Title

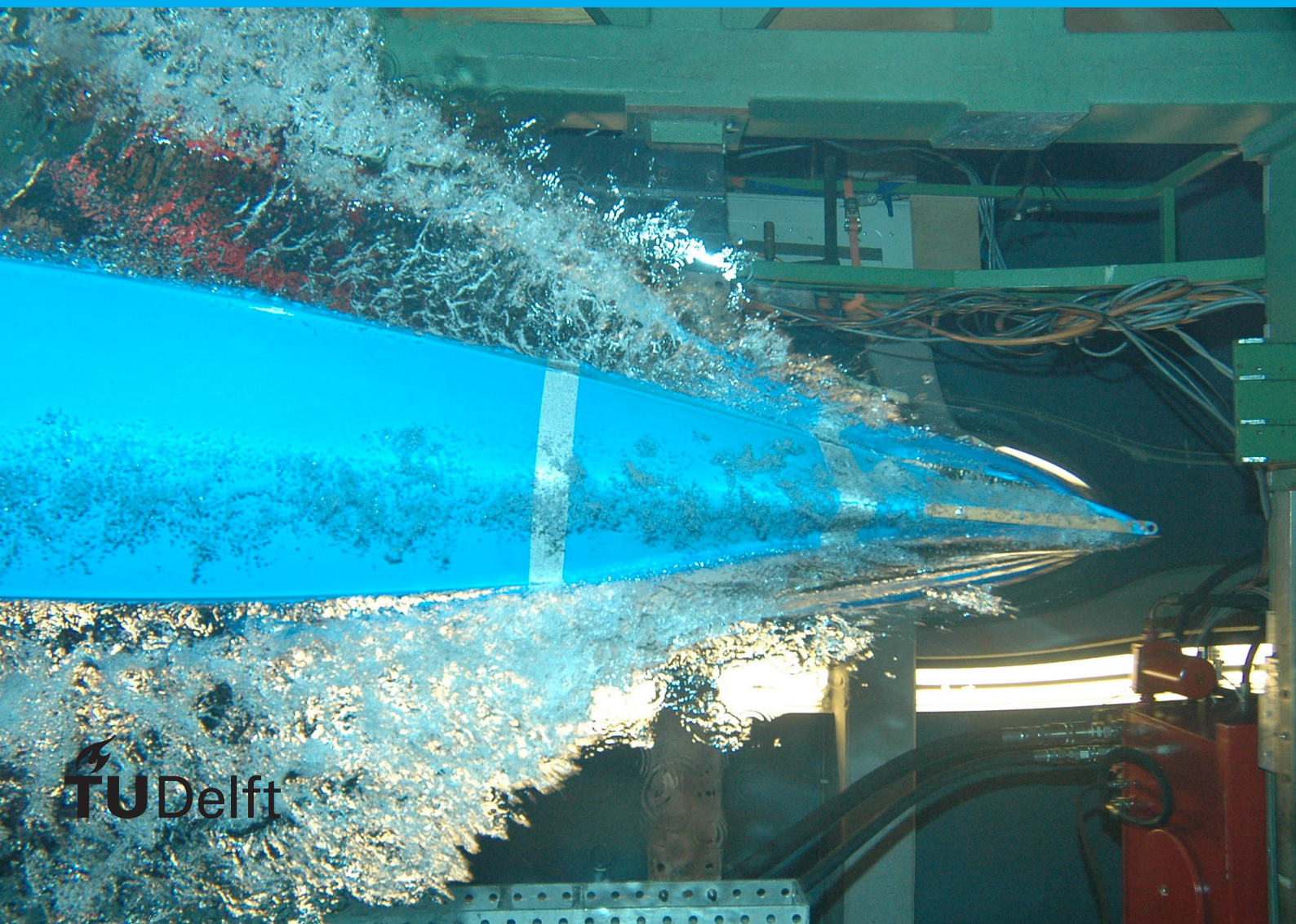
;

Optional subtitle

M. Meuleman

Cover Text
possibly
spanning multiple lines

ISBN 000-00-0000-000-0



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

Abstract

Preface

Preface...

M. Meuleman
Delft, January 2013

Contents

1 Introduction

1.1. Optical Music Recognition

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.

The field of Optical Music Recognition (OMR) works toward solving this problem by finding methods that can translate the scanned formats to a format that can provide semantic meaning of the music scores to computers, so that these aforementioned applications can become available to the written music domain. Translating text from an image to computer-readable text has long been solved in the field of Optical Character Recognition (OCR)[citation needed:]. Its musical counterpart however remains for a large part unsolved. A lot of this is due to the more complex structures we see in music scores. Where reading text is a single dimensional operation, classifying each character in a line of text, reading music is a two dimensional operation, where the dimensions are time on the horizontal axis and pitch on the vertical axis. 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 to improve readability for musicians. Two connected notes can also be translated relative to each other, meaning the connection can become stretched or tilted. Also there are symbol that are often small in comparison to the notes, but have significant meaning, such as dots, dashes and accents. Draw some examples All of these illustrated cases can be combined and many of these combinations occur frequently in music scores, making it impossible to apply simple heuristics when translating music scores to a computer-understandable format. Many approaches have been taken to apply machine learning and end-to-end learning [citation needed:]to this problem. There are promising results in there, but for a lot of use cases the translated music has to almost perfectly correspond to the original.

This is especially true for scores of classical music. Many classical scores are complex, and the scanned format of many of these scores are often hard to read because they are old editions or scanned in low quality. This, combined with the fact that there is a lot of classical music available in the public domain makes digitization in this computer-understandable format especially attractive for classical music.

1.2. TROMPA

TROMPA (Towards Richer Online Music Public-domain Archives) is an international organization of scientists and scholars that push towards this goal of making more applications available to the music domain and increase engagement with classical music and music scores[citation needed:]Some more info about TROMPA. A group of researchers at Delft University of Technology is part of this organization. This group focusses on translating the scans of music scores into the MEI (Music Encoding Initiative) format, an XML-like format for music scores that allows computers to give semantic meaning to these scores[citation needed:]. This process is far from perfect, and therefore research is currently ongoing in crowd computing solutions.

1.3. Crowd Sourced solutions

Maybe the crowd can help out a bit

1.3.1. Struggles for crowd sourcing

Repetition legitimises Repetition legitimises Repetition legitimises

But maybe see if we can do without transcribing the same measure 100 times

Also, if we could measure the complexity of a measure somewhat easily, this could help with prioritizing tasks

1.4. Research questions

Main research question

> How can we apply standard image processing methods to improve the process of transcribing orchestral music scores?

1.5. Main contributions

1.6. Outline

2

Related work

2.1. OMR

- OMR is an extension of OCR, but with many more complex problems. The 2 dimensional nature of OMR, in contrast to the 1 dimensional nature of OCR, can cause troubles in accurately transcribing music [citation needed:]. Besides this, there is a vast collection of symbols possible in sheet music, with a very uneven occurrence distribution, resulting in a large number of false positives for the symbols that occur infrequently (Chen, Stolterman, 2016). There is also a large portion of written music that adheres strongly to a fixed ruleset, advocating for a rule-based recognition system, which is offset against the small but long tail of exceptions on these rules, making any rule-based recognition system inaccurate in too many cases.
- Throughout the last two decades a generally accepted OMR pipeline has been devised, first started by Bellini et al. (2004??) and later refined by Rebelo et al. (2012). Various stages of this pipeline remain unsolved.
- To find solution for these open issues, two trends can be seen over the last decade or so: deep learning and human-aided recognition. (include lots of sources for both pls)
- In deep learning, promising results are shown for certain stagese of the pipeline (Gallego, Calvo-Zaragoza, 2017), (Pacha, Calvo-Zaragoza, Hajic, 2019) as well as for a full-pipeline end-to-end learning approach (Calvo-Zaragoza, Valer-Mas, 2017), (Wel, Ullrich, 2017). The main difficulty with the deep learning approach is the cost of collecting data. There is only a small collection of datasets available, getting more of these is expensive.
- Meanwhile human-aided solutions have also started to get some traction. Examples are the Allegro system (Burghardt, Spanner, (2017), human-in-the-loop systems on measure level (Chen, Stolterman, 2016) and on symbol level (Chen, Raphael, 2016)

2.2. Crowd computing

2.3. 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 [4]

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.

2.4. 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 [2] and its derivative, the MUSCIMA++ [3] for handwritten data, mainly aimed at staff line removal and symbol classification, and the MeasureBoundingBoxAnnotations dataset [4]. Both the MUSCIMA++ and version 2 of the MeasureBoundingBoxAnnotations datasets have annotations for bounding boxes of individual measures.

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 [1]).

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. [4]). An example of such a structure is given in Figure **??**, 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.

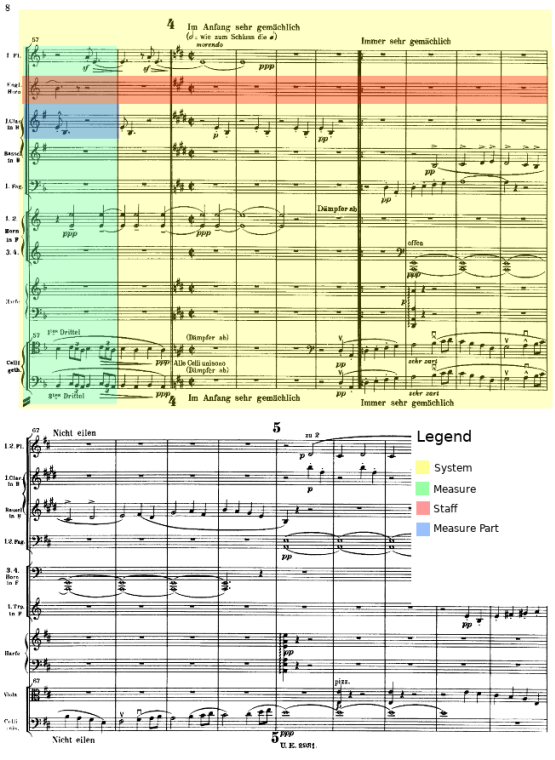


Figure 4.1: Example structure of a page of music

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.

Bibliography

[1] 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.

[2] 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. 15:243–251, 2012. doi: 10.1007/s10032-011-0168-2. URL http://www.cvc.uab.es/cvcmuscima.

[3] 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.

[4] 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.