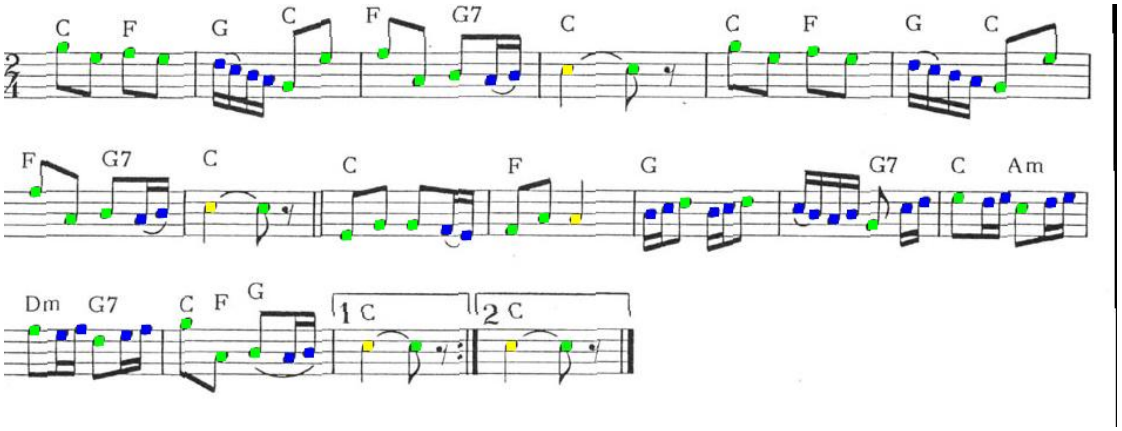


Optical Music Recognition

Project Assignment

for the course 2013/ht2
TNM034 Advanced Image Processing

Jessica Larsson *<jesbe726>*
Birgit Saalfeld *<birsa476>*

Linköpings universitet, Norrköping
December 15, 2013

Contents

List of Figures

1 Introduction

Optical Music Recognition (OMR) is an extension of the Optical Character Recognition field and aims at translating music symbols from printed music sheets into digital format. The first documented approach of an OMR system was made by Pruslin in the 1960s. His program was able to detect primarily music notes, without being able to recognize other important characters such as pauses and G-clefs. Since then OMR has been an international research effort and has become quite advanced, but with challenges yet to be solved. [1]

This project, made in the course tnm034 - Advanced Image Processing at Linköpings University, aims at implementing an OMR system much like the one Pruslin had accomp lished. Only a subset of all music symbols should be detected. The notes to be detected are quarter notes (crotchets) and eight notes (quavers) without taking any other characters and notes into account.

The program should take in an image normalized to the interval [0, 1] of captured sheet music and return as result a string of detected notes. The captured sheet music is either a scanned image or an image taken with a camera. The pitures taken with camera can be for example distorted or not uniformly illuminated. Therefore preprocessing is necessary for some pictures.

This paper presents our implementation for Optical Music Recognition. The tool used to implement the OMR program is MATLAB, a powerful instrument that already has all the basic morphological operations implemented, reducing the amount of handwritten code. The focus will therefore be based on the combination of known techniques, like morpholgical operations instead of how explaining the theory behind those algorithms.

According to software engineering standards all komponents should be reusable. Therefore the main process of detecting the notes is decoupled from all prior steps. Preprocessing leads to a normalized input image. This image is binarized, complemented, cutted and rotated. If the input image is camera taken, more preprocessing steps are necessary to derive such a normalized picture, see figure 1. During normalization, it is necessary to detect already the staff systems. With the staff information and the normalized image it is possible to create image variations. They will be used for detection of note heads and later on for detection of note values. Combining the obtained information it is possible to build the result string.

In the following chapters the shown substeps of figure 1 will be described in more detail. The order of presentation is following the execution order. As written above the software is divided in two parts, so in this documentation every chapter deals with one of those parts.

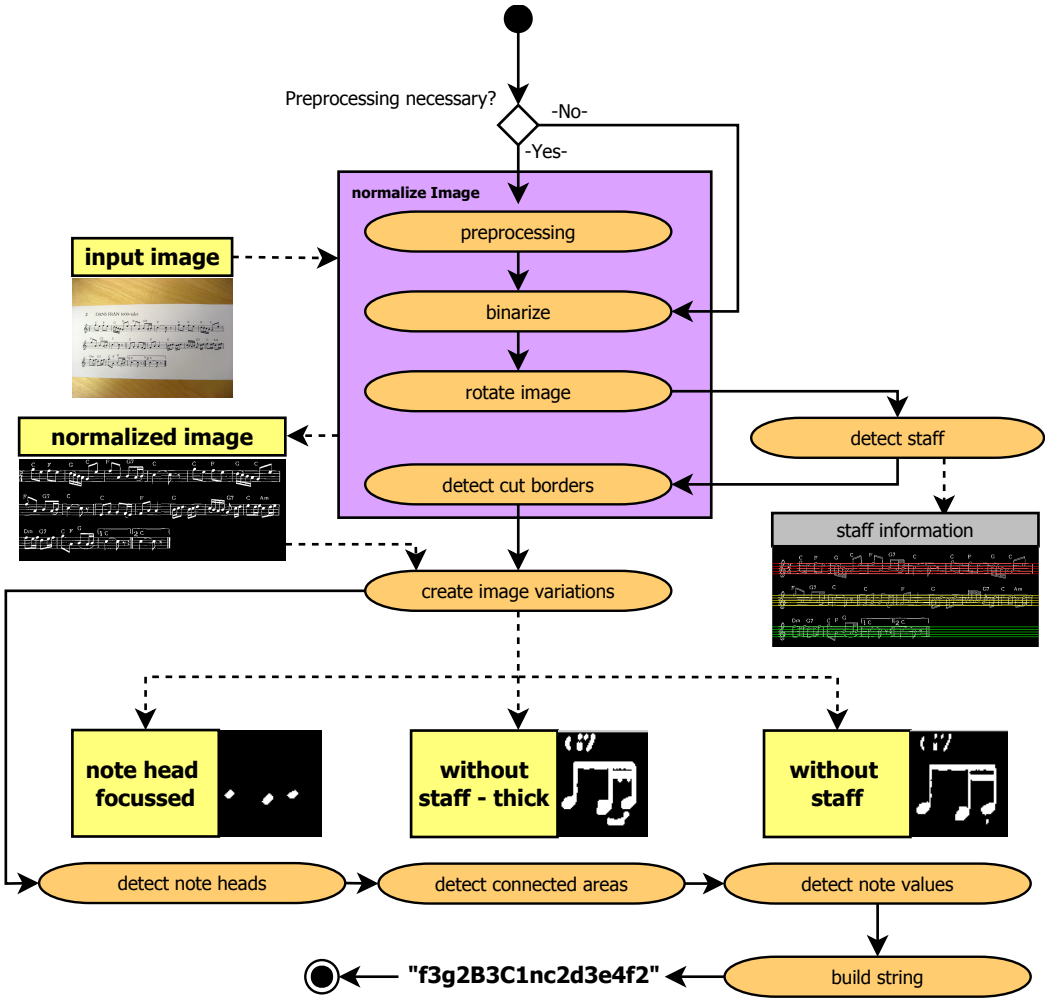


Figure 1: Simplified flow chart diagram

2 Normalize Images

There are two types of input images that schould be evaluated with this software. One type are the scanned images without background information. More challenging are the pictures taken with camera. They introduces problems like distortion, out of focus, nonuniform illumitation or note detection disturbing background. Due to this reasons the software is devided into two parts. The main part note detection from scanned images. To extend this, the second part deals with preprocessing in order to convert the camera taken picture into a scanned liked picture.

The software developed for the Advanced Image Processing course should be able to detect notes. To reduce the complexity of the task, limitations where introduced for this project:

- All music sheets are computer generated and not handwritten
- Detect only quaters and crotches
- Always assume G-clefs
- The rotated image never exceeds ± 90 degrees
- All images taken with the camera are taken in front of the same background

2.1 Preprocessing

Prepocessing is necessary for images with poor quality or pictures taken with a camera, because they can contain a lof of additional information, like backgound or they can be distorted.

To handle the table in the background it is useful to focus on the blue channel of the image. If this channel differs a lot between the center and the sides of the input image there is with high probability a table on the image. With this blue channel information the table can get cutted away.

This code to undo the distortion is not implemented yet. One idea how to handle distortion is use of vandermonde matrix. This algorithms takes certain positions form the distorted images. Additionaly it needs the information where these points are located in the non distorted image. Then it calculates the transformation from one image into the other.

Furhtermore for following steps a global illumination has to be guaranteed. Global illumination can be reached by apply sliding maximum over the image, as shown in one of the course lectures.

For all scanned images the preprocessing should not change anything.

2.2 Rotate Image

Another important step for normalization is the rotation of the picture. Therefore the rotation angle has to be determined.

The rotation is achieved by first applying a Sobel operator to enhance all edges and then calculate the desired rotation degree, using the Hough transformation. From this hough transformation the 10 highest peaks are extracted and shown as blue lines in figure 2. There are even sometimes false positives. Those can be the borders of the sheet or ascending/descinding note rows, like the red marked in figure 2.

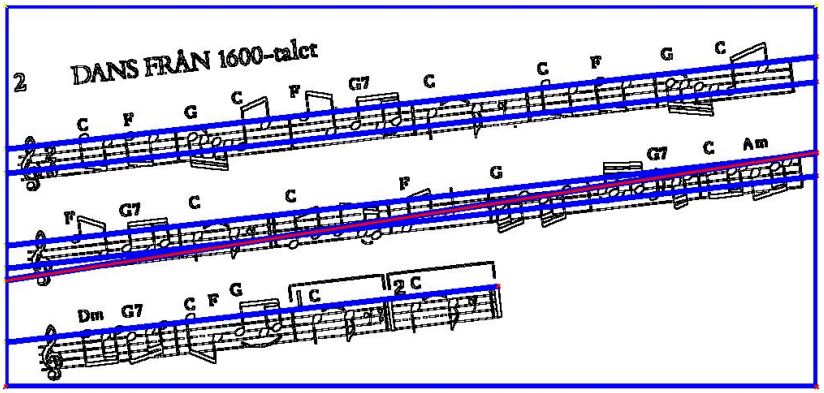


Figure 2: 10 Houghlines (blue) in detect rotation, including false positives, like the red markes one

To avoid taking those false positives into account when calculating the rotation degree, only the 5 most similar lines are selected. The selection throws away the outliers, like vertical borders. The similarity of the lines is determined by comparing the absoulte value of their slopes. Removing the outlies ensures that no picture borders and no ascending/descending note rows are taken into account. From those remaining (random staff) lines the mean value is found for the optimal rotation, as shown in figure 3.

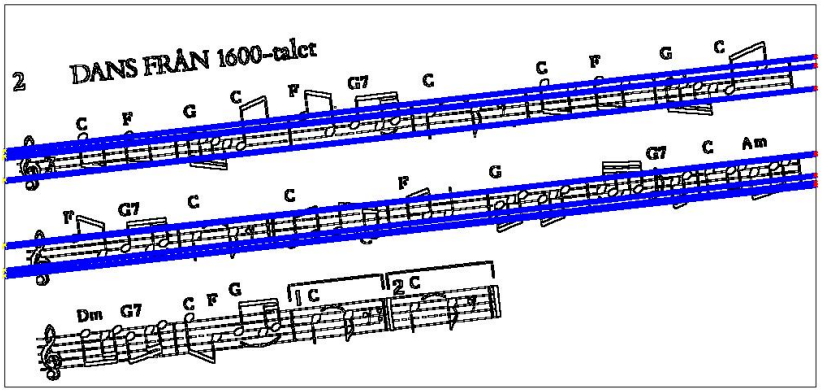


Figure 3: 5 remaining Houghlines (blue) in detect rotation

2.3 Detect staff system

The staff systems are an important source of information. They yield information about where note heads are expected to be found and which pitch the note is. One staff system consists of five staff lines, all with roughly the same thickness (*staffHeight*) and all with the same space between one another (*staffSpace*). [2]

To detect the positions of the staff systems a horizontal projection is made on the complemeted binarized image, see figure 4. In this rotated image the staff lines are horizon-

tal. By counting the white pixels per row each staff line delivers a high amount of white pixels. So every high peak in the horizontal projection is with a high probability a staff line.

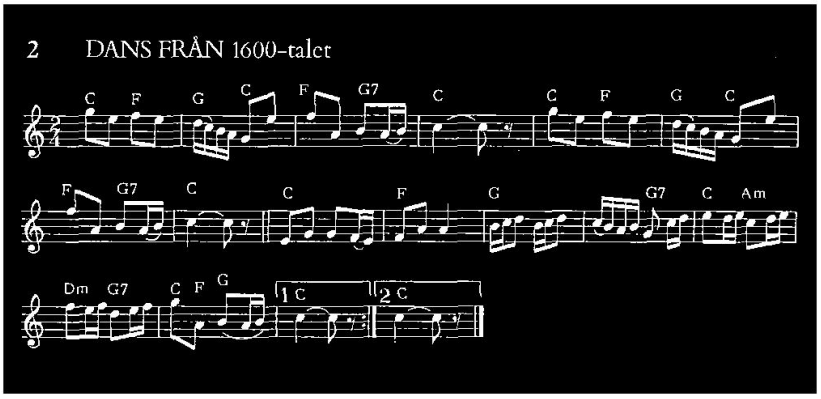


Figure 4: Binarized, complemented image

Due to the *staffHeight* is thicker than one pixel in most cases it is necessary to concentrate this cluster of peaks to one peak. This can be achieved using a low pass filter, see figure 5. After this filtering the peaks are lower, but then the peaks are more concentrated.

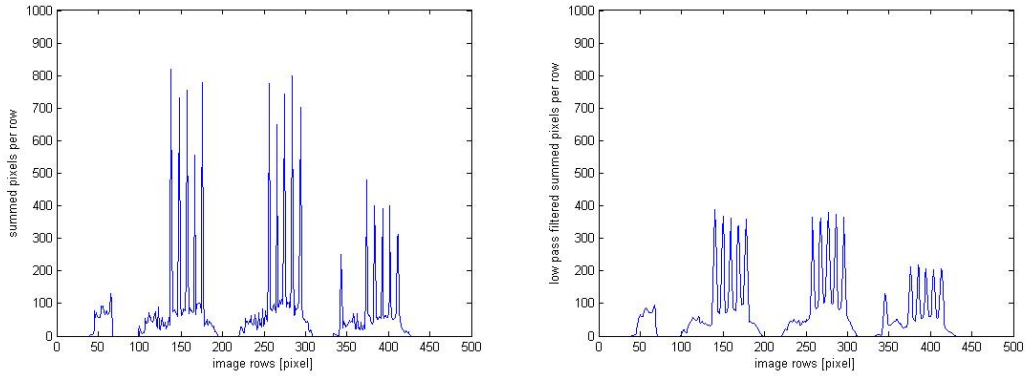


Figure 5: Horizontal projection unfiltered and filtered for finding staff lines

The histogramm of figure 6 shows the frequency of the normalized staff peak values. In the right half there are 10 peaks from the two long staff lines. In the middle part the short staff line can be seen. Everything below this values is not a staff line. For finding the threshold between staff lines and other lines Otsu's method [3] is used.

Extracting locations of the peaks higher than threshold from the projection returns the positions (row numbers) of all staff line centers.

To detect the correct threshold is not always straight forward. In figure 7(a) it is easy for humans to detect the peaks caused by staff lines. But the alogirhtm determines the threshold with figure 7(b) and there it is not as clear, which values belong to the

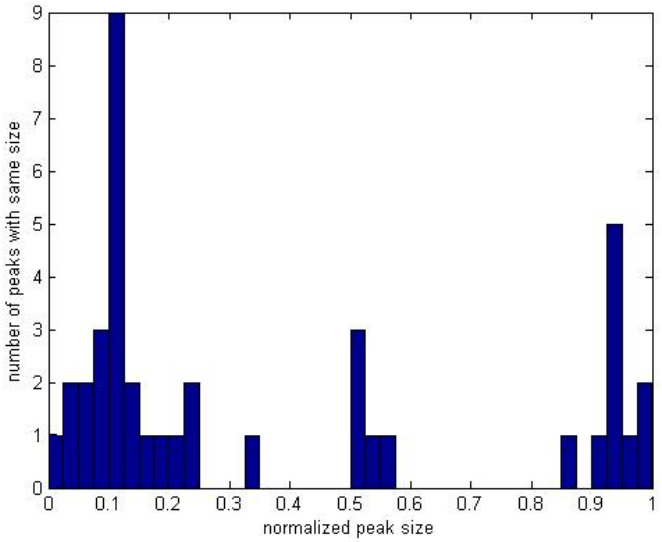


Figure 6: Frequency of the normalized staff peak values

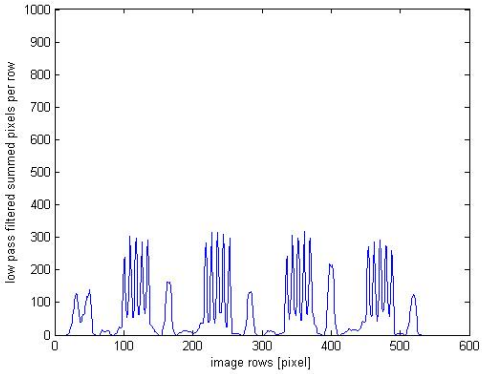
staff lines. Due to this there are sometimes to many lines marked as staff lines, see figure 7(c). Especially the music title or text to sing delivers false positives. To detect the false positives are several steps necessary. First k-means algorithm is applied to sort the lines into different clusters. Each cluster represents a staff system and has to consist out of exact 5 lines. The number of seeds for k-means is determined by number of lines found modulo 5. The seed positions are marked in gray in figure 7(d). Then k-means defines the clusters for staff lines, see figure 7(e). Now the false positives can be easily detected by the distance of the lines per each cluster. After throwing away the outliers the result is the detected staff lines, see figure 7(f).

With the knowledge of the staff system a big step in note detection is done. Now it is possible to determine *staffSpace* and *staffHeight*. Those values are needed in nearly every following step.

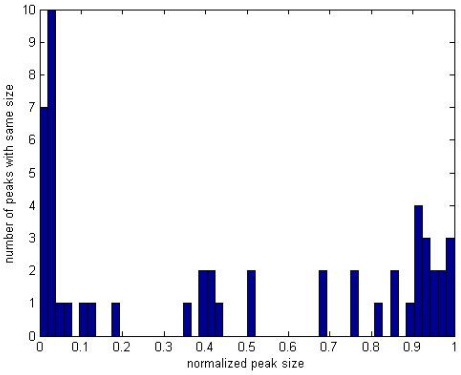
2.4 Detect cut borders

Only the part with notes is interesting for further steps. So everything else can be cutted away. This means for example title or background like table. The picture has 4 edges and this delivers 4 sides where a axis aligned cut is possible.

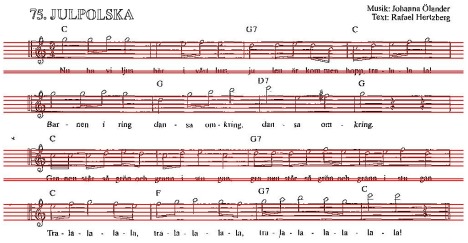
The rule to find the cutpositions above and below the staff system is quite easy. As the beams of the notes could go above and below their staff space, it is not possible to cut directly before the first and after the last staff system. Instead an offset has to be added. This offset is a little bit bigger than one staff system. So this offset is added to the end of the last staff system and substracted from the beginning of the first staff system to define the upper and the lower cut.



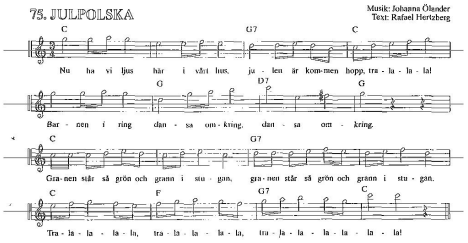
(a) Horizontal projection filtered for finding staff lines



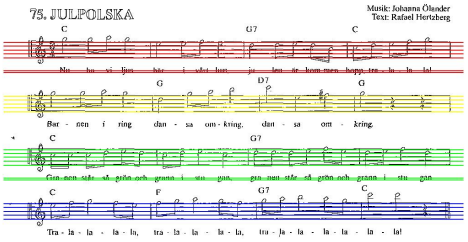
(b) Frequency of the normalized staff peak values



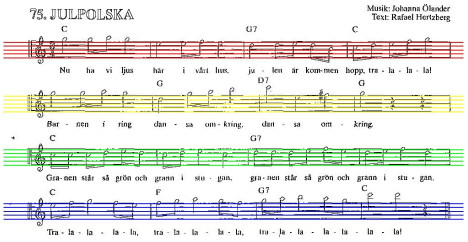
(c) Detected lines (red) with high probability to be a staff line



(d) Seed positions (gray) for k-means algorithm based on detected staff lines



(e) Detected staff lines in clusters, each color shows one cluster

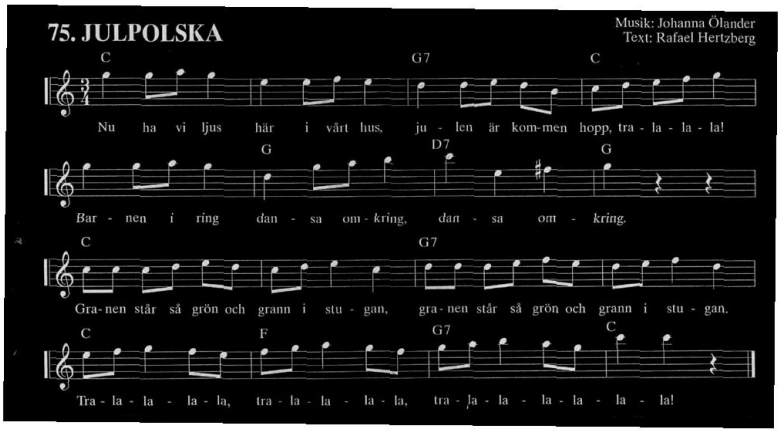


(f) Detected staff lines in clusters after removing outliers

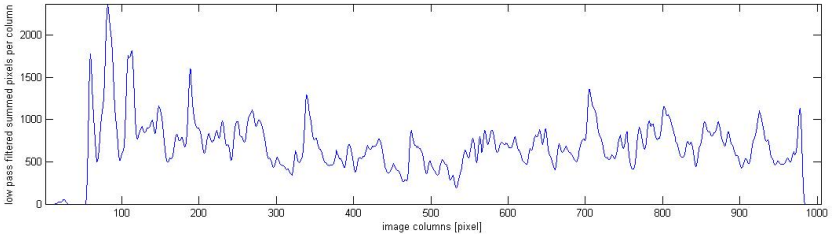
Figure 7: Detect staff lines with false positives due to text lines

Defining the left cut border is a bit more complicated. As given in the task description Treble Clef (G-clef) is always assumed. Therefore the G-clef should be removed. Otherwise it will be the reason for false positives in note detection later on. But how to detect the G-clef? In the vertical projection of the image the G-clef can be identified as highest peak in the left quad, see figure 8. Due to this reason the left cut is set to be at the position of the first minima after the first peak, because this is the position, where the G-clef ends. A cut on the right side was not necessary for this implementation.

Now the image is normalized and well prepared for the following steps. Figure 9 shows the image from figure 8 as normalized version.



(a) Binarized, complemented, rotated image



(b) Vertical filtered projection (of counting white pixels)

Figure 8: Image and vertical projection to define left cut border

3 Detect Notes

After the normalization process, the image is assumed to be a note sheet containting less unnecessary information than the original sheet. The normalized image is the image source used for detection. In the following chapters describes first the process of getting different image variations that provides different informations. After extracting these informations and combine them, it is possible to determine the note values and their pitch. Finally out of this data the string is derived.

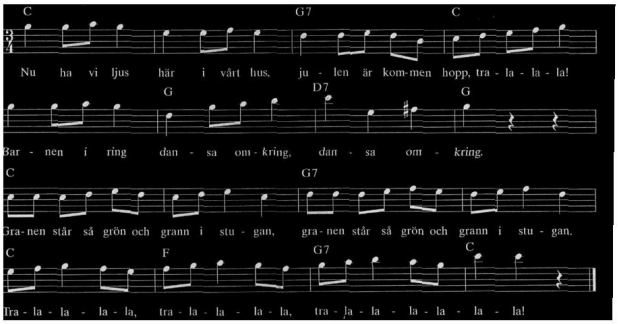


Figure 9: Example for normalized image

3.1 Create Image Variations

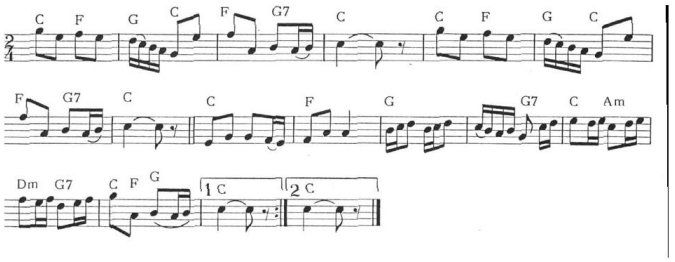
A set of different images of the sheet music is derived through various morphological operations. This set includes images with:

- note heads only
- no staff lines
- no staff lines, strenghtend all remaining areas

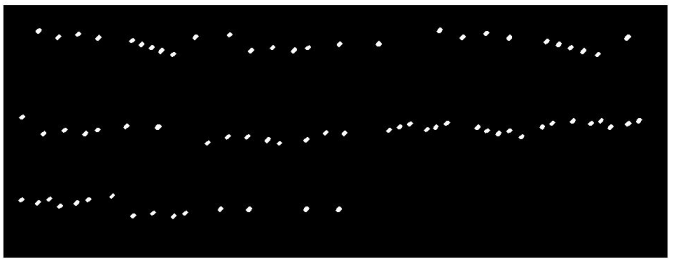
Note heads only To remove everything except note heads, the normalized image, compare figure 10(a), is opened with a disk element. The size of the disk element depends on the *staffSpace*, see figure 10(b).

No staff lines, strengthened all remaining areas The main goal with this variation is to get all the connected areas like quavers connected with a note beam. To get the best result, the normalized image can not be used directly. Instead the rotation and cut algorithms from the normalizing step are applied to the original image. Then this gray value image is opened with a vertical line structuring element to remove the staff lines. This result is then binarized after enhancing the vertical information and opened once more with the same element. Finally, the image is dilated with a cross shaped structuring element. This step ensures that note heads barely connected to their steam get connected, see figure 10(c).

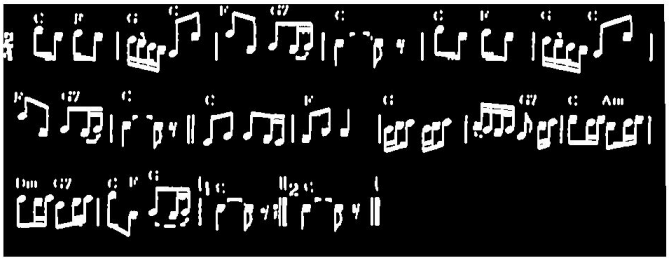
No staff lines Next, in order to get an image with no staff lines and only rare information, the normalized image is opened twice with a vertical line structuring element. This results in figure 10(d). As this image contains to much noise, an additional operation is needed. The modified images gets opened again, ensuring that all the areas remaining have at least a size of 2 times *staffSpace* pixels, compare figure 10(e).



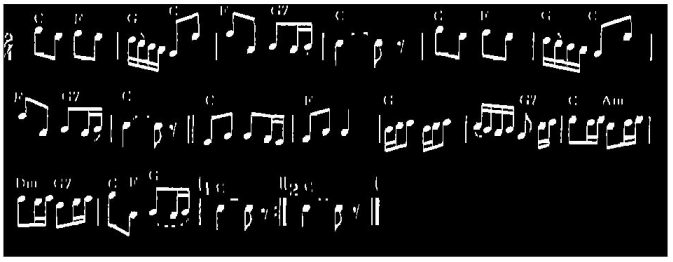
(a) Original image, rotated and cutted



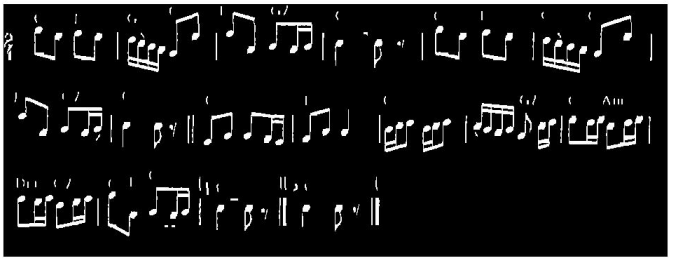
(b) Focused on note heads



(c) Removed staff and strengthened areas



(d) Removed staff



(e) Removed staff and noise

Figure 10: Different image variations

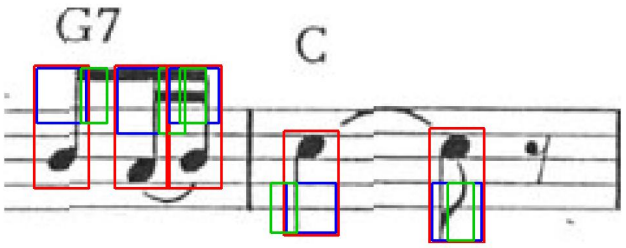


Figure 11: Bounding boxes with color coding, see next figures

3.2 Detect note heads

In this step the note head focused image variation is used. This image contains the note heads only. To locate the density positions of the note heads, the matlab functions bwlabel followed by regionsprops with the centroid property are used. To give the notes the correct order for building the result string the notes get sorted by their staff membership and x position.

3.3 Detect connected areas

After detection of all connected areas in the image with strengthened objects, the areas get checked for their size. Each note including its steam is expected to be at least 3 times *staffSpace* high. All areas that do not reach this minimum hight gets deleted. The remaining area boundung boxes are stored for the note value detection. In case of the note being a quaver, even the note beams are inside this axis aligned boxes.

3.4 Detect note values

This keyfunction combines a lot of previous calculated informations to finally detect the note values.

To show how the algorithm works, the processing for 5 notes will be explained in this chapter. Figure 11 shows the notes that are explored.

The general algorithm is for every note the same. For each note head position make a look up in the axis aligned boxes. If the note head position belongs to a real note, there is a box, which contains this position. Otherwise the note head is with a high probability not a note head. Continue with those note heads that were in a box.

For every note take the found containing box. This boxes represents the note and its surrounding. To be able to get the note and its surrounding information the image without staff information is accessed and the information inside the box is extracted.

Next all these image parts are normalized. This normalization means that all the notes can be handled in the same way. Therefore all image information above the center of the notehead is flipped downwards. This results in an image like figure 12(a).

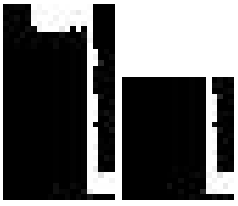
The note head is now cutted away so that it doesn't inlfuence the analysing of the beams, see figure 12(b). On this image a horizontal projections is made, see figure 12(c). This projection shows the beam position by it's maximum value.

In general, music note values are defined by the side regarding the steam that has the most information. For example in figure 13(b) and (c) the side that determines the note value is the right side. If there is no further information except the steam itself, the note is classified as a crotch, see figure 15(c).

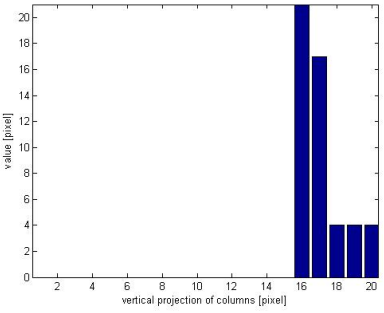
Once the side of the important information is known, only this side is further analysed. Therefore a new extraction is made. This new extraction has a fixed boxsize and is positioned next to the steam on the important side, see figure 12(d).

From this image a horizontal projection is made, see figure 12(e). To reduce noise again a low pass filter is applied, see figure 12(e). By counting the number of peaks in this last image it is possible, to define the value of the note. One beam means a quavers and more than one beam is not considered in this application.

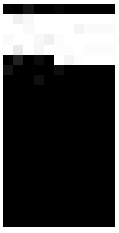
The following figures show the extracted images and plots for different cases.



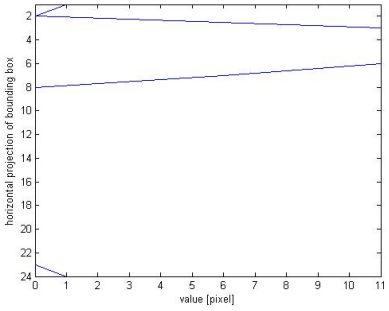
(a) red (b) blue



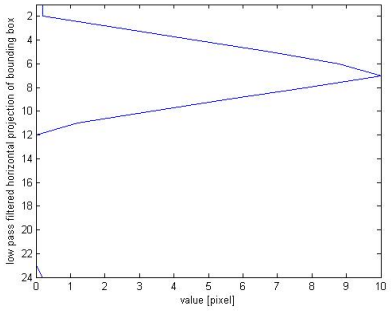
(c) vertical projection of columns



(d) green

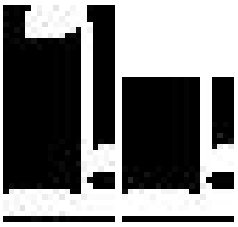


(e) horizontal projection of rows

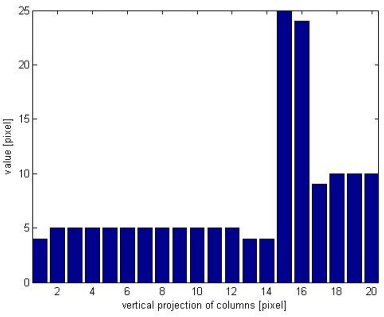


(f) filtered horizontal projection of rows

Figure 12: Processing of note detection first note

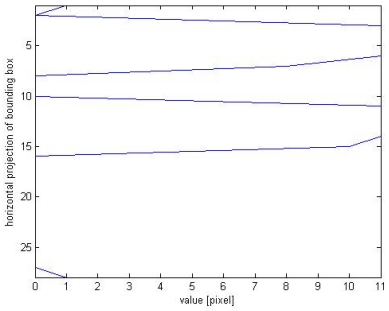


(a) red (b) blue

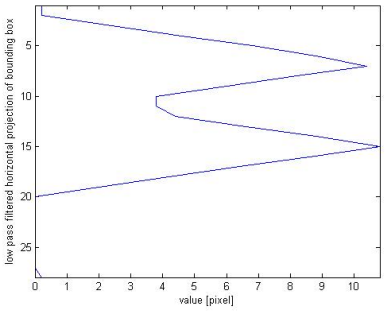


(c) vertical projection of columns

(d) green

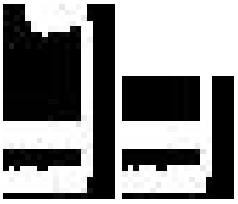


(e) horizontal projection of rows

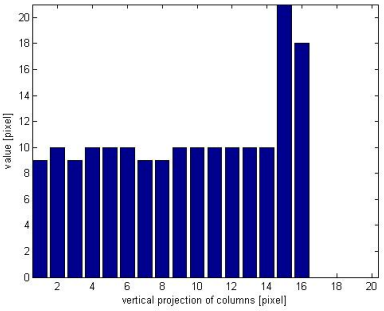


(f) filtered horizontal projection of rows

Figure 13: Processing of note detection second note

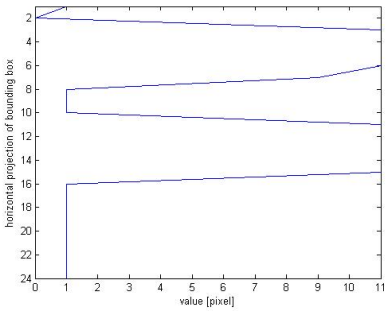


(a) red (b) blue

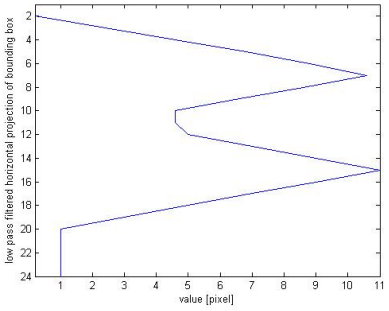


(c) vertical projection of columns

(d) green

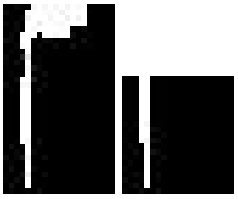


(e) horizontal projection of rows

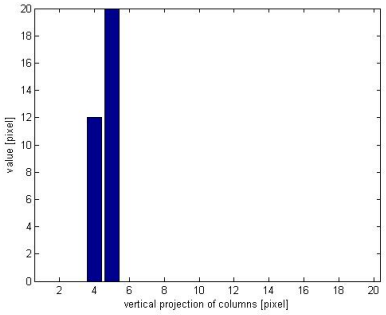


(f) filtered horizontal projection of rows

Figure 14: Processing of note detection third note

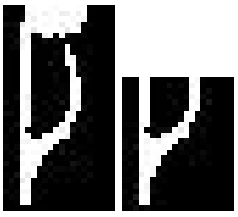


(a) red (b) blue

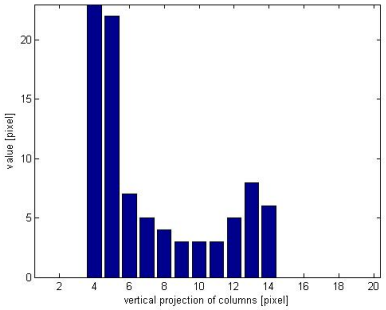


(c) vertical projection of columns

Figure 15: Processing of note detection fourth note

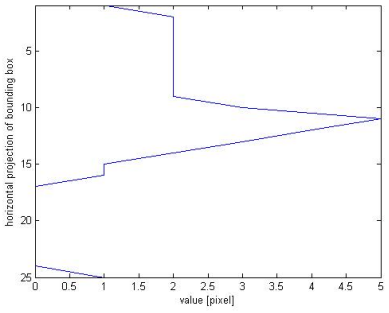


(a) red (b) blue

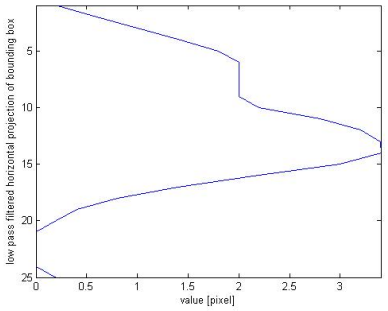


(c) vertical projection of columns

(d) green



(e) horizontal projection of rows



(f) filtered horizontal projection of rows

Figure 16: Processing of note detection fifth note

3.5 Build string and draw

Finally all of the information is evaluated to generate the string to give back.

The output should follow the following convention: (see even figure **??**)

- Uppercase letters for crotches
- Lowercase letters for quavers
- Numbers from 1 to 4 indicating the pitch of each individual note
- Multiple notes: Register each note separately, from top to bottom
- Line break (new staff) shall be encoded with the character n

With the knowledge of note value and the absolut positon of the note head in the staff system the note pitch can be detected straight forward.

To have not only the string feedback this application is able to print the detected notes by color coding on the images. This way supplies a much faster way to manually identify correct and incorrect detections. Figure 17 shows one of those images. The images for all test images can be found in the appendix.

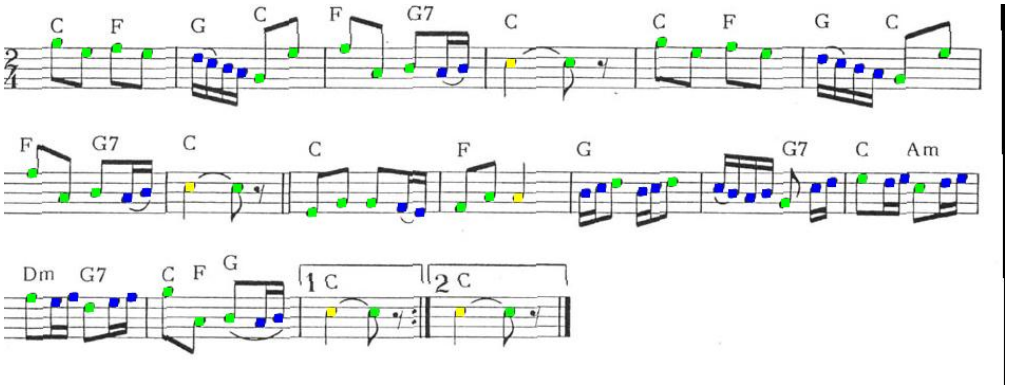


Figure 17: Recognoized notes drawn on original, cutted image. Color coding: blue-not considered, green-quavers, yellow-crotches

4 Alternative Approaches

There are different other approaches to determine notes. Especially when all notes and symboles have to be recognized.

One other possibilty is the application of template matching. But then the *staffSize* becomes more important. For every detection different templates have to be used regarding

the *staffSize*. In [4] an approach is described to adapt the patterns during matching time, to be more robust in symbol detection.

5 Conclusion, Result and Analysis

In this report the implementation of our OMR software is described. With a lot of examples the functionality is explained and the results for the test images are availabe in the appendix.

The achieved software works very well for scanned image as input. There are some minor mistakes, like detecting notes out of a break. This could be solved by using template matching. The template matching at that position would show, that the object is not a note.

If the quality of the scanned notes is poor and parts of the notes are missing, then the recognition fails. This is a general problem for optical recognition software. One idea how to solve this problem is the use of more intelligent algorithms. For example between 3 notes with beams, there is a high probability that there is a fourth note inside, when there is enough space, see fourth staff system in figure 25. This post processing can be very complex.

The step detect borders, described in chapter 2.4 is made for convenience purpose only. We make use of the preconditions allowing us to throw away unnecessary information, making the rest of the analysis a lot easier. The advantage of this approach is obvious: we do not get the false positives from the dot of the G clef, instead though there is a small chance that we do loose correct notes if they happen to be too close. This works really well for us, but it not suited for general music recognition.

The preprocessing works only rudimentary. There are possibilities to extend the preprocessing with the written stages.

All in all within this project work a software was developed which works realy well for the given images. Nevertheless software can always be improved.

References

[1] BAINBRIDGE, David ; BELL, Tim: The Challenge of Optical Music Recognition. In: *Computers and the Humanities* 35 (2001), Nr. 2, 95 – 121. http://dx.doi.org/10.1023/A:1002485918032, Abruf: 2009-05-15. – DOI 10.1023/A:1002485918032

[2] DALITZ, Christoph ; DROETTBOOM, Michael ; PRANZAS, Bastian ; FUJINAGA, Ichiro: A Comparative Study of Staff Removal Algorithms. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 30 (2008), Nr. 5, 753-766. http://dblp.uni-trier.de/db/journals/pami/pami30.html#DalitzDPF08

[3] MATHWORKS: *Documentation Center*. Version: 2013. http://www.mathworks.se/help/images/ref

[4] MCPHERSON, John R.: Introducing Feedback into an Optical Music Recognition System. In: *Proceedings of the Third International Conference on Music Information Retrieval*. Paris, France, October 13-17 2002, S. 259–260. – http://ismir2002.ismir.net/proceedings/03-SP01-2.pdf

Appendix

On the following sides are the results for the scanned test images.

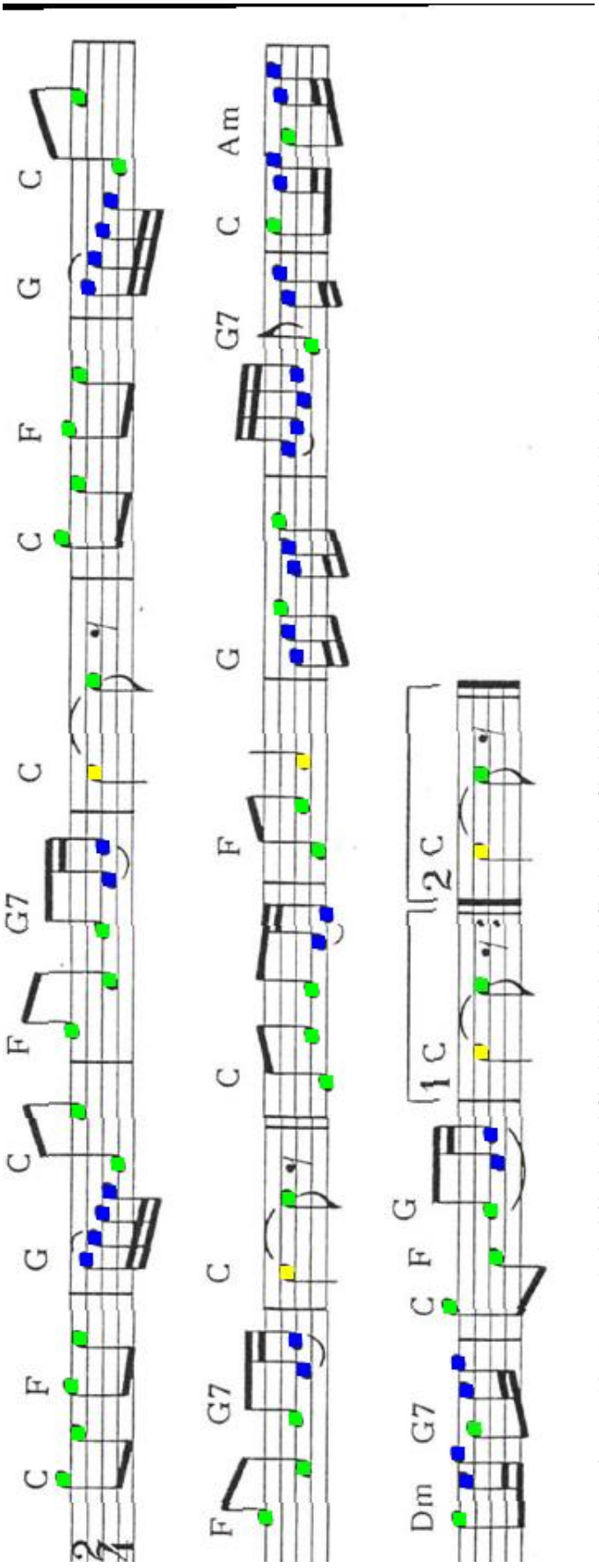


Figure 18: Dance ’g3e3f3e3g2e3f3a2b2C3c3g3e3f3e3g2e3nf3a2b2C3c3e2g2g2f2a2A2d3d3g2e3c3nf3d3g3a2b2C3c3C3c3’

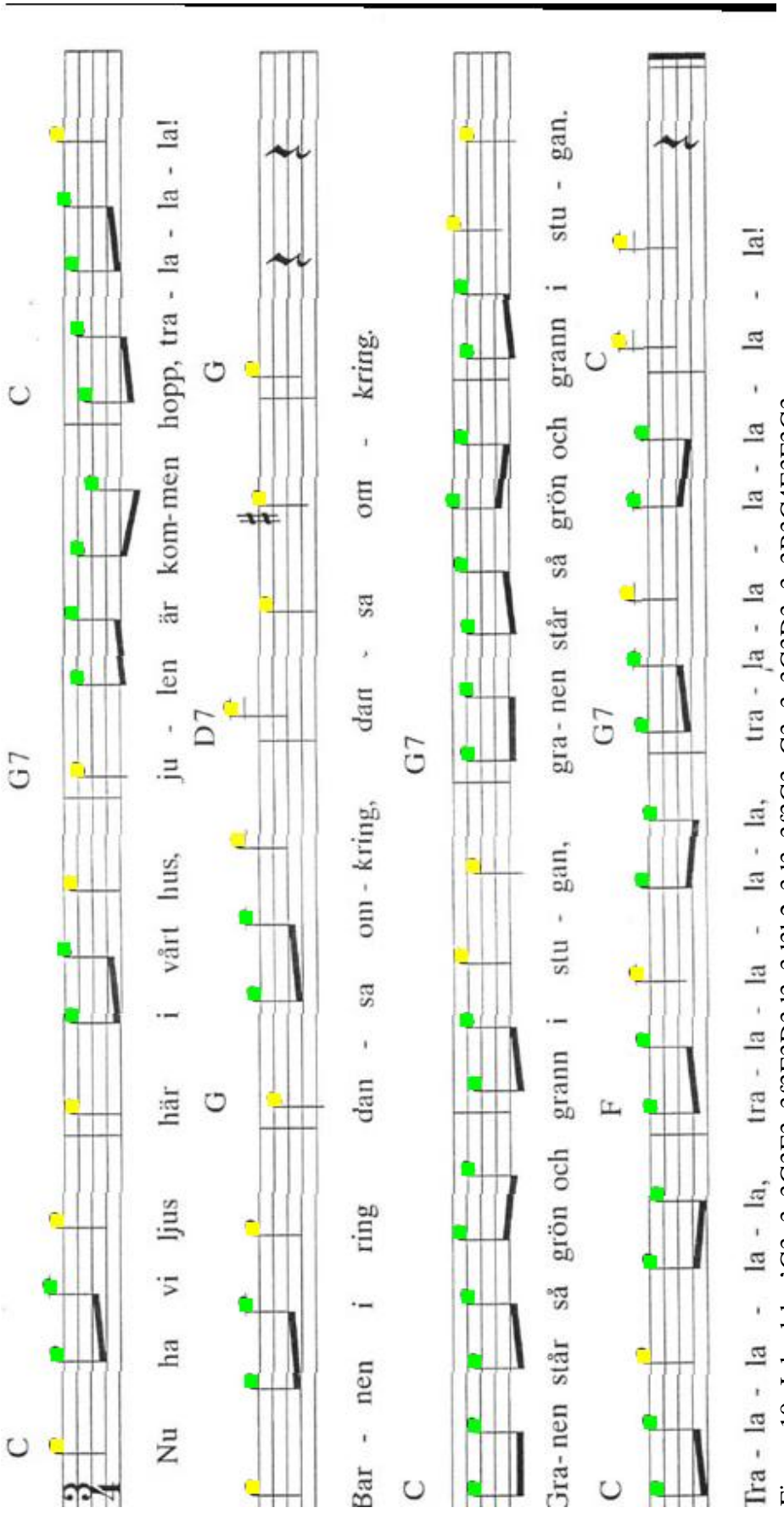


Figure 19: Julpolska 'G3g3a3G3E3e3f3E3D3d3e3d3b2c3d3e3f3G3nG3g3a3G3D3g3a3B3C4E3F3G3n c3c3c3d3e3d3c3d3E3C3d3d3d3e3f3e3d3e3F3D3ne3f3G3f3e3f3g3A3g3f3g3a3B3a3g3C4C4'

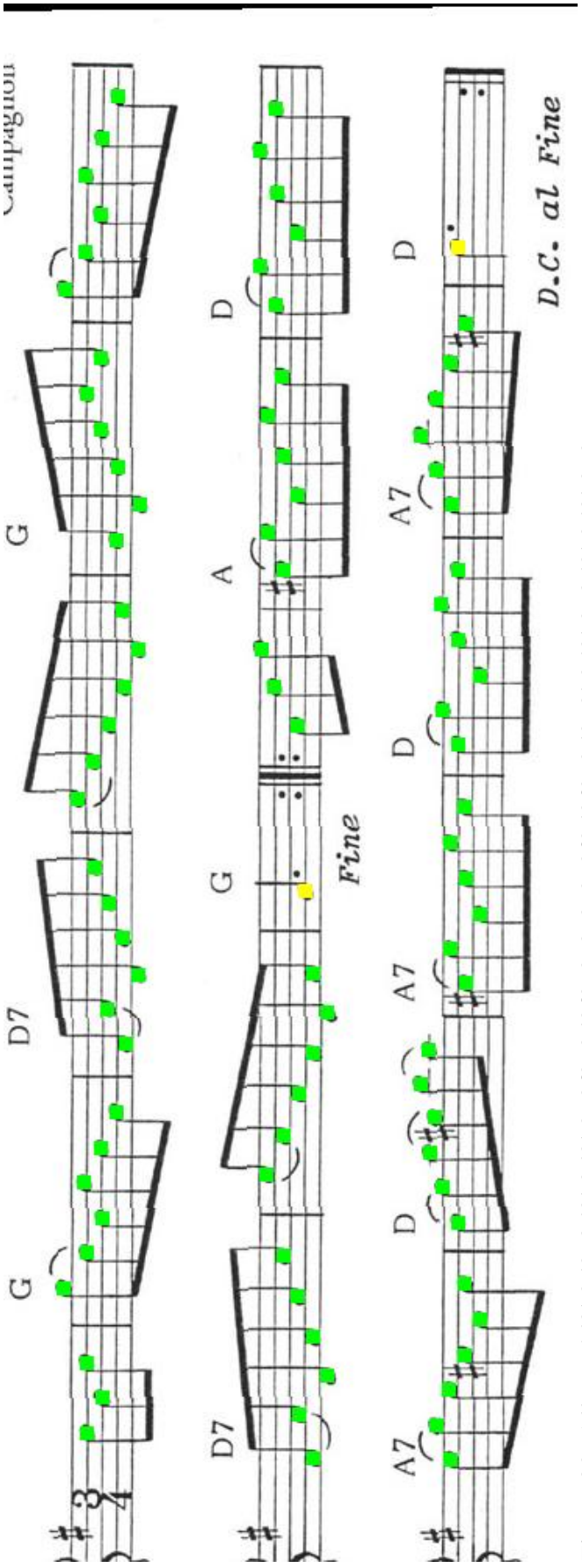


Figure 20: Allegro ’d3b2d3g3d3b2d3b2g2f2a2d2f2a2c3e3c3a2f2d2f2g2d2g2b2d3b2g3d3b2d3b2g2n
f2a2d2f2a2c3e3c3a2f2d2f2G2a2d3f3c3e3a2c3e3c3d3f3a2d3f3d3n
e3g3e3c3a2c3d3f3a3g3b3a3c3e3a2c3e3c3d3f3a2d3f3d3e3g3b3g3e3c3D3’

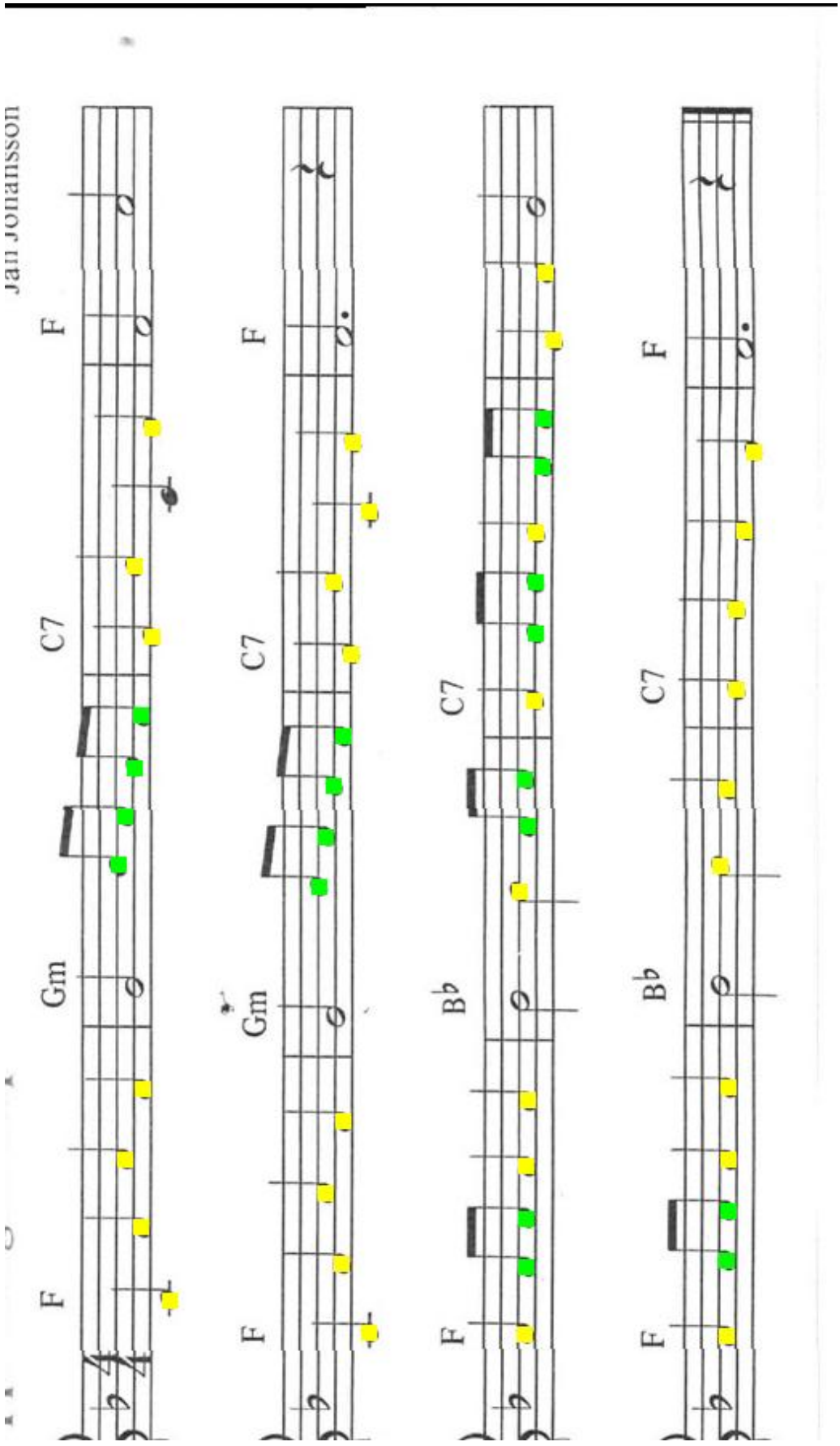


Figure 21: Pippi Langstrump ’C2F2A2F2b2a2g2f2E2G2E2nC2F2A2F2b2a2g2f2E2G2C2E2n A2a2a2A2A2B2a2a2G2g2g2G2f2f2E2F2nA2a2a2A2A2B2A2G2G2F2E2’

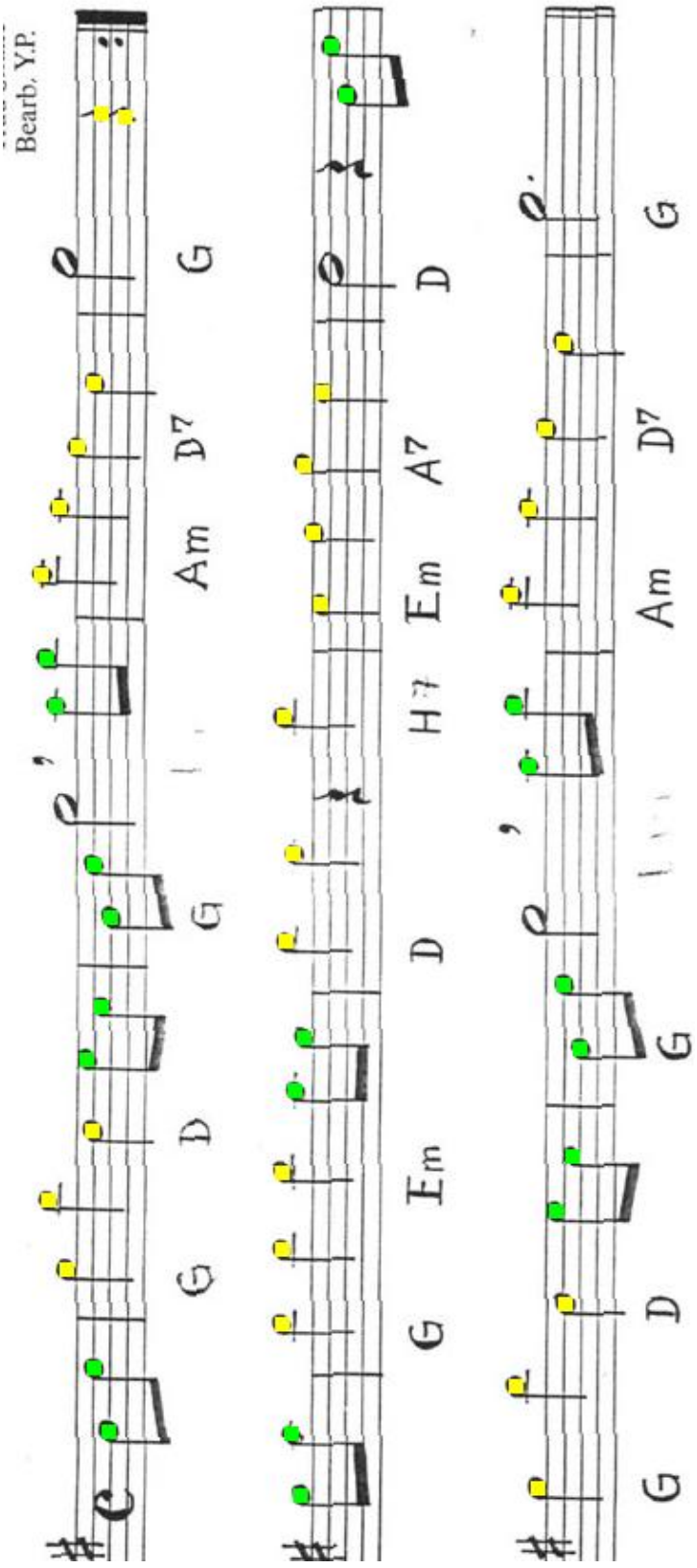


Figure 22: Bred dina vingar ’b2d3G3C4D3e3c3b2d3b3c4C4A3F3D3G2C3n
a3b3C4C4C4a3g3B3B3C4E3F3G3E3b2d3n
G3C4D3e3c3b2d3a3c4C4A3F3D3’

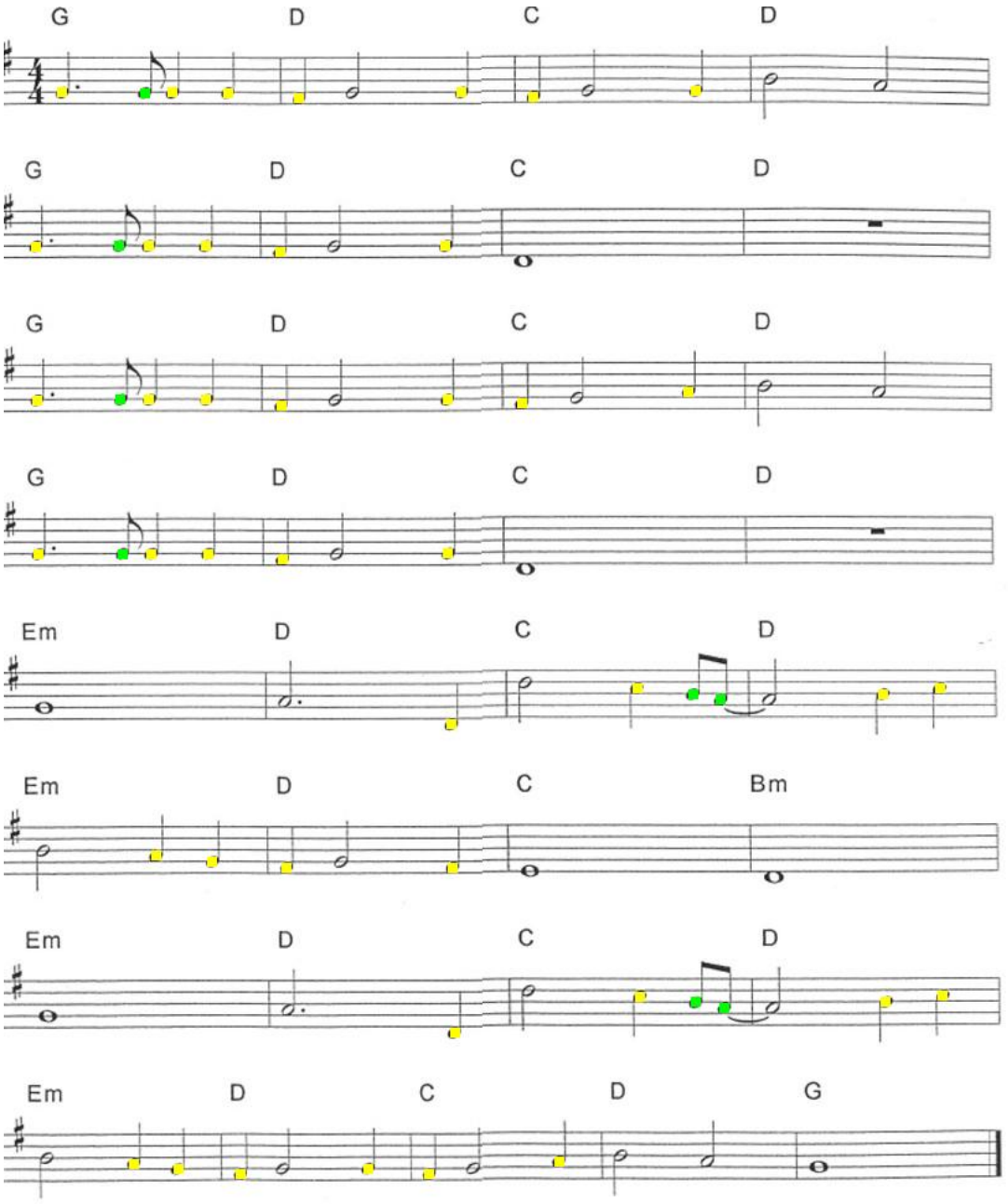


Figure 23: Titanic
’G2g2G2G2F2G2F2G2n
G2g2G2G2F2G2n
G2g2G2G2F2G2F2A2n
G2g2G2G2F2G2n
D2C3b2a2B2C3n
A2G2F2F2n
D2C3b2a2B2C3n
A2G2F2G2F2A2’

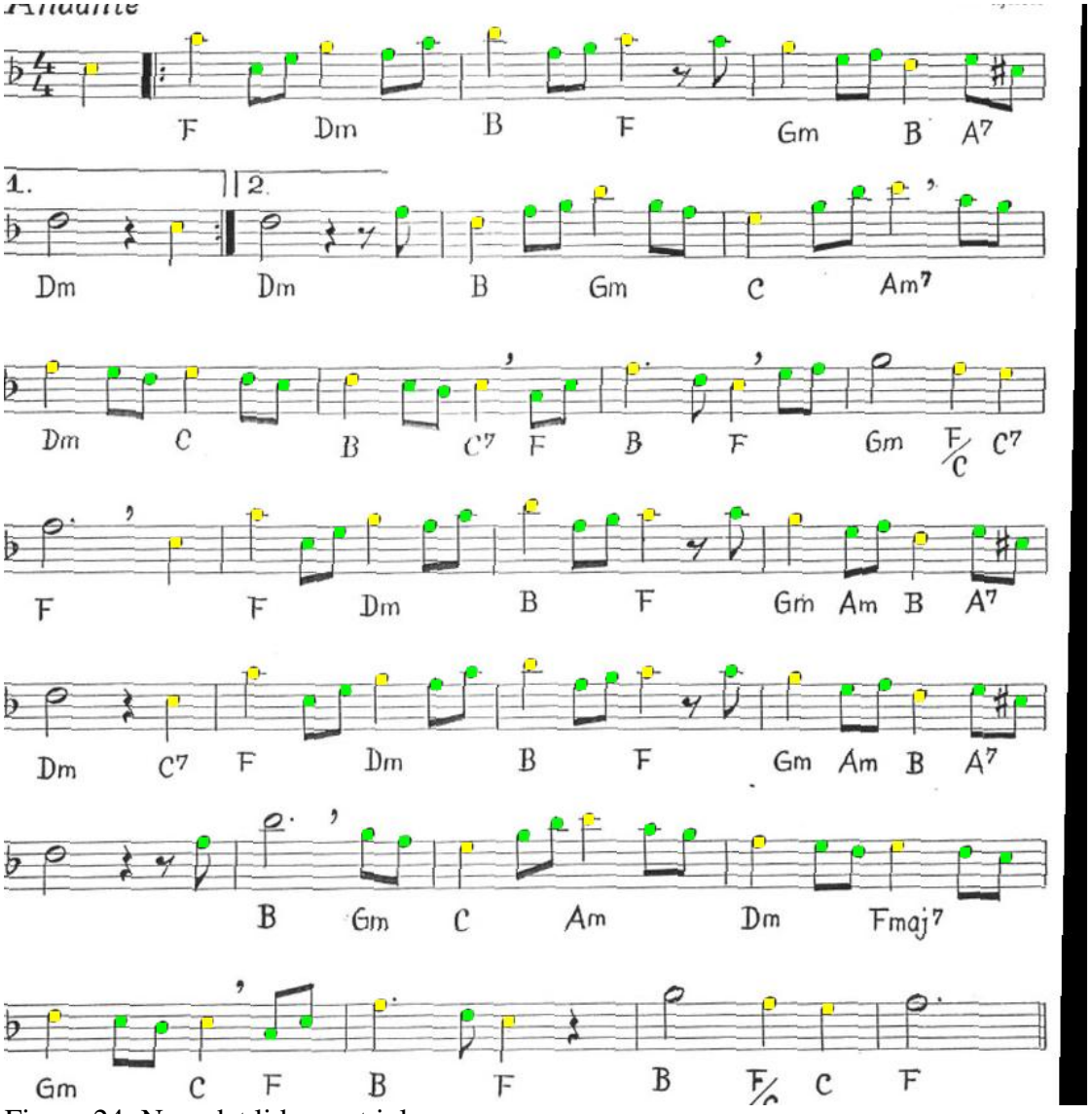


Figure 24: Naer det lider mot jul

’D3B3d3e3G3f3a3C4f3g3B3a3G3e3f3D3e3c3n
C3f3D3f3g3C4g3f3E3g3c4C4a3g3n
F3e3d3E3d3c3D3c3b2C3a2c3F3d3C3e3f3F3E3n
C3A3c3e3G3f3a3C4f3g3A3b3G3e3f3D3e3c3n
C3A3c3e3G3f3a3C4f3g3A3a3G3e3f3D3e3c3n
f3g3f3E3g3b3C4a3g3F3e3d3E3d3c3n
D3c3b2C3a2c3F3d3C3F3E3’

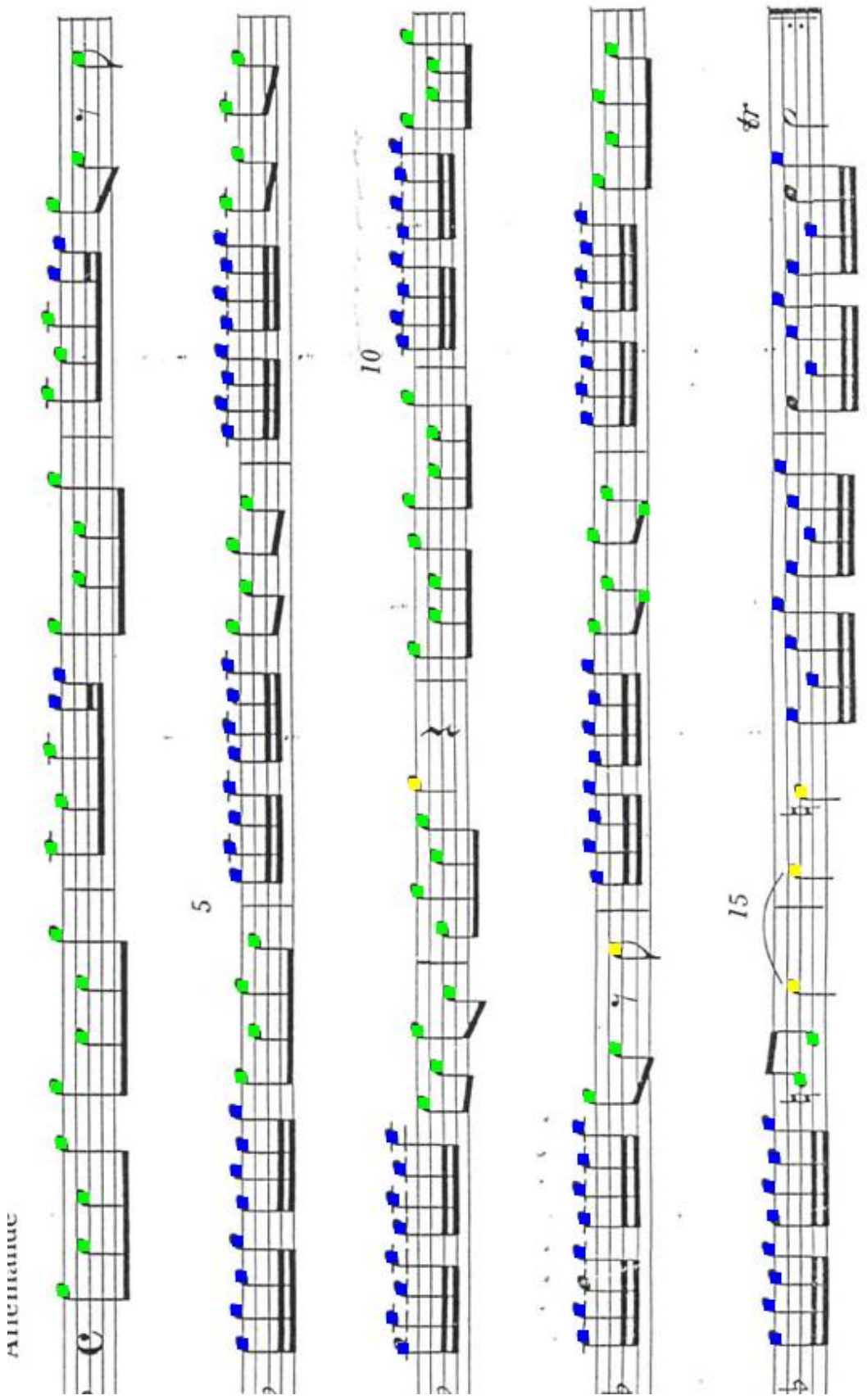


Figure 25: Allemande ’f3c3c3f3g3c3c3g3a3f3a3g3c3c3a3a3g3a3a3d3d3n
f3d3f3d3g3e3g3e3b3g3b3g3n e3c3f3a2b2f3c3e3F3g3c3c3f3g3c3d3a3a3d3d3a3n
g3c3C3g3f2d3g3f2e3f3d3f3d3n b2g2C3C3B2’