Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Musical Image Recognition

Javier de la Rica

Musical Image Recognition
Javier de la Rica Palma
MAE – Final Report
June 2017

1. INTRODUCTION

Musical Image Recognition has been the subject of research for decades. Its traditional application has been the extraction of existing printed sheet music to digitized formats. The aim of this project is to learn and train certain skills in the field of image processing, instead of creating a complete musical image recognition software.

An accessible and easy-to-use application in this area would improve and provide an amazing tool for improving the musical education experience. The main goal of the project is to develop the algorithms needed to parse digital sheet music images and to provide a playback system for the parsed music notes.

A theoretical background is explained below, in which all the necessary steps and musical theory are shown.

First, some musical theory as the compasses and the difference between the black and white notes are explained. It is also shown a state of the art and a block diagram of the whole software to deliver a proper explanation and a big picture of the project.

Next, the different computed functions are explained one by one, as to understand the whole functionality and need of each one.

Finally, some final results and examples are shown in order to explain the functionality of the software.

2. THEORETICAL BACKGROUND

2.1. Musical Theory

In order to understand the whole Project, a bit introduction to Musical Theory and the singing of scales is needed.

In this project some basic musical theory is needed, as to know the difference between the different possible notes, and different musical objects in a sheet.

In this software we only check if a note is black or white, which sets the duration of the note. The duration of a black note is just a second, while the duration of a white note is four times the duration of a black note.

There are also different keys, as shown in the picture above, which set the meaning of each note depending on which staff line they are set.

Figure 1. *Possible keys.*

Nevertheless, the software developed in this project only recognizes the first key, in order to make the project simpler and easier to use.

Finally, when all the notes are read and recognized, the system will have to translate the notes into frequencies in order to let the software to play the frequencies using a sinusoidal signal.

In the following image a big picture of the whole project is explained in a block diagram, showing the steps the software follows in order to play and obtain the final results:

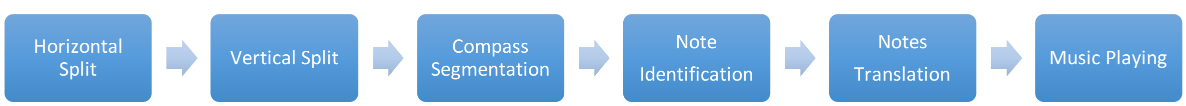

Figure 2. *Block Diagram of the whole project.*

3. MAIN FUNCTIONS

3.1. Segmentation

In order to read the complete score it is necessary to focus on each of the elements of the music piece. For that reason, the first aim to achieve is to divide the image horizontally using a Horizontal Split in order to divide every horizontal element corresponding to the staff lines and every other information as the title and the author.

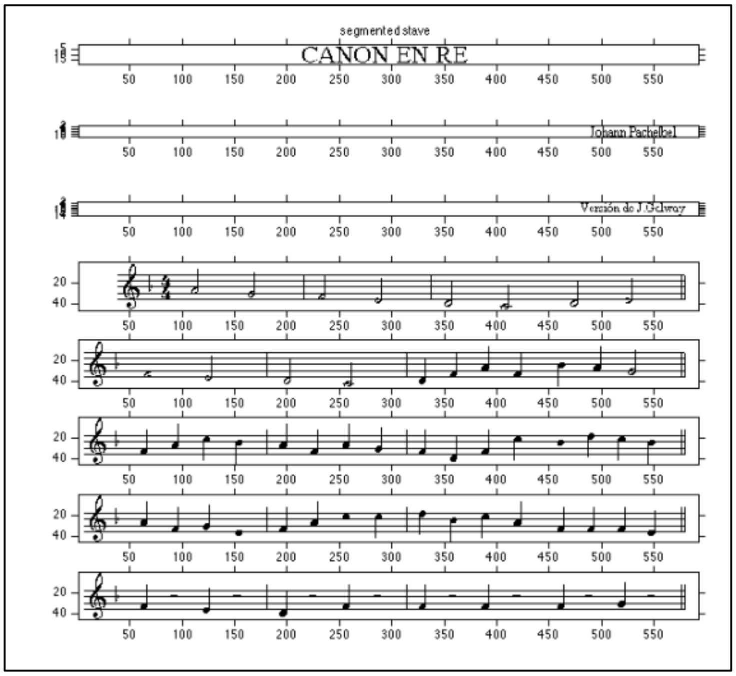

Figure 3. *Output of the Horizontal Split function.*

Once all the horizontal elements are separated, the second step is the staff detection. With this function we obtain the most important images, corresponding to the ones containing staff lines. These are recognized computing horizontal histograms for every image, and selecting those which have 5 maximum peaks corresponding to the 5 staff lines.

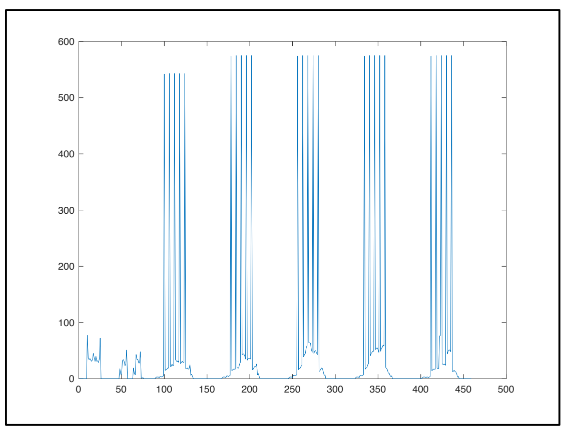

Figure 4. *Horizontal Histogram corresponding to the 5 staff lines in the canon.jpg image.*

The next step is to divide all the compasses with a vertical split. This segmentation is done taking into account that each bar-line presents a maximum in a vertical histogram. Later, in order to divide every element of each compass, is made with vertical splits again.

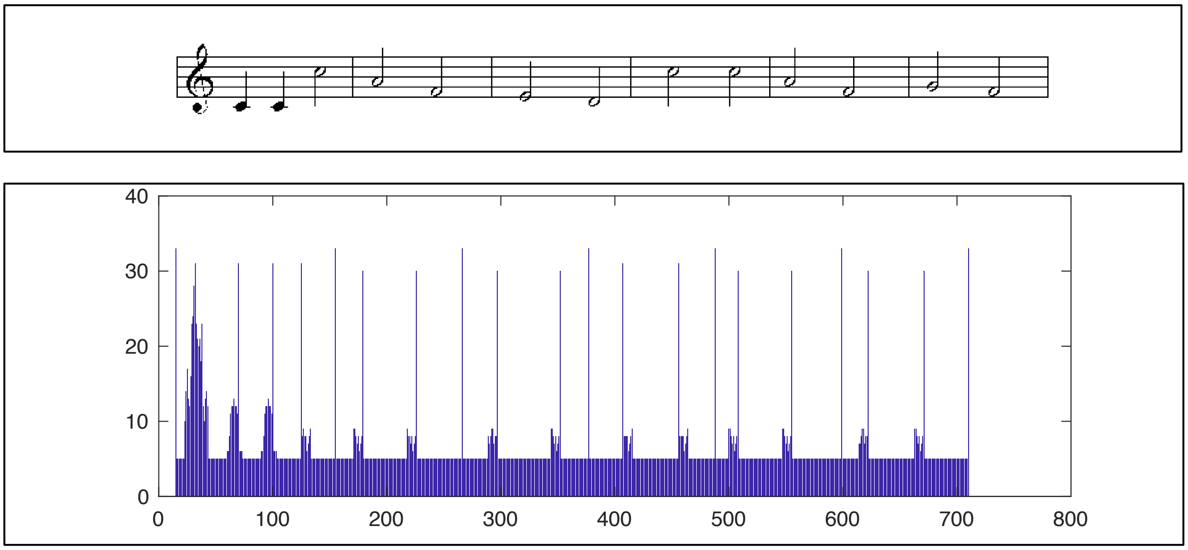

Figure 5. *Vertical Split and Vertical histogram of the staff lines.*

3.2. Note identification

At this point every element in the sheet music is in a separate image. In order to recognize and identify which element is each one, all the images are cleaned up to erase the staff lines, so it is easier to recognize the notes. The images are then compared using the correlation with the elements stored in a previously saved database.

The database used is very basic, containing only the elements needed to identify the scores given. It is designed in three different tables; one for the notes, one for the silences and the other one for other objects.

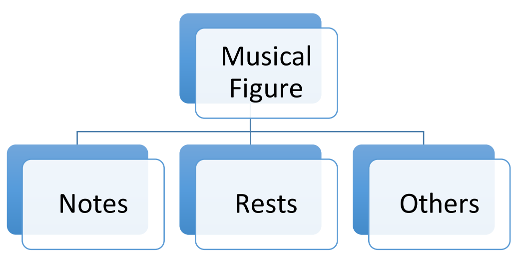

Figure 6. *Database hierarchy.*

database{1, 1}

	1	2
1	'redonda'	*12x20 logical*
2	'blanca'	*49x16 double*
3	'blanca'	*49x16 double*
4	'negra'	*48x16 double*
5	'negra'	*48x16 double*
6	'blanca'	*24x12 logical*

database{1, 2}

	1	2
1	'silencio de redonda'	*6x21 logical*
2	'silencio de blanca'	*6x22 logical*
3	'silencio de negra'	*36x12 logical*

database{1, 3}

	1	2
1	'clave de sol'	*134x48 logical*
2	'bemol'	*82x25 logical*
3	'compas: 3/4'	*33x12 logical*
4	'compas: 4/4'	*25x9 uint8*
5	'Fin'	*45x15 uint8*

Figure 7. *Insight of the database.*

On the one hand, if a note is detected, we need to compute its pitch in order to recognize the note. On the other hand, if a silence is detected, we only need the information of its time length.

The pitch of each note is computed with the horizontal histograms in order to isolate the note head. The image containing the note is divided in horizontal areas, computed according to the sheet, its size and the space between the staff lines. The pitch is then computed checking in which area the note head is placed.

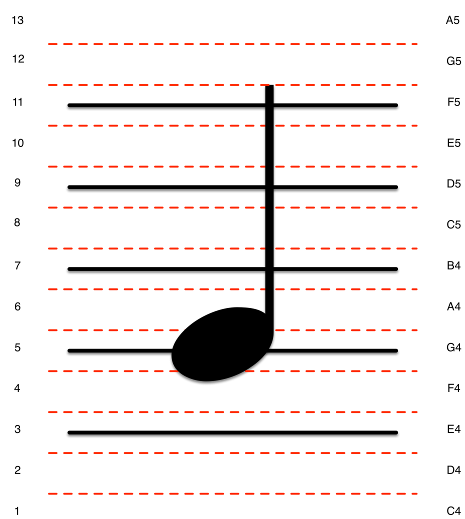

Figure 8. *Space division.*

Finally, once the software detects the String of characters corresponding to the notes of the music sheet, the system should play the song correctly. To do so, a function called *playing* is computed, in which every note from the database is related to a single frequency, depending on the corresponding following scale.

```
{'c4';'d4';'e4';'f4';'g4';'a4';'b4';'c5';'d5';'e5';'f5';'g5';'a5'}
```

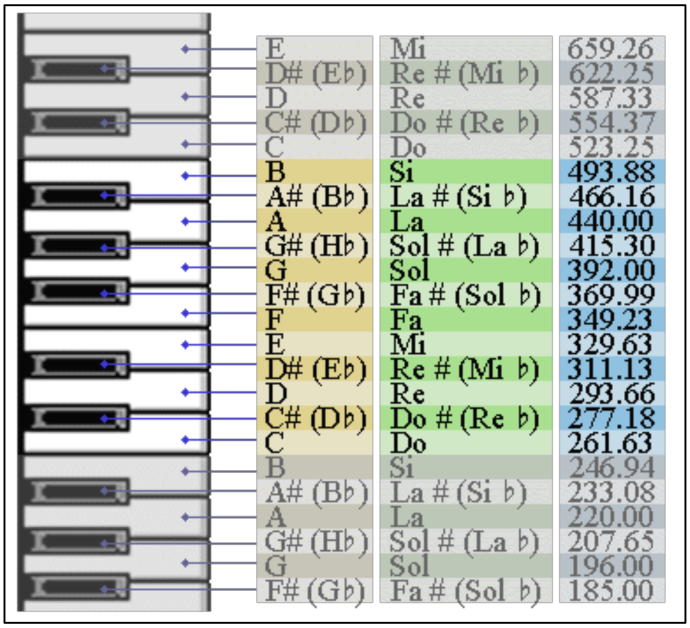

Figure 9. *Correspondences between the musical notes and frequencies.*

A function called *translator* computes the translation between the vector obtained in the previous step, which consists in a String with all the information from the music piece as to the notes, the scale, etc. In order to play the piece, we need to translate these notes into a vector with the corresponding frequencies so the software can read and play them.

As the database used is quite simple, the translation can be computed just with an if else or switch case for each of the possible notes.

In the *playing* function, the sampling frequency is established and the same maximum time of playing is set to every single note to be played. The amplitude is also set to a minimum of 2. Then, the software reads the translated vector of frequencies with a for loop in order to play the music piece.

The main aim of the Project is to compute a GUI so the software is user-friendly, and the user can choose an input music sheet to play. That is the reason why at the end a *MusicReader* function was computed so everyone can easily try and test the software.

This function allows the user to choose the musical piece he/she wants the software to read accessing to the files explorer. Once the file is chosen, the musical sheet is shown in the figure and it shows the notes down the image.

Finally, the software allows the user to play the song to see if it is correctly translated and lets the user save the audio file if wanted.

4. RESULTS

Once all the software is computed, and the GUI is completed, I checked the correct functionality of the whole project. In the following pictures the GUI is shown and the whole function and test of the software is explained.

First, the main of the GUI is shown, where all the possibilities are shown in different pushbuttons. The user can choose the image in a .jpg format, then show the chords of the music piece to play it later. The software also allows the user to save the music piece in a .wav file and it has a Help button in case the user need some help while using the program.

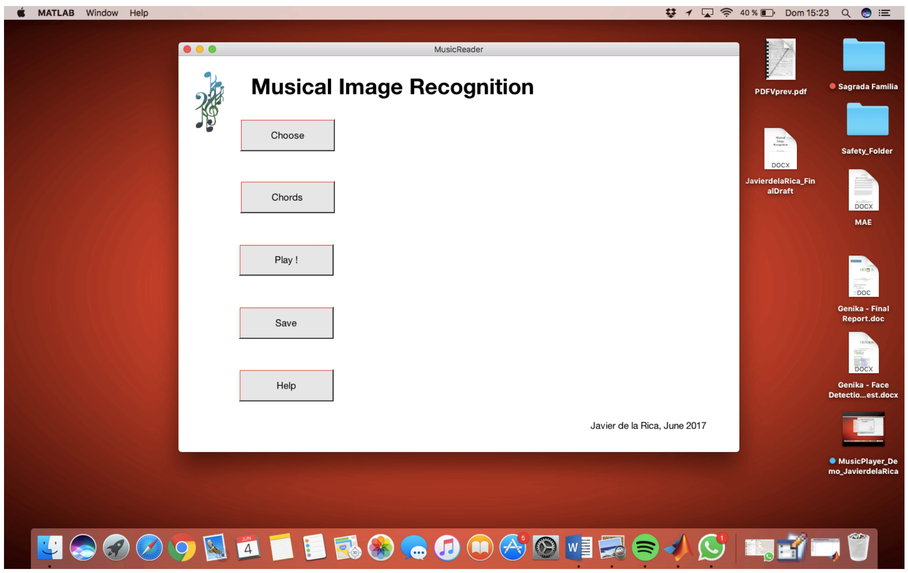

Figure 10. *Main of the Music Reader software.*

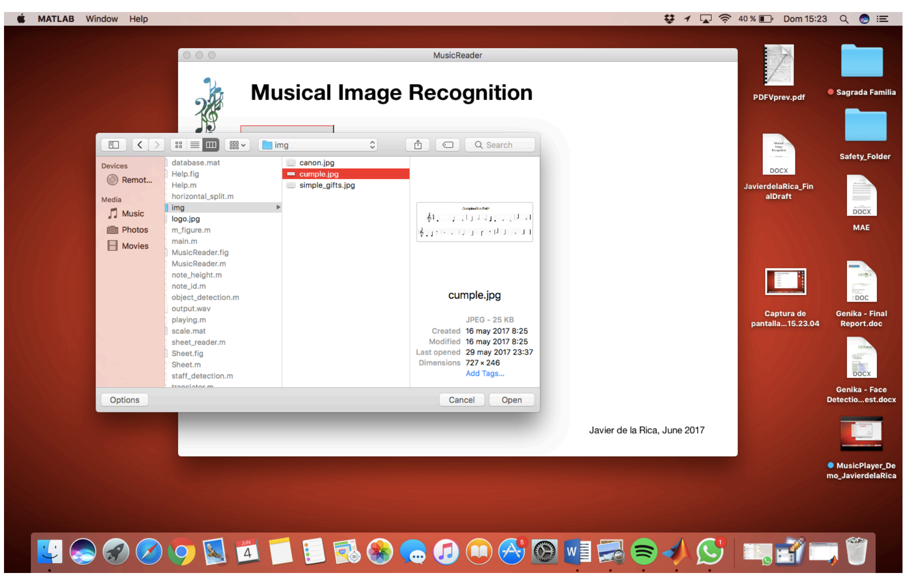

Figure 11. *Choosing the music piece to read.*

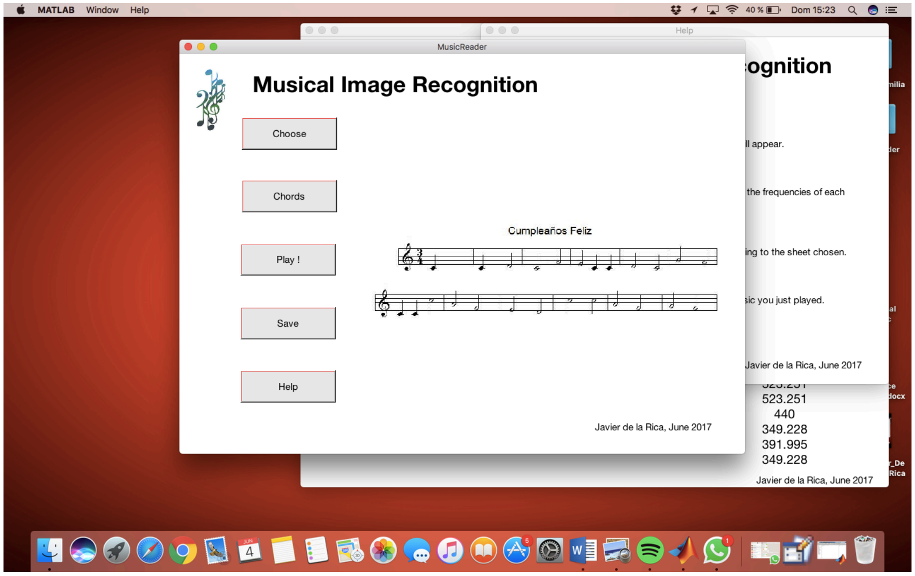

Figure 12. *Main of the Music Reader once the song is chosen.*

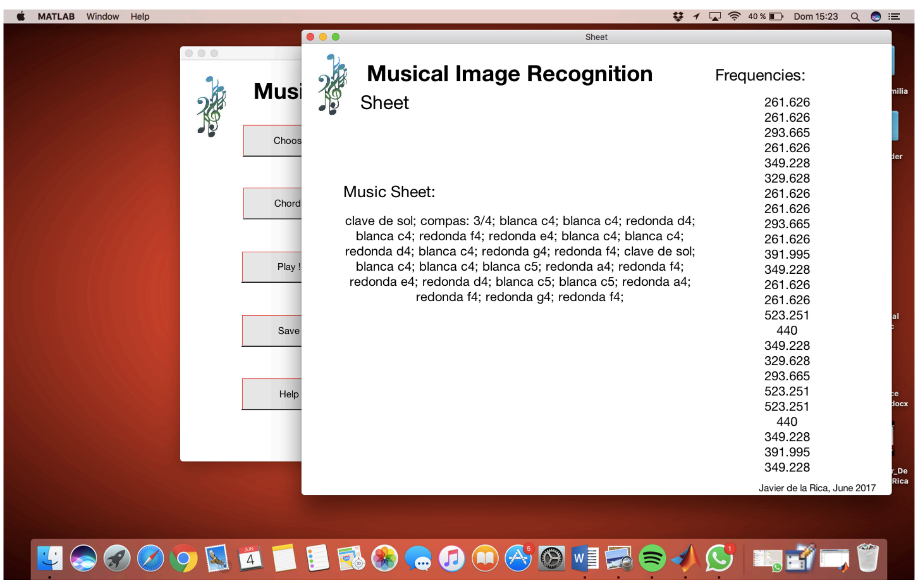

Figure 13. *Showing the Music Sheets and the corresponding frequencies.*

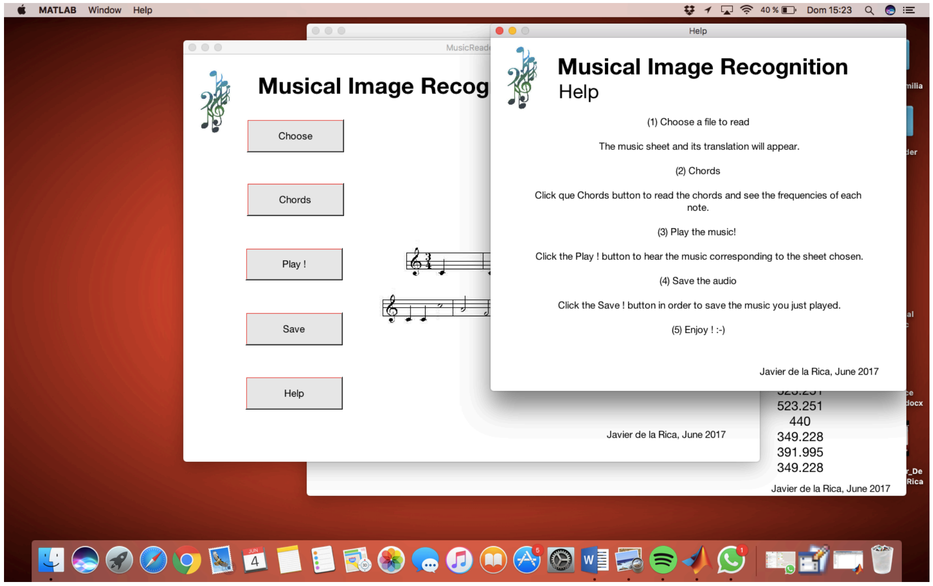

Figure 14. *Help button leads to the whole explanation of the software.*

5. CONCLUSIONS

This project established a Musical Image Reader, in which almost every different musical objects are taken into account.

The first step the software takes is the horizontal and vertical splits, in order to have the most important parts of the image in different images, so it is easier to compute and recognize the notes. The program also computes a compass segmentation in order to isolate the different notes from the whole musical piece.

The software computes then the note identification using the provided database and computing the pitch of the note depending on the staff lines corresponding to the compass in which the note is.

Finally, the result of all the segmentations and identifications lead to a single string in which all the information of the music piece is saved, as the key, the compasses and each one of the notes. The first attempt provides the notes in the Anglo-Saxon language, so later the notes have to be translated into frequencies in order to let the software play the different notes.

With the different notes translated into frequencies, the software plays the music piece computing a sinusoidal signal with all the amplitude, timing and frequencies set.

In order to make the software easy to use and user-friendly, a GUI is computed as shown above, with the different options and a help window in order to guide the user along the program.

The final result of the project is satisfactory, although at the moment it only reads basic music sheets as the *Happy Birthday* or the *Song of Joy*. It only reads musical pieces with the SOL key and separate notes. Ideally, the program can be extended so it can read and play any music sheet.

The only problem of the current version is that some of the pieces it can read, the software plays them in an awkward way, as the program has no information about the decay, amplitude, etc. The program has only the information of the SOL key and the frequencies of the corresponding frequencies, but is has no information about the melody it has to follow while playing the music piece.

6. REFERENCES

[1] Steve Dai, Chun-Wei Lee, Ye Tian, *Optical Music Recognition Implementation and Application,*

[2] MacMillan, Karl, Michael Droettboom, and Ichiro Fujinaga. "*Gamera: Optical Music Recognition in a New Shell.*" Web.

[3] P. Bellini, I. Bruno, P. Nesi, *Optical Music Sheet Segmentation*, International Conference on WEB Delivering of Music, 2001.

[4] Antonio José Moya, Fernando Pérez, *Optical Music Recognition,* Universidad de Granada, 2012.

[5] Andy Zeng, *Optical Music Recognition CS 194-26 Final Project Report,* University of California, Berkeley.

[6] Albert Ng, Asif Khan, *Automated Instructional Aid for Reading Sheet Music*, Standford University.