

Optical Character Recognition for Musical Notes and Playback

Kartik B. Bhargav, *Software Engineer, Airwatch*, Kaushik M.K., *Software Engineer, Y-MediaLabs*,
Kaustubh B. Bhargav, *Software Engineer, Scientific-Games*, Nataraj C.R., *Professor, Dept. of E&C SJCE Mysore*,
and Neelanjana E.K., *Hardware Engineer, General Motors*

Abstract—OCR for music recognition and playback aims to address the problem of music data acquisition. It accomplishes this by using OCR to recognize the music score and converts it into MusicXML file format. MusicXML file is the industry standard way of representing western musical notation. The conversion of the recognized symbols to MusicXML file format enables the user to store, edit, play and share his music. The main motivation behind the project is to develop software like “Cam Scanner” for musical score sheets that can be used by learners, professional musicians and publishers with ease. The character recognition has been implemented through template matching using normalized cross correlation. The Music table of recognized symbols with (x,y)coordinates, staff number, duration, stem, octave, step and alter is generated, which will be helpful to test the correctness of the recognized symbols. Currently 10 different instruments such as guitar, violin, mandolin, melodica, banjo, koto, kazoo, electric guitar, sitar, mezzo-soprano are supported for music playback by using Muse Score software. The implementation was done through in Matlab(version R2014a - 8.3.0.532).

Keywords—OCR, MusicXML, Cam Scanner, Template matching, Normalized cross correlation, Matlab.

I. INTRODUCTION

A. Subsection Heading Here

II. MUSICXML

MusicXML (Music Extensive Markup Language) is a digital sheet music interchange and distribution format. The goal is to create a universal format for common Western music notation, similar to the role that the MP3 format serves for recorded music.

The musical information is designed to be usable by notation programs, sequencers and other performance programs, music education programs and music databases. It is designed from the ground up for sharing sheet music files between applications and for archiving sheet music files for use in the future. MusicXML files are readable and usable by a wide range of music notation applications. MusicXML complements the native file formats used by several musical score writing programs, which are designed for rapid and interactive use. MusicXML files are the standard for sharing interactive sheet music. Using MusicXML, users can create music in one program and share the results back and forth with people using other programs.

MusicXML was based primarily on two academic music formats:

- The MuseData format, developed by Walter Hewlett at the Center for Computer Assisted Research in the Humanities (CCARH), located at Stanford University .
- The Humdrum format, developed by David Huron, based at Ohio State University.

III. SOFTWARE REQUIREMENTS

The input image acquired has to be subjected to various processing operations for recognising the characters in the music score sheet. Thus, the platform used for the implementation should support image processing operations. Matlab has extensive support for image processing operations through Image Processing Toolbox”. Matlab also has Data and File Management Toolbox which can be used to read, write and generate XML files. MusicXML being one of the types of XML file, the toolbox will be able to handle it’s basic operations also. Hence Matlab becomes a good choice for implementation. Matlab version R2014a - 8.3.0.532 has been used for the implementation of the algorithm. For reading MusicXML files MuseScore can be used. MuseScore is a free and opensource score writer software with rich features.

IV. IMPLEMENTATION

This section presents the steps involved in converting the captured image of a score sheet into MusicXML file.

A. Image Acquisition

The music score sheet image is acquired from MuseScore software. The file format used is “png”. A sample image of score sheet is shown in figure2

B. Image Preprocessing

The image acquired will be in rgb format and is converted into grey scale image. The purpose of converting an RGB image into gray scale is due to the fact that it eliminates the hue and saturation information, as they contribute very less for the total appearance of the image and retains the needed luminance (intensity). The main advantages of image preprocessing are it helps in reducing the noise in images, Variations in illumination and viewing geometry between images (for optical sensors) can be corrected by image pre-processing techniques and helps to convert the image into a form that is more suitable for further operations like morphological operations and feature extraction.

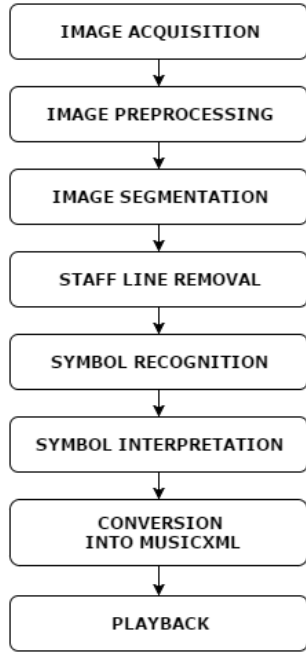


Fig. 1. Flow chart for the steps involved conversion of image of score sheet to MusicXML file

C. Image Segmentation

Image segmentation is the process of partitioning an image into parts or regions. This division into parts is often based on the characteristics of the pixels in the image. Thresholding method of segmenting method is used here to convert a grey scale image into binary image. This transformation is important because the morphological operations like dilation, erosion, etc. can be done efficiently on a binary image. Examples of a segmented image and a non-segmented image are shown in Figure 4.4 and Figure 4.3. Figure 4.3 contains pixel values in the gray scale and each pixel can take any value out of the available 256 gray scale levels. As seen from figures 4.3 and 4.4, image segmentation converts the gray scale image into a binary image.

D. Morphological Image processing

Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. Morphological operations rely only on the relative ordering of pixel values, not on their numerical values and therefore are especially suited to the processing of binary images. Morphological techniques typically probe an image with a small template known as a Structuring Element (SE). The SE is very small compared to the image size. They are usually made up of ones and zeroes. The SEs usually have odd orders like 3x3, 5x5, 7x7 etc. because, while performing any morphological operations, the centre pixel of the SE is placed on each and every pixel of the image and the operation is thus performed. The SE is positioned at all possible locations in the image and it is compared with the corresponding neighborhood

pixels. By marking the locations where SE fits or hits the image, information about the structure of the image can be obtained. The SE “fits image if for each of its pixels that is set to 1 (foreground), the corresponding image pixel is also set to 1

. Dilation and erosion are morphological image processing techniques which are extensively used in staff line removal.

E. Staff Line Removal

Removing staff lines is important, because in the score sheet essentially the same symbol will be placed on different staff lines. Without this step, the same symbol will be treated as different when they are on different staff lines. For removal of staff lines morphological image processing techniques such as dilation and erosion are used. The image is subjected to dilation two times which is then followed by erosion.

1) *Dilation*: The dilation of an image f by a structuring element s (denoted by $f \oplus s$) produces a new binary image

$$g = f \oplus s$$

with ones in all locations (x,y) of a structuring elements origin at which that structuring element s hits the the input image f , i.e. $g(x,y) = 1$ if s hits f and 0 otherwise, repeating f , i.e. $g(x,y) = 1$ if s hits f and 0 otherwise, repeating.

The holes enclosed by a single region and gaps between different regions become smaller, and small intrusions into boundaries of a region are filled in. Figure 4.5 shows the effects of dilation on an input image.

2) *Erosion*: The erosion of a binary image f by a structuring element s (denoted by $f \ominus s$) produces a new binary image

$$g = f \ominus s$$

with ones in all locations (x,y) of a structuring elements origin at which that structuring element s fits the input image f , i.e. $g(x,y) = 1$ if s fits f and 0 otherwise, repeating for all pixel coordinates (x,y) .

Erosion with small square structuring elements shrinks an image by stripping away a layer of pixels from both the inner and outer boundaries of regions. The holes and gaps between different regions become larger and small details are eliminated.

Larger structuring elements have a more pronounced effect, the result of erosion with a large structuring element being similar to the result obtained by iterated erosion using a smaller structuring element of the same shape. If s_1 and s_2 are a pair of structuring elements identical in shape, with s_2 twice the size of s_1 , then Erosion removes small-scale details from a binary image but simultaneously reduces the size of regions of interest, too. By subtracting the eroded image from the original image, boundaries of each region can be found:

$$b = f - (f \ominus s)$$

where f is an image of the regions, s is structuring element, and b is an image of the region boundaries. Figure 4.6 shows the effects of erosion on an input image

APPENDIX A PROOF OF THE FIRST ZONKLAR EQUATION

Some text for the appendix.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.



Kartik Bhargav Top entrepreneur