Linköpings universitet | Insitutionen för teknik och naturvetenskap
Advanced Image Processing TNM034 | Fall 2018

Optical Music Recognition (OMR)

Sarah Fosse, sarfo265
Joakim Konac, joako217
My Rydberg, myry009
Johanna Westberg, johwe476

Examiner: Daniel Nyström

LINKÖPINGS UNIVERSITET

Linköpings universitet
SE-581 83 Linköping
013–28 10 00, www.liu.se

Abstract

This is a project in the course Advanced Image Processing TNM034 at Linköping University. The course concentrate on pattern recognition in the context of optical reading of printed sheet music to implement software for Optical Music Recognition (OMR).

The project will be divided into sub problems, such as pre-processing, segmentation, matching objects, feature extraction and decision theory. Only a selected range of all musical notes are included in this task.

MATLAB is used to read the images and to implement the software to detect the notes. When the software has compiled the output consists of a string with all the notes described in letters.

Contents

1 Introduction

1.1 Background

This report is a part of the examination in the course TNM034 Advanced Image Processsing at Linköping University.

1.2 Aim

To create a software to read sheet music from photos and extract the notes found to create an output in the shape of a string with the all the notes in letters.

1.3 Limitations

The implementations should read the pitch of the notes, it will only classify notes from G1 to E4. It will also be limited to only read crotchets (quarter notes) and quavers (eighth notes). It will be assumed that the Treble Clef (G-clef) exists on every sheet music. Open notes and notes with more than one flag will be ignored. Also rest, dots, ties and other symbols other than Figure 1 will be ignored. The characteristics and appearance of the notes will be further explained under the section "Background Theory".

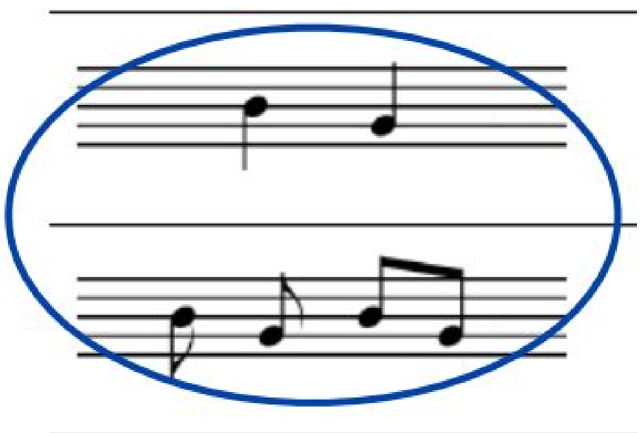


Figure 1: All the notes that will be included by the implementation.

2 Background Theory

2.1 Notes

A sheet of music is divided into separate staffs. One staff consists of five staff lines, a starting clef, bar lines and the notes and pauses. The treble clef combined with staff lines and the intervening spaces indicate the pitch of the notes (see Figure 2). The head of the note can rest upon a line or between two, indicating the specific pitch. Ledger lines are used to support the reader when determining the pitch when the note reaches over the staff.

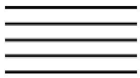
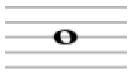
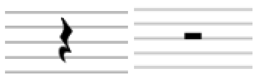
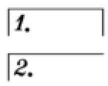
Symbol	Description
	**Staff and staff lines**, each staff has these five staff lines and intervening spaces.
	**Treble clef**, the clef defines the position of the first G after middle C (C2) to be at the second line from the bottom.
	**Semibreve** (whole note), a note that last one bar or four beats. The note has an open notehead and no stem.
	**Minim** (half note), a note that last half a bar or two beats. The note has an open notehead and a stem.
	**Crotchet** (quarter note), a note that last a quarter of a bar or 1 beat. The note has a closed notehead and a stem.
	**Quaver** (eighth note), a note that last a eighth of a bar or a half beat. The note has a closed notehead and one flag from the stem. Two quavers in succession are combined with one beam. For notes of shorter length the number of flags and beams increase.
	**Semiquaver** (sixteenth note), a note that last a sixteenth of a bar or a quarter beat. The note has a closed notehead and two flags, when combined with other semiquavers it has two beams.
	**Rest**, denotes a pause in the music.
	**Volta brackets**, used when a repeated passage is to be played with different endings on different playings.

Figure 2: Some symbols and properties of music notation.

2.2 Optical Music Recognition (OMR)

OMR is a computer system that can read printed music in shape of sheet music. The aim is that OMR will process the majority of the symbols on a sheet of music. It originates from Optical Character Recognition (OCR) which is the mechanical conversion from typed letters into machine text and has existed since the 50's.

There are however more challenges involved when reading sheet music compared to reading text. Sheet music is more complicated to categorize due to the different notes being more similar than different characters in a text, yet the small differences between them communicate large changes in the reading of the music. There are also properties found in music that is not found in printed text as seen in Figure 3. [1]

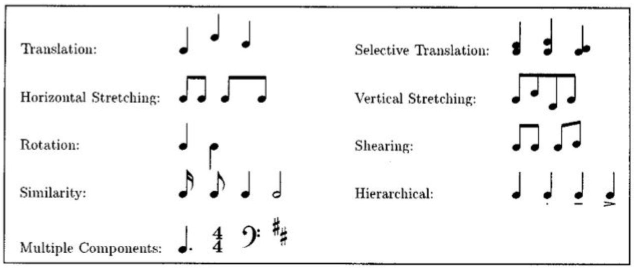


Figure 3: Properties of the notes.

2.3 Cross-correlation

Cross-correlation is a method to measue the similarity between the image we are processing and a template image. Matlab has an implemented function normxcorr2, which returns the resulting similarities between the input image and a template image.

2.4 Hough transform

The Hough transform computes the Standard Hough Transform on an binary image. It focus on finding lines and uses the parametric representation of a line described by Equation 1.

$$\rho = x * cos(\theta) + y * sin(\theta) \tag{1}$$

In MATLAB the Hough function calculates the distance from the from the origin to the line, theta is the angle in degrees between the x-axis and the vector. H is the parameter which corresponds to the values for the distance and the angle for rows and columns, see Figure 4 [3].

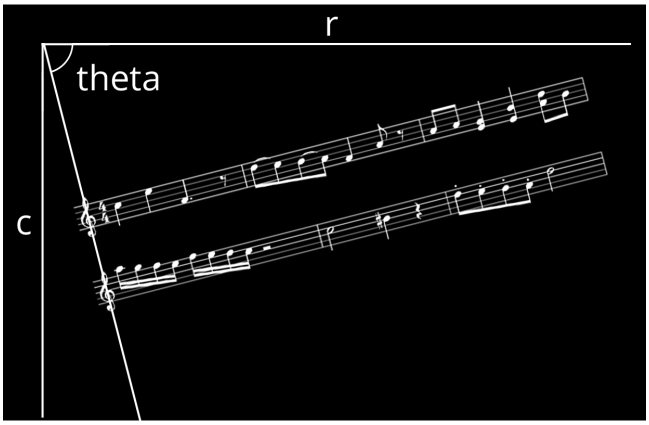


Figure 4: A rotated image with the angle theta.

2.5 Horizontal projection

A horizontal projection is the accumulated value of each row in an image. In a binary image in particular the value corresponds to the number of pixels with value. The projection can be used to find horizontal lines, these lines will create peaks in the histogram of the projection [4].

To find lines in the image, the lines have to be both straight and horizontal, otherwise the projection will not show any significant peaks as the line will not accumulate on the same row in the image. As seen in Figure 5 the image is at an small angle and the projection shows no clear peaks. However in Figure 6 the image has been rotated and the projection shows five distinct peaks.

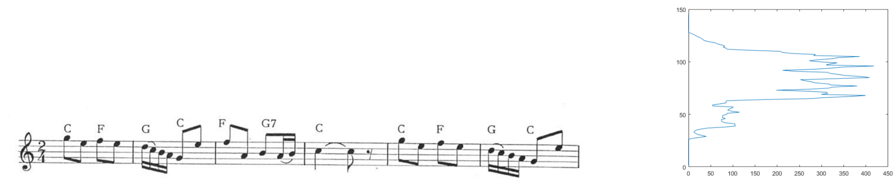


Figure 5: Staff with small rotation and horizontal projection.

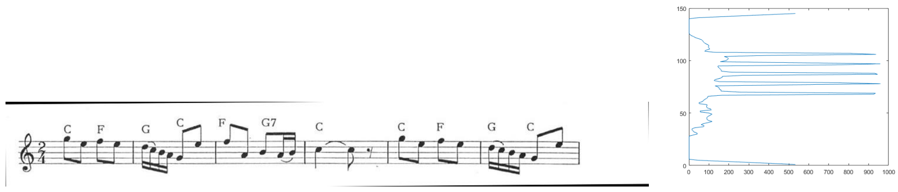


Figure 6: Straight staff and horizontal projection, distinct peaks for the lines.

2.6 Erode and Dilate

Erosion and dilation are used in this task when processing images. What dilation does is to expand an objects borders to give it a filled appearance (see Figure 7), thus making the object bigger and blend its components together. Erosion on the other hand reduces the object and makes it thinner, which is the complete opposite of dilation (see Figure 8).

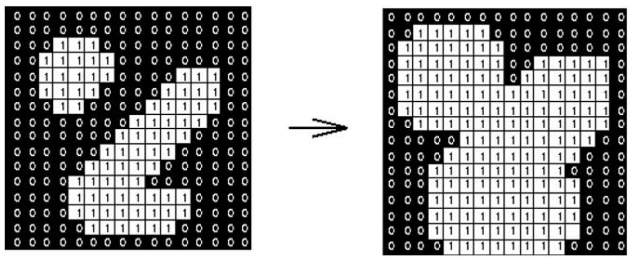


Figure 7: Dilation.

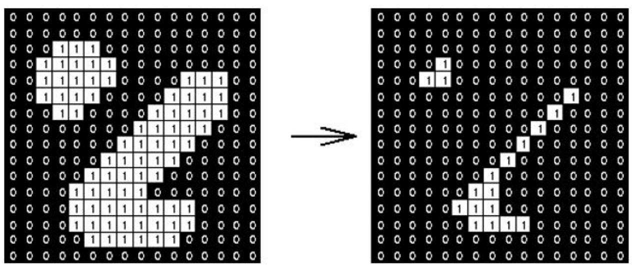


Figure 8: Erosion.

3 Method

3.1 Implementations and algorithms used

The project is divided into sub problems that are attacked in the following order so extract the notes found in a music sheet:

3.1.1 Preprocessing

All images are converted to grayscale images with double precision, this is to achieve good precision during all the processing and because the interesting part of the images, the sheet music, is already black and white.

3.1.2 Rotate image

This is the second most important step in the process after the preprocessing since all the following functions after this wont function correctly if the image is not completely straight. The following implementations are very sensitive so the rotating function must operate even on the smallest angles.

The function used is called *rotate.m* and takes a grayscale image and creates an binary inverted image to compute the angle of rotation. This image is used in the Hough transform to find lines in the image and the angles they are rotated. The theta resolution was set to 0.01 to get a detailed graph. The function will return multiple theta, some will correspond to the angle of the staffs and some will correspond to other lines in the image.

To determine which angles that are relevant and which ones that are diverging from the others a selection process is done. It will remove angles that diverge more than a varying limit from the mean angle of the combined angles. The limit is set to 60% of the distance between the largest angle and smallest angle. This will remove outliers, but include all angles that are relatively similar.

The final angle of rotation is calculated as the mean of the remaining angles. The original grayscale image is rotated with this angle and by using bicubic interpolation. This is the image the function returns.

Matlab functions used: imcomplement, im2bw, hough, houghpeaks, houghlines, imrotate.

3.1.3 Find staffs

The image is read in Matlab in a function called *findlines.m*. The lines that the function will be searching for are the ones seen in Figure 9.

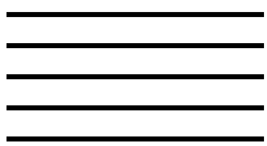


Figure 9: The staff consisting of five lines and four spaces.

A Horizontal projection is done on the image to find where the staff lines are which means that where the horizontal projection has its peaks there will be lines. All peaks are not relevant, therefore a threshold is chosen to only include peaks higher than half of the highest peak. This will only include relevant peaks and lines. The peaks also tell the coordinates for the lines, where the y-coordinates are the most relevant since they also tell the distance between the lines.

The y-coordinates can be used to measure the average space between the lines in the staff which is useful in many ways. It can be used to determine where the ledger lines are which is not seen in the sheet music but there can be notes in that height. The average space can also determine the coordinate system to see where the notes are positioned on the y-axis which tells what pitch the notes have. The average space is also a good way of telling the scale of an image and can be a resource when rescaling the image.

The function returns the positions of all the lines in the staffs, where the staff line (the top line in a staff) is in every staff on the y-axis and the average space between the staff lines. The positions of the staff lines are indicated by green lines as seen in Figure 10.

Figure 10: What it looks like in MATLAB when the staffs are detected.

Matlab functions used: imcomplement, im2bw, findpeaks.

3.1.4 Split image

The function *splitsystem.m* split a sheet music into separate images with only one staff each. It uses the positions of the first line in every staff and the average space between the lines to determine where the image should be cropped in order to get an image with the right size.

3.1.5 Remove lines

The function implemented in this case is called *removelines.m* and its purpose is to remove the staff lines that are found in the image and return a binary image in black and white without the lines, see Figure 11.

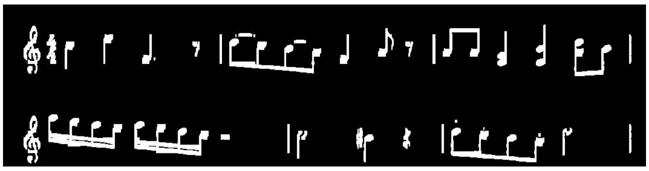


Figure 11: A binary image without the staff lines.

The input is an image which will be converted into a black and white image as in the previous functions. The lines are removed using first erosion and then dilation, by using a vertical matrix with the dimensions 5x1.

3.1.6 Find notes

When the staff has been removed, the next step in the process is to remove all the vertical lines, which results in an image with note heads and beams and to open the image with a template resembling a note head. With these two images, it is possible to pinpoint where the center of the noteheads are with the function **imfindcircles**. With this information, it is possible to scan the areas above and under the note head, to check if there is any beams present. This is done by measuring if there are pixel values present in the cropped subimages that equals to 1. If that is the case, there are white pixels present in the area above or under the note head. This will result in the current character being processed becoming lower cased, which in this task represents an eight note.

3.1.7 Define pitch

Defining the pitch of a note requires a function that measures the height of a note in a system, where the height is defined by a y-coordinate. The remaining objects are in a circular shape that resembles note heads. These objects (see figure 12) are mapped and the center point coordinates are registered. The height of the y-coordinate is then used to estimate the pitch. After finding the pitch the notes are saved as a string.

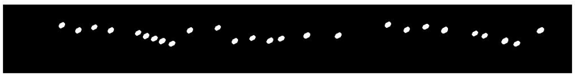


Figure 12: A binary image with only the note heads remaining.

4 Result

A combination of the implementations described above results in there being a string generated by the program. This string is an estimation of the notes being present in the images. For example, Figure 13 below results in this string being generated after being processed by the program: C2F2A2F2B2a2G2E2G2C2E2n C2F2A2B2a2f2E2G2C2E2n A2A2a2A2A2B2a2G2g2g2G2f2f2E2F2n A2A2a2A2A2B2A2G2G2F2E2n.

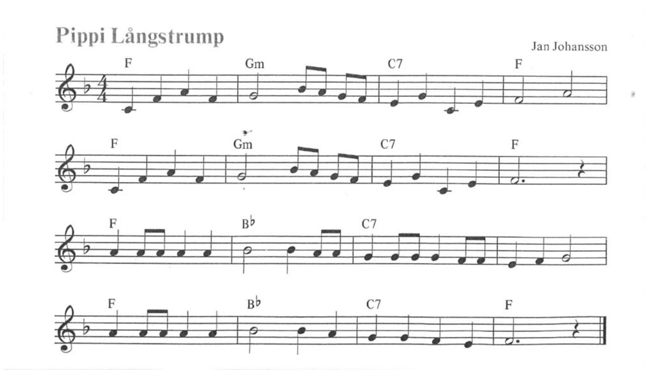


Figure 13: Notational sheet of the "Pippi Långstrump" song.

5 Discussion

Different qualities of the images of the sheet music affect the level of readability and the validity of the results in the end. Obstacles such as skew images, slightly rotated images, bad lightening in the photo are examples what can cause bad results.

A perfect image is a straight image in black and white captured by a scanner which gives the optimal quality in detecting notes in a sheet music. The detection of notes is simplified by preprocessing the image in the best possible beforehand to ensure the validity to be as high as possible.

The Hough transform is used to find straight lines of the staff in the image and the angel of which the sheet is rotated. A problem however with Hough transform is that it will not only find the long lines, but all straight lines. To combat this problem outlying angles have been removed before the rotation is applied. The method used to do this might allow angles close to the correct angle to influence the final rotation. A solution not tried could be to rotate the image multiple times with greater precision.

Another problem can be the layout of the sheet music. The staffs do not have a fixed length and can cause problems when detecting the lines because the length is unknown. It is however rather safe to assume the staffs to be the longest lines in the image, as most of them reach from one side of the paper to the next. The final staff is on the other hand different, as they can be shorter. In the function *findlines.m* a threshold for which lines to count is set to half of the longest line found, if the last staff is shorter than that the notes on that staff will not be read. Completely removing this threshold will result in undesirable lines such as beams and other horizontal features will be included and confuse algorithms later in the process. It is however rare that the last staff is much shorter than the rest and if lost easy to transcribe.

Something that has not been considered is how the existence of volta brackets will affect the horizontal projection and how to stop them from being classified as staff lines. All of the tried images have relatively easy music and does not contain any complex notation, therefore the code has not been modified to accommodate this. Other features might also result in high peaks in the horizontal projection and therefore influence the outcome.

Finding the note heads is most of the time not a problem, the problem is that other shapes can be read as a note heads. To make this problem as small as possible different thresholds are tested. The treble clef can sometimes be read as a note head which can result in a wrong note being included, however this is rare. After finding the note heads the pitch is decided by testing where on the y-axis the center of the note head is located, this function works very well and it is only in few cases the pitch is defined incorrectly.

6 Conclusion

Although a fairly accurate result can be generated, it is difficult to achieve perfect results when testing the program with a large number of input images. An OMR program will therefore have the risk of generating somewhat faulty results if the program is not optimized to a very high degree. The program developed for this task works fairly well and generates a result which is comparable to the original notational sheet.

References

[1] D. Bainbridge and T. Bell, *The Challenge of Optical Music Recognition*, Computers and the Humanities 35: 95–121, 2001

[2] A. Rebelo, G. Capela, Jaime S. Cardoso, *Optical recognition of music symbols A comparative study*, 13:19–31, Published online: 17 November 2009.

[3] MATLAB, *Hough Transform*, https://se.mathworks.com/help/images/ref/hough.html (Accessed November 2016)

[4] D. Nyström, *Advanced Image Processing TNM034, Lecture 2*, https://liuonline.sharepoint.com/sites/TNM034/TNM034-2016HT/CourseDocuments/Lectures/Lecture_2.pdf (Accessed November 2016)