Title

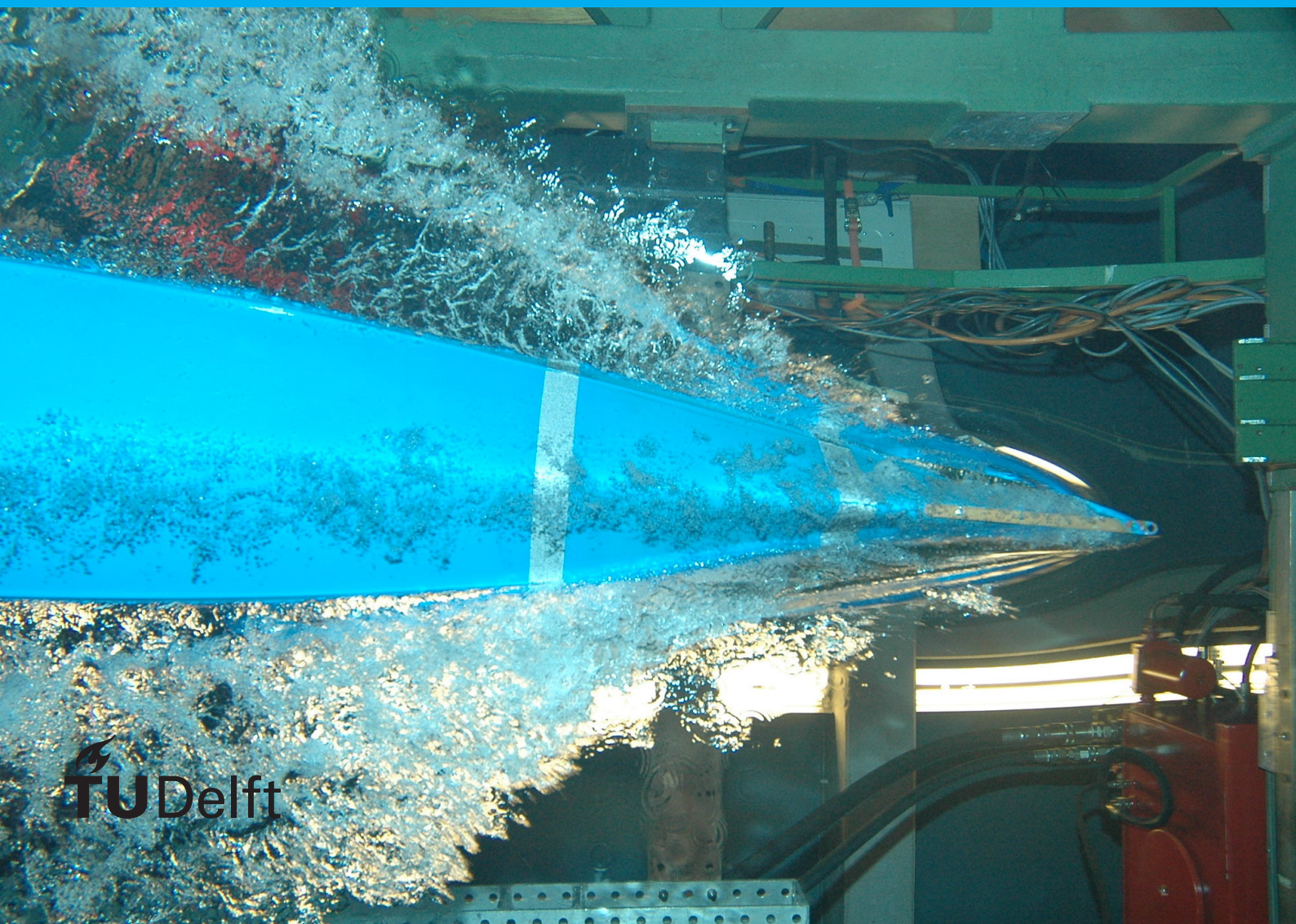
;

Optional subtitle

M. Meuleman

Cover Text
possibly
spanning multiple lines

ISBN 000-00-0000-000-0



This page intentionally left blank

Title

Optional subtitle

by

M. Meuleman

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday January 1, 2013 at 10:00 AM.

Student number:	4375629	
Project duration:	March 1, 2012 – January 1, 2013	
Thesis committee:	Prof. dr. ir. J. Doe,	TU Delft, supervisor
	Dr. E. L. Brown,	TU Delft
	Ir. A. Aaronson,	Acme Corporation

TU Delft

This page intentionally left blank

Abstract

This page intentionally left blank

Preface

Preface...

M. Meuleman
Delft, January 2013

This page intentionally left blank

Contents

This page intentionally left blank

1 Introduction

Rewrite intro

Music has always been communicated in two ways: aural transmission, where music is played or performed and people can listen to it, and written transmission, where music is formalized in a document. These written formats come in many forms, one of which is western musical notation. These documents, mostly referred to as (music) scores, used to be handwritten, but were later printed on paper and can nowadays be found in digital format on computers, mostly as images or scans. Quite some effort has been taken to collect all of these digital scores and store them in central places or make them available to the public where possible. An example is the IMSLP library, where vast amounts of scores for classical music in the public domain are available. Collections such as these give rise to many opportunities, such as affordable scores and sheet music for musicians, or easy access to musical data for researchers. The formats in which these scores are stored do have their limitations however. When text is contained in PDF documents (or many other document forms for that matter), the text is recognized by the computer as text, meaning it can be searched, edited and reformatted. For music this is not the case. Scanned music scores are simply stored as images in PDF documents, meaning that none of the applications for text can be applied to these scores. This means that a lot of applications that the digital age provides cannot yet be applied to music scores and as such there is a huge gap for potential left open.

1.1. Music notation

Music notation is the manner in which music is communicated in written form. The de facto standard for this since the seventeeth century has been Common Western Music Notation (CWMN), of which the general principles are shared by almost all music that is written since then. CWMN consists of one or more music parts, which are depicted from left to right, synchronized in time. The music is divided up into measures, which provide logical divisions and keep track of the current position in the music while playing. These measures are seperated by barlines. Since measures are used to track the position of music in time, the different parts all have the same amount of measures, synchronized in time. To make sure these parts fit on pages, they are broken up at barlines and placed under each other on the page, or on the next page if there is no more space on the page left. This pattern resumes till the end of the piece. Each such section that spans a page's width is called a system, which consists of a set a staffs, generally one for each part, all divided into synchronized measures. We call the collection of measures that overlap at a given point in time a system measure, to differentiate them from the individual measures. It can be customary to leave out staffs of a system if that part has nothing to play for the duration of that system, causing difference in the number of staffs between systems. In Figure 4.1 examples of all these elements of CWMN are shown. In first of the two systems is highlighted in yellow, the second staff of the first system is highlighted in red, the first system measure is highlighted in green and the first measure of the third part is highlighted in blue. When counting, it also becomes clear that the first system has one staff more than the second system, to save space on the page.

When going to a smaller level, we touch upon the contents of the individual measures. The vertical placement of notes indicate their pitch, together with the clefs and accidentals that might occur on the page or in the measure. Their horizontal relation to each other, as well as the type of note, flags, and rests, indicate the rhythm in which the notes need to be played. Those at least are the very basic principles, ignoring a vast amount of details. One special case that is worth noting is the empty measure, which consists entirely of a

single, full-measure rest without any notes. This indicates that in this measure the instrument has no role to play, examples can be seen in figure 4.1, e.g. the last four measures of the top three staffs.

1.2. Introduction into Optical Music Recognition

By digitizing the contents of a music score, the information contained in it can be processed by a computer. This opens up a large amount of possibilities for various target audiences. First of all the field of musicology can benefit greatly from this, since digital scores are often not availble. Cultural heritage can be safeguarded through digital scores, since large amounts of historical significant collections of music are currently only available on paper. Musicians can be equiped with a "digital music stand", where the music score can be easily adapted to account for readability issues, or error corrections and annotations can be easily made and shared with other musicians. Furthermore there is the obvious advantage of compiling large databases of digital information, creating possibilities for search and browsing, which are widely available for textual data, and further applications such as melody retrieval and similarity studies, just to name a few.

The digitizing process, or transcribing, is still relatively often done manually by experts. Several music notation applications can be found nowadays, which make their workflow more efficient. Examples of these applications are Sibelius [1], MuseScore [2] and Finale [3], in which users can input music either by keyboard and mouse, or by using a MIDI keyboard. The large drawback of these systems is that inputting music is time-consuming and takes experience, and is therefore expensive. Automatic transcription of music scores could therefore dramatically improve this process, and this is what the field of Optical Music Recognition, or OMR, is researching.

Automating transcription is unfortunately not an easy task. There are a lot of parallels to be drawn with the field of Optical Character Recognition, which focusses on transcribing written text. However, where most OCR tasks are considered to be solved, this does not hold for OMR. There are several reasons for this. First of all, where OCR is generally a one dimensional problem where text is vertically evenly spaced out, OMR is a two dimensional problem, where time and frequency make up the horizontal and vertical axis of sheet music, respectively [6]. This can result in influence between symbols that are spaced out far across the page, such as a clef at the start of a page, which indicated what frequency the written notes entail, or accidentals that span a measure or even an entire page's width. Additionally, in music different symbols can be stacked on top of one another to signify multi-tones or chords, or can be connected horizontally to one another. Stacked or connected notes can also be translated both horizontally and vertically relative to each other, meaning the connection can become stretched or tilted, see [3] for some examples. Besides, these translations and connections can often be done in different ways, or they can be purposefully left out. Decisions on if and how these tranformations are used often depend on preferences of the musician, or on larger contextual requirements, such as horizontal spacing to accommodate space for measures that are played at the same time. Furthermore there are symbols that are relatively small in comparison to the notes, but have significant meaning, such as dots, dashes and accents. There is also a large imbalance in the frequency in which symbols occur, increasing the likelihood of false positives when classifying these symbols [11]. All of these difficulties make it impossible for simple systems to accurately transcribe music scores.

The OMR field has been active for over 50 years now, starting with Pruslin in 1966 [22]. Since then, a number of different approaches have been tried, which ultimately led to a generally agreed upon pipeline in which the OMR task could be subdivided into smaller subtasks. This pipeline was first coined by Bainbridge and Bell [3] and was later elaborated on by Rebelo et al. [24] in a vast literature review. Both papers indicate that the vast majority of research up to that point could be segmented into the various stages of this pipeline, and in fact a lot of research from that point on falls into place when considering this framework. There have been deviations though, since not all stages are equally easily solvable. With the increasing research and application of deep learning, attempts at full-pipeline and end-to-end OMR have been started to occur as well [8, 21]. The preliminary results show promise, but the research effort is scoped very tightly, making it so that a lot of work needs to be done before these approaches can be considered generally applicable. This is also hampered by the main struggle of deep learning; the lack of sufficient data. Some datasets have been compiled, mainly for object classifications, but it is hard to conceive that these will be sufficient for reliable classification given the large amount of variability that these objects can occur in due to differences in handwriting, typesetting and deformations in printing and scanning.

[1] https://www.avid.com/sibelius
[2] https://musescore.org/
[3] https://www.finalemusic.com/

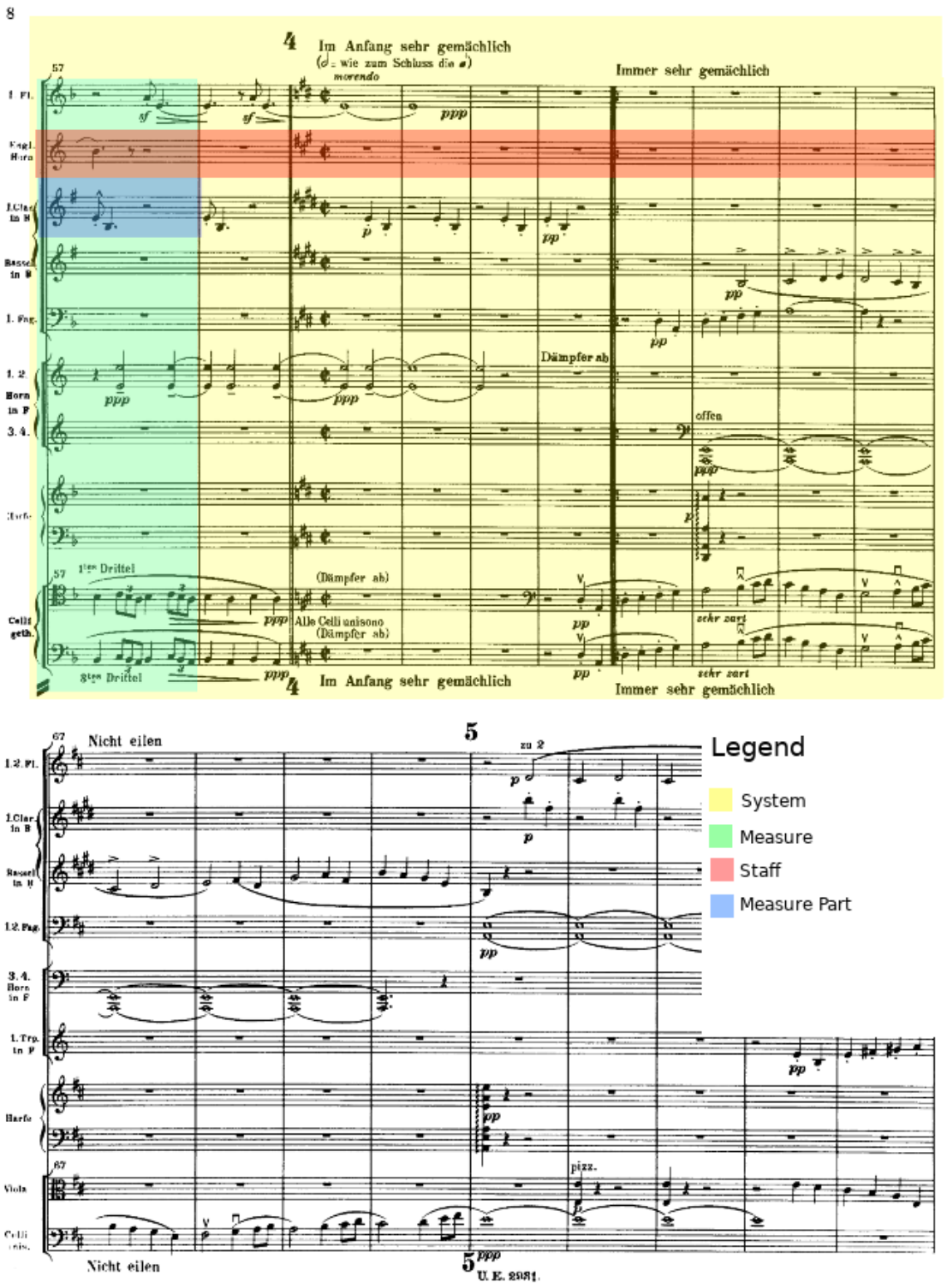


Figure 1.1: Example structure of a page of music

Trying to overcome all of these problems has led to the rise of human-aided OMR, where the focus lies on human feedback in several stages of the pipeline, in the hope that this will improve the performance of such an OMR system in comparison to fully automated versions. Examples of such research are MacMillan et al. [17], who have developed Gamera, an ease-to-use scripting environment for OMR, Chen et al. [10, 11], who have been developing an adapted OMR system called Ceres that takes human feedback into account, and Burghardt and Spanner [5], who developed Allegro which takes a more user-centered approach and provides an easy framework for users to transcribe the contents of simple handwritten scores.

1.3. TROMPA

Towards Richer Online Music Public-domain Archives, or TROMPA, is an international organization of scientists, scholars and musicians that push towards the goal of making more applications available to the music domain and increase engagement with classical music and music scores [2]. One of their main research themes is called "Scanned score analysis" and focusses on improving OMR and bringing together different techniques to make large scale transcription of music scores feasible, so that those transcriptions can be used in a variety of other research themes. Within this research theme, work is being done on human-aided OMR through crowd computing, in which a large userbase is asked to interact with the results of an OMR system and through that correct and check the results of the system.

Await TROMPA deliverable

Viewing these results, one of the issues that arises when dealing with large music works like this is the repetitiveness of correcting and checking measures of the music score. Especially in large ensemble works, it occurs frequently that several instruments or instrument groups don't play continuously, but instead have parts of the piece where there is a gap in their score. Since these instrument groups still need to be aligned with the rest of the ensemble in the music score, these parts are indicated by empty measures. Over the different instrument groups, the total of these empty measures can add up quickly, which raises the question whether this characteristic can be utilized to improve the workflow of this human-aided OMR system, and perhaps reduce the work that has to be performed by users.

1.4. Problem statement

This leads to the main goal of this thesis. We want to investigate whether the work needed for this OMR task can be made easier by means of finding similarities in the music scores on the measure level. If it is indeed possible to find such similarities, and we can separate these similar measures with reasonable accuracy from the remainder of the score, data that is collected on these measures, be it through automated transcription or human input, could be shared across similar measures. This also means that the benefits of such a system would not be limited to the TROMPA project, but could be applied in a wider OMR context. These similarities will be studied on PDF documents that contain the scores, since those documents are the main targets for (automatic) transcription in the TROMPA project. The scope of music is defined as follows:

- We focus on ensemble music, that is, music written for ensembles of musicians. These ensembles can range from small chamber music ensembles, which usually consist of two to eight musicians, to full-sized symphony orchestras. This scope is a logical one given the goal this thesis is persuing, since similarities in music, such as empty measures, are more likely to occur in larger ensembles. Music written for individual permormers will most likely contain very little empty measures.
- We focus on music scores that are available in typeset fonts, since the vast majoriy of these ensemble works can be found as typeset in public domain collections such as the Petrucci Music Library (IMSLP [1]).

To be able to find these similarities, we first need to be able to accurately find the measures that are found in a given score. Therefore a preliminary step needs to be taken which can segment the music score in PDF format into its individual measures. This is the first part of our research goal, after which we can focus on the similarity study.

1.5. Research questions

Now that we have defined the problem, we define the following main research question:

RQ: How can we find similarities in music scores on the measure level?

To structure this further, we provide the following subquestions:

SQ1: What previous work has been done on measure detecting and similarity finding and how can we build upon this?

SQ2: What data is available which can be used to design and evaluate these systems?

SQ3: How can we reliably detect the measures in a music score?

SQ4: How can we find similarities between detected measures?

1.6. Main contributions

1.7. Outline

This page intentionally left blank

2

Related work

2.1. Existing datasets

When working with written music score data, there are a few datasets already available in the OMR field. Examples are the CVC-MUSCIMA [13] and its derivative, the MUSCIMA++ [15] for handwritten data, mainly aimed at staff line removal and symbol classification, and the MeasureBoundingBoxAnnotations dataset [29]. Both the MUSCIMA++ and version 2 of the MeasureBoundingBoxAnnotations datasets have annotations for bounding boxes of individual measures.

2.2. Measure detectors

The segmenter currently embedded in the OMR pipeline is the one taken from the work of Waloschek, Hadjakos and Pacha [1]. Their approach consisted of manually annotating measures on pages of orchestral scores, defining a distance metric between annotated measures and training a CNN to detect measures in new input data. There are a few shortcomings to this approach. First of all, this model is far from perfect and makes too many mistakes to be used in a reasonable manner in this OMR pipeline. Even on pages that contain high quality scans and straight barlines, the detection is not perfect and therefore requires post-processing, see Figure 1 for two examples. Second, the measures that are detected are restricted to separation over time only. This means that when applying this to music scores with multiple instruments or voices -which is common in orchestral music, choir music, or piano scores- the segmenter will only segment horizontally, but leaves the different voicings grouped together in the same block. Besides this, the model is fixed and cannot be easily improved upon. Retraining the model to overcome the mistakes it currently makes would require manual annotation of these pages which is a very costly process. Finally, this model is relatively slow compared to other approaches. Insert mentions of [29]

Figure 1: Two examples of errors of the CNN method. In the first example we see that a large part of the entire page is classified as a single measure, in the second examples we see that smaller subsections of measures are detected as measures.

2.3. Dynamic Time Warping

2.4. Similarity finding

2.5. Paper shorts

2.5.1. OMR

- Bellini 2007 [4] Overview of state-of-the-art, with a call-to-action for more research. Attempts to provide more standard metrics for evaulation of OMR systems.
- Bainbridge 2001 [3] Describes properties of music that make OMR a difficult task, review of OMR research pre-2000. Overview of sub-tasks for OMR pipeline (check if [24] corresponds with this), with common solutions.
- Rebelo 2012[24] A more detailed outline of the different subtasks that are involved in OMR, with a large set of available research linked to those subtasks. []

- Byrd 2015 [6] Set of definitions for complexity of music notation, set of performance metrics for page quality and score complexity, provides a small corpus of music as a baseline for a proper OMR testbed.
- Raphael 2011 [23] Presents a first version of an OMR system that 1) first segments a page into systems, system measures and individual measures, and 2) uses the measures as a basic unit for recognition. Recognition is divided into rigid symbols (clefs, rests, etc) and composite symbols (notes with stems, chords, beamed groups, etc). See also [10]
- Calvo-Zaragoza 2019 [9] A paper that aims to be a primer in OMR. The authors argue that the OMR community is hurting from a lack of framework in which to work, missing clear task definitions and evaluation methods. It provides a taxonomy of OMR, and a taxonomy of OMR applications.

2.5.2. Difficulties of OMR

- Bainbridge 2001 [3]
- Bellini 2007 [4]
- Homenda 2006 [16]
- Fujinaga 1996 [14]
- Byrd 2015 [6]

2.5.3. OMR pipeline

- Bainbridge 2001 [3]
- Rebelo 2012 [24]

2.5.4. Full-pipeline or end-to-end OMR

- Calvo-Zaragoza 2017 [8] First attempt at end-to-end OMR. Using a convolutional recurrent nn with synthetically obtained monophonic scores with limited length and symbols. Classifies concurrent symbols, but lacks polyphonic and cross-symbol spanning elements, such as slurs, ties, text, etc.
- Calvo-Zaragoza 2018 [7] Builds upon previous paper, but now with a newly created dataset: PRiMuS. Results are quite good, but still limited to monophonic music, and there are some dubious claims in the conclusion: In previous paper the review of Rebelo et al. [24] showed "results have not been very promising so far", however the task in this paper "can be solved succesfully using the considered ... approach". Also, claiming "performance comparable to that of commercial systems" after a "qualitative comparative study of selected examples" seems quite a stretch.
- Wel 2017 [27] Convolutional and recurrent nn that translates a line of sheet music to (pitch, duration) pairs. All data is monophonic, and the resulting format discards all markup, such as beamed note groups, stem direction etc.

2.5.5. Human-aided OMR

- Petros 2020 [25] Designing micro-tasks in such a way that crowd sourcing can be used on a highly varied crowd. User interaction is through Amazon Mechanical Turk, tasks are 1 to 3 measures long, and differ between providing the score, the audio, or both. Users were asked to compare two versions of the and indicate whether they are the same. [4, 5, 9, 10]
- Burghardt 2017 [5] The Allegro system is a framework in which users can manually transcribe music scores. It was used to transcribe a collection of handwritten folk music that contains single melodic lines only. Includes references to crowd sourcing and OCR correction papers. [3, 4, 23, 24]
- Chen 2016 [10] Human directed OMR takes an existing recognition engine and brings human input into the loop. In three stages, staff recognition, system identification and measure contents recognition, users are employed to correct errors made by the system by drawing fundamentels such as stems and notes, and labelling them manually.

- Chen 2016(b) [11] Human interactive OMR system called Ceres that also employs human input, but works on a symbol-level instead of a measure-level. Users can add contraints to the system by identifying a pixel as a classified object, and the system re-recognizes the contents of a page within these constraints.

2.5.6. OMR techniques

- Otsu 1979 [20] binarization technique
- Bainbridge 2001 [3] Staff line detection through horizontal projections, possibly only on start and end of staff line to avoid noise through music symbols.
- Staff line height and staff space height determined through run-length encoding [24]
- Dalitz 2008 [12] Comparing several staff removal algorithms, which are implemented in the Gamera toolkit.
- Vigliensoni 2013 [28] Optical measure recognition technique described. Staff detection is taken from [] and barline detection is developed by the authors.

2.5.7. Clustering

- Senin 2008 [26] Dynamic Time Warping overview, basic algorithm, customizations and optimizations with some examples.
- Niennattrakul 2006 [18] Clustering multimedia data with time series. Comparing Euclidean and Dynamic Time Warping distances in a k-mediods clustering algorith.
- Niennattrakul 2007 [19] Same topic as previous paper, but now showing why k-means is inferior to k-mediods clustering when using DTW measure as distance metric.

Computational musicology (music pattern recognition)
OCR clustering?
Alessio Bazzica

2.5.8. Crowd sourcing

[5, 25]

This page intentionally left blank

3

Data description

For this work, one of the focus points is music scores for larger ensembles. In this section we will describe the requirements for this data and the way the data is collected.

3.1. Requirements

There are a few requirements we want to hold the data to. Mainly we want to focus on larger ensemble music scores. These can range from chamber music ensembles for five instruments to full sized symphnoy orchestras. Most of the music scores suited for ensembles of these sizes are typeset instead of handwritten, since these scores are generally made available by publishers. Although we limit ourselves to typeset sources, we want to make sure to include different typesets and fonts, which correspond with the types of music scores often worked with by ensembles. Furthermore we aim to include pieces from different time periods, since the composition of ensembles has been subject to change over the last centuries.

3.2. Aquisition

Aquisition was done through IMSLP. Various music scores were chosen based on the criteria described above. Scores range from small ensembles to symphony orchestras, even including an orchestra with choir piece. We have included pieces from the classical and romantic eras. We also selected a piece that is written for orchestra and choir, to test if the approaches taken in this work can also extend to those applications. The selected pieces are shown in Table 3.1. In this table we see the titles and composers of the pieces, as well as some characteristics of the pieces, such as ensemble composition, image quality and tightness (both from [6]).

From all of the selected pieces, the highest quality score from IMSLP was selected. The pdf was split into single page PNG images with 300 dpi. Since we want to work with the musical contents only, the auxiliary pages that contain no music, such as covers, table of contents, additional explanations etcetera where discarded. The remaining images where binarized and saved as PNG once more.

From all these pages the bounding boxes of the staves, the measures and the individual measures were found. For this task, first all systems, staffs and measures were manually counted for each page. The measure detector layed out in Chapter 4 was then run for each all the pages, and the amount of systems, staffs and measures was compared to the previously found amount. When these counts are equal, the correct positions of staffs and measures should be selected, assuming that the measure detector finds staffs and measures before it finds anything else. To verify this assumption, the found staffs and measures were overlayed on the pages and the result was manually checked. Pages which had missing or extraneous staffs or measures, or pages where the measure detector found objects besides staffs and measures first were manually corrected.

The resulting dataset, containing the original PDFs, the binarized PNG images and the bounding box annotations in JSON format were made available through Insert Github link?.o

Title	Composer	Ensemble composition	#Pages	Image quality	Tigh
Septett, opus 20	Beethoven, L. van	Cl, Bs, Ho, Vl, Vo, Cl, Db	40		
La Mer	Debussy, C.	3324-4331 + 2 Crn., strings, percussion, harp	137		
l'Apprenti Sorcier	Dukas, P.	3234-4231 + 2 Crn., strings, percussion, harp	74		
Symphony No. 104	Haydn, J.	2222-2200, strings, timpani	62		
Psalm 42	Mendelssohn, F.	2222-2200, strings, timpani, choir, soprano	67		
Symphony No. 31	Mozart, W.A.	2222-2200, strings, timpani	40		
Symphony No. 4	Schubert, F.	2222-4200, strings, timpani	60		

Table 3.1: List of selected music scores.

4

Measure detector

In this chapter we will lay out the work that was done for the measure detector. First we will cover the general structure of music scores and the assumptions made that follow from that structure. After that we will cover the implementation of the measure detector, followed by the evaluation of results with both existing datasets and the data collected as described in Chapter 3.

4.1. Music scores

To understand the steps that need to be taken when implementing a measure detector, we first describe the general structure of a music score in this section. Each page of a music score contains one or more sytems. A system can be defined as an excerpt of a full page, having all the voices or instruments synchronized in time. A system will span the total width of the page, minus page margins, and subsequent systems will be put one beneath the other, continuing on subsequent pages depending on space. Each system can be seen as a grid of voices and measures. The rows of this grid represent the voices, each voice has a single staff, and the columns represent the measures. Each cell in this grid we then define as an individual measure. Terminology on this can get a little ambiguous, since in general music practise the term measure is used interchangably for both columns and cells in this grid, since generally musicians will mostly work with individual parts instead of the whole score, which unifies both these concepts since the columns in these individual parts are only of size 1. For clarity from now on we will refer to the columns as system measures and to the cells as measures, since this is done in the literature as well (e.g. [29]). An example of such a structure is given in Figure 4.1, where a system, staff, measure and individual measure are outlined on a page of music. From this example it is already clear that there is more content on a page of music than just this grid, but for our purposes, the definition as given above should suffice.

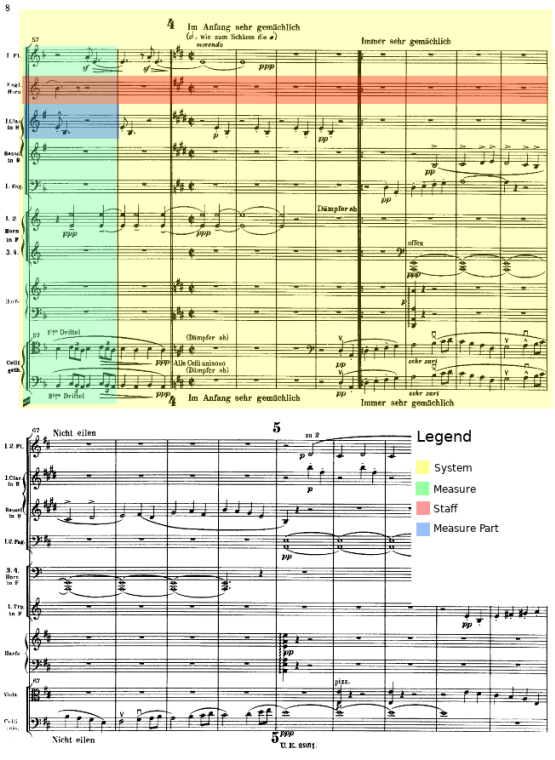


Figure 4.1: Example structure of a page of music

Note that although it frequently occurs that different instruments or voices share a staff, this only makes a semantic difference, not a syntactical one. The purpose of this measure detector is to find the syntax of the music only, having a single voice versus multiple voices in a single staff can only change a staff rom containing

monophonic to polyphonic contents, but this difference does not influence the measure detector, since it is expected to be able to handle polyphonic contents anyways, since polyphonic instruments also frequently occur in music scores.

4.2. Detecting the grid

This general structure of a page as layed out above will be used to detect the positions of the rows and columns of this grid on a page of music. Before any segmentation is done, some preprocessing is performed on the page. First we use morphology operations to find profiles of the vertical and horizontal line segments on the page. Using the horizontal profile, any rotation in the page that might have occurred due to scanning of the original score is rectified. Next the rotated page is inverted, making it white on black, thresholded by using Otso thresholding [citation needed:] and then Gaussian blur is applied to smooth out some noise. The detection steps will be performed on the resulting image, complemented by the obtained profiles for vertical and horizontal line segments.

The first step of detection is done at the system level. There is a high level of separation between systems on a page, there are no connected components between them, which allows for segmentation through binary propagation. This binary propagation step will fill in each of the systems, allowing for easy detection of a few large blocks on the page, each of which is a candidate to be a system. A candidate system is only considered to be a system if there are horizontal and vertical lines detected in that system, this is to filter out extraneous parts on the page such as titles and text. From here on, in each system the vertical and horizontal profiles are used to detect the columns and rows in the system. First the staffs are detected by using the SciPy peak detection algorithm [citation needed:] on the cutout of the horizontal profile containing only the current system in question. The detected peaks are grouped based on individual distance; if a next peak is further away twice the mean distance between peaks, a new grouping is created. These groupings are considered to be staffs in the system. This method is prefered over simply grouping five concurrent peaks together, which could also be an option considering common western music notation generally uses staffs consisting of five lines. However, this does not always hold true, percussion instruments for example can be notated with single lines only, and noise on a page might cause the peak detection algorithm to miss out on a line, making the results incorrect as well. With the staffs detected, the measures are now found using a similar method. Peak detection is performed on the cutout of the vertical profile, again containing only the current system, but removing the information contained within the boundaries of the previously detected staffs. Since we are detecting vertical lines, there is the possibility that note staffs are detected as well. Especially when a large part of the ensemble plays a note at the same time, this can yield a false positive while trying to find the measure boundaries. Therefore the information contained within the staffs, where most of the note staffs can be found, is removed. Since the barlines span the entire hight of the system, these can still be detected, even though some of that information is also removed. Now that the rows and columns of the grid are detected, the individual measures are simply found at the intersection. Note however that horizontally there is empty space between the boundaries of staffs. This is considered disputed teritory. Notes, accidentals and other markings of both the above and below staffs are allowed to spill over in this space, therefore there is no straight forward way to determine which part of this space belongs to which staff, that division is in a lot of cases non-linear. How this is handled can differ between applications of this detector, therefore no fixed solution is given here, however two imperfect possibilities are given here. Both of them suffer from the fact that they try a linear division, which is not always a possibility.

4.2.1. Smallest intersection method

The smallest intersection method works in two parts: first a set of baselines per system are established, and then these baselines are corrected for each individual block in that system. The baselines are established from the vertical intensity profile of the system. Each of the bars consist of 5 small peaks, corresponding to the 5 lines in a bar. These 5 peaks grouped together can be detected as one broader peak. The baselines are set as the middle points between each of these detected peaks. Next the baselines are corrected on a per block basis. This second step is necessary, since in the scores it can occur that notes and related annotations can cross an established baseline into the “territory” of measures above or below it. This crossing over can change per block, and therefore a correction per block is necessary. This second step finds within a predefined distance from the baseline the points with the lowest value in the intensity profile. These points indicate that when segmenting at these points, the least amount of information, indicated by white pixels, will be segmented, and therefore these points should be considered as good segmenting points. When finding these candidate

points a small margin from the minimum intensity value is taken, and the point closest to the baseline is chosen as the segmenting point.

4.2.2. Largest region method

The largest region method divides the region in between two peaks into regions, where regions are separated from each other by intensity values above a certain threshold. The largest of these regions is taken, as this indicated the largest part between two measures where there is little to no information. The middle of this region is chosen as the segmenting point. In Figure 3 two examples are given of segmented pages using the image processing approach with the largest region method.

Figure 3: Two examples of score pages segmented with the largest region method. Evaluation The drawback of this image processing method is that performance of the segmenter is hard to evaluate. The CNN based method had in this respect the advantage of having annotated examples. Unfortunately, these examples are not applicable to evaluation of the image processing approach, since that segments a level deeper; where the CNN approach segments at the block level, the image processing approach segments at the measure level. Currently evaluation has to be done by hand because of the lack of a corpus of segmented music scores. A tool is under development that can hopefully make this process more efficient and over time can hopefully help to create a corpus of segmented measures.

This page intentionally left blank

Bibliography

[1] Imslp website - main page. https://imslp.org/wiki/Main_Page. Accessed: 2021-05-03.

[2] Trompa website - about. https://trompamusic.eu/about-trompa. Accessed: 2021-05-03.

[3] David Bainbridge and Tim Bell. The challenge of optical music recognition. Computers and the Humanities, 35:95–121, 2001. ISSN 00104817. doi: 10.1023/A:1002485918032. URL https://link.springer.com/article/10.1023/A:1002485918032. Outlines the history of OMR research from 1960 - 2000, contains useful examples of why OMR is much more difficult than OCR.
.

[4] Pierfrancesco Bellini, Ivan Bruno, and Paolo Nesi. Assessing optical music recognition tools, 2007. URL https://www.jstor.org/stable/4618021?seq=1&cid=pdf-.

[5] Manuel Burghardt and Sebastian Spanner. Allegro: User-centered design of a tool for the crowdsourced transcription of handwritten music scores. pages 15–20, 2017.

[6] Donald Byrd and Jakob Grue Simonsen. Towards a standard testbed for optical music recognition: Definitions, metrics, and page images. Journal of New Music Research, 44:169–195, 2015. ISSN 1744-5027. doi: 10.1080/09298215.2015.1045424. URL https://www.tandfonline.com/action/journalInformation?journalCode=nnmr20.

[7] Jorge Calvo-Zaragoza and David Rizo. End-to-end neural optical music recognition of monophonic scores. Applied Sciences, 8, 2018. doi: 10.3390/app8040606. URL https://musescore.org.

[8] Jorge Calvo-Zaragoza, Jose J Valero-Mas, and Antonio Pertusa. End-to-end optical music recognition using neural networks. pages 23–27, 2017. URL http://lilypond.org/.

[9] Jorge Calvo-Zaragoza, Jan Hajič, and Alexander Pacha. Understanding optical music recognition. ACM Computing Surveys, 53:1–35, 8 2019. doi: 10.1145/3397499. URL http://arxiv.org/abs/1908.03608http://dx.doi.org/10.1145/3397499.

[10] Liang Chen and Christopher Raphael. Human-directed optical music recognition. Electronic Imaging, 2016:1–9, 2016.

[11] Liang Chen, Erik Stolterman, and Christopher Raphael. Human-interactive optical music recognition. pages 647–653, 8 2016. doi: 10.5281/ZENODO.1416184. URL https://zenodo.org/record/1416184.

[12] Christoph Dalitz, Michael Droettboom, Bastian Pranzas, and Ichiro Fujinaga. A comparative study of staff removal algorithms. IEEE transactions on pattern analysis and machine intelligence, 30:753–766, 2008.

[13] Alicia Fornés, Anjan Dutta, Albert Gordo, and Josep Lladós. Cvc-muscima: a ground truth of handwritten music score images for writer identification and staff removal. International Journal on Document Analysis and Recognition (IJDAR), 15:243–251, 2012. doi: 10.1007/s10032-011-0168-2. URL http://www.cvc.uab.es/cvcmuscima.

[14] Ichiro Fujinaga. Adaptive optical music recognition, 6 1996. URL moz-extension://b1bc1f1d-22b8-4225-8d70-0b4c95e8cc68/enhanced-reader.html?openApp&pdf=http%3A%2F%2Fwww.music.mcgill.ca%2F~ich%2Fresearch%2Fdiss%2FFujinagaDiss.pdf.

[15] Jan Hajič and Pavel Pecina. The muscima++ dataset forhandwritten optical music recognition. 14th IAPR International Conference on Document Analysis and Recognition, 2017. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8269947.

[16] Wladyslaw Homenda and Marcin Luckner. Automatic knowledge acquisition: Recognizing music notation with methods of centroids and classifications trees. pages 3382–3388, 2006. ISBN 0780394909. doi: 10.1109/ijcnn.2006.247339.

[17] Karl Macmillan, Michael Droettboom, and Ichiro Fujinaga. Gamera: Optical music recognition in a new shell. 2002. URL https://www.researchgate.net/publication/255637263.

[18] Vit Niennattrakul and Chotirat Ann Ratanamahatana. Clustering multimedia data using time series. pages 372–379, 2006. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4021117.

[19] Vit Niennattrakul and Chotirat Ann Ratanamahatana. On clustering multimedia time series data using k-means and dynamic time warping. pages 733–738, 2007. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4197360.

[20] Nobuyuki Otsu. A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, 9:62–66, 1979.

[21] Alexander Pacha, Jorge Calvo-Zaragoza, and Jan Hajič jr. Learning notation graph construction for full-pipeline optical music recognition. pages 75–82, 11 2019. doi: 10.5281/ZENODO.3527744. URL https://zenodo.org/record/3527744.

[22] D Pruslin. Automatic recognition of sheet music, 1966. a critical survey of music image analysis. Structured document image analysis. Springer, Heidelberg, pages 405–434, 1992.

[23] Christopher Raphael and Jingya Wang. New approaches to optical music recognition. pages 305–310, 10 2011. doi: 10.5281/ZENODO.1414856. URL https://zenodo.org/record/1414856. Work in progress research on detecting and classifying notes and chords in IMSLP data, with an example of Beethoven sonata for violin and orchesrr.

[24] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre R.S. Marcal, Carlos Guedes, and Jaime S. Cardoso. Optical music recognition: state-of-the-art and open issues. International Journal of Multimedia Information Retrieval, 1:173–190, 10 2012. ISSN 2192662X. doi: 10.1007/s13735-012-0004-6. URL http://www.finalemusic.com/.

[25] Ioannis Petros Samiotis, Sihang Qiu, Andrea Mauri, Cynthia C S Liem, Christoph Lofi, and Alessandro Bozzon. Microtask crowdsourcing for music score transcriptions: An experiment with error detection. 2020.

[26] Pavel Senin. Dynamic time warping algorithm review. Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA, 855:1–23, 2008.

[27] Eelco van der Wel and Karen Ullrich. Optical music recognition with convolutional sequence-to-sequence models. 10 2017. doi: 10.5281/ZENODO.1415664. URL https://zenodo.org/record/1415664.

[28] Gabriel Vigliensoni, Gregory Burlet, and Ichiro Fujinaga. Optical measure recognition in common music notation. pages 125–130, 2013. URL https://github.com/DDMAL/.

[29] Frank Zalkow, Angel Villar Corrales, T J Tsai, Vlora Arifi-Müller, and Meinard Müller. Tools for semi-automatic bounding box annotation of musical measures in sheet music. 2019. URL https://www.audiolabs-erlangen.de/resources/MIR/2019-ISMIR-LBD-Measures.

[30] Barbara Zitová and Jan Flusser. Image registration methods: A survey. Image and Vision Computing, 21: 977–1000, 2003. ISSN 02628856. doi: 10.1016/S0262-8856(03)00137-9.