LUCIANO ROSI MENGARELLI

A NOVEL APPROACH FOR THE COMPARISON OF DIFFERENT OPTICAL MUSIC RECOGNITION METHODS

Thesis presented to the Graduate in Informatics (PPGIa) of the Pontifícia Universidade Católica do Paraná (PUCPR) as a partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Curitiba - PR, Brasil

2023

LUCIANO ROSI MENGARELLI

A NOVEL APPROACH FOR THE COMPARISON OF DIFFERENT OPTICAL MUSIC RECOGNITION METHODS

Thesis presented to the Graduate in Informatics (PPGIa) of the Pontifícia Universidade Católica do Paraná (PUCPR) as a partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Concentration Area: Data Science / Knowledge Discovery and Machine Learning

Supervisor: Carlos N. Silla Jr.

Curitiba - PR, Brasil

2023

LUCIANO ROSI MENGARELLI
A Novel Approach for the comparison of Different Optical Music Recognition Methods
LUCIANO ROSI MENGARELLI. – Curitiba - PR, Brasil, 2023-
102 p. : il. (algumas color.) ; 30 cm.

Supervisor: Carlos N. Silla Jr.

Tese –
Pontifícia Universidade Católica do Paraná (PUCPR)
Graduate Program in Computer Science - PPGIa, 2023.

1. OMR. 2. Evaluation. 3. Ground Truth. 4. Comparison.

LUCIANO ROSI MENGARELLI

A Novel Approach for the comparison of Different Optical Music Recognition Methods

Thesis presented to the Graduate in Informatics (PPGIa) of the Pontifícia Universidade Católica do Paraná (PUCPR) as a partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Approved Work. Curitiba - PR, Brasil, December 5, 2023:

Carlos N. Silla Jr.
Orientador(a)

Professor(a)
Convidado 1

Professor(a)
Convidado 2

Curitiba - PR, Brasil
2023

To my children, Rhuan Pablo dos Santos Mengarelli and Gabriel Furlan Mengarelli.
To my mother, Angela Maria Nardelli Rosi, still taking care of me.
And to my father, Hugo Daniel Mengarelli, that unfortunelly leaved us before I could finish this work.
Everything done with love is a great legacy.

Acknowledgements

I would like to thank my advisor, Carlos Silla Jr., for all the support and for suggesting the theme, allowing me to unite two great passions: music and computers.

I would like to thank Júlio Nievola, Ricardo Cerri and Alceu Britto for their insightful suggestions during the Doctoral qualification, improving this document and helping me organize my ideas.

I would like to thank my brother, Rodrigo Rosi Mengarelli, for his patience in moments of distress and, also, his advises and friendship.

I would like to thank my love, Michele Roberto Lima, for never let me give up and always see the best of me, even when I couldn't.

And I could not forget to thank for the exemption of financial fees provided by the Araucaria Foundation and the Coordination for the Improvement of Higher Education Personnel (CAPES).

"But we cannot do it all at once; it is a sequence. An unfolding process.
We can only control the end by making a choice at each step."
Philip K. Dick, The Man in the High Castle

Abstract

Music is rhythm, timbre, tones, intensity and performance. Conventional Western Music Notation (CWMN) is used to generate Music Scores in order register music on paper. Optical Music Recognition (OMR) studies techniques and algorithms for converting music scores into a readable format for computers. Despite a large number of works studying OMR techniques and more than 50 years of research, the evaluation of OMR results is still an open issue. This Doctoral Research proposes the first automated method to compare the final results of different OMR systems. The method works using MusicXML files and it was evaluated using a ground-truth dataset and two different OMR Systems. A full systematic literature review is conduced and presented for OMR Evaluation. Alignment algorithms are also described and adapted to handle the challenges presented. The method, dataset and all files used in the experiments will be made freely available on GitHub together with this work's publication. Therefore, the research community will be able to use and improve it. Experiments were conduced with the provided dataset. Overall meaningful results and detailed analysis are presented showing a possible way to automatically evaluate the performance of OMR systems.

Keywords: OMR, Evaluation, Comparison.

List of Figures

List of Tables

List of Algorithms

List of abbreviations and acronyms

CWMN	Conventional Western Music Notation
CNN	Convolutional neural network
FN	False Negatives
FP	False Positives
ICDAR	International Conference on Document Analysis and Recognition
JSON	JavaScript Object Notation, compact format for data exchange
MIDI	Musical Instrument Digital Interface, file format with musical information
MusicXML	File standard designed to allow the storage of music score documents. Specification, schemas and DTDs for MusicXML can be found online on <https://www.musicxml.com/for-developers/>.
OMR	Optical Music Recognition
PDF	Portable Document Format, independent document format developed by Adobe Systems
PNG	Portable Network Graphis, image file type
TN	True Negatives
TP	True Positives

Contents

1 Introduction

Music is rhythm, timbre, tones, intensity. Musicians, using their instruments, should follow a list of steps to play a song. Conventional Western Music Notation (CWMN) is used to compose documents called "Music Scores" (figure 1) in order to describe those steps allowing historical register and music sharing (MENGARELLI et al., 2020). This notation is formed by symbols that indicate the tones, the duration and the way to perform them (MODAYUR et al., 1993a).

Music scores are usually written on paper and can deteriorate over time. That's why there is a great concern in saving and restoring them. The digitalisation has been used as a solution to preserve music scores, also facilitating the distribution and digital processing of them. "The development of general image processing methods for object recognition has contributed to the development of several important algorithms for Optical Music Recognition (OMR)" (REBELO et al., 2012). Since the mid 1960s, OMR has been a research area that study techniques and algorithms for converting or interpreting the music score into a readable format for computers (CHEN; CHEN; TENG, 2013). The main goal is to teach computers how to read music(CALVO-ZARAGOZA; JR; PACHA, 2020). Machine learning algorithms are also widely used for recognizing symbols as neural networks, decision tree, random forest and SVM(MENGARELLI et al., 2020).

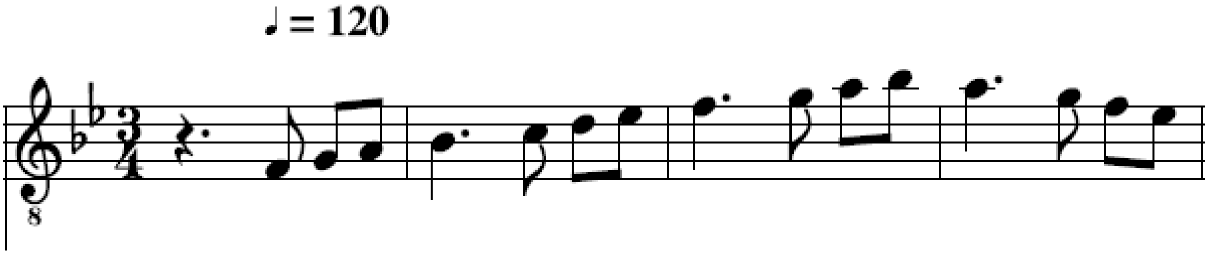



Figure 1 – Music Score Sample

1.1 Justification

According to (CALVO-ZARAGOZA; JR; PACHA, 2020) "for over 50 years, researchers have been trying to teach computers to read music notation, referred to as Optical Music Recognition (OMR)." The development of an OMR system presents several challenges, as we shall see in further detail in Chapter 2.

Another important limitation of OMR research is how to effectively compare the output results of different OMR systems. As noted in (BYRD; SIMONSEN, 2015), one "question of interest to quite a few people is: what is the most accurate OMR system, at least for a specific collection of page images? Unfortunately, the answer is 'No one knows, and there's no practical way to find out'". (BYRD; SIMONSEN, 2015) also discussed the issue concerning the lack of automatic OMR evaluation methods: "Thus far, every evaluation of OMR system accuracy we know of has been done entirely by hand, a process that is itself error-prone as well as time consuming – and, of course, the larger the test collection, the more time consuming it is." This issue is also mentioned in (HAJIC JR et al., 2018): which states that the comparison of "existing systems is hard, because there are few symbol detection results reported, and even fewer full-pipeline OMR results". For (BUGGE et al., 2011) "Error counting in OMR is notoriously difficult and ambiguous".

OMR systems are still not perfect and can sometimes provide wrong results since the challenges are great in this area. The resulting file from an OMR system often needs to be manually analyzed and corrected in order to have a perfect correspondence with the original scanned music sheet. For example, in the work (BYRD; SCHINDELE, 2006) which considers pitch and duration, all errors were counted manually: "this confirmed the hand error count results showing that all of the programs did worse on more complex music".

Although the important corrections are related to music information, most of the metrics and comparisons only consider the algorithm performance, disregarding music information and the real impact of the differences in the musical context. Usually the results shows some kind of metric to indicate the number of correctly identified primitive symbols, similar to using machine learning for image classification. For example, in (WEN et al., 2015) an OMR system with CNN with no need of image segmentation is presented. The results consider only music symbols accuracy, precision and recall, but do not address music context: "Basically, the symbols from the same class are compared with each other. The symbols will be saved as one symbol if their positions are close enough.". The work of (YOO; KIM; LEE, 2008) presents recognition rate results only on musical notes classification: "we do not deal with non musical note classification.", also disregarding note pitch, rests, other symbols.

Byrd and Simonsen (BYRD; SIMONSEN, 2015) have proposed rules, definitions and metrics for a Standard Testbed for Optical Music Recognition, addressing also the counting of errors. It is an important step in an area that suffers from this lack for over 20 years. Their work addresses issues as: image quality level, complexity of notation, tightness of spacing and rules for error counting.

For (HAJIC JR, 2018) "The major problem in OMR evaluation is that given a ground truth encoding of a score and the output of a recognition system, there is no automatic method capable of reliably computing how well the recognition system performs that would (1) be rigorously described and evaluated, (2) have a public implementation, (3) give meaningful results."

Despite more than 50 years of research, the development of an OMR system still presents several challenges and there is no automatic method to compare the output results of different OMR Systems. "The major problem in OMR evaluation is that given a ground truth encoding of a score and the output of a recognition system, there is no automatic method capable of reliably computing how well the recognition system performs that would (1) be rigorously described and evaluated, (2) have a public implementation, (3) give meaningful results." (HAJIC JR, 2018)

Based on these works it is clear that there is a consensus within the OMR research community that there is still a need for the development of a computation-based approach that is able to compare the output results of different OMR Systems. Furthermore, this problem was also recognized in a recent paper by (CALVO-ZARAGOZA; JR; PACHA, 2020), since OMR Evaluation was listed as one of the Open Issues in the field.

The main contribution of this work is to propose a new automated method to evaluate and compare the output of different OMR systems. This method is to the best of the authors knowledge the first method that allows the automatic comparison of different Full OMR systems (MENGARELLI et al., 2020) (i.e. OMR systems that receives as input a visual image file of a score and outputs a musicXML file). The proposed method is able to deal with several Challenges (presented in Section 2.5) for the task. The method is described is detail in Chapter 4. To evaluate the proposed method (Chapter 5) we have compared the performance of two OMR systems in the Synthetic Score Databased created by Christoph Dalitz (DALITZ; MICHALAKIS; PRANZAS, 2008). In Section 6 we present the final considerations about the proposed method as well as future research steps.

1.2 Objectives

The main objective of this work is to propose a new method for the evaluation of OMR systems considering the musical context in which they are inserted.

In order to achieve this goal, the following specific objectives are considered:

1. To establish (or develop/adapt) a dataset with ground-truth examples in MusicXML format.

2. To develop a novel computational method that is able to compare the MusicXML output of different OMR systems.

3. To provide the full computational system or framework to help other researchers to evaluate the output of their OMR systems.

1.3 Hypothesis Statements

Hypothesis 1. It is possible to make an automated computational method to compare OMR systems, considering important transcription elements beyond primitive symbols counting.

1.4 Scientific and Technical Contributions

The main scientific contribution of this work is a novel method for the comparison of different OMR systems using MusicXML data.

Along this work we have conducted a literature review presented in chapters 2 and 3. That review was published as a Journal Paper in the Multimedia Tools and Applications Journal in 2020 (MENGARELLI et al., 2020).

Also the method, algorithms and experiments described in chapters 4 and 5 were compiled and are under review in the Multimedia Tools and Applications Journal.

We addressed a field issue that remains open for many years, providing a complete analysis of the challenges and the research area. Experiments were performed and empirical evidences were collected. We also edited a ground-truth dataset of MusicXML files based on the Synthetic Score Databased created by Christoph Dalitz (DALITZ; MICHALAKIS; PRANZAS, 2008).

As Technical contributions, the software implementation of the proposed method is made freely available for the research community, as well as any datasets and ground truths developed during this research. The software and the datasets are available at this link [1].

Considering that the source code and algorithms of the method are freely available, the method can evolve and reach even better results with the help of, and also helping, the research community.

[1] https://github.com/mengarelli/omr-comparison

On chapter 4, the implementation and "Sophistication" of the General Method for Sequence Comparison(NEEDLEMAN; WUNSCH, 1970) can also help another works on aligning more complex structures.

1.5 Chapters Organization

This work presents a research proposing a solution for the evaluation and comparison of OMR methods. In order to establish this solution, this document was organized in chapters. Chapter 2 describes how OMR systems work, the most commonly used methods, their metrics, issues and challenges. Chapter 3 presents a systematic literature review on metrics and methods of evaluation, also comparing OMR systems and algorithms, needed for the understanding of the work and the proposed solution. Chapter 4 explains the proposed method for evaluation of OMR and presents algorithms, alignment algorithms adaptions and simulaion scenarios results. Chapter 5 presents the experiments conduced with the groundtruth files based on the Synthetic Score Databased created by Christoph Dalitz (DALITZ; MICHALAKIS; PRANZAS, 2008), also a detailed analisys of one music score. Chapter 6 shows the conclusion and discusses the next steps of this work.

2 Optical Music Recognition (OMR) Evaluation: Issues and Challenges

An OMR system (figure 2), including also the image pre-processing as part of it, can be divided in three stages(REBELO; CAPELA; CARDOSO, 2010):

1. Stage 1 (S1 - See figure 3): Image pre-processing (binarization, noise removal, blurring, deskewing, etc);
2. Stage 2 (S2 - See figure 4): Recognition of musical symbols from images. Usually also subdivided in:
   a) Staff lines detection and removal;
   b) Symbols primitives segmentation;
   c) Symbols recognition;
3. Stage 3 (S3): Reconstruction of the musical information in order to identify the music notation;
4. Stage 4 (S4): Construction of a model containing the symbolical representation of the music score.

One of the most important algorithms in the image pre-processing stage (S1) is the binarization. The scanned image is analyzed to determine what information belongs to the music score (musical symbols and lines) and what should be disregarded (background and noise), reducing the amount of information to be processed(REBELO et al., 2012).

The Staff Detection and Removal is a procedure with great impact for the OMR, since it isolates the music symbols through the removal of the staff lines of the score and can sometimes result in loss of information for the next steps of the process. For this

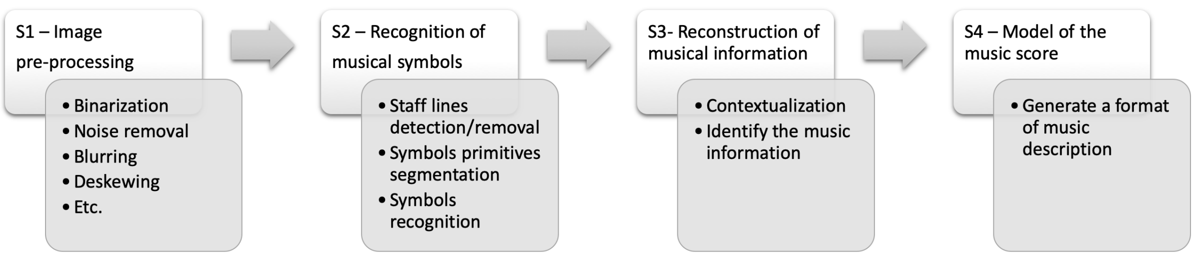


Figure 2 – OMR Stages (REBELO; CAPELA; CARDOSO, 2010)

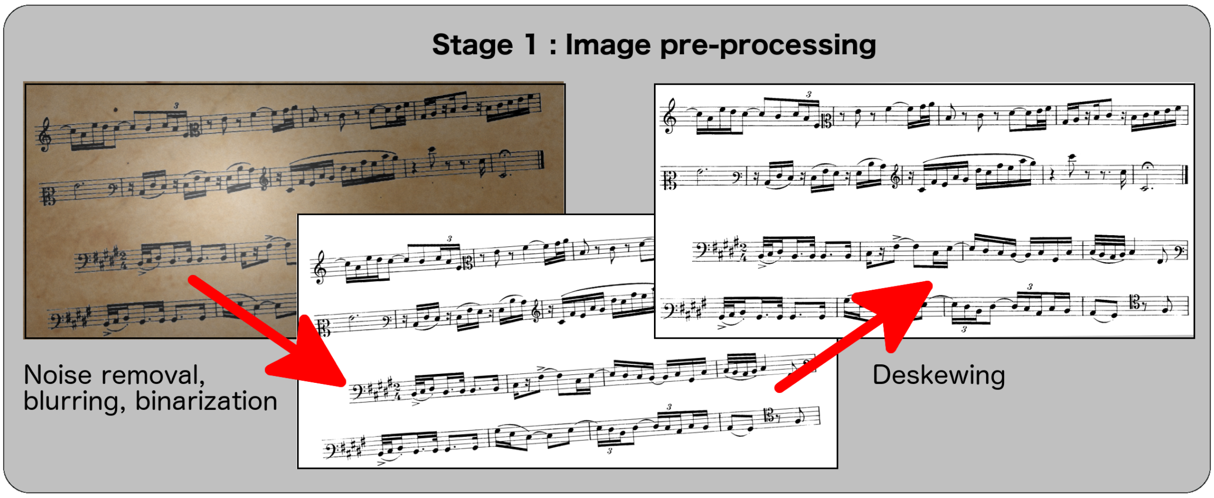


Figure 3 – OMR Stage 1(MENGARELLI et al., 2020)

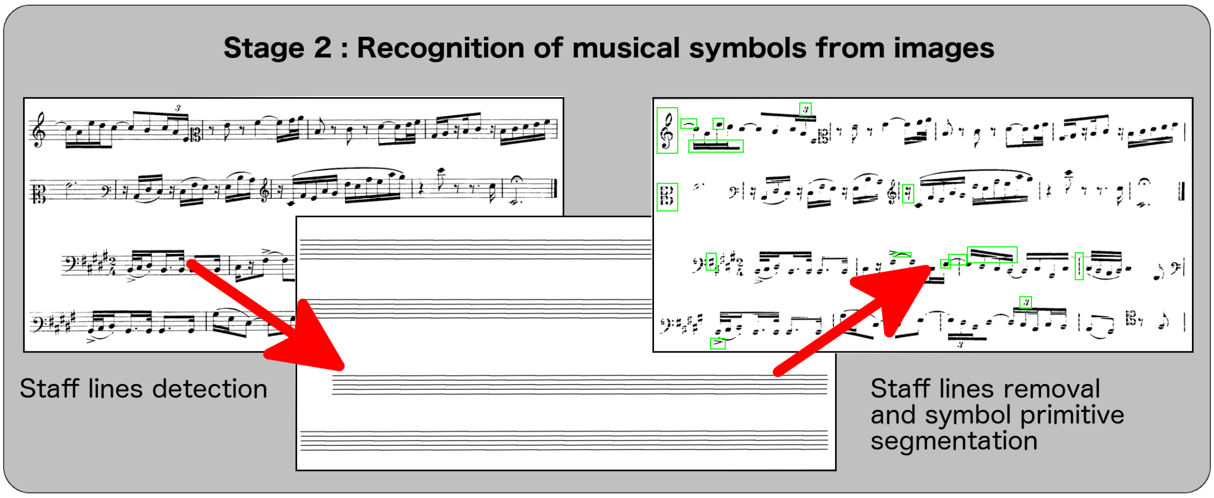


Figure 4 – OMR Stage 2(MENGARELLI et al., 2020)

reason, there are some authors who suggest OMR algorithms that do not remove the staff lines.(REBELO et al., 2012).

Segmentation and recognition of symbols extracts elementary graphic symbols like dots, rests, note heads and stems (see Table 1), usually using a classification task and occasionally with an alternative approach. All objects extracted are then classified again now considering size, bounding, linking, connections and other features in order to recognize higher level elements of the musical notation. There is a great variety of algorithms for the later task.(REBELO; CAPELA; CARDOSO, 2010).

The later two stages are often interconnected. Here is where graphical and syntactic rules are used to contextualize the information transforming the symbols recognized on the previous stage in music information. Once all the music information is produced, the last stage is responsible for generating a format of music description, like MIDI files and more recently MusicXML is the more often used format.(REBELO et al., 2012).

Table 1 – Music Symbols (BELLINI; BRUNO; NESI, 2007)

Category	Example
Empty Note Head	
Black Note Head	
Augmentation Dot	
Rest Duration 4/4	
Rest Duration 2/4	
Rest Duration 1/4	
Rest Duration 1/8	
Rest Duration 1/16	
Rest Duration 1/32	
Rest Duration 1/64	
Barline Single	
Barline Double	
Barline End	
Barline Start Refrain	
Barline End Refrain	
Slur	

Category	Example
Sharp	
Flat	
Natural	
Double Sharp	
Double Flat	
Treble Clef	
Bass Clef	
Tenor Clef	
Hook 1 (1/8)	
Hook 2 (1/16)	
Hook 3 (1/32)	
Hook 4 (1/64)	
Beam 1 (1/8)	
Beam 2 (1/16)	
Beam 3 (1/32)	
Beam 4 (1/64)	

Category	Example
Accent	>
Number 1	1
Number 2	2
Number 3	3
Number 4	4
Number 5	5
Number 6	6
Number 7	7
Number 8	8
Number 9	9
Piano	*p*
Forte	*f*
Comma	
C	
Staccato	
Fermata	
Mordent	
Turn	
Grace Note	
Trill	*tr*
Tenuto	

Currently there is a great deal of work on OMR, commercial and free OMR systems. The question is: Which are the best algorithms? Which full system is more accurate? The problem is that nobody knows this answer. Therefore, it is of great importance to evaluate OMR systems through metrics and comparisons.(BYRD; SIMONSEN, 2015).

2.1 Evaluation and Metrics - Current Scenario

Most analysis of OMR systems and their results are based on statistic metrics as accuracy, precision, recovery and error rate. There are some datasets available that might be used to test OMR systems, as shown on table 2.

However, there is not yet a set of standards and methods developed to aid the evaluation and comparison of OMR systems.

Table 2 – Datasets

Dataset	**Summary**	**Groudtruth**
The Synthetic Score Database by Christoph Dalitz	Historical scores, modern scores and tabs images are in PNG, 300dpi.	Yes
CVC-Muscima Database	Handwriten scores images (300dpi). 1,000 images for writer identification and 12,000 for staff removal, including deformations. Transcriptions from 50 different musicians writing the same 20 scores.	Yes
IMSLP - Petrucci Music Library	Library of musical scores, works and recordings from contemporary composers.	No
HOMUS Database	Handwritten musical score images (149ppi) drawn by 100 musicians including 32 musical symbols resulting a set of 15,200 samples in 32 models.	Yes
SNU Dataset for Online Music Symbol Recognition	Handwritten musical score images drawn by 18 musicians resulting a set of 1,716 samples.	Yes
Digital Scriptorium	Until September of 2017 there ware 8,133 manuscripts and 75,922 image with 34 participants from distinct locations.	No
MUSICNETWORK OMR Assessment	Seven printed scores images with three results of OMR systems each.	No
NEUMES Project	Medieval chants manuscripts.	Yes
(OMR-ChSR6306) OMR Chord Separation and Recognition 6306 Database	Database of 6,306 musical chords images captured from music scores using SamsungNote 2 camera. Destined to chord recognition.	Yes

2.2 Problem

OMR systems are still not perfect and can sometimes provide wrong results since the challenges are great in this area. The resulting file from an OMR system often needs to be manually analyzed and corrected in order to have a perfect correspondence with the original scanned music sheet. For example, in the work (BYRD; SCHINDELE, 2006) which considers pitch and duration, all errors were counted manually: "this confirmed the hand error count results showing that all of the programs did worse on more complex music".

Although the important corrections are related to music information, most of the metrics and comparisons only consider the algorithm performance, disregarding music information and the real impact of the differences in the musical context. Usually the results shows some kind of metric to indicate the number of correctly identified primitive symbols, similar to using machine learning for image classification. For example, in (WEN et al., 2015) an OMR system with CNN with no need of image segmentation is presented. The results consider only music symbols accuracy, precision and recall, but do not address music context: "Basically, the symbols from the same class are compared with each other. The symbols will be saved as one symbol if their positions are close enough.". The work of (YOO; KIM; LEE, 2008) presents recognition rate results only on musical notes classification: "we do not deal with non musical note classification.", also disregarding note pitch, rests, other symbols.

Although there are some databases that can be used to develop, test and compare existing OMR systems, a set of standards and methods has not been established yet. Most of the metrics are only statistical, but that is not the best scenario for OMR systems.

Despite there is no agreement about how to measure errors yet, some authors highlight the importance of considering errors with different weights. Bugge et al.(BUGGE et al., 2011) have presented a detailed work about error counting when prioritizing sound. Their work converts the MusicXML final result in a simpler version and uses a sequence alignment algorithm to compare results among several OMR systems with very specific error rules.

Byrd and Simonsen (BYRD; SIMONSEN, 2015) have proposed rules, definitions and metrics for a Standard Testbed for Optical Music Recognition, addressing also the counting of errors. It is an important step in an area that suffers from this lack for over 20 years. Their work addresses issues as: image quality level, complexity of notation, tightness of spacing and rules for error counting.

So how to develop a method and a computational system capable of allowing comparisons between full OMR systems considering the context of music?

Section 3.3.2 on chapter 3 describes 25 works with some kind of metric for full OMR systems. Each work uses it's own metrics for evaluation. For instance, (KATO; INOKUCHI, 1992) try to improve the evaluation using the repair count to calculate recognition rate, "The recognition rate is calculated from the count of modified and appended words in repairing the obtained result" (KATO; INOKUCHI, 1992). (BYRD; SCHINDELE, 2006) "ended up relying almost entirely on the hand error counts, plus expert opinions and documentation". Other works calculate the recognition rate or accuracy of primitive symbols (YADID-PECHT et al., 1996) (LUANGNAPA et al., 2012) (LUANGNAPA et al., 2012) (WU; JANG, 2014a) (PADILLA et al., 2014) (WU; JANG, 2014b) (WEN et al., 2015). (BUGGE et al., 2011) address some ambiguity and difficulty on counting errors. Last, but not least, each work uses a different dataset and only a few works provide the dataset.

2.3 Music Scores, Symbols and Rhythm

Music is a language that can cross cultural and geographic borders. Music scores are used to communicate and understand that language. Music score is a graphical representation of the music. In order to better understand some of the errors made by OMR software, it is important to understand some basic concepts that make a music score. In this chapter, the music scores, their musical elements and rhythm are explored to show the their crucial role on musical interpretation.

2.3.1 Music Scores: The music written language

Music scores are written documents containing execution information of a music composition. Those information are presented through musical elements as notes, rhythms, rests, dynamics and articulation.

The music notation has evolved over the history, being influenced by several music styles. The western music notation is widely used and is the focus of this study. The western music notation represents the music information on a pentagram, also know as staff, represented by five horizontal lines.(JONES, 2022)

2.3.2 Musical Elements on Western Music Notation

Music scores are composed of several music elements to provide information to musicians. Some of those elements are:

2.3.2.1 Notes

Notes represent the sound and are placed mostly on the lines or on the space between the lines of the staff. Higher pitch or lower pitch notes can be represented by drawing additional floating lines upper or lower the staff.

2.3.2.2 Rests

Just as notes represent sound, rests represent silence.

2.3.2.3 Rhythm

The shape of the note or rest tells how long it is, that is, it's duration.

2.3.2.4 Dynamics

The dynamics is used to describe the sound intensity variation, suggesting lower or higher intensity playing on different parts of the music.

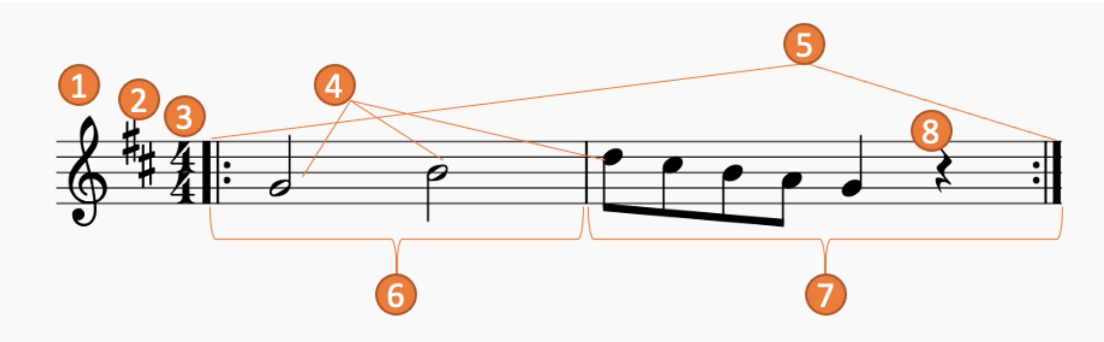
2.3.2.5 Articulation

Articulation refers to the way notes are connected with each other, for instance: staccato (short notes highlighted) or legato (soft connected notes).

2.3.3 The Basic Elements of a Music Score

Figure 5 illustrates an example with some of the most common symbols where:

Figure 5 – Basic Elements of a Music Score - Musical Symbols



1) G Clef; 2) Key signature; 3) Time signature; 4) Notes;
5) Repeat signs; 6) First measure; 7) Second measure; 8) Rest;

1. The Clef: The Clef indicates what are the notes in each of the spaces and lines of the music score. In this example, we are showing a standard G clef, which means that the second line (counting in a bottom-up fashion) of the measure is the note G.

2. The Key Signature: The Key signature indicates what is the harmonic field of the song by using accidentals like sharps (#) and flats (b). In this example, since we are using two sharps, it indicates that the song is in D Major or B Minor.

3. The Time Signature: The Time Signature is related with the duration of each measure, which impacts how many notes can be in it. In this example, we are using a time signature of 4/4.

4. Notes: There are different formats that can be used to indicate the duration of the notes, but the position of the note in relation to the clef, indicates which note should be played. In this example the notes are G, B in the first measure and D, C#, B, A, G in the second measure.

5. Repeat Signs: There are different symbols to represent them and they are commonly used in music notation to save page space and avoid writing the exact same notes two of more times. In this example the repetition sign used indicates that once the

second measure is played, the interpreter need to play the first and second measures again.

6. First Measure: As explained before, the measure duration is defined by the time signature and when the measure is full a vertical bar indicate the end of this measure and the beginning of the next. In this case the measure is two half notes long, equivalent to 4 quarter notes.

7. Second Measure: In this case, the end of the measure is not a simple vertical bar, because there is a repeat sign that indicates the end of the measure.

8. Rest: As with the notes there are different symbols to represent the rests according to their duration. In this example we are using a quarter note rest.

2.4 Preliminary Experiments

In this chapter we present some preliminary experiments that were performed with the goal of better understanding some of the common errors made by the OMR system and the technical aspects involved with dealing with MusicXML files.

2.4.1 Experimental Setup

For this preliminary experiment, we have used some music scores from the Henrique Pinto - Guitar Initiation Book (PINTO, 1978). The OMR system used to generate the MusicXML files was Musescore (MuseScore BVBA, 2018).

2.4.2 Experimental Results

For the analysis of these preliminary results, since no dataset with ground-truth in MusicXML was available, we have performed a manual error analysis. Figures 6 and 7 show some mistakes with different impacts on final results.

Figure 6 shows a problem detecting the time signature of the score. The letter "C" is equivalent to a 4/4 time signature, meaning that the measure should have a duration of 4 figures of quarter note. Instead, the OMR system detected 2/4, that is, 2 figures of quarter note. Also some triplets were misdetected because the confusion between "m" and "3", since "m" indicates the "middle finger" while "3" is an indication of triplet.

Figure 7 shows two errors with less impact on the final result. The first error is the last note missing in the third measure of the second line. This could be easily corrected with an editor since the alignment was not broken. The second error is the introduction of an articulation symbol, which indicates that the note should be played "staccato", that is, quickly. This could also be easily fixed with an editor.

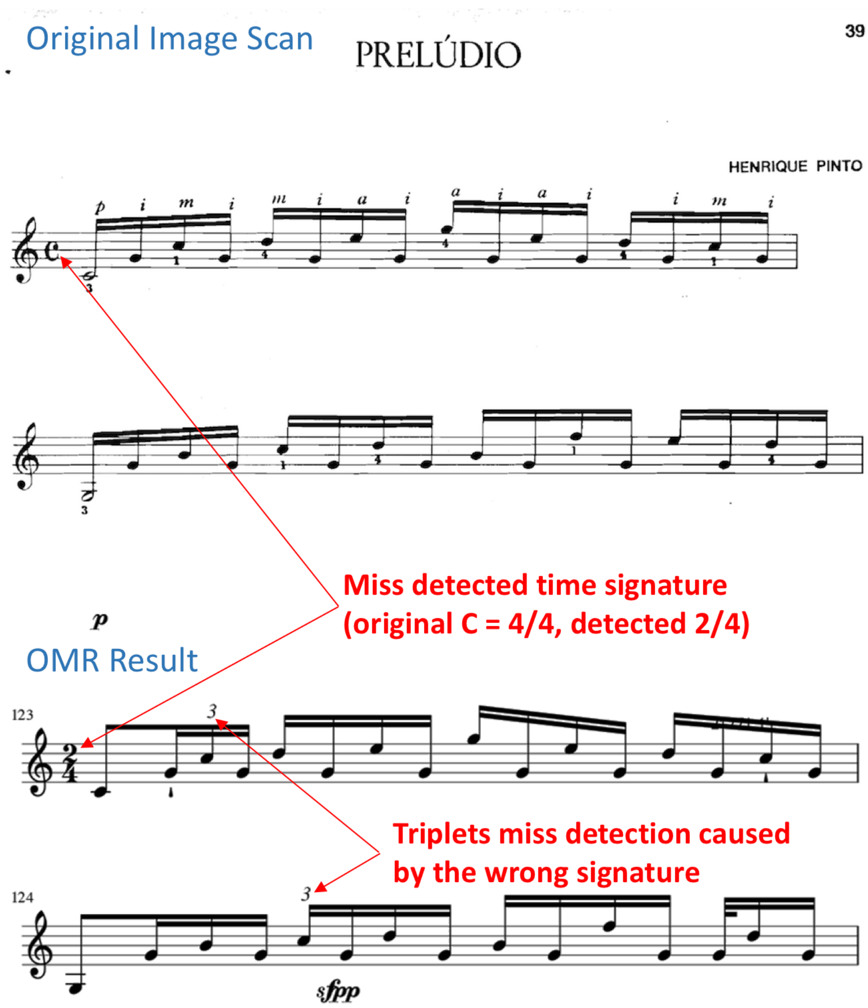


Figure 6 – Preliminary Experiments - OMR Mistakes Example 1(MENGARELLI et al., 2020)

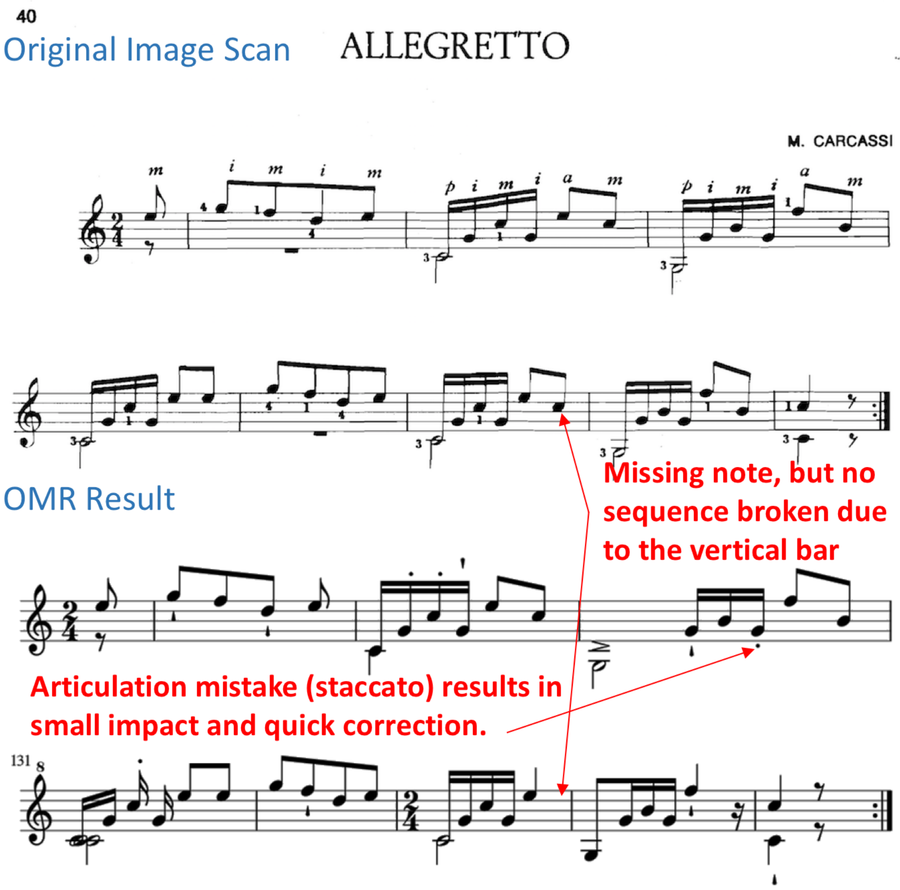
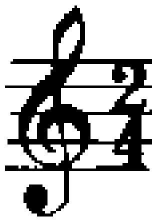
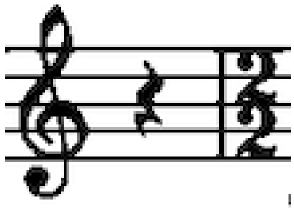


Figure 7 – Preliminary Experiments - OMR Mistakes Example 2(MENGARELLI et al., 2020)

Figure 8 – Preliminary Experiments - Time Signature Error

Original Score	OMR Result
	

These two examples, shows some of the analysis that were manually performed. After conducting this experiment, it was possible to identify different types of errors made by the OMR software.

Figure 8 shows a time signature error, where the original score has a time of 2/4 (i.e. duration of 2 figures of a quarter note) and the OMR system detected it as a time signature of 2/2 (i.e. duration of 2 figures of half note).

Figure 9 shows a note duration error, where the first note of the original score is a half note and the OMR system detected it as a whole note.

Figure 9 – Preliminary Experiments - Note Duration Misclassified

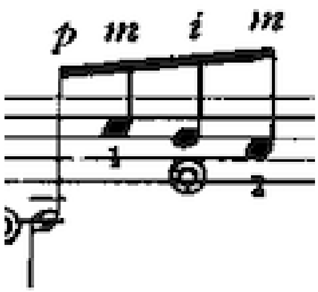
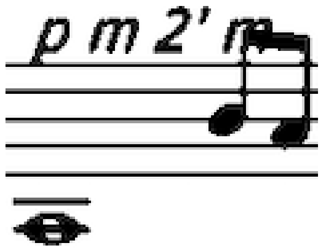
Original Score	OMR Result
	

Figure 10 shows a missing note that was not recognized by the OMR system. The ground truth has a half note and three eighth notes while the OMR system detected the half note but only two eighth notes.

Figure 11 shows a false positive situation, where the OMR system detects a note that does not exist. The ground truth has only one half note and three eighth notes while the OMR system detected the half note and four eighth notes.

Figure 12 shows another false positive situation where the finger numbers "1" and "3" were detected as a rest that does not exist.

Figure 10 – Preliminary Experiments - Note Missing

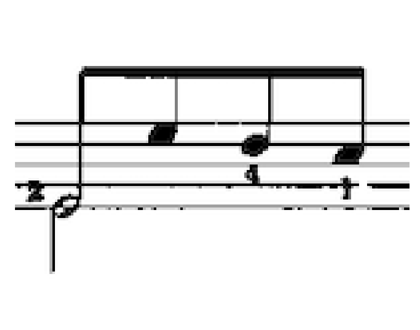
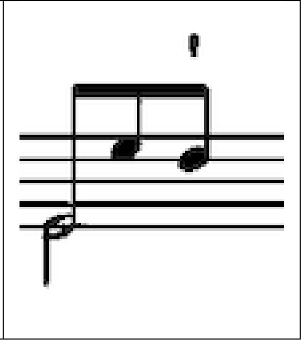
Original Score	OMR Result
	

Figure 11 – Preliminary Experiments - Non Existent Note

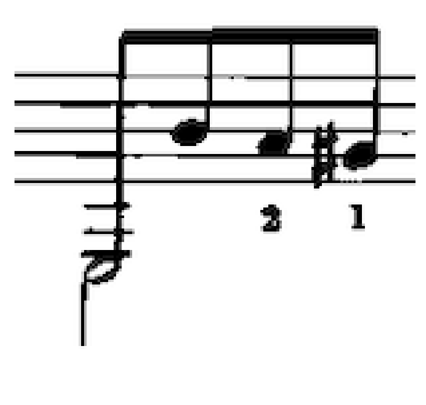
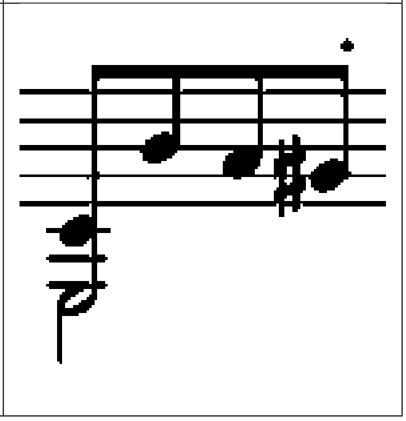
Original Score	OMR Result
	

Figure 12 – Preliminary Experiments - Non Existent Rest

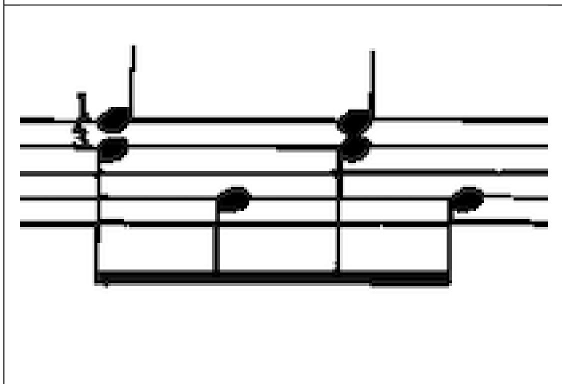
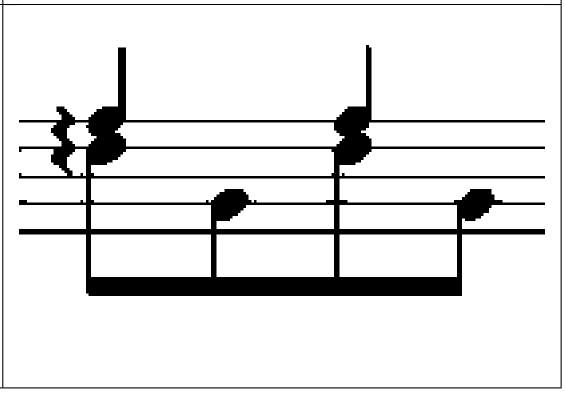
Original Score	OMR Result
	

Figure 13 shows a repetition text missing. "Fine" indicates the end of the repetition and was not detected by the OMR system.

In figure 14 the dot beside the note represents a "dotted note", that is, the note duration is increased by half. The OMR system detected two dots meaning that the note duration is increased by half and then by half of half, that is, a quarter.

Figure 15 shows a classification problem where the G clef on the ground truth is

Figure 13 – Preliminary Experiments - Missing Repetition Text

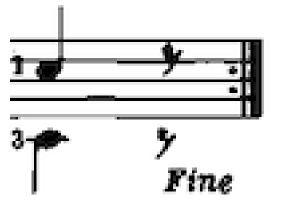
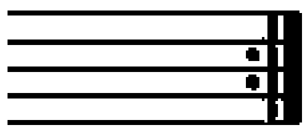
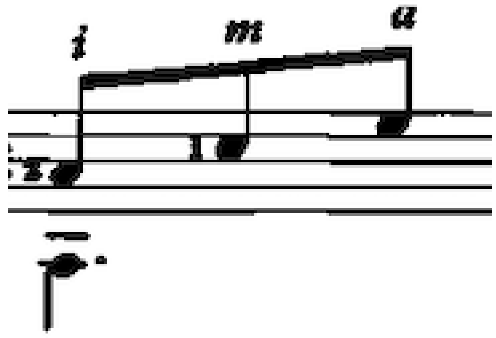
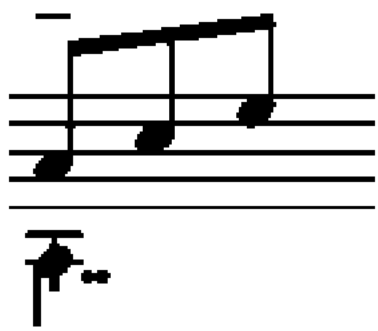
Original Score	OMR Result
	

Figure 14 – Preliminary Experiments - Non Existent Dot

Original Score	OMR Result
	

detected as a C clef by the OMR system. This will make all notes change for the score.

Figure 15 – Preliminary Experiments - Clef Misclassified

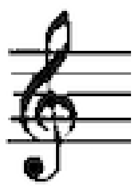
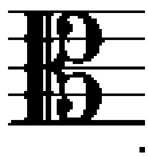
Original Score	OMR Result
	

Figure 16 shows another false positive where the OMR system detected a new time signature in the measure in the meddle of the score.

Figure 17 shows a missing accidental situation (the two sharp symbols denoted by # before the first set of notes) but the OMR system was not able to detect those signs.

And finally, figure 18 presents a similar error, but the OMR system failed to identify the sharp symbol in the key signature of the music score.

Figure 16 – Preliminary Experiments - Non Existent Time Signature

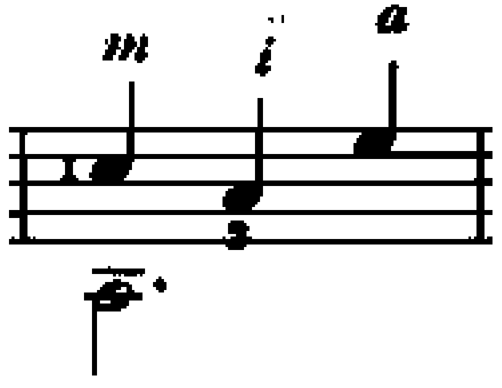
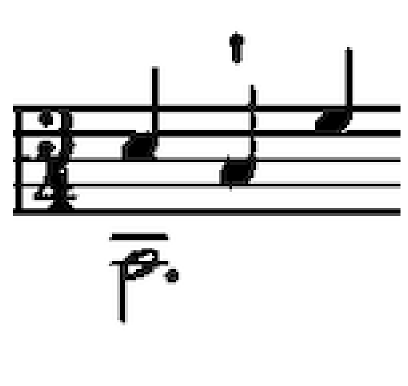
Original Score	OMR Result
	

Figure 17 – Preliminary Experiments - Missing Accidentals (Flat, Sharp, etc)

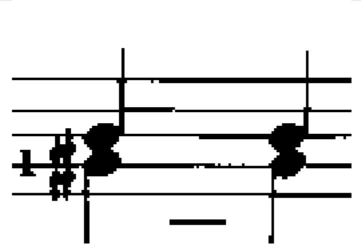
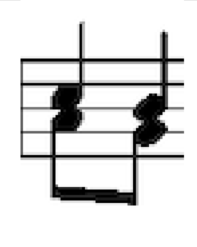
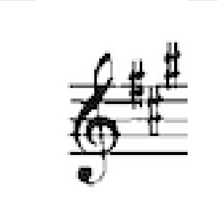
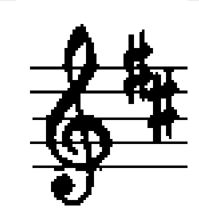
Original Score	OMR Result
	

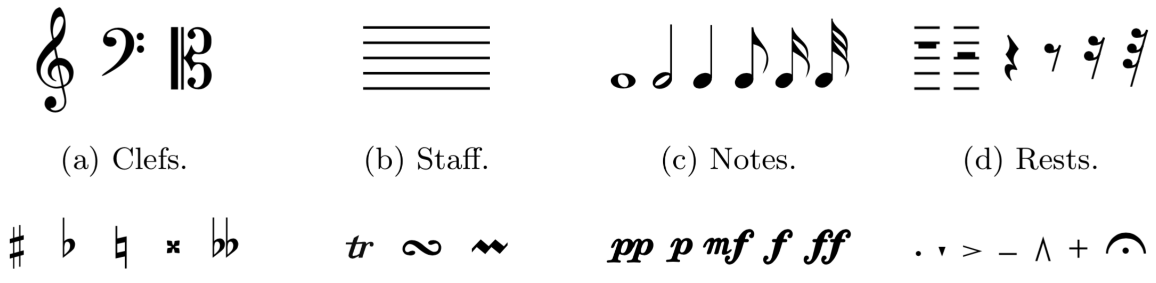
Figure 18 – Preliminary Experiments - Key Signature Misclassified

Original Score	OMR Result
	

These preliminary experiments were important to identify some of the challenges that we are dealing with when evaluating OMR. In the next section we present those challenges.

2.5 Challenges

Optical music recognition area presents several challenges, for example: staff removal and symbol segmentation(OH et al., 2017), ambiguities during graphical primitives detection and classification(BARÓ; RIBA; FORNÉS, 2016), imbalance of classes and overlapping elements(JASTRZEBSKA; LESINSKI, 2016). Concerning handwritten scores, more challenges appear: high variability of handwritten styles, pens, papers and traces(BARÓ; RIBA; FORNÉS, 2016). Considering the use of digital cameras, there are still: paper and



(a) Clefs. (b) Staff. (c) Notes. (d) Rests.

(e) Accidentals. (f) Ornaments. (g) Dynamics. (h) Articulations.

Figure 19 – Music notation symbols(NOVOTNỲ; POKORNỲ, 2015)

staves bending, no parallelism between paper and camera resulting in size variations for elements, shadow and light influence(NA; KIM, 2016).

Another important matter to be considered is that different errors during the OMR process can cause distinct levels of impact on final results. Music notation symbols can be organized considering their information type:

> "Clefs (Fig. 19a) determine the pitches for each line and space of the staff (Fig. 19b), accidentals (Fig. 19e) temporarily modify the pitch of following notes. The pitch of notes itself (Fig. 19c) is indicated by their vertical placement on the staff, and their appearance affects the relative duration. Ornaments (Fig. 19f) change the pitch pattern of individual notes. Rests (Fig. 19d) indicate a relative duration of silence. Dynamics (Fig. 19g) signify the varying loudness. Articulations (Fig. 19h) change the timbre or duration of a note."(NOVOTNỲ; POKORNỲ, 2015)

Analyzing those types, it's clear that miss identifying a dynamic (Fig. 19g) or an articulation (Fig. 19h) symbol will cause less damage to the final result than miss identifying rests (Fig. 19d) or notes (Fig. 19g) that can change the melody and the alignment of the entire music.

Considering the results obtained through the preliminary experiments, in this section we present some elements and errors of music scores to illustrate the challenges involving the automatic comparing of OMR results.

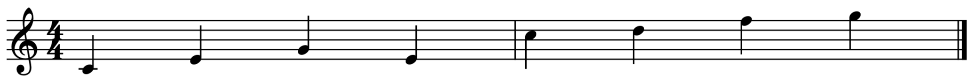
An automatic method, to be useful, needs to handle at least the basic symbols and error cases presented here.

2.5.1 Simulation's Scenarios

In order to understand and propose the method, it is important to enumerate some of the possible errors resulting from an OMR system. For this reason, in this section we present some of the possible errors that the method must be able to handle.

In an attempt to make the identification of the errors simpler for the reader, in this section we present as a ground-truth, the music score presented in figure 20.

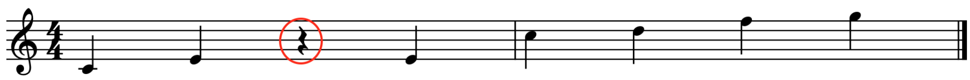
Figure 20 – Scenarios - Music Score Example - Ground-truth



2.5.1.1 Scenarios - Error Case #1: Rest Instead of Note

One of the common errors of OMR systems is that it can fail to identify a particular symbol or note. For this error case, Figure 21 shows on the music clef a missing note that was wrongly substituted by a rest of the same duration. Figure 21 also shows the output of the standard OMR metric (accuracy).

Figure 21 – Scenarios - Music Score - Error Case 1



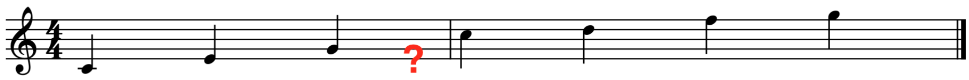
```
Standard Metric – accuracy: 7/8
```

2.5.1.2 Error Case #2: Incorrect Music Score Due to Missing Note

Another issue that can occur when an OMR system fails to recognise a note, is that it simply ignores the note existence and breaks the correct duration of the measure. Figure 22 presents an example of this error case where the OMR system failed to recognise the 4th note of the 1st measure (An E4 note) of the ground-truth and simply ignored the time signature of the measure. The measure should have in total a duration equivalent to four quarter notes. As shown in Figure 22 the resulting measure has only three quarter notes, therefore, there is an error related with the duration of the whole measure. Furthermore, we can see that the standard metric output is not able to discern between the different type of errors when comparing the output results of 7/8 in error case 2 with the output result of 7/8 in the error case 1.

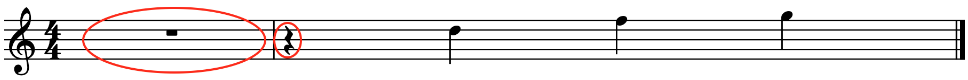
2.5.1.3 Error Case #3: Partial Recognition

Another error that can happen in OMR systems is that it correctly identifies some (but not all) parts of the music score. Figure 23 shows the first whole measure with only

Figure 22 – Scenarios - Music Score - Error Case 2


```
Standard Metric – accuracy: 7/8
```

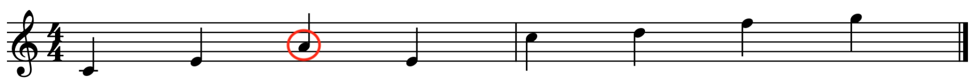
one rest and the second measure has only a missing note that was wrongly substituted by a rest of same duration.

Figure 23 – Scenarios - Music Score - Error Case 3

```
Standard Metric – accuracy: 3/8
```

2.5.1.4 Error Case #4: Pitch Error

Another possible problem in OMR systems, is that it correctly identifies the note duration, but incorrectly identifies the note pitch. For example, in Figure 24 the second note in the first measure was recognized as a A4 instead of a G4. Although the standard metric result of 8/8 may come as a surprise to reader, according to (RAPHAEL; JIN, 2014) most OMR researchers "have focused on the primitive level, identifying the number of closed note heads, open note heads, clefs, quarter rests, beams, stems, flags, etc.".

Figure 24 – Scenarios - Music Score - Error Case 4

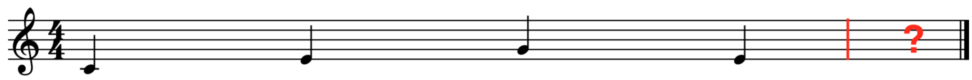


```
Standard Metric – accuracy: 8/8
```

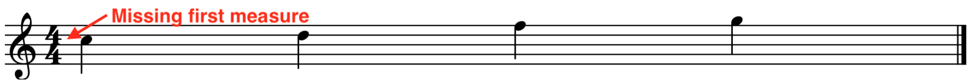
2.5.1.5 Error Case #5: Missing Measures

Another issue that will have a direct impact on the evaluation of an OMR system, is in the cases where the system fails to identify one or more whole measures. In order to illustrate this issue, let us consider the examples presented in Figures 25 and 26. Figure 25 shows an example where the second whole measure is missing while Figure 26 shows an example where the first whole measure is missing.

The missing measures error case brings an important issue to discussion, because the missing measures can affect the alignment when comparing a full music score. In this example, the lack of identification of the first measure will generate several errors, since the comparison will be made using the first measure of the ground-truth with the only measure available. Without an alignment algorithm, the comparison will indicate 4 wrong notes and a missing measure. Instead, the correct comparison should indicate that the first measure is missing and the second measure is perfect.

Figure 25 – Scenarios - Music Score - Error Case 5 - Second Measure Missing


```
Standard Metric – accuracy: 4/8
```

Figure 26 – Scenarios - Music Score - Error Case 5 - First Measure Missing


```
Standard Metric – accuracy: 4/8
```

2.5.2 Alignment Issues

According with the last error case shown above, it's important to consider cases that can affect the total alignment of the score. One additional note at the beginning of the measure can result in all notes wrong, for instance. One error detecting an early measure of the score, can misalign all following measures.

2.5.2.1 Alignment Case #1: Measure Alignment

The first alignment case, on figure 27, is an example when the OMR System fails to detect one measure in the middle of the score. In this case, the third measure was missed. In order to do the best comparison the fourth measure of the ground-truth should be compared to the third measure of the OMR result and the third measure of the ground-truth should be declared missing.

2.5.2.2 Alignment Case #2: Voice Alignment

This example shows different voicing order. Each OMR system can organize voices its own way. In figure 28 red color represents voice 1 and green color represents voice 2. If

Figure 27 – Scenarios - Music Score - Aligment Case 1

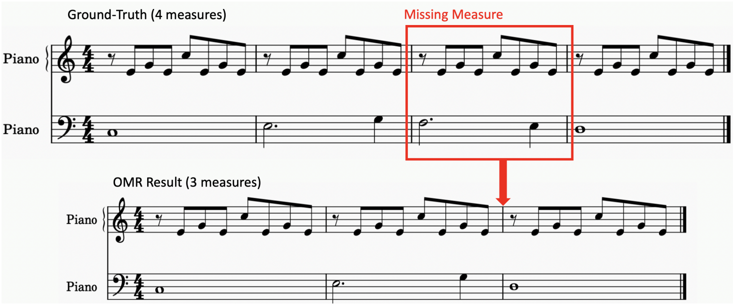
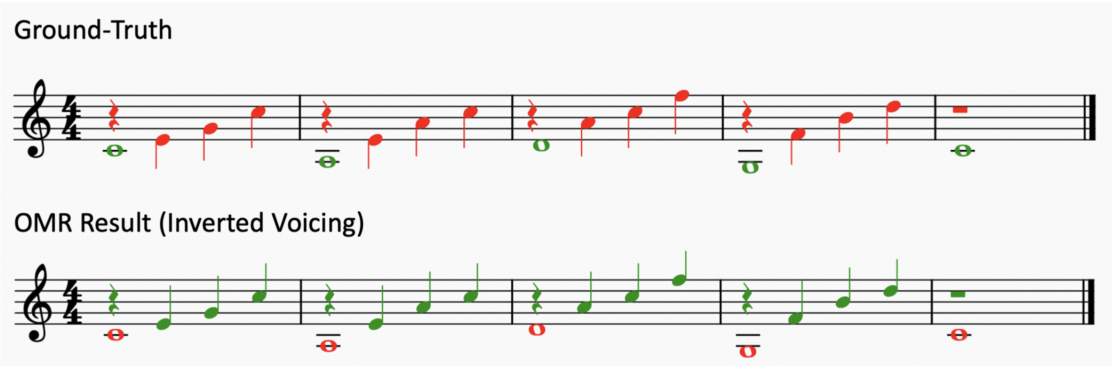


Figure 28 – Scenarios - Music Score - Aligment Case 2



the algorithm compares 1 to 1 and 2 to 2, the result will be all notes wrong.

2.5.2.3 Alignment Case #3: Notes Inside Chords Alignment

Figure 29 is a tricky case, because there are two different situations. The first one is a chord, set of parallel notes, with one note missing in the middle. The second situation is a chord of two notes compared with a note alone, not a chord, and it can came in any sequence, top to bottom or bottom to top.

2.5.2.4 Alignment Case #4: Notes/Chord Alignment Inside Measures

At last, but not least, figure 30 shows that the sequence of notes/chords inside the measure can be affected by misidentifying or missing elements. Another case where alignment algorithms should be applied.

Figure 29 – Scenarios - Music Score - Aligment Case 3

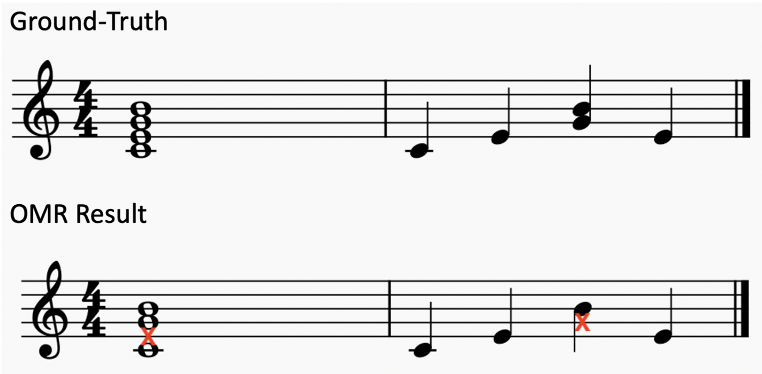
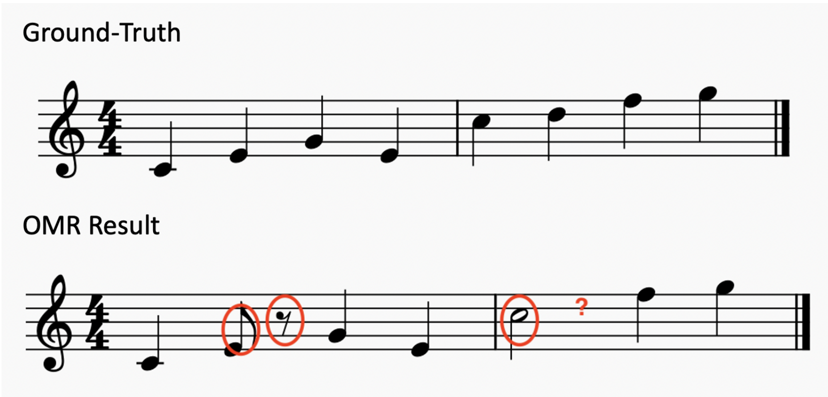



Figure 30 – Scenarios - Music Score - Aligment Case 4

2.6 Final Considerations

In this chapter, we described how an OMR system works, detailing its stages and current used metrics and evaluation methods. Also, the problems and challenges were presented, along with the preliminary experiments and scenarios.

3 Related Works

A systematic literature review searching for metrics and methods of evaluation and comparison of OMR systems and algorithms has been performed(MENGARELLI et al., 2020). Through the research it was reported that each metric was applied in some specific procedure of the OMR. Section 3.1 explains the research protocol, the objectives, the used databases and how the research was executed. Section 3.3 organizes the information obtained from the research and displays metrics, stages, comparisons, OMR datasets and related works.

3.1 Research Protocol

The research protocol defines the process that will be applied to perform the review aiming the proposed objective. The protocol comprises the research questions, the search strategy and the inclusion and exclusion criteria. The chosen research question for this work is:

- How are OMR systems evaluated?

The main objective is to enumerate and compare methods of evaluation and metrics in order to allow future works to use them and establish new standards focusing the improvement this area. Three outcomes are expected from this work:

- Identify more relevant works regarding evaluation and metrics for OMR;
- Identify new needs and possibilities in this area;
- Allow comparative studies among several existing methods;

3.1.1 Keyword Definition

ORM unites two distinct areas: computing and music. It makes necessary to search for synonyms and keywords that can be found in it's works, taking into account the acronym "OMR" that can be found in several other areas. The keywords were chosen to find all available publications on the subject and are shown in the table 3:

3.1.2 Online Libraries and Search Strings

The online libraries used in this research are shown in the table 4:

Table 3 – Research Protocol - Keywords

Test
Measurement
Metrics
Evaluation
Assessment
Comparison
Optical Music Recognition
OMR
Music Score
Music Scores

Table 4 – Research Protocol - Research Bases

1. ACM(MACHINERY, 2017)
2. IEEE(IEEE, 2017)
3. Scopus(B.V., 2017)
4. SpringerLink(AG, 2017)
5. Taylor & Francis Online(LIMITED, 2017)

For each base it was needed to use a specific search string due to limitations and syntax differences:

1. **ACM:** Several searches combining the String:

   +(test* measur* metric* evaluat* assess* compar*) +("Optical Music Recognition" OMR "Music Score" "Music Scores")

   - With all the following Strings:

     +measur* +"Optical Music Recognition"

     +measur* + OMR

     +metric* + "Optical Music Recognition"

     +metric* + OMR

     +evaluat* + "Optical Music Recognition"

     +evaluat* + OMR

     +assess* + "Optical Music Recognition"

     +assess* + OMR

     +compar* + "Optical Music Recognition"

     +compar* + OMR

2. **IEEE:** Advanced search was used with the option "Metadata Only". Two strings where needed.

   (measur* OR metric* OR evaluat* OR assess* OR compar*) AND ("Optical Music Recognition" OR OMR OR "Music Score" OR "Music Scores") (test*) AND ("Optical Music Recognition" OR OMR OR "Music Score" OR "Music Scores")

3. **Scopus:**

   (test* OR measur* OR metric* OR evaluat* OR assess* OR compar*) AND ("Optical Music Recognition" OR OMR OR "Music Score" OR "Music Scores")

4. **SpringerLink:**

   (test* OR measur* OR metric* OR evaluat* OR assess* OR compar*) AND ("Optical Music Recognition" OR OMR OR "Music Score" OR "Music Scores")

5. **Taylor & Francis:** Search modified to search only for Title:

   (test* OR measur* OR metric* OR evaluat* OR assess* OR compar*) AND ("Optical Music Recognition" OR OMR OR "Music Score" OR "Music Scores")

3.1.3 Inclusion and Exclusion Criteria

The selection of articles was planned to be done in two different steps. The first step was to eliminate most unrelated articles quickly, using only the article title and abstract. The exclusion criteria were defined as:

- Exclude non-English articles;
- Exclude articles with title clearly not related to OMR;
- Exclude articles which abstract clearly indicates it's not related to OMR;
- Exclude duplicated articles;

The second step was more laborious because the entire article was considered and analyzed. The inclusion and exclusion criteria were defined as:

- Exclusion Criteria;
  - Exclude articles not related to OMR;
  - Exclude articles related to OMR but with no dataset, no comparison and no metric;
  - Exclude duplicated articles;
- Inclusion Criteria;
  - Include articles containing any metric or evaluation of OMR;
  - Include articles containing comparison among OMR systems or algorithms;
  - Include articles containing OMR datasets;

3.2 Conducting the Review

In this section it is demonstrated how the research was conducted.

3.2.1 Selection of Primary Papers

After the execution of the searches on the databases, a total of 932 articles were found and organized in the Mendeley[1] platform (Reference manager and research network), which assisted in the pre-selection of the articles by excluding articles with duplicate titles, reducing them to 802 articles to be analyzed.

At this point, 5 teams were formed and each researcher was part of 2 teams. Those remaining articles were distributed to the teams in order to eliminate unrelated articles and duplicated titles only by analyzing abstract as described on the research protocol. Each team evaluated the articles applying the exclusion criteria defined for the first step.

At the end of this step, 624 articles were discarded and 178 articles were considered for the more careful analysis.

3.2.2 Detailed Analysis in Pairs

Each team now was responsible for 36 articles, except one group that was responsible for 34. Each team evaluated articles contents applying the inclusion and exclusion criteria defined for the second step.

3.2.3 Selected Articles

The results were organized as spreadsheets and validated using the Kappa coefficient, since it's an efficient method to verify the level of agreement of the teams(VIERA; GARRETT et al., 2005).

When the value one (1) is obtained for this coefficient, it means that the agreement between the team mates is 100%. The closer to zero is the result, the lower is the agreement. See table 5

Table 5 – Review - Kappa

Member 1	**Member 2**	**Kappa**
João	Luciano	0,63
Bruno	Luciano	1,00
Maicon	William	0,88
João	William	0,66
Maicon	Bruno	0,81
Overall Kappa		**0,81**

[1] https://www.mendeley.com/

At the final of this step, 94 articles were selected to be part of the results as shown on figure 31 and figure 32.

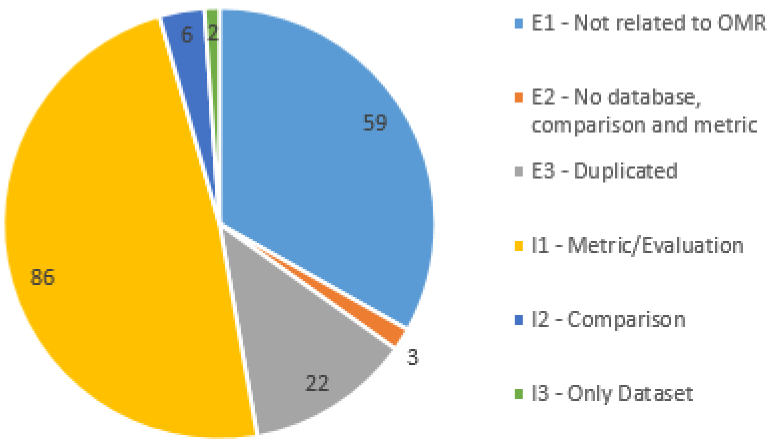


Figure 31 – Review - Detailed Analysis Results by Inclusion (I) and Exclusion (E) Criteria

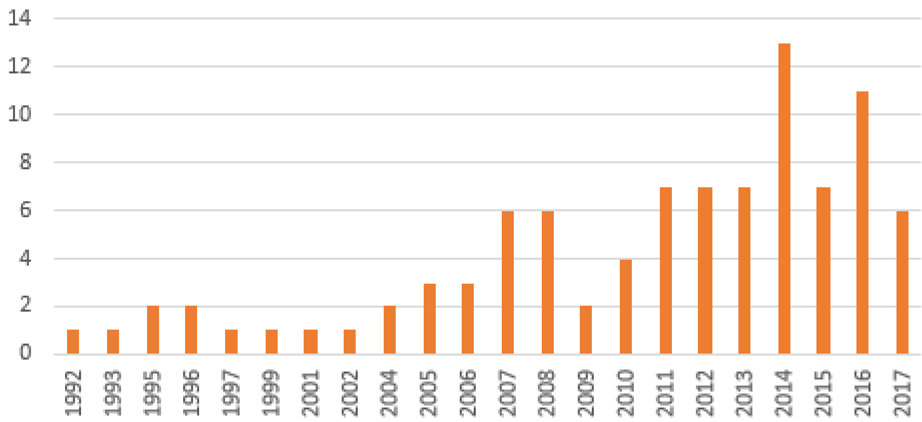


Figure 32 – Review - Histogram by Year

3.3 Results

All the selected articles were analyzed and classified regarding metrics, datasets, stages of OMR, type of music score and existing comparisons. The concept was to identify all datasets and metrics, to find the examples of comparison and provide lists of articles in an organized form. Some works could not be classified due to their particularity, as: writer recognition for historical archives(BRUDER; IGNATOVA; MILEWSKI, 2004), layout analysis for ancient handwritten scores(CAMPOS et al., 2016), audio and score alignment(İZMIRLI; SHARMA, 2012), handwritten score alignment(RIBA; FORNÉS; LLADÓS, 2015), watermarking for music scores(SCHMUCKER, 2002) and works on Chinese numeric music notation(WU; JANG, 2014b)(WU; JANG, 2014a)(WU, 2016).

3.3.1 Metrics

The majority of the 94 selected articles uses accuracy as metric, many times referenced as Recognition Rate or Classification Rate. Several statistical metrics can also be found such as: Precision, Recall, Error Rate, TP (True Positives), TN (True Negatives), FP (False Positives), FN (False Negatives), Specificity, Sensitivity, F-Measure (F-Score) and Errors type I and II. On table 6 the metrics were classified considering their scope and stage, and the following categories were selected:

- M1 - Metrics presented on general results or at the end of entire process.
- M2 - Metrics presented to evaluate the staff detection and removal.
- M3 - Metrics presented to evaluate the symbol segmentation/classification/recognition.
- M4 - Metrics presented to evaluate the result music notation, but only 3 works were presented:

  (BYRD; SCHINDELE, 2006) uses Note pitch error (%), Duration error (%) and Accuracy.

  (PUGIN et al., 2008) uses recall and precision.

  (CHEN; DUAN, 2016) uses F-Measure, precision and recall.

Table 6 – Review - Metrics

	M1-General		**M2-Staff Det./Rem.**		**M3-Symbol Rec.**	
Metric	**Q**	**Ref.**	**Q**	**Ref.**	**Q**	**Ref.**
accuracy	20	L1	5	L2	20	L3
precision	3	L4	6	L5	9	L6
TP/TN	2	L7	0		1	L8
FP/FN	2	L9	0		5	L10
error rate	4	L11	6	L12	5	L13
recall	3	L14	7	L15	5	L16
specificity	0		3	L17	0	
sensitivity	0		0		2	L18
F-Measure	0		8	L19	0	

Table 7 – Review - Metrics References

Ref.	Works

L1	(KATO; INOKUCHI, 1992) (MIYAO; NAKANO, 1995) (REED; PARKER, 1996) (PUGIN; BURGOYNE; FUJINAGA, 2007a) (SZWOCH, 2008) (YOO; KIM; LEE, 2008) (FORNES et al., 2009) (LIU, 2012) (LUANGNAPA et al., 2012) (REBELO; MARÇAL; CARDOSO, 2013) (TAMBOURATZIS, 2013) (CHEN; CHEN; TENG, 2013) (FAHN; LU, 2014) (PADILLA et al., 2014) (KODIROV et al., 2014) (CALVO-ZARAGOZA et al., 2015) (MONTAGNER; HIRATA; HIRATA, 2014) (LIU; ZHOU; XU, 2015) (WEN et al., 2015) (ADAMSKA et al., 2015)
L2	(WIJAYA; BAINBRIDGE, 1999) (SZWOCH, 2005) (PIĄTKOWSKA et al., 2012) (MONTAGNER; HIRATA; HIRATA, 2014) (MONTAGNER; HIRATA; HIRATA, 2017)
L3	(YADID-PECHT et al., 1996) (BAINBRIDGE; BELL, 1997) (SU; TEW; CHEN, 2001) (ROSSANT; BLOCH, 2004) (ROSSANT; BLOCH, 2005) (LUCKNER, 2006) (ROSSANT; BLOCH, 2007) (FORNÉS; LLADÓS; SÁNCHEZ, 2007) (DALITZ; MICHALAKIS; PRANZAS, 2008) (SHARIF et al., 2009) (REBELO; CAPELA; CARDOSO, 2010) (TAMBOURATZIS, 2011) (VIGLIENSONI et al., 2011) (LUANGNAPA et al., 2012) (MEHTA; BHATT, 2015) (LESINSKI; JASTRZEBSKA, 2015) (PEREIRA et al., 2016) (NA; KIM, 2016) (JASTRZEBSKA; LESINSKI, 2016) (OH et al., 2017)
L4	(BURGOYNE et al., 2007) (MONTAGNER; HIRATA; HIRATA, 2014) (WEN et al., 2015)
L5	(FORNÉS; LLADÓS; SÁNCHEZ, 2005) (DUTTA et al., 2010) (DINH et al., 2016) (PEDERSOLI; TZANETAKIS, 2016) (VO et al., 2016) (CHEN; DUAN, 2016)
L6	(LÓPEZ et al., 2005) (FORNÉS; LLADÓS; SÁNCHEZ, 2005) (PUGIN et al., 2008) (RAMIREZ; OHYA, 2010) (LESINSKI; JASTRZEBSKA, 2015) (PEREIRA et al., 2016) (NA; KIM, 2016) (BARÓ; RIBA; FORNÉS, 2016) (JASTRZEBSKA; LESINSKI, 2016)
L7	(LIU, 2012) (LIU; ZHOU; XU, 2015)
L8	(MEHTA; BHATT, 2015)
L9	(LIU, 2012) (LIU; ZHOU; XU, 2015)
L10	(MODAYUR et al., 1993b) (LÓPEZ et al., 2005) (FORNÉS; LLADÓS; SÁNCHEZ, 2005) (RAPHAEL; WANG, 2011) (MEHTA; BHATT, 2015)
L11	(ROSSANT; BLOCH, 2007) (DALITZ; MICHALAKIS; PRANZAS, 2008) (FANG; GUI-FA, 2015)

L12	(BAINBRIDGE; BELL, 1997) (CARDOSO; REBELO, 2010) (DUTTA et al., 2010) (FORNES et al., 2011) (SU et al., 2012) (TIMOFTE; GOOL, 2012)
L13	(ROSSANT; BLOCH, 2004) (DALITZ; MICHALAKIS; PRANZAS, 2008) (THOMAS; WAGNER; CLAUSEN, 2011) (JASTRZEBSKA; LESINSKI, 2016) (CALVO-ZARAGOZA; ONCINA, 2017)
L14	(BURGOYNE et al., 2007) (MONTAGNER; HIRATA; HIRATA, 2014) (WEN et al., 2015)
L15	(DUTTA et al., 2010) (MONTAGNER; HIRATA; HIRATA, 2014) (DINH et al., 2016) (PEDERSOLI; TZANETAKIS, 2016) (VO et al., 2016) (CHEN; DUAN, 2016) (MONTAGNER; HIRATA; HIRATA, 2017)
L16	(MODAYUR et al., 1993b) (PUGIN et al., 2008) (RAMIREZ; OHYA, 2010) (NA; KIM, 2016) (BARÓ; RIBA; FORNÉS, 2016)
L17	(MONTAGNER; HIRATA; HIRATA, 2014) (VO et al., 2016) (MONTAGNER; HIRATA; HIRATA, 2017)
L18	(LESINSKI; JASTRZEBSKA, 2015) (JASTRZEBSKA; LESINSKI, 2016)
L19	(MÄRGNER; ABED, 2014) (FORNÉS et al., 2013b) (DINH et al., 2016) (VO et al., 2016) (CHEN; DUAN, 2016) (CALVO-ZARAGOZA; MICÓ; ONCINA, 2016) (CALVO-ZARAGOZA; PERTUSA; ONCINA, 2017) (CALVO-ZARAGOZA; VIGLIENSONI; FUJINAGA, 2017)

3.3.2 Stages of OMR

Nearly 50% of the articles are focused only on the Stage 2 of OMR, therefore the recognition of musical symbols. The distribution can be checked on table 8. Some of the collected data about OMR and its stages and methods are presented on the following sections, resulting on a taxonomy shown on figure 33. As the main focus of the paper is on metrics and evaluation of Optical Music Recognition systems, more detailed information can be obtained in some OMR reviews found during this work. Fornés and Sánchez(FORNÉS; SÁNCHEZ, 2014) provide a review of OMR methods and stages, Rebelo et al.(REBELO et al., 2012) present the State-of-art of OMR systems. There are also works focused only on a specific task as binarization methods(BURGOYNE et al., 2007) and comparison works regarding staff removal(DALITZ et al., 2008) and symbols recognition(DALITZ et al., 2008).

Table 8 – Review - Stages Analysis

Stage	**Qtd.**	**Perc.**	**References**

S1	4	4,7%	(BURGOYNE et al., 2007) (PUGIN; BURGOYNE; FUJINAGA, 2007a) (PINTO et al., 2011) (VO et al., 2016)
S1, S2	8	9,4%	(RAMIREZ; OHYA, 2010) (VIGLIENSONI et al., 2011) (PIĄTKOWSKA et al., 2012) (TAMBOURATZIS, 2013) (CHEN; CHEN; TENG, 2013) (CALVO-ZARAGOZA et al., 2015) (DINH et al., 2016) (PEDERSOLI; TZANETAKIS, 2016)
S2	42	49,4%	(MODAYUR et al., 1993b) (MIYAO; NAKANO, 1995) (YADID-PECHT et al., 1996) (BAINBRIDGE; BELL, 1997) (WIJAYA; BAINBRIDGE, 1999) (SU; TEW; CHEN, 2001) (SZWOCH, 2005) (LÓPEZ et al., 2005) (ROSSANT; BLOCH, 2005) (FORNÉS; LLADÓS; SÁNCHEZ, 2005) (REBELO et al., 2007) (PUGIN; BURGOYNE; FUJINAGA, 2007b) (BELLINI; BRUNO; NESI, 2007) (DALITZ et al., 2008) (FORNÉS; LLADÓS; SÁNCHEZ, 2007) (CARDOSO; REBELO, 2010) (DUTTA et al., 2010) (REBELO; CAPELA; CARDOSO, 2010) (TAMBOURATZIS, 2011) (RAPHAEL; WANG, 2011) (FORNES et al., 2011) (LIU, 2012) (SU et al., 2012) (MALIK et al., 2013) (VISANIY et al., 2013) (TIMOFTE; GOOL, 2012) (MONTAGNER; HIRATA; HIRATA, 2014) (NHAT; LEE, 2014) (MONTAGNER; HIRATA; HIRATA, 2014) (FORNÉS et al., 2013b) (MEHTA; BHATT, 2015) (LESINSKI; JASTRZEBSKA, 2015) (FANG; GUI-FA, 2015) (PEREIRA et al., 2016) (NA; KIM, 2016) (BARÓ; RIBA; FORNÉS, 2016) (CALVO-ZARAGOZA; MICÓ; ONCINA, 2016) (MONTAGNER; HIRATA; HIRATA, 2017) (CALVO-ZARAGOZA; PERTUSA; ONCINA, 2017) (CALVO-ZARAGOZA; VIGLIENSONI; FUJINAGA, 2017) (OH et al., 2017) (CALVO-ZARAGOZA; ONCINA, 2017)
S2, S3	3	3,5%	(ROSSANT; BLOCH, 2007) (SHARIF et al., 2009) (JASTRZEBSKA; LESINSKI, 2016)
S3	2	2,4%	(REBELO; MARÇAL; CARDOSO, 2013) (LIU; ZHOU; XU, 2015)
S3, S4	1	1,2%	(BAUMANN, 1995)

All	25	29,4%	(KATO; INOKUCHI, 1992) (REED; PARKER, 1996) (ROSSANT; BLOCH, 2004) (BYRD; SCHINDELE, 2006) (LUCKNER, 2006) (SZWOCH, 2008) (YOO; KIM; LEE, 2008) (DALITZ; MICHALAKIS; PRANZAS, 2008) (PUGIN et al., 2008) (THOMAS; WAGNER; CLAUSEN, 2011) (BUGGE et al., 2011) (LUANGNAPA et al., 2012) (REBELO et al., 2012) (FAHN; LU, 2014) (PADILLA et al., 2014) (KODIROV et al., 2014) (MÄRGNER; ABED, 2014) (WU; JANG, 2014b) (WU; JANG, 2014a) (RAPHAEL; JIN, 2014) (FORNÉS; SÁNCHEZ, 2014) (BYRD; SIMONSEN, 2015) (WEN et al., 2015) (ADAMSKA et al., 2015) (CHEN; DUAN, 2016)

3.3.2.1 Stage 1: Image Pre-Processing

Concerning the image pre-processing, it was possible to identify different methods for each kind of task. Noise removal has examples using morphological operations (FORNÉS; LLADÓS; SÁNCHEZ, 2007)(FORNES et al., 2009) and median filters (ADAMSKA et al., 2015)(CAMPOS et al., 2016). Morphological operations were also used for blurring correction. Most of the works considering image skew uses the Hough Transform Algorithm.

Binarization was the algorithm with more examples and it's important to highlight the comparative survey found(BURGOYNE et al., 2007). This algorithm usually begins with a gray-scale conversion. Otsu's(OTSU, 1979) threshold was the most used method (CARDOSO; REBELO, 2010)(TIMOFTE; GOOL, 2012)(CALVO-ZARAGOZA et al., 2015)(WEN et al., 2015)(CAMPOS et al., 2016). However other methods were presented, such as: adaptive and heuristic techniques(FORNÉS; LLADÓS; SÁNCHEZ, 2007)(PINTO et al., 2011)(LUANGNAPA et al., 2012), Niblack(FORNES et al., 2009), constant threshold(TAMBOURATZIS, 2011)(TAMBOURATZIS, 2013), adaptive filtering(ADAMSKA et al., 2015) and Gaussian Mixture Markov Random Field (GMMRF)(VO et al., 2016).

3.3.2.2 Stage 2: Recognition of Musical Symbols

It's important to highlight the large number of works and competitions(FORNES et al., 2011)(VISANIY et al., 2013)(FORNÉS et al., 2013a)(FORNÉS et al., 2013b) involving this stage, mainly discussing staff lines detection and removal. For this task, the majority of works use Histograms/Y-Projection(SZWOCH, 2005)(ROSSANT; BLOCH, 2007)(LUANGNAPA et al., 2012)(İZMIRLI; SHARMA, 2012)(TAMBOURATZIS, 2013)(CALVO-ZARAGOZA et al., 2015)(MEHTA; BHATT, 2015)(ADAMSKA et al., 2015). Other methods are also used, such as: median filters(LÓPEZ et al., 2005)(FORNÉS; LLADÓS;

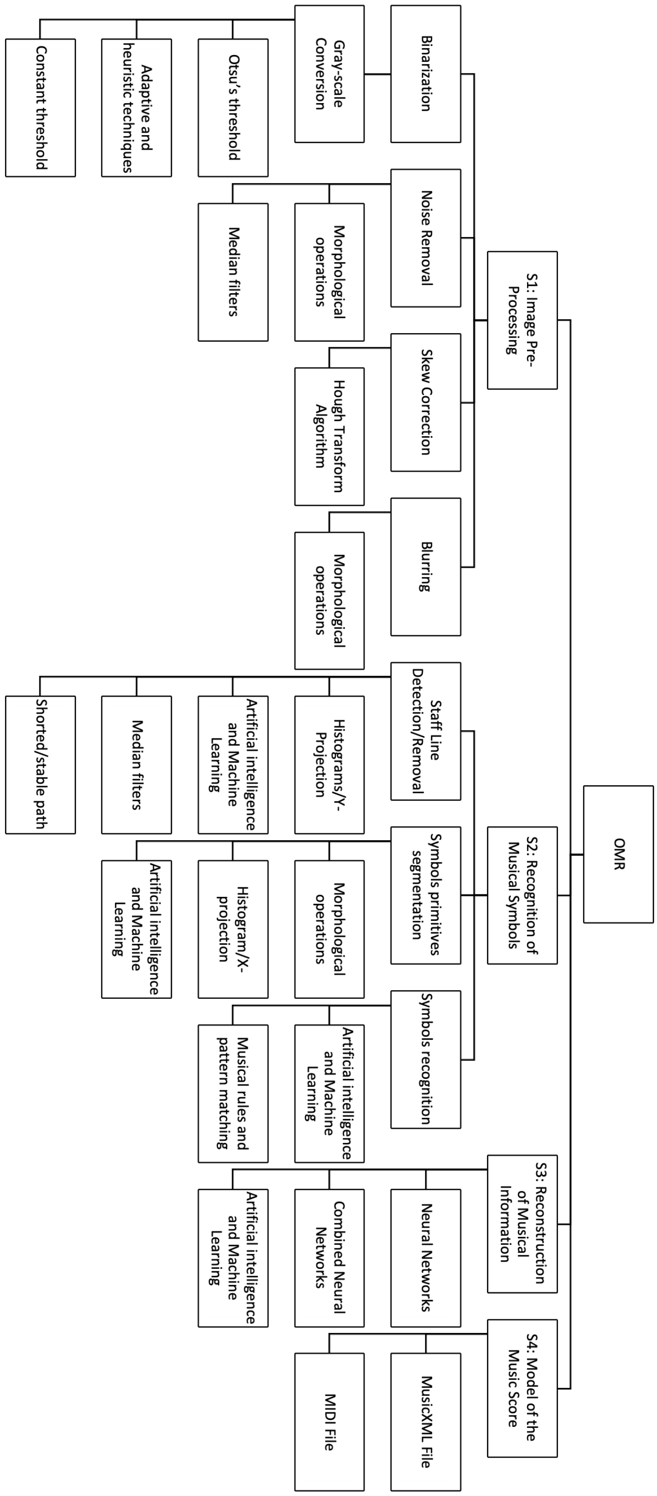


Figure 33 – Optical Music Recognition - Taxonomy

SÁNCHEZ, 2005)(FORNÉS; LLADÓS; SÁNCHEZ, 2007)(FORNES et al., 2009), morphological operations(FORNÉS; LLADÓS; SÁNCHEZ, 2005)(MONTAGNER; HIRATA; HIRATA, 2014), shorted/stable path(REBELO et al., 2007)(WEN et al., 2015)(NA; KIM, 2016)(DINH et al., 2016) and artificial intelligence methods(PIĄTKOWSKA et al., 2012)(CALVO-ZARAGOZA; MICÓ; ONCINA, 2016)(MONTAGNER; HIRATA; HIRATA, 2017)(CALVO-ZARAGOZA; PERTUSA; ONCINA, 2017)(CALVO-ZARAGOZA; VIGLIENSONI; FUJINAGA, 2017)(MONTAGNER; HIRATA; HIRATA, 2014). There are also many papers proposing new methods for staff detection/removal(SCHMUCKER, 2002)(DUTTA et al., 2010)(SU et al., 2012)(TIMOFTE; GOOL, 2012)(CHEN; CHEN; TENG, 2013) and comparative studies(DALITZ et al., 2008)(FORNES et al., 2011)(VISANIY et al., 2013)(FORNÉS et al., 2013a)(FORNÉS et al., 2013b). There is also a paper proposing a method for staff detection without having to remove it(CALVO-ZARAGOZA et al., 2015).

After staff detection/removal, usually, comes the symbol primitive segmentation using methods such as: hough transformation(ROSSANT; BLOCH, 2007), morphological operations(FORNÉS; LLADÓS; SÁNCHEZ, 2007)(MALIK et al., 2013)(MODAYUR et al., 1993b), histogram/x-projection(LUANGNAPA et al., 2012)(TAMBOURATZIS, 2013)(CALVO-ZARAGOZA et al., 2015), templates(İZMIRLI; SHARMA, 2012), hierarchical decomposition(MEHTA; BHATT, 2015), combined neural network(WEN et al., 2015), SVM classifier(CHEN; DUAN, 2016).

Once the primitive segmentation is ready, the next step is the symbol recognition. There are examples using: fuzzy model(SU; TEW; CHEN, 2001)(ROSSANT; BLOCH, 2004)(ROSSANT; BLOCH, 2005)(ROSSANT; BLOCH, 2007), musical rules and pattern matching(LUANGNAPA et al., 2012)(WEN et al., 2015), neural networks(TAMBOURATZIS, 2011)(TAMBOURATZIS, 2013)(LUCKNER, 2006)(MEHTA; BHATT, 2015)(PEREIRA et al., 2016), decision tree(CHEN; CHEN; TENG, 2013)(ADAMSKA et al., 2015)(JASTRZEBSKA; LESINSKI, 2016), kNN(KODIROV et al., 2014)(JASTRZEBSKA; LESINSKI, 2016), random forest(LESINSKI; JASTRZEBSKA, 2015) workflow for primitive assembly(LIU; ZHOU; XU, 2015), SVM(FANG; GUI-FA, 2015)(OH et al., 2017) and finite state machines(CALVO-ZARAGOZA; ONCINA, 2017). There is also a comparative study on symbols recognition(REBELO; CAPELA; CARDOSO, 2010).

3.3.2.3 Stage 3: Reconstruction of Musical Information

Very few papers address this stage. Neural networks(MIYAO; NAKANO, 1995), three Combined Neural Networks (CNN) using majority vote(REBELO; MARÇAL; CARDOSO, 2013) and semantic reconstruction are some of the methods adopted for this stage.

3.3.2.4 Stage 4: Model of the Music Score

Some papers generate XML, MIDI or equivalent files at the end of process(BUGGE et al., 2011)(LUANGNAPA et al., 2012)(PADILLA et al., 2014)(ADAMSKA et al., 2015).

There are also full OMR systems and frameworks such as COMSCAN(SHARIF et al., 2009), Lemon(REED; PARKER, 1996), Gamera Gamut(PUGIN et al., 2008) and Aruspix(VIGLIENSONI et al., 2011).

3.3.3 Comparisons

Concerning the existing comparisons, 26 articles present some kind of comparison and they were divided in two categories:

- C1 - Comparison with commercial systems: SmartScore, MIDISCAN, NoteScan, PhotoScore, SharpEye.
- C2 - Comparison with other algorithms.

See table 9 for more details.
See table 10 for commercial systems used for comparisons.

Table 9 – Review - Articles with Comparison

Category	**Quant.**	**Metrics**
C1 - Comparison with commercial systems	12	accuracy(CHEN; CHEN; TENG, 2013); FP(REED; PARKER, 1996); visual comparison(ROSSANT; BLOCH, 2004)(ROSSANT; BLOCH, 2005); error rate(BYRD; SCHINDELE, 2006)(ROSSANT; BLOCH, 2007); error/rank(BUGGE et al., 2011); TP/FP/FN/accuracy(LIU, 2012)(RAPHAEL; JIN, 2014)(LIU; ZHOU; XU, 2015); precision/recall(BARÓ; RIBA; FORNÉS, 2016); very specific and detailed metrics(BELLINI; BRUNO; NESI, 2007)

C2 - Comparison with other algorithms	14	accuracy(PUGIN; BURGOYNE; FUJINAGA, 2007a)(FORNÉS; LLADÓS; SÁNCHEZ, 2007)(YOO; KIM; LEE, 2008)(REBELO; CAPELA; CARDOSO, 2010)(VIGLIENSONI et al., 2011)(THOMAS; WAGNER; CLAUSEN, 2011)(MONTAGNER; HIRATA; HIRATA, 2017); recall/precision(BURGOYNE et al., 2007)(PUGIN et al., 2008); visual comparison(REBELO et al., 2007); error rate(DALITZ et al., 2008)(FORNES et al., 2011)(SU et al., 2012); F-Measure/precision/recall/specificity(VO et al., 2016)

Table 10 – Review - Commercial Systems

Commercial System	**References**
Capella-Scan	(BUGGE et al., 2011)
MIDISCAN	(REED; PARKER, 1996)
NoteScan	(REED; PARKER, 1996)
O^3MR	(BELLINI; BRUNO; NESI, 2007)
PhotoScore	(BYRD; SCHINDELE, 2006) (BARÓ; RIBA; FORNÉS, 2016)
SharpEye	(BELLINI; BRUNO; NESI, 2007) (LIU, 2012) (CHEN; CHEN; TENG, 2013) (RAPHAEL; JIN, 2014) (LIU; ZHOU; XU, 2015)
SmartScore	(ROSSANT; BLOCH, 2004) (ROSSANT; BLOCH, 2005) (BYRD; SCHINDELE, 2006) (ROSSANT; BLOCH, 2007) (BELLINI; BRUNO; NESI, 2007) (BUGGE et al., 2011) (LIU, 2012) (LIU; ZHOU; XU, 2015)

It was possible to identify two events competitions: "ICDAR / GREC 2011 Competition: Writer Identification and Staff Removal"(FORNES et al., 2011)[2] and "ICDAR / GREC 2013 Competition on Music Scores: Staff Removal"(VISANIY et al., 2013)[3]. Both are focused on Stage 2: staff removal and writer recognition of handwritten music scores. The writer recognition is usually made using classification algorithms after the staff removal. There was also an extension of the ICDAR / GREC 2011 Competition with some new image distortion creating tree levels of difficulty(FORNÉS et al., 2013a).

It's possible to see on table 9 the difficulty of establishing benchmarks and rankings. Each work uses different metrics and datasets for comparing to commercial systems and

[2] http://www.cvc.uab.es/cvcmuscima/competition/index.htm
[3] http://www.cvc.uab.es/cvcmuscima/competition2013/

with other methods.

3.4 Final Considerations

In this chapter, we presented a full systematic literature review of metrics and methods of evaluation and comparison of OMR systems. All details of the research were presented, including a taxonomy for Optical Music Recognition. The results confirm the need of an automatic and more contextualized evaluation method.

4 Proposed Evaluation Method

This work proposes a new method with a public implementation and a ground-truth dataset in MusicXML to help the evaluation of OMR systems. It also establishs a standard way to compare the final results of full OMR systems, using the MusicXML result, considering the following characteristics:

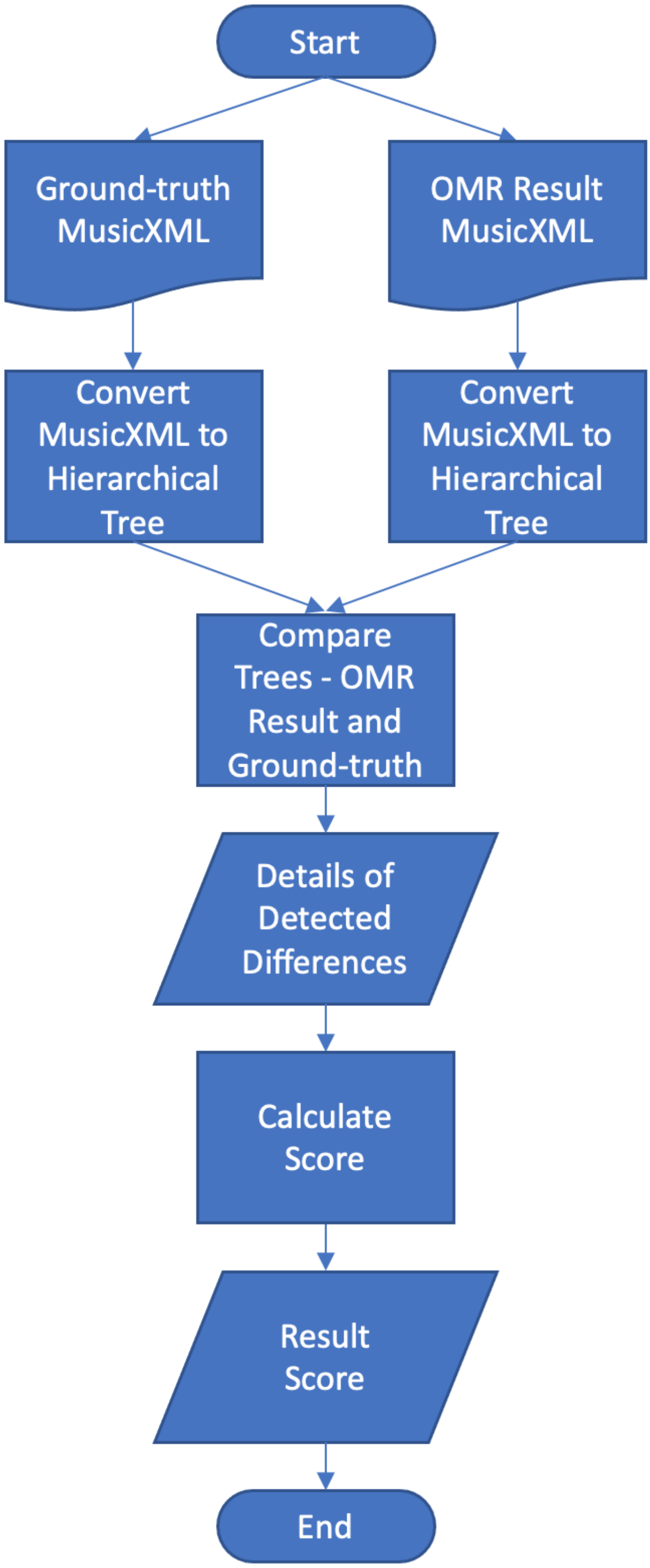
- Only most important music information is considered. Positional, layout, fonts and non musical information are ignored.
- Measures and notes reorganized and realigned to allow the comparison between different systems.
- For this first version, only systems, measures, attributes, notes and rests are considered.

Figure 34 presents an overview of the proposed method. The proposed method can be divided into the following steps: (1) Dealing with the Ground-truth MusicXML data and the MusicXML provided by the OMR system; (2) Conversion of the MusicXML to a Hierarchical Tree Structure; (3) Comparison of the Generated Trees; (4) Presentation of the detected differences; (5) Score calculation based on the detected differences; (6) Presentation of the resulting score. In the following subsections, each of these steps will be presented in more detail.

The method will always compare the ground-truth MusicXML file with the resulting MusicXML file of the OMR System. Each error should be identified and then classified in a specific type that should represent a weight in points. The sum of the points will represent the amount of errors generated by the OMR system, meaning that a perfect system should achive 0 points. The weight of each error type can be determined by the difficulty of correcting it using a music score editor, as Musescore(MuseScore BVBA, 2018), for example. It's possible to establish also different sets of weights considering different aspects like, for instance, the hearing impact of the errors.

This work will use the tree depth to calculate points for each error, the more deep is the error, the less point it represents.

Figure 34 – Proposed Method - General Flowchart



4.1 Dealing with MusicXML Data

In order to deal with MusicXML data it is important to understand its format. The reason behind using MusicXML data in our work is due to the fact that it is an industry standard for sharing, representing, opening and editing music scores in any software. (SZWOCH, 2008) states that "Though there are many different formats available only MusicXML format gained the status of a de facto standard and is accepted by most commercial and academic applications, allowing for musical information exchange between different score-writing, music recognition and music sequencer software".

The MusicXML file standard is designed to allow the storage of music score documents. In order to do that, it has, besides the musical information, information about sources, layouts, fonts and position of elements, such as `<page-height>, <page-width>`; and information regarding the visual position of the notes (e.g. `note default=x"12.36" default-y="-10.00">` along with actual music information (e.g. `<pitch> <step>D</step>`).

Specification, schemas and DTDs for MusicXML can be found online on <https://www.musicxml.com/for-developers/>.

4.2 Hierarchical Conversion

An analysis of the MusicXML format shows that the musical information within the MusicXML has a hierarchical structure. Figure 35 shows some important aspects of the hierarchical information present in the Music XML.

The proposed method is based on the hierarchical structure of MusicXML files. Each staff of the music score is identified as a "part" in the MusicXML file. Figure 36 shows an example from the Dalitz's database where there are 4 parts, each one representing an independent voice of the music.

The main idea of the method is to create a tree structure for each "part" of the MusicXML file, treating each measure as a node of the tree and then each note or rest as a node of the measure and so on. This way, it is possible to compare each part independently.

There are some additional invisible information on the score, but very important for comparison. MusicXML files can have chords, that represents notes in parallel, and independent voices. Those two information and the measure can have alignment issues. In order to align elements with algorithms, the tree structure adopted is shown on figure 37. Figure 38 shows some of those elements in the music score.

Figure 35 – Proposed Method - MusicXML Hierarchy Example

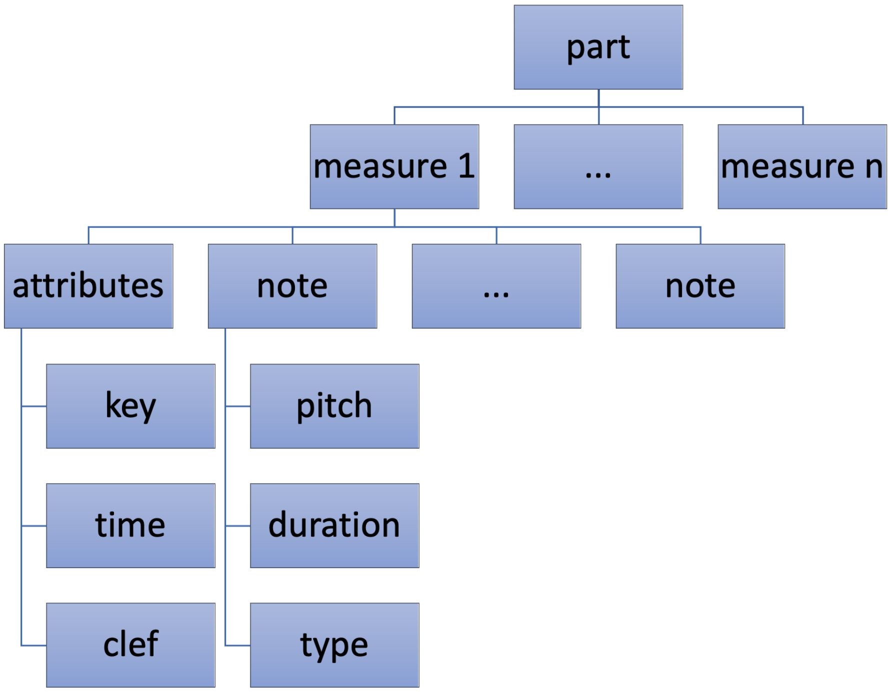


Figure 36 – Proposed Method - Parts and Measures - Brahms score from the Dalitz Dataset

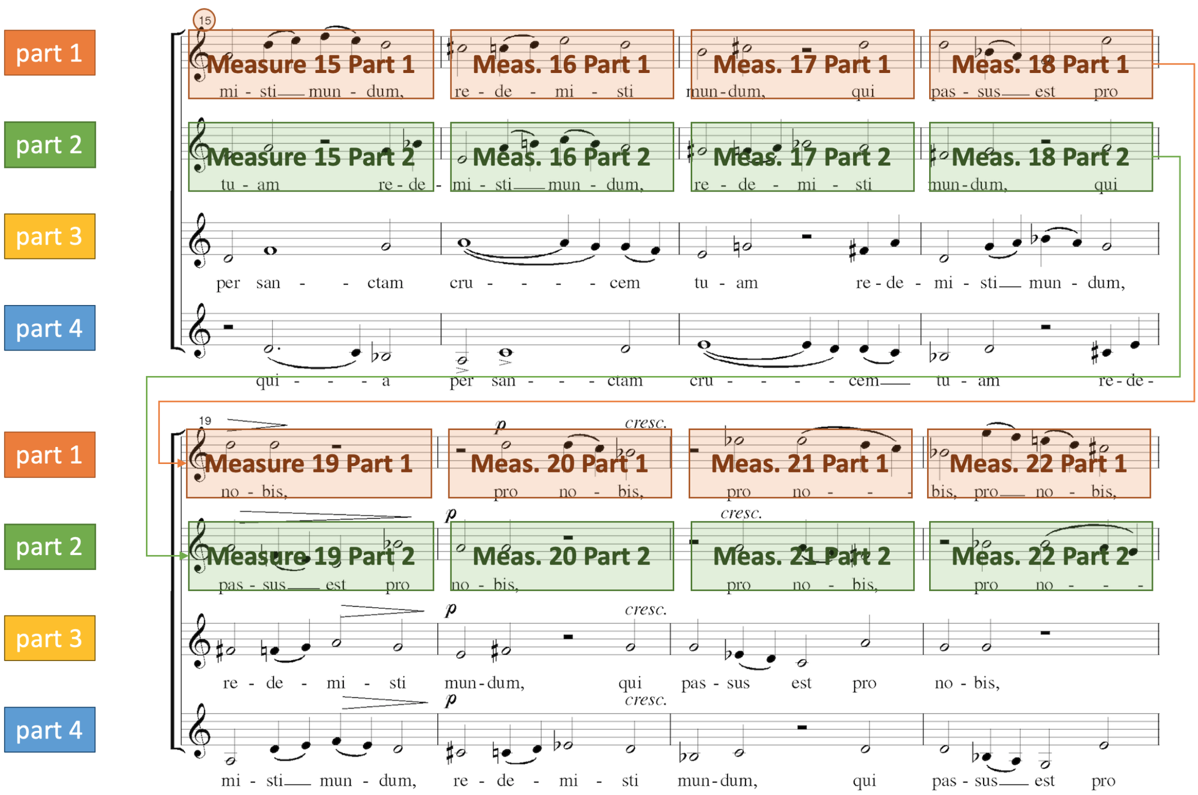


Figure 37 – Proposed Method - Adopted Tree Structure

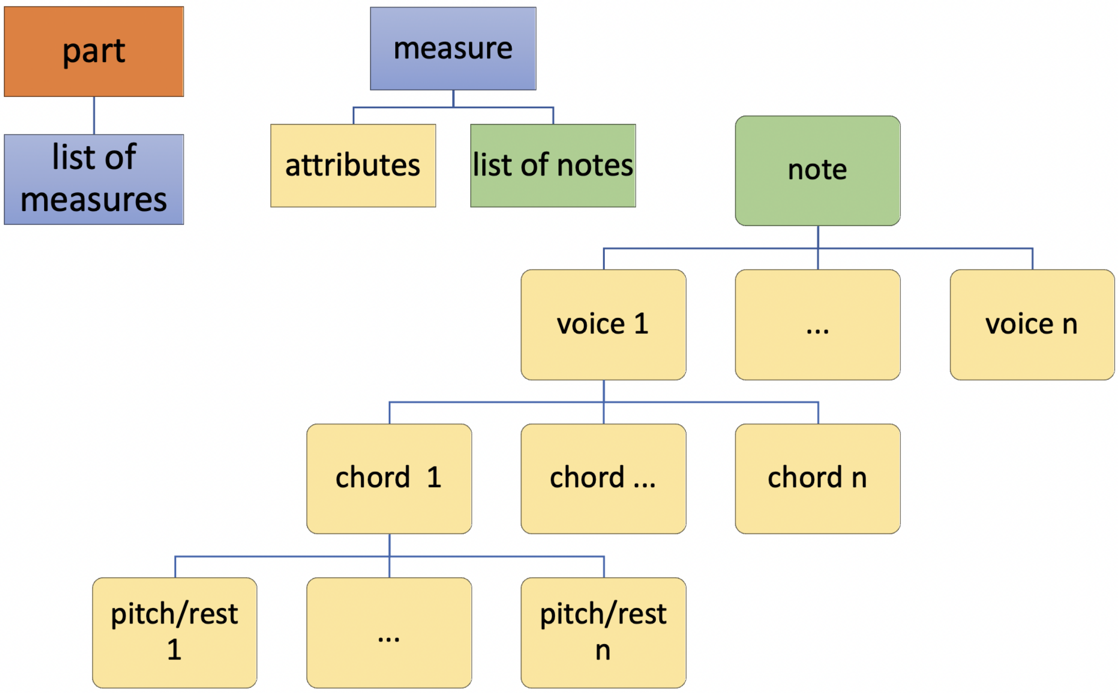
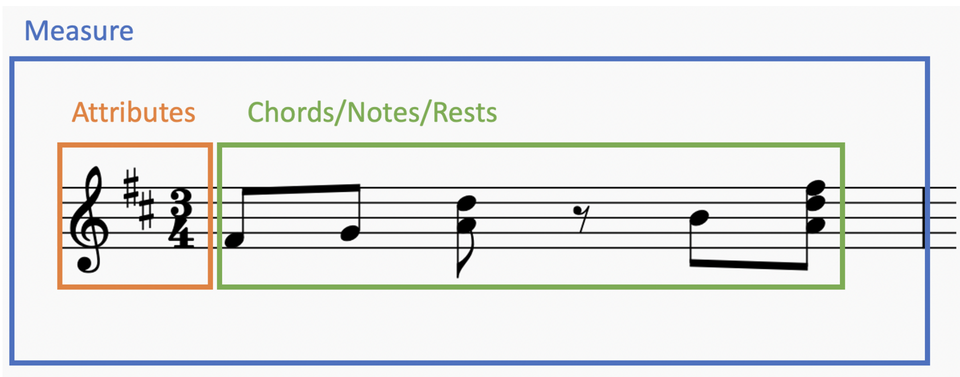


Figure 38 – Proposed Method - Used Elements on the Score



4.3 Detailed Example of Hierarchical Conversion

Figure 39 is a good example of music score with some of the elements needed to understand the conversion. This small example was edited on Musescore (MuseScore BVBA, 2018) and exported to MusicXML generating a file with more than 500 lines [1]. Even after cleaning all positional information and some instrument description, the file was still near 500 lines [2]. In order to understand the conversion, we will present some

[1] This file is available at https://github.com/mengarelli/omr-comparison
[2] This file is available at https://github.com/mengarelli/omr-comparison

slices of the file.

Figure 40 shows the music score with the representation of some elements in MusicXML to facilitate the understanding of the conversion method.

Figure 39 – Proposed Method - Music Score Example for Conversion

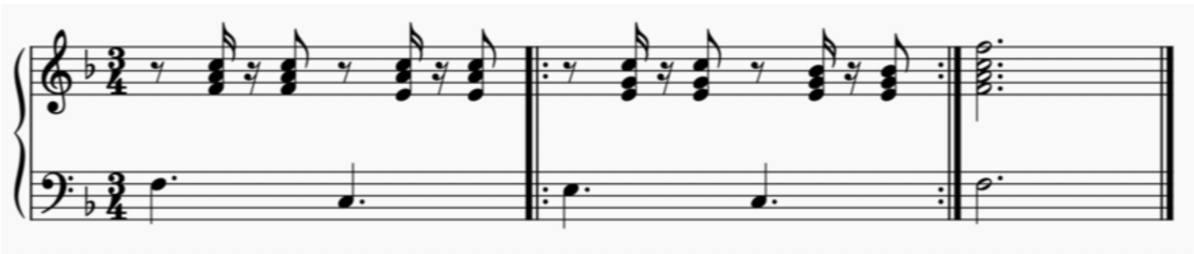
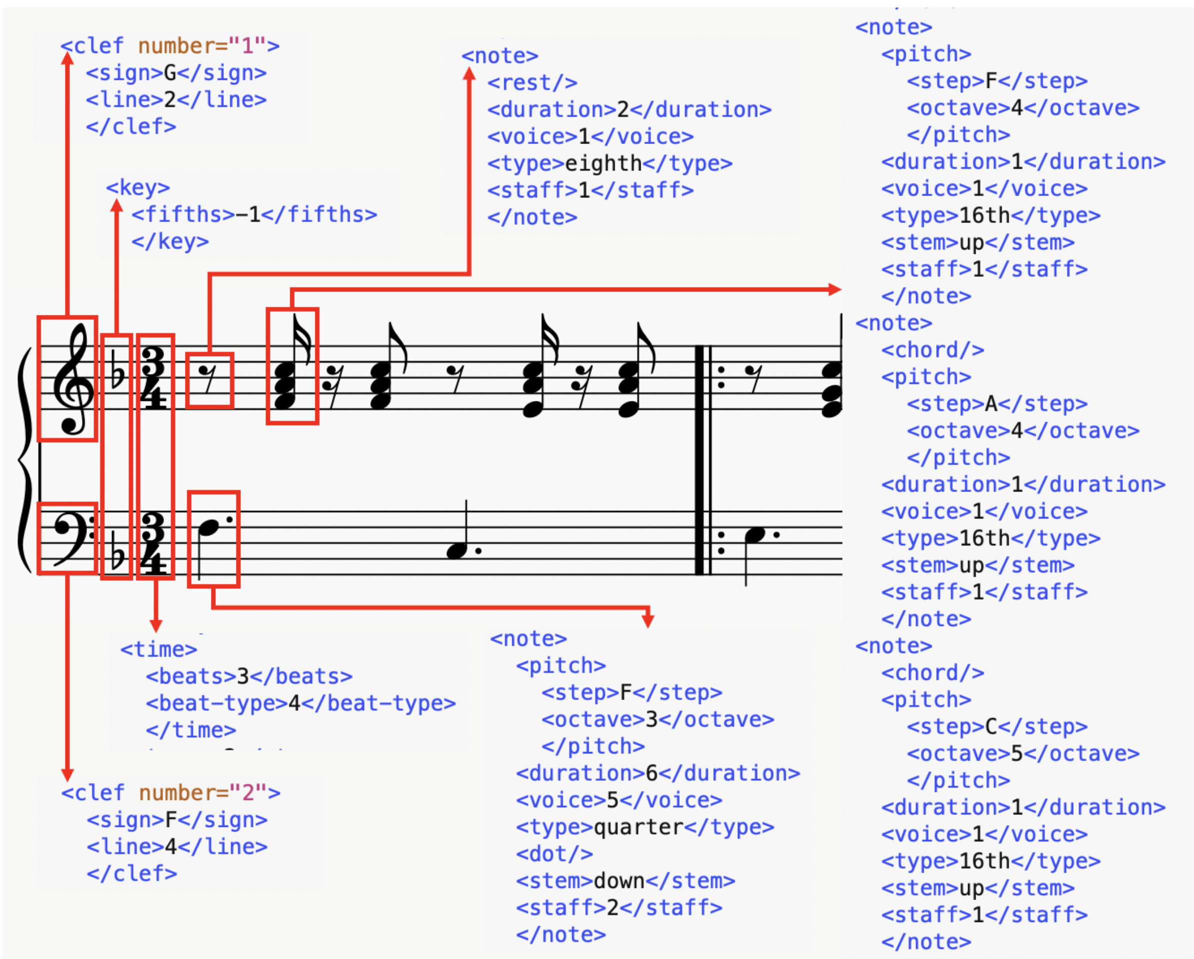
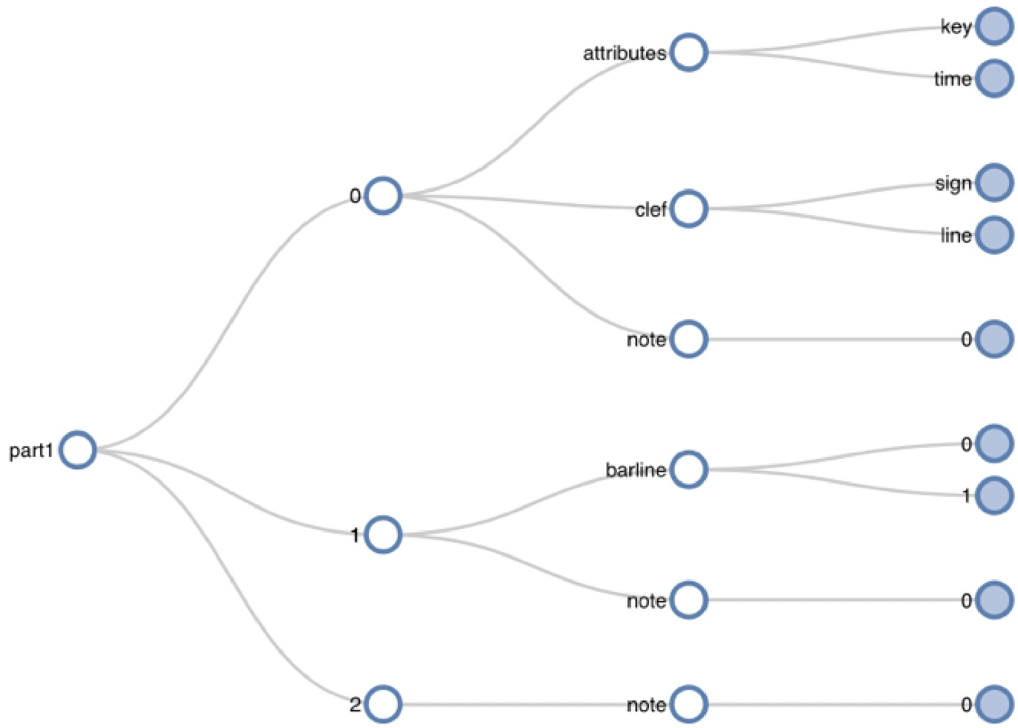



Figure 40 – Proposed Method - Music Score and MusicXML Elements



In this example, both staves are presented together as only one part representing the grand staff, combining treble and bass clef. This can be a problem because most of the OMR systems analyze each staff independently. Therefore the method always break combined staves in independent staves as shown on figures 41 and 42. Also figure 43 shows the notes structure for measure 2 of part 1, treble clef.

Figure 41 – Proposed Method - Part 1 of Score Example, Treble Clef



Once the information inside the MusicXML is converted, the result is a list of parts, each part being a list of measures represented by a tree sctruture. The next step is to compare each part of the ground-truth with the respective part of the OMR Result, comparing measure by measure through the comparison of tree elements.

Figure 42 – Proposed Method - Part 2 of Score Example, Bass Clef

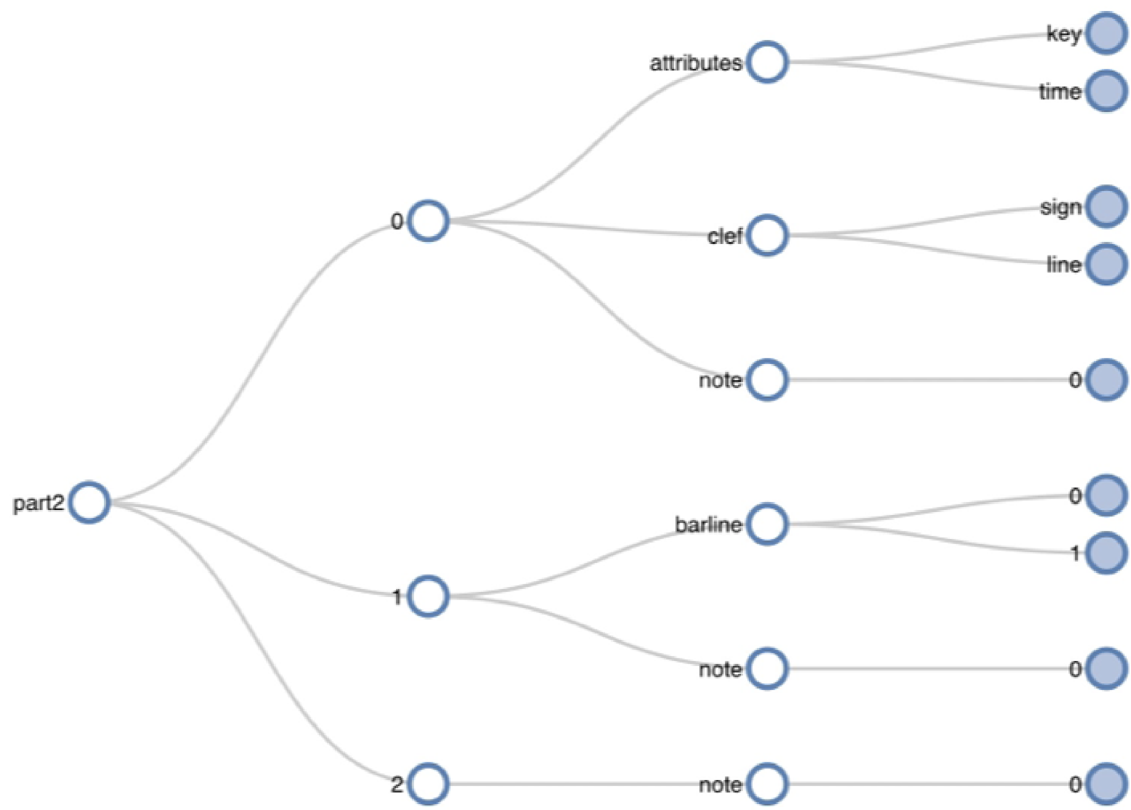
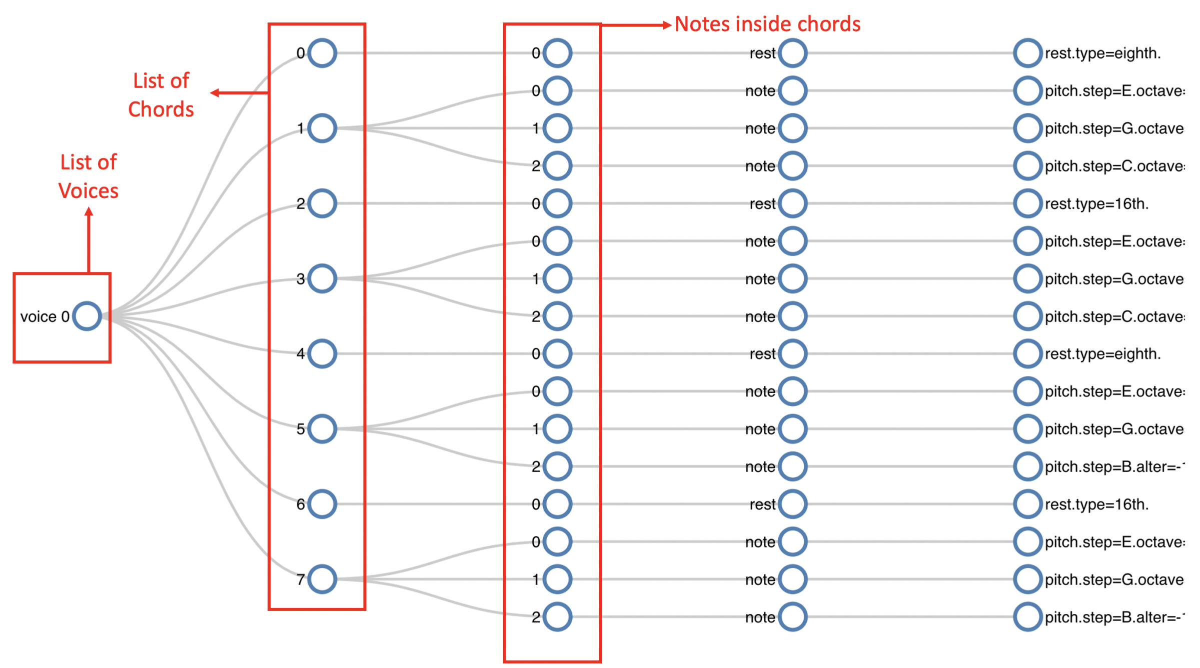


Figure 43 – Proposed Method - Measure 2 of Part 1 of Score Example



4.4 Hierarchical Conversion Algorithm

Basically, the conversion algorithm receives a MusicXML file and returns an object with lists and properties, similar to a JSON format. The algorithms are responsible for converting the different levels inside the model. Algorithm 1 starts splitting parts and processing all measures inside each part by calling algorithms 2 and 3. The latter calls algorithm 4 to import each note, voice and chord inside the measure as the structure shown on figure 37.

Algorithm 1 Function for Converting MusicXML file

```
function CONVERTXML(XMLFile)
    xmltree = XMLParser(XMLFile)
    musictree = {}
    partid = 0
    for each part ∈ xmltree.iter('part') do
        firstmeasure = part.iter('measure')                 ▷ Check for pair of clefs
        if count(firstmeasure.iter('clef')) > 1 then
            for each clef ∈ firstmeasure.iter('clef') do
                clefs[clef.attrib['number']] = {}
            end for
        else
            clefs[null] = {}
        end if
        for each measure ∈ part.iter('measure') do
            for each key ∈ clefs.keys() do
                measures = clefs[key]
                newmeasure = {}
                measures.append(newmeasure)
                procAttributes(measure, newmeasure, key)    ▷ See Alg. 2
                procMeasure(measure, newmeasure, key)       ▷ See Alg. 3
            end for
        end for
        for each key ∈ clefs.keys() do
            measures = clefs[key]
            partid = partid + 1
            musictree['part' + partid] = measures
        end for
    end for
    return musictree
end function
```

Algorithm 2 Function for Processing Measure Attributes

```
function procAttributes(measure, newmeasure, clefnum)
    newmeasure['attributes'] = {
        "key": {
            "fifths": measure['key']['fifths']
        },
        "time": {
            "beats": measure['time']['beats'],
            "beat-type": measure['time']['beat-type']
        }
    }
    for each clef ∈ measure.iter('clef') do
        if (clef.attrib['number'] = clefnum) then
            newmeasure['clef'] = {
                "fifths": measure['clef']['sign'],
                "beat-type": measure['clef']['line']
            }
        end if
    end for
end function
```

Algorithm 3 Function for Processing Measure

```
function procMeasure(measure, item, clefnum)
    newmeasure["barline"] = {
            "location": measure['barline']['location'],
            "repeat": measure['barline']['direction']
    }
    procNotes(measure, item, clefnum)                ▷ See Alg. 4
end function
```

Algorithm 4 Function for Processing Notes

```
function procNotes(measure, father, clefnum)
    voices = {}
    lastVoice = null
    for each note ∈ measure.iter('note') do
        type = note['type']
        staff = note['staff']
        if type and staff = clefnum then
            newNote = getNoteInfo(note)                    ▷ See Alg. 5
            chord = 'chord' ∈ note
            currentVoice = {}
            textVoice = note['voice']
            if textVoice = nullandchord then
                textVoice = lastVoice
            end if
            if textVoice ∈ voices then
                currentVoice = voices[textVoice]
            else
                voices[textVoice] = currentVoice
            end if
            lastVoice = textVoice
            if chord then currentVoice[-1].append(newNote)
            elsecurrentVoice.append(newNote)
            end if
        end if
    end for
    father['note'] = voices
end function
```

Algorithm 5 Function for Getting Note/Rest Information

```
function getNoteInfo(note)
    newNote =
    noterest = ""
    typerest = False
    if ('rest' ∈ note) then
        noterest = "rest."
        typerest = True
    else
        pitch = note['pitch']['step'] + "." + note['pitch']['alter'] + "." + note['pitch']['octave'] + "."
        noterest = "pitch." + pitch
    end if
    noterest = noterest + "type=" + type + "."
    if 'dot' ∈ note then
        noterest = noterest + 'dot'
    end if
    if typerest then
        newNote['rest'] = noterest
    else
        newNote['note'] = noterest
    end if
    return newNote
end function
```

4.5 Alignment Algorithm

In order to handle all alignment cases described on section 2.5.2, the General Method for Sequence Comparison(NEEDLEMAN; WUNSCH, 1970) was adapted for comparing measures, voices, chords and notes.

4.5.1 Needleman–Wunsch Algorithm

The General Method for Sequence Comparison(NEEDLEMAN; WUNSCH, 1970) was proposed for the sequence comparison of amino acids in proteins, finding the largest number of matches between two proteins. It is based on a matrix used to represent all possible combinations between two sequences. Initially, 1 is the assumed value for a match, otherwise zero. However, the same work talks about the "sophistication" of the method using different values for matchs, gaps or other theories about the significance of the amino acids. To simplify the understanding of the method, let's consider comparing strings of characters. Matches will correspond to the value 1, while misses and spaces, sometimes called gaps, will correspond to the value -1. If two strings are chosen as "abc" and "abec", the method proposes the creation of a matrix with the height equals the length of the first string plus one (3 + 1) and the width equals the length of the second string plus one (4 + 1). A perfect diagonal inside the matrix would represent a perfect match between the two strings. Each horizontal dislocation means a gap on the first string, while each vertical dislocation means a gap on the second string. Figure 44 shows the initial values, starting from zero and increasing one gap horizontally and vertically. The first cell to be calculated is the yellow one, on position [1,1]. The value will be the higher among 3 values:

- The sum of previous diagonal cell [0,0] (0) with the comparison between the characters of the equivalent position: a = a? (1) In this case, the result is 1.
- The sum of the value of the cell above (-1) with space value (-1). In this case, -2.
- The sum of the value of the cell on left (-1) with space value (-1). In this case, -2.

So the value calculated for the yellow cell is the higher, 1. The same is done to each empty cell of the matrix, resulting the matrix on figure 45. The best path can be obtained starting from the last cell and following the calculations results towards the beginning as indicated by the red arrows.
Algorithm 6 shows the construction of the matrix, where fc is the comparison function, shown on algorithm 7, and d is the space value, or gap value, -1 in this example. With the understanding of the Sequence-Comparison algorithm, it's possible to "sophisticate" the comparison function to deal with complex objects instead of characters. Those custom comparison functions are described bellow.

Figure 44 – Alignment Algorithm - Needleman-Wunsch Matrix - 01

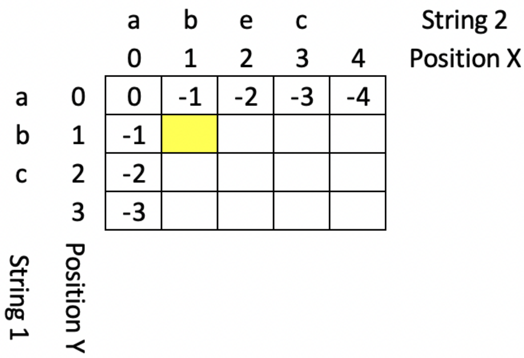
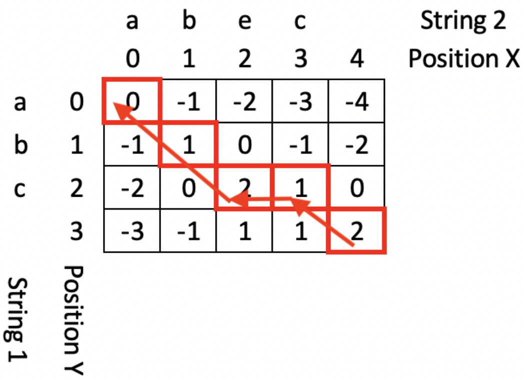


Figure 45 – Alignment Algorithm - Needleman-Wunsch Matrix - 02



4.5.2 Custom Algorithms

A set of custom comparison algorithms was designed to handle the comparison between trees of notes and measures to handle Alignment case #1 (2.5.2.1. The comparison function used to align measures is algorithm 8 which starts by creating a list of possible permutations of voices list as shown on figure 37 in order to handle Alignment Case #2 (2.5.2.2). This is done to make sure all possible combinations are tested to get the best result. For instance, if there are 3 voices, the possible combinations are: 123, 132, 213, 231, 312, 321. At the end, the best result is considered. Each comparison is made by the next algorithm 9 that works also on alignment of chords as Alignment Case #3 (2.5.2.3), as on algorithm 10, in order to identify missing notes or nonexistent notes among correct detected notes. The comparison of chords on algorithm 11 is made by comparing the list of notes of each chord. The last algorithms 13, 14, 15 and 12 are used to support the score calculation based on error level. Some constants used on the algorithms are: $measureErrorRate = 50$, $levelRate = 10$, $maxDepthLevel = 4$ and $needlemanMatrixSpace = -200$. The last one is used as

Algorithm 6 Needleman–Wunsch Matrix

```
function NEEDLEMANMATRIX(var1, var2, fc, d)
    max1 = length(var1) + 1
    max2 = length(var2) + 1
    for i = 0, i < max1, i++ do
        af[i][0] = d * i
    end for
    for j = 0, j < max2, j++ do
        af[0][j] = d * i
    end for
    for i = 1, i < max1, i++ do
        for j = 1, j < max2, j++ do
            match = af[i - 1][j - 1] + fc(var1[i - 1], var2[j - 1])
            delete = af[i - 1][j] + d
            insert = af[i][j-1] + d
            af[i][j] = max(match, delete, insert)
        end for
    end for
    return af
end function
```

Algorithm 7 Simple Comparison Function

```
function FC(c1, c2)
    if c1 = c2 then
        return 1
    else
        return -1
    end if
end function
```

the cost of missing measures. The Neddleman's algorithm (NEEDLEMAN; WUNSCH, 1970) is called as: *needlemanAlign(groundtruthPart, omrResultPart, fcCompareMeasures)* After the alignment of measures and voices made by algorithms describe above, the alignment algorithm is called one more time for the chords inside each voice for all measures in order to handle Alignment Case #4 (2.5.2.4). After that all possible results are analyzed and the best score is selected.

Algorithm 8 Function for Comparing Measures

```
function fcCompareMeasures(measure1, measure2)
    possible = permutations(measure2['note'])
    bestIndex = -1
    bestValue = 0
    i = 0
    difs = list()
    for each possiblePermut ∈ possible do
        pontos = compareVoices(var1['note'], possiblePermut)
        if bestValue < pontos or bestIndex < 0 then
            bestValue = pontos
            bestIndex = i
        end if
        i = i + 1
    end for
    return bestValue
end function
```

Algorithm 9 Function for Comparing Voices

```
function compareVoices(voice1, voice2)
    voicesCount = min(length(var1), length(var2))
    total = 0
    for i = 0, i < voicesCount, i++ do
        total = total + alignChords(voice1, voice2, i)
    end for
    return total
end function
```

Algorithm 10 Function for Align Chords

```
function alignChords(voice1, voice2, index)
    var1 = voice1[index]
    var2 = voice2[index]
    chordsCount = min(length(var1), length(var2))
    for seq = 0, i < chordsCount, seq++ do
        size = max(length(var1[seq]), length(var2[seq]))
        while length(var1[seq]) < size do
            var1[seq].append(∅)
        end while
        while length(var2[seq]) < size do
            var2[seq].append(∅)
        end while
        possible = permutations(var2[seq])
        bestIndex = -1
        bestValue = 0
        i = 0
        difs = list()
        for each possiblePermut ∈ possible do
            ddiff = compareNotes(var1[seq], possiblePermut, index, seq)
            pontos = calcDeepdiff(ddiff)
            if bestValue < pontos or bestIndex < 0 then
                bestValue = pontos
                bestIndex = i
            end if
            i = i + 1
        end for
    end for
    return bestValue
end function
```

Algorithm 11 Function for Comparing Notes

```
function compareNotes(lst1, lst2, index1, index2)
    text = "note"
    dd['values_changed'] = {}
    dd['dictionary_item_added'] = {}
    dd['dictionary_item_removed'] = {}
    notesCount = min(length(lst1), length(lst2))
    total = 0
    for i = 0, i < notesCount, i++ do
        if lst1[i] = ∅ then
            if lst2[i] ≠ ∅ then
                dd['dictionary_item_added'].append("root['" + text + "']["  + str(index1) + "][" + str(index2) + "]" + extractType(lst2[i]))
            end if
        else
            if lst2[i] = ∅ then
                dd['dictionary_item_removed'].append("root['" + text + "'][" + str(index1) + "][" + str(index2) + "]" + extractType(lst1[i]))
            else
                if lst1[i] ≠ lst2[i] then
                    dd['values_changed'].append("root['" + text + "'][" + str(index1) + "][" + str(index2) + "]" + extractType(lst2[i]))
                end if
            end if
        end if
    end for
    return dd
end function
```

Algorithm 12 Function for Extract Note's Type

```
function extractType(item)
    if "pitch" ∈ item then
        return "['note']"
    else
        return "['rest']"
    end if
end function
```

Algorithm 13 Function for Calculate Points of a Comparison Result

```
function calcDeepdiff(diffdict)
    ret = 0
    ret = ret + countScoreLevel(diffdict, 'dictionary_item_added')
    ret = ret + countScoreLevel(diffdict, 'dictionary_item_removed')
    ret = ret + countScoreLevel(diffdict, 'values_changed')

    return ret
end function
```

Algorithm 14 Function for Calculate Score Based on Key

```
function countScoreLevel(diffdict, key)
    totalScore = 0
    if key ∈ item then
        listItens = diffdict[key]
        for each item ∈ listItens do
            itemScore = calcItemScore(item)
            totalScore = totalScore + itemScore
        end for
    end if
    return totalScore
end function
```

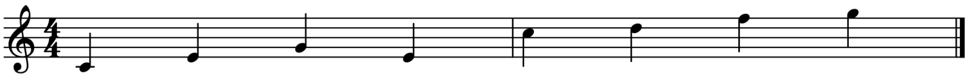
Algorithm 15 Function for Calculate Item Score

```
function calcItemScore(item)
    itemScore = 0
    if "attributes" ∈ item then
        level = 4
    else
        level = length(splitLevels(item))
    end if
    itemScore = max((maxDepthLevel - level) * levelRate, 1)
    return itemScore
end function
```

4.6 Simulation's Scenarios and Outcomes

All scenarios proposed in section 2.5.1 have been used as references during the method development and evolution. With the implementation of the method, all of them were used as test cases and the outcomes are presented bellow. To facilitate the understanding, some images are replicated here as the ground-truth on image 46.

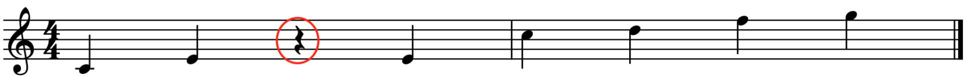
Figure 46 – Outcomes - Music Score Example - Ground-truth



4.6.1 Error Case #1: Rest Instead of Note

Figure 47 shows a simple metric (accuracy) and the detailed output of the proposed method. Despite the error points are the same, the method provides additional information about the error.

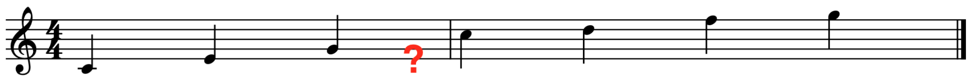
Figure 47 – Outcame - Error Case 1



```
Standard Metric - accuracy: 7/8

Proposed Method Output:
Alignment File 1: MM
Alignment File 2: MM
Measure: 1
.. part1
.... root['note'][0][2][0]['rest'] - non_existent_items - 1
.... root['note'][0][2][0]['note'] - missing_items - 1

Total Score: 2 error points
```

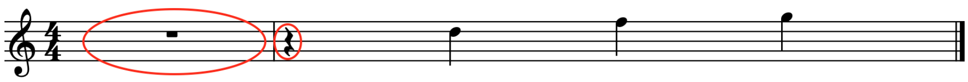
4.6.2 Error Case #2: Incorrect Music Score Due to Missing Note

Figure 48 shows a simple metric (accuracy) and the detailed output of the proposed method. For this example, when we look at the accuracy and the proposed method output, it becomes clear that the standard metric is not able to discern between the different type of errors when comparing the output results of 7/8 in error case 2 with the output result of 7/8 in the error case 1. The proposed method goes beyond this limitation and is be able to pinpoint the differences between different types of errors.

Figure 48 – Outcome - Error Case 2

```
Standard Metric – accuracy: 7/8

Proposed Method Output:
Alignment File 1: MM
Alignment File 2: MM
Measure: 1
.. part1
.... root['note'][0][3][0]['note'] – missing_items – 1

Total Score: 1 error point
```

4.6.3 Error Case #3: Partial Recognition

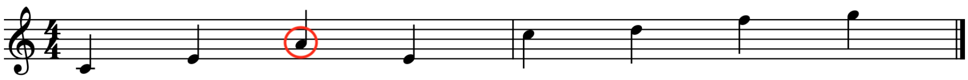
Figure 49 shows the capacity of handling different types of errors with their related error points. The first measure indicates the absence of notes. The measure is detected, but empty. So the error level is higher. While second measure presents a missing note and, also, the detection of a rest instead.

Figure 49 – Outcome - Error Case 3

```
Standard Metric – accuracy: 3/8

Proposed Method Output:
Alignment File 1: MM
Alignment File 2: MM
Measure: 1
.. part1
.... root['note'][0] – missing_items – 20
Measure: 2
.. part1
.... root['note'][0][0][0]['rest'] – non_existent_items – 1
.... root['note'][0][0][0]['note'] – missing_items – 1

Total Score: 22 error points
```

4.6.4 Error Case #4: Pitch Error

Figure 50 is an example where the proposed method shines while accuracy ignores the error.

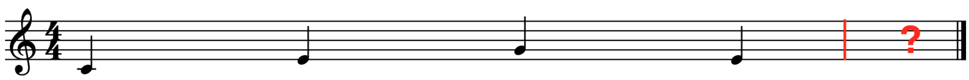
Figure 50 – Outcome - Error Case 4



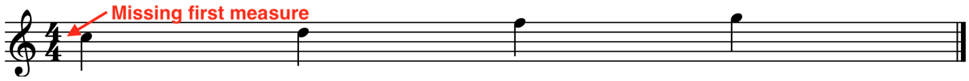
```
Standard Metric - accuracy: 8/8

Proposed Method Output:
Alignment File 1: MM
Alignment File 2: MM
Measure: 1
.. part1
.... root['note'][0][2][0]['note'] - incorrect_values - 1

Total Score: 1 error point
```

4.6.5 Error Case #5: Missing Measures

Figures 51 and 52 demonstrate that the proposed method can handle the alignment problem. The information of the missing measure is easily visible on "Alignment File 2". For the second scenario, as the first measure is missing, the method also shows information of clefs and attributes that should be on the first measure.

Figure 51 – Outcome - Error Case 5 - Second Measure Missing


```
Standard Metric - accuracy: 4/8

Proposed Method Output:
Alignment File 1: MM
Alignment File 2: M_
Measure: 2
.. part1
.... - missing_measures - 50

Total Score: 50 error points
```

Figure 52 – Outcome - Error Case 5 - First Measure Missing


```
Standard Metric – accuracy: 4/8 (0/8 if positions are considered)

Proposed Method Output:
Alignment File 1: MM
Alignment File 2: _M
Measure: 1
.. part1
....  – missing_measures – 50
Measure: 2
.. part1
.... root['attributes'] – non_existent_items – 1
.... root['clef'] – non_existent_items – 30

Total Score: 81 error points
```

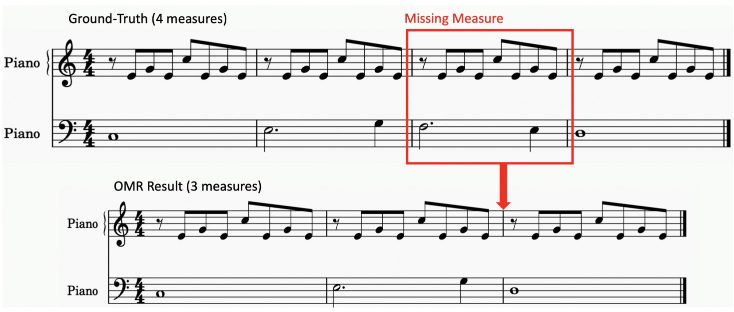
4.7 Alignment Cases

The following alignment cases will demonstrate different types of alignment problems being handled, considering the m̈easure error rate≐ 50 points, therefore each missing measure or non existing measure will count as 50 points.

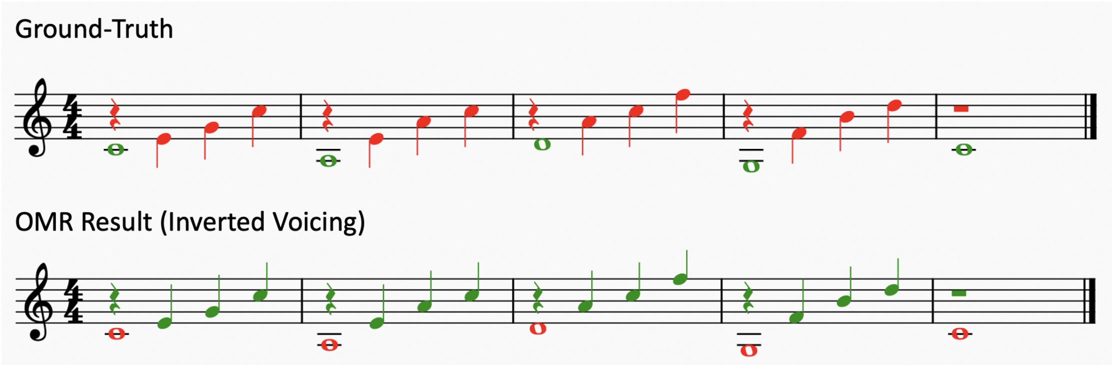
4.7.1 Alignment Case #1: Measure Alignment

53 shows the result of the method and, also, the detected alignment.

Figure 53 – Outcome - Aligment Case 1




```
Score: 100
Alignment File 1: MMMM
Alignment File 2: MM_M
Measure: 3
.. part1
....  – missing_measures – 50
.. part2
....  – missing_measures – 50
```

Figure 54 – Outcome - Aligment Case 2


```
Score: 0
Alignment File 1: MMMMM
Alignment File 2: MMMMM
```

4.7.2 Alignment Case #2: Voice Alignment

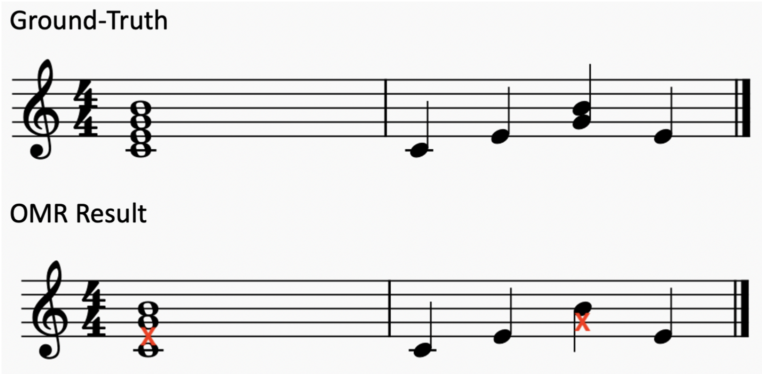
In figure 54 red color represents voice 1 and green color represents voice 2, and despite the inverted order, the method recognizes the result as a perfect match by alternating voices and searching for the best fit.

4.7.3 Alignment Case #3: Notes Inside Chords Alignment

As explained on chapter 2, this case represents two different situations: comparing chords with missing notes and comparing chord with a lonely note. For the first situation, the alignment algorithm do the job, but the second situation was the reason for the hierarchy model to use chords even for lonely notes. The comparison is always made between list of notes, even if there is only one note, it's represented as a list of one note. Figure 55 shows the result of the method and can be analyzed as:
On measure 1, part1, first voice [0], first chord/list of notes [0], second note [1] is missing.
On measure 2, part1, first voice [0], third chord/list of notes [2], first note [0] is missing.

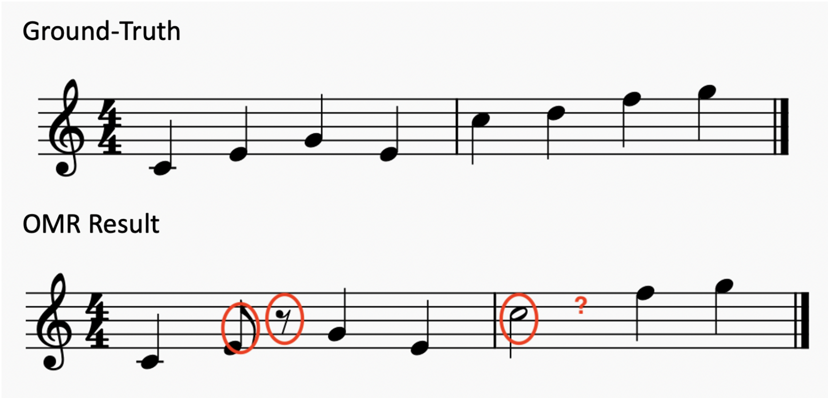
4.7.4 Alignment Case #4: Notes/Chord Alignment Inside Measures

Figure 56 shows the result of the method and can be analysed as:
On measure 1, part1, first voice [0], second chord/list of notes [1], first note [0] is incorrect.
On measure 1, part1, first voice [0], third chord/list of notes [1], first rest [0] does not exist.
On measure 2, part1, first voice [0], first chord/list of notes [0], first note [0] is missing.

Figure 55 – Outcome - Aligment Case 3


```
Score: 2
Alignment File 1: MM
Alignment File 2: MM
Measure: 1
.. part1
.... root['note'][0][0][1] - missing_items - 1
Measure: 2
.. part1
.... root['note'][0][2][0] - missing_items - 1
```

On measure 2, part1, first voice [0], second chord/list of notes [1], first note [0] is incorrect.

Figure 56 – Outcome - Aligment Case 4


```
Score: 4
Alignment File 1: MM
Alignment File 2: MM
Measure: 1
.. part1
.... root['note'][0][1][0]['note'] - incorrect_values - 1
.... root['note'][0][2][0]['rest'] - non_existent_items - 1
Measure: 2
.. part1
.... root['note'][0][0][0]['note'] - missing_items - 1
.... root['note'][0][1][0]['note'] - incorrect_values - 1
```

4.8 Dataset

The Synthetic Score Database by Christoph Dalitz[3] was chosen as the dataset to create the Ground-truth files, because it was the most used dataset for OMR works for printed scores on the research presented on Chapter 3. It's a test set from Gamera Kit used to evaluate staff removal algorithms, contains historical scores, modern scores and tablatures. The images are in PNG format with 300dpi and were generated from 32 music scores using different softwares.
For this work, each one of the 18 pages of modern music scores was manually edited in order to generate the ground-truth MusicXML file.

4.9 Current Method Limitations

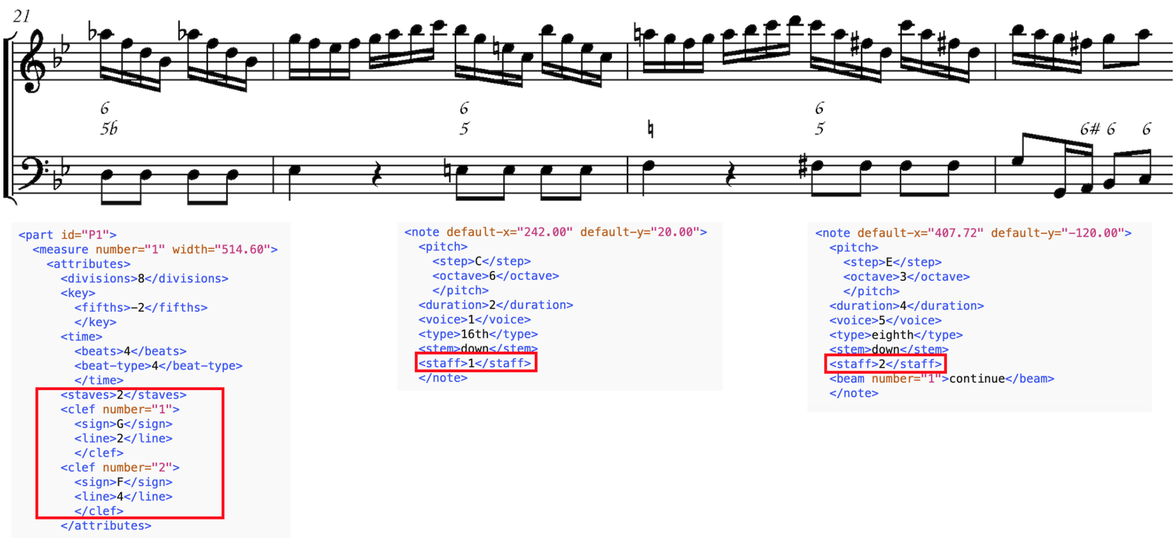
As described on the method description, for this first version, only systems, measures, attributes, notes and rests are considered. Those are the basic and essential elements for music scores. Another symbols as ornaments, dynamics and articulations should be addressed on future works.

[3] <http://gamera.informatik.hsnr.de/addons/musicstaves/testset-musicstaves.tar.gz>

4.10 Proposed Method Description

There are also cases where a pair of staves is represented as one only "part" in the MusicXML. This usually happens when it's a piano pair of staves. In those cases, many OMR systems break it in two parts, however some OMR systems and music editors join both in one "part". Figure 57 shows another example from Dalitz's database where there is a pair of piano staves. In that case, the method breaks the piano "part" into two independent parts taking only the notes belonging to each part allowing the correct comparison for both kinds of MusicXML representation.

Figure 57 – Proposed Method - Parts and Measures - Bellinzani score from Dalitz



4.11 Final Considerations

In this chapter, we explained the proposed evaluation method, describing the initial hierarchical conversion, detailing its algorithm and the alignment algorithms with their customisation. We also presented the simulations scenarios and their outcomes, confirming the expected behavior. The dataset and some method limitations were also provided.

5 Experiments

In this section we present some experiments that were performed with the goal of better evaluating the proposed method and the technical aspects involved with comparing MusicXML files.

5.1 Experimental Setup

For this experiment, we have used the 18 ground-truth MusicXML files created based on the 18 scores of The Synthetic Score Database by Christoph Dalitz[1]. In order to have OMR results, we have chosen 2 different cloud-based OMR systems:

1. The Musescore free experimental service that convert PDF Sheet music.[2]
2. Newzik Web convertion from PDF to a LiveScore.[3]

5.2 Experimental Results

Table 11 shows the result scores of running the open method implementation on the OMR results of both systems.

Newzik scored perfectly on files buxtehude.pdf and williams.pdf, but has failed converting dalitz03.pdf, while Musescore has failed to convert bellinzani.pdf and had most of the worst scores.

5.2.1 Detailed Analysis

In order to better understanding the errors, we will analyze detailed Musescore's results for buxtehude.pdf. The public implementation of the method shows the detailed information as the example presented in Figure 58:

The note or rest error is indicated as:
['note'][voice][chord][note]['rest or note'], so the example *['note'][0][2][0]['note'] - incorrect values* indicate that the note on the first voice third position (zero based) is wrong. As it's not a chord it's represented as a "chord" of one note.

Let's verify each part of the result using images to show the errors.

[1] http://gamera.informatik.hsnr.de/addons/musicstaves/testset-musicstaves.tar.gz
[2] https://musescore.com/import
[3] https://newzik.com/

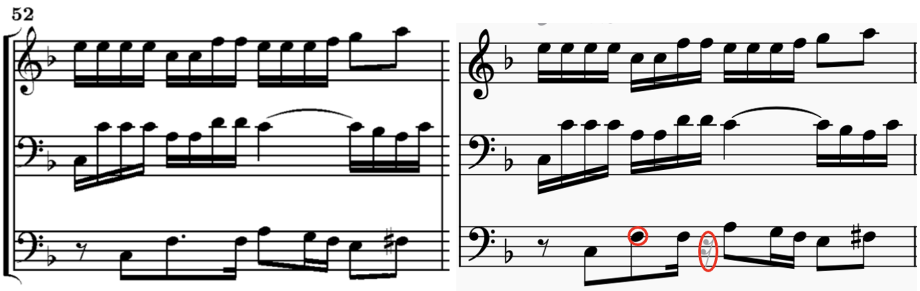
Table 11 – Experiments - Experimental Results for The Synthetic Score Database by Christoph Dalitz

Original	**PDF**	**MusicXML**	**Musescore**	**Newzik**
bach.png	bach.pdf	bach.xml	507	609
bellinzani.png	bellinzani.pdf	bellinzani.xml	Failed	225
brahms02.png	brahms02.pdf	brahms02.xml	118	8
bruckner01.png	bruckner01.pdf	bruckner01.xml	54	72
buxtehude.png	buxtehude.pdf	buxtehude.xml	44	0
carcassi01.png	carcassi01.pdf	carcassi01.xml	17	2
dalitz03.png	dalitz03.pdf	dalitz03.xml	571	Failed
diabelli.png	diabelli.pdf	diabelli.xml	121	119
mahler.png	mahler.pdf	mahler.xml	342	283
pmw01.png	pmw01.pdf	pmw01.xml	275	189
pmw03.png	pmw03.pdf	pmw03.xml	143	106
pmw04.png	pmw04.pdf	pmw04.xml	602	512
rameau.png	rameau.pdf	rameau.xml	955	322
schumann.png	schumann.pdf	schumann.xml	783	414
tye.png	tye.pdf	tye.xml	306	66
victoria09.png	victoria09.pdf	victoria09.xml	347	312
wagner.png	wagner.pdf	wagner.xml	495	448
williams.png	williams.pdf	williams.xml	12	0

Figure 58 – Experiments - Detailed Analysis Example for buxtehude.pdf

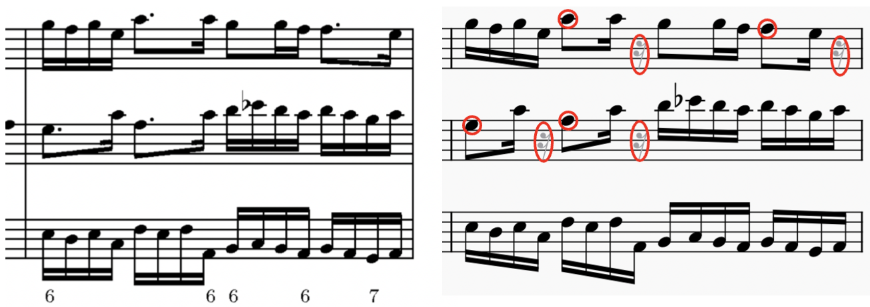
```
Score: 44
Alignment File 1: MMMMMMMMMMMM
Alignment File 2: MMMMMMMMMMMM
Measure: 1
.part3
..['note'][0][2][0]['note'] - incorrect_values - 1
..['note'][0][4][0]['rest'] - non_existent_items - 1
Measure: 3
.part1
..['note'][0][4][0]['note'] - incorrect_values - 1
..['note'][0][6][0]['rest'] - non_existent_items - 1
..['note'][0][10][0]['note'] - incorrect_values - 1
..['note'][0][12][0]['rest'] - non_existent_items - 1
.part2
..['note'][0][0][0]['note'] - incorrect_values - 1
..['note'][0][2][0]['rest'] - non_existent_items - 1
..['note'][0][3][0]['note'] - incorrect_values - 1
..['note'][0][5][0]['rest'] - non_existent_items - 1
Measure: 4...and so on
```

Let's verify each part of the result using Figures 59 to 69. In these figures, on the left side of the image is a measure of the original score, on right side of the image if the output of the Newzik Web software, with the errors highlighted and in the lower part of the image if the output of the proposed system which is able to automatically identify the different types of errors.

Figure 59 – Experiments - buxtehude - Detailed Analysis - Measure 1


```
Measure: 1
.part3
..['note'][0][2][0]['note'] - incorrect_values - 1
..['note'][0][4][0]['rest'] - non_existent_items - 1
```

Figure 60 – Experiments - buxtehude - Detailed Analysis - Measure 3




```
Measure: 3
.part1
..['note'][0][4][0]['note'] - incorrect_values - 1
..['note'][0][6][0]['rest'] - non_existent_items - 1
..['note'][0][10][0]['note'] - incorrect_values - 1
..['note'][0][12][0]['rest'] - non_existent_items - 1
.part2
..['note'][0][0][0]['note'] - incorrect_values - 1
..['note'][0][2][0]['rest'] - non_existent_items - 1
..['note'][0][3][0]['note'] - incorrect_values - 1
..['note'][0][5][0]['rest'] - non_existent_items - 1
```

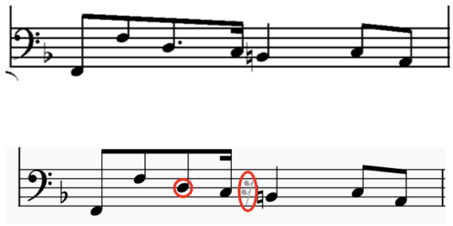
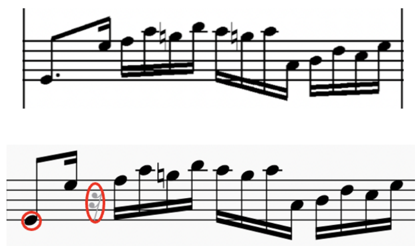
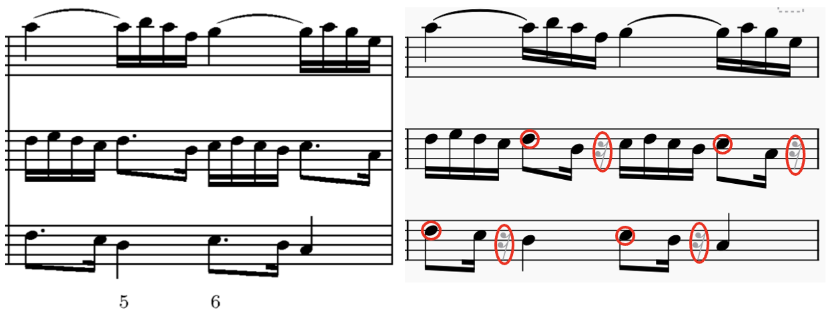
Figure 61 – Experiments - buxtehude - Detailed Analysis - Measure 4

```
Measure: 4
.part3
..['note'][0][2][0]['note'] – incorrect_values – 1
..['note'][0][4][0]['rest'] – non_existent_items – 1
```

Figure 62 – Experiments - buxtehude - Detailed Analysis - Measure 5

```
Measure: 5
.part2
..['note'][0][0][0]['note'] – incorrect_values – 1
..['note'][0][2][0]['rest'] – non_existent_items – 1
```

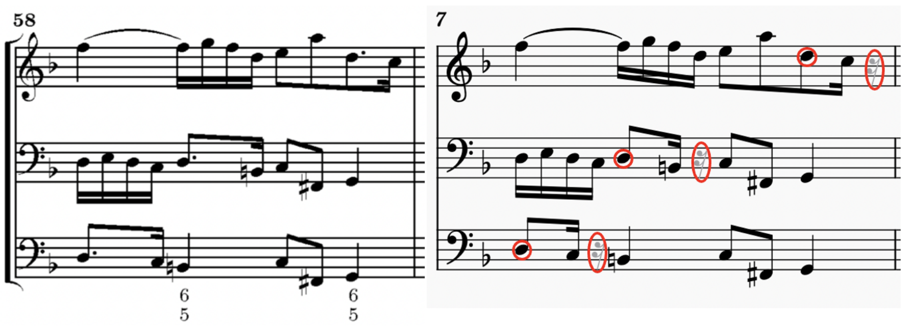
5.2.2 Measure Alignment Case

The music score bellinzani.pdf has a peculiar property, some of its measures begin on one line and end on the next as shown on figure 70. This means that one half of the measure is separated of the other half. This can confuse the OMR algorithm to detect two measures instead of one.

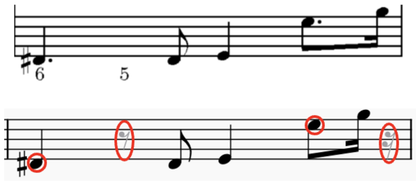
Analyzing the detailed information provided by the method when comparing Newzik OMR result, it's possible to see the measure alignment working as expected. The method highlights the missing notes of the measure and the non-existing measures, while adjusts the position of the measures to keep comparing them correctly.

Figure 63 – Experiments - buxtehude - Detailed Analysis - Measure 6


```
Measure: 6
.part2
..['note'][0][4][0]['note'] - incorrect_values - 1
..['note'][0][6][0]['rest'] - non_existent_items - 1
..['note'][0][11][0]['note'] - incorrect_values - 1
..['note'][0][13][0]['rest'] - non_existent_items - 1
.part3
..['note'][0][0][0]['note'] - incorrect_values - 1
..['note'][0][2][0]['rest'] - non_existent_items - 1
..['note'][0][4][0]['note'] - incorrect_values - 1
..['note'][0][6][0]['rest'] - non_existent_items - 1
```

Figure 64 – Experiments - buxtehude - Detailed Analysis - Measure 7


```
Measure: 7
.part1
..['note'][0][7][0]['note'] - incorrect_values - 1
..['note'][0][9][0]['rest'] - non_existent_items - 1
.part2
..['note'][0][4][0]['note'] - incorrect_values - 1
..['note'][0][6][0]['rest'] - non_existent_items - 1
.part3
..['note'][0][0][0]['note'] - incorrect_values - 1
..['note'][0][2][0]['rest'] - non_existent_items - 1
```

Figure 65 – Experiments - buxtehude - Detailed Analysis - Measure 9


```
Measure: 9
.part3
..['note'][0][0][0]['note'] - incorrect_values - 1
..['note'][0][1][0]['rest'] - non_existent_items - 1
..['note'][0][4][0]['note'] - incorrect_values - 1
..['note'][0][6][0]['rest'] - non_existent_items - 1
```

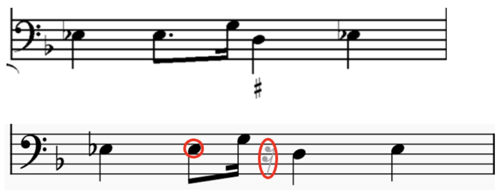
Figure 66 – Experiments - buxtehude - Detailed Analysis - Measure 10

```
Measure: 10
.part3
..['note'][0][1][0]['note'] - incorrect_values - 1
..['note'][0][3][0]['rest'] - non_existent_items - 1
```

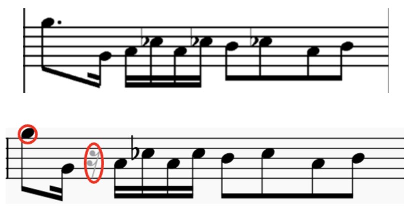
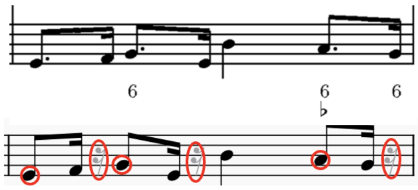
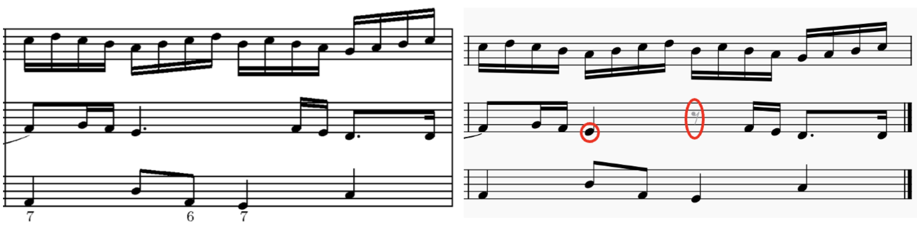
Figure 67 – Experiments - buxtehude - Detailed Analysis - Measure 11

```
Measure: 11
.part2
..['note'][0][0][0]['note'] - incorrect_values - 1
..['note'][0][2][0]['rest'] - non_existent_items - 1
```

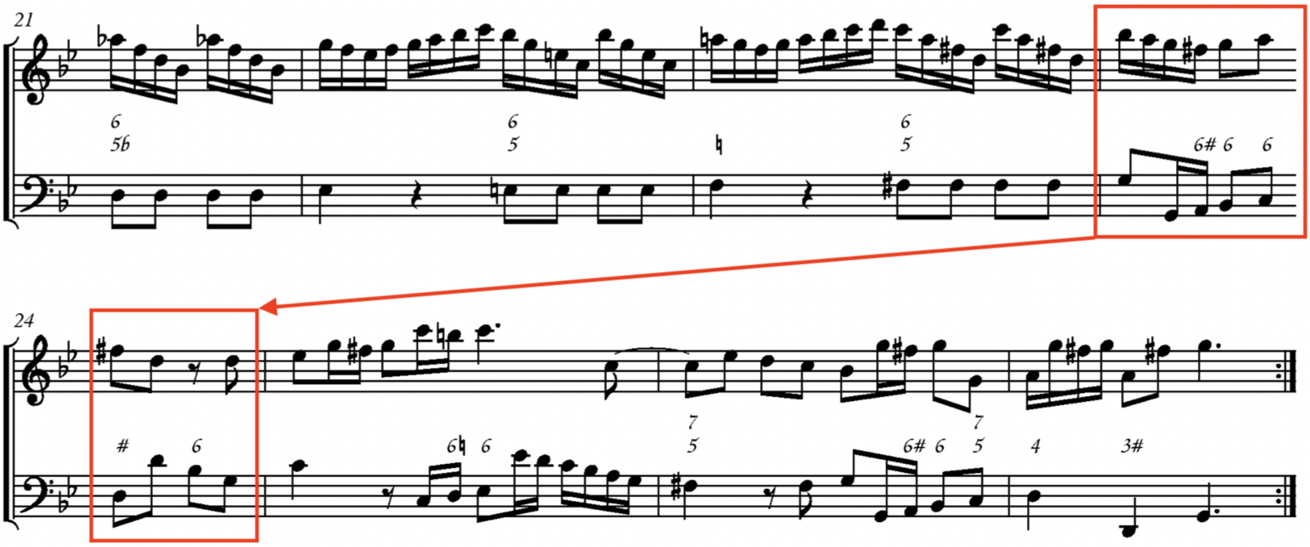
Figure 68 – Experiments - buxtehude - Detailed Analysis - Measure 12


```
Measure: 12
.part3
..['note'][0][0][0]['note'] - incorrect_values - 1
..['note'][0][2][0]['rest'] - non_existent_items - 1
..['note'][0][3][0]['note'] - incorrect_values - 1
..['note'][0][5][0]['rest'] - non_existent_items - 1
..['note'][0][7][0]['note'] - incorrect_values - 1
..['note'][0][9][0]['rest'] - non_existent_items - 1
```

Figure 69 – Experiments - buxtehude - Detailed Analysis - Measure 14


```
Measure: 14
.part2
..['note'][0][3][0]['note'] - incorrect_values - 1
..['note'][0][4][0]['rest'] - non_existent_items - 1
```

Figure 70 – Experiments - Measure Alignment Case - Bellinzani - Divided Measure




```
Score: 225
Alignment File 1: MMMM_MMMMMMMMM_MM
Alignment File 2: MMMMMMMMMMMMMMMMM
Measure: 4
.part1
..['note'][0][6][0]['note'] – missing_items – 1
..['note'][0][7][0]['note'] – missing_items – 1
..['note'][0][8][0]['rest'] – missing_items – 1
..['note'][0][9][0]['note'] – missing_items – 1
.part2
..['note'][0][5][0]['note'] – missing_items – 1
..['note'][0][6][0]['note'] – missing_items – 1
..['note'][0][7][0]['note'] – missing_items – 1
..['note'][0][8][0]['note'] – missing_items – 1
Measure: 5
.part1
.– non_existent_measures – 50
.part2
.– non_existent_measures – 50
Measure: 8 ..an so on
```

5.3 Source Code and Dataset Available

The public implementation and the ground-truth dataset in MusicXML are available on GitHub [4]. The folder structure is described bellow:

- `omr-comparison` .... (main python programs)
  - `alignment-studies` .... (string alignment(NEEDLEMAN; WUNSCH, 1970))
  - `dataset` .... (datasets, ground-truth and OMR outputs)
    - `dalitz_modern` .... (PNG images from Dalitz Dataset)
    - `fur_elise` .... (simple score with the first part of Fur Elise - Beethoven)
    - `musicxml` .... (Ground-truth MusicXML files)
    - `pdf` .... (PDF generated by PNG files on dalitz_modern)
    - `result_musescore` .... (MusicXML files generated by Musescore OMR)
    - `result_newzik` .... (MusicXML files generated by Newzik)
    - `scenarios` .... (MusicXML files for all scenarios on experiments)
  - `output` .... (Method output for all Musescore and Newzik OMR)

The python programs used to conduce the experiments are available on the root folder:

- `omr-comparison`
  - `compare-all.py` .... (program used to generate the table 11)
  - `compare-scenarios.py` .... (program used to generate the output for 4.6)
  - `compare-alignments.py` .... (program used to generate the output for 4.7)
  - `compare-ds.py` ... (program to compare and show details of a chosen dataset file)
  - `omr_comparison.py` .... (full implementation for 4)

5.4 Final Considerations

In this chapter, we described the setup used to conduct the experiments. The full dataset was converted using Musescore and Newzik OMR systems and all score results were presented. We also provided a detailed description of each error on the Buxtehude music score. One case of measure alignment is also described using the Bellinzani music score. The results highlight the rich level of detail provided by the method. The source code and the dataset were shared with the research community.

[4] https://github.com/mengarelli/omr-comparison

6 Conclusion

Along this work, we presented a practical new computer-based method to compare results among OMR systems. The public implementation and the ground-truth dataset in MusicXML are already available on GitHub https://github.com/mengarelli/omr-comparison. Therefore, a new door is open, enabling other researchers to try the new method and help to improve it. New ground-truth files can be added to the dataset allowing the evolution of the method and the comparison regarding different levels of difficulty and kinds of challenge.

On Chapter 2 we presented an overview of OMR systems, their stages, most commonly used methods, metrics, issues and challenges. On Chapter 3 we conducted a literature review following the recommended protocols in order to avoid bias and contribute to the community. Those chapters resulted a journal paper published in the Multimedia Tools and Applications Journal in 2020 (MENGARELLI et al., 2020).

On chapters 4 and 5 we presented the new method, it's structures, all algorithms, experiments and results, generating another journal paper waiting for publishing in the Multimedia Tools and Applications Journal.

As stated at the beginning of this work, there was not an automatic method to evaluate the performance of an OMR system over a given ground-truth. (HAJIC JR, 2018) says "The major problem in OMR evaluation is that given a ground truth encoding of a score and the output of a recognition system, there is no automatic method capable of reliably computing how well the recognition system performs that would (1) be rigorously described and evaluated, (2) have a public implementation, (3) give meaningful results." Answering that, the automatic method proposed in this work: (1) was rigorously described and evaluated, (2) has a public implementation, (3) gives meaningful results.

New symbols and structures should be added to the method. This first version does not handle ornaments, dynamics and articulations yet. Another possible improvement can be to handle staff detection or key signature errors without detecting all notes wrong, because despite musically they are wrong, it's only one mistaken staff or key signature.

The detailed information provided by the method can be used as feedback for machine learning methods, allowing self correction. The idea of using the same method to search for similarities can also be explored. The results provided by the method make it possible to generate both quantitative and qualitative analyses.

This work represents a big step on closing one of the open issues in the field and, hopefully, will be a great help to the OMR research community.

References

ADAMSKA, Julia; PIECUCH, Mateusz; PODGÓRSKI, Mateusz; WALKIEWICZ, Piotr; LUKASIK, Ewa. Mobile system for optical music recognition and music sound generation. *Proceedings of the IFIP International Conference on Computer Information and Industrial Management*, p. 571–582, 2015. Quoted 4 times on pages 47, 50, 52, and 53.

AG, Springer International Publishing. *Springer Link*. 2017. <<https://link.springer.com/>>. Acesso em: 2017-07-05. Quoted on page 42.

BAINBRIDGE, David; BELL, TC. Dealing with superimposed objects in optical music recognition. The Institution of Engineering and Technology, 1997. Quoted 3 times on pages 47, 48, and 49.

BARÓ, Arnau; RIBA, Pau; FORNÉS, Alicia. Towards the recognition of compound music notes in handwritten music scores. In: IEEE. *Proceedings of the 15th IEEE International Conference on Frontiers in Handwriting Recognition (ICFHR)*. [S.l.], 2016. p. 465–470. Quoted 6 times on pages 34, 47, 48, 49, 53, and 54.

BAUMANN, Stephan. A simplified attributed graph grammar for high-level music recognition. In: *Proceedings of the 3rd IEEE International Conference on Document Analysis and Recognition*. [S.l.: s.n.], 1995. v. 2, p. 1080–1083. Quoted on page 49.

BELLINI, Pierfrancesco; BRUNO, Ivan; NESI, Paolo. Assessing optical music recognition tools. *Computer Music Journal*, MIT Press, v. 31, n. 1, p. 68–93, 2007. Quoted 5 times on pages 10, 23, 49, 53, and 54.

BRUDER, Ilvio; IGNATOVA, Temenushka; MILEWSKI, Lars. Knowledge-based scribe recognition in historical music archives. *Lecture notes in computer science*, Springer, p. 304–316, 2004. Quoted on page 45.

BUGGE, Esben Paul; JUNCHER, Kim Lundsteen; MATHIASEN, Brian Søborg; SIMONSEN, Jakob Grue. Using sequence alignment and voting to improve optical music recognition from multiple recognizers. 2011. Quoted 5 times on pages 17, 25, 50, 53, and 54.

BURGOYNE, John Ashley; PUGIN, Laurent; EUSTACE, Greg; FUJINAGA, Ichiro. A comparative survey of image binarisation algorithms for optical recognition on degraded musical sources. In: *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*. [S.l.: s.n.], 2007. p. 509–512. Quoted 5 times on pages 47, 48, 49, 50, and 54.

B.V., Elsevier. *Scopus*. 2017. <<https://www.scopus.com/>. Acesso em: 2017-07-05. Quoted on page 42.

BYRD, Donald; SCHINDELE, Megan. Prospects for improving omr with multiple recognizers. In: *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*. [S.l.: s.n.], 2006. p. 41–46. Quoted 7 times on pages 17, 24, 25, 46, 50, 53, and 54.

BYRD, Donald; SIMONSEN, Jakob Grue. Towards a standard testbed for optical music recognition: Definitions, metrics, and page images. *Journal of New Music Research*, Taylor & Francis, v. 44, n. 3, p. 169–195, 2015. Quoted 4 times on pages 17, 23, 25, and 50.

CALVO-ZARAGOZA, Jorge; BARBANCHO, Isabel; TARDÓN, Lorenzo J; BARBANCHO, Ana M. Avoiding staff removal stage in optical music recognition: application to scores written in white mensural notation. *Pattern Analysis and Applications*, Springer, v. 18, n. 4, p. 933–943, 2015. Quoted 4 times on pages 47, 49, 50, and 52.

CALVO-ZARAGOZA, Jorge; JR, Jan Hajič; PACHA, Alexander. Understanding optical music recognition. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 53, n. 4, p. 1–35, 2020. Quoted 3 times on pages 16, 17, and 18.

CALVO-ZARAGOZA, Jorge; MICÓ, Luisa; ONCINA, Jose. Music staff removal with supervised pixel classification. *International Journal on Document Analysis and Recognition (IJDAR)*, Springer, v. 19, n. 3, p. 211–219, 2016. Quoted 3 times on pages 48, 49, and 52.

CALVO-ZARAGOZA, Jorge; ONCINA, Jose. Recognition of pen-based music notation with finite-state machines. *Expert Systems with Applications*, Elsevier, v. 72, p. 395–406, 2017. Quoted 3 times on pages 48, 49, and 52.

CALVO-ZARAGOZA, Jorge; PERTUSA, Antonio; ONCINA, Jose. Staff-line detection and removal using a convolutional neural network. *Machine Vision and Applications*, Springer, p. 1–10, 2017. Quoted 3 times on pages 48, 49, and 52.

CALVO-ZARAGOZA, Jorge; VIGLIENSONI, Gabriel; FUJINAGA, Ichiro. Staff-line detection on grayscale images with pixel classification. In: *Proceedings of Iberian Conference on Pattern Recognition and Image Analysis*. [S.l.: s.n.], 2017. p. 279–286. Quoted 3 times on pages 48, 49, and 52.

CAMPOS, Vicente Bosch; CALVO-ZARAGOZA, Jorge; TOSELLI, Alejandro H; RUIZ, Enrique Vidal. Sheet music statistical layout analysis. In: *Proceedings of the 15th IEEE International Conference on Frontiers in Handwriting Recognition (ICFHR)*. [S.l.: s.n.], 2016. p. 313–318. Quoted 2 times on pages 45 and 50.

CARDOSO, Jaime S; REBELO, Ana. Robust staffline thickness and distance estimation in binary and gray-level music scores. In: *Proceedings of the 20th IEEE International Conference on Pattern Recognition (ICPR)*. [S.l.: s.n.], 2010. p. 1856–1859. Quoted 3 times on pages 48, 49, and 50.

CHEN, Liang; DUAN, Kun. Midi-assisted egocentric optical music recognition. In: *Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. [S.l.: s.n.], 2016. p. 1–9. Quoted 5 times on pages 46, 47, 48, 50, and 52.

CHEN, Y.-S.; CHEN, F.-S.; TENG, C.-H. AN optical music recognition system for skew or inverted musical scores. *International Journal of Pattern Recognition and Artificial Intelligence*, v. 27, n. 7, 2013. Quoted 6 times on pages 16, 47, 49, 52, 53, and 54.

DALITZ, Christoph; DROETTBOOM, Michael; PRANZAS, Bastian; FUJINAGA, Ichiro. A comparative study of staff removal algorithms. *IEEE Transactions on Pattern Analysis* *and Machine Intelligence*, IEEE, v. 30, n. 5, p. 753–766, 2008. Quoted 4 times on pages 48, 49, 52, and 54.

DALITZ, Christoph; MICHALAKIS, Georgios K; PRANZAS, Christine. Optical recognition of psaltic byzantine chant notation. *International Journal of Document Analysis and Recognition (IJDAR)*, Springer, v. 11, n. 3, p. 143–158, 2008. Quoted 6 times on pages 18, 19, 20, 47, 48, and 50.

DINH, Cong Minh; YANG, Hyung Jeong; LEE, Guee Sang; KIM, Soo Hyung. Fast lyric area extraction from images of printed korean music scores. *IEICE TRANSACTIONS on Information and Systems*, The Institute of Electronics, Information and Communication Engineers, v. 99, n. 6, p. 1576–1584, 2016. Quoted 4 times on pages 47, 48, 49, and 52.

DUTTA, Anjan; PAL, Umapada; FORNES, Alicia; LLADOS, Josep. An efficient staff removal approach from printed musical documents. In: *Proceedings of the 20th IEEE International Conference on Pattern Recognition (ICPR)*. [S.l.: s.n.], 2010. p. 1965–1968. Quoted 4 times on pages 47, 48, 49, and 52.

FAHN, Chin-Shyurng; LU, Kai-Jay. Humanoid recognizing piano scores techniques. In: *Proceedings of the IEEE International Conference on Information Science, Electronics and Electrical Engineering (ISEEE)*. [S.l.: s.n.], 2014. v. 3, p. 1397–1402. Quoted 2 times on pages 47 and 50.

FANG, Yang; GUI-FA, Teng. Visual music score detection with unsupervised feature learning method based on k-means. *International Journal of Machine Learning and Cybernetics*, Springer, v. 6, n. 2, p. 277–287, 2015. Quoted 3 times on pages 47, 49, and 52.

FORNES, Alicia; DUTTA, Anjan; GORDO, Albert; LLADOS, Josep. The icdar 2011 music scores competition: Staff removal and writer identification. In: *Proceedings of the IEEE International Conference on Document Analysis and Recognition (ICDAR)*. [S.l.: s.n.], 2011. p. 1511–1515. Quoted 5 times on pages 48, 49, 50, 52, and 54.

FORNÉS, Alicia; DUTTA, Anjan; GORDO, Albert; LLADÓS, Josep. The 2012 music scores competitions: staff removal and writer identification. In: *Graphics Recognition. New Trends and Challenges*. [S.l.]: Springer, 2013. p. 173–186. Quoted 3 times on pages 50, 52, and 54.

FORNÉS, Alicia; KIEU, Van Cuong; VISANI, Muriel; JOURNET, Nicholas; DUTTA, Anjan. The icdar/grec 2013 music scores competition: Staff removal. In: SPRINGER. *International Workshop on Graphics Recognition*. [S.l.], 2013. p. 207–220. Quoted 4 times on pages 48, 49, 50, and 52.

FORNÉS, Alicia; LLADÓS, Josep; SÁNCHEZ, Gemma. Primitive segmentation in old handwritten music scores. In: SPRINGER. *International Workshop on Graphics Recognition*. [S.l.], 2005. p. 279–290. Quoted 3 times on pages 47, 49, and 52.

FORNÉS, Alicia; LLADÓS, Josep; SÁNCHEZ, Gemma. Old handwritten musical symbol classification by a dynamic time warping based method. In: SPRINGER. *International Workshop on Graphics Recognition*. [S.l.], 2007. p. 51–60. Quoted 5 times on pages 47, 49, 50, 52, and 54.

FORNES, Alicia; LLADÓS, Josep; SÁNCHEZ, Gemma; BUNKE, Horst. On the use of textural features for writer identification in old handwritten music scores. In: *Proceedings of the 10th IEEE International Conference on Document Analysis and Recognition (ICDAR)*. [S.l.: s.n.], 2009. p. 996–1000. Quoted 3 times on pages 47, 50, and 52.

FORNÉS, Alicia; SÁNCHEZ, Gemma. Analysis and recognition of music scores. In: *Handbook of Document Image Processing and Recognition*. [S.l.]: Springer, 2014. p. 749–774. Quoted 2 times on pages 48 and 50.

HAJIC JR, Jan. A case for intrinsic evaluation of optical music recognition. 2018. Quoted 2 times on pages 18 and 92.

HAJIC JR, Jan; DORFER, Matthias; WIDMER, Gerhard; PECINA, Pavel. Towards full-pipeline handwritten omr with musical symbol detection by u-nets. In: *Proceedings of the 19th International Conference on Music Information Retrieval (ISMIR)*. [S.l.: s.n.], 2018. p. 225–232. Quoted on page 17.

IEEE. *IEEE Xplore Digital Library*. 2017. <<http://ieeexplore.ieee.org/search/>>. Acesso em: 2017-07-05. Quoted on page 42.

İZMIRLI, Özgür; SHARMA, Gyanendra. Bridging printed music and audio through alignment using a mid-level score representation. In: *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*. [S.l.: s.n.], 2012. p. 61–66. Quoted 3 times on pages 45, 50, and 52.

JASTRZEBSKA, Agnieszka; LESINSKI, Wojciech. Optical music recognition as the case of imbalanced pattern recognition: a study of single classifiers. In: *Knowledge, Information and Creativity Support Systems: Recent Trends, Advances and Solutions*. [S.l.]: Springer, 2016. p. 493–505. Quoted 5 times on pages 34, 47, 48, 49, and 52.

JONES, Catherine Schmidt. *Understanding Basic Music Theory*. [S.l.: s.n.], 2022. Quoted on page 26.

KATO, Hirokazu; INOKUCHI, Seiji. A recognition system for printed piano music using musical knowledge and constraints. *Structured Document Image Analysis*, Springer Berlin Heidelberg, p. 435–455, 1992. Quoted 3 times on pages 25, 47, and 50.

KODIROV, Elyor; HAN, Sejin; LEE, Guee-Sang; KIM, YoungChul. Music with harmony: chord separation and recognition in printed music score images. In: ACM. *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*. [S.l.], 2014. p. 50. Quoted 3 times on pages 47, 50, and 52.

LESINSKI, Wojciech; JASTRZEBSKA, Agnieszka. Optical music recognition: Standard and cost-sensitive learning with imbalanced data. In: SPRINGER. *IFIP International Conference on Computer Information Systems and Industrial Management*. [S.l.], 2015. p. 601–612. Quoted 4 times on pages 47, 48, 49, and 52.

LIMITED, Informa UK. *Taylor & Francis Online*. 2017. <<http://www.tandfonline.com/>>. Acesso em: 2017-08-08. Quoted on page 42.

LIU, Xiaoxiang. Note symbol recognition for music scores. *Intelligent Information and Database Systems*, Springer, p. 263–273, 2012. Quoted 4 times on pages 47, 49, 53, and 54.

LIU, Xiaoxiang; ZHOU, Mi; XU, Peng. A robust method for musical note recognition. In: IEEE. *Computer-Aided Design and Computer Graphics (CAD/Graphics), 2015 14th International Conference on*. [S.l.], 2015. p. 212–213. Quoted 5 times on pages 47, 49, 52, 53, and 54.

LÓPEZ, B et al. Staff and graphical primitive segmentation in old handwritten music scores. *Artificial Intelligence Research and Development*, IOS Press, v. 131, p. 83, 2005. Quoted 3 times on pages 47, 49, and 50.

LUANGNAPA, Nawapon; SILPAVARANGKURA, Thongchai; NUKOOLKIT, Chakarida; MONGKOLNAM, Pornchai. Optical music recognition on android platform. *Advances in Information Technology*, Springer, v. 344, p. 106–115, 2012. Quoted 5 times on pages 25, 47, 50, 52, and 53.

LUCKNER, Marcin. Recognition of noised patterns using non-disruption learning set. In: *Proceedings of the 6th IEEE International Intelligente Systems Desing and Applications (ISDA)*. [S.l.: s.n.], 2006. v. 1, p. 557–562. Quoted 3 times on pages 47, 50, and 52.

MACHINERY, Association for Computing. *ACM Digital Library*. 2017. <<https://dl.acm.org/>>. Acesso em: 2017-07-05. Quoted on page 42.

MALIK, Rakesh; ROY, Partha Pratim; PAL, Umapada; KIMURA, Fumitaka. Handwritten musical document retrieval using music-score spotting. In: *Proceedings of the 12th IEEE International Conference on Document Analysis and Recognition (ICDAR)*. [S.l.: s.n.], 2013. p. 832–836. Quoted 2 times on pages 49 and 52.

MÄRGNER, Volker; ABED, Haikal El. Tools and metrics for document analysis systems evaluation. *Handbook of Document Image Processing and Recognition*, Springer, p. 1011–1036, 2014. Quoted 2 times on pages 48 and 50.

MEHTA, Apurva A; BHATT, Malay S. Optical music notes recognition for printed piano music score sheet. In: *Proceedings of the IEEE International Conference on Computer Communication and Informatics (ICCCI)*. [S.l.: s.n.], 2015. p. 1–6. Quoted 4 times on pages 47, 49, 50, and 52.

MENGARELLI, Luciano; KOSTIUK, Bruno; VITÓRIO, João G; TIBOLA, Maicon A; WOLFF, William; SILLA, Carlos N. Omr metrics and evaluation: a systematic review. *Multimedia Tools and Applications*, Springer, v. 79, n. 9, p. 6383–6408, 2020. Quoted 9 times on pages 8, 16, 18, 19, 22, 29, 30, 41, and 92.

MIYAO, Hidetoshi; NAKANO, Yasuaki. Head and stem extraction from printed music scores using a neural network approach. In: *Proceedings of the 3rd IEEE International Conference on Document Analysis and Recognition (ICDAR)*. [S.l.: s.n.], 1995. v. 2, p. 1074–1079. Quoted 3 times on pages 47, 49, and 52.

MODAYUR, Bharath R.; RAMESH, Visvanathan; HARALICK, Robert M.; SHAPIRO, Linda G. MUSER: A prototype musical score recognition system using mathematical morphology. *Machine Vision and Applications*, v. 6, n. 2-3, p. 140–150, 1993. ISSN 09328092. Quoted on page 16.

MODAYUR, Bharath R; RAMESH, Visvanathan; HARALICK, Robert M; SHAPIRO, Linda G. Muser: A prototype musical score recognition system using mathematical morphology. *Machine Vision and Applications*, Springer, v. 6, n. 2, p. 140–150, 1993. Quoted 4 times on pages 47, 48, 49, and 52.

MONTAGNER, Igor dos Santos; HIRATA, Roberto; HIRATA, Nina ST. A machine learning based method for staff removal. In: *Proceedings of the 20th IEEE International Conference on Pattern Recognition (ICPR)*. [S.l.: s.n.], 2014. p. 3162–3167. Quoted 4 times on pages 47, 48, 49, and 52.

MONTAGNER, Igor S; HIRATA, Nina ST; HIRATA, Roberto. Staff removal using image operator learning. *Pattern Recognition*, Elsevier, v. 63, p. 310–320, 2017. Quoted 5 times on pages 47, 48, 49, 52, and 54.

MONTAGNER, Igor S; HIRATA, Roberto; HIRATA, Nina ST. Learning to remove staff lines from music score images. In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. [S.l.: s.n.], 2014. p. 2614–2618. Quoted 4 times on pages 47, 48, 49, and 52.

MuseScore BVBA. *MuseScore*. 2018. Disponível em: <https://musescore.org/>. Quoted 3 times on pages 28, 56, and 60.

NA, In Seop; KIM, Soo Hyung. Music symbol recognition by a lag-based combination model. *Multimedia Tools and Applications*, Springer, p. 1–17, 2016. Quoted 5 times on pages 35, 47, 48, 49, and 52.

NEEDLEMAN, Saul B; WUNSCH, Christian D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, Elsevier, v. 48, n. 3, p. 443–453, 1970. Quoted 4 times on pages 20, 68, 70, and 91.

NHAT, Vo Quang; LEE, GueeSang. Adaptive line fitting for staff detection in handwritten music score images. In: ACM. *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*. [S.l.], 2014. p. 99. Quoted on page 49.

NOVOTNỲ, Jiri; POKORNỲ, Jaroslav. Introduction to optical music recognition: Overview and practical challenges. In: *Proceedings of the 15th Annual Workshop of Databases Texts Specifications Objects (DATESO)*. [S.l.: s.n.], 2015. p. 65–76. Quoted 2 times on pages 8 and 35.

OH, Jiyong; SON, Sung Joon; LEE, Sangkuk; KWON, Ji-Won; KWAK, Nojun. Online recognition of handwritten music symbols. *International Journal on Document Analysis and Recognition (IJDAR)*, Springer, v. 20, n. 2, p. 79–89, 2017. Quoted 4 times on pages 34, 47, 49, and 52.

OTSU, Nobuyuki. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 9, n. 1, p. 62–66, 1979. Quoted on page 50.

PADILLA, Victor; MARSDEN, Alan; MCLEAN, Alex; NG, Kia. Improving omr for digital music libraries with multiple recognisers and multiple sources. In: ACM. *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*. [S.l.], 2014. p. 1–8. Quoted 4 times on pages 25, 47, 50, and 53.

PEDERSOLI, Fabrizio; TZANETAKIS, George. Document segmentation and classification into musical scores and text. *International Journal on Document Analysis and Recognition (IJDAR)*, Springer, v. 19, n. 4, p. 289–304, 2016. Quoted 3 times on pages 47, 48, and 49.

PEREIRA, Roberto M Pinheiro; MATOS, Caio EF; JUNIOR, Geraldo Braz; ALMEIDA, João DS de; PAIVA, Anselmo C de. A deep approach for handwritten musical symbols recognition. In: ACM. *Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web*. [S.l.], 2016. p. 191–194. Quoted 3 times on pages 47, 49, and 52.

PIĄTKOWSKA, Weronika; NOWAK, Leszek; PAWŁOWSKI, Marcin; OGORZAŁEK, Maciej. Stafflines pattern detection using the swarm intelligence algorithm. *Computer Vision and Graphics*, Springer, p. 557–564, 2012. Quoted 3 times on pages 47, 49, and 52.

PINTO, Henrique. Iniciação ao violão, vol. 1. *São Paulo: Ed. Ricordi*, 1978. Quoted on page 28.

PINTO, Telmo; REBELO, Ana; GIRALDI, Gilson A; CARDOSO, Jaime S. Music score binarization based on domain knowledge. In: *Proceedings of the Iberian Conference on Pattern recognition and Image Analysis (IbPRIA)*. [S.l.: s.n.], 2011. p. 700–708. Quoted 2 times on pages 49 and 50.

PUGIN, Laurent; BURGOYNE, John Ashley; FUJINAGA, Ichiro. Goal-directed evaluation for the improvement of optical music recognition on early music prints. In: ACM. *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*. [S.l.], 2007. p. 303–304. Quoted 3 times on pages 47, 49, and 54.

PUGIN, Laurent; BURGOYNE, John Ashley; FUJINAGA, Ichiro. Reducing costs for digitising early music with dynamic adaptation. In: SPRINGER. *International Conference on Theory and Practice of Digital Libraries*. [S.l.], 2007. p. 471–474. Quoted on page 49.

PUGIN, Laurent; HOCKMAN, Jason; BURGOYNE, John Ashley; FUJINAGA, Ichiro. Gamera versus aruspix–two optical music recognition approaches. In: CITESEER. *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*. [S.l.], 2008. Quoted 6 times on pages 46, 47, 48, 50, 53, and 54.

RAMIREZ, Carolina; OHYA, Jun. Symbol classification approach for omr of square notation manuscripts. In: *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*. [S.l.: s.n.], 2010. p. 549–554. Quoted 3 times on pages 47, 48, and 49.

RAPHAEL, Christopher; JIN, Rong. Optical music recognition on the international music score library project. In: *DRR*. [S.l.: s.n.], 2014. p. 90210F–90210F. Quoted 4 times on pages 37, 50, 53, and 54.

RAPHAEL, Christopher; WANG, Jingya. New approaches to optical music recognition. In: *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*. [S.l.: s.n.], 2011. p. 305–310. Quoted 2 times on pages 47 and 49.

REBELO, Ana; CAPELA, Artur; COSTA, Joaquim F Pinto Da; GUEDES, Carlos; CARRAPATOSO, Eurico; CARDOSO, Jaime S. A shortest path approach for staff line detection. In: *Proceedings of the 3rd International Conference on automated production of cross media content for multi-channel distribution (AXMEDIS)*. [S.l.: s.n.], 2007. p. 79–85. Quoted 3 times on pages 49, 52, and 54.

REBELO, A.; CAPELA, G.; CARDOSO, Jaime S. Optical recognition of music symbols. *International Journal on Document Analysis and Recognition*, v. 13, n. 1, p. 19–31, 2010. ISSN 14332833. Quoted 7 times on pages 8, 21, 22, 47, 49, 52, and 54.

REBELO, Ana; FUJINAGA, Ichiro; PASZKIEWICZ, Filipe; MARCAL, Andre R. S.; GUEDES, Carlos; CARDOSO, Jaime S. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, v. 1, n. 3, p. 173–190, oct 2012. ISSN 2192-6611. Disponível em: <http://link.springer.com/10.1007/s13735-012-0004-6>. Quoted 5 times on pages 16, 21, 22, 48, and 50.

REBELO, Ana; MARÇAL, André RS; CARDOSO, Jaime S. Global constraints for syntactic consistency in omr: an ongoing approach. In: SPRINGER. *International Conference Image Analysis and Recognition*. [S.l.], 2013. p. 734–741. Quoted 3 times on pages 47, 49, and 52.

REED, K Todd; PARKER, James R. Automatic computer recognition of printed music. In: IEEE. *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*. [S.l.], 1996. v. 3, p. 803–807. Quoted 4 times on pages 47, 50, 53, and 54.

RIBA, Pau; FORNÉS, Alicia; LLADÓS, Josep. Towards the alignment of handwritten music scores. In: SPRINGER. *International Workshop on Graphics Recognition*. [S.l.], 2015. p. 103–116. Quoted on page 45.

ROSSANT, Florence; BLOCH, Isabelle. A fuzzy model for optical recognition of musical scores. *Fuzzy sets and systems*, Elsevier, v. 141, n. 2, p. 165–201, 2004. Quoted 6 times on pages 47, 48, 50, 52, 53, and 54.

ROSSANT, Florence; BLOCH, Isabelle. Optical music recognition based on a fuzzy modeling of symbol classes and music writing rules. In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. [S.l.: s.n.], 2005. v. 2, p. II–538. Quoted 5 times on pages 47, 49, 52, 53, and 54.

ROSSANT, Florence; BLOCH, Isabelle. Robust and adaptive omr system including fuzzy modeling, fusion of musical rules, and possible error detection. *EURASIP Journal on Applied Signal Processing*, Hindawi Publishing Corp., v. 2007, n. 1, p. 160–160, 2007. Quoted 6 times on pages 47, 49, 50, 52, 53, and 54.

SCHMUCKER, Martin. Staff line features as information carrier. In: *Proceedings of the 2nd International Conference on Web Delivering of Music (WEDELMUSIC)*. [S.l.: s.n.], 2002. p. 168–175. Quoted 2 times on pages 45 and 52.

SHARIF, Muhammad; ARSHAD, Quratul-Ain; RAZA, Mudassar; KHAN, Wazir Zada. [comscan]: an optical music recognition system. In: ACM. *Proceedings of the 7th International Conference on Frontiers of Information Technology*. [S.l.], 2009. p. 34. Quoted 3 times on pages 47, 49, and 53.

SU, Bolan; LU, Shijian; PAL, Umapada; TAN, Chew Lim. An effective staff detection and removal technique for musical documents. In: *Proceedings of the 10th IAPR International Workshop on Document Analysis Systems (DAS)*. [S.l.: s.n.], 2012. p. 160–164. Quoted 4 times on pages 48, 49, 52, and 54.

SU, Mu-Chun; TEW, Chee-Yuen; CHEN, Hsin-Hua. Musical symbol recognition using som-based fuzzy systems. In: *Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference*. [S.l.: s.n.], 2001. v. 4, p. 2150–2153. Quoted 3 times on pages 47, 49, and 52.

SZWOCH, Mariusz. A robust detector for distorted music staves. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 3691 LNCS, p. 701–708, 2005. ISSN 03029743. Quoted 3 times on pages 47, 49, and 50.

SZWOCH, Mariusz. Using musicxml to evaluate accuracy of omr systems. *Diagrammatic Representation and Inference*, Springer, p. 419–422, 2008. Quoted 3 times on pages 47, 50, and 58.

TAMBOURATZIS, Tatiana. Identification of key music symbols for optical music recognition and on-screen presentation. In: *Proceedings of the IEEE International Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2011. p. 1935–1942. Quoted 4 times on pages 47, 49, 50, and 52.

TAMBOURATZIS, Tatiana. The digital music stand as a minimal processing custom-made optical music recognition system, part 1: Key music symbol recognition. *International Journal of Intelligent Systems*, Wiley Online Library, v. 28, n. 5, p. 474–504, 2013. Quoted 4 times on pages 47, 49, 50, and 52.

THOMAS, Verena; WAGNER, Christian; CLAUSEN, Michael. Ocr based post processing of omr for the recovery of transposing instruments in complex orchestral scores. In: *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*. [S.l.: s.n.], 2011. p. 411–416. Quoted 3 times on pages 48, 50, and 54.

TIMOFTE, Radu; GOOL, Luc Van. Automatic stave discovery for musical facsimiles. In: *Proceedings of the Asian Conference on Computer Vision*. [S.l.: s.n.], 2012. p. 510–523. Quoted 4 times on pages 48, 49, 50, and 52.

VIERA, Anthony J; GARRETT, Joanne M et al. Understanding interobserver agreement: the kappa statistic. *Fam Med*, v. 37, n. 5, p. 360–363, 2005. Quoted on page 44.

VIGLIENSONI, Gabriel; BURGOYNE, John Ashley; HANKINSON, Andrew; FUJINAGA, Ichiro. Automatic pitch recognition in printed square-note notation. In: *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*. [S.l.: s.n.], 2011. Quoted 4 times on pages 47, 49, 53, and 54.

VISANIY, Muriel; KIEU, Van Cuong; FORNÉS, Alicia; JOURNET, Nicholas. Icdar 2013 music scores competition: Staff removal. In: *Proceedings of the 12th IEEE International Conference on Document Analysis and Recognition (ICDAR)*. [S.l.: s.n.], 2013. p. 1407–1411. Quoted 4 times on pages 49, 50, 52, and 54.

VO, Quang Nhat; KIM, Soo Hyung; YANG, Hyung Jeong; LEE, Gueesang. An mrf model for binarization of music scores with complex background. *Pattern Recognition Letters*, Elsevier, v. 69, p. 88–95, 2016. Quoted 5 times on pages 47, 48, 49, 50, and 54.

WEN, Cuihong; REBELO, Ana; ZHANG, Jing; CARDOSO, Jaime. A new optical music recognition system based on combined neural network. *Pattern Recognition Letters*, Elsevier, v. 58, p. 1–7, 2015. Quoted 6 times on pages 17, 25, 47, 48, 50, and 52.

WIJAYA, K; BAINBRIDGE, D. Staff line restoration. 1999. Quoted 2 times on pages 47 and 49.

WU, Fu-Hai Frank. An evaluation framework of optical music recognition in numbered music notation. In: *Proceedings of the IEEE International Symposium on Multimedia (ISM)*. [S.l.: s.n.], 2016. p. 626–631. Quoted on page 45.

WU, Fu-Hai Frank; JANG, Jyh-Shing Roger. An architecture for optical music recognition of numbered music notation. In: ACM. *Proceedings of International Conference on Internet Multimedia Computing and Service*. [S.l.], 2014. p. 241. Quoted 3 times on pages 25, 45, and 50.

WU, Fu-Hai Frank; JANG, Jyh-Shing Roger. On the way to ambient media for sheet music by techniques of information retrieval. In: *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. [S.l.: s.n.], 2014. p. 1–6. Quoted 3 times on pages 25, 45, and 50.

YADID-PECHT, Orly; GERNER, Moty; DVIR, Lior; BRUTMAN, Eliyahu; SHIMONY, Uri. Recognition of handwritten musical notes by a modified neocognitron. *Machine vision and applications*, Springer, v. 9, n. 2, p. 65–72, 1996. Quoted 3 times on pages 25, 47, and 49.

YOO, JaeMyeong; KIM, GiHong; LEE, Gueesang. Mask matching for low resolution musical note recognition. In: *Proceedings of the IEEE International Symposium on Signal Processing and Informaiton Technology*. [S.l.: s.n.], 2008. p. 223–226. Quoted 5 times on pages 17, 25, 47, 50, and 54.