Sheet Vision
Optical Music Recognition with Template Matching

André Nascimento (73152) and João Verdasca (62355)
Departamento de Eletrónica,
Telecomunicações e Informática
Universidade de Aveiro, Portugal
Email: {andrenascimento, jfrverdasca}@ua.pt

***Abstract*—This article describes the challenges posed by optical music recognition by exploring the features of OpenCV and Template Matching method.**

First, the problem is described, then it is presented the steps that divide the recognition algorithm, starting by Camera Calibration, Perspective Transformation, Image Processing needed and finally the Music Character Recognition.

Afterwards it is shown the results obtained and an analysis.

The article concludes by discussing the results obtained by exploration of the template matching method usage.

***Index Terms*—Optical Music Recognition, Template Matching, Camera Calibration, Perspective Transform, Music Data Acquisition**

I. Introduction

Since always the human vision and the notion of an image was crucial for the human being knowledge, as one once said, *'a picture is worth a thousand words'*, it can tell different words to each one that analysis it and perform an introspective thought.

Like an image, a song can make people feel different type of ways and also remember things and moments lived when that music was used to be listen more often, it is not just the flow of the notes, it is a mixture of feelings and knowledge.

Thus, joining these both concepts of image and music, we could obtain a interesting and powerful combination.

So, in this article, it is shown an implementation of a musical sheet reader, being able from a picture, to read and distinguish song notes, the acquisition of feelings provided by the music it's up to the listener.

II. Implementation

The implementation of this musical reader is divided into four main groups, being the Camera Calibration, Perspective Transformation, Image Processing and Music Character Recognition. These groups are responsible for the image acquisition as well as the image aligning, rectification and preparing it, processing, towards the usage of template matching techniques that will be presented later.

A. Camera Calibration

The first and crucial step into this musical sheet recognition is to calibrate successfully the camera so that we can obtain the image and remove possible lens distortion made by the camera internal configuration.

For this process it was photographed a chess on a certain distance on every possible perspective in order to achieve a good calibration.

When getting a new capture, if distorted, like on figure 1, it is rectified as it is shown on image figure 2.

Fig. 1. Distorted Image Capture

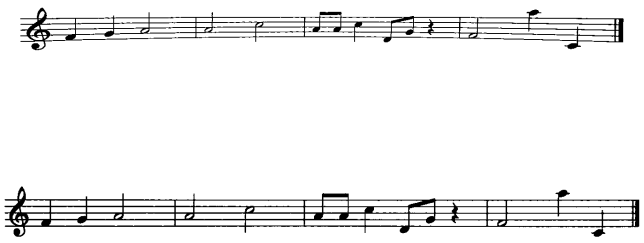

Fig. 2. Undistortion

B. Perspective Transformation

To complete the camera calibration and guarantee a well aligned image acquisition, it was used four chess as markers, and, by finding the center of each chess it is applied an perspective transformation by using the calculated matrix from *getPerspectiveTransform* and then using it on the *warpPerspective* method, observed on figure 3.

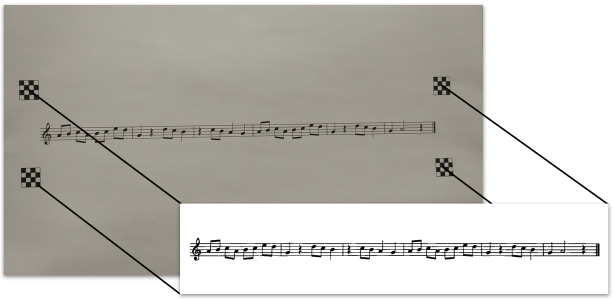

Fig. 3. Perspective Transformation Process

C. Image Processing

On to the image processing, it was applied a customized pre-processing before the usage of template matching.

There were used a couple of filters and morphological operations in order to have a noise free and a final image where each symbol to be detected could be well distinguished from each other, distinguish an half note from an quarter (tempos that have a similar musical character), thus, first it was applied a threshold to obtain an only black and white clean image, then it was applied, by using an *MORPH_ELLIPSE* kernel configuration, an erosion (*erode*) to eliminate the lines, a dilation (*dilate*) to reconstruct the image without the lines, then it was applied the Canny Edge detector to get the contours and finally another erosion to get a well defined contours instead of a very dilated one, so that in the end it was obtained an image to be processed where there are only known characters presented, similar to the ones used as templates on the template matching method.

An example of this image processing algorithm is shown on figure 4, where it was selected a zone of the musical sheet where there are similar characters, and on figure 5 it can be seen that there are only symbols to be detected, in this case an eight, quarter, an half note and a compass split, and still we have similar characters, but less similar than the first image, so that, template matching will perform better on these cases of similarity.

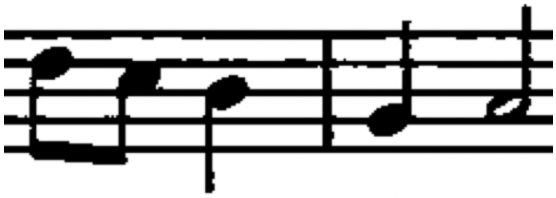

Fig. 4. Cropped Sheet example

III. Template Matching Algorithm for Music Character Recognition

On this section it will be presented the flow of the template matching algorithm and how it was managed to work in order to get, at every iteration, the best matching.

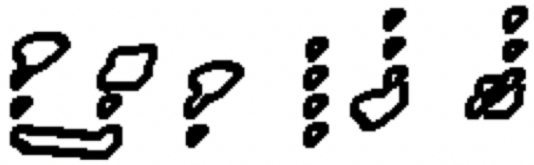

Fig. 5. Sheet Cropped Pre-Processed example

On the beginning, it was selected a determined number of notes to be recognized since that besides that the notes are simpler than recognizing handwritten characters, there are various number of different notes to be recognized but having a high level of similarity, making this process even harder, so, the following study focus only on to the basic notes and symbols like the quarter note, eight note, half note, pause, compass split, sheet end and the clave.

Collected the templates examples regarding the chosen characters to be recognized, we have the necessary data collected to start the matching process.

A. The Matching Process

Given an auxiliary post-processed image and the main one, this matching process will focus on the processed image and draw the detection on the main one.

Templates also pass through the pre-processment phase.

The process starts by matching all the templates with the processed image, and the template candidate with the higher confidence level is selected, and the result matrix given by that template helps to indentify the location of the match, thus, on the main image it is drawn the match with the assigned label, and on the auxiliary image the same match zone is painted has white, removing it for the next iteration.

The algorithm will keep on iterating over the processed image and removing the found and high classified templates till the image is only white and fail to match anything, figure 8.

The recognition process cannot be done in only one run, for example, if we know that there is t symbols to be recognized on the sheet, and we pick the t higher candidates and get their location, it sounds to be a good idea but on the other side, it isn't accurate at all.

At every new iteration the candidates do change due to that in every new match, the last best candidate match is removed, changing the values of the candidates at every iteration.

We took an example of the first matching round, on figure 6 and the last matching round, 7, where the xx axis are the symbols to be recognizing (there are seven distinct symbols) and the yy axis denotes the confidence value of each template match.

As it can be observed, if we take the higher template candidates on figure 6 they won't correspond to the symbols presented on figure 8 and candidates confidence level do change at every iteration because we are deleting the template

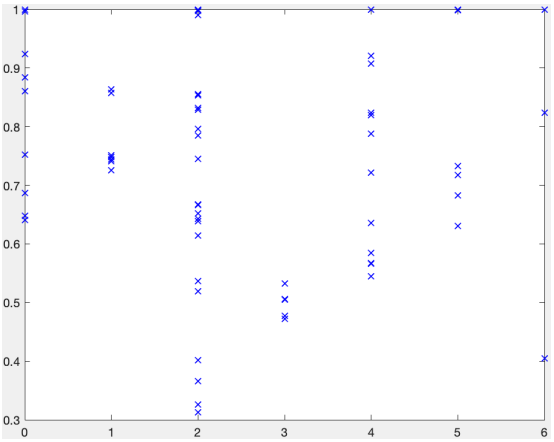

Fig. 6. Matching Candidates, first iteration

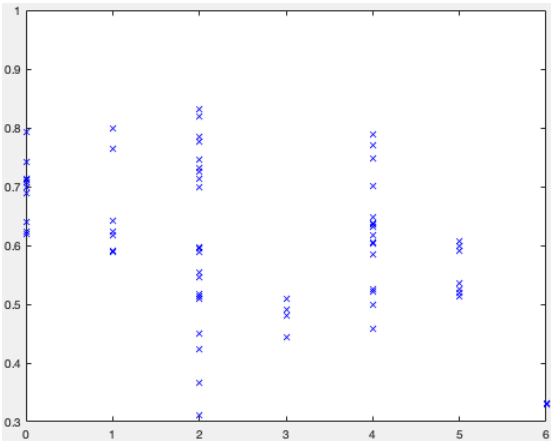

Fig. 7. Matching Candidates, last iteration

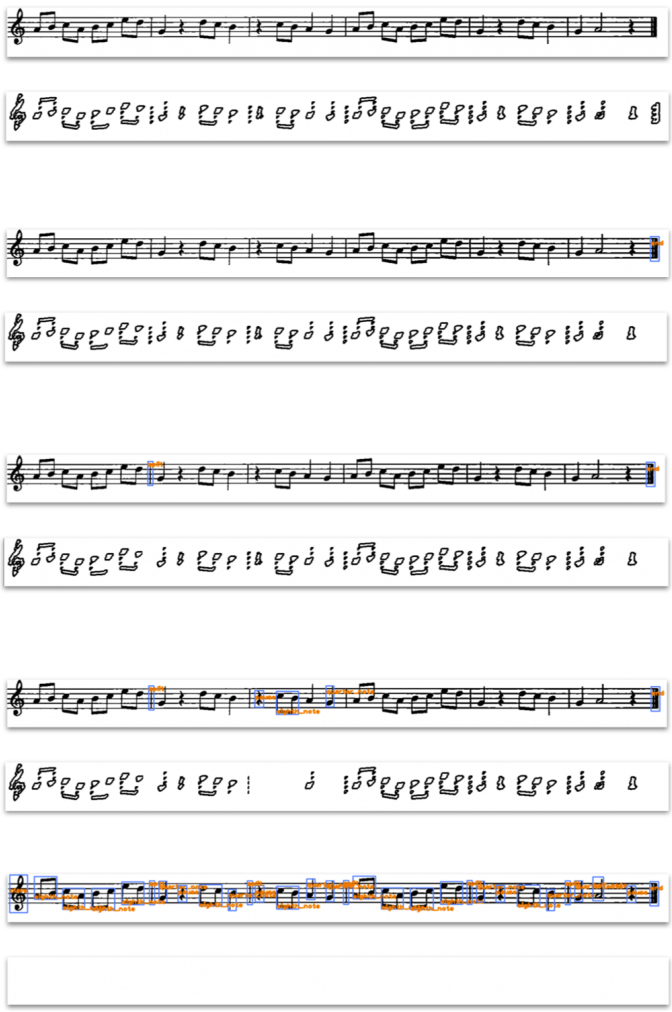

Fig. 8. Matching iterations example

that was detected, on sheet, after we get it right for the next iteration.

B. Tempo

On a musical sheet, the note tempo is denoted by its symbol. For this case, each template that represents a note, also does represent its tempo on the music.

Example of templates are shown on figure 9.

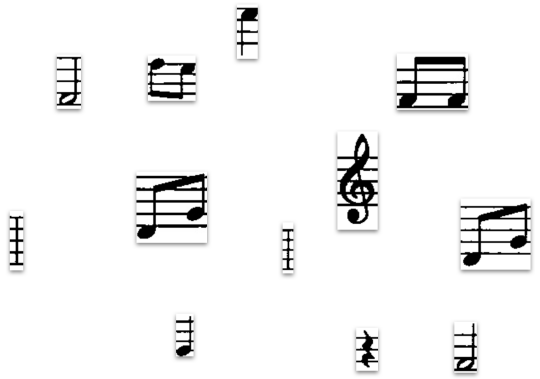

Fig. 9. Template examples

C. Tone

After we recognize the tempo, in other words, getting a good match symbol, we then get the location of the match, we get it and then we proceed to an analysis on way to get the tone of the note.

First, duo to the usage of markers to get a perspective transformation, the sheet is always on the same scale, thus, the symbols do have the same size on different sheets, the reason why we didn't choose to use scaling on the template matching, and the sheet lines are always divided with the same distance.

Assuming this, we started by dividing the processed image to get the location of each sheet line and also the lines that we denote as imaginary lines that are between each line, getting a total of third-teen lines, figure 10.

After knowing the position of each line, we will then analyze the recognized template, being the best candidate at

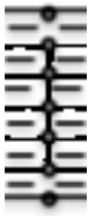

Fig. 10. Sheet Lines

that iteration, and we will look for ellipses forms and get the center of the detected circles.

For this process, figure 11, we opted once again to apply a processing sequence using dilate and erode, deleting the sheet lines, filling the empty spaces of ellipses with holes (half note symbol), and then another erosion to get a smaller ellipse where we could then find the perfect spot to draw a circle and get his center accurately.

Fig. 11. Ellipse form detection process

With the found circles center coordinates, figure 12, we can then find the nearest lines (the one with less absolute distance), predicting the tone(s) for the template that we are analyzing, figure 13.

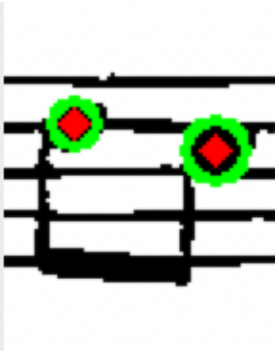

Fig. 12. Note detection example

IV. Results

To test our system, we've used the sheet presented on figure 8, the sheet isn't that simple and can mix the symbols we want to recognize and evaluate.

The following sheet examples on figures 14, 15, 16 and 17 were obtained by the same cellphone.

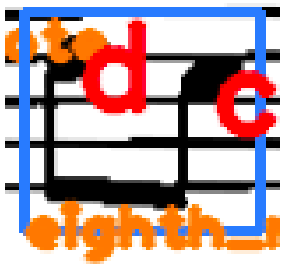

Fig. 13. Note detection example

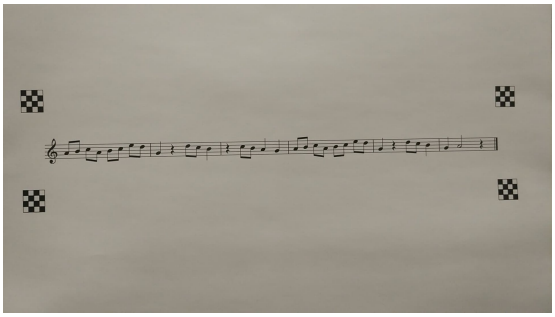

Fig. 14. First example

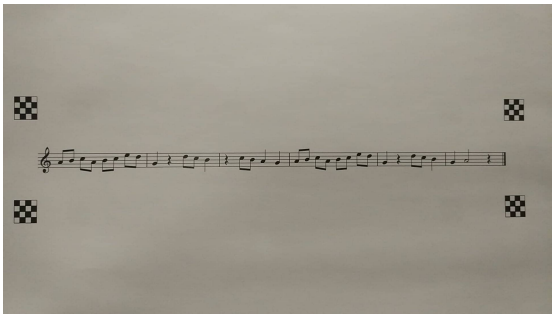

Fig. 15. Second example

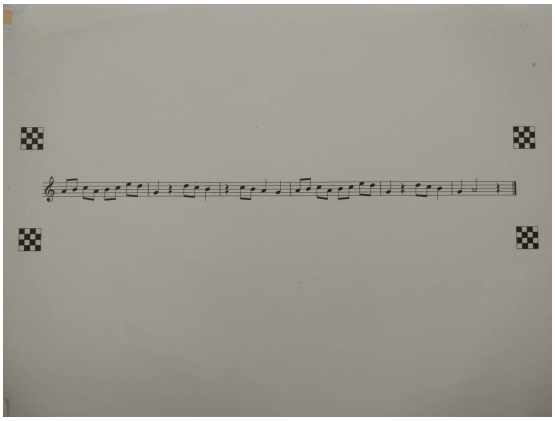

Fig. 16. Third example

The result of each example is pretty similar, and it is presented on figure 18.

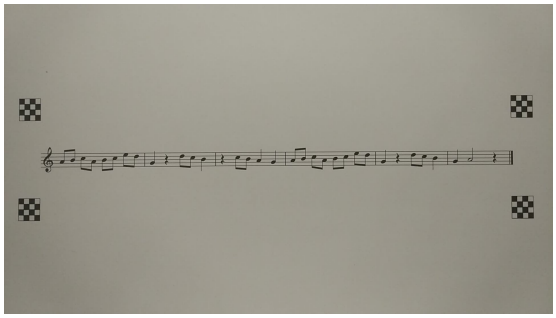

Fig. 17. Fourth example

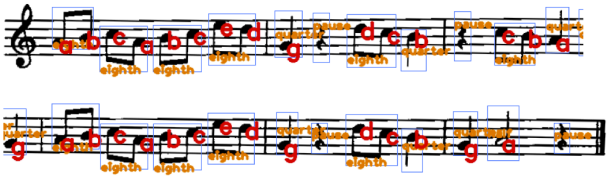

Fig. 18. Final Result

On this sheet, it was expected that the system would detect correctly 30 symbols, we've got the results shown on table I.

TABLE I
RESULTS ON SHEET PROCESSING

Examples	Correct	Incorrect	Precision
First	30	0	100%
Second	29	1	96.7%
Third	23	7	76.7%
Fourth	30	0	100%

By analyzing the images, the first, figure 14, and fourth example, figure 17, revealed better scores.

The behavior presented on the third example, figure 16, can be justified due to the proximity of the camera, being the picture that has the lowest zoom.

Also one must notice that all of these examples have a medium quality of light and the exposure to light can change the accuracy of the system, being this factor one of the limitations of template matching. This bad behavior can trigger problems like miss-detection and fail to identify notes, like on figure 19, where the system found a note *c* when shouldn't be any ellipse form / note.

V. CONCLUSIONS

To finalize this project, we can conclude that we have achieved the main objectives defined on the beginning phase.

We have studied and explored the methods offered by OpenCV, useful for image processing, pattern and character recognition and last but not least, methods for camera calibration that is crucial to this develop system to work, since that we do need an aligned, undistorted/rectified image to simplify our process and reduce the prediction/recognition error of each symbol.

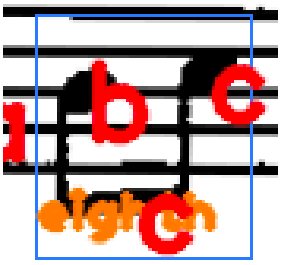

Fig. 19. Note detection error

During the development of this work, it was tried a classifier model to learn the symbols, however, the model wasn't learning the symbols and was decorating the symbols, so we found out that for simpler cases like this, exploring the functionalities of template matching would be the best approach for our system to work.

At the end, the results were satisfying and we can conclude the project with a good feeling of accomplishment and the feeling that we've used the material learned on class, we also got the feeling that the template matching approach made by us did fit well on this scenario.

For future work, the system could be expanded to recognize even more symbols and even handwritten ones. The system also could be even more optimized to reduce miss-detection errors due to the illumination exposure factor and for the lack of zoom.

REFERENCES

[1] OpenCV library, https://opencv.org/

[2] Deepa Berchmans, S S Kumar, *Optical character recognition: An overview and an insight), signals transported over an optic link*, https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=arnumber=6993174

[3] A. Rebelo, G. Capela, Jaime S. Cardoso, *Optical recognition of music symbols, A comparative study*, https://www.researchgate.net/publication/220163392_Optical_recognition_of_music_symbols_-_A_comparative_study

[4] Cascade Classification, https://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html