C.S. MUSE: Musical Note Recognition System

Lauro A. Salazar Garcia

Advisor: Dr. Mahmoud K. Quweider

Department of Computer and Information Sciences

University of Texas at Brownsville and Texas Southmost College

*Abstract –* The recognition of musical notes is a complex problem in the area of character recognition. Currently there exists no standard in the fields of Computer Vision and Image Processing to solve this problem in a manner that is both efficient and adequate. We propose an experiment in which isolated musical notes are identified according to (1) their duration (quarter notes, half notes, and whole notes), (2) to their relative frequency (given by the position on the staff) and (3) by their shape. This paper describes methods to provide an algorithm which is simple, accurate, and fast. The future of this research is very promising, with potential for development ranging in areas such as robotics to education.

I. Introduction

*A. Musical Background*

Reading musical notation is similar to reading a standard alphabet in a written language. There are mainly three types of notes discussed in this paper; whole notes, half notes, and quarter notes. All of which can be placed at different positions of the staff and thus give their duration and frequency. (Fig. 1, Fig 3) Others exist such as eight notes, sixteenth notes, thirty-second notes, and sixty-fourth notes. (Fig. 2)



Fig.1 Note Duration



Fig.2 Eight Notes, Sixteenth Notes

The clef identifies the pitch depending on the instrument. We used the treble clef, which is the first symbol from left-to-right. Lower pitched instruments use a different notation called the bass clef. Some examples of instruments in the treble clef generally include guitar, flute, and saxophone. Some bass clef instruments include French horns, bass guitar, tuba, and cello. The piano is a mixture of both.



Fig. 3 Frequency name in treble clef

In conclusion for notation covered in this experiment; the time signature states how many notes are in a measure. For example, the 4/4 time signature means there exists four quarter notes per measure (Fig.1). This is the most common time signature in popular music.

II. Recognition Method

*A. Data*

A set of testing images was created using a musical score software. This software creates standardized images of musical score like the previous Fig. 1 & 2. These images were later cropped manually to a convenient size for manipulation. This method enabled us to use data that would be consistent throughout our project and contained a minimum amount of noise. We can take advantage of this standardization for the counts of pixels in the image, matching, and shape. There were a total of twenty-seven images and sounds for playback created. Three for each musical note in the range of the staff we are using (Fig.3). Each note was created at three different durations: quarter, half, and whole. The following is an example of a test image (Fig. 4):



Fig. 4 Frequency: C, duration: quarter

*B. Image Processing and Isolation*

It is convenient for us that most of our data sets are of black and white origin, but there are still slight variations in color and noise in the image that are not seen. In order for our data sets to be usable, we used an automatic threshold algorithm and changed our image to pure black and white normalizing the image down to 0’s and 1’s. This got rid of the staff lines as well, which we do not need for this experiment.

The same was applied to an image of an empty staff which contained the same symbols except for our note. Now that both the test image and the empty staff image were normalized, we can subtract the empty staff by the test image in order to obtain our isolated note (Fig. 5).



Fig. 5 Isolated quarter note

After finding our isolated note, we tested edge detection algorithms and found that the Sobel detection delivered the best results in all the cases of quarter notes, whole notes, and half notes.

*C. Template Matching by Correlation*

*Definition -* Correlation can be used to locate features within an image; in this context, correlation is often called template matching. By multiplying the Fourier Transform of one function by the complex conjugate of the FT of the other gives the FT of their correlation [3]. Fourier Transform is often called the frequency domain representation of a function. It describes which frequencies are present in the original function [4].

A template image was created which contains only the note which it is going to be matched. This note was processed and its edge was detected as well. This improves the chances of matching.



Fig. 6 Template Image

We take the template image (Fig. 6) and then match it to our isolated note (Fig. 5). The process by which correlation works is extremely fast, surpassing matching methods such as pyramid matching. The match is almost instant since it converts the image into the frequency domain and is unambiguous to noise in the image. Several notes and symbols on the staff can be identified by this method.



Fig. 7 Quarter note matched

Since our set of data is standardized, from our match we are also given the row and column at which the note is located. This gives us the position on the staff, and thus the frequency of the note. From our match, we can say that this is a C note, but we still have to find the duration and shape.

*D. Image Histogram*

An image histogram is a count of each individual color in an image saved into a bin. From the image histogram we can use the counts of normalized pixels to find the duration of the note. Quarter notes have a higher count of pixels than half notes and whole notes (Fig. 1).

After finding our matched image, we take the window that was created to segment the note on the position it is at. We use this image to create the two bin histogram (Fig. 8).

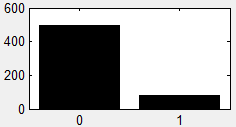
**

Fig. 8 Quarter note histogram

From our previous steps, we can identify our note as being of frequency “C” by the matched position and of duration “quarter” by the histogram; but we still need to check for one more thing. It happens that in our set of standardized data, half notes and whole notes have about the same pixel count. The solution to this problem is very simple; we need to find its shape.

*E. Elliptic Fourier Descriptors*

*Definition FD* – Fourier descriptors characterize a contour by a set of numbers that represent the frequency content of a whole shape. Based on frequency analysis, we can select a small set of numbers (the Fourier coefficients) that describe a shape rather than any noise (i.e. the noise affecting the spatial position of the boundary pixels). First, we have to define a representation of a curve. It is important to notice that although a curve in an image is composed of discrete pixels, Fourier descriptors are developed for continuous curves. The first element of the Fourier components (the d.c. component) is simple the average value of the x and y coordinate, giving the coordinates of the centre point of the boundary expressed in complex form. The second component essentially gives the radius of the circle that best fits the points. Fourier descriptors are regenerative, meaning we can reconstruct the object’s shape from them [1 (pp. 285-287)].

*Elliptic FD* – Elliptic Fourier Descriptors maintain the description of the curve in a 2D space [5]. This is achieved by considering that the image space defines the complex plane. That is, each pixel is represented by a complex number. The first coordinate represents the real part, while the second coordinate represents the imaginary part. Thus, a curve is defined as *c(t) = x(t) + jy(t).* This representation is very general and can be extended to obtain the elliptic Fourier description of irregular curves [1(pp. 301-303)].

We will use the trigonometric representation of *E.F.D.* These can be computed by the discrete approximation given by:

The trigonometric representation of a curve from the elliptic Fourier descriptors can be described as:

By default, the number of coefficients is half of the number of points that define the curve. The number of coefficients used defines the level of detail of the characterization. [1 (pp. 304-309)]

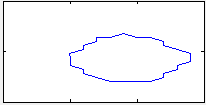
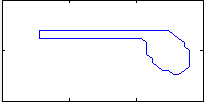
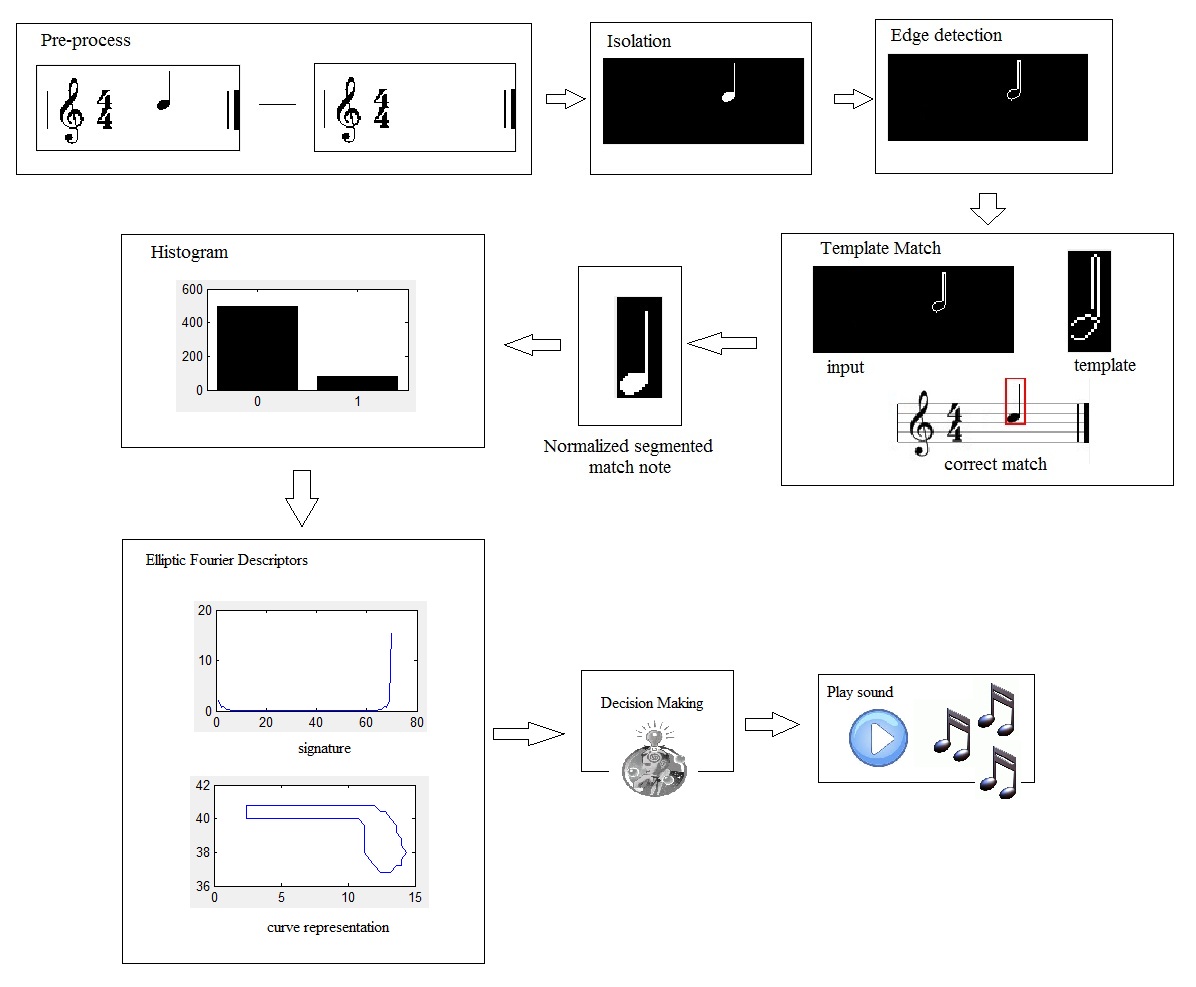


Fig. 9 Half/Quarter note shape (left) whole note shape (right)

If we are able to find the shape, then we will be able to differentiate between half/quarter notes and whole notes (Fig. 9).

System Layout



III. Results

Overall we achieved good results. The algorithm produced an 85% accuracy rate for our test cases. Only four out of the twenty-seven cases were disqualified due to their lack of matching. It is difficult for the algorithm to produce a continuous curve if the image is too close to the edge.

We used the signatures from our Fourier descriptors to discriminate the notes from Half/Quarter to Whole, a histogram for the counts in pixels to find duration, and our template matching algorithm to find our locations on the staff. To produce a sound, we created a look-up table which looked for the information of each note and designated a recorded wav file of the note from the decision. It takes about 0.38 seconds with plotting and output display in Matlab to detect each note, without plotting and output this process only takes about 0.07 seconds. This is extremely fast. It is reported on some studies that human recognition time lies between 0.27 seconds and 0.53 seconds depending on the complexity of the musical score [6][7]. This means as this algorithm becomes more complex, it still stands a chance to function at the same pace or faster than a human being.

In conclusion, this procedure seems like an adequate solution to this problem. Perhaps there are other ways to improve the matching like taking into account the lines on the staff. There are still other approaches to be tried, and as the musical score becomes more complex, so will the algorithm. Future improvements include detection for several notes and a statistical model for decision making, for example: Neural Networks and Hidden Markov Model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Note | Duration | Pixels | Coefficients | Played |
| A | Half | 53 | 69 | Y |
| A | Quarter | 82 | 70 | Y |
| A | Whole | 52 | 30 | Y |
| B | Half | 53 | 69 | Y |
| B | Quarter | 80 | 70 | Y |
| B | Whole | 54 | 30 | Y |
| C | Half | 52 | 68 | Y |
| C | Quarter | 82 | 70 | Y |
| C | Whole | 56 | 30 | Y |
| D | Half | 53 | 69 | Y |
| D | Quarter | 80 | 69 | Y |
| D | Whole | 53 | 30 | Y |
| E | Half | 53 | 69 | Y |
| E | Quarter | 80 | 70 | Y |
| E | Whole | 53 | 30 | Y |
| Espace | Half | X | X | N |
| Espace | Quarter | X | X | N |
| Espace | Whole | 55 | 30 | Y |
| F | Half | X | X | N |
| F | Quarter | X | X | N |
| F | Whole | 54 | 30 | Y |
| Fspace | Half | 53 | 69 | Y |
| Fspace | Quarter | 81 | 70 | Y |
| Fspace | Whole | 56 | 30 | Y |
| G | Half | 52 | 69 | Y |
| G | Quarter | 78 | 69 | Y |
| G | Whole | 53 | 30 | Y |
| Total |  |  | 23/27 | 85% |

References

1. Nixon, Mark, and Alberto Aguado. Feature Extraction & Image Processing. Oxford: Elsevier, 2008.
2. Weisstein, Eric W. "Cross-Correlation." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/ Cross-Correlation.html
3. Graham, James R. “Convolution, Correlation, & Fourier Transforms.” 10/25/2005. Berkeley. March 2010 http://www.ugastro.berkeley.edu/infrared/ir\_clusters/convolution2.pdf
4. Wikipedia. “Fourier Transform.” May 6 2010.http://en.wikipedia.org/wiki/ Fourier\_transform
5. Granlund, G.H., Fourier Preprocessing for Hand Print Character Recognition, *IEEE Trans. Comput.,* 21**,** pp. 195-201, 1972
6. Wikipedia. “Eye movement in music reading.” May 8 2010. http://en.wikipedia.org/wiki/Eye\_movement\_in\_music\_reading
7. Souter T (2001) Eye movement and memory in the sight reading of keyboard music. Doctoral dissertation, University of Sydney