Alignment of handwritten music scores

Ivan Santos Panadero

Resum— Hi ha musicòlegs que dediquen el seu temps a analitzar obres musicals de fa més d'un segle per enllaçar-les amb altres ja existents del mateix autor però escrites per mans diferents. És una tasca tediosa, doncs són moltes les representacions que s'han pogut fer d'una mateixa obra al llarg del temps, i la variabilitat d'escriptura entre aquestes pot ser molt ample. La finalitat doncs, seria la de tenir una base de dades variada d'aquestes composicions antigues per a l'estudi, reproducció i difusió. Aquest treball es divideix en dues fases. La primera, consistent en la detecció dels elements presents en cada un dels compassos d'una partitura a partir de la transcripció existent de la partitura, conseguint així un alineament guiat. La segona tractarà d'analitzar aquest alineament. Els resultats obtinguts són encoratjadors.

Paraules clau— Alineament, Partitures musicals escrites a mà, Reconeixement òptic musical, Zoning, Blurred Shape Model

Abstract— There are musicologists that spend their time in analyzing musical pieces of more than a century ago in order to link them to another pre-existing pieces from the same author but written by different hands. It is a tedious task, since there are many representations done of a single piece through the time, and the writing variability among those representations can be extensive. The purpose would be in having a varied database of these old compositions for the study, reproduction and difusion. This work is divided into two phases. The first one, constitent in the detection of primitive present elements in each of the measures of a score using the existing transcription of the piece, thus obtaining the desired guided alignment. The second one will seek to analyze this alignment. Obtained results are encouraging.

Index Terms— Alignment, Handwritten music scores, Optical music recognition, Zoning, Blurred Shape Model

1 Introduction

Optical music recognition is a well-known problem in the computing industry [1], [2]. For over five decades [3], it has drawn the attention of researchers across the globe, even though the existing challenges in musical data acquisition [4].

Many musical pieces written a long time ago are only accessible via photocopies or as original manuscripts, but these days demands better technological solutions, so they can be addressable to everyone who may want to analyse or reproduce the work.

The optical music recognition (OMR) is an open problem, despite the fact of existing commercial solutions, such as PhotoScore [5], SmartScore [6], SharpEye [7]... However, the acquisition of musical data from handwritten scores is not that easy nor trivial. The high variability of each person's writting style makes it a difficult task to approach. In addition to that, the condition of musical pieces is usually far from ideal, thereby aggravates the detection process.

• *E-mail de contacte: ivan.santos@e-campus.uab.cat*
• *Menció realitzada: Computació.*
• *Treball tutoritzat per: Alicia Fornés-Bisquerra, Arnau Baró-Mas* (CC. de la computació)
• *Curs 2018/19*

The capabilities of a proper detection and alignment are numerous: any musician could scan the piece and automatically search for the MIDI version of it to play along; a photo could be taken in a museum of the original piece and some information such as the royal composer, dates or a playing sample could be returned to the visitor; or even for educational purposes, as children could write a musical piece on the chalkboard with the teacher's guidance so they would learn the fundamental basis of music.

We have to remember that optical music recognition (OMR) is slightly different to optical character recognition (OCR), given that OCR must retrieve its data in a sequential form, unlike OMR where there might be other coexisting voices in parallel.

This paper is divided into the following sections. First, the objectives of the project will be discussed. Thereafter, the state of the art of optical music recognition (OMR) is commented in section 2. The methodology, project planning and technological questions are described in section 3. Section 4 will describe the processing phase, section 5 will discuss the common challenges to be faced in a problem like this. The alignment output is shown in section 6, and obtained results are shown in section 7. Finally, conclusion and future lines of work are reasoned in section 8.

1.1 Objectives

The main purpose of this work is not to create nor detail a new way for optical music recognition, but for research, analysis and apprenticeship intentions. Many other options already exist that adequately meet the objective, so this paper is merely intended for the exploration of already functional solutions.

This work aims to achieve a good detection system of each of the bars of a music piece, which is given in picture files. These files must be of fairly high quality, since we are going to crop every bar to treat each one individually to acquire our music data.

Thus, the tasks to be done are the following ones:

- Analyze which technique would be better for the detection phase.
- Develop the system for getting every bar to treat.
- Develop the detection system depending of the data.
- Correctly align measures from the transcripted piece with detected elements of measures from the handwritten piece.
- Evaluate obtained results.

2 State-of-the-art

There are mainly two modes for optical music recognition based on the input data: offline and online [8]. Offline mode is based on the processing of images of music; this is, according to gathered data from processed pixels. On the other side, online mode is conditioned by the way the sequence is written, or how the strokes were entered in the system (a better segmentation is achieved).

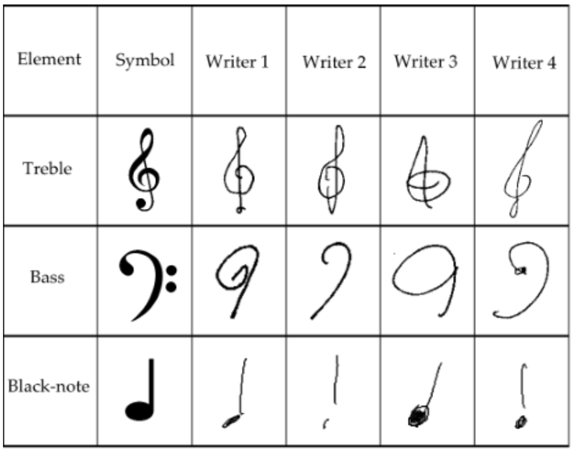

Fig. 1: Variability in handwriting. Extracted from [9] and [10].

Obviously, printed scores are much easier to process and apply the recognition phase than handwriting due to the variability of the latter. In Figure 1 we can see how a symbol can vary in handwritten pieces for different hands.

The recognition mode that this work is going to follow is offline, so we are going to see the most common techniques of OMR nowadays.

2.1 Commercial solutions

As told before, PhotoScore, SmartScore and other commercial solutions really approach very well the open problem of OMR. By the way, these solutions are mainly focused on obtaining a transcripted score given a printed one.

For the moment, they are not much advanced in giving good handwritten solutions, but we expect that in a nearly future, best software solutions are going to be developed.

2.2 Neural networks

The truly state of the art of optical music recognition is in the use of neural networks. As stated in [11], recurrent neural networks (RNN) are very useful to model time series, this means that the network contains some sort of memory (outputs from neurons in hidden layers are inputs for some neurons, whose output now falls in the aforementioned layer). This solution works well for handwriting recognition or speech recognition series.

In [12], another approach is presented using combined neural networks (CNN), which aims to obtain the best classifier possible given a combination of other classificators.

We have to take in mind that these solutions need a wide base of good samples to train the models, otherwise the neural networks would not be able to achieve considerable results on their own.

2.3 Common solutions

Other feasible solutions (including the ones mentioned in section 4) are the use of classification algorithms such as k-NN (k-Nearest Neighbors) or k-Means, whose outputs can be every class (every required symbol).

Also, the use of HMM (Hidden Markov Models) [13] are widely used for pattern recognition, therefore they are useful for the recognition goal.

A solution based on Dynamic Time Warping algorithm [14] is also to be taken in consider, not because is a time series algorithm, but because the method allows to compute the minimum matching cost for two symbols.

3 Methodology

The chosen methodology for the development of this project is the incremental. I think it is the most worthwhile one because of the need of working with iterations.

In each iteration of the project, a delivery of the current situation is being held, being this one more step in the course of the project development.

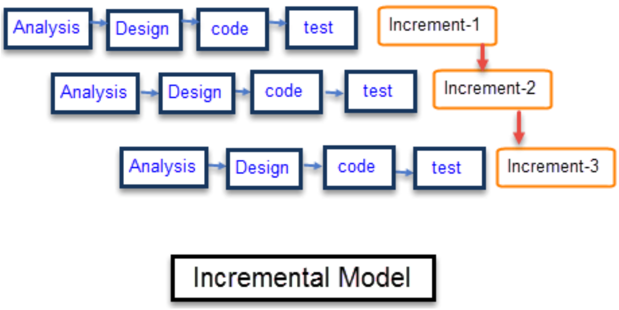

Fig. 2: Incremental model method for software development.

In Figure 2 the incremental model is sketched out. We can see that for each increment (loop or iteration) an analysis step is executed, followed by a re-thinking of the design or new designing, and finally a coding and testing tasks.

3.1 Planning

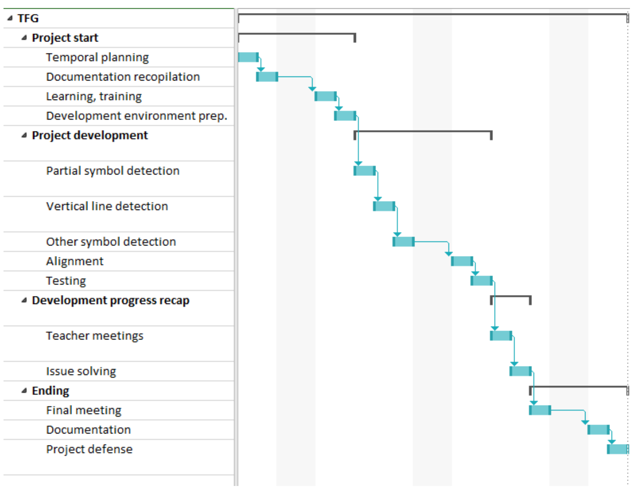

Fig. 3: Project planning

The planning of the project is shown in Figure 3. The tasks are:

1. Project start: Probably the most crucial task, because a proper gathering of documentation and learning period is the key to a successful development.
2. Project development: As the title says, this is the task where the development is being made, but obviously the learning phase is going to be present all along the project.
3. Development progress recap: Once the development task is done, we must look back and analyse what can be done better, what can be improved and what is not necessary.
4. Ending: The final phase of the project. Here, the documentation and the defense is done. After that, the project can be considered as closed.

3.2 Environment

We depart from having a total of 6 image files of a music score, corresponding to the musical piece *Motet: Laudate Dominum*, represented by the composer Pau Llinás, an 18-th century catalan composer. Motets are usually polyphonic compositions, which are meant to be sung in churchs.

Also, two .xml files containing information of the piece are provided. They correspond to the transcription that the musicologist made of the piece. These files include a lot of information of the musical piece, such as every single bar with the notes inside it, the present lyrics if any, clefs, note duration, the pitch...

We can import these .xml files into software such as Finale [15] or MuseScore [16] if we want to be able to see (in a musical way) the transcription of the musicologist.

The programming language to be used is Python, among other possibilities. The fact is, that this language provides efficient tools for processing images (such as the OpenCV library, or the SciPy environment).

Finally, we also have a set of .txt files where the coordinates of each bar are represented. With this information, we can easily crop the score into individual measures for the further processing. This files will ease the detection phase.

3.3 Pipeline

As we have briefly commented before, an image containing part of the musical piece is given to the system. Thanks to the cropping coordinates we obtain a subset of measures, every one being correctly labeled to its score.

Once done, we are ready to start the alignment phase. This pase consists in getting every single element that the transcription have, and for every element, we search it thoroughly in the handwritten measure. According to the searched element, one or another process will be carried out. This processing will be further explained in section 4.

After that, the obtained alignment will be analyzed qualitatively, to discuss if the system works accordingly or not.

Figure 4 shows the aforementioned tasks. Specifically, the first part of the pipeline corresponds to the cropping phase above mentioned, and the second part of it corresponds to section 4.

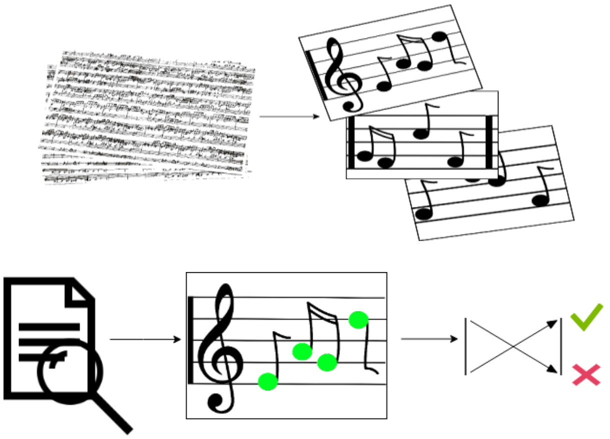

Fig. 4: Pipeline. Top corresponds to cropping phase, bottom to aligning phase.

4 Processing

In this section, the processing phase is stated. We have to take in mind that a process is being held according to the nature of the element. This means that, if a black-headed note is being searched, the finding process will be different from the white-headed note finding process.

4.1 Black-headed note detection

The fundamental and most obvious characteristic of black-headed notes as they name state, is that they get a higher proportion of ink than other notes. This plays in our favor, because thanks to morphology techniques we can easily segment these ink areas and skip the rest.

Concretely, we get the thresholded image of a measure, and then apply a morphological closing in order to "paint" all remaining small holes inside possible notes (Figure 5).

Finally, we only have to find the contours of the image to have an approximation of the covered area by a note.

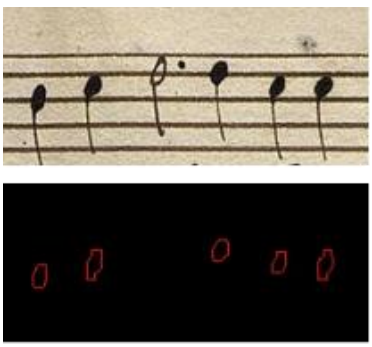

Fig. 5: Contours of black-headed notes.

We have to keep in mind that this is not enough for the correct detection, as any kind of existing ink stain in the measure could be considered as a note. The next subsection explains how can we filter this ink spots out in order to get only desired notes.

4.1.1 Hough Transform

To avoid as much as possible a bad detection, vertical note lines (or stems) are detected via the probabilistic Hough transform [17].

This transform is better in computation time than the classic Hough transform because it gets only a random subset of points for line detection. Nevertheless, transform parameters must be correctly tuned in order to get only vertical lines, and for a better detection, not the entire measure is passed to the function, but a crop containing just the found contour.

This method returns a list of possible vertical lines found, so in order to get the best matching line to the current analyzed note I follow the next strategy:

- Get the midpoint of every returned line
- Compute the euclidean distance between the contour centroid and this midpoint.
- The closest distance of a given line is the best found line.

We finally state that if a stem is detected within the range of a contour, a black-headed note (quarter, eighth or sixteenth) is detected.

4.2 White-headed note detection

The previous thresholded image now is processed by deleting the found contours. This is, by drawing a rectangle where a black-headed note may be.

After that, an horizontal and vertical projection is made (see Figure 6). With this projections, we can now determine where the position of a white-headed note can be, because the projection tells us where resides a higher ink proportion.

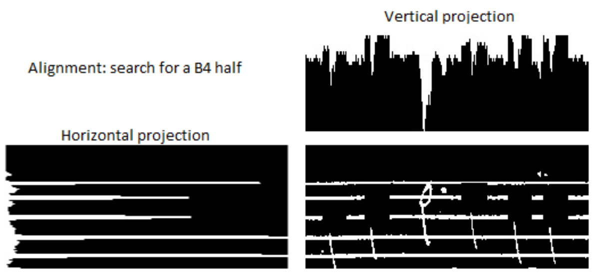

Fig. 6: Measure projections.

We know that a half or a whole note exists, but to know where it can be, we have to know the nature of the note, and the .xml files contains all this information. For example, for the previous half note in Figure 5, we know that it is a B in 4th octave. This entire musical piece is set on the C clef, and knowing that the clef determine the pitch and melody of the music piece, we are sure that the nature of the note is going to probably be in a particular region.

If a peak is found in the vertical projection, we now ensure the position by looking at horizontal projection, whose peaks correspond to the pentagram lines of the measure.

Now, we can assure that a white-headed note is present in this particular coordinates.

4.3 Accidentals and rests detection

For the detection of accidentals and rests, we have to use another technique. A symbol descriptor fits perfect this task, and because of that, two descriptors are tested in this project: Zoning and Blurred Shape Model descriptors. In the next subsections, both will be explained and compared.

This task is not trivial, since the musical notation has changed with time. Besides, the handwriting technique varies roughly among musicians. In Figure 7, some examples of accidental and rests will be shown to know the symbols we are searching for.

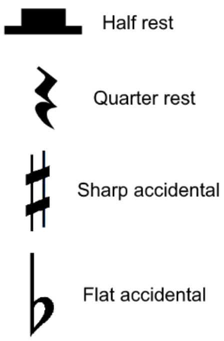

Fig. 7: Common complex-to-spot musical symbols

4.3.1 Zoning

Zoning, described in [18], relies on superimposing a grid of nxn pixels through the image, and for every grid, the percentage of black pixels are computed. This method is very fast and commonly used in the search for symbols.

In Figure 8 we can see how Zoning works for different nxn grids.

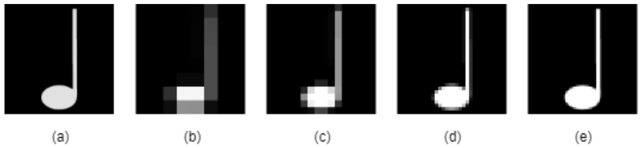

Fig. 8: Zoning descriptor. (a) Base image. (b) 8x8 regions. (c) 16x16 regions. (d) 32x32 regions. (e) 64x64 regions.

By enlarging the grid, (less regions) the image gets more blurred, because a bigger area is used in the percentage computation.

By the way, this descriptor has a problem: the computed regions discriminate the other regions, and therefore, the result of the descriptor; meaning that it returns an unsatisfactory result.

If a crucial region falls on a particular region, only this particular region will be aware of it, but not it's neighbors.

For this particular problem, a better descriptor exists, and is described in the next section.

4.3.2 BSM

BSM stands for Blurred Shape Model, and it is described in [19]. For every point forming the shape of a symbol, is treated as a feature to compute the descriptor. The contribution of every single shape point is weighted according to the distance between the point and the centroid of its neighboring regions.

This means that, unlike the Zoning descriptor, each point of the shape contributes to its neighbors, not making it a bad descriptor if a crucial part of a symbol falls in one unique region.

The output of the descriptor is a vector histogram of nxn subregions, and each position corresponds to spatial distribution of each shape point. If a shape point is closer to a region, it scores more than another far away from the point.

BSM is a robust descriptor given rotations and elastic deformations of the shapes, although the search for the optimal descriptor continues nowadays.

In Figure 9, different outputs for the descriptor are shown.

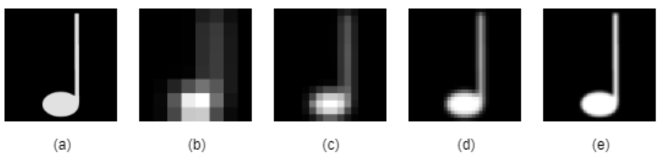

Fig. 9: BSM descriptor. (a) Base image. (b) 8x8 regions. (c) 16x16 regions. (d) 32x32 regions. (e) 64x64 regions.

As we can see in difference with Zoning descriptor, BSM gets actually a more blurred version for the same number of regions, thus showing the contribution among regions.

4.4 Other strategies

There are other simple strategies that can be followed in order to achieve a better alignment:

- Get only the contours for black-headed detection that meet a specific standard, this is, by area for example. Contours that surpass a high threshold could be ink stains, and the ones that not surpass a low threshold could be simple morphology errors.
- If a note has been detected before, use its coordinates to search from there. For this project we are searching for notes sequentially, so another possible note previous to the last detected gets automatically discarded.
- For a better black-headed note alignment, we can use the horizontal projection used in white-headed note detection, to check where a note may fall given its nature.
- Only a crop of the left of a note is passed to the descriptors if an accidental is present, in order to improve computation time of them. As a general rule, accidentals live to the left of the note, so there is no need to check at either the right or even the entire measure.
- A re-crop to the entire measure is being done, in order to avoid as much as possible bad detections. We aim to get only the important information, and this is by skipping all the present noise in a measure.

With this little strategies, we can assure an even better alignment. But, nevertheless, we have some challenges to overcome or at least try to.

5 Challenges

We have to keep in mind that this handwritten musical piece is from around 1700, and its simple conservation can actually be a challenge for its processing.

The challenges and/or problems found are the following:

1. Ink stains and dirt are present all over the music score. There is not much to be done, but if a bigger contour is found, it can be filtered out given the strategies abovementioned. If not, it may be considered as a correct note.
2. It is, in fact, handwritten. The high variability in the writing technique and the quality of the degraded paper through the years makes it even more harder.
3. Also, the musical writing style of that time can differ greatly from how we understand it today. There are some accidentals found on top of the notes, and not in the near left as it should be by nowadays.
4. Musicologist interpretation. It could be possible that the musicologist who did the transcription thought that one measure was one way and not the other as the composer wrote. This does not mean that the interpretation is wrong, but that a musicologist thought that the most correct way of writing some measure was his way, or maybe adapted to the current times.
5. Corrections by the musicologist. Continuing from the previous point, it could be that the musicologist saw flaws in the handwritten piece, and that he or she made the appropriate changes.
6. Flaws by the musicologist. We are humans, and as such, we make mistakes. There is no evidence that the transcription is 100% correct, and probably errors are present.

Despite all this challenges, we can try and anticipate to them in order to get significant good results.

6 Alignment

As mentioned in the Pipeline subsection (3.3), we get every single element of the "printed" version (the transcripted piece), and then we search for it into the handwritten measure.

This process's output have the following structure for a single measure:

```
### Alignment for measure 79 ###
Found: whole G4 note at coords (51,64), sharp accidental not found.
Found: quarter A4 note at coords (116,55).
Found: quarter B4 note at coords (157,42).
Found: eighth C5 note at coords (217,36), sharp accidental found.
Missed: eighth D5 not found.
Found: quarter E5 note at coords (242,26).

Accuracy: 83.33%
```

Please note that the coordinates described in the output are not from the whole handwritten image, but for this particular cropped measure. I found it irrelevant as the real goal is to analyze if it is correctly found in the measure.

In the next section, results of applying the previously mentioned strategies are being discussed.

7 Results

To obtain significant results, from around 240 measures of the entire musical piece we will get 50, which I consider sufficient to test the detection system. All 50 measures have been taken from the entire work, so they are not sequential, but from all over the 6 images corresponding to the score.

Also, the measures contain as much variance as possible, meaning that they include wholes, eighths, quarters, notes with accidentals… We want to have a realistic qualitative analysis of the system.

The results are described in terms of Precision and Recall.

Given that the system relies on the searching of elements, precision and recall metrics fits best for our purpose. This metric is used in pattern recognition, binary classification and information retrieval, and the terms are described as follows:

- Precision: Fraction of retrieved elements that are relevant. Equation (1) shows the calculations.

$$Precission = \frac{\#relevant\ found}{\#found}\ or\ \frac{TP}{TP+FP} \quad (1)$$

- Recall: Fraction of relevant documents that are successfully retrieved. Equation (2) shows the calculations.

$$Recall = \frac{\#relevant\ found}{\#relevant}\ or\ \frac{TP}{TP+FN} \quad (2)$$

The metrics shall be calculated by class, this is, by note typology.Concretely, we will have 4 different classes: half, quarter, eighth and whole metrics.

	Precision	Recall	TP	FP	FN
Half	0,8491	0,9375	45	8	3
Quarter	0,9529	0,9643	81	4	3
Eighth	0,9608	0,98	49	2	1
Whole	0,7727	0,8947	17	5	2

Table 1: Results in terms of Precision and Recall. All values are between [0-1].

Table 1 shows the obtained results. They are obtained by a grand-total of 220 notes among all measures.

	F1 Score
Half	0,8911
Quarter	0,9586
Eighth	0,9703
Whole	0,8292

Table 2: Results in terms of F1 Score. All values are between [0-1].

It could be that the metrics of Precision and Recall were good in Precision because the system said that a note is a correct note, but we could still get a bad Recall given that a lot of relevant notes are not detected. For this reason, an harmonic average of Precision and Recall is used, called F1 Score (see Table 2). This metric is computed as for equation (3).

$$F1\ Score = 2 * \left(\frac{Precision * Recall}{Precision + Recall}\right) \quad (3)$$

The good thing about this metric is that if a bad result appears in Precision or Recall, this harmonic average decreases dramatically, thus F1 Score gives us a more general overview of how well our system works.

For the calculation of the results, we take into account that:

- TP or True Positives: All notes correctly detected and found.
- FP or False Positives: All notes found incorrectly, at wrong place on the measure.
- FN or False Negative: All notes not found on the measure.

These metrics describe well the performance of our system, because metrics such as accuracy shown in the previous section are not realistic. It is computated according to the obtained results, but they can be wrong. A note can be labeled as found, but it could be an ink stain, dust in the score or even misplaced by another. Because of that, we have to talk about the confidence of our method, and the best way is to use the abovementioned metrics.

On the other hand, Precision and Recall metrics are calculated in a visual way, meaning that they are more precise than the computed accuracy.

Regarding accidental detection: from the previous 50 measures mentioned, there are a total of 11 present accidentals. Of those, only 4 are being correctly detected, giving us a rate of $4/11$ = 36,36% detection.

This relatively small value is given due to discrepancies among the handwritten piece and the transcription, probably due to corrections of the musicologist. In addition to this and as a matter of fact, some accidentals descriptors such as the sharp one is often misrecognized as a note, given the common peculiarities of the accidental (a whole in the middle, ink by the sides).

With regards to rest detecion: with this method, it is currently not possible to correctly detect any of them. We have to understand that the common rests such as whole, half or quarter are usually little dots or lines of ink. This makes it nearly impossible to detect, because of all the false positives retrieved. Besides, the computational time of the BSM descriptor for sliding a window in the entire measure searching for a single rest is far from ideal.

8 Conclusion

Obtained results are promising. It is clear that white-headed notes are harder to detect, and so is shown in the results table.

We have to keep in mind that, although we are talking about detection, this project is about alignment, so the results to be discussed is how well our method aligns with the existing transcription.

The best results are obtained by the eighth class, closely followed by the quarter class. Obviously, with a good morphology process, the contour detection gets all relevant black-headed notes present in the measure, because the ink level of these notes is significant.

Talking about the best, by obtaining a percentage of 96,08% of Precision we can assure that almost every eighth note found among all notes found are correctly detected. With a Recall of 98%, we can say that nearly all eighth relevant notes are retrieved, barely without noise.

By the other side, the worst results are obtained by the whole class, followed by the half class. The pre-process of deleting black-headed notes works very well, but the projection strategy relies on the amount of present ink, and whole/half notes are empty on the inside. Despite that, results are not bad.

For that, a musicologist running this method with the existing transcription could observe the differences that can exist among them: how a symbol was interchanged by another, or even what the composer thought to be an enhancement to the musical work, to ease parts for the musicians who would perform the piece. In concrete, a musical piece can have many many interpretations, every single of it with different annotations, changes in the melody or tempo… For all this reasons, the differences between passages are interesting for the study.

Encouraging results are achieved with the proposed techniques, but further future work can be done. I would like to improve the white-headed note detection, as it is clearly limiting the method. Apart from that, a better generalisation of the method could be important, because the developed strategies might fail if applied in another score. This sentence is easier said than done, because other scores can be structurally different from the ones provided and the algorithm should certainly be tuned. Finally, I would love to implement a better alignment method according to the state-of-the-art, like neural networks, but this method could need probably thousands of handwritten music samples, and I am not 100% sure if the computational time would make this strategy feasible.

In Appendix A1, further results of detection are shown in a qualitative way.

Acknowledgements

I would like to thanks my tutors Alicia Fornés Bisquerra and Arnau Baró Mas for all the given support even when all seemed lost.

Also, my most sincere gratitude to my mother, my fiancée Sara and my in-laws, whose support and patience turned out to be infinite in moments of pure anxiety.

References

[1] Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marcal, A. R., Guedes, C., & Cardoso, J. S. (2012). *Optical music recognition: state-of-the-art and open issues*. International Journal of Multimedia Information Retrieval, 1(3), 173-190.

[2] Rebelo, A., Capela, G., & Cardoso, J. S. (2010*). Optical recognition of music symbols.* International Journal on Document Analysis and Recognition (IJDAR), 13(1), 19-31.

[3] Pruslinn, D. (1966). *Automatic recognition of sheet music* (Vol. 21, pp. 15-27). Taplinger.

[4] Brainbridge, D., & Bell, T. (2001). *The challenge of optical music recognition*. Computers and the Humanities, 35(2), 95-121.

[5] PhotoScore: https://www.neuratron.com/photoscore.htm

[6] SmartScore: https://www.musitek.com/

[7] SharpEye: http://www.visiv.co.uk/

[8] Hajic jr, J. *Handwritten Optical Music Recognition* (Doctoral dissertation, phd thesis proposal. http://ufal.mff.cuni.cz/zabokrtsky/pgs/thesis_proposal/jan-hajic-jr-proposal.pdf, 2017. [Online]).

[9] Fornés, A., Lladós, J., & Sánchez, G. (2007, September). *Old handwritten musical symbol classification by a dynamic time warping based method*. In International Workshop on Graphics Recognition (pp. 51-60). Springer, Berlin, Heidelberg.

[10] Calvo-Zaragoza, J., & Oncina, J. (2014, August). *Recognition of pen-based music notation: the HOMUS dataset*. In Pattern Recognition (ICPR), 2014 22nd International Conference on (pp. 3038-3043). IEEE.

[11] Baró, A., Riba, P., Calvo-Zaragoza, J., & Fornés, A. (2017, November). *Optical Music Recognition by Recurrent Neural Networks*. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on* (Vol. 2, pp. 25-26). IEEE.

[12] Wen, C., Rebelo, A., Zhang, J., & Cardoso, J. (2015). *A new optical music recognition system based on combined neural network*. *Pattern Recognition Letters*, *58*, 1-7.

[13] Pugin, L. (2006, October). *Optical Music Recognitoin of Early Typographic Prints using Hidden Markov Models*.

In *ISMIR* (pp. 53-56).

[14] Fornés, A., Lladós, J., & Sánchez, G. (2007, September). *Old handwritten musical symbol classification by a dynamic time warping based method*. In *International Workshop on Graphics Recognition* (pp. 51-60). Springer, Berlin, Heidelberg.

[15] Finale: https://www.finalemusic.com/

[16] MuseScore: https://musescore.com/

[17] Stephens, R. S. (1991). *Probabilistic approach to the Hough transform*. Image and vision computing, *9*(1), 66-71.

[18] Fornés, A. (2005). *Analysis of old handwritten musical scores*. *Master's Thesis,* Universitat Autònoma de Barcelona, Spain.

[19] Escalera, S., Fornés, A., Pujol, O., Radeva, P., Sánchez, G., & Lladós, J. (2009). *Blurred shape model for binary and grey-level symbol recognition*. Pattern Recognition Letters, *30*(15), 1424-1433.

APPENDIX

A1. Qualitative results

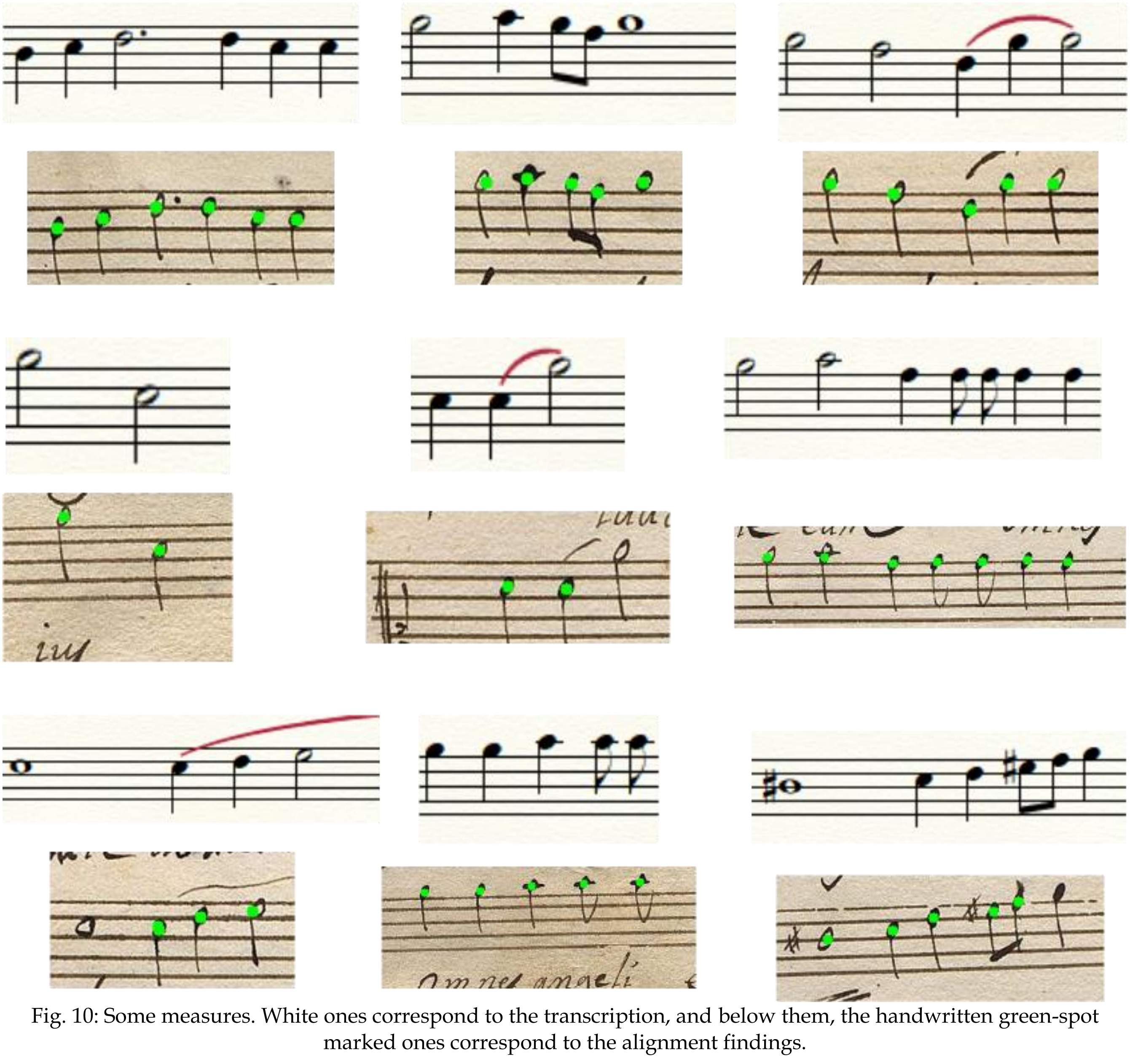

Fig. 10: Some measures. White ones correspond to the transcription, and below them, the handwritten green-spot marked ones correspond to the alignment findings.