Copista - OMR System for Historical Musical Collection Recovery

Marcos Laia, Flávio Schiavoni, Daniel Madeira, Dárlinton Carvalho,
João Pedro Moreira, Avner de Paulo, and Rodrigo Ferreira

Computer Science Department
Federal University of São João del-Rei
São João Del Rei – MG – Brasil
marcoslaia@gmail.com, {fls,dmadeira,darlinton}@ufsj.edu.br
joaopmoferreira@gmail.com, avnerpaulo.mg@gmail.com,
rodrigoferreira001@hotmail.com

Resumo Optical Music Recognition (OMR) is a Computer Science field applied to Music that deals with problems like recognition of handwritten scores. This paper presents a project called "Copista" proposed to investigate techniques and develop a software to recognize handwritten scores especially regarding a historical musical collection. The proposed system is useful to collection preservation and as supporting further development based on OMR. "Copista" is the Brazilian word for Scribe, someone who writes music scores.

1 Introduction

Some of the most important music collections in Brazil, dated from the beginning of 18th century, are located in São João Del Rei, Tiradentes and Prados. These collections include several musical genre and are the work of hundred composers from this historical Brazilian region.

The Music Department of Federal University of São João Del Rei started a program to describe and catalog these collections, called "Memória Viva" (Living Memory), trying to provide these collections to public audience. The main aspect regarding these collections is that the old sheets have several marks of degradation like folding, candle wax, tears and even bookworm holes, as depicted in Fig. 1.

In order to help the processing of these music papers, a partnership of Music Department with the Computer Science Department in the same University arose. This partnership involved several researchers on the creation of an application called Copista, a software to help musicians to rewrite music scores collections based on a digital copy of them. The project Copista comprises the digital image acquisition from the original files, digitally recovery of the files and transcript the music files to a symbolic music representation.

Each step on this process would return partial results that are important to preserve these historical collections. The scanned original files are valuable to musicology since they keep historical features of each sheet. The digital recovered

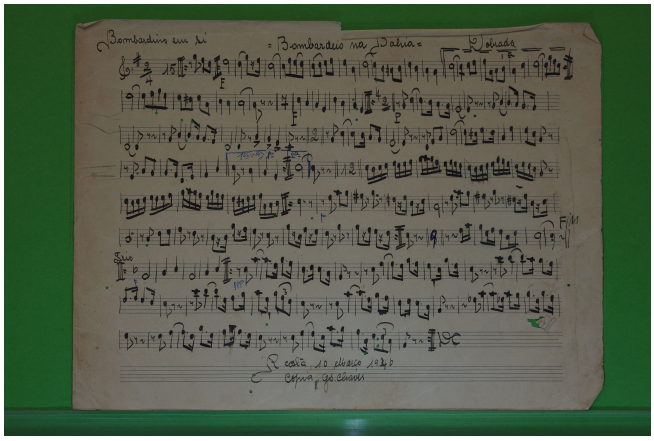


Fig. 1: Example of music score present in the collections

files are also important since it can be easier to read and distributed them. The symbolic music representation is another important artifact since it is easier to work with these files to check and correct some transcription problem.

The remainder of this paper is organized as follows: Section 2 presents the Copista system, Section 3 presents some Partial Results and Section 4 presents this article Conclusion.

2 The Copista

The Copista system is proposed as a tool to convert handwriting scores into a digital music representation. The applications used to interpret music scores are called Optical Music Recognition (OMR) [26] [5]. These applications are similar to Optical Character Recognition (OCR) tools but they should be able to convert handwriting scores into symbolic music. In spite of existing tools that converts handwriting scores into editable scores, most of these tools a) do not work with manuscript scores[5], b) are very expensive and c) are not open source, being impossible to adapt them to this project. All these reasons helped us to decide to build a brand new tool on the OMR field.

To develop such tool, we divided the OMR process into some distinct parts: the image acquisition, image preprocessing and digital image recovery, the recognition of musical Symbols with Computer Vision, the Music Notation Reconstruction and the symbolic music output, as depicted in Fig. 2.

2.1 Image Acquisition

The Copista input is a handwriting score from regional historical collections. In these collections, it is common for the scores, many of them centuries old, have

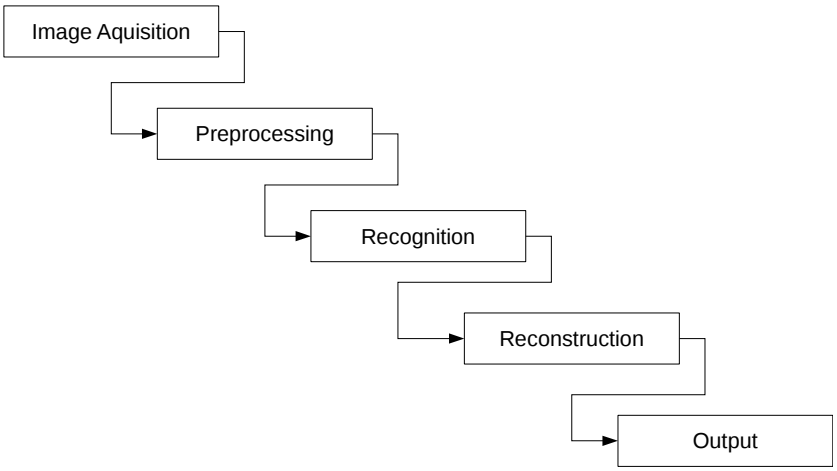


Fig. 2: The Copista Framework

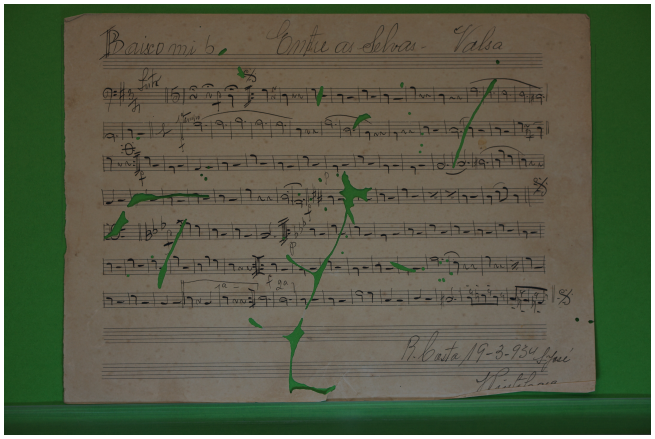


Fig. 3: Damaged score

been used in Masses and processions, and have folds, candle wax marks, dents, tears and other damage, as can be seen in Fig. 3:

For this reason, firstly a physical restoration of the scores of collections are being performed. Once this restoration is performed, the score should be digitized to be processed by Copista.

During the Acquisition process, a high resolution camera is being used. We used a green chroma key under the original score to facilitate the identification of the sheet damage.

2.2 Preprocessing

It is common that ancient papers and manuscripts suffer degradation over time. Constant handling can rub the ink out, creating differences in shades or various marks. In the example of sheet music, candle wax drippings and sweaty hands creates marks in each document in several cases. In addition, improper storage caused folds, wrinkles, tears and holes caused by bookworms in the sheet music. All these marks are not relevant and need to be removed to make the recognition process more adequate.

There is no universal technique available for preprocessing, as for each document a specific treatment set may be required. Nonetheless, two steps can be highlighted as the basic pre-processing process for the Copista:

1. artifacts removal
2. color thresholding

The first step involves removing all artifacts (i.e. marks) non-important to the recognition process. These artifacts, which become noise in the acquired image, cover the stains, rips, holes and all marks that are not part of the score. The paper itself can be considered noise, because it is not part of the score itself. Holes and rips on the paper are the hardest artifacts presented, because they alter the paper format, while erasing the data on the score.

Therefore, this step comprises a set of algorithms. Image filtering [10,31] and hole filling [3] are necessary. The chroma key used in acquisition step helps to make holes easier to spot. Consequently, the hole-filling algorithm needs it in order to remove all of them efficiently. At the end of this step, the brightness and contrast are enhanced in order to clarify the acquired image, passing it along to the next step.

With noise removed, the score needs to be converted to a black and white format. After the color conversion a two-level thresholding processing (binarization) is employed in order to achieve the final objective. This process simplifies score representation, cutting off color and grey variations. The thresholding process can be classified into two categories: global or local thresholding.

Global methods use only one value to classify all pixels on the image, regardless of whether the area it belongs has more or less noise. Values higher than the threshold become white, while lower values become black. By using only one threshold, global methods tend to be easier to implement and computationally cheaper. However, noises that occur in only one part of the image will influence the decision-making algorithm, which can lead to undesirable results.

To work around this problem, the local thresholding methods work with input image subsets, calculating the optimal threshold by region. Higher adaptivity are achieved by local methods, by allowing the decision-making in a region depend only in it, regardless of it neighborhood. Better results are expected on cases where different noises appear on different areas of the input image, but at a higher computational cost.

As the project's target scores have artifacts like sweat marks and candle drippings, which does not occur throughout the area, local methods tend to be more suitable for the Copista.

In this step then, the set of filtering techniques to remove different noises and to efficiently threshold input images should be evaluated. The evaluation of the results can be accomplished through a standard music content, which is already known, together with the next step of Copista.

2.3 Recognition of Musical Symbols

The step of Recognition of Musical Symbols employs computer vision techniques in certain specific steps:

1. Define meaning areas as staves
2. Clean the meaning area to only objects of interest
3. Definition of descriptors for each object
4. Classification of all recognize objetcs

The segmentation step [13] allows to separate elements such as lines and other notations to be trained. The lines can still be used to define the location of a notation. For example, the height of the notes according to their position in relation to the lines separating different overlapping symbols [4] and of different sizes or rotated positions [21].

Each notation can be described by a set of features [18]. Each feature may represent something of the image to be recognized as edges, curvatures, blisters, ridges and points of interest. The features extracted are then used in a pattern recognition process [11,27], after being qualified and quantized to a statistical analysis by a filter to reduce uncertainty between the heights of notes or artifacts present in input images.

This step can use a Kalman filter [19] that will allow the correction of data generated by the features extraction. By combining computer vision techniques in OMR, there is a higher gain for generating such data, ensuring the integrity and fidelity to that which is present in the document.

In addition, computer vision techniques used for other applications such as character recognition [9], handwriting recognition [33], augmented reality with low resolution markers [12] can also be used to complete this process step.

2.4 Music Notation Reconstruction

In the OMR process, the reconstruction stage of symbolic representation should receive data from the computer vision and map them to an alphabet of musical symbols. This mapping may include the validation of a given symbol or a set of symbols to aid the recognition step as to the correctness of a given graphical element with an analysis from the notational model[14] or based on a musical context[20]. The validation may occur by creating a set of lexical, syntax and / or semantics rules, that define the symbolic representation format.

A major issue of defining a symbolic musical representation is to find a sufficient generic representation, very flexible but at the same time restricted in relation to its rules to allow a validation of the musical structure as a whole[30].

Most of the existing models is part of a hierarchical musical structure[7] where there is an overview of the music divided into several staves (lines), which are divided into bars and these bars time to time and notes. For this project, it will be added to the model an even deeper hierarchy which will include information on the scores and the page of the score. A computational possibility to achieve such representation is to use an object-oriented model [32], to define the representation of a set of objects with attributes valued.

Such valued attributes should store the musical notation of a symbol as well as register symbol information within the image. For this reason, we divide the musical symbolic representation for OMR in two parts, one that represents the music information and another that represents the image information.

The valued data of the original image that was found a musical symbol are necessary to allow a reassessment of erroneously recognized data. This would request the computer vision to remade a given symbol validation conference automatically.

Other original image data may be stored relate to the initial processing made in the image. Information such as brightness, contrast, color, rotation, translation, histogram and what steps were performed to remove the artifacts becomes necessary for preprocessing can be adjusted by changing these parameters in an attempt to improve the quality of page reading.

2.5 Output

This last step generates a file representing the original score using a Symbolic representation. The definition of the symbolic representation format is a critical task in the development of this tool. This setting will influence the tool development since the validation of recognized symbols in the representation model can assist the learning algorithm of computer vision stage and thus reduce the need for human intervention in the process of transcription of digitized music.

The output of the tool should be as interoperable as possible in order to allow any possibility of editing and human intervention to correct a generated score, if this is necessary. Human correction performed in a score with identification problems can serve as a new entry in the system as it would enable a new learning step for the proposed algorithms.

The evaluation of adaptation takes into account a) the symbols used in these scores b) the hierarchical computational representation of this set of symbols, c) the lexical, syntactic and semantic rules to allow scores correction in the symbolic format and d) converting this set of symbols to commonly used formats in musical applications.

3 Partial Results

The recognition process of musical scores is done through steps that include image preprocessing (removal of possible noise and artifacts), segmentation (separation of elements in the images), detection, classification and recognition of musical elements.

This functionality separation created a chain of processes that may be changed individually based on successes and errors. Based on this chain, our first implementation separated each step of processing independently allowing each part to use a different programming language and exchanging data through file exchange.

Next, we present separated outcome from every phase of our process chain.

3.1 Image acquisition

The first issue faced during the Image Acquisition step regards the paper size. The music sheets are bigger than A4 sheet, so they do not fit in a regular table scanner. Moreover, considering the popularization of smartphones with high-resolution cameras, we decided to establish a camera-based setup for image acquisition. Consequently, the generated dataset is built taking into the account the use of the proposed approach in a more dynamic environment, leveraging from commonly available new technologies.

Nevertheless, it is also important to identify accurately the page borders and contours in order to verify how consistent the dataset is. Therefore, the image acquisition step uses a set of predefined rules to scan like keep image proportion, scan all files with the same distance, use the same chroma key under music files, and scan both side of paper independently if there are information on both side. Fig. 4 illustrates the built setup to accomplish the image acquisition.

Fig. 4: Image acquisition setup

The Acquisition phase generates image files to the Preprocessing phase. These file libraries are also considered a first outcome of the project because we keep original data as it is.

3.2 Preprocessing

The input of the preprocessing phase is the acquired digital image. This step prepares the image for computer vision process. For this step, initially the input file pass through a crop algorithm, to eliminate the area outside the score. This is done to erase the chroma key area outside the paper. After that, next step involves detecting holes inside score and classify the images according to the size of their most visible defects. Handling efficiently the holes are the hardest challenge on preprocessing. So, after the crop, these holes are measured using a connected components detection algorithm, using the easily spotted chrome key color to find the components.

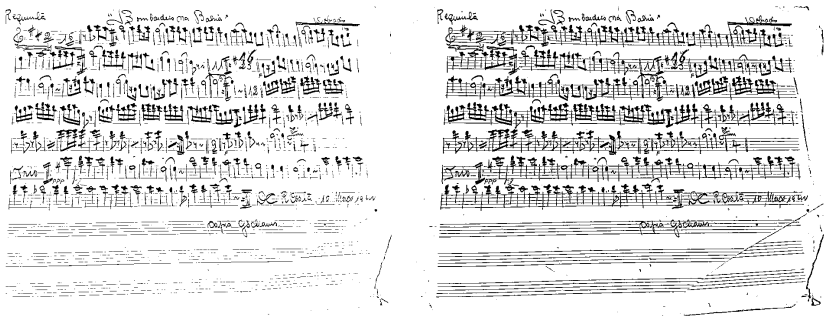
With all holes measured, one can classify the scores according to the degree of degradation suffered. Scores with higher count of holes or with bigger holes are classified as *highly damaged*. Smaller holes classifies the input score as *mild damaged*. Finally, if the scores has minimum holes or no holes, it is classified as *no damaged*. Thus, it is possible to analyze if the score has to pass through all preprocessing steps or if it can skip some. In this initial stage, only the scores classified as *no damaged* are being processed, while the team investigates how to handle the holes with context-aware filling techniques. This classification is also a partial outcome and can help to evaluate how damage is a collection.

After classification, the scores are converted to grayscale and after that, the image contrast is increased using histogram equalization. The high contrast increases the difference between the shades in each region and help the binarization to better decide if each pixel is background or object. Fig. 5 show the results of same method, with and without histogram equalization. Using histogram equalization allowed to erase less information from the image, keeping almost all the lines.

Using histogram equalized inputs, three binarization algorithms have already been tested: Niblack, Savuola and Wolf. All three methods works locally and the results are shown in Fig. 6. These are the final images in this stage of the Copista flow, and will be the inputs for the next step.

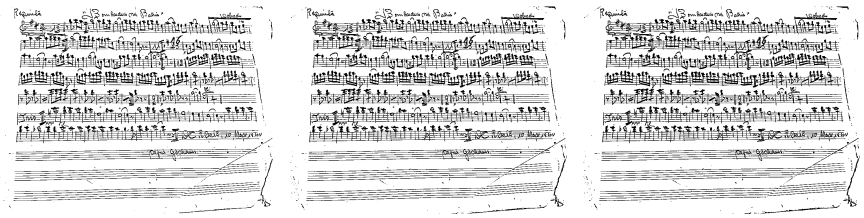
3.3 Recognition and Description

The initial algorithms on Recognition step used image comparison to identify the music elements on the score. To ensure an initial sample of elements, a set of non-manuscript figures was used in our preliminary tests. We choose to use non-manuscript scores despite the fact of these images have good contours and a predictable shapes. Since we did not use scanned files in this stage, we did not use preprocessing in our initial tests. After these tests, it will be possible to use our algorithms on the target collections, adapting the algorithms if necessary.



(a) Niblack without histogram equalization (b) Niblack after histogram equalization

Fig. 5: Difference between binarization without and with histogram equalization



(a) Niblack (b) Savuola (c) Wolf

Fig. 6: Tested methods with same score

We started the elements recognition, performing a search for the staves on the image. The staves are considered the meaning area on this step since its location can be used to delimit the boundary of notes and marks. To discover the staves we used the pixels projection of the image, depicted in Fig. 7

The staves are defined as pentagrams, which are five peak at the graphic, representing the five lines in each stave [34]. As it is possible to have notes and other graphical elements above or below the staves, we considered as our meaning area an vertical extension of the staves, as presented in Fig. 8.

After the definition of the staves, the five lines used to define the pentagrams are eliminated from the image, taking care to not remove or damage any element located over the line. Once the line is removed, it is easier to search for objects on the staff, as notice in Fig. 9

Once we have a score without lines, algorithms to recognize objects is applied to find music notes and marking. These algorithms will detach every element of the stave for future recognition, description and classification as illustrated in Fig. 10.

The detached elements will have an associated value as a unique identifier during the recognition process. The background image is displayed with the

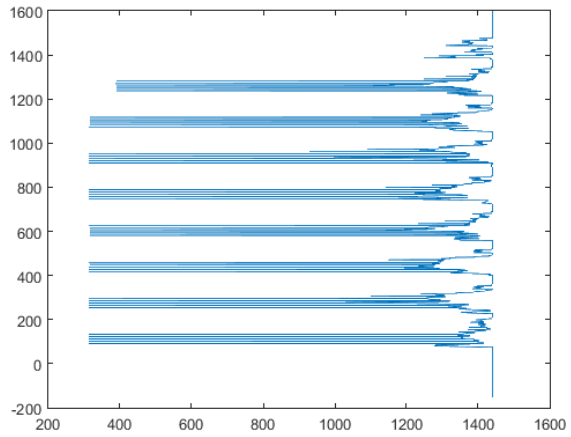


Fig. 7: Pixels Projection

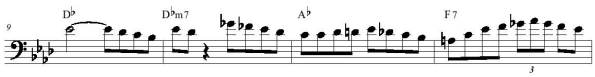


Fig. 8: The staves as the meaning area of the document

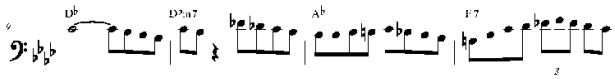


Fig. 9: Stave without lines

Fig. 10: First object recognized in a stave

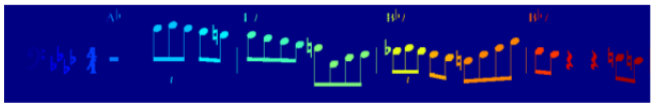


Fig. 11: Score with labeled elements

smallest value (in Fig. 11, the value is equal to 0) and so on. Thus, if the image has 200 elements, the last element will be labeled 199.

Fig. 12: Elements found in score image: (a) first, (b) sixteenth, (c) nineteenth e (d) thirty-fifth

In separation step of the detected elements, each element is clipped from input image and its pixels normalized to 1 (white) for object and 0 (black) to background, as shown in Fig. 12.

The background is left in black, because for this stage is used the invariants Hu moments, where the description of each element is made. Hu moments are based on invariant moments (non-variance to scale, translation and rotation)[35]. Hu moments are a vector of features of the image. This vector can be used to compare two graphical elements and identify an unknown object based on a known object from a dictionary. The first Hu moment, for example, provides the center of the element. One advantage of using Hu moments is that the element may be on different scales (displayed larger or smaller) or different positions, rotated or mirrored on the image.

The Recognition activity output is a text file containing a list of valuable elements identified on the staff with its location and other features value like size, identification, precision of identification process and so on.

3.4 Reconstruction

The input data of our reconstruction process is a textual file containing information about every recognized element of the original sheet. We created a Object-oriented model to represent the original document that includes the musical data of the original document and the image information about the original document. Thus, it will be possible to evaluate each score element based on their image. Our class diagram is depicted in Fig.13.

These class representation would help us to represent the recognized data and also validate it. The data validation can use some compilers techniques like a Syntax Analyzer to verify several features like: a) an accident or a dynamic symbol is not used before a rest, b) the sum of note times in a section should not be bigger than it could, c) it is not normal to have a clef or a time signature in the middle of a section, d) a natural symbol is not used in a line or space that is not changed with sharp or flat, d) a del segno symbol must be used with a segno symbol. All these validation are not a rigid rule but a clue that maybe something is wrongly recognized. Some of these rules can be implemented using a free context grammar, like the position of a clef in the section, and some must use an attribute grammar, like the sum of note times in a section.

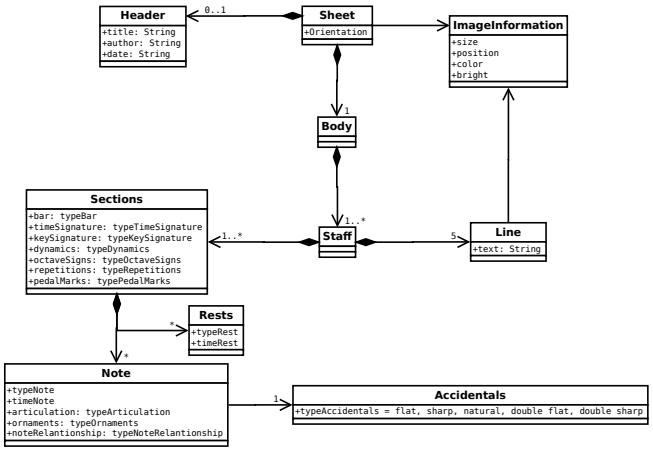


Fig. 13: Object-oriented representation of a Symbolic Music

Another important aspect of our Object model is the possibility to convert it into a common Symbolic Music format file. Next section will present a list of researched formats that can be used to this task.

3.5 Output

The tool output must be compatible with some existent tool to allow score editions and corrections. For this reason, we listed several Symbolic Music Notation file formats that could aim a good output choice.

The researched file formats that can be used as an output format are:

- ABC[23]
- MusicXML[14]
- Lilypond[22]
- Music21[1][8]
- GUIDO[17]

All these formats are ASCii and are input file format for several Score Editors. Also, there are several tools to convert one format to other and they are a kind of interchangeable music formats. We also researched other formats like MIDI[2] and NIFF (Notation Interchange File Format)[15] that were discarded since they use a binary file format.

4 Conclusion

This project triggered the joint research collaboration from different areas of Computer Science like Computer Vision, Image Processing, Computer Music, Artificial Intelligence and Compilers. The union of these areas should help the

development of the desired tool in the project and bringing gains for interdisciplinary research in the area of Computer Science. In addition to collaborating as interdisciplinary research in science, the project will also assist in the area of music creating an open-source tool for recognition and rewriting scores.

The first steps of this project involved the research of techniques and computational tools to be used in each step of Copista flow. The survey of these algorithms allowed preliminary tests in every planned activity with good initial results. The next steps of the project should merge the raised techniques and codes through individual steps of this research in a first functional prototype. Possibly, this first prototype will still work with digital music and non-handwritten for training recognition of a neural network to be used for decision-making in relation to the correctness of an identified symbol.

Another step that should be taken soon is to integrate the data representation with the Computer Vision step and to verify all elements identified by a symbolic music compiler. This step should also assist in the training tool, being another step in seeking a more suitable result for the proposed objective.

Referências

1. Ariza, C. and Cuthbert, M. (2011). The music21 stream: A new object model for representing, filtering, and transforming symbolic musical structures. Ann Arbor, MI: MPublishing, University of Michigan Library.
2. Association, M. M. et al. (1996). The complete MIDI 1.0 detailed specification: incorporating all recommended practices. MIDI Manufacturers Association.
3. Avidan, S. and Shamir, A. (2007). Seam carving for content-aware image resizing. InACM Transactions on graphics (TOG), volume 26, page 10. ACM.
4. Bainbridge, D. and Bell, T. (1997). Dealing with superimposed objects in optical music recognition.
5. Bainbridge, D. and Bell, T. (2001). The challenge of optical music recognition. Computers and the Humanities, 35(2):95–121.
6. Bernsen, J. (1986). Dynamic thresholding of gray-level images. In International Conference on Pattern Recognition.
7. Buxton, W., Reeves, W., Baecker, R., and Mezei, L. (1978). The use of hierarchy and instance in a data structure for computer music. Computer Music Journal, pages 10–20.
8. Cuthbert, M. S. and Ariza, C. (2010). music21: A toolkit for computer-aided musicology and symbolic music data.
9. Dori, D., Doerman, D., Shin, C., Haralick, R., Phillips, I., Buchman, M., and Ross, D. (1996). Handbook on optical character recognition and document image analysis, chapter the representation of document structure: a generic object-process analysis.
10. Fujinaga, I. (2004). Staff detection and removal. Visual perception of music notation: on-line and off-line recognition, pages 1–39.
11. Fukunaga, K. (2013). Introduction to statistical pattern recognition. Academic press.
12. Furht, B. (2011). Handbook of augmented reality. Springer Science & Business Media.
13. Gonzalez, R. C., Woods, R. E., and Eddins, S. L. (2004). Digital image processing using MATLAB. Pearson Education India.

14. Good, M. (2001). Musicxml for notation and analysis. The virtual score: representation, retrieval, restoration, 12:113–124.
15. Grande, C. (1997). The notation interchange file format: A windows-compliant approach. In Beyond MIDI, pages 491–512. MIT Press.
16. Hewlett, W. B. (1997). Beyond midi. chapter MuseData: Multipurpose Representation, pages 402–447. MIT Press, Cambridge, MA, USA.
17. Hoos, H. H., Hamel, K. A., Renz, K., and Kilian, J. (1998). The guido notation format – a novel approach for adequately representing score-level music.
18. Koendrik, J. J. (1992). Computational vision (book). Ecological Psychology, 4(2):121–128.
19. Laia, M. A. d. M. (2013). Filtragem de Kalman não linear com redes neurais embarcada em uma arquitetura reconfigurável para uso na tomografia de Raios-X para amostras da física de solos. PhD thesis, Universidade de São Paulo.
20. Medina, R. A., Smith, L. A., and Wagner, D. R. (2003). Content-based indexing of musical scores. In Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '03, pages 18–26, Washington, DC, USA. IEEE Computer Society.
21. Mundy, J. L., Zisserman, A., et al. (1992). Geometric invariance in computer vision, volume 92. MIT press Cambridge.
22. Nienhuys, H.-W. and Nieuwenhuizen, J. (2003). Lilypond, a system for automated music engraving. In Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003), volume 1. Citeseer.
23. Oppenheim, I., Walshaw, C., and Atchley, J. (2010). The abc standard 2.0.
24. Otsu, N. (1975). A threshold selection method from gray-level histograms. Automatica, 11(285-296):23–27.
25. Pinto, T., Rebelo, A., Giraldi, G., and Cardoso, J. S. (2011). Music score binarization based on domain knowledge. In pattern recognition and image analysis, pages 700–708. Springer.
26. Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marcal, A., Guedes, C., and Cardoso, J. (2012). Optical music recognition: state-of-the-art and open issues. International Journal of Multimedia Information Retrieval, 1(3):173–190.
27. Ripley, B. D. (1996). Pattern recognition and neural networks. Cambridge university press.
28. Sauvola, J. and Pietikainen, M. (2000). Adaptive document image binarization. PATTERN RECOGNITION, 33:225–236.
29. Seixas, F. L., Martins, A., Stilben, A. R., Madeira, D., Assumpção, R., Mansur, S., Victer, S. M., Mendes, V. B., and Conci, A. (2008). Avaliação dos métodos para a segmentação automática dos tecidos do encéfalo em ressonância magnética. Simpósio de Pesquisa Operacional e Logística da Marinha SPOLM.
30. Selfridge-Field, E. (1997). Beyond codes: issues in musical representation. In Beyond MIDI, pages 565–572. MIT Press.
31. Szwoch, M. (2007). Guido: A musical score recognition system. In icdar, pages 809–813.
32. Travis Pope, S. (1996). Object-oriented music representation. Organised Sound, 1(01):56–68.
33. Xu, L., Krzyzak, A., and Suen, C. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. Systems, Man and Cybernetics, IEEE Transactions on, 22(3):418–435.
34. A. G. S. e Thiago Margarida, Reconhecimento automatco de smbolos em partituras musicais

35. M.-K. Hu, Visual pattern recognition by moment invariants, InformationTheory, IRE Transactions on, vol. 8, no. 2, pp. 179187, 1962.