainly because of compression distortion. The recognition results were relatively worse than previous scores.

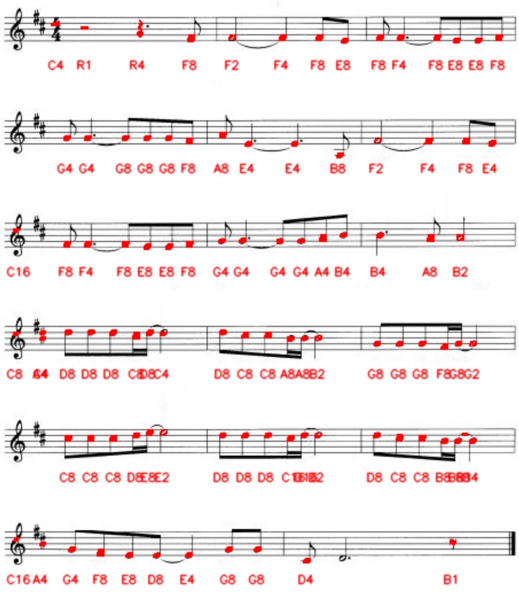

C4 R1 R4 F8 F2 F4 F8 E8 F8 F4 F8 E8 E8 F8
G4 G4 G8 G8 G8 F8 A8 E4 E4 B8 F2 F4 F8 E4
C16 F8 F4 F8 E8 E8 F8 G4 G4 G4 G4 A4 B4 B4 A8 B2
C8 D8 D8 D8 D8 D8 C8 C8 A8 G8 G8 G8 G8
C8 C8 C8 D8 D8 D8 D8 C8 C8
C16 A4 G4 F8 E8 D8 E4 G8 G8 D4 B1

Fig. 14 Recognition results of the musical score *At This Moment*

Fig. 15 shows the recognition results for the musical score *My Herd*. The quality was also relatively poor mainly because of compression distortion. The recognition results were relatively worse than previous scores especially in beat recognition (most recognition failure occurred at dots).

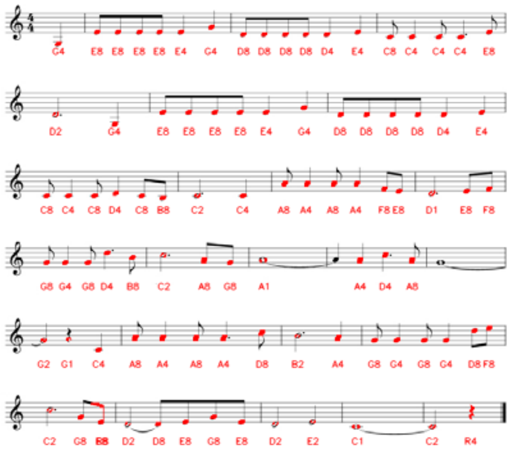

G4 E8 E8 E8 E8 E4 G4 D8 D8 D8 D8 D4 E4 C8 C4 C4 C4 E8
D2 G4 E8 E8 E8 E8 E4 G4 D8 D8 D8 D8 D4 E4
C8 C4 C8 D4 C8 B8 C2 C4 A8 A4 A8 A4 F8 E8 D1 E8 F8
G8 G4 G8 D4 B8 C2 A8 G8 A1 A4 D4 A8
G2 G1 C4 A8 A4 A8 A4 D8 B2 A4 G8 G4 G8 G4 D8 F8
C2 G8 B8 D2 D8 E8 G8 E8 D2 E2 C1 C2 R4

Fig. 15 Recognition results of the musical score *My Herd*

IV. Conclusion

In this study, we proposed an *Automatic Music Score Recognition System Using Digital Image Processing*. The technical approaches included: staff region segmentation, image preprocessing, note recognition and accidental and rest recognition. Our system was developed to automatically detect and recognize musical notes, accidentals and rests in printed musical scores.

The results showed that the detection and the recognition rates of our system were 96.3% and 91.7%, respectively. While the results were limited and could be affected by image quality, our system was shown to achieve effective detection and recognition for musical symbols, such as notes, accidentals, and rests. Ultimately, our system could be incorporated with a media player that could play music/songs with inputs of musical scores.

While our system has been demonstrated with success for different types of musical scores, our system was still limited with respect to the complexities of some musical scores (e.g., chords or other complex symbols). Improvement of our systems is still required for such musical scores.

At present, our system was evaluated using personal computers with inputs of digital images. The system could be further integrated in applications for smart-phones or other mobile devices with built-in digital cameras. In addition, this system could be also used as a learning tool for players (e.g., piano players, guitar players, etc.) to play the music/songs even though they may not be familiar with the music/songs.

REFERENCES

[1] N. Otsu, "A Threshold Selection Method form Gray-Level Histograms," *IEEE Transactions on Systems*, pp. 62-66, 1979.

[2] R.G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 690-706, 1996.

[3] Cheng-Lin Liu, M. Koga and H. Fujisawa, "Lexicon-Driven Segmentation and Recognition of Handwritten Character Strings for Japanese Address Reading," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1425-1437, 2002.

[4] M. Sotoodeh and F. Tajeripour, "Staff Detection and Removal Using Derivation and Connected Component Analysis," *IEEE 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, pp. 54-57, 2012.

[5] Chen Genfang, Zhang Liyin, Zhang Wenjun and Wang Qiuqiu, "Detecting the Staff-lines of Musical Score with Hough Transform and Mathematical Morphology," *IEEE International Conference on Multimedia Technology (ICMT)* , pp. 1-4, 2010.

[6] A. Dutta, U. Pal, A. Fornes and J. Llados, "An Efficient Staff Removal Approach from Printed Musical Documents," *IEEE International Conference on Pattern Recognition (ICPR)*, pp.1965-1968, 2010.

[7] JaeMyeong Yoo, GiHong Kim and Gueesang Lee, "Mask Matching for Low Resolution Musical Note Recognition," *IEEE International Symposium on Signal Processing and Information Technology*, pp. 223-226, 2008.

[8] F.Toyama, K. Shioji and J. Miyamichi, "Symbol Recognition of Printed Piano Scores with Touching Symbols," *Pattern Recognition, ICPR 18th International Conference*, pp. 480-483, 2006.

[9] F.Rossant and I. Bloch, "Optical Music Recognition Based on a Fuzzy Modeling of Symbol Classes and Music Writing Rules," *Pattern Recognition Letters, vol.23,* pp. 1129-1141, 2002.

[10] K.T. Reed and J.R.Parker, "Automatic Computer Recognition of Printed Music," *in Proceedings of the ICPR,* pp.803-807, 1996.

Open Science Index, Computer and Information Engineering Vol:9, No:7, 2015 publications.waset.org/10005799/pdf